aws

Panduan Developer

Amazon SageMaker

Amazon SageMaker: Panduan Developer

Table of Contents

Apa itu Amazon SageMaker?

Amazon SageMaker adalah layanan pembelajaran mesin (ML) yang dikelola sepenuhnya. Dengan SageMaker, para ilmuwan dan pengembang data dapat dengan cepat dan percaya diri membangun, melatih, dan menerapkan model ML ke dalam lingkungan host yang siap produksi. Ini memberikan pengalaman UI untuk menjalankan alur kerja ML yang membuat alat SageMaker ML tersedia di beberapa lingkungan pengembangan terintegrasi (IDE).

Dengan SageMaker, Anda dapat menyimpan dan berbagi data Anda tanpa harus membangun dan mengelola server Anda sendiri. Ini memberi Anda atau organisasi Anda lebih banyak waktu untuk secara kolaboratif membangun dan mengembangkan alur kerja ML Anda, dan melakukannya lebih cepat. SageMaker menyediakan algoritma ML terkelola untuk berjalan secara efisien terhadap data yang sangat besar dalam lingkungan terdistribusi. Dengan dukungan bring-your-own-algorithms dan kerangka kerja bawaan, SageMaker menawarkan opsi pelatihan terdistribusi fleksibel yang menyesuaikan dengan alur kerja spesifik Anda. Dalam beberapa langkah, Anda dapat menerapkan model ke lingkungan yang aman dan dapat diskalakan dari konsol. SageMaker

Topik

Harga untuk Amazon SageMaker

Untuk informasi tentang batas Tingkat AWS Gratis dan biaya penggunaan SageMaker, lihat SageMakerHarga Amazon.

Apakah Anda pengguna Amazon SageMaker pertama kali?

Jika Anda adalah pengguna pertama kali SageMaker, kami sarankan Anda menyelesaikan yang berikut ini:

1. Ikhtisar pembelajaran mesin dengan Amazon SageMaker— Dapatkan ikhtisar siklus hidup pembelajaran mesin (ML) dan pelajari tentang solusi yang ditawarkan. Halaman ini menjelaskan

konsep-konsep kunci dan menjelaskan komponen inti yang terlibat dalam membangun solusi AI dengan SageMaker.

2. Memulai— Pelajari cara mengatur dan menggunakan SageMaker berdasarkan kebutuhan Anda.
3. Gunakan HTML otomatis, tanpa kode, atau kode rendah— Pelajari tentang opsi HTML kode rendah dan tanpa kode yang menyederhanakan alur kerja ML dengan mengotomatiskan tugas pembelajaran mesin. Opsi ini adalah alat pembelajaran ML yang bermanfaat karena memberikan visibilitas ke dalam kode dengan membuat buku catatan untuk setiap tugas ML otomatis.
4. Gunakan lingkungan pembelajaran mesin yang ditawarkan oleh SageMaker— Biasakan diri Anda dengan lingkungan ML yang dapat Anda gunakan untuk mengembangkan alur kerja ML Anda, seperti informasi dan contoh tentang ready-to-use dan model kustom.
5. Jelajahi topik lainnya — Gunakan daftar isi Panduan SageMaker Pengembang untuk menjelajahi lebih banyak topik. Misalnya, Anda dapat menemukan informasi tentang tahapan siklus hidup ML, inIkhtisar pembelajaran mesin dengan Amazon SageMaker, dan berbagai solusi yang SageMaker ditawarkan.
6. SageMakerSumber daya Amazon — Lihat berbagai sumber daya pengembang yang SageMaker menawarkan.

Ikhtisar pembelajaran mesin dengan Amazon SageMaker

Bagian ini menjelaskan alur kerja machine learning (ML) yang khas dan merangkum cara menyelesaikan tugas-tugas tersebut dengan Amazon. SageMaker

Dalam pembelajaran mesin, Anda mengajarkan komputer untuk membuat prediksi atau kesimpulan. Pertama, Anda menggunakan algoritma dan contoh data untuk melatih model. Kemudian Anda mengintegrasikan model Anda ke dalam aplikasi Anda untuk menghasilkan kesimpulan secara real time dan dalam skala besar.

Diagram berikut menggambarkan alur kerja khas untuk membuat model pembelajaran mesin. Ini mencakup tiga tahap dalam aliran melingkar yang akan kita bahas secara lebih rinci di bawah ini: menghasilkan contoh data, melatih model, dan menyebarkan model.

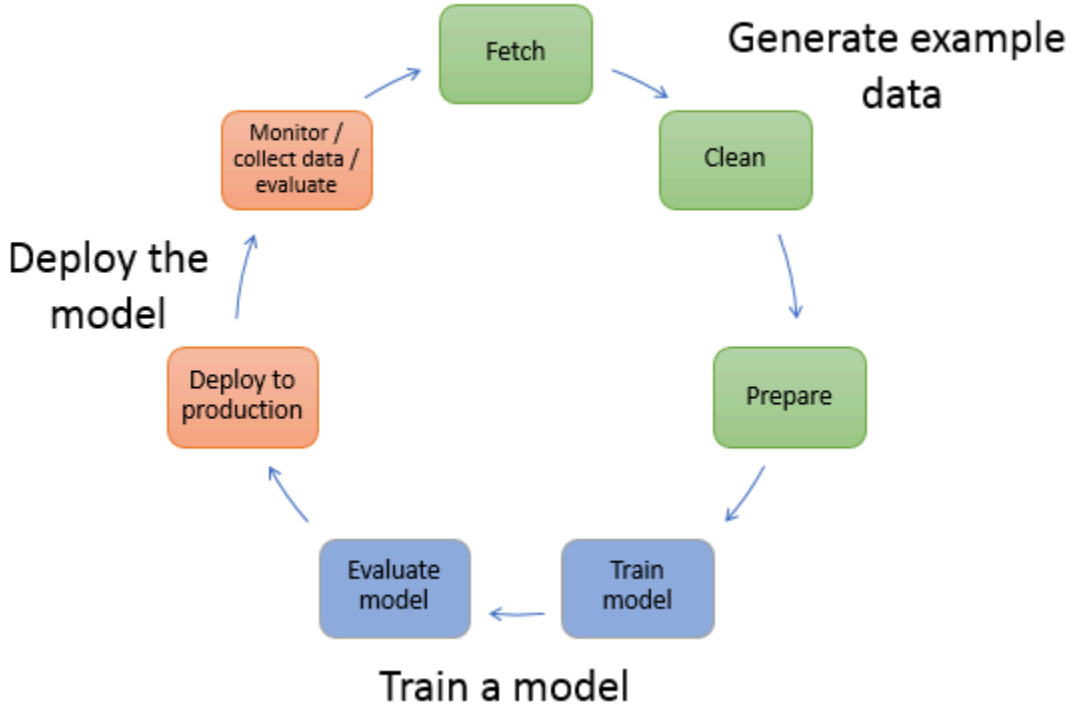


Diagram menggambarkan bagaimana melakukan kegiatan berikut dalam skenario yang paling umum:

1. Menghasilkan contoh data — Untuk melatih model, Anda memerlukan contoh data. Jenis data yang Anda butuhkan tergantung pada masalah bisnis yang Anda ingin model untuk memecahkan (kesimpulan yang Anda ingin model untuk menghasilkan). Misalnya, Anda ingin membuat model untuk memprediksi angka dari gambar input digit tulisan tangan. Untuk melatih model seperti itu, Anda memerlukan contoh gambar angka tulisan tangan.

   Ilmuwan data sering mencurahkan waktu untuk mengeksplorasi dan memproses data contoh sebelum menggunakannya untuk pelatihan model. Untuk memproses data sebelumnya, Anda biasanya melakukan hal berikut:

   a. Ambil data — Anda mungkin memiliki repositori data contoh internal, atau Anda mungkin menggunakan kumpulan data yang tersedia untuk umum. Biasanya, Anda menarik dataset atau dataset ke dalam satu repositori.

   b. Bersihkan data — Untuk meningkatkan pelatihan model, periksa data dan bersihkan, sesuai kebutuhan. Misalnya, jika data Anda memiliki `country name` atribut dengan nilai `United States` danUS, Anda dapat mengedit data agar konsisten.

c. Mempersiapkan atau mengubah data — Untuk meningkatkan kinerja, Anda dapat melakukan transformasi data tambahan. Misalnya, Anda dapat memilih untuk menggabungkan atribut. Jika model Anda memprediksi kondisi yang memerlukan de-icing pesawat terbang, alih-alih menggunakan atribut suhu dan kelembaban secara terpisah, Anda dapat menggabungkan atribut tersebut menjadi atribut baru untuk mendapatkan model yang lebih baik.

Di SageMaker, Anda dapat memproses data contoh menggunakan SageMaker API dengan SageMaker Python SDK di lingkungan pengembangan terintegrasi (IDE). Dengan SDK for Python (Boto3), Anda dapat mengambil, menjelajahi, dan menyiapkan data Anda untuk pelatihan model. Untuk informasi tentang persiapan data, pemrosesan, dan transformasi data Anda, lihat Mempersiapkan dataMemproses data, danBuat, simpan, dan bagikan fitur dengan Amazon SageMaker Feature Store.

2. Melatih model — Pelatihan model mencakup pelatihan dan evaluasi model, sebagai berikut:

   - Melatih model — Untuk melatih model, Anda memerlukan algoritme atau model dasar yang telah dilatih sebelumnya. Algoritma yang Anda pilih tergantung pada sejumlah faktor. Untuk solusi bawaan, Anda dapat menggunakan salah satu algoritma yang SageMaker menyediakan. Untuk daftar algoritma yang disediakan oleh SageMaker dan pertimbangan terkait, lihat. Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih Untuk solusi pelatihan berbasis UI yang menyediakan algoritma dan model, lihat. SageMaker JumpStart

     Anda juga membutuhkan sumber daya komputasi untuk pelatihan. Bergantung pada ukuran kumpulan data pelatihan dan seberapa cepat Anda membutuhkan hasilnya, Anda dapat menggunakan sumber daya mulai dari satu instans tujuan umum hingga kluster instans GPU terdistribusi. Untuk informasi selengkapnya, lihat Latih Model dengan Amazon SageMaker.

   - Mengevaluasi model — Setelah Anda melatih model Anda, Anda mengevaluasinya untuk menentukan apakah keakuratan kesimpulan dapat diterima. Untuk melatih dan mengevaluasi model Anda, Anda dapat menggunakan SageMaker Python SDK untuk mengirim permintaan ke model untuk inferensi melalui salah satu IDE yang tersedia. Untuk informasi selengkapnya tentang mengevaluasi model Anda, lihatMemantau data dan kualitas model.

3. Menyebarkan model — Anda secara tradisional merekayasa ulang model sebelum Anda mengintegrasikannya dengan aplikasi Anda dan menerapkannya. Dengan layanan SageMaker hosting, Anda dapat menerapkan model Anda secara independen, yang memisahkannya dari kode aplikasi Anda. Untuk informasi selengkapnya, lihat Menyebarkan model untuk inferensi.

Pembelajaran mesin adalah siklus yang berkelanjutan. Setelah menerapkan model, Anda memantau kesimpulan, mengumpulkan lebih banyak data berkualitas tinggi, dan mengevaluasi model untuk mengidentifikasi penyimpangan. Anda kemudian meningkatkan akurasi kesimpulan Anda dengan memperbarui data pelatihan Anda untuk memasukkan data berkualitas tinggi yang baru dikumpulkan. Saat lebih banyak contoh data tersedia, Anda terus melatih ulang model Anda untuk meningkatkan akurasi.

SageMaker Fitur Amazon

Amazon SageMaker menyertakan fitur-fitur berikut.

Topik

Fitur baru untuk RE: Invent 2023

SageMaker termasuk fitur baru berikut untuk re: Invent 2023.

SageMaker Obrolan kanvas untuk persiapan data

SageMaker Obrolan kanvas untuk persiapan data membantu Anda membuat alur persiapan data menggunakan LLM.

Editor Kode

Code Editor memperluas Studio sehingga Anda dapat menulis, menguji, men-debug, dan menjalankan analisis dan kode pembelajaran mesin Anda di lingkungan berdasarkan Visual Studio Code - Open Source (“Code-OSS”).

Wadah pembelajaran mendalam untuk inferensi model besar

SageMaker telah mengganti kernel NCCL default dengan kernel yang dioptimalkan inferensi untuk meningkatkan pemanfaatan GPU dan menawarkan kinerja yang membedakan terhadap OSS.

Terapkan model untuk inferensi waktu nyata

SageMaker Inferensi memberikan pengalaman pengembang dan abstraksi antarmuka pengguna untuk membantu Anda memulai lebih cepat dengan penerapan model.

SageMaker pelanggan sekarang dapat meningkatkan pemanfaatan instans komputasi mereka yang dipercepat dengan menerapkan hingga ribuan model ke SageMaker titik akhir dengan throughput terjamin dan auto-scaling berdasarkan per model.

SageMakerGambar Distribusi

SageMaker Distribusi adalah kumpulan gambar Docker yang dirancang untuk pembelajaran mesin, ilmu data, dan analisis data. Gambar tersedia di Studio, Studio Lab, notebook Studio, dan Github.

Penyederhanaan orientasi domain

Pengalaman orientasi SageMaker Domain Amazon yang disederhanakan dan dipandu dengan kemampuan baru untuk pengguna tunggal dan administrator organisasi. Kemampuannya mencakup integrasi IAM Identity Center langsung, manajemen kebijakan akses yang halus, manajemen dan konfigurasi SageMaker aplikasi yang mulus, serta konfigurasi VPC dan penyimpanan.

Amazon S3 Express Satu Zona

Amazon S3 Express One Zone adalah kelas penyimpanan baru yang menyediakan akses milidetik satu digit untuk aplikasi yang paling sensitif terhadap latensi. Amazon S3 Express One Zone memungkinkan pelanggan untuk mengumpulkan penyimpanan objek mereka dan menghitung sumber daya dalam satu AWS Availability Zone, mengoptimalkan kinerja komputasi dan biaya dengan peningkatan kecepatan pemrosesan data.

Evaluasi model pondasi (FMEval)

Evaluasi model Foundation (FMEval) membantu Anda mengukur risiko penyediaan konten yang tidak akurat, beracun, atau bias dengan model bahasa Anda sehingga Anda dapat memilih yang terbaik untuk kasus penggunaan Anda. Bawa dataset kustom Anda sendiri atau gunakan built-in untuk mengevaluasi model bahasa apa pun. FMEval terintegrasi dengan puluhan model fondasi berbasis teks SageMaker JumpStart atau membawa milik Anda sendiri. Anda juga dapat membuat evaluasi yang disesuaikan menggunakan pustaka FMEval.

SageMaker HyperPod

SageMaker HyperPod adalah kemampuan SageMaker yang menyediakan lingkungan pembelajaran mesin yang selalu aktif pada cluster tangguh sehingga Anda dapat menjalankan beban kerja pembelajaran mesin apa pun untuk mengembangkan model pembelajaran mesin besar seperti model bahasa besar (LLM) dan model difusi.

Jupyterai

Jupyter AI dan Code Whisperer telah dimasukkan ke Distribusi. SageMaker Dengan pembaruan ini, pengguna Studio atau Code Editor dapat dengan mudah menggunakan AI generatif dari notebook mereka dan memanfaatkan fitur penyelesaian kode Code Whisperer.

JupyterLab di Studio

JupyterLab di Studio meningkatkan latensi dan keandalan untuk Studio Notebook

SageMakerLowongan Notebook

SageMaker Pekerjaan Notebook menyediakan dukungan SDK untuk pekerjaan notebook sehingga Anda dapat menjadwalkan pekerjaan notebook Anda secara terprogram.

SageMaker Pipa

SageMaker Pipelines memberi Anda opsi untuk mengonversi kode pembelajaran mesin lokal Anda menjadi langkah SageMaker Pipeline, dari mana Anda dapat membuat dan menjalankan pipeline.

SageMakerpenyaringan cerdas

SageMaker smart sifting adalah kemampuan SageMaker Pelatihan yang meningkatkan efisiensi kumpulan data pelatihan Anda dan mengurangi total waktu dan biaya pelatihan.

SageMakerStudio

Studio adalah pengalaman berbasis web terbaru untuk menjalankan alur kerja ML. Studio menawarkan serangkaian IDE, termasuk Code Editor, aplikasi Jupyterlab baru, RStudio, dan Studio Classic.

Lingkungan pembelajaran mesin

SageMaker termasuk lingkungan pembelajaran mesin berikut.

SageMaker kemampuan geospasial

Bangun, latih, dan terapkan model ML menggunakan data geospasial.

SageMaker Kanvas

Layanan Auto ML yang memberikan orang-orang tanpa pengalaman coding kemampuan untuk membangun model dan membuat prediksi dengan mereka.

SageMaker Studio

Lingkungan pembelajaran mesin terintegrasi tempat Anda dapat membangun, melatih, menyebarkan, dan menganalisis semua model Anda dalam aplikasi yang sama.

SageMaker Studio Lab

Layanan gratis yang memberi pelanggan akses ke sumber daya AWS komputasi di lingkungan berdasarkan sumber terbuka JupyterLab.

RStudio di Amazon SageMaker

Lingkungan pengembangan terintegrasi untuk R, dengan konsol, editor penyorotan sintaks yang mendukung eksekusi kode langsung, dan alat untuk merencanakan, riwayat, debugging, dan manajemen ruang kerja.

Fitur utama

SageMaker termasuk fitur utama berikut dalam urutan abjad tidak termasuk awalan apa pun. SageMaker

AI Augmented AI Amazon

Bangun alur kerja yang diperlukan untuk tinjauan manusia terhadap prediksi ML. Amazon A2I membawa tinjauan manusia ke semua pengembang, menghilangkan beban berat yang tidak terdiferensiasi yang terkait dengan membangun sistem tinjauan manusia atau mengelola sejumlah besar pengulas manusia.

Langkah AutoML

Buat pekerjaan AutoML untuk melatih model secara otomatis di SageMaker Pipelines.

SageMaker Autopilot

Pengguna tanpa pengetahuan pembelajaran mesin dapat dengan cepat membangun model klasifikasi dan regresi.

Transformasi Batch

Praproses kumpulan data, jalankan inferensi saat Anda tidak memerlukan titik akhir yang persisten, dan kaitkan catatan input dengan kesimpulan untuk membantu interpretasi hasil.

SageMaker Klarifikasi

Tingkatkan model pembelajaran mesin Anda dengan mendeteksi potensi bias dan membantu menjelaskan prediksi yang dibuat model.

Kolaborasi dengan ruang bersama

Ruang bersama terdiri dari JupyterServer aplikasi bersama dan direktori bersama. Semua profil pengguna di SageMaker Domain Amazon memiliki akses ke semua ruang bersama di Domain.

SageMaker Data Wrangler

Impor, analisis, siapkan, dan featurisasi data di SageMaker Studio. Anda dapat mengintegrasikan Data Wrangler ke dalam alur kerja pembelajaran mesin Anda untuk menyederhanakan dan merampingkan pra-pemrosesan data dan rekayasa fitur menggunakan sedikit atau tanpa pengkodean. Anda juga dapat menambahkan skrip dan transformasi Python Anda sendiri untuk menyesuaikan alur kerja persiapan data Anda.

Widget persiapan data Wrangler Data

Berinteraksi dengan data Anda, dapatkan visualisasi, jelajahi wawasan yang dapat ditindaklanjuti, dan perbaiki masalah kualitas data.

SageMaker Debugger

Periksa parameter dan data pelatihan selama proses pelatihan. Secara otomatis mendeteksi dan memperingatkan pengguna untuk kesalahan yang umum terjadi seperti nilai parameter menjadi terlalu besar atau kecil.

SageMaker Manajer Tepi

Optimalkan model khusus untuk perangkat edge, buat dan kelola armada dan jalankan model dengan runtime yang efisien.

SageMaker Elastic Inference

Mempercepat throughput dan mengurangi latensi mendapatkan kesimpulan real-time.

SageMaker Eksperimen

Manajemen dan pelacakan eksperimen. Anda dapat menggunakan data yang dilacak untuk merekonstruksi eksperimen, secara bertahap membangun eksperimen yang dilakukan oleh rekan kerja, dan melacak garis keturunan model untuk kepatuhan dan verifikasi audit.

SageMaker Toko Fitur

Toko terpusat untuk fitur dan metadata terkait sehingga fitur dapat dengan mudah ditemukan dan digunakan kembali. Anda dapat membuat dua jenis toko, toko Online atau Offline. Toko Online dapat digunakan untuk latensi rendah, kasus penggunaan inferensi real-time dan Toko Offline dapat digunakan untuk pelatihan dan inferensi batch.

SageMaker Ground Truth

Kumpulan data pelatihan berkualitas tinggi dengan menggunakan pekerja bersama dengan pembelajaran mesin untuk membuat kumpulan data berlabel.

SageMaker Ground Truth Plus

Fitur pelabelan data turnkey untuk membuat kumpulan data pelatihan berkualitas tinggi tanpa harus membangun aplikasi pelabelan dan mengelola tenaga kerja pelabelan sendiri.

SageMaker Rekomendasi Inferensi

Dapatkan rekomendasi tentang jenis dan konfigurasi instans inferensi (misalnya jumlah instans, parameter kontainer, dan pengoptimalan model) untuk menggunakan model dan beban kerja MLmu.

Tes bayangan inferensi

Evaluasi setiap perubahan pada infrastruktur yang melayani model Anda dengan membandingkan kinerjanya dengan infrastruktur yang saat ini digunakan.

SageMaker JumpStart

Pelajari tentang SageMaker fitur dan kemampuan melalui solusi 1-klik yang dikurasi, contoh buku catatan, dan model terlatih yang dapat Anda terapkan. Anda juga dapat menyempurnakan model dan menerapkannya.

SageMaker Pelacakan Silsilah

Lacak garis keturunan alur kerja pembelajaran mesin.

SageMaker Pipa Bangunan Model

Membuat dan mengelola pipa pembelajaran mesin yang terintegrasi langsung dengan SageMaker pekerjaan.

SageMaker Kartu Model

Dokumentasikan informasi tentang model ML Anda di satu tempat untuk tata kelola dan pelaporan yang efisien di seluruh siklus hidup ML.

SageMaker Dasbor Model

Ikhtisar visual yang dibuat sebelumnya dari semua model di akun Anda. Dasbor Model mengintegrasikan informasi dari SageMaker Model Monitor, mengubah pekerjaan, titik akhir, pelacakan garis keturunan, CloudWatch sehingga Anda dapat mengakses informasi model tingkat tinggi dan melacak kinerja model dalam satu tampilan terpadu.

SageMaker Model Monitor

Memantau dan menganalisis model dalam produksi (titik akhir) untuk mendeteksi penyimpangan data dan penyimpangan dalam kualitas model.

SageMaker Registri Model

Pembuatan versi, pelacakan artefak dan garis keturunan, alur kerja persetujuan, dan dukungan lintas akun untuk penerapan model pembelajaran mesin Anda.

SageMaker Neo

Latih model pembelajaran mesin sekali, lalu jalankan di mana saja di cloud dan di tepi.

Alur Kerja Berbasis Notebook

Jalankan notebook SageMaker Studio Anda sebagai pekerjaan terjadwal yang tidak interaktif.

Pra-pemrosesan

Menganalisis dan memproses data sebelumnya, menangani rekayasa fitur, dan mengevaluasi model.

SageMaker Proyek

Buat solusi end-to-end ML dengan CI/CD dengan menggunakan SageMaker proyek.

Pembelajaran Penguatan

Maksimalkan imbalan jangka panjang yang diterima agen sebagai hasil dari tindakannya.

SageMaker Manajer Peran

Administrator dapat menentukan izin hak istimewa terkecil untuk aktivitas MLM umum menggunakan peran IAM berbasis persona kustom dan yang telah dikonfigurasi sebelumnya.

SageMaker Titik Akhir Tanpa Server

Opsi endpoint tanpa server untuk menghosting model ML Anda. Secara otomatis menskalakan kapasitas untuk melayani lalu lintas titik akhir Anda. Menghapus kebutuhan untuk memilih jenis instance atau mengelola kebijakan penskalaan pada titik akhir.

Ekstensi Studio Klasik Git

Ekstensi Git untuk memasukkan URL repositori Git, mengkloningnya ke lingkungan Anda, mendorong perubahan, dan melihat riwayat komit.

SageMaker Notebook Studio

Generasi SageMaker notebook berikutnya yang mencakup integrasi AWS IAM Identity Center (IAM Identity Center), waktu start-up yang cepat, dan berbagi sekali klik.

SageMaker Studio Notebook dan Amazon EMR

Temukan, sambungkan, buat, hentikan, dan kelola kluster EMR Amazon dengan mudah dalam konfigurasi akun tunggal dan lintas akun langsung dari Studio. SageMaker

SageMaker Kompiler Pelatihan

Latih model pembelajaran mendalam lebih cepat pada instans GPU yang dapat diskalakan yang dikelola oleh. SageMaker

Memulai

Ada banyak cara untuk menggunakan Amazon SageMaker dan fitur-fiturnya dengan berbagai tingkat akses. Di halaman ini, kami memberikan petunjuk tentang cara mengatur prasyarat dan pengguna dan organisasi onboard ke lingkungan pembelajaran mesin (ML) yang ditawarkan oleh. SageMaker Untuk daftar SageMaker fitur, lihat SageMaker Fitur Amazon.

> **Note**
>
> Jika Anda ingin menggunakan lingkungan Amazon SageMaker Studio Lab ML atau jika SageMaker Domain Amazon Akun AWS dan Amazon telah disiapkan untuk Anda, Anda tidak perlu mengikuti pengaturan ini. Untuk mulai menggunakan lingkungan SageMaker ML, cari lingkungan dalam panduan pengembang ini untuk memulai. Jika URL masuk diberikan kepada Anda, Anda dapat mengaturnya dengan mengikuti petunjuk diAkses Domain setelah onboarding.

Pengalaman SageMaker orientasi menyederhanakan cara pengguna tunggal dan administrator organisasi besar mengatur dan mengelola SageMaker akses mereka melalui Domain Amazon. SageMaker Sebelum menyiapkan, pertama-tama ikuti langkah-langkah Mengatur SageMaker Prasyarat Amazon untuk mendaftar AWS akun dan membuat pengguna administratif. Pengalaman orientasi menyediakan opsi berikut.

Siapkan untuk pengguna tunggal: Pengalaman satu klik yang ramah untuk pengguna tunggal. Ini membantu Anda mengatur Domain Anda dengan pengaturan default. Setelah memenuhi prasyarat, ikuti Mengatur SageMaker Prasyarat Amazon langkah-langkahnya. Cepat onboard ke Domain Amazon SageMaker

Siapkan untuk organisasi: step-by-step Panduan dengan bantuan kontekstual untuk administrator ML tingkat lanjut dan perusahaan. Ini membantu Anda mengatur Domain Anda dengan pengaturan khusus. Administrator organisasi dapat melakukan onboard pengguna atau grup mereka melalui otentikasi, menggunakan AWS Identity and Access Management (IAM) atauAWS IAM Identity Center, menentukan kebijakan akses tercakup bawah, mengelola SageMaker aplikasi dan konfigurasi, dan menyiapkan konfigurasi VPC. Setelah memenuhi prasyaratMengatur SageMaker Prasyarat Amazon, gunakan salah satu instruksi berikut.

- Untuk menyiapkan Domain kustom bagi pengguna yang menggunakan IAM, lihatOrientasi khusus ke SageMaker Domain Amazon menggunakan IAM.

- Untuk menyiapkan Domain kustom untuk grup yang menggunakan autentikasi di Pusat Identitas IAM, lihat. Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center

Setelah Domain diatur, pengguna administratif dapat Lihat dan Edit Domain dan pengguna dapat mengakses lingkungan ML menggunakan URL masuk atau membuka SageMaker konsol dan memilih lingkungan dari panel navigasi kiri.

Topik

- Mengatur SageMaker Prasyarat Amazon
- Ikhtisar SageMaker Domain Amazon
- Wilayah dan kuota yang didukung

Mengatur SageMaker Prasyarat Amazon

Pada bagian ini, Anda mendaftarkan AWS akun dan membuat pengguna admin AWS Identity and Access Management (IAM).

Jika Anda baru menggunakan SageMaker, kami menyarankan agar Anda membacaApa itu Amazon SageMaker?.

Topik

- Buat AWS Akun
- Buat Pengguna Administratif dan Grup
- AWS CLIPrasyarat

Buat AWS Akun

Pada bagian ini, Anda mendaftarkan akun AWS. Jika sudah memiliki akun AWS, Anda dapat melewati langkah ini.

Saat Anda mendaftar ke Amazon Web Services (AWS), AWS akun Anda secara otomatis terdaftar untuk semua AWS layanan, termasuk SageMaker. Anda hanya membayar biaya layanan yang Anda gunakan.

Untuk membuat akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.

2. Ikuti petunjuk secara online.

   Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

   Saat Anda mendaftar Akun AWS, Pengguna root akun AWS akan dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, tetapkan akses administratif ke pengguna administratif, dan hanya gunakan pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Tulis ID AWS akun Anda karena Anda membutuhkannya untuk tugas berikutnya.

Buat Pengguna Administratif dan Grup

Saat Anda membuat AWS akun, Anda mendapatkan identitas masuk tunggal yang memiliki akses penuh ke semua AWS layanan dan sumber daya di akun tersebut. Identitas ini disebut pengguna root akun AWS. Masuk ke AWS konsol menggunakan alamat email dan kata sandi yang Anda gunakan untuk membuat akun memberi Anda akses penuh ke semua sumber AWS daya di akun Anda.

Kami sangat menyarankan agar Anda tidak menggunakan akun pengguna root untuk tugas sehari-hari, bahkan tugas administratif. Sebaliknya, patuhi praktik terbaik Keamanan di IAM, dan buat pengguna administratif. Kemudian, kunci kredensial pengguna akar dengan aman dan gunakan kredensial itu untuk melakukan beberapa tugas manajemen akun dan layanan saja.

Untuk membuat pengguna administratif

1. Buat pengguna administratif di AWS akun Anda. Untuk petunjuk, lihat Membuat pengguna administratif di Panduan Pengguna IAM.

   > **Note**
   >
   > Kami berasumsi bahwa Anda menggunakan kredensi pengguna administrator untuk latihan dan prosedur dalam panduan ini. Jika Anda memilih untuk membuat dan menggunakan pengguna lain, berikan izin minimum kepada pengguna tersebut. Untuk informasi selengkapnya, lihat Mengautentikasi dengan Identitas.

2. Pastikan bahwa pengguna administrator Anda memiliki AmazonSageMakerFullAccesskebijakan, serta kebijakan dengan konten berikut yang diperlukan untuk membuat SageMaker domain.

Untuk informasi selengkapnya tentang cara membuat kebijakan IAM, lihat Membuat kebijakan IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:*"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:domain/*",
                "arn:aws:sagemaker:*:*:user-profile/*",
                "arn:aws:sagemaker:*:*:app/*",
                "arn:aws:sagemaker:*:*:flow-definition/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "servicecatalog:*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

AWS CLIPrasyarat

Prasyarat berikut diperlukan untuk onboard ke Domain menggunakan. AWS CLI

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensil, lihat Memahami dan mendapatkan kredensil Anda AWS.

Ikhtisar SageMaker Domain Amazon

Untuk memiliki akses ke sebagian besar SageMaker lingkungan dan sumber daya Amazon, Anda harus menyelesaikan proses orientasi SageMaker Domain Amazon menggunakan SageMaker konsol atau. AWS CLI Untuk panduan yang menjelaskan cara memulai penggunaan SageMaker berdasarkan cara Anda ingin mengakses SageMaker, dan jika perlu cara mengatur Domain, lihatMemulai.

Amazon SageMaker Domain terdiri atas beberapa hal berikut ini:

- Volume Amazon Elastic File System (Amazon EFS) terkait
- Daftar pengguna yang berwenang
- Berbagai konfigurasi keamanan, aplikasi, kebijakan, dan Amazon Virtual Private Cloud (Amazon VPC)

Saat melakukan orientasi, Anda dapat memilih untuk menggunakan AWS Identity and Access Management (IAM) atau AWS IAM Identity Center untuk metode otentikasi. Saat menggunakan autentikasi IAM, Anda dapat memilih prosedur Penyiapan untuk pengguna tunggal atau Pengaturan untuk organisasi. Penyiapan RStudio hanya tersedia saat menggunakan prosedur Pengaturan untuk organisasi.

> **Note**
>
> Jika Anda onboard menggunakan autentikasi IAM dan ingin beralih ke otentikasi menggunakan IAM Identity Center nanti, Anda harus menghapus Domain yang Anda buat. Kemudian, Anda perlu mengimpor ulang semua notebook dan data pengguna lain yang Anda buat secara manual. Untuk informasi selengkapnya, lihat Hapus SageMaker Domain Amazon.

Cara termudah untuk membuat SageMaker Domain Amazon adalah dengan mengikuti prosedur Pengaturan untuk pengguna tunggal dari SageMaker konsol. Pengaturan untuk pengguna tunggal menggunakan pengaturan default. Pengaturan ini termasuk notebook yang dapat dibagikan dan akses internet publik. Untuk informasi tentang pengaturan default, lihatPengaturan default.

Untuk kontrol lebih lanjut, termasuk opsi menggunakan otentikasi menggunakan IAM Identity Center dan RStudio, gunakan prosedur Pengaturan untuk organisasi.

Otentikasi menggunakan IAM Identity Center

Untuk menggunakan otentikasi menggunakan IAM Identity Center dengan Domain, Anda harus onboard ke organisasi di. AWS Organizations

> **Note**
>
> AWS OrganizationsAkun harus berada dalam AWS Wilayah yang sama dengan Domain.

Otentikasi menggunakan IAM Identity Center memberikan manfaat berikut dibandingkan autentikasi IAM:

- Anggota yang diberi akses ke Domain memiliki URL masuk unik yang langsung membuka Domain, dan mereka masuk dengan kredenal Pusat Identitas IAM mereka. Saat Anda menggunakan autentikasi IAM, Anda harus masuk melalui konsol. SageMaker

  Untuk informasi selengkapnya tentang cara mengakses Domain Anda dengan autentikasi IAM Identity Center, lihat. Akses Domain setelah onboarding
- Organizations mengelola anggotanya di IAM Identity Center, bukan Domain. Anda dapat menetapkan beberapa anggota akses ke Domain secara bersamaan. Saat Anda menggunakan autentikasi IAM, Anda harus menambahkan dan mengelola anggota secara manual, satu per satu, menggunakan panel kontrol Domain.

Topik

- SageMaker Domain Amazon
- Cepat onboard ke Domain Amazon SageMaker
- Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center
- Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM
- Pilih VPC Amazon

SageMaker Domain Amazon

Amazon SageMaker Domain mendukung lingkungan SageMaker machine learning (ML). Sebuah SageMaker Domain terdiri atas beberapa entitas berikut. Untuk langkah-langkah orientasi untuk membuat Domain, lihatIkhtisar SageMaker Domain Amazon.

- Domain Amazon SageMaker terdiri dari volume Amazon Elastic File System (Amazon EFS) terkait; daftar pengguna resmi; dan berbagai konfigurasi keamanan, aplikasi, kebijakan, dan Amazon Virtual Private Cloud (Amazon VPC). Pengguna dalam Domain dapat berbagi file buku catatan dan artefak lainnya satu sama lain. Suatu akun dapat memiliki beberapa Domain. Untuk informasi lebih lanjut tentang beberapa Domain, lihatGambaran Umum Beberapa Domain.
- UserProfile: Profil pengguna mewakili satu pengguna dalam Domain. Ini adalah cara utama untuk mereferensikan pengguna untuk tujuan berbagi, pelaporan, dan fitur berorientasi pengguna lainnya. Entitas ini dibuat saat pengguna melakukan onboard ke SageMaker Domain Amazon. Untuk informasi selengkapnya tentang profil pengguna, lihatProfil Pengguna Domain.
- Ruang bersama: Ruang bersama terdiri dari JupyterServer aplikasi bersama dan direktori bersama. Semua pengguna dalam Domain memiliki akses ke ruang bersama. Semua profil pengguna dalam Domain memiliki akses ke semua ruang bersama di Domain. Untuk informasi selengkapnya tentang spasi bersama, lihatBerkolaborasi dengan ruang bersama.
- Aplikasi: Aplikasi mewakili aplikasi yang mendukung pengalaman membaca dan mengeksekusi notebook, terminal, dan konsol pengguna. Jenis aplikasi bisa JupyterServer,, R KernelGatewayStudioServerPro, atau RSession. Seorang pengguna mungkin memiliki beberapa aplikasi aktif secara bersamaan.

Tabel berikut menjelaskan nilai status untuk`Domain`,, `UserProfile`s`hared space`, dan `App` entitas. Jika berlaku, mereka juga memberikan langkah pemecahan masalah.

Nilai Transformasi

Nilai	Deskripsi
Tertunda	Pembuatan Domain yang sedang berlangsung.
InService	Pembuatan Domain yang Sukses
Memperbarui	Pembaruan Domain yang sedang berlangsung.
Deleting	Penghapusan Domain yang sedang berlangsung.
Failed	Penciptaan Domain yang gagal. Panggil `DescribeDomain` API untuk melihat alasan kegagalan pembuatan Domain. Hapus Domain yang gagal dan buat ulang Domain setelah

Nilai	Deskripsi
	memperbaiki kesalahan yang disebutkan di`FailureReason` .
Update_Failed	Update Domain yang tidak berhasil. Panggil `DescribeDomain` API untuk melihat alasan kegagalan pembaruan Domain. Panggil `UpdateDomain` API setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .
Delete_Failed	Penghapusan Domain yang gagal. Panggil `DescribeDomain` API untuk melihat alasan kegagalan penghapusan Domain. Karena penghapusan gagal, Anda mungkin memiliki beberapa sumber daya yang masih berjalan, tetapi Anda tidak dapat menggunakan atau memperbarui Domain tersebut. Panggil `DeleteDomain` API lagi setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .

`UserProfile`nilai status

Nilai	Deskripsi
Tertunda	Penciptaan yang sedang berlangsung`UserProfile` .
InService	Penciptaan yang suksesUserProfile .
Memperbarui	Pembaruan yang sedang berlangsung dari`UserProfile` .
Deleting	Penghapusan yang sedang berlangsung dari. `UserProfile`

Nilai	Deskripsi
Failed	Penciptaan yang gagal. `UserProfile` Panggil `DescribeUserProfile` API untuk melihat alasan kegagalan `UserProfile` pembuatan. Hapus yang gagal `UserProfile` dan buat ulang setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .
Update_Failed	Pembaruan yang tidak berhasil dari. `UserProfile` Panggil `DescribeUserProfile` API untuk melihat alasan kegagalan `UserProfile` pembaruan. Panggil `UpdateUserProfile` API lagi setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .
Delete_Failed	Penghapusan yang tidak berhasil. `UserProfile` Panggil `DescribeUserProfile` API untuk melihat alasan kegagalan `UserProfile` penghapusan. Karena penghapusan gagal, Anda mungkin memiliki beberapa sumber daya yang masih berjalan, tetapi Anda tidak dapat menggunakan atau memperbarui file tersebut. `UserProfile` Panggil `DeleteUserProfile` API lagi setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .

nilai status spasi bersama

Nilai	Deskripsi
Tertunda	Penciptaan ruang bersama yang berkelanjutan.
InService	Penciptaan ruang bersama yang sukses.

Nilai	Deskripsi
Deleting	Penghapusan ruang bersama yang sedang berlangsung.
Failed	Penciptaan ruang bersama yang gagal. Panggil `DescribeSpace` API untuk melihat alasan kegagalan pembuatan ruang bersama. Hapus ruang bersama yang gagal dan buat ulang setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .
Update_Failed	Pembaruan ruang bersama yang tidak berhasil. Panggil `DescribeSpace` API untuk melihat alasan kegagalan pembaruan ruang bersama. Panggil `UpdateSpace` API lagi setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .
Delete_Failed	Penghapusan ruang bersama yang tidak berhasil. Panggil `DescribeSpace` API untuk melihat alasan kegagalan penghapusan ruang bersama. Karena penghapusan gagal, Anda mungkin memiliki beberapa sumber daya yang masih berjalan, tetapi Anda tidak dapat menggunakan atau memperbarui ruang bersama. Panggil `DeleteSpace` API lagi setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .
Dihapus	Penghapusan ruang bersama yang berhasil.

Appnilai status

Nilai	Deskripsi
Tertunda	Penciptaan yang sedang berlangsungApp.

Nilai	Deskripsi
InService	Penciptaan yang suksesApp.
Deleting	Penghapusan yang sedang berlangsung dari. App
Failed	Penciptaan yang gagal. App Panggil `DescribeApp` API untuk melihat alasan kegagalan App pembuatan. Panggil `CreateApp` API lagi setelah memperbaiki kesalahan yang disebutkan di`FailureReason` .
Dihapus	Penghapusan yang berhasil. App

Pemeliharaan aplikasi

Setidaknya sekali setiap 90 hari, SageMaker melakukan pembaruan keamanan dan kinerja ke perangkat lunak yang mendasarinya untuk aplikasi Amazon SageMaker Studio Classic JupyterServer dan KernelGateway, SageMaker Canvas, dan Amazon SageMaker Data Wrangler. Beberapa item pemeliharaan, seperti peningkatan sistem operasi, mengharuskan SageMaker aplikasi Anda offline untuk waktu yang singkat selama jendela pemeliharaan. Karena pemeliharaan ini membuat aplikasi offline, Anda tidak dapat melakukan operasi apa pun saat perangkat lunak yang mendasarinya sedang diperbarui. Saat aktivitas pemeliharaan sedang berlangsung, status aplikasi beralih dari InServiceke Pending. Ketika pemeliharaan selesai, status aplikasi beralih kembali ke InService. Jika tambalan gagal, maka status aplikasi menjadi Gagal. Jika aplikasi dalam keadaan Gagal, sebaiknya buat aplikasi baru dengan jenis yang sama. Untuk informasi tentang membuat aplikasi Studio Classic, lihatMatikan dan Perbarui Aplikasi SageMaker Studio Classic dan Studio Classic. Untuk informasi tentang membuat aplikasi SageMaker Canvas, lihatKelola aplikasi.

Untuk informasi lebih lanjut, hubungi https://aws.amazon.com/premiumsupport/.

Topik

- Prasyarat
- Gambaran Umum Beberapa Domain
- Isolasi sumber daya domain

- Mengatur Default untuk Domain
- Melampirkan Sistem File Kustom ke Domain atau Profil Pengguna
- Environment
- Lihat dan Edit Domain
- Hapus SageMaker Domain Amazon
- Profil Pengguna Domain
- Grup Pusat Identitas IAM dalam Domain

Prasyarat

Untuk menggunakan fitur yang tersedia di SageMaker Domain Amazon, Anda harus terlebih dahulu melakukan onboard ke Domain. Untuk informasi lebih lanjut, lihat Onboard ke SageMaker Domain Amazon.

Jika Anda berinteraksi dengan Domain Anda menggunakanAWS CLI, Anda juga harus menyelesaikan prasyarat berikut.

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensil, lihat Memahami dan mendapatkan kredensil Anda AWS.

Gambaran Umum Beberapa Domain

Amazon SageMaker mendukung pembuatan beberapa SageMaker Domain Amazon dalam satu Wilayah AWS untuk setiap akun. Domain Tambahan di Wilayah memiliki fitur dan kemampuan yang sama dengan Domain pertama di suatu Wilayah. Setiap Domain dapat memiliki pengaturan Domain yang berbeda. Profil pengguna yang sama tidak dapat ditambahkan ke beberapa Domain dalam satu Wilayah dalam akun yang sama. Untuk informasi selengkapnya tentang batas Domain, lihat SageMaker titik akhir dan kuota Amazon.

Topik

- Propagasi tanda otomatis
- Pemfilteran tampilan sumber daya domain
- Pengisian kembali tag Domain

Propagasi tanda otomatis

Secara default, SageMaker sumber daya apa pun yang mendukung penandaan dan dibuat dari dalam UI Studio Classic setelah 30/11/2022 secara otomatis ditandai dengan tag ARN Domain. Tag ARN Domain didasarkan pada ID Domain dari Domain tempat sumber daya dibuat. Daftar berikut menjelaskan satu-satunya SageMaker sumber daya yang tidak mendukung propagasi tag otomatis, serta panggilan API yang terpengaruh di mana tag tidak dikembalikan karena tidak disetel secara otomatis.

Anda juga dapat menggunakan tag ini untuk alokasi biaya menggunakanAWS Billing and Cost Management. Untuk informasi selengkapnya, lihat Menggunakan tag alokasi AWS biaya.

> **Note**
>
> Semua SageMaker `List` API tidak mendukung isolasi sumber daya berbasis tag. `default`Aplikasi, yang mengelola UI Studio, tidak diberi tag secara otomatis.

SageMaker sumber daya	Pilih IAM
ImageVersionArn	• describe-image-version • update-image-version • delete-image-version
ModelCardExportJobArn	describe-model-card-export-pekerjaan
ModelPackageArn	mendeskripsikan-tindakan

Pemfilteran tampilan sumber daya domain

Secara default, SageMaker memfilter sumber daya yang ditampilkan dalam Studio Classic di tingkat domain. SageMakermengimplementasikan penyaringan sumber daya di Studio Classic menggunakan `sagemaker:domain-arn` tag yang dilampirkan ke SageMaker sumber daya.

> **Note**
>
> Ini hanya berlaku untuk UI Studio Classic. SageMaker tidak mendukung pemfilteran sumber daya menggunakan secara AWS CLI default.

Menggunakan penyaringan sumber SageMaker daya ini, SageMaker hanya menampilkan sumber daya yang dibuat di domain, serta SageMaker sumber daya yang tidak memiliki `sagemaker:domain-arn` tag yang terkait dengannya. Sumber daya yang tidak ditandai ini dibuat di luar konteks domain atau dibuat sebelum 30/11/2022. Anda dapat menambahkan tag ke sumber daya yang tidak ditandai ini untuk penyaringan yang lebih baik dengan mengikuti langkah-langkahnya. Pengisian kembali tag Domain Sumber daya yang dibuat di domain lain secara otomatis disaring.

Semua sumber daya yang dibuat di ruang bersama secara otomatis difilter ke ruang itu.

Pengisian kembali tag Domain

Jika Anda telah membuat sumber daya di Domain sebelum 30/11/2022, sumber daya tersebut tidak secara otomatis ditandai dengan tag Nama Sumber Daya Amazon Domain (ARN).

Untuk secara akurat mengatribusikan sumber daya ke Domain masing-masing, Anda harus menambahkan tag Domain ke sumber daya yang ada menggunakanAWS CLI, sebagai berikut.

1. Petakan semua SageMaker sumber daya yang ada dan ARN masing-masing ke Domain yang ada di akun Anda.
2. Jalankan perintah berikut dari mesin lokal Anda untuk menandai sumber daya dengan ARN dari Domain masing-masing sumber daya. Ini harus diulang untuk setiap SageMaker sumber daya di akun Anda.

```
aws resourcegroupstaggingapi tag-resources \
    --resource-arn-list arn:aws:sagemaker:region:account-id:space/domain-id/space-name \
    --tags sagemaker:domain-arn=arn:aws:sagemaker:region:account-id:domain/domain-id
```

Isolasi sumber daya domain

Anda dapat mengisolasi sumber daya antara masing-masing Domain di akun dan Wilayah menggunakan kebijakan. AWS Identity and Access Management Dengan isolasi sumber SageMaker daya, sumber daya, seperti model, eksperimen, pekerjaan pelatihan, dan saluran pipa yang dibuat dalam satu Domain, tidak dapat diakses dari Domain lain. Topik berikut menunjukkan cara membuat kebijakan IAM baru yang membatasi akses ke sumber daya di Domain ke profil pengguna dengan tag Domain, serta cara melampirkan kebijakan ini ke peran eksekusi IAM Domain. Anda harus mengulangi proses ini untuk setiap Domain di akun Anda. Untuk informasi selengkapnya tentang tag Domain dan pengisian ulang tag ini, lihatGambaran Umum Beberapa Domain.

Konsol

Bagian berikut menunjukkan cara membuat kebijakan IAM baru yang membatasi akses ke sumber daya di Domain ke profil pengguna dengan tag Domain, serta cara melampirkan kebijakan ini ke peran eksekusi IAM Domain, dari konsol Amazon SageMaker .

1. Buat kebijakan IAM bernama `StudioDomainResourceIsolationPolicy-`*`domain-id`* dengan dokumen kebijakan JSON berikut dengan menyelesaikan langkah-langkah dalam Membuat kebijakan IAM (konsol).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAPIs",
            "Effect": "Allow",
            "Action": "sagemaker:Create*",
            "NotResource": [
                "arn:aws:sagemaker:*:*:domain/*",
                "arn:aws:sagemaker:*:*:user-profile/*",
                "arn:aws:sagemaker:*:*:space/*"
            ]
        },
        {
            "Sid": "ResourceAccessRequireDomainTag",
            "Effect": "Allow",
            "Action": [
                "sagemaker:Update*",
                "sagemaker:Delete*",
                "sagemaker:Describe*"
            ],
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/sagemaker:domain-arn": "domain-arn"
                }
            }
        },
        {
            "Sid": "AllowActionsThatDontSupportTagging",
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeImageVersion",
                "sagemaker:UpdateImageVersion",
                "sagemaker:DeleteImageVersion",
                "sagemaker:DescribeModelCardExportJob",
                "sagemaker:DescribeAction"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteDefaultApp",
            "Effect": "Allow",
            "Action": "sagemaker:DeleteApp",
            "Resource": "arn:aws:sagemaker:*:*:app/domain-id/*/jupyterserver/
default"
        }
    ]
}
```

2. Lampirkan `StudioDomainResourceIsolationPolicy-`*`domain-id`* kebijakan ke peran eksekusi Domain dengan menyelesaikan langkah-langkah dalam Memodifikasi peran (konsol).

AWS CLI

Bagian berikut menunjukkan cara membuat kebijakan IAM baru yang membatasi akses ke sumber daya di Domain ke profil pengguna dengan tag Domain, serta cara melampirkan kebijakan ini ke peran eksekusi Domain, dari. AWS CLI

1. Buat file `StudioDomainResourceIsolationPolicy-`*`domain-id`* dengan nama dengan konten berikut dari mesin lokal Anda.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAPIs",
            "Effect": "Allow",
            "Action": "sagemaker:Create*",
            "NotResource": [
                "arn:aws:sagemaker:*:*:domain/*",
                "arn:aws:sagemaker:*:*:user-profile/*",
                "arn:aws:sagemaker:*:*:space/*"
            ]
        },
        {
            "Sid": "ResourceAccessRequireDomainTag",
            "Effect": "Allow",
            "Action": [
                "sagemaker:Update*",
                "sagemaker:Delete*",
                "sagemaker:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/sagemaker:domain-arn": "domain-arn"
                }
            }
        },
        {
            "Sid": "AllowActionsThatDontSupportTagging",
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeImageVersion",
                "sagemaker:UpdateImageVersion",
                "sagemaker:DeleteImageVersion",
                "sagemaker:DescribeModelCardExportJob",
                "sagemaker:DescribeAction"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteDefaultApp",
            "Effect": "Allow",
            "Action": "sagemaker:DeleteApp",
```

```
            "Resource": "arn:aws:sagemaker:*:*:app/domain-id/*/jupyterserver/
default"
        }
    ]
}
```

2. Buat kebijakan IAM baru menggunakan `StudioDomainResourceIsolationPolicy-domain-id` file.

```
aws iam create-policy --policy-name StudioDomainResourceIsolationPolicy-domain-id
 --policy-document file://StudioDomainResourceIsolationPolicy-domain-id
```

3. Lampirkan kebijakan yang baru dibuat ke peran baru atau yang sudah ada yang digunakan sebagai peran eksekusi Domain.

```
aws iam attach-role-policy --policy-arn arn:aws:iam:account-
id:policy/StudioDomainResourceIsolationPolicy-domain-id --role-name domain-
execution-role
```

Mengatur Default untuk Domain

Dengan SageMaker, Anda dapat mengatur pengaturan default untuk sumber daya Anda di tingkat SageMaker Domain Amazon. Pengaturan default ini digunakan dalam pembuatan sumber daya dalam Domain. Bagian berikut mencantumkan pengaturan default untuk Domain dan memberikan informasi tentang penggunaan kunci konteks saat mengatur default.

Topik

- Pengaturan default domain
- Kunci konteks

Pengaturan default domain

Anda dapat mengatur default berikut saat membuat atau memperbarui Domain. Nilai yang diteruskan pada profil pengguna dan tingkat ruang bersama mengganti default yang ditetapkan pada tingkat Domain.

- DefaultUserSettings
- DefaultSpaceSettings

> **Note**
>
> `DefaultSpaceSettings`shanya mendukung penggunaan JupyterLab 3 ARN gambar untuk`SageMakerImageArn`. Untuk informasi selengkapnya, lihat JupyterLab Pembuatan Versi.

```
"DefaultSpaceSettings": {
      "ExecutionRole": "string",
      "JupyterServerAppSettings": {
         "DefaultResourceSpec": {
            "InstanceType": "string",
            "LifecycleConfigArn": "string",
            "SageMakerImageArn": "string",
            "SageMakerImageVersionArn": "string"
         },
         "LifecycleConfigArns": [ "string" ]
      },
      "KernelGatewayAppSettings": {
         "CustomImages": [
            {
               "AppImageConfigName": "string",
               "ImageName": "string",
               "ImageVersionNumber": number
            }
         ],
         "DefaultResourceSpec": {
            "InstanceType": "string",
            "LifecycleConfigArn": "string",
            "SageMakerImageArn": "string",
            "SageMakerImageVersionArn": "string"
         },
         "LifecycleConfigArns": [ "string" ]
      },
      "SecurityGroups": [ "string" ]
   }
```

Kunci konteks

Anda dapat menambahkan kunci konteks ke kebijakan IAM yang membuat Domain. Ini membatasi nilai yang dapat dilewati pengguna untuk bidang tersebut. Daftar berikut menunjukkan kunci konteks yang didukung Domain dan di mana mereka diimplementasikan.

- `sagemaker:ImageArns`
  - Diimplementasikan sebagai bagian dari**`DefaultUserSettings`**: `SagemakerImageArn` di `DefaultUserSettings.JupyterServerAppSettings` dan`DefaultUserSettings.KernelGatewayAppSettings`. `CustomImages`di`DefaultUserSettings.KernelGatewayAppSettings`.
  - Diimplementasikan sebagai bagian dari**`DefaultSpaceSettings`**: `SagemakerImageArn` di `DefaultSpaceSettings.JupyterServerAppSettings` dan`DefaultSpaceSettings.KernelGatewayAppSettings`. `CustomImages`di`DefaultSpaceSettings.KernelGatewayAppSettings`.
- `sagemaker:VpcSecurityGroupIds`
  - Diimplementasikan sebagai bagian dari**`DefaultUserSettings`**: `SecurityGroups` in`DefaultUserSettings`.
  - Diimplementasikan sebagai bagian dari**`DefaultSpaceSettings`**: `SecurityGroups` in`DefaultSpaceSettings`.
- `sagemaker:DomainSharingOutputKmsKey`

  Diimplementasikan sebagai bagian dari**`DefaultUserSettings`**: `S3KmsKeyId` in`DefaultSpaceSettings.SharingSettings`.

Anda tidak dapat membatasi pengguna untuk meneruskan nilai yang tidak kompatibel saat menggunakan kunci konteks untuk default. Misalnya, nilai untuk `SageMakerImageArn` ditetapkan sebagai bagian dari `DefaultUserSettings` dan `DefaultSpaceSettings` harus kompatibel. Anda tidak dapat mengatur nilai default yang tidak kompatibel berikut ini. Untuk informasi selengkapnya tentang ARN JupyterLab versi yang tersedia, lihatMenyetel JupyterLab versi default.

- Hanya ARN JupyterLab versi 1 yang dapat digunakan untuk nilai di `SageMakerImageArn` `DefaultUserSettings`
- Hanya ARN JupyterLab versi 3 yang dapat digunakan untuk nilai di `SageMakerImageArn` `DefaultSpaceSettings`

Melampirkan Sistem File Kustom ke Domain atau Profil Pengguna

Saat Anda membuat Domain, Amazon SageMaker secara otomatis mengaitkannya dengan volume Amazon Elastic File System (Amazon EFS) yang SageMaker dibuat untuk Anda. Anda juga memiliki opsi untuk mengaitkan Domain dengan sistem file Amazon EFS kustom yang telah Anda buatAkun AWS. Sistem file ini tersedia untuk semua pengguna yang termasuk dalam Domain ketika mereka menggunakan Amazon SageMaker Studio. Pengguna dapat melampirkan sistem file ke ruang apa pun yang mereka buat untuk aplikasi yang didukung: JupyterLab dan Editor Kode. Kemudian, setelah menjalankan spasi dan memulai aplikasi, mereka dapat mengakses data, kode, atau artefak lain yang berisi sistem file.

Jika Anda tidak ingin mengizinkan semua pengguna untuk Domain untuk mengakses sistem file, Anda dapat melampirkannya ke profil pengguna tertentu sebagai gantinya. Jika Anda melakukannya, sistem file hanya tersedia di spasi yang dibuat pengguna terkait.

Anda dapat melampirkan sistem file kustom dengan menggunakan Amazon SageMaker API, AWS SDK, atau file. AWS CLI Anda tidak dapat melampirkan sistem file kustom dengan menggunakan SageMaker konsol.

Prasyarat

Sebelum Anda dapat melampirkan sistem file Amazon EFS kustom ke Domain, Anda harus memenuhi persyaratan berikut:

- Anda memiliki sistem file Amazon EFS diAkun AWS. Untuk langkah-langkah membuatnya, lihat Membuat sistem file Amazon EFS Anda di Panduan Pengguna Amazon Elastic File System.
- Sebelum Studio dapat mengakses sistem file Anda, Studio harus memiliki target mount di setiap subnet yang Anda kaitkan dengan Domain. Untuk informasi selengkapnya tentang menetapkan target mount ke subnet, lihat Membuat dan mengelola target mount dan grup keamanan di Panduan Pengguna Amazon Elastic File System.
- Untuk setiap target pemasangan, Anda harus menambahkan grup keamanan yang SageMaker dibuat Amazon di Akun AWS saat Anda membuat Domain. Nama grup keamanan memiliki format`security-group-for-inbound-nfs-`*`domain-id`*.
- Izin IAM Anda harus memungkinkan Anda untuk menggunakan tindakan. `elasticfilesystem:DescribeMountTargets` Untuk informasi selengkapnya tentang tindakan ini, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon Elastic File System di Referensi Otorisasi Layanan.

Melampirkan sistem file kustom dengan AWS CLI

Untuk melampirkan sistem file kustom ke Domain atau profil pengguna denganAWS CLI, Anda meneruskan `CustomFileSystemConfigs` definisi ketika Anda menggunakan salah satu perintah berikut:

- buat-domain
- pembaruan-domain
- create-user-profile
- update-user-profile

Example perintah buat-domain dengan sistem file kustom

Contoh berikut melampirkan sistem file ke Domain baru.

```
aws sagemaker create-domain --domain-name domain-name \
--vpc-id vpc-id --subnet-ids subnet-ids --auth-mode IAM \
--default-user-settings file://default-user-settings.json \
--default-space-settings "ExecutionRole=execution-role-arn"
```

Dalam contoh ini, file `default-user-settings.json` memiliki pengaturan berikut, yang mencakup `CustomFileSystemConfigs` tombol `CustomPosixUserConfig` dan.

```
{
    "ExecutionRole": "execution-role-arn",
    "CustomPosixUserConfig":
    {
        "Uid": UID,
        "Gid": GID
    },
    "CustomFileSystemConfigs":
    [
        {
            "EFSFileSystemConfig":
            {
                "FileSystemId": "file-system-id",
                "FileSystemPath": "/"
            }
        }
    ]
```

```
}
```

Konfigurasi contoh ini memiliki kunci-kunci berikut:

- `ExecutionRole`— Peran eksekusi default untuk pengguna Domain.
- `CustomPosixUserConfig`— Identitas POSIX default yang digunakan untuk operasi sistem file. Anda dapat menggunakan pengaturan ini untuk menerapkan struktur izin POSIX yang ada ke profil pengguna yang mengakses sistem file kustom. Pada tingkat izin POSIX, Anda dapat mengontrol pengguna mana yang dapat mengakses sistem file dan file atau data mana yang dapat mereka akses.

  Anda juga dapat menerapkan `CustomPosixUserConfig` pengaturan ke profil pengguna saat Anda menggunakan `update-user-profile` perintah `create-user-profile` atau. Pengaturan yang Anda terapkan pada profil pengguna akan menggantikan pengaturan yang Anda terapkan pada Domain terkait.
- `Uid`— ID pengguna POSIX. Default-nya adalah 200001.
- `Gid`— ID grup POSIX. Default-nya adalah 1.
- `CustomFileSystemConfigs`— Pengaturan untuk sistem file khusus (hanya sistem file Amazon EFS yang didukung).

  Anda juga dapat menerapkan `CustomFileSystemConfigs` pengaturan ke profil pengguna saat Anda menggunakan `update-user-profile` perintah `create-user-profile` atau. Profil pengguna akan memiliki akses ke sistem file tersebut serta apa pun yang Anda lampirkan ke Domain mereka.
- `EFSFileSystemConfig`- Pengaturan untuk sistem file Amazon EFS khusus.
- `FileSystemId`— ID sistem file Amazon EFS Anda.
- `FileSystemPath`— Jalur ke direktori sistem file yang dapat diakses oleh pengguna Domain di ruang mereka di Studio. Pengguna yang diizinkan hanya dapat mengakses direktori ini dan di bawahnya. Jalur default adalah root sistem file:`/`.

SageMaker membuat tautan simbolis di jalur berikut:. `/home/sagemaker-user/custom-file-systems/`*`file-system-type`*`/`*`file-system-id`* Dengan ini, pengguna Domain dapat menavigasi ke sistem file kustom dari dalam direktori home mereka`/home/sagemaker-user`.

Setelah Anda melampirkan sistem file kustom ke Domain, pengguna Domain dapat melampirkan sistem file ke spasi ketika mereka menggunakan perintah create-space.

Example perintah create-space dengan sistem file kustom

Contoh berikut melampirkan sistem file ke spasi baru.

```
aws sagemaker create-space \
--space-name space-name \
--domain-id domain-id \
--ownership-settings "OwnerUserProfileName=user-profile-name" \
--space-sharing-settings "SharingType=Private" \
--space-settings file://space-settings.json
```

Dalam contoh ini, file `space-settings.json` memiliki pengaturan berikut, yang mencakup `CustomFileSystems` konfigurasi dengan `FileSystemId` kunci.

```
{
    "AppType": "JupyterLab",
    "JupyterLabAppSettings":
    {
        "DefaultResourceSpec":
        {
            "InstanceType": "ml.t3.xlarge"
        }
    },
    "CustomFileSystems":
    [
        {
            "EFSFileSystem":
            {
                "FileSystemId": "file-system-id"
            }
        }
    ]
}
```

Environment

Halaman ini memberikan informasi tentang modifikasi pada lingkungan SageMaker Domain Amazon. Ini termasuk gambar kustom, konfigurasi siklus hidup, dan repositori git yang dilampirkan ke lingkungan Domain. Ini juga dapat dilampirkan ke ruang bersama menggunakan AWS CLI dengan meneruskan nilai ke perintah create-space menggunakan parameter. `space-settings`

Untuk informasi selengkapnya tentang membawa gambar Amazon SageMaker Studio Classic kustom, lihat Membawa SageMaker gambar Anda sendiri.

Untuk informasi selengkapnya tentang membawa gambar RStudio kustom, lihat Membawa gambar Anda sendiri ke RStudio. SageMaker

Untuk petunjuk tentang penggunaan konfigurasi siklus hidup dengan Studio Classic, lihat Menggunakan Konfigurasi Siklus Hidup dengan Amazon Studio. SageMaker

Untuk informasi tentang melampirkan repositori git ke Domain, lihat Melampirkan Repos Git yang Disarankan ke. SageMaker

Selesaikan prosedur berikut ini untuk melihat gambar kustom, konfigurasi siklus hidup, dan repositori git yang dilampirkan ke lingkungan Domain.

Buka halaman Lingkungan

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain untuk membuka halaman Lingkungan.
5. Pada halaman Detail domain, pilih tab Lingkungan.

Lihat dan Edit Domain

Topik ini menunjukkan cara melihat daftar Domain Amazon Anda, melihat detail SageMaker Domain, dan mengedit setelan Domain dari SageMaker konsol Amazon atau AWS Command Line Interface (AWS CLI).

Topik

- Lihat Domain
- Edit pengaturan Domain

Lihat Domain

Bagian berikut menunjukkan cara melihat daftar Domain Anda, dan detail masing-masing Domain dari SageMaker konsol atau. AWS CLI

Konsol

Halaman ikhtisar Domain konsol memberikan informasi tentang struktur Domain, dan menyediakan daftar Domain Anda. Diagram struktur Domain halaman menggambarkan komponen Domain dan bagaimana mereka berinteraksi satu sama lain.

Prosedur berikut menunjukkan cara melihat daftar Domain Anda dari SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.

Untuk melihat detail Domain, selesaikan prosedur berikut. Halaman ini memberikan informasi tentang pengaturan umum untuk Domain, termasuk nama, ID Domain, peran eksekusi yang digunakan untuk membuat Domain, dan metode otentikasi Domain.

1. Dari daftar Domain, pilih Domain yang ingin Anda buka halaman Pengaturan domain.
2. Pada halaman Detail domain, pilih tab Pengaturan domain.

AWS CLI

Jalankan perintah berikut dari terminal mesin lokal Anda untuk melihat daftar Domain dariAWS CLI.

```
aws sagemaker list-domains --region region
```

Edit pengaturan Domain

Anda dapat mengedit pengaturan Domain dari SageMaker konsol atauAWS CLI. Pertimbangan berikut berlaku saat memperbarui pengaturan Domain.

- Jika `DefaultUserSettings` dan `DefaultSpaceSettings` disetel, mereka tidak dapat di-unset.
- `DefaultUserSettings.ExecutionRole`hanya dapat diperbarui jika tidak ada aplikasi yang berjalan di profil pengguna mana pun di dalam Domain. Nilai ini tidak dapat di-unset.
- `DefaultSpaceSettings.ExecutionRole`Hanya dapat diperbarui jika tidak ada aplikasi yang berjalan di salah satu ruang bersama dalam Domain. Nilai ini tidak dapat di-unset.

- Jika Domain dibuat dalam mode VPC saja, SageMaker secara otomatis menerapkan pembaruan ke pengaturan grup keamanan yang ditentukan untuk Domain ke semua ruang bersama yang dibuat di Domain.
- `DomainId`tidak dapat diperbarui.

Bagian berikut menunjukkan cara mengedit pengaturan Domain dari SageMaker konsol atauAWS CLI.

Konsol

Anda dapat mengedit Domain dari SageMaker konsol menggunakan prosedur berikut ini.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda buka halaman Pengaturan domain.
5. Pada halaman Detail Domain, Anda dapat mengonfigurasi dan mengelola detail Domain Anda dengan memilih tab yang sesuai.
6. Untuk mengonfigurasi pengaturan umum, pada halaman Detail domain pilih tab Pengaturan domain lalu pilih Edit.

AWS CLI

Jalankan perintah berikut dari terminal mesin lokal Anda untuk memperbarui Domain dari fileAWS CLI. Untuk informasi selengkapnya tentang strukturdefault-user-settings, lihat CreateDomain.

```
aws sagemaker update-domain \
--domain-id domain-id \
--default-user-settings default-user-settings \
--default-space-settings default-space-settings \
--domain-settings-for-update settings-for-update \
--region region
```

Hapus SageMaker Domain Amazon

Domain terdiri dari daftar pengguna yang berwenang, pengaturan konfigurasi, dan volume Amazon Elastic File System (Amazon EFS). Volume Amazon EFS berisi data untuk pengguna, termasuk

notebook, sumber daya, dan artefak. Pengguna dapat memiliki beberapa aplikasi (aplikasi) yang mendukung pengalaman membaca dan mengeksekusi notebook, terminal, dan konsol pengguna.

Anda dapat menghapus Domain Anda menggunakan salah satu dari berikut ini:

- Konsol AWS
- AWS Command Line Interface (AWS CLI)
- SageMaker SDK

Bagian berikut menjelaskan cara menghapus Domain dan persyaratan untuk melakukannya.

Persyaratan

Anda harus memenuhi persyaratan berikut untuk menghapus Domain.

- Anda harus memiliki izin admin untuk menghapus Domain.
- Anda hanya dapat menghapus aplikasi dengan status yang `InService` ditampilkan sebagai Siap di Domain. Untuk menghapus Domain yang berisi, Anda tidak perlu menghapus aplikasi yang statusnya`Failed`. Di Domain, upaya untuk menghapus aplikasi dalam status gagal menghasilkan kesalahan.
- Untuk menghapus Domain, Domain tidak dapat berisi profil pengguna atau spasi bersama. Untuk menghapus profil pengguna atau ruang bersama, profil atau ruang pengguna tidak dapat berisi aplikasi yang tidak gagal.

  Ketika Anda menghapus sumber daya ini, hal berikut ini terjadi:

  - Aplikasi — Data (file dan notebook) di direktori home pengguna disimpan. Data notebook yang belum disimpan hilang.
  - Profil pengguna — Pengguna tidak bisa lagi masuk ke Domain. Pengguna kehilangan akses ke direktori home mereka, tetapi data tidak dihapus. Admin dapat mengambil data dari volume Amazon EFS yang disimpan di bawah volume pengguna. Akun AWS
- Untuk mengalihkan mode otentikasi dari IAM ke IAM Identity Center, Anda harus menghapus Domain.

Fungsi

File Anda disimpan dalam volume Amazon EFS sebagai cadangan. Cadangan ini mencakup file di direktori yang dipasang, yang `/home/sagemaker-user` untuk Amazon SageMaker Studio Classic dan `/root` untuk kernel.

Saat Anda menghapus file dari direktori yang dipasang ini, kernel atau aplikasi dapat memindahkan file yang dihapus ke folder sampah tersembunyi. Jika folder sampah berada di dalam direktori yang dipasang, file-file tersebut disalin ke volume Amazon EFS dan akan dikenakan biaya. Untuk menghindari biaya Amazon EFS ini, Anda harus mengidentifikasi dan membersihkan lokasi folder sampah. Lokasi folder sampah untuk aplikasi dan kernel default adalah. `~/.local/` Ini dapat bervariasi tergantung pada distribusi Linux yang digunakan untuk aplikasi atau kernel khusus. Untuk informasi lebih lanjut tentang volume Amazon EFS, lihatKelola Volume Penyimpanan Amazon EFS Anda di SageMaker Studio Classic.

Saat Anda menggunakan SageMaker konsol untuk menghapus Domain, volume Amazon EFS terlepas tetapi tidak dihapus. Perilaku yang sama terjadi secara default ketika Anda menggunakan AWS CLI atau SageMaker Python SDK untuk menghapus Domain. Namun, ketika Anda menggunakan AWS CLI atau SageMaker Python SDK, Anda dapat mengatur `RetentionPolicy HomeEfsFileSystem=Delete` to untuk menghapus volume Amazon EFS bersama dengan Domain.

Menghapus SageMaker Domain Amazon (konsol)

Cara menghapus domain

1. Buka konsol SageMaker.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih Domain yang ingin Anda hapus.
5. Ulangi langkah-langkah berikut untuk setiap pengguna dalam daftar Profil Pengguna.

    a. Pilih pengguna.

    b. Pada halaman Detail Pengguna, untuk setiap aplikasi yang tidak gagal dalam daftar Aplikasi, pilih Tindakan.

    c. Dari daftar dropdown, pilih Delete.

    d. Pada kotak dialog Hapus aplikasi, pilih Ya, hapus aplikasi. Kemudian masukkan hapus di bidang konfirmasi, dan pilih Hapus.

e. Saat Status ditampilkan sebagai Dihapus untuk semua aplikasi, pilih Edit.

f. Pada halaman Edit Pengguna, pilih Hapus pengguna.

g. Pada kotak dialog Hapus pengguna, pilih Ya, hapus pengguna. Kemudian masukkan hapus di bidang konfirmasi, dan pilih Hapus.

> **Important**
>
> Ketika pengguna dihapus, mereka kehilangan akses ke volume Amazon EFS yang berisi data mereka, termasuk notebook dan artefak lainnya. Data tidak dihapus dan dapat diakses oleh administrator.

6. Ketika semua pengguna dihapus, pilih tab Manajemen ruang.
7. Ulangi langkah-langkah berikut untuk setiap ruang bersama dalam daftar Spasi.

   a. Pilih nama ruang bersama.

   b. Pilih Hapus aplikasi untuk setiap aplikasi.

   c. Pada kotak dialog Hapus aplikasi, pilih Ya, hapus aplikasi. Kemudian masukkan hapus di bidang konfirmasi, dan pilih Hapus.

   d. Pilih Batalkan.

   e. Pilih ruang bersama.

   f. Pilih Hapus.

   g. Pada kotak dialog Hapus ruang, pilih Ya, hapus spasi. Kemudian masukkan hapus di bidang konfirmasi, dan pilih Hapus spasi.
8. Ketika semua pengguna dan spasi bersama dihapus, pilih tab Pengaturan domain.
9. Pilih Edit.
10. Pada halaman Pengaturan umum, pilih Hapus Domain.
11. Pada kotak dialog Hapus Domain, pilih Ya, hapus Domain. Kemudian masukkan hapus di bidang konfirmasi, dan pilih Hapus.

Menghapus SageMaker Domain Amazon (AWS CLI)

Cara menghapus domain

1. Ambil daftar Domain di akun Anda.

```
aws --region Region sagemaker list-domains
```

2. Ambil daftar aplikasi untuk Domain yang akan dihapus.

```
aws --region Region sagemaker list-apps \
    --domain-id-equals DomainId
```

3. Hapus setiap aplikasi dalam daftar.

```
aws --region Region sagemaker delete-app \
    --domain-id DomainId \
    --app-name AppName \
    --app-type AppType \
    --user-profile-name UserProfileName
```

4. Ambil daftar profil pengguna di Domain.

```
aws --region Region sagemaker list-user-profiles \
    --domain-id-equals DomainId
```

5. Hapus setiap profil pengguna dalam daftar.

```
aws --region Region sagemaker delete-user-profile \
    --domain-id DomainId \
    --user-profile-name UserProfileName
```

6. Ambil daftar spasi bersama di Domain.

```
aws --region Region sagemaker list-spaces \
    --domain-id DomainId
```

7. Hapus setiap ruang bersama dalam daftar.

```
aws --region Region sagemaker delete-space \
    --domain-id DomainId \
    --space-name SpaceName
```

8. Hapus Domain. Untuk juga menghapus volume Amazon EFS, tentukan`HomeEfsFileSystem=Delete`.

```
aws --region Region sagemaker delete-domain \
```

```
    --domain-id DomainId \
    --retention-policy HomeEfsFileSystem=Retain
```

Profil Pengguna Domain

Profil pengguna mewakili satu pengguna dalam SageMaker Domain Amazon. Profil pengguna adalah cara utama untuk mereferensikan pengguna untuk tujuan berbagi, pelaporan, dan fitur berorientasi pengguna lainnya. Entitas ini dibuat saat pengguna melakukan onboard ke SageMaker Domain Amazon. Profil pengguna dapat memiliki (paling banyak) satu JupyterServer aplikasi di luar konteks ruang bersama. Aplikasi Studio Classic profil pengguna secara langsung terkait dengan profil pengguna dan memiliki direktori Amazon EFS yang terisolasi, peran eksekusi yang terkait dengan profil pengguna, dan aplikasi Kernel Gateway. Profil pengguna juga dapat membuat aplikasi lain dari konsol atau dari Amazon SageMaker Studio.

Topik

- Tambah dan Hapus Profil Pengguna
- Lihat Profil Pengguna dan Detail Profil Pengguna

Tambah dan Hapus Profil Pengguna

Bagian berikut menunjukkan cara menambahkan dan menghapus profil pengguna dari SageMaker Domain Amazon menggunakan SageMaker konsol atau AWS Command Line Interface (AWS CLI).

Topik

- Pilih nama pengguna
- Hapus profil

Pilih nama pengguna

Bagian berikut menunjukkan cara menambahkan profil pengguna ke Domain menggunakan SageMaker konsol atauAWS CLI.

Setelah Anda menambahkan profil pengguna ke domain, pengguna dapat login menggunakan URL. Jika domain digunakan AWS IAM Identity Center untuk otentikasi, pengguna akan menerima email yang berisi URL untuk masuk ke domain. Jika domain menggunakanAWS Identity and Access Management, Anda dapat membuat URL untuk profil pengguna menggunakan CreatePresignedDomainUrl

Tambahkan profil pengguna dari konsol

Anda dapat menambahkan profil pengguna ke Domain dari SageMaker konsol dengan mengikuti prosedur ini.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda tambahkan profil pengguna.
5. Pada halaman Detail domain, pilih tab Profil pengguna.
6. Pilih Tambahkan pengguna. Ini membuka halaman baru.
7. Gunakan nama default untuk profil pengguna Anda atau menambahkan nama kustom.
8. Untuk peran eksekusi, pilih opsi dari pemilih peran. Jika Anda memilih Masukkan ARN peran IAM kustom, peran tersebut harus memiliki, setidaknya, kebijakan kepercayaan terlampir yang SageMaker memberikan izin untuk mengambil peran tersebut. Untuk informasi selengkapnya, lihat SageMakerPeran.

   Jika Anda memilih Buat peran baru, kotak dialog Buat peran IAM akan terbuka:

   a. Untuk bucket S3 yang Anda tentukan, tentukan bucket Amazon S3 tambahan yang dapat diakses oleh pengguna notebook Anda. Jika Anda tidak ingin menambahkan akses ke ember lainnya, pilih Tidak Ada.

   b. Pilih Buat peran. SageMaker menciptakan peran IAM baru,`AmazonSageMaker-ExecutionPolicy`, dengan AmazonSageMakerFullAccesskebijakan terlampir.
9. (Opsional) Tambahkan tag ke profil pengguna. Semua sumber daya yang dibuat profil pengguna akan memiliki tag ARN Domain dan tag ARN profil pengguna. Tag ARN Domain didasarkan pada ID Domain, sedangkan tag ARN profil pengguna didasarkan pada nama profil pengguna.
10. Pilih Berikutnya.
11. Di bawah JupyterLab Versi default, pilih JupyterLab versi dari dropdown yang akan digunakan sebagai default untuk profil pengguna Anda. Untuk informasi tentang memilih JupyterLab versi, lihat JupyterLabPembuatan Versi.
12. Di JumpStart bagian SageMaker Proyek dan, Anda memiliki dua opsi. Anda dapat menerima Proyek dan JumpStart pengaturan default, atau Anda dapat menyesuaikan apakah profil pengguna dapat membuat proyek dan menggunakan JumpStart. Untuk informasi selengkapnya, lihat Izin SageMaker Studio yang Diperlukan untuk Menggunakan Proyek.

13. Pilih Berikutnya.
14. (Opsional) Jika Domain memiliki lisensi RStudio yang terkait, pilih apakah Anda ingin membuat pengguna dengan salah satu otorisasi berikut:
    - Tidak sah
    - Pilih IAM
    - Pengguna RStudio
15. Pilih Berikutnya.
16. Untuk konfigurasi izin dasar Canvas, pilih apakah akan menetapkan izin minimum yang diperlukan untuk menggunakan aplikasi SageMaker Canvas.
17. (Opsional) Untuk konfigurasi peramalan deret waktu: Untuk memberikan izin pengguna untuk peramalan deret waktu di SageMaker Canvas, biarkan opsi Aktifkan peramalan deret waktu diaktifkan. Ini diaktifkan secara default.
18. (Opsional) Jika Anda meninggalkan Aktifkan peramalan deret waktu diaktifkan, pilih Buat dan gunakan peran eksekusi baru. Atau, jika Anda sudah memiliki IAM role dengan izin Amazon Forecast yang diperlukan, pilih Gunakan peran eksekusi yang ada. Lihat informasi yang lebih lengkap di Metode pengaturan peran IAM.
19. Pilih Kirim.

Buat profil pengguna dari AWS CLI

Untuk membuat profil pengguna di Domain dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda. Untuk informasi tentang ARN JupyterLab versi yang tersedia, lihatMenyetel JupyterLab versi default.

```
aws --region region \
sagemaker create-user-profile \
--domain-id domain-id \
--user-profile-name user-name \
--user-settings '{
  "JupyterServerAppSettings": {
    "DefaultResourceSpec": {
      "SageMakerImageArn": "sagemaker-image-arn",
      "InstanceType": "system"
    }
  }
}'
```

Hapus profil

Semua aplikasi yang diluncurkan oleh profil pengguna harus dihapus untuk menghapus profil pengguna. Bagian berikut menunjukkan cara menghapus profil pengguna dari Domain menggunakan SageMaker konsol atauAWS CLI.

Hapus profil pengguna dari konsol

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain tempat Anda menghapus profil pengguna.
5. Pada halaman Detail domain, pilih tab Profil pengguna.
6. Pilih profil pengguna yang ingin Anda hapus.
7. Pada halaman Detail Pengguna, untuk setiap aplikasi yang tidak gagal dalam daftar Aplikasi, pilih Tindakan.
8. Dari daftar dropdown, pilih Delete.
9. Pada kotak dialog Hapus aplikasi, pilih Ya, hapus aplikasi. Kemudian masukkan hapus di bidang konfirmasi, dan pilih Hapus.
10. Saat Status ditampilkan sebagai Dihapus untuk semua aplikasi, pilih Edit.
11. Pada halaman Edit Pengguna, pilih Hapus pengguna.
12. Pada pop-up Hapus pengguna, pilih Ya, hapus pengguna.
13. Masukkan hapus di bidang untuk mengonfirmasi penghapusan.
14. Pilih Hapus.

Hapus profil pengguna dari AWS CLI

Untuk menghapus profil pengguna dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda.

```
aws sagemaker delete-user-profile \
--region region \
--domain-id domain-id \
--user-profile-name user-name
```

Lihat Profil Pengguna dan Detail Profil Pengguna

Topik ini menunjukkan cara melihat daftar profil pengguna di SageMaker Domain Amazon, dan melihat detail untuk profil pengguna dari SageMaker konsol atau AWS Command Line Interface (AWS CLI).

Topik

Lihat profil pengguna

Bagian berikut menjelaskan cara melihat daftar profil pengguna di Domain dari SageMaker konsol atauAWS CLI.

Lihat profil pengguna dari konsol

Selesaikan prosedur berikut ini untuk melihat daftar profil pengguna di Domain dari SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda lihat daftar profil pengguna.
5. Pada halaman Detail domain, pilih tab Profil pengguna.

Lihat profil pengguna dari AWS CLI

Untuk melihat profil pengguna di Domain dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda.

```
aws sagemaker list-user-profiles \
--region region \
--domain-id domain-id
```

Lihat detail profil pengguna

Bagian berikut menjelaskan cara untuk melihat detail profil pengguna dari SageMaker konsol atauAWS CLI.

Melihat detail profil pengguna dari konsol

Selesaikan prosedur berikut ini untuk melihat detail profil pengguna dari SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda lihat daftar profil pengguna.
5. Pada halaman Detail domain, pilih tab Profil pengguna.
6. Pilih profil pengguna yang detailnya ingin Anda lihat.

Lihat detail profil pengguna dari AWS CLI

Untuk mendeskripsikan profil pengguna dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda.

```
aws sagemaker describe-user-profile \
--region region \
--domain-id domain-id \
--user-profile-name user-name
```

Grup Pusat Identitas IAM dalam Domain

Jika Anda menggunakan AWS IAM Identity Center autentikasi untuk SageMaker Domain Amazon, Anda dapat menambahkan dan mengedit akses grup dan pengguna ke Domain. Untuk informasi selengkapnya tentang autentikasi IAM Identity Center, lihat Apa itu IAM Identity Center? . Topik berikut menunjukkan cara mengelola pengguna dan grup IAM Identity Center yang memiliki akses ke Domain.

Topik

- Lihat grup dan pengguna
- Tambahkan grup dan pengguna
- Hapus grup

Lihat grup dan pengguna

Selesaikan prosedur berikut untuk melihat daftar grup dan pengguna Pusat Identitas IAM dari SageMaker konsol Amazon.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda buka halaman Pengaturan domain.
5. Pada halaman Detail domain, pilih tab Grup.

Tambahkan grup dan pengguna

Bagian berikut menunjukkan cara menambahkan grup dan pengguna ke domain dari SageMaker konsol atauAWS CLI.

> **Note**
>
> Jika domain dibuat sebelum 1 Oktober 2023, Anda hanya dapat menambahkan grup dan pengguna ke domain dari SageMaker konsol.

Konsol SageMaker

Selesaikan prosedur berikut ini untuk menambahkan grup dan pengguna ke Domain Anda dari SageMaker konsol.

1. Pada tab Grup, pilih Tetapkan pengguna dan grup.
2. Pada halaman Tetapkan pengguna dan grup, pilih pengguna dan grup yang ingin Anda tambahkan.
3. Pilih Tetapkan pengguna dan grup.

AWS CLI

Selesaikan prosedur berikut untuk menambahkan grup dan pengguna ke domain Anda dariAWS CLI.

1. Ambil domain dengan panggilan untuk mendeskripsikan domain. `SingleSignOnApplicationArn` `SingleSignOnApplicationArn`adalah ARN dari aplikasi yang dikelola di IAM Identity Center.

```
aws sagemaker describe-domain \
--region region \
--domain-id domain-id
```

2. Kaitkan pengguna atau grup dengan domain. Untuk mencapai ini, teruskan `SingleSignOnApplicationArn` nilai yang dikembalikan dari perintah deskripsi-domain sebagai `application-arn` parameter dalam panggilan ke create-application-assignment.Anda juga harus meneruskan tipe dan ID entitas untuk diasosiasikan.

```
aws sso-admin create-application-assignment \
--application-arn application-arn \
--principal-id principal-id \
--principal-type principal-type
```

Hapus grup

Selesaikan prosedur berikut untuk menghapus grup dari Domain Anda dari SageMaker konsol. Untuk informasi tentang menghapus pengguna, lihatHapus profil.

1. Di tab Grup, pilih grup yang ingin Anda hapus.
2. Pilih Unassign grup.
3. Pada jendela pop-up, pilih Ya, batalkan penetapan grup.
4. Masukkan unassign di bidang.
5. Pilih Unassign grup.

Cepat onboard ke Domain Amazon SageMaker

Topik ini menjelaskan cara onboard ke SageMaker Domain Amazon dengan menggunakan prosedur Mengatur untuk pengguna tunggal dari SageMaker konsol. Prosedur ini mengatur Domain Anda dalam satu klik dan menggunakan otentikasi AWS Identity and Access Management (IAM).

Untuk informasi tentang cara mengatur Domain Anda dengan setelan lanjutan, lihat Orientasi khusus menggunakan IAM pengaturan autentikasi IAM kustom dan Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center pengaturan autentikasi Pusat Identitas IAM kustom.

Dukungan RStudio saat ini tidak tersedia saat orientasi menggunakan prosedur Pengaturan untuk pengguna tunggal. Untuk menggunakan RStudio, Anda harus onboard menggunakan prosedur Pengaturan untuk organisasi.

Topik

- Pengaturan cepat untuk pengguna tunggal
- Pengaturan default

Pengaturan cepat untuk pengguna tunggal

Onboard ke Domain dengan cepat menggunakan Siapkan untuk pengguna tunggal dari SageMaker halaman arahan

Setelah memenuhi prasyarat, ikuti langkah-langkahnya. Mengatur SageMaker Prasyarat Amazon

1. Buka konsol SageMaker .
2. Di bawah New to SageMaker? , pilih Siapkan untuk pengguna tunggal. Domain dan profil pengguna Anda dibuat secara otomatis.

Onboard ke Domain dengan cepat menggunakan Pengaturan untuk pengguna tunggal dari halaman Domain

1. Buka konsol SageMaker .
2. Buka panel navigasi di sebelah kiri.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih Create domain (Buat domain).
5. Pilih Siapkan untuk pengguna tunggal (Pengaturan cepat). Domain dan profil pengguna Anda dibuat secara otomatis.

Sekarang Anda telah onboard ke Domain, Anda dapat meluncurkan aplikasi dengan membuka SageMaker konsol dan memilih lingkungan dari panel navigasi kiri.

Untuk informasi tentang menambahkan pengguna ke Domain Anda, lihatTambah dan Hapus Profil Pengguna.

Pengaturan default

Saat Anda melakukan onboard ke SageMaker Domain Amazon menggunakan prosedur Penyiapan untuk pengguna tunggal, SageMaker gunakan default berikut sebagai bagian dari pembuatan domain dan profil pengguna. Anda tidak dapat mengubah default ini menggunakan prosedur Set up for single user.

- Nama domain: SageMaker secara otomatis menetapkan nama domain dengan stempel waktu dalam format berikut.

  ```
  QuickSetupDomain-YYYYMMDDTHHMMSS
  ```

- Nama profil pengguna: SageMaker secara otomatis menetapkan nama profil pengguna dengan stempel waktu dalam format berikut.

  ```
  default-YYYYMMDDTHHMMSS
  ```

- Peran eksekusi domain: SageMaker membuat peran IAM baru dan melampirkan kebijakan. `AmazonSageMakerFullAccess` Saat menggunakan penyiapan cepat dan Amazon SageMaker Studio yang diperbarui adalah pengalaman default Anda, peran IAM Anda juga menyertakan `AmazonSageMakerCanvasFullAccess`, `AmazonSageMakerCanvasAIServicesAccess`, `AmazonS3FullAccess`kebijakan.
- Peran eksekusi profil pengguna: SageMaker menetapkan peran eksekusi profil pengguna ke peran IAM yang sama yang digunakan untuk peran eksekusi domain.
- Peran eksekusi ruang bersama: SageMaker menetapkan peran eksekusi ruang bersama ke peran IAM yang sama yang digunakan untuk peran eksekusi domain.
- SageMaker Peran peramalan deret waktu kanvas: SageMaker membuat peran IAM baru dengan izin yang diperlukan untuk menggunakan fitur peramalan deret waktu SageMaker Canvas.
- Bucket Amazon S3: SageMaker membuat bucket Amazon S3 bernama dengan format berikut untuk berbagi notebook.

  ```
  sagemaker-studio-XXXXXXXXXXXXXXX
  ```

- Amazon VPC: SageMaker memilih VPC publik dengan logika berikut.

  1. Jika ada VPC default dengan subnet terkait di Wilayah, SageMaker gunakan itu.

2. Jika tidak ada VPC default atau VPC default tidak memiliki subnet terkait, maka gunakan VPC yang SageMaker ada dengan subnet terkait. Jika ada beberapa VPC yang ada, SageMaker dapat memilih salah satu dari mereka.

Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center

Topik ini menjelaskan cara onboard ke SageMaker Domain Amazon menggunakan autentikasi menggunakan AWS IAM Identity Center dari SageMaker konsol atau. AWS CLI Untuk informasi tentang pengaturan IAM Identity Center untuk digunakan dengan Domain, lihatSiapkan Pusat Identitas IAM untuk digunakan dengan Domain Amazon SageMaker . Untuk informasi tentang cara onboard menggunakan autentikasi AWS Identity and Access Management (IAM), lihat atau. Masukan Cepat Orientasi khusus menggunakan IAM

Topik

- Onboard dari konsol
- Onboard dari AWS CLI
- Akses Domain setelah onboarding
- Siapkan Pusat Identitas IAM untuk digunakan dengan Domain Amazon SageMaker

Onboard dari konsol

Untuk onboard ke Domain menggunakan IAM Identity Center

1. Buka konsol SageMaker .
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari halaman Domain, pilih Buat domain.
5. Pada halaman Siapkan SageMaker Domain, pilih Siapkan untuk organisasi.
6. Pilih Konfigurasi.

Langkah 1: Detail domain

1. Untuk Nama Domain, masukkan nama yang unik untuk Domain Anda. Misalnya, ini bisa berupa nama proyek atau tim Anda.
2. Pilih Berikutnya.

Langkah 2: Pengguna dan Aktivitas ML

Pilih grup atau buat pengguna untuk Domain dan berikan mereka izin untuk aktivitas ML yang Anda ingin mereka akses.

Dalam instruksi pengaturan ini kami menggunakan opsi IAM Identity Center. Untuk menggunakan otentikasi di Pusat Identitas IAM, Anda harus menjadi anggota organisasi di. AWS Organizations Untuk informasi tentang menyiapkan Pusat Identitas IAM, lihatSiapkan Pusat Identitas IAM untuk digunakan dengan Domain Amazon SageMaker .

Peran IAM yang Anda konfigurasikan dalam langkah ini ditetapkan ke grup Pusat Identitas IAM yang dipilih dan semua pengguna yang merupakan bagian dari grup tersebut.

1. Di bawah Bagaimana Anda ingin mengakses Studio? , pilih Pusat AWS Identitas.
2. Di bawah Siapa yang akan menggunakan Studio? pilih pengguna atau grup Pusat Identitas IAM, lalu pilih Pilih. Anda perlu menyiapkan Amazon SageMaker Studio dalam Wilayah yang sama di mana Pusat Identitas IAM Anda dikonfigurasi. Anda dapat mengubah Wilayah Domain Anda dengan memilih Wilayah dari daftar tarik-turun di kanan atas konsol atau Anda dapat mengubah Wilayah Pusat Identitas IAM Anda dengan menavigasi ke portal akses. AWS
3. Di bawah Aktivitas ML apa yang mereka lakukan? Anda dapat menggunakan peran yang sudah ada dengan memilih Gunakan peran yang sudah ada atau Anda dapat membuat peran baru dengan memilih Buat peran baru dan memeriksa aktivitas ML yang ingin Anda akses peran tersebut. Anda dapat memilih paling banyak 10 aktivitas ML.
4. Saat memilih aktivitas MLM, Anda mungkin perlu memenuhi persyaratan. Untuk memenuhi persyaratan, pilih Tambah dan lengkapi persyaratan.
5. Setelah semua persyaratan terpenuhi, pilih Berikutnya.

Langkah 3: Aplikasi

Pada langkah ini, Anda dapat mengonfigurasi aplikasi yang telah Anda aktifkan pada langkah sebelumnya. Untuk informasi lebih lanjut tentang aktivitas ML, lihatReferensi aktivitas MLM.

Jika aplikasi belum diaktifkan, Anda menerima peringatan untuk aplikasi itu. Untuk mengaktifkan aplikasi yang belum diaktifkan, kembali ke langkah sebelumnya dengan memilih Kembali dan ikuti instruksi sebelumnya.

SageMaker Konfigurasi studio:

Di bawah SageMaker Studio, Anda memiliki opsi untuk memilih antara versi Studio baru dan klasik sebagai pengalaman default Anda. Ini berarti memilih lingkungan ML mana yang akan berinteraksi dengan Anda setelah membuka Studio.

- SageMaker Studio - New mencakup beberapa lingkungan pengembangan terintegrasi (IDE) dan aplikasi, termasuk Amazon SageMaker Studio Classic. Jika dipilih, Studio Classic IDE memiliki pengaturan default. Untuk informasi tentang pengaturan default, lihatPengaturan default.
- SageMaker Studio Classic termasuk IDE Jupyter. Jika dipilih, Anda dapat mengonfigurasi konfigurasi Studio Classic.

  Untuk informasi tentang Studio Classic, lihatAmazon SageMaker Studio Klasik.

SageMaker Konfigurasi kanvas:

Jika Anda mengaktifkan Amazon SageMaker Canvas, lihat Memulai dengan menggunakan Amazon SageMaker Canvas petunjuk dan detail konfigurasi untuk orientasi.

SageMaker Konfigurasi Studio Klasik:

Jika Anda memilih SageMaker Studio - Baru (disarankan) sebagai pengalaman default Anda, Studio Classic IDE memiliki pengaturan default. Untuk informasi tentang pengaturan default, lihatPengaturan default.

Jika Anda memilih Studio Classic sebagai pengalaman default, Anda dapat memilih untuk mengaktifkan atau menonaktifkan berbagi sumber daya notebook. Sumber daya notebook mencakup artefak seperti keluaran sel dan repositori Git. Untuk informasi selengkapnya tentang sumber daya Notebook, lihatBagikan dan Gunakan Notebook Amazon SageMaker Studio Classic.

Jika Anda mengaktifkan berbagi sumber daya notebook:

1. Di bawah lokasi S3 untuk sumber daya notebook yang dapat dibagikan, masukkan lokasi Amazon S3 Anda.
2. Di bawah kunci Enkripsi - opsional, biarkan sebagai Tanpa Enkripsi Kustom atau pilih AWS KMS kunci yang ada atau pilih Masukkan ARN kunci KMS dan masukkan ARN kunci AWS KMS Anda.

3. Di bawah Preferensi berbagi keluaran sel Notebook, pilih Izinkan pengguna berbagi keluaran sel atau Nonaktifkan berbagi keluaran sel.

konfigurasi RStudio:

Untuk mengaktifkan RStudio, Anda memerlukan lisensi RStudio. Untuk mengaturnya, lihatLisensi RStudio.

1. Di bawah RStudio Workbench, verifikasi bahwa lisensi RStudio Anda terdeteksi secara otomatis. Untuk informasi selengkapnya tentang mendapatkan lisensi RStudio dan mengaktifkannya SageMaker, lihat. Lisensi RStudio
2. Pilih jenis instans untuk meluncurkan Server RStudio Anda. Untuk informasi selengkapnya, lihat Jenis StudioServerPro contoh R.
3. Di bawah Izin, buat peran Anda atau pilih peran yang ada. Peran harus memiliki kebijakan izin berikut. Kebijakan ini memungkinkan StudioServerPro aplikasi R untuk mengakses sumber daya yang diperlukan. Ini juga memungkinkan Amazon SageMaker untuk secara otomatis meluncurkan StudioServerPro aplikasi R ketika StudioServerPro aplikasi R yang ada dalam `Failed` status `Deleted` atau. Untuk informasi tentang menambahkan izin ke peran, lihat Memodifikasi kebijakan izin peran (konsol).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "license-manager:ExtendLicenseConsumption",
                "license-manager:ListReceivedLicenses",
                "license-manager:GetLicense",
                "license-manager:CheckoutLicense",
                "license-manager:CheckInLicense",
                "logs:CreateLogDelivery",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DeleteLogDelivery",
                "logs:Describe*",
                "logs:GetLogDelivery",
                "logs:GetLogEvents",
                "logs:ListLogDeliveries",
```

```
                "logs:PutLogEvents",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery",
                "sagemaker:CreateApp"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Di bawah RStudio Connect, tambahkan URL untuk server RStudio Connect Anda. RStudio Connect adalah platform penerbitan untuk aplikasi Shiny, laporan R Markdown, dasbor, plot, dan banyak lagi. Saat Anda onboard ke RStudio aktif SageMaker, server RStudio Connect tidak dibuat. Untuk informasi selengkapnya, lihat URL RStudio Connect.
5. Di bawah RStudio Package Manager, tambahkan URL untuk RStudio Package Manager Anda. SageMaker membuat repositori paket default untuk Package Manager saat Anda onboard RStudio. Untuk informasi lebih lanjut tentang RStudio Package Manager, lihatManajer Package RStudio.
6. Pilih Selanjutnya.

Konfigurasi Editor Kode:

Jika Anda mengaktifkan Editor Kode, lihat Editor Kode untuk ikhtisar dan detail konfigurasi.

Langkah 4: Jaringan

Pilih bagaimana Anda ingin Studio terhubung ke AWS layanan lain.

Anda dapat memilih untuk menonaktifkan akses internet ke Studio Anda dengan menentukan jenis akses jaringan hanya Virtual Private Cloud (VPC) Virtual Private Cloud (VPC). Jika memilih opsi ini, Anda tidak dapat menjalankan buku catatan Studio kecuali VPC Anda memiliki titik akhir antarmuka ke SageMaker API dan waktu proses, atau gateway Network Address Translation (NAT) dengan akses internet, dan grup keamanan Anda mengizinkan koneksi keluar. Untuk informasi selengkapnya tentang Amazon VPC, lihatPilih VPC Amazon.

Jika Anda memilih Virtual Private Cloud (VPC) Hanya langkah-langkah berikut yang diperlukan. Jika Anda memilih akses internet publik, dua langkah pertama berikut diperlukan.

1. Di bawah VPC, pilih ID VPC Amazon.

2. Di bawah Subnet, pilih satu subnet atau lebih. Jika Anda tidak memilih subnet apa pun, SageMaker gunakan semua subnet di Amazon VPC. Kami menyarankan Anda menggunakan beberapa subnet yang tidak dibuat di Availability Zone yang dibatasi. Menggunakan subnet di Availability Zone yang dibatasi ini dapat mengakibatkan kesalahan kapasitas yang tidak mencukupi dan waktu pembuatan aplikasi yang lebih lama. Untuk informasi selengkapnya tentang Availability Zone, lihat Availability Zone.
3. Dalam Grup keamanan, pilih satu subnet atau beberapa.

Jika hanya VPC yang dipilih, SageMaker secara otomatis menerapkan pengaturan grup keamanan yang ditentukan untuk Domain ke semua ruang bersama yang dibuat di Domain. Jika hanya Internet publik yang dipilih, SageMaker tidak menerapkan pengaturan grup keamanan ke spasi bersama yang dibuat di Domain.

Langkah 5: Penyimpanan

Anda memiliki opsi untuk mengenkripsi data Anda. Sistem file Amazon Elastic File System (Amazon Elastic) dan Amazon Elastic Block Store (Amazon EBS) yang dibuat untuk Anda saat Anda membuat Domain. Ukuran Amazon EBS digunakan oleh Editor Kode dan JupyterLab spasi.

Anda tidak dapat mengubah kunci enkripsi setelah mengenkripsi sistem file Amazon EFS dan Amazon EBS Anda. Untuk mengenkripsi sistem file Amazon EFS dan Amazon EBS, Anda dapat menggunakan konfigurasi berikut.

- Di bawah kunci Enkripsi - opsional, biarkan sebagai Tanpa Enkripsi Kustom atau pilih kunci KMS yang ada atau pilih Masukkan ARN kunci KMS dan masukkan ARN kunci KMS Anda.
- Di bawah Ukuran ruang default - opsional, masukkan ukuran ruang default.
- Di bawah Ukuran ruang maksimum - opsional, masukkan ukuran ruang maksimum.

Langkah 6: Tinjau dan buat

Peninjauan pengaturan Domain Anda. Jika Anda perlu mengubah pengaturan, pilih Edit di sebelah langkah yang relevan. Setelah Anda mengonfirmasi bahwa pengaturan Domain Anda akurat, pilih Kirim dan Domain dibuat untuk Anda. Proses ini mungkin memerlukan waktu beberapa menit.

Onboard dari AWS CLI

Gunakan perintah berikut untuk onboard ke Domain menggunakan otentikasi menggunakan IAM Identity Center dari. AWS CLI

1. Buat peran eksekusi yang digunakan untuk membuat Domain dan lampirkan AmazonSageMakerFullAccesskebijakan. Anda juga dapat menggunakan peran yang ada, setidaknya, memiliki kebijakan kepercayaan terlampir yang memberikan SageMaker izin untuk mengambil peran tersebut. Untuk informasi selengkapnya, lihat SageMaker Peran.

```
aws iam create-role --role-name execution-role-name --assume-role-policy-
document file://execution-role-trust-policy.json
aws iam attach-role-policy --role-name execution-role-name --policy-arn
 arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

2. Dapatkan Amazon Virtual Private Cloud (Amazon VPC) default akun Anda.

```
aws --region region ec2 describe-vpcs --filters Name=isDefault,Values=true --query
 "Vpcs[0].VpcId" --output text
```

3. Dapatkan daftar subnet di Amazon VPC default.

```
aws --region region ec2 describe-subnets --filters Name=vpc-id,Values=default-vpc-
id --query "Subnets[*].SubnetId" --output json
```

4. Buat Domain dengan meneruskan ID VPC Amazon default, subnet, dan peran eksekusi ARN. Anda juga harus memberikan SageMaker gambar ARN. Untuk informasi tentang ARN JupyterLab versi yang tersedia, lihatMenyetel JupyterLab versi default.

```
aws --region region sagemaker create-domain --domain-name domain-name --vpc-
id default-vpc-id --subnet-ids subnet-ids --auth-mode SSO --default-user-
settings "ExecutionRole=arn:aws:iam::account-number:role/execution-role-
name,JupyterServerAppSettings={DefaultResourceSpec={InstanceType=system,SageMakerImageArn=i
arn}}" \ --query DomainArn --output text
```

5. Verifikasi bahwa Domain telah dibuat.

```
aws --region region sagemaker  list-domains
```

Akses Domain setelah onboarding

Setelah Anda diberi akses ke Domain, Anda akan dikirimi email yang mengundang Anda untuk membuat kata sandi dan menggunakan IAM Identity Center. Email juga berisi URL untuk masuk ke Domain. Untuk informasi selengkapnya tentang masuk dan durasi sesi, lihat Cara masuk ke portal pengguna.

Setelah Anda mengaktifkan akun Anda, buka URL Domain, masuk, dan tunggu profil pengguna Anda dibuat. Pada kunjungan berikutnya, Anda hanya perlu menunggu aplikasi Studio dimuat.

Tandai URL. URL juga tersedia di halaman pengaturan Domain.

Siapkan Pusat Identitas IAM untuk digunakan dengan Domain Amazon SageMaker

Untuk menggunakan otentikasi di Pusat Identitas IAM, Anda harus menjadi anggota organisasi di. AWS Organizations Jika Anda bukan anggota organisasi, Anda dapat membuatnya dengan mengikuti langkah-langkah dalam Tutorial: Membuat dan mengonfigurasi organisasi.

Autentikasi Multi-Faktor (MFA) diaktifkan secara default ketika Anda membuat instans IAM Identity Center. Pengguna diminta untuk masuk dengan MFA saat perangkat, browser, atau lokasi mereka berubah. Sebagai praktik keamanan terbaik, kami sangat menyarankan untuk mengaktifkan MFA untuk identitas tenaga kerja Anda. Untuk informasi selengkapnya, lihat Mengelola perangkat MFA di Pusat Identitas IAM.

Setelah membuat organisasi dan pengguna, Anda dapat membuat profil SageMaker pengguna untuk pengguna tersebut di Pusat Identitas IAM sebagai berikut.

1. Dari SageMaker konsol Amazon: — Anda dapat menggunakan SageMaker konsol Amazon untuk membuat profil pengguna bagi pengguna di Pusat Identitas IAM. Jika pengguna di Pusat Identitas IAM belum dikaitkan dengan Domain, itu secara otomatis dikaitkan.
2. Menggunakan AWS CLI atau AWS CloudFormation — Pengguna di Pusat Identitas IAM yang ditetapkan ke Domain dapat membuat profil pengguna menggunakan SageMaker konsol, AWS CLI atauAWS CloudFormation.
   - Pertama, Anda harus menggunakan konsol Pusat Identitas IAM untuk menetapkan pengguna atau grup pengguna ke Domain. Untuk informasi selengkapnya tentang penetapan aplikasi, lihat Menetapkan akses pengguna.
   - Kemudian, gunakan AWS CLI atau AWS CloudFormation untuk membuat profil SageMaker pengguna untuk pengguna.

> **Note**
>
> Untuk menyederhanakan administrasi izin akses, sebaiknya gunakan IAM Identity Center untuk menetapkan grup ke Domain (bukan menetapkan pengguna). Grup mengizinkan izin diberikan atau ditolak ke beberapa pengguna sekaligus. Pengguna dapat dipindahkan dari grup atau ke grup lain jika diperlukan. Saat menetapkan akses pengguna ke aplikasi, Pusat

Identitas IAM saat ini tidak mendukung pengguna yang ditambahkan ke grup bersarang. Jika pengguna ditambahkan ke grup bersarang, mereka mungkin menerima pesan `"You do not have any applications"` galat saat login. Penugasan harus dilakukan ke grup langsung yang menjadi anggota pengguna.

Kembali ke halaman Domain untuk melanjutkan ke onboard menggunakan autentikasi IAM Identity Center.

Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM

Topik ini menjelaskan cara onboard ke SageMaker Domain Amazon menggunakan prosedur Penyiapan untuk organisasi untuk autentikasi AWS Identity and Access Management (IAM) dari SageMaker konsol atau. AWS CLI Untuk onboard lebih cepat menggunakan IAM, lihat. Masukan Cepat

Untuk informasi tentang cara onboard menggunakan AWS IAM Identity Center (IAM Identity Center), lihat. Orientasi khusus menggunakan IAM Identity Center

Onboard menggunakan konsol

Untuk onboard ke Domain menggunakan IAM

1. Buka konsol SageMaker .
2. Di panel navigasi sebelah kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari halaman Domain, pilih Buat domain.
5. Pada halaman Setup SageMaker Domain, pilih Siapkan untuk organisasi.
6. Pilih Konfigurasi.

Langkah 1: Detail domain

1. Untuk Nama Domain, masukkan nama yang unik untuk Domain Anda. Misalnya, ini bisa berupa nama proyek atau tim Anda.
2. Pilih Berikutnya.

Langkah 2: Pengguna dan Aktivitas ML

Pilih grup atau buat pengguna untuk Domain dan berikan izin untuk aktivitas ML mana yang akan mereka akses.

Dalam instruksi pengaturan ini, kami menggunakan opsi Login melalui IAM.

Peran IAM yang Anda konfigurasikan pada langkah ini ditetapkan untuk semua pengguna yang Anda tambahkan dalam langkah ini.

1. Di bawah Bagaimana Anda ingin mengakses Studio? , pilih Login melalui IAM.
2. Di bawah Siapa yang akan menggunakan Studio? tambahkan nama profil pengguna. Untuk menambahkan nama profil pengguna, pilih Tambah pengguna, masukkan nama profil pengguna, lalu pilih Pilih.
3. Di bawah Aktivitas ML apa yang mereka lakukan? Anda dapat menggunakan peran yang sudah ada dengan memilih Gunakan peran yang sudah ada atau Anda dapat membuat peran baru dengan memilih Buat peran baru dan memeriksa aktivitas ML yang ingin Anda akses peran tersebut. Anda dapat memilih paling banyak 10 aktivitas ML.
4. Saat memilih aktivitas MLM, Anda mungkin perlu memenuhi persyaratan. Untuk memenuhi persyaratan, pilih Tambah dan lengkapi persyaratan.
5. Setelah semua persyaratan terpenuhi, pilih Berikutnya.

Langkah 3: Aplikasi

Pada langkah ini, Anda dapat mengonfigurasi aplikasi yang telah Anda aktifkan pada langkah sebelumnya. Untuk informasi lebih lanjut tentang aktivitas ML, lihatReferensi aktivitas MLM.

Jika aplikasi belum diaktifkan, Anda menerima peringatan untuk aplikasi itu. Untuk mengaktifkan aplikasi yang belum diaktifkan, kembali ke langkah sebelumnya dengan memilih Kembali dan ikuti instruksi sebelumnya.

SageMaker Konfigurasi studio:

Di bawah SageMaker Studio, Anda memiliki opsi untuk memilih antara versi Studio baru dan klasik sebagai pengalaman default Anda. Ini berarti memilih lingkungan ML mana yang akan berinteraksi dengan Anda setelah membuka Studio.

- SageMaker Studio - New mencakup beberapa lingkungan pengembangan terintegrasi (IDE) dan aplikasi, termasuk Amazon SageMaker Studio Classic. Jika dipilih, Studio Classic IDE memiliki pengaturan default. Untuk informasi tentang pengaturan default, lihatPengaturan default.
- SageMaker Studio Classic termasuk IDE Jupyter. Jika dipilih, Anda dapat mengonfigurasi konfigurasi Studio Classic.

  Untuk informasi tentang Studio Classic, lihatAmazon SageMaker Studio Klasik.

SageMaker Konfigurasi kanvas:

Jika Anda mengaktifkan Amazon SageMaker Canvas, lihat Memulai dengan menggunakan Amazon SageMaker Canvas petunjuk dan detail konfigurasi untuk orientasi.

SageMaker Konfigurasi Studio Klasik:

Jika Anda memilih SageMaker Studio - Baru (disarankan) sebagai pengalaman default Anda, Studio Classic IDE memiliki pengaturan default. Untuk informasi tentang pengaturan default, lihatPengaturan default.

Jika Anda memilih Studio Classic sebagai pengalaman default, Anda dapat memilih untuk mengaktifkan atau menonaktifkan berbagi sumber daya notebook. Sumber daya notebook mencakup artefak seperti keluaran sel dan repositori Git. Untuk informasi selengkapnya tentang sumber daya Notebook, lihatBagikan dan Gunakan Notebook Amazon SageMaker Studio Classic.

Jika Anda mengaktifkan berbagi sumber daya notebook:

1. Di bawah lokasi S3 untuk sumber daya notebook yang dapat dibagikan, masukkan lokasi Amazon S3 Anda.
2. Di bawah kunci Enkripsi - opsional, biarkan sebagai Tanpa Enkripsi Kustom atau pilih AWS KMS kunci yang ada atau pilih Masukkan ARN kunci KMS dan masukkan ARN kunci AWS KMS Anda.
3. Di bawah Preferensi berbagi keluaran sel Notebook, pilih Izinkan pengguna berbagi keluaran sel atau Nonaktifkan berbagi keluaran sel.

konfigurasi RStudio:

Untuk mengaktifkan RStudio Anda akan memerlukan lisensi RStudio. Untuk mengaturnya, lihatLisensi RStudio.

1. Di bawah RStudio Workbench, verifikasi bahwa lisensi RStudio Anda terdeteksi secara otomatis. Untuk informasi selengkapnya tentang mendapatkan lisensi RStudio dan mengaktifkannya SageMaker, lihat. Lisensi RStudio
2. Pilih jenis instans untuk meluncurkan Server RStudio Anda. Untuk informasi selengkapnya, lihat Jenis StudioServerPro contoh R.
3. Di bawah Izin, buat peran Anda atau pilih peran yang ada. Peran harus memiliki kebijakan izin berikut. Kebijakan ini memungkinkan StudioServerPro aplikasi R untuk mengakses sumber daya yang diperlukan. Ini juga memungkinkan Amazon SageMaker untuk secara otomatis meluncurkan StudioServerPro aplikasi R ketika StudioServerPro aplikasi R yang ada dalam `Failed` status `Deleted` atau. Untuk informasi tentang menambahkan izin ke peran, lihat Memodifikasi kebijakan izin peran (konsol).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "license-manager:ExtendLicenseConsumption",
                "license-manager:ListReceivedLicenses",
                "license-manager:GetLicense",
                "license-manager:CheckoutLicense",
                "license-manager:CheckInLicense",
                "logs:CreateLogDelivery",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DeleteLogDelivery",
                "logs:Describe*",
                "logs:GetLogDelivery",
                "logs:GetLogEvents",
                "logs:ListLogDeliveries",
                "logs:PutLogEvents",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery",
                "sagemaker:CreateApp"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

4. Di bawah RStudio Connect, tambahkan URL untuk server RStudio Connect Anda. RStudio Connect adalah platform penerbitan untuk aplikasi Shiny, laporan R Markdown, dasbor, plot, dan banyak lagi. Saat Anda onboard ke RStudio aktif SageMaker, server RStudio Connect tidak dibuat. Untuk informasi selengkapnya, lihat URL RStudio Connect.
5. Di bawah RStudio Package Manager, tambahkan URL untuk RStudio Package Manager Anda. SageMaker membuat repositori paket default untuk Package Manager saat Anda onboard RStudio. Untuk informasi lebih lanjut tentang RStudio Package Manager, lihatManajer Package RStudio.
6. Pilih Selanjutnya.

Konfigurasi Editor Kode:

Jika Anda mengaktifkan Editor Kode, lihat Editor Kode untuk ikhtisar dan detail konfigurasi.

Langkah 4: Jaringan

Pilih bagaimana Anda ingin Studio terhubung ke AWS layanan lain.

Anda dapat memilih untuk menonaktifkan akses internet ke Studio Anda dengan menentukan jenis akses jaringan hanya Virtual Private Cloud (VPC) Virtual Private Cloud (VPC). Jika memilih opsi ini, Anda tidak dapat menjalankan buku catatan Studio kecuali VPC Anda memiliki titik akhir antarmuka ke SageMaker API dan waktu proses, atau gateway Network Address Translation (NAT) dengan akses internet, dan grup keamanan Anda mengizinkan koneksi keluar. Untuk informasi selengkapnya tentang Amazon VPC, lihatPilih VPC Amazon.

Jika Anda memilih Virtual Private Cloud (VPC) Hanya langkah-langkah berikut yang diperlukan. Jika Anda memilih akses internet publik, dua langkah pertama berikut diperlukan.

1. Di bawah VPC, pilih ID VPC Amazon.
2. Di bawah Subnet, pilih satu subnet atau lebih. Jika Anda tidak memilih subnet apa pun, SageMaker gunakan semua subnet di Amazon VPC. Kami menyarankan Anda menggunakan beberapa subnet yang tidak dibuat di Availability Zone yang dibatasi. Menggunakan subnet di Availability Zone yang dibatasi ini dapat mengakibatkan kesalahan kapasitas yang tidak mencukupi dan waktu pembuatan aplikasi yang lebih lama. Untuk informasi selengkapnya tentang Availability Zone, lihat Availability Zone.
3. Dalam Grup keamanan, pilih satu subnet atau beberapa.

Jika hanya VPC yang dipilih, SageMaker secara otomatis menerapkan pengaturan grup keamanan yang ditentukan untuk Domain ke semua ruang bersama yang dibuat di Domain. Jika hanya Internet publik yang dipilih, SageMaker tidak menerapkan pengaturan grup keamanan ke spasi bersama yang dibuat di Domain.

Langkah 5: Penyimpanan

Anda memiliki opsi untuk mengenkripsi data Anda. Sistem file Amazon Elastic File System (Amazon Elastic) dan Amazon Elastic Block Store (Amazon EBS) yang dibuat untuk Anda saat Anda membuat Domain. Ukuran Amazon EBS digunakan oleh Editor Kode dan JupyterLab spasi.

Anda tidak dapat mengubah kunci enkripsi setelah mengenkripsi sistem file Amazon EFS dan Amazon EBS Anda. Untuk mengenkripsi sistem file Amazon EFS dan Amazon EBS, Anda dapat menggunakan konfigurasi berikut.

- Di bawah kunci Enkripsi - opsional, biarkan sebagai Tanpa Enkripsi Kustom atau pilih kunci KMS yang ada atau pilih Masukkan ARN kunci KMS dan masukkan ARN kunci KMS Anda.
- Di bawah Ukuran ruang default - opsional, masukkan ukuran ruang default.
- Di bawah Ukuran ruang maksimum - opsional, masukkan ukuran ruang maksimum.

Langkah 6: Tinjau dan buat

Peninjauan pengaturan Domain Anda. Jika Anda perlu mengubah pengaturan, pilih Edit di sebelah langkah yang relevan. Setelah Anda mengonfirmasi bahwa pengaturan Domain Anda akurat, pilih Kirim dan Domain dibuat untuk Anda. Proses ini mungkin memerlukan waktu beberapa menit.

Onboard menggunakan AWS CLI

Gunakan perintah berikut untuk onboard ke Domain menggunakan otentikasi menggunakan IAM dari. AWS CLI

1. Buat peran eksekusi yang digunakan untuk membuat Domain dan lampirkan AmazonSageMakerFullAccesskebijakan. Anda juga dapat menggunakan peran yang ada, setidaknya, memiliki kebijakan kepercayaan terlampir yang memberikan SageMaker izin untuk mengambil peran tersebut. Untuk informasi selengkapnya, lihat SageMaker Peran.

   ```
   aws iam create-role --role-name execution-role-name
   aws iam attach-role-policy --role-name execution-role-name --policy-arn
    arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
   ```

2. Dapatkan Amazon Virtual Private Cloud (Amazon VPC) default akun Anda.

```
aws --region region ec2 describe-vpcs --filters Name=isDefault,Values=true --query
 "Vpcs[0].VpcId" --output text
```

3. Dapatkan daftar subnet di Amazon VPC default.

```
aws --region region ec2 describe-subnets --filters Name=vpc-id,Values=default-vpc-
id --query "Subnets[*].SubnetId" --output json
```

4. Buat Domain dengan meneruskan ID VPC Amazon default, subnet, dan peran eksekusi ARN. Anda juga harus memberikan SageMaker gambar ARN. Untuk informasi tentang ARN JupyterLab versi yang tersedia, lihatMenyetel JupyterLab versi default.

```
aws --region region sagemaker create-domain --domain-name domain-name --vpc-
id default-vpc-id --subnet-ids subnet-ids --auth-mode IAM --default-user-
settings "ExecutionRole=arn:aws:iam::account-number:role/execution-role-
name,JupyterServerAppSettings={DefaultResourceSpec={InstanceType=system,SageMakerImageArn=i
arn}}" \ --query DomainArn --output text
```

5. Verifikasi bahwa Domain telah dibuat.

```
aws --region region sagemaker  list-domains
```

Pilih VPC Amazon

Topik ini memberikan informasi terperinci tentang memilih Amazon Virtual Private Cloud (Amazon VPC) ketika Anda melakukan onboard ke Amazon Domain. SageMaker Untuk informasi selengkapnya tentang orientasi ke SageMaker Domain, lihatIkhtisar SageMaker Domain Amazon.

Secara default, SageMaker Domain menggunakan dua VPC Amazon. Satu Amazon VPC dikelola oleh Amazon SageMaker dan menyediakan akses internet langsung. Anda menentukan VPC Amazon lainnya, yang menyediakan lalu lintas terenkripsi antara Domain dan volume Amazon Elastic File System (Amazon EFS) Anda.

Anda dapat mengubah perilaku ini sehingga SageMaker mengirimkan semua lalu lintas melalui VPC Amazon yang Anda tentukan. Ketika Anda memilih opsi ini, Anda harus menyediakan subnet, grup keamanan, dan titik akhir antarmuka yang diperlukan untuk berkomunikasi dengan SageMaker

API dan SageMaker runtime, dan berbagai AWS layanan, seperti Amazon Simple Storage Service (Amazon S3) dan CloudWatch Amazon, yang digunakan oleh Studio.

Saat Anda onboard ke SageMaker Domain, Anda memberi tahu SageMaker untuk mengirim semua lalu lintas melalui VPC Amazon Anda dengan menyetel jenis akses jaringan ke VPC saja.

Untuk menentukan informasi Amazon VPC

Saat Anda menentukan entitas VPC Amazon (yaitu, VPC Amazon, subnet, atau grup keamanan) dalam prosedur berikut, salah satu dari tiga opsi disajikan berdasarkan jumlah entitas yang Anda miliki saat ini. Wilayah AWS Perilakunya adalah sebagai berikut:

- Satu entitas — SageMaker menggunakan entitas itu. Ini tidak dapat diubah.
- Beberapa entitas — Anda harus memilih entitas dari daftar dropdown.
- Tidak ada entitas — Anda harus membuat satu atau beberapa entitas untuk menggunakan Domain. Pilih Buat <entity>untuk membuka konsol VPC di tab browser baru. Setelah Anda membuat entitas, kembali ke halaman Memulai Domain untuk melanjutkan proses orientasi.

Prosedur ini merupakan bagian dari proses orientasi SageMaker Domain Amazon saat Anda memilih Pengaturan untuk organisasi. Informasi Amazon VPC Anda ditentukan di bawah bagian Jaringan.

1. Pilih jenis akses jaringan.

   > **Note**
   >
   > Jika hanya VPC yang dipilih, SageMaker secara otomatis menerapkan pengaturan grup keamanan yang ditentukan untuk Domain ke semua ruang bersama yang dibuat di Domain. Jika hanya Internet publik yang dipilih, SageMaker tidak menerapkan pengaturan grup keamanan ke spasi bersama yang dibuat di Domain.

   - Hanya internet publik — Lalu lintas EFS non-Amazon melewati Amazon VPC yang SageMaker dikelola, yang memungkinkan akses internet. Lalu lintas antara Domain dan volume Amazon EFS Anda adalah melalui Amazon VPC yang ditentukan.
   - Hanya VPC — Semua SageMaker lalu lintas melalui VPC Amazon dan subnet yang ditentukan. Anda harus menggunakan subnet yang tidak memiliki akses internet langsung dalam mode VPC saja. Akses Internet dinonaktifkan secara default.
2. Pilih Amazon VPC.

3. Pilih satu subnet atau beberapa subnet. Jika Anda tidak memilih subnet apa pun, SageMaker gunakan semua subnet di Amazon VPC. Kami menyarankan Anda menggunakan beberapa subnet yang tidak dibuat di Availability Zone yang dibatasi. Menggunakan subnet di Availability Zone yang dibatasi ini dapat mengakibatkan kesalahan kapasitas yang tidak mencukupi dan waktu pembuatan aplikasi yang lebih lama. Untuk informasi selengkapnya tentang Availability Zone, lihat Availability Zone.
4. Pilih grup keamanan. Jika Anda memilih Internet Publik saja, langkah ini opsional. Jika Anda memilih VPC saja, langkah ini diperlukan.

> **Note**
>
> Untuk jumlah maksimum grup keamanan yang diizinkan, lihat UserSettings.

Untuk persyaratan Amazon VPC dalam mode VPC saja, lihat. Connect SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal

Wilayah dan kuota yang didukung

Untuk tipe instans AWS Elastic Compute Cloud (Amazon EC2) SageMaker dan tipe instans Elastic Compute Cloud (Amazon EC2) dan tipe instans Amazon Elastic Compute Cloud (Amazon EC2) dan tipe instans Elastic Compute Cloud (Amazon EC2). SageMaker

Untuk daftar titik akhir SageMaker layanan untuk setiap Wilayah, lihat SageMaker titik akhir dan kuota Amazon di. Referensi Umum AWS

Kuota

Untuk daftar SageMaker kuota, lihat SageMaker titik akhir Amazon dan kuota di. Referensi Umum AWS

Konsol Service Quotas memberikan informasi tentang kuota layanan Anda. Anda dapat menggunakan konsol Service Quotas untuk melihat kuota layanan default Anda atau meminta penambahan kuota. Untuk meminta peningkatan kuota yang dapat disesuaikan, lihat Meminta peningkatan kuota.

Anda dapat menyiapkan templat permintaan kuota untuk AWS Organisasi yang secara otomatis meminta peningkatan kuota selama pembuatan akun. Untuk informasi selengkapnya, lihat Menggunakan templat permintaan Service Quotas.

Gunakan HTML otomatis, tanpa kode, atau kode rendah

Amazon SageMaker menawarkan fitur-fitur berikut untuk mengotomatiskan tugas pembelajaran mesin utama dan menggunakan solusi tanpa kode atau kode rendah.

- Amazon SageMaker Autopilot adalah kumpulan fitur yang mengotomatiskan tugas-tugas utama dari proses pembelajaran mesin otomatis (AutoML). Ini mengeksplorasi data Anda, memilih algoritme yang relevan dengan jenis masalah Anda, dan menyiapkan data untuk memfasilitasi pelatihan dan penyetelan model.
- SageMaker JumpStart menyediakan model sumber terbuka yang sudah terlatih untuk berbagai jenis masalah untuk membantu Anda memulai pembelajaran mesin. Anda dapat melatih dan menyetel model ini secara bertahap sebelum penerapan. JumpStart juga menyediakan templat solusi yang menyiapkan infrastruktur untuk kasus penggunaan umum, dan contoh notebook yang dapat dieksekusi untuk pembelajaran mesin. SageMaker

Topik

SageMaker Autopilot

> **Important**
>
> Pada 30 November 2023, fitur Autopilot bermigrasi ke Amazon SageMaker Canvas sebagai bagian dari pengalaman Studio yang diperbarui, memberikan ilmuwan data kemampuan tanpa kode untuk tugas-tugas seperti persiapan data, rekayasa fitur, pemilihan algoritme, pelatihan dan penyetelan, inferensi, pemantauan model berkelanjutan, dan banyak lagi. SageMaker Canvas mendukung berbagai kasus penggunaan, termasuk visi komputer, peramalan permintaan, pencarian cerdas, dan AI generatif.
> Pengguna Studio Classic dapat terus menggunakan Autopilot sebagai fitur mandiri. Namun, kami mendorong pengguna yang lebih menyukai kenyamanan antarmuka pengguna untuk mengeksplorasi menjalankan tugas SageMaker AutoML mereka di dalam Canvas. Pengguna dengan pengalaman pengkodean dapat terus menggunakan semua instruksi API dan SDK apa pun yang didukung untuk implementasi teknis.

Semua instruksi terkait UI dalam panduan ini berkaitan dengan fitur mandiri Autopilot sebelum bermigrasi ke Amazon Canvas. SageMaker Pengguna yang mengikuti petunjuk ini harus menggunakan Studio Classic.

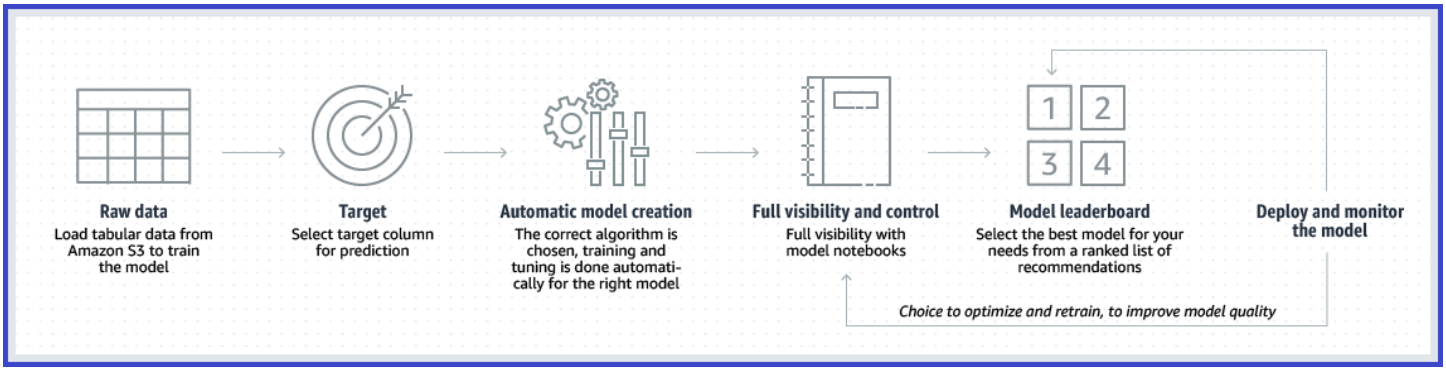
Amazon SageMaker Autopilot adalah rangkaian fitur yang menyederhanakan dan mempercepat berbagai tahapan alur kerja pembelajaran mesin dengan mengotomatiskan proses pembuatan dan penerapan model pembelajaran mesin (AutoML).

Autopilot melakukan tugas-tugas utama berikut yang dapat Anda gunakan pada autopilot atau dengan berbagai tingkat bimbingan manusia:

- Analisis dan pra-pemrosesan data: Autopilot mengidentifikasi jenis masalah spesifik Anda, menangani nilai yang hilang, menormalkan data Anda, memilih fitur, dan secara keseluruhan menyiapkan data untuk pelatihan model.
- Pemilihan model: Autopilot mengeksplorasi berbagai algoritma dan menggunakan teknik resampling validasi silang untuk menghasilkan metrik yang mengevaluasi kualitas prediktif algoritme berdasarkan metrik objektif yang telah ditentukan.
- Optimasi Hyperparameter: Autopilot mengotomatiskan pencarian untuk konfigurasi hyperparameter yang optimal.
- Pelatihan dan evaluasi model: Autopilot mengotomatiskan proses pelatihan dan mengevaluasi berbagai kandidat model. Ini membagi data menjadi set pelatihan dan validasi, melatih kandidat model yang dipilih menggunakan data pelatihan, dan mengevaluasi kinerja mereka pada data tak terlihat dari set validasi. Terakhir, ini memberi peringkat kandidat model yang dioptimalkan berdasarkan kinerja mereka dan mengidentifikasi model berkinerja terbaik.
- Penerapan model: Setelah Autopilot mengidentifikasi model berkinerja terbaik, Autopilot memberikan opsi untuk menerapkan model secara otomatis dengan menghasilkan artefak model dan titik akhir yang mengekspos API. Aplikasi eksternal dapat mengirim data ke titik akhir dan menerima prediksi atau kesimpulan yang sesuai.

Autopilot mendukung pembuatan model pembelajaran mesin pada kumpulan data besar hingga ratusan GB.

Diagram berikut menguraikan tugas-tugas proses AutoML ini yang dikelola oleh Autopilot.



Bergantung pada tingkat kenyamanan Anda dengan proses pembelajaran mesin dan pengalaman pengkodean, Anda dapat menggunakan Autopilot dengan berbagai cara:

- Menggunakan UI Studio Classic, pengguna dapat memilih antara pengalaman tanpa kode atau memiliki beberapa tingkat input manusia.

  > **Note**
  >
  > Hanya eksperimen yang dibuat dari data tabular untuk tipe masalah seperti regresi atau klasifikasi yang tersedia melalui UI Studio Classic.

- Menggunakan AutoML API, pengguna dengan pengalaman pengkodean dapat menggunakan SDK yang tersedia untuk membuat pekerjaan AutoML. Pendekatan ini memberikan fleksibilitas dan opsi penyesuaian yang lebih besar dan tersedia untuk semua jenis masalah.

Autopilot saat ini mendukung jenis masalah berikut:

> **Note**
>
> Untuk masalah regresi atau klasifikasi yang melibatkan data tabular, pengguna dapat memilih di antara dua opsi: menggunakan antarmuka pengguna Studio Classic atau Referensi API. Tugas seperti klasifikasi teks dan gambar, peramalan deret waktu, dan penyempurnaan model bahasa besar tersedia secara eksklusif melalui API Autopilot versi 2. Untuk pengguna Python, sebaiknya gunakan AWS SDK for Python (Boto3)sebagai Amazon SageMaker Python SDK saat ini tidak didukung untuk Autopilot API versi 2.
> Pengguna yang lebih menyukai kenyamanan antarmuka pengguna dapat menggunakan Amazon SageMaker Canvas untuk mengakses model pra-terlatih dan model dasar AI

generatif, atau membuat model khusus yang disesuaikan untuk teks tertentu, klasifikasi gambar, kebutuhan peramalan, atau AI generatif.

- Klasifikasi regresi, biner, dan multikelas dengan data tabular yang diformat sebagai file CSV atau Parket di mana setiap kolom berisi fitur dengan tipe data tertentu dan setiap baris berisi pengamatan. Tipe data kolom yang diterima meliputi numerik, kategoris, teks, dan deret waktu yang terdiri dari string angka yang dipisahkan koma.
  - Untuk membuat pekerjaan Autopilot sebagai percobaan percontohan menggunakan referensi SageMaker API, lihat. Membuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API
  - Untuk membuat pekerjaan Autopilot sebagai percobaan percontohan menggunakan UI Studio Classic, lihat. Membuat eksperimen Autopilot Regresi atau Klasifikasi untuk data tabular menggunakan UI Studio Classic
  - Jika Anda seorang administrator yang ingin melakukan pra-konfigurasi infrastruktur, jaringan, atau parameter keamanan default eksperimen Autopilot di Studio Classic UI, lihat. Konfigurasikan parameter default eksperimen Autopilot (untuk administrator)
- Klasifikasi teks dengan data yang diformat sebagai file CSV atau Parket di mana kolom menyediakan kalimat yang akan diklasifikasikan, sementara kolom lain harus memberikan label kelas yang sesuai. Lihat Membuat pekerjaan AutoML untuk klasifikasi teks menggunakan API.
- Klasifikasi gambar dengan format gambar seperti PNG, JPEG, atau kombinasi keduanya. Lihat. Membuat pekerjaan AutoML untuk klasifikasi gambar menggunakan API
- Peramalan deret waktu dengan data deret waktu yang diformat sebagai file CSV atau Parket. Lihat. Buat pekerjaan AutoML untuk peramalan deret waktu menggunakan API
- Fine-tuning model bahasa besar (LLM) untuk pembuatan teks dengan data yang diformat sebagai file CSV atau Parquet. Lihat. Buat pekerjaan AutoML untuk menyempurnakan model pembuatan teks menggunakan API

Selain itu, Autopilot membantu pengguna memahami bagaimana model membuat prediksi dengan secara otomatis menghasilkan laporan yang menunjukkan pentingnya setiap fitur individu. Ini memberikan transparansi dan wawasan tentang faktor-faktor yang mempengaruhi prediksi, yang dapat digunakan oleh tim risiko dan kepatuhan serta regulator eksternal. Autopilot juga menyediakan laporan kinerja model, yang mencakup ringkasan metrik evaluasi, matriks kebingungan, berbagai visualisasi seperti kurva karakteristik operasi penerima dan kurva penarikan presisi, dan banyak lagi. Isi spesifik dari setiap laporan bervariasi tergantung pada jenis masalah eksperimen Autopilot.

Laporan penjelasan dan kinerja untuk kandidat model terbaik dalam eksperimen Autopilot tersedia untuk jenis masalah klasifikasi data teks, gambar, dan tabel.

Untuk kasus penggunaan data tabular seperti regresi atau klasifikasi, Autopilot menawarkan visibilitas tambahan tentang bagaimana data diperdebatkan dan bagaimana kandidat model dipilih, dilatih, dan disetel dengan menghasilkan notebook yang berisi kode yang digunakan untuk mengeksplorasi data dan menemukan model berkinerja terbaik. Notebook ini menyediakan lingkungan interaktif dan eksplorasi untuk membantu Anda mempelajari tentang dampak berbagai input atau trade-off yang dibuat dalam eksperimen. Anda dapat bereksperimen lebih lanjut dengan kandidat model berkinerja lebih tinggi dengan membuat modifikasi Anda sendiri pada eksplorasi data dan notebook definisi kandidat yang disediakan oleh Autopilot.

Dengan Amazon SageMaker, Anda hanya membayar untuk apa yang Anda gunakan. Anda membayar sumber daya komputasi dan penyimpanan yang mendasarinya di dalam SageMaker atau AWS layanan lain, berdasarkan penggunaan Anda. Untuk informasi selengkapnya tentang biaya penggunaan SageMaker, lihat SageMakerHarga Amazon.

Topik

- Membuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API
- Membuat pekerjaan AutoML untuk klasifikasi gambar menggunakan API
- Membuat pekerjaan AutoML untuk klasifikasi teks menggunakan API
- Buat pekerjaan AutoML untuk peramalan deret waktu menggunakan API
- Buat pekerjaan AutoML untuk menyempurnakan model pembuatan teks menggunakan API
- Membuat eksperimen Autopilot Regresi atau Klasifikasi untuk data tabular menggunakan UI Studio Classic
- Notebook SageMaker contoh Amazon Autopilot
- Kuota Amazon SageMaker Autopilot
- Panduan Referensi API untuk Amazon SageMaker Autopilot

Membuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API

Anda dapat membuat eksperimen Autopilot untuk data tabular secara terprogram dengan memanggil tindakan `CreateAutoMLJobV2`API dalam bahasa apa pun yang didukung oleh Autopilot atau. AWS CLI

Untuk informasi tentang cara tindakan API ini diterjemahkan ke dalam fungsi dalam bahasa pilihan Anda, lihat bagian Lihat Juga `CreateAutoMLJobV2` dan pilih SDK. Sebagai contoh, untuk pengguna Python, lihat sintaks permintaan lengkap dari in. `create_auto_ml_job_v2` AWS SDK for Python (Boto3)

> **Note**
>
> CreateAutoMLJobv2 dan DescribeAutoMLJobv2 adalah versi baru CreateAutoMLJob dan DescribeAutoMLJob yang menawarkan kompatibilitas mundur.
> Kami merekomendasikan menggunakan`CreateAutoMLJobV2`. `CreateAutoMLJobV2`dapat mengelola jenis masalah tabel yang identik dengan versi sebelumnya`CreateAutoMLJob`, serta jenis masalah non-tabular seperti klasifikasi gambar atau teks, atau peramalan deret waktu.

Minimal, semua eksperimen pada data tabular memerlukan spesifikasi nama eksperimen, menyediakan lokasi untuk data input dan output, dan menentukan data target mana yang akan diprediksi. Secara opsional, Anda juga dapat menentukan jenis masalah yang ingin Anda selesaikan (regresi, klasifikasi, klasifikasi multiclass), pilih strategi pemodelan Anda (ansambel bertumpuk atau optimasi hiperparameter), pilih daftar algoritma yang digunakan oleh pekerjaan Autopilot untuk melatih data, dan banyak lagi.

Setelah eksperimen berjalan, Anda dapat membandingkan uji coba dan mempelajari detail langkah pra-pemrosesan, algoritme, dan rentang hiperparameter dari setiap model. Anda juga memiliki opsi untuk mengunduh laporan penjelasan dan kinerjanya. Gunakan buku catatan yang disediakan untuk melihat hasil eksplorasi data otomatis atau definisi model kandidat.

Berikut ini adalah kumpulan parameter permintaan input wajib dan opsional untuk tindakan `CreateAutoMLJobV2` API. Anda dapat menemukan informasi alternatif untuk versi sebelumnya dari tindakan ini,`CreateAutoMLJob`. Namun, kami sarankan untuk menggunakan`CreateAutoMLJobV2`.

Temukan panduan tentang cara memigrasi `CreateAutoMLJob` ke `CreateAutoMLJobV2` inMigrasikan CreateAuto MLJob ke MLJobv2 CreateAuto.

Parameter yang diperlukan

CreateAutoMLJobV2

Saat menelepon `CreateAutoMLJobV2` untuk membuat eksperimen Autopilot untuk data tabular, Anda harus memberikan nilai berikut:

- An `AutoMLJobName` untuk menentukan nama pekerjaan Anda.
- Setidaknya satu `AutoMLJobChannel` `AutoMLJobInputDataConfig` untuk menentukan sumber data Anda.
- Baik `AutoMLJobObjective` metrik dan jenis masalah pembelajaran terawasi pilihan Anda (klasifikasi biner, klasifikasi multikelas, regresi) di`AutoMLProblemTypeConfig`, atau tidak sama sekali. Untuk data tabular, Anda harus memilih `TabularJobConfig` sebagai jenis. `AutoMLProblemTypeConfig` Anda mengatur masalah pembelajaran yang diawasi dalam `ProblemType` atribut. `TabularJobConfig`
- `OutputDataConfig`Untuk menentukan jalur keluaran Amazon S3 untuk menyimpan artefak pekerjaan AutoML Anda.
- A `RoleArn` untuk menentukan ARN dari peran yang digunakan untuk mengakses data Anda.

CreateAutoMLJob

Saat memanggil `CreateAutoMLJob` untuk membuat eksperimen AutoML, Anda harus memberikan empat nilai berikut:

- An `AutoMLJobName` untuk menentukan nama pekerjaan Anda.
- Setidaknya satu `AutoMLChannel` `InputDataConfig` untuk menentukan sumber data Anda.
- `OutputDataConfig`Untuk menentukan jalur keluaran Amazon S3 untuk menyimpan artefak pekerjaan AutoML Anda.
- A `RoleArn` untuk menentukan ARN dari peran yang digunakan untuk mengakses data Anda.

Semua parameter lainnya adalah opsional.

Parameter opsional

Bagian berikut memberikan detail beberapa parameter opsional yang dapat Anda teruskan ke tindakan `CreateAutoMLJobV2` API saat menggunakan data tabular. Anda dapat menemukan informasi alternatif untuk versi sebelumnya dari tindakan ini,`CreateAutoMLJob`. Namun, kami sarankan untuk menggunakan`CreateAutoMLJobV2`.

Cara mengatur mode pelatihan pekerjaan AutoML

Untuk data tabular, kumpulan algoritme yang dijalankan pada data Anda untuk melatih kandidat model Anda bergantung pada strategi pemodelan Anda (`ENSEMBLING`atau`HYPERPARAMETER_TUNING`). Berikut ini detail cara mengatur mode pelatihan ini.

Jika Anda tetap kosong (ataunull), Mode disimpulkan berdasarkan ukuran kumpulan data Anda.

Untuk informasi tentang ansambel bertumpuk Autopilot dan metode pelatihan optimasi hiperparameter, lihat Mode pelatihan dan dukungan algoritme

CreateAutoMLJobV2

Untuk data tabular, Anda harus memilih TabularJobConfig sebagai jenis. AutoMLProblemTypeConfig

Anda dapat mengatur metode pelatihan pekerjaan AutoML V2 dengan parameter. TabularJobConfig.Mode

CreateAutoMLJob

Anda dapat mengatur metode pelatihan pekerjaan AutoML dengan parameter. AutoMLJobConfig.Mode

Cara memilih fitur dan algoritme untuk melatih pekerjaan AutoML

Pilihan fitur

Autopilot menyediakan langkah-langkah pra-pemrosesan data otomatis termasuk pemilihan fitur dan ekstraksi fitur. Namun, Anda dapat secara manual menyediakan fitur yang akan digunakan dalam pelatihan dengan FeatureSpecificatioS3Uri atribut.

Fitur yang dipilih harus terkandung dalam file JSON dalam format berikut:

```
{ "FeatureAttributeNames":["col1", "col2", ...] }
```

Nilai yang tercantum dalam ["col1", "col2", ...] peka huruf besar/kecil. Mereka harus berupa daftar string yang berisi nilai unik yang merupakan himpunan bagian dari nama kolom dalam data input.

> **Note**
>
> Daftar kolom yang disediakan sebagai fitur tidak dapat menyertakan kolom target.

CreateAutoMLJobV2

Untuk data tabular, Anda harus memilih `TabularJobConfig` sebagai jenis. `AutoMLProblemTypeConfig`

Anda dapat mengatur URL ke fitur yang Anda pilih dengan `TabularJobConfig.FeatureSpecificatioS3Uri` parameter.

CreateAutoMLJob

Anda dapat mengatur `FeatureSpecificatioS3Uri` atribut AutoMl CandidateGenerationConfig dalam CreateAutoMLJob API dengan format berikut:

```
{
    "AutoMLJobConfig": {
        "CandidateGenerationConfig": {
            "FeatureSpecificationS3Uri":"string"
            },
       }
  }
```

Pemilihan algoritma

Secara default, pekerjaan Autopilot Anda menjalankan daftar algoritme yang telah ditentukan sebelumnya pada kumpulan data Anda untuk melatih kandidat model. Daftar algoritma tergantung pada mode pelatihan (`ENSEMBLING`atau`HYPERPARAMETER_TUNING`) yang digunakan oleh pekerjaan.

Anda dapat memberikan subset dari pemilihan algoritme default.

CreateAutoMLJobV2

Untuk data tabular, Anda harus memilih `TabularJobConfig` sebagai jenis. `AutoMLProblemTypeConfig`

Anda dapat menentukan array yang dipilih `AutoMLAlgorithms` dalam `AlgorithmsConfig` atribut CandidateGenerationConfig.

Berikut ini adalah contoh `AlgorithmsConfig` atribut yang mencantumkan tepat tiga algoritma (“xgboost”, “fasai”, “catboost”) di `AutoMLAlgorithms` bidangnya untuk mode pelatihan ensembling.

```
{
   "AutoMLProblemTypeConfig": {
        "TabularJobConfig": {
          "Mode": "ENSEMBLING",
          "CandidateGenerationConfig": {
            "AlgorithmsConfig":[
               {"AutoMLAlgorithms":["xgboost", "fastai", "catboost"]}
            ]
         },
       },
     },
  }
```

CreateAutoMLJob

Anda dapat menentukan array yang dipilih `AutoMLAlgorithms` dalam `AlgorithmsConfig` atribut CandidateGenerationConfigAutoML.

Berikut ini adalah contoh `AlgorithmsConfig` atribut yang mencantumkan tepat tiga algoritma (“xgboost”, “fasai”, “catboost”) di `AutoMLAlgorithms` bidangnya untuk mode pelatihan ensembling.

```
{
   "AutoMLJobConfig": {
        "CandidateGenerationConfig": {
            "AlgorithmsConfig":[
               {"AutoMLAlgorithms":["xgboost", "fastai", "catboost"]}
            ]
         },
     "Mode": "ENSEMBLING"
  }
```

Untuk daftar algoritma yang tersedia per pelatihan`Mode`, lihat `AutoMLAlgorithms`. Untuk detail tentang setiap algoritma, lihatMode pelatihan dan dukungan algoritme.

Cara menentukan kumpulan data pelatihan dan validasi pekerjaan AutoML

Anda dapat memberikan kumpulan data validasi dan rasio pemisahan data khusus Anda sendiri, atau membiarkan Autopilot membagi kumpulan data secara otomatis.

CreateAutoMLJobV2

Setiap `AutoMLJobChannel` objek (lihat parameter AutoML yang diperlukanJobInputDataConfig) memilikiChannelType, yang dapat diatur ke salah satu `training` atau `validation` nilai yang menentukan bagaimana data akan digunakan saat membangun model pembelajaran mesin. Setidaknya satu sumber data harus disediakan dan maksimal dua sumber data diperbolehkan: satu untuk data pelatihan dan satu untuk data validasi.

Bagaimana Anda membagi data menjadi kumpulan data pelatihan dan validasi tergantung pada apakah Anda memiliki satu atau dua sumber data.

- Jika Anda hanya memiliki satu sumber data, `ChannelType` diatur ke secara `training` default dan harus memiliki nilai ini.
  - Jika `ValidationFraction` nilai dalam tidak `AutoMLDataSplitConfig` disetel, 0.2 (20%) data dari sumber ini digunakan untuk validasi secara default.
  - Jika `ValidationFraction` diatur ke nilai antara 0 dan 1, dataset dibagi berdasarkan nilai yang ditentukan, di mana nilai menentukan fraksi dari dataset yang digunakan untuk validasi.
- Jika Anda memiliki dua sumber data, `ChannelType` salah satu `AutoMLJobChannel` objek harus diatur ke`training`, nilai default. Sumber data lainnya harus diatur ke`validation`. `ChannelType` Kedua sumber data harus memiliki format yang sama, baik CSV atau Parket, dan skema yang sama. Anda tidak boleh menetapkan nilai untuk `ValidationFraction` dalam kasus ini karena semua data dari setiap sumber digunakan untuk pelatihan atau validasi. Menyetel nilai ini menyebabkan kesalahan.

CreateAutoMLJob

Setiap `AutoMLChannel` objek (lihat parameter yang diperlukan InputDataConfig) memilikiChannelType, yang dapat diatur ke salah satu `training` atau `validation` nilai yang menentukan bagaimana data akan digunakan saat membangun model pembelajaran mesin. Setidaknya satu sumber data harus disediakan dan maksimal dua sumber data diperbolehkan: satu untuk data pelatihan dan satu untuk data validasi.

Bagaimana Anda membagi data menjadi kumpulan data pelatihan dan validasi tergantung pada apakah Anda memiliki satu atau dua sumber data.

- Jika Anda hanya memiliki satu sumber data, `ChannelType` diatur ke secara `training` default dan harus memiliki nilai ini.

- Jika `ValidationFraction` nilai dalam tidak `AutoMLDataSplitConfig`disetel, 0.2 (20%) data dari sumber ini digunakan untuk validasi secara default.
- Jika `ValidationFraction` diatur ke nilai antara 0 dan 1, dataset dibagi berdasarkan nilai yang ditentukan, di mana nilai menentukan fraksi dari dataset yang digunakan untuk validasi.

- Jika Anda memiliki dua sumber data, `ChannelType` salah satu `AutoMLChannel` objek harus diatur ke`training`, nilai default. Sumber data lainnya harus diatur ke`validation`. `ChannelType` Kedua sumber data harus memiliki format yang sama, baik CSV atau Parket, dan skema yang sama. Anda tidak boleh menetapkan nilai untuk `ValidationFraction` dalam kasus ini karena semua data dari setiap sumber digunakan untuk pelatihan atau validasi. Menyetel nilai ini menyebabkan kesalahan.

Untuk informasi tentang validasi split dan cross-validasi di Autopilot lihat. Validasi silang di Autopilot

Cara mengatur jenis masalah pekerjaan AutoML

CreateAutoMLJobV2

Untuk data tabular, Anda harus memilih `TabularJobConfig` sebagai jenis. `AutoMLProblemTypeConfig`

Anda selanjutnya dapat menentukan jenis masalah pembelajaran yang diawasi (klasifikasi biner, klasifikasi multiclass, regresi) yang tersedia untuk kandidat model pekerjaan AutoML Anda V2 dengan parameter. `TabularJobConfig.ProblemType`

CreateAutoMLJob

Anda dapat mengatur jenis masalah pada pekerjaan AutoML dengan parameter. `CreateAutoPilot.ProblemType` Ini membatasi jenis preprocessing dan algoritma yang Autopilot coba. Setelah pekerjaan selesai, jika Anda telah mengatur`CreateAutoPilot.ProblemType`, maka `ResolvedAttribute.ProblemType` kecocokan yang `ProblemType` Anda tetapkan. Jika Anda menyimpannya kosong (ataunull), `ProblemType` disimpulkan atas nama Anda.

> **Note**
>
> Dalam beberapa kasus, Autopilot tidak dapat menyimpulkan `ProblemType` dengan kepercayaan diri yang cukup tinggi, dalam hal ini Anda harus memberikan nilai agar pekerjaan berhasil.

Cara menambahkan bobot sampel ke pekerjaan AutoML

Anda dapat menambahkan kolom bobot sampel ke kumpulan data tabular Anda dan kemudian meneruskannya ke pekerjaan AutoML Anda untuk meminta baris kumpulan data untuk ditimbang selama pelatihan dan evaluasi.

Support untuk bobot sampel hanya tersedia dalam mode ensembling. Bobot Anda harus numerik dan non-negatif. Poin data dengan nilai bobot tidak valid atau tidak ada dikecualikan. Untuk informasi selengkapnya tentang metrik objektif yang tersedia, lihatMetrik tertimbang autopilot.

CreateAutoMLJobV2

Untuk data tabular, Anda harus memilih `TabularJobConfig` sebagai jenis. `AutoMLProblemTypeConfig`

Untuk mengatur bobot sampel saat membuat eksperimen (lihat CreateAutoMLJobv2), Anda dapat meneruskan nama kolom bobot sampel di `SampleWeightAttributeName` atribut objek. `TabularJobConfig` Ini memastikan bahwa metrik objektif Anda menggunakan bobot untuk pelatihan, evaluasi, dan pemilihan kandidat model.

CreateAutoMLJob

Untuk menetapkan bobot sampel saat membuat eksperimen (lihat CreateAutoMLJob), Anda dapat meneruskan nama kolom bobot sampel di `SampleWeightAttributeName` atribut objek AutomlChannel. Ini memastikan bahwa metrik objektif Anda menggunakan bobot untuk pelatihan, evaluasi, dan pemilihan kandidat model.

Migrasikan CreateAuto MLJob ke MLJobv2 CreateAuto

Kami menyarankan pengguna `CreateAutoMLJob` untuk bermigrasi ke`CreateAutoMLJobV2`.

Bagian ini menjelaskan perbedaan parameter input antara CreateAutoMLJob dan CreateAutoMLJobv2 dengan menyoroti perubahan posisi, nama, atau struktur objek dan atribut permintaan input antara dua versi.

- Minta atribut yang tidak berubah antar versi.

```
{
   "AutoMLJobName": "string",
   "AutoMLJobObjective": {
      "MetricName": "string"
   },
   "ModelDeployConfig": {
      "AutoGenerateEndpointName": boolean,
      "EndpointName": "string"
   },
   "OutputDataConfig": {
      "KmsKeyId": "string",
      "S3OutputPath": "string"
   },
   "RoleArn": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

- Minta atribut yang mengubah posisi dan struktur antar versi.

  Atribut berikut berubah posisi:`DataSplitConfig`,`Security Config`,`CompletionCriteria`,`Mode`,`FeatureSpecificationS3Uri`,`SampleWeightAttributeNam`

  CreateAutoMLJob

```
{
    "AutoMLJobConfig": {
        "Mode": "string",
        "CompletionCriteria": {
            "MaxAutoMLJobRuntimeInSeconds": number,
            "MaxCandidates": number,
            "MaxRuntimePerTrainingJobInSeconds": number
        },
```

```
        "DataSplitConfig": {
            "ValidationFraction": number
        },
        "SecurityConfig": {
            "EnableInterContainerTrafficEncryption": boolean,
            "VolumeKmsKeyId": "string",
            "VpcConfig": {
            "SecurityGroupIds": [ "string" ],
            "Subnets": [ "string" ]
            }
        },
        "CandidateGenerationConfig": {
            "FeatureSpecificationS3Uri": "string"
        }
    },
    "GenerateCandidateDefinitionsOnly": boolean,
    "ProblemType": "string"
}
```

CreateAutoMLJobV2

```
{
    "AutoMLProblemTypeConfig": {
        "TabularJobConfig": {
            "Mode": "string",
            "ProblemType": "string",
            "GenerateCandidateDefinitionsOnly": boolean,
            "CompletionCriteria": {
                "MaxAutoMLJobRuntimeInSeconds": number,
                "MaxCandidates": number,
                "MaxRuntimePerTrainingJobInSeconds": number
            },
            "FeatureSpecificationS3Uri": "string",
            "SampleWeightAttributeName": "string",
            "TargetAttributeName": "string"
        }
    },
    "DataSplitConfig": {
        "ValidationFraction": number
    },
    "SecurityConfig": {
        "EnableInterContainerTrafficEncryption": boolean,
        "VolumeKmsKeyId": "string",
```

```
        "VpcConfig": {
            "SecurityGroupIds": [ "string" ],
            "Subnets": [ "string" ]
        }
    }
}
```

- Atribut berikut mengubah posisi dan struktur antar versi.

  Berikut JSON menggambarkan bagaimana AutoML. JobConfig CandidateGenerationConfigdari tipe AutoML CandidateGenerationConfig dipindahkan ke AutoML. ProblemTypeConfig TabularJobConfig. CandidateGenerationConfigdari tipe CandidateGenerationConfigdi V2.

  CreateAutoMLJob

```
{
   "AutoMLJobConfig": {
      "CandidateGenerationConfig": {
         "AlgorithmsConfig": [
            {
               "AutoMLAlgorithms": [ "string" ]
            }
         ],
         "FeatureSpecificationS3Uri": "string"
      }
}
```

  CreateAutoMLJobV2

```
{
    "AutoMLProblemTypeConfig": {
        "TabularJobConfig": {
            "CandidateGenerationConfig": {
                "AlgorithmsConfig": [
                    {
                    "AutoMLAlgorithms": [ "string" ]
                    }
                ],
            },
        }
    },
}
```

- Minta atribut yang mengubah nama dan struktur.

  JSON berikut mengilustrasikan bagaimana InputDataConfig(Sebuah array dari AutoMLChannel) berubah menjadi AutoML(JobInputDataConfigSebuah array dari AutoML) di V2. JobChannel
  Perhatikan bahwa atribut `SampleWeightAttributeName` dan `TargetAttributeName` bergerak keluar `InputDataConfig` dan masuk`AutoMLProblemTypeConfig`.

  CreateAutoMLJob

```
{
    "InputDataConfig": [
        {
            "ChannelType": "string",
            "CompressionType": "string",
            "ContentType": "string",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "string",
                    "S3Uri": "string"
                }
            },
            "SampleWeightAttributeName": "string",
            "TargetAttributeName": "string"
        }
    ]
}
```

  CreateAutoMLJobV2

```
{
    "AutoMLJobInputDataConfig": [
        {
            "ChannelType": "string",
            "CompressionType": "string",
            "ContentType": "string",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "string",
                    "S3Uri": "string"
                }
            }
        }
    ]
```

```
}
```

Kumpulan data autopilot dan jenis masalah

Untuk data tabular (yaitu data di mana setiap kolom berisi fitur dengan tipe data tertentu dan setiap baris berisi pengamatan), Autopilot memberi Anda opsi untuk menentukan jenis masalah pembelajaran yang diawasi yang tersedia untuk kandidat model pekerjaan AutoML, seperti klasifikasi biner atau regresi, atau mendeteksinya atas nama Anda berdasarkan data yang Anda berikan.

Topik

- Kumpulan data autopilot, tipe data, dan format
- Jenis masalah autopilot

Kumpulan data autopilot, tipe data, dan format

Autopilot mendukung data tabular yang diformat sebagai file CSV atau sebagai file Parket: setiap kolom berisi fitur dengan tipe data tertentu dan setiap baris berisi pengamatan. Properti dari dua format file ini sangat berbeda.

- CSV (comma-separated-values) adalah format file berbasis baris yang menyimpan data dalam teks biasa yang dapat dibaca manusia yang merupakan pilihan populer untuk pertukaran data karena didukung oleh berbagai aplikasi.
- Parket adalah format file berbasis kolom di mana data disimpan dan diproses lebih efisien daripada format file berbasis baris. Ini membuat mereka menjadi pilihan yang lebih baik untuk masalah data besar.

Tipe data yang diterima untuk kolom termasuk numerik, kategoris, teks, dan deret waktu yang terdiri dari string angka yang dipisahkan koma. Jika Autopilot mendeteksi itu berurusan dengan urutan deret waktu, ia memprosesnya melalui transformator fitur khusus yang disediakan oleh perpustakaan tsfresh. Pustaka ini mengambil deret waktu sebagai input dan output fitur seperti nilai absolut tertinggi dari deret waktu atau statistik deskriptif pada autokorelasi. Fitur keluaran ini kemudian digunakan sebagai input ke salah satu dari tiga jenis masalah.

Autopilot mendukung pembuatan model pembelajaran mesin pada kumpulan data besar hingga ratusan GB. Untuk detail tentang batas sumber daya default untuk kumpulan data input dan cara meningkatkannya, lihat Kuota Autopilot.

Jenis masalah autopilot

Untuk data tabular, Anda lebih lanjut menentukan jenis masalah pembelajaran yang diawasi yang tersedia untuk kandidat model sebagai berikut:

Regresi

Regresi memperkirakan nilai variabel target dependen berdasarkan satu atau lebih variabel atau atribut lain yang berkorelasi dengannya. Contohnya adalah prediksi harga rumah menggunakan fitur seperti jumlah kamar mandi dan kamar tidur, luas persegi rumah dan taman. Analisis regresi dapat membuat model yang mengambil satu atau lebih fitur ini sebagai input dan memprediksi harga rumah.

Klasifikasi biner

Klasifikasi biner adalah jenis pembelajaran yang diawasi yang menetapkan individu ke salah satu dari dua kelas yang telah ditentukan dan saling eksklusif berdasarkan atribut mereka. Ini diawasi karena model dilatih menggunakan contoh di mana atribut disediakan dengan objek berlabel dengan benar. Diagnosis medis untuk apakah seseorang memiliki penyakit atau tidak berdasarkan hasil tes diagnostik adalah contoh klasifikasi biner.

Klasifikasi multiclass

Klasifikasi multiclass adalah jenis pembelajaran yang diawasi yang menugaskan seorang individu ke salah satu dari beberapa kelas berdasarkan atributnya. Ini diawasi karena model dilatih menggunakan contoh di mana atribut disediakan dengan objek berlabel dengan benar. Contohnya adalah prediksi topik yang paling relevan dengan dokumen teks. Sebuah dokumen dapat diklasifikasikan sebagai tentang, katakanlah, agama atau politik atau keuangan, atau tentang salah satu dari beberapa kelas topik yang telah ditentukan sebelumnya.

Mode pelatihan dan dukungan algoritme

Autopilot mendukung berbagai mode pelatihan dan algoritma untuk mengatasi masalah pembelajaran mesin, melaporkan metrik kualitas dan objektif, dan menggunakan validasi silang secara otomatis, bila diperlukan.

Mode pelatihan

SageMaker Autopilot dapat secara otomatis memilih metode pelatihan berdasarkan ukuran dataset, atau Anda dapat memilihnya secara manual. Pilihannya adalah sebagai berikut:

- Ensembling — Autopilot menggunakan AutoGluonperpustakaan untuk melatih beberapa model dasar. Untuk menemukan kombinasi terbaik untuk kumpulan data Anda, mode ansambel menjalankan 10 uji coba dengan pengaturan model dan parameter meta yang berbeda. Kemudian Autopilot menggabungkan model-model ini menggunakan metode ansambel susun untuk membuat model prediktif yang optimal. Untuk daftar algoritma yang didukung Autopilot dalam mode ensembling untuk data tabular, lihat bagian dukungan Algoritma berikut.
- Optimasi Hyperparameter (HPO) — Autopilot menemukan versi terbaik dari model dengan menyetel hyperparameters menggunakan optimasi Bayesian atau optimasi multi-fidelity saat menjalankan pekerjaan pelatihan pada dataset Anda. Mode HPO memilih algoritme yang paling relevan dengan kumpulan data Anda dan memilih rentang hiperparameter terbaik untuk menyetel model Anda. Untuk menyetel model Anda, mode HPO menjalankan hingga 100 uji coba (default) untuk menemukan pengaturan hiperparameter optimal dalam rentang yang dipilih. Jika ukuran dataset Anda kurang dari 100 MB, Autopilot menggunakan optimasi Bayesian. Autopilot memilih optimasi multi-fidelity jika dataset Anda lebih besar dari 100 MB.

  Dalam optimasi multi-fidelity, metrik terus dipancarkan dari wadah pelatihan. Uji coba yang berkinerja buruk terhadap metrik objektif yang dipilih dihentikan lebih awal. Uji coba yang berkinerja baik dialokasikan lebih banyak sumber daya.

  Untuk daftar algoritma yang didukung Autopilot dalam mode HPO, lihat bagian dukungan Algoritma berikut.
- Otomatis — Autopilot secara otomatis memilih mode ensembling atau mode HPO berdasarkan ukuran dataset Anda. Jika dataset Anda lebih besar dari 100 MB, Autopilot memilih HPO. Jika tidak, ia memilih mode ansambel. Autopilot dapat gagal membaca ukuran kumpulan data Anda dalam kasus berikut.
  - Jika Anda mengaktifkan mode Virtual Private Cloud (VPC), untuk pekerjaan AutoML tetapi bucket S3 yang berisi kumpulan data hanya mengizinkan akses dari VPC.
  - Input S3 DataType dari dataset Anda adalah a. `ManifestFile`
  - Masukan S3Uri berisi lebih dari 1000 item.

  Jika Autopilot tidak dapat membaca ukuran dataset Anda, default memilih mode HPO.

> **Note**
>
> Untuk runtime dan kinerja yang optimal, gunakan mode pelatihan ansambel untuk kumpulan data yang lebih kecil dari 100 MB.

Dukungan algoritma

Dalam mode HPO, Autopilot mendukung jenis algoritma pembelajaran mesin berikut:

- Linear learner — Algoritma pembelajaran yang diawasi yang dapat memecahkan masalah klasifikasi atau regresi.
- XGBoost — Algoritma pembelajaran yang diawasi yang mencoba memprediksi variabel target secara akurat dengan menggabungkan ansambel perkiraan dari serangkaian model yang lebih sederhana dan lebih lemah.
- Algoritma pembelajaran mendalam — Perceptron multilayer (MLP) dan jaringan saraf tiruan feedforward. Algoritma ini dapat menangani data yang tidak dapat dipisahkan secara linier.

> **Note**
>
> Anda tidak perlu menentukan algoritma yang akan digunakan untuk masalah pembelajaran mesin Anda. Autopilot secara otomatis memilih algoritma yang sesuai untuk dilatih.

Dalam mode ansambel, Autopilot mendukung jenis algoritma pembelajaran mesin berikut:

- LightGBM - Kerangka kerja yang dioptimalkan yang menggunakan algoritma berbasis pohon dengan peningkatan gradien. Algoritma ini menggunakan pohon yang tumbuh dalam lebar, bukan kedalaman, dan sangat dioptimalkan untuk kecepatan.
- CatBoost— Kerangka kerja yang menggunakan algoritme berbasis pohon dengan peningkatan gradien. Dioptimalkan untuk menangani variabel kategoris.
- XGBoost — Kerangka kerja yang menggunakan algoritme berbasis pohon dengan peningkatan gradien yang tumbuh secara mendalam, bukan luasnya.
- Random Forest — Algoritma berbasis pohon yang menggunakan beberapa pohon keputusan pada sub-sampel acak data dengan penggantian. Pohon-pohon dibagi menjadi simpul optimal di setiap tingkat. Keputusan setiap pohon dirata-ratakan bersama untuk mencegah overfitting dan meningkatkan prediksi.

- Extra Trees — Algoritma berbasis pohon yang menggunakan beberapa pohon keputusan pada seluruh kumpulan data. Pohon-pohon dibelah secara acak di setiap tingkat. Keputusan setiap pohon dirata-ratakan untuk mencegah overfitting dan untuk meningkatkan prediksi. Pohon tambahan menambahkan tingkat pengacakan dibandingkan dengan algoritma hutan acak.
- Model Linear — Kerangka kerja yang menggunakan persamaan linier untuk memodelkan hubungan antara dua variabel dalam data yang diamati.
- Obor jaringan saraf — Model jaringan saraf yang diimplementasikan menggunakan Pytorch.
- Neural network fast.ai — Model jaringan saraf yang diimplementasikan menggunakan fast.ai.

Metrik dan validasi

Panduan ini menunjukkan metrik dan teknik validasi yang dapat Anda gunakan untuk mengukur kinerja model pembelajaran mesin. Amazon SageMaker Autopilot menghasilkan metrik yang mengukur kualitas prediktif kandidat model pembelajaran mesin. Metrik yang dihitung untuk kandidat ditentukan menggunakan array MetricDatumtipe.

Metrik Autopilot

Daftar berikut berisi nama-nama metrik yang saat ini tersedia untuk mengukur kinerja model dalam Autopilot.

> **Note**
>
> Autopilot mendukung bobot sampel. Untuk mempelajari lebih lanjut tentang bobot sampel dan metrik objektif yang tersedia, lihat. Metrik tertimbang autopilot

Berikut ini adalah metrik yang tersedia.

`Accuracy`

Rasio jumlah item yang diklasifikasikan dengan benar dengan jumlah total item yang diklasifikasikan (benar dan salah). Ini digunakan untuk klasifikasi biner dan multiclass. Akurasi mengukur seberapa dekat nilai kelas yang diprediksi dengan nilai aktual. Nilai untuk metrik akurasi bervariasi antara nol (0) dan satu (1). Nilai 1 menunjukkan akurasi sempurna, dan 0 menunjukkan ketidakakuratan sempurna.

AUC

Metrik area under the curve (AUC) digunakan untuk membandingkan dan mengevaluasi klasifikasi biner dengan algoritma yang mengembalikan probabilitas, seperti regresi logistik. Untuk memetakan probabilitas ke dalam klasifikasi, ini dibandingkan dengan nilai ambang batas.

Kurva yang relevan adalah kurva karakteristik operasi penerima. Kurva memplot tingkat positif sebenarnya (TPR) prediksi (atau recall) terhadap tingkat positif palsu (FPR) sebagai fungsi dari nilai ambang batas, di atasnya prediksi dianggap positif. Meningkatkan ambang menghasilkan lebih sedikit positif palsu, tetapi lebih banyak negatif palsu.

AUC adalah area di bawah kurva karakteristik operasi penerima ini. Oleh karena itu, AUC memberikan ukuran agregat dari kinerja model di semua ambang batas klasifikasi yang mungkin. Skor AUC bervariasi antara 0 dan 1. Skor 1 menunjukkan akurasi sempurna, dan skor satu setengah (0,5) menunjukkan bahwa prediksi tidak lebih baik daripada pengklasifikasi acak.

BalancedAccuracy

`BalancedAccuracy`adalah metrik yang mengukur rasio prediksi akurat untuk semua prediksi. Rasio ini dihitung setelah menormalkan positif sejati (TP) dan negatif sejati (TN) dengan jumlah total nilai positif (P) dan negatif (N). Ini digunakan dalam klasifikasi biner dan multiclass dan didefinisikan sebagai berikut: 0,5* ((TP/P) + (TN/N)), dengan nilai mulai dari 0 hingga 1. `BalancedAccuracy`memberikan ukuran akurasi yang lebih baik ketika jumlah positif atau negatif sangat berbeda satu sama lain dalam kumpulan data yang tidak seimbang, seperti ketika hanya 1% email adalah spam.

F1

`F1`Skor adalah rata-rata harmonik dari presisi dan ingatan, didefinisikan sebagai berikut: F1 = 2 * (presisi * recall)/(presisi + recall). Ini digunakan untuk klasifikasi biner ke dalam kelas yang secara tradisional disebut sebagai positif dan negatif. Prediksi dikatakan benar ketika mereka cocok dengan kelas mereka yang sebenarnya (benar), dan salah ketika tidak.

Presisi adalah rasio prediksi positif sejati untuk semua prediksi positif, dan itu termasuk positif palsu dalam kumpulan data. Presisi mengukur kualitas prediksi ketika memprediksi kelas positif.

Ingat (atau sensitivitas) adalah rasio prediksi positif sejati untuk semua contoh positif aktual. Ingat mengukur seberapa lengkap model memprediksi anggota kelas yang sebenarnya dalam kumpulan data.

Skor F1 bervariasi antara 0 dan 1. Skor 1 menunjukkan kinerja terbaik, dan 0 menunjukkan yang terburuk.

`F1macro`

`F1macro`Skor tersebut menerapkan penilaian F1 untuk masalah klasifikasi multiclass. Hal ini dilakukan dengan menghitung presisi dan recall, dan kemudian mengambil mean harmonik mereka untuk menghitung skor F1 untuk setiap kelas. Terakhir, `F1macro` rata-rata skor individu untuk mendapatkan skor. `F1macro` `F1macro`Skor bervariasi antara 0 dan 1. Skor 1 menunjukkan kinerja terbaik, dan 0 menunjukkan yang terburuk.

`InferenceLatency`

Latensi inferensi adalah perkiraan jumlah waktu antara membuat permintaan untuk prediksi model untuk menerimanya dari titik akhir waktu nyata tempat model digunakan. Metrik ini diukur dalam hitungan detik dan hanya tersedia dalam mode ansambel.

`LogLoss`

Loss log, juga dikenal sebagai cross-entropy loss, adalah metrik yang digunakan untuk mengevaluasi kualitas output probabilitas, bukan output itu sendiri. Ini digunakan dalam klasifikasi biner dan multiclass dan dalam jaring saraf. Ini juga merupakan fungsi biaya untuk regresi logistik. Kehilangan log adalah metrik penting untuk menunjukkan kapan model membuat prediksi yang salah dengan probabilitas tinggi. Nilai berkisar dari 0 hingga tak terbatas. Nilai 0 mewakili model yang memprediksi data dengan sempurna.

`MAE`

Kesalahan absolut rata-rata (MAE) adalah ukuran seberapa berbeda nilai prediksi dan aktual, ketika dirata-ratakan pada semua nilai. MAE biasanya digunakan dalam analisis regresi untuk memahami kesalahan prediksi model. Jika ada regresi linier, MAE mewakili jarak rata-rata dari garis yang diprediksi ke nilai aktual. MAE didefinisikan sebagai jumlah kesalahan absolut dibagi dengan jumlah pengamatan. Nilai berkisar dari 0 hingga tak terhingga, dengan angka yang lebih kecil menunjukkan kecocokan model yang lebih baik dengan data.

`MSE`

Mean squared error (MSE) adalah rata-rata perbedaan kuadrat antara nilai prediksi dan aktual. Ini digunakan untuk regresi. Nilai MSE selalu positif. Semakin baik model dalam memprediksi nilai aktual, semakin kecil nilai MSE.

`Precision`

Presisi mengukur seberapa baik suatu algoritma memprediksi positif sejati (TP) dari semua hal positif yang diidentifikasi. Ini didefinisikan sebagai berikut: Presisi = TP/ (TP+FP), dengan nilai

mulai dari nol (0) hingga satu (1), dan digunakan dalam klasifikasi biner. Presisi adalah metrik penting ketika biaya positif palsu tinggi. Misalnya, biaya positif palsu sangat tinggi jika sistem keselamatan pesawat dianggap aman untuk terbang. Positif palsu (FP) mencerminkan prediksi positif yang sebenarnya negatif dalam data.

`PrecisionMacro`

Makro presisi menghitung presisi untuk masalah klasifikasi multiclass. Ini dilakukan dengan menghitung presisi untuk setiap kelas dan skor rata-rata untuk mendapatkan presisi untuk beberapa kelas. `PrecisionMacro`Skor berkisar dari nol (0) hingga satu (1). Skor yang lebih tinggi mencerminkan kemampuan model untuk memprediksi positif sejati (TP) dari semua positif yang diidentifikasi, dirata-ratakan di beberapa kelas.

`R2`

R^2, juga dikenal sebagai koefisien determinasi, digunakan dalam regresi untuk mengukur seberapa banyak model dapat menjelaskan varians variabel dependen. Nilai berkisar dari satu (1) ke negatif (-1). Angka yang lebih tinggi menunjukkan fraksi yang lebih tinggi dari variabilitas yang dijelaskan. `R2`nilai mendekati nol (0) menunjukkan bahwa sangat sedikit variabel dependen yang dapat dijelaskan oleh model. Nilai negatif menunjukkan kecocokan yang buruk dan bahwa model tersebut dikalahkan oleh fungsi konstan. Untuk regresi linier, ini adalah garis horizontal.

`Recall`

Ingat mengukur seberapa baik algoritme memprediksi dengan benar semua positif sejati (TP) dalam kumpulan data. Positif sejati adalah prediksi positif yang juga merupakan nilai positif aktual dalam data. Ingat didefinisikan sebagai berikut: Ingat = TP/ (TP+FN), dengan nilai mulai dari 0 hingga 1. Skor yang lebih tinggi mencerminkan kemampuan model yang lebih baik untuk memprediksi positif sejati (TP) dalam data. Ini digunakan dalam klasifikasi biner.

Ingat penting ketika menguji kanker karena digunakan untuk menemukan semua hal positif yang sebenarnya. Positif palsu (FP) mencerminkan prediksi positif yang sebenarnya negatif dalam data. Seringkali tidak cukup untuk mengukur hanya ingatan, karena memprediksi setiap output sebagai positif sejati menghasilkan skor ingatan yang sempurna.

`RecallMacro`

`RecallMacro`Menghitung penarikan kembali untuk masalah klasifikasi multiclass dengan menghitung recall untuk setiap kelas dan skor rata-rata untuk mendapatkan recall untuk beberapa kelas. `RecallMacro`skor berkisar dari 0 hingga 1. Skor yang lebih tinggi mencerminkan kemampuan model untuk memprediksi positif sejati (TP) dalam kumpulan data, sedangkan positif sejati mencerminkan prediksi positif yang juga merupakan nilai positif aktual dalam data.

Seringkali tidak cukup untuk mengukur hanya ingatan, karena memprediksi setiap output sebagai positif sejati akan menghasilkan skor ingatan yang sempurna.

`RMSE`

Kesalahan kuadrat rata-rata akar (RMSE) mengukur akar kuadrat dari perbedaan kuadrat antara nilai prediksi dan aktual, dan dirata-ratakan pada semua nilai. Ini digunakan dalam analisis regresi untuk memahami kesalahan prediksi model. Ini adalah metrik penting untuk menunjukkan adanya kesalahan model besar dan outlier. Nilai berkisar dari nol (0) hingga tak terhingga, dengan angka yang lebih kecil menunjukkan kecocokan model yang lebih baik dengan data. RMSE tergantung pada skala, dan tidak boleh digunakan untuk membandingkan kumpulan data dengan ukuran yang berbeda.

Metrik yang dihitung secara otomatis untuk kandidat model ditentukan oleh jenis masalah yang ditangani.

- Regresi:`InferenceLatency`,,,`MAE`, `MSE` `R2` `RMSE`
- Klasifikasi biner: `Accuracy`AUC,`BalancedAccuracy`,`F1`,`InferenceLatency`,,`LogLoss`,`Precision`, `Recall`
- Klasifikasi multiclass:`Accuracy`,,,`BalancedAccuracy`,,`F1macro`, `InferenceLatency` `LogLoss` `PrecisionMacro` `RecallMacro`

Metrik tertimbang autopilot

> **Note**
>
> Autopilot mendukung bobot sampel dalam mode ansambel hanya untuk semua metrik yang tersedia dengan pengecualian dan. `Balanced Accuracy` `InferenceLatency` `BalanceAccuracy`dilengkapi dengan skema pembobotannya sendiri untuk kumpulan data yang tidak seimbang yang tidak memerlukan bobot sampel. `InferenceLatency`tidak mendukung bobot sampel. Baik objektif `Balanced Accuracy` maupun `InferenceLatency` metrik mengabaikan bobot sampel yang ada saat melatih dan mengevaluasi model.

Pengguna dapat menambahkan kolom bobot sampel ke data mereka untuk memastikan bahwa setiap pengamatan yang digunakan untuk melatih model pembelajaran mesin diberi bobot yang

sesuai dengan persepsi pentingnya model. Ini sangat berguna dalam skenario di mana pengamatan dalam kumpulan data memiliki berbagai tingkat kepentingan, atau di mana kumpulan data berisi jumlah sampel yang tidak proporsional dari satu kelas dibandingkan dengan yang lain. Menetapkan bobot untuk setiap pengamatan berdasarkan pentingnya atau kepentingan yang lebih besar bagi kelas minoritas dapat membantu kinerja keseluruhan model, atau memastikan bahwa model tidak bias terhadap kelas mayoritas.

Untuk informasi tentang cara meneruskan bobot sampel saat membuat eksperimen di UI Studio Classic, lihat Langkah 7 di Membuat eksperimen Autopilot menggunakan Studio Classic.

Untuk informasi tentang cara meneruskan bobot sampel secara terprogram saat membuat eksperimen Autopilot menggunakan API, lihat Cara menambahkan bobot sampel ke pekerjaan AutoML di Membuat eksperimen Autopilot secara terprogram.

Validasi silang di Autopilot

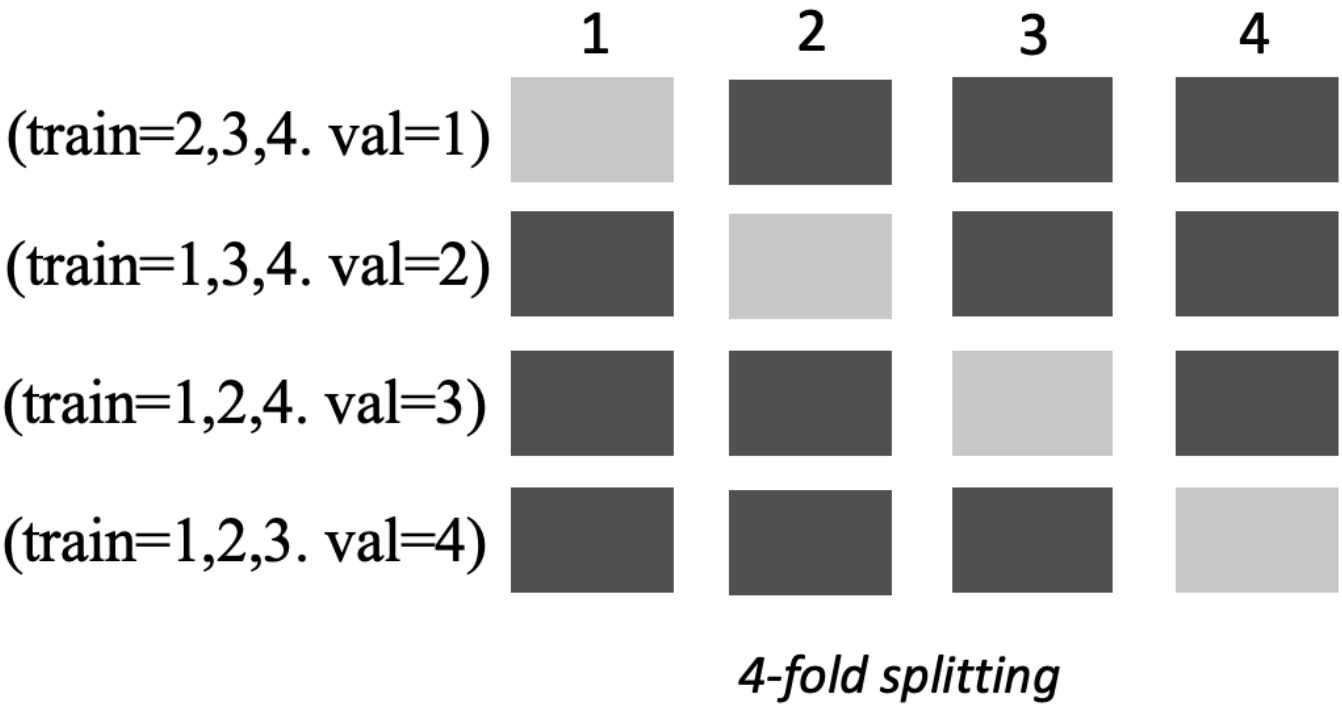
Validasi silang digunakan untuk mengurangi overfitting dan bias dalam pemilihan model. Hal ini juga digunakan untuk menilai seberapa baik model dapat memprediksi nilai-nilai dari dataset validasi yang tak terlihat, jika dataset validasi diambil dari populasi yang sama. Metode ini sangat penting saat melatih kumpulan data yang memiliki jumlah instance pelatihan terbatas.

Autopilot menggunakan validasi silang untuk membangun model dalam optimasi hyperparameter (HPO) dan mode pelatihan ensemble. Langkah pertama dalam proses validasi silang Autopilot adalah membagi data menjadi k-folds.

Pemisahan K-lipat

K-fold splitting adalah metode yang memisahkan dataset pelatihan input menjadi beberapa kumpulan data pelatihan dan validasi. Dataset dibagi menjadi sub-sampel k berukuran sama yang disebut lipatan. Model kemudian dilatih pada $k-1$ lipatan dan diuji terhadap lipatan k^{th} yang tersisa, yang merupakan kumpulan data validasi. Proses ini diulang k kali menggunakan kumpulan data yang berbeda untuk validasi.

Gambar berikut menggambarkan pemisahan k-fold dengan k = 4 lipatan. Setiap lipatan direpresentasikan sebagai baris. Kotak berwarna gelap mewakili bagian dari data yang digunakan dalam pelatihan. Kotak berwarna terang yang tersisa menunjukkan kumpulan data validasi.



Autopilot menggunakan validasi silang k-fold untuk mode optimasi hiperparameter (HPO) dan mode ansambel.

Anda dapat menerapkan model Autopilot yang dibuat menggunakan validasi silang seperti yang Anda lakukan dengan Autopilot atau model lainnya. SageMaker

Modus HPO

Validasi silang K-fold menggunakan metode pemisahan k-fold untuk validasi silang. Dalam mode HPO, Autopilot secara otomatis mengimplementasikan validasi silang k-fold untuk kumpulan data kecil dengan 50.000 instans pelatihan atau lebih sedikit. Melakukan validasi silang sangat penting saat melatih kumpulan data kecil karena melindungi terhadap overfitting dan bias seleksi.

Mode HPO menggunakan nilai k 5 pada masing-masing algoritma kandidat yang digunakan untuk memodelkan dataset. Beberapa model dilatih pada split yang berbeda, dan model disimpan secara terpisah. Ketika pelatihan selesai, metrik validasi untuk masing-masing model dirata-ratakan untuk menghasilkan metrik estimasi tunggal. Terakhir, Autopilot menggabungkan model dari uji coba dengan metrik validasi terbaik ke dalam model ansambel. Autopilot menggunakan model ansambel ini untuk membuat prediksi.

Metrik validasi untuk model yang dilatih oleh Autopilot disajikan sebagai metrik objektif di papan peringkat model. Autopilot menggunakan metrik validasi default untuk setiap jenis masalah yang

ditangani, kecuali jika Anda menentukan sebaliknya. Untuk daftar semua metrik yang digunakan Autopilot, lihat. Metrik Autopilot

Misalnya, dataset Perumahan Boston hanya berisi 861 sampel. Jika Anda membangun model untuk memprediksi harga jual rumah menggunakan dataset ini tanpa validasi silang, Anda berisiko melatih dataset yang tidak mewakili stok perumahan Boston. Jika Anda membagi data hanya sekali menjadi subset pelatihan dan validasi, lipatan pelatihan mungkin hanya berisi data terutama dari pinggiran kota. Akibatnya, Anda akan melatih data yang tidak mewakili seluruh kota. Dalam contoh ini, model Anda kemungkinan akan terlalu cocok dengan pilihan bias ini. Validasi silang K-fold dapat mengurangi risiko kesalahan semacam ini dengan memanfaatkan data yang tersedia secara penuh dan acak untuk pelatihan dan validasi.

Validasi silang dapat meningkatkan waktu pelatihan rata-rata 20%. Waktu pelatihan juga dapat meningkat secara signifikan untuk kumpulan data yang kompleks.

> **Note**
>
> Dalam mode HPO, Anda dapat melihat metrik pelatihan dan validasi dari setiap lipatan di Log Anda. `/aws/sagemaker/TrainingJobs` CloudWatch Untuk informasi selengkapnya tentang CloudWatch Log, lihatLog SageMaker Acara Amazon dengan Amazon CloudWatch.

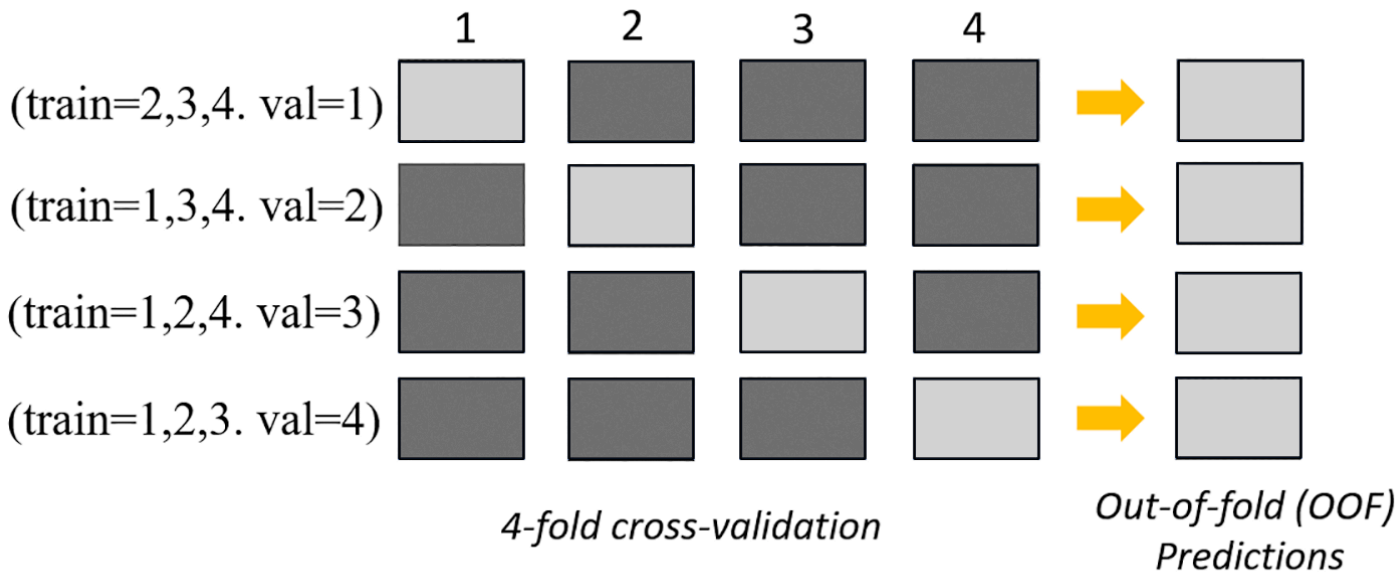
Mode ansambel

> **Note**
>
> Autopilot mendukung bobot sampel dalam mode ansambel. Untuk daftar metrik yang tersedia yang mendukung bobot sampel, lihat. Metrik Autopilot

Dalam mode ansambel, validasi silang dilakukan terlepas dari ukuran kumpulan data. Pelanggan dapat menyediakan kumpulan data validasi dan rasio pemisahan data khusus mereka sendiri, atau membiarkan Autopilot membagi kumpulan data secara otomatis menjadi rasio pemisahan 80-20%. Data pelatihan kemudian dibagi menjadi k -lipatan untuk validasi silang, di mana nilai k ditentukan oleh mesin. AutoGluon Sebuah ansambel terdiri dari beberapa model pembelajaran mesin, di mana setiap model dikenal sebagai model dasar. Model dasar tunggal dilatih pada (k-1) lipatan dan membuat out-of-fold prediksi pada lipatan yang tersisa. Proses ini diulang untuk semua k lipatan, dan prediksi out-of-fold (OOF) digabungkan untuk membentuk satu set prediksi. Semua model dasar dalam ansambel mengikuti proses yang sama untuk menghasilkan prediksi OOF.

Gambar berikut menggambarkan validasi k-fold dengan k = 4 lipatan. Setiap lipatan direpresentasikan sebagai baris. Kotak berwarna gelap mewakili bagian dari data yang digunakan dalam pelatihan. Kotak berwarna terang yang tersisa menunjukkan kumpulan data validasi.

Di bagian atas gambar, di setiap lipatan, model dasar pertama membuat prediksi pada kumpulan data validasi setelah pelatihan pada kumpulan data pelatihan. Pada setiap lipatan berikutnya, kumpulan data mengubah peran. Dataset yang sebelumnya digunakan untuk pelatihan sekarang digunakan untuk validasi, dan ini juga berlaku secara terbalik. Pada akhir k lipatan, semua prediksi digabungkan untuk membentuk satu set prediksi yang disebut prediksi (OOF). out-of-fold Proses ini diulang untuk setiap model n dasar.



Prediksi OOF untuk setiap model dasar kemudian digunakan sebagai fitur untuk melatih model susun. Model susun mempelajari bobot penting untuk setiap model dasar. Bobot ini digunakan untuk menggabungkan prediksi OOF untuk membentuk prediksi akhir. Kinerja pada dataset validasi menentukan basis atau model susun mana yang terbaik, dan model ini dikembalikan sebagai model akhir.

Dalam mode ansambel, Anda dapat memberikan kumpulan data validasi Anda sendiri atau membiarkan Autopilot membagi kumpulan data input secara otomatis menjadi kumpulan data 80% dan kumpulan data validasi 20%. Data pelatihan kemudian dibagi menjadi k -lipatan untuk validasi silang dan menghasilkan prediksi OOF dan model dasar untuk setiap lipatan.

Prediksi OOF ini digunakan sebagai fitur untuk melatih model susun, yang secara bersamaan mempelajari bobot untuk setiap model dasar. Bobot ini digunakan untuk menggabungkan prediksi OOF untuk membentuk prediksi akhir. Kumpulan data validasi untuk setiap lipatan digunakan untuk

penyetelan hiperparameter dari semua model dasar dan model susun. Kinerja pada kumpulan data validasi menentukan model basis atau susun mana yang merupakan model terbaik, dan model ini dikembalikan sebagai model akhir.

Penyebaran dan prediksi model Amazon SageMaker Autopilot

Panduan SageMaker Autopilot Amazon ini mencakup langkah-langkah untuk penerapan model, menyiapkan inferensi waktu nyata, dan menjalankan inferensi dengan pekerjaan batch.

Setelah Anda melatih model Autopilot Anda, Anda dapat menerapkannya untuk mendapatkan prediksi dengan salah satu dari dua cara:

1. Gunakan Inferensi waktu nyata untuk mengatur titik akhir dan mendapatkan prediksi secara interaktif.
2. Gunakan Inferensi Batch untuk membuat prediksi secara paralel pada batch pengamatan pada seluruh kumpulan data.

> **Note**
>
> Untuk menghindari biaya yang tidak perlu: Setelah titik akhir dan sumber daya yang dibuat dari penerapan model tidak lagi diperlukan, Anda dapat menghapusnya. Untuk informasi tentang penetapan harga instans menurut Wilayah, lihat SageMaker Harga Amazon.

Inferensi waktu nyata

Inferensi real-time sangat ideal untuk beban kerja inferensi di mana Anda memiliki persyaratan real-time, interaktif, latensi rendah. Bagian ini menunjukkan bagaimana Anda dapat menggunakan inferensi real-time untuk mendapatkan prediksi secara interaktif dari model Anda.

Untuk menerapkan model yang menghasilkan metrik validasi terbaik dalam eksperimen Autopilot, Anda memiliki beberapa opsi. Misalnya, saat menggunakan Autopilot di SageMaker Studio Classic, Anda dapat menerapkan model secara otomatis atau manual. Anda juga dapat menggunakan SageMaker API untuk menerapkan model Autopilot secara manual.

Tab berikut menunjukkan tiga opsi untuk menerapkan model Anda. Instruksi ini mengasumsikan bahwa Anda telah membuat model di Autopilot. Jika Anda tidak memiliki model, lihatMembuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API. Untuk melihat contoh untuk setiap opsi, buka setiap tab.

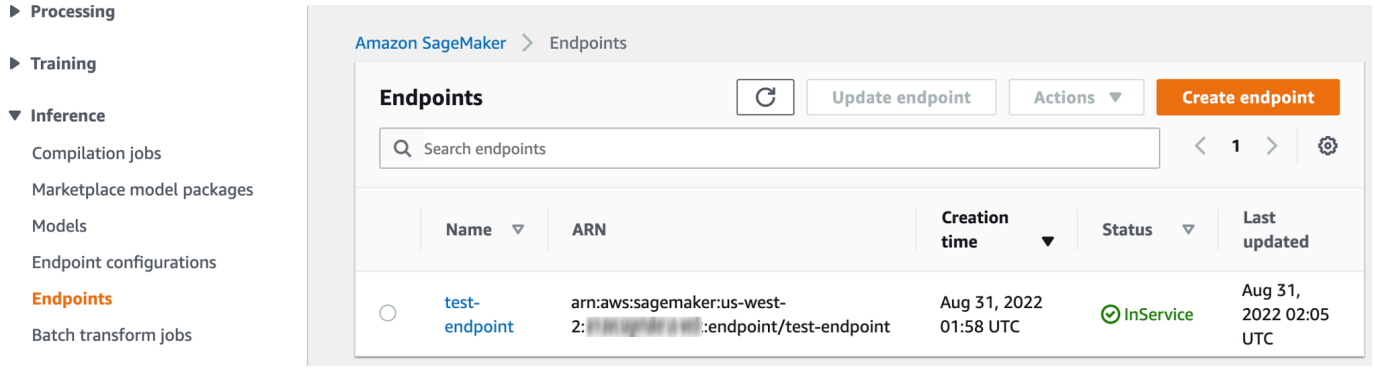
Terapkan menggunakan Antarmuka Pengguna Autopilot (UI)

UI Autopilot berisi menu tarik-turun yang bermanfaat, sakelar, tooltips, dan lainnya untuk membantu Anda menavigasi penerapan model. Anda dapat menggunakan salah satu dari prosedur berikut: Otomatis atau Manual.

- Automatic Deployment: Untuk secara otomatis menerapkan model terbaik dari eksperimen Autopilot ke titik akhir
  1. Buat eksperimen di SageMaker Studio Classic.
  2. Alihkan nilai Auto deploy ke Yes.

  > **Note**
  >
  > Penerapan otomatis akan gagal jika kuota sumber daya default atau kuota pelanggan Anda untuk instans titik akhir di Wilayah terlalu terbatas. Dalam mode optimasi hyperparameter (HPO), Anda harus memiliki setidaknya dua instance ml.m5.2xlarge. Dalam mode ansambel, Anda harus memiliki setidaknya satu instance ml.m5.12xlarge. Jika Anda mengalami kegagalan terkait kuota, Anda dapat meminta peningkatan batas layanan untuk instance SageMaker endpoint.

- Penerapan Manual: Untuk menerapkan model terbaik secara manual dari eksperimen Autopilot ke titik akhir
  1. Buat eksperimen di SageMaker Studio Classic.
  2. Alihkan nilai Auto deploy ke No.
  3. Pilih model yang ingin Anda gunakan di bawah Nama model.
  4. Pilih tombol Deployment dan pengaturan lanjutan berwarna oranye yang terletak di sebelah kanan papan peringkat. Ini membuka tab baru.
  5. Konfigurasikan nama titik akhir, jenis instance, dan informasi opsional lainnya.
  6. Pilih model Deploy oranye untuk menyebarkan ke titik akhir.
  7. Periksa kemajuan proses pembuatan titik akhir di https://console.aws.amazon.com/sagemaker/ dengan menavigasi ke bagian Endpoints. Bagian itu terletak di menu tarik-turun Inferensi di panel navigasi.
  8. Setelah status endpoint berubah dari Creating menjadi InService, seperti yang ditunjukkan di bawah ini, kembali ke Studio Classic dan panggil endpoint.



Terapkan menggunakan API SageMaker

Anda juga dapat memperoleh inferensi real-time dengan menerapkan model Anda menggunakan panggilan API. Bagian ini menunjukkan lima langkah proses ini menggunakan AWS Command Line Interface (AWS CLI) cuplikan kode.

Untuk contoh kode lengkap untuk kedua AWS CLI perintah dan AWS SDK untuk Python (boto3), buka tab langsung mengikuti langkah-langkah ini.

1. Dapatkan definisi kandidat

   Dapatkan definisi wadah kandidat dari InferenceContainers. Definisi kandidat ini digunakan untuk membuat SageMaker model.

   Contoh berikut menggunakan DescribeAutoMLJob API untuk mendapatkan definisi kandidat untuk kandidat model terbaik. Lihat AWS CLI perintah berikut sebagai contoh.

   ```
   aws sagemaker describe-auto-ml-job --auto-ml-job-name <job-name> --region <region>
   ```

2. Daftar kandidat

   Contoh berikut menggunakan ListCandidatesForAutoMLJob API untuk daftar semua kandidat. Lihat AWS CLI perintah berikut sebagai contoh.

   ```
   aws sagemaker list-candidates-for-auto-ml-job --auto-ml-job-name <job-name> --
   region <region>
   ```

3. Buat SageMaker model

   Gunakan definisi container dari langkah sebelumnya untuk membuat SageMaker model dengan menggunakan CreateModelAPI. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker create-model --model-name '<your-custom-model-name>' \
                    --containers ['<container-definition1>, <container-
definition2>, <container-definition3>]' \
                    --execution-role-arn '<execution-role-arn>' --region '<region>
```

4. Buat konfigurasi titik akhir

   Contoh berikut menggunakan CreateEndpointConfigAPI untuk membuat konfigurasi endpoint. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker create-endpoint-config --endpoint-config-name '<your-custom-endpoint-
config-name>' \
                    --production-variants '<list-of-production-variants>' \
                    --region '<region>'
```

5. Buat titik akhir

   AWS CLIContoh berikut menggunakan CreateEndpointAPI untuk membuat titik akhir.

```
aws sagemaker create-endpoint --endpoint-name '<your-custom-endpoint-name>' \
                    --endpoint-config-name '<endpoint-config-name-you-just-created>'
 \
                    --region '<region>'
```

   Periksa kemajuan penerapan titik akhir Anda dengan menggunakan API. DescribeEndpoint Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-endpoint –endpoint-name '<endpoint-name>' –region <region>
```

   Setelah `EndpointStatus` perubahan`InService`, titik akhir siap digunakan untuk inferensi waktu nyata.

6. Memanggil titik akhir

   Struktur perintah berikut memanggil titik akhir untuk inferensi real-time.

```
aws sagemaker invoke-endpoint --endpoint-name '<endpoint-name>' \
                  --region '<region>' --body '<your-data>' [--content-type]
 '<content-type>' <outfile>
```

Tab berikut berisi contoh kode lengkap untuk menerapkan model dengan AWS SDK untuk Python (boto3) atau file. AWS CLI

AWS SDK for Python (boto3)

1. Dapatkan definisi kandidat dengan menggunakan contoh kode berikut.

```
import sagemaker
import boto3

session = sagemaker.session.Session()

sagemaker_client = boto3.client('sagemaker', region_name='us-west-2')
job_name = 'test-auto-ml-job'

describe_response = sm_client.describe_auto_ml_job(AutoMLJobName=job_name)
# extract the best candidate definition from DescribeAutoMLJob response
best_candidate = describe_response['BestCandidate']
# extract the InferenceContainers definition from the caandidate definition
inference_containers = best_candidate['InferenceContainers']
```

2. Buat model dengan menggunakan contoh kode berikut.

```
# Create Model
model_name = 'test-model'
sagemaker_role = 'arn:aws:iam:444455556666:role/sagemaker-execution-role'
create_model_response = sagemaker_client.create_model(
   ModelName = model_name,
   ExecutionRoleArn = sagemaker_role,
   Containers = inference_containers
)
```

3. Buat konfigurasi endpoint dengan menggunakan contoh kode berikut.

```
endpoint_config_name = 'test-endpoint-config'

instance_type = 'ml.m5.2xlarge'
# for all supported instance types, see
# https://docs.aws.amazon.com/sagemaker/latest/APIReference/
API_ProductionVariant.html#sagemaker-Type-ProductionVariant-InstanceType    #
 Create endpoint config

endpoint_config_response = sagemaker_client.create_endpoint_config(
```

```
    EndpointConfigName=endpoint_config_name,
    ProductionVariants=[
        {
            "VariantName": "variant1",
            "ModelName": model_name,
            "InstanceType": instance_type,
            "InitialInstanceCount": 1
        }
    ]
)

print(f"Created EndpointConfig: {endpoint_config_response['EndpointConfigArn']}")
```

4. Buat titik akhir dan gunakan model dengan contoh kode berikut.

```
# create endpoint and deploy the model
endpoint_name = 'test-endpoint'
create_endpoint_response = sagemaker_client.create_endpoint(
                                            EndpointName=endpoint_name,

 EndpointConfigName=endpoint_config_name)
print(create_endpoint_response)
```

   Periksa status pembuatan titik akhir dengan menggunakan contoh kode berikut.

```
# describe endpoint creation status
status = sagemaker_client.describe_endpoint(EndpointName=endpoint_name)
["EndpointStatus"]
```

5. Memanggil endpoint untuk inferensi real-time dengan menggunakan struktur perintah berikut.

```
# once endpoint status is InService, you can invoke the endpoint for inferencing
if status == "InService":
  sm_runtime = boto3.Session().client('sagemaker-runtime')
  inference_result = sm_runtime.invoke_endpoint(EndpointName='test-endpoint',
 ContentType='text/csv', Body='1,2,3,4,class')
```

AWS Command Line Interface (AWS CLI)

1. Dapatkan definisi kandidat dengan menggunakan contoh kode berikut.

```
aws sagemaker describe-auto-ml-job --auto-ml-job-name 'test-automl-job' --
region us-west-2
```

2. Buat model dengan menggunakan contoh kode berikut.

```
aws sagemaker create-model --model-name 'test-sagemaker-model'
--containers '[{
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-sklearn-
automl:2.5-1-cpu-py3", DOC-EXAMPLE-BUCKET1
    "ModelDataUrl": "s3://DOC-EXAMPLE-BUCKET/output/model.tar.gz",
    "Environment": {
        "AUTOML_SPARSE_ENCODE_RECORDIO_PROTOBUF": "1",
        "AUTOML_TRANSFORM_MODE": "feature-transform",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
        "SAGEMAKER_PROGRAM": "sagemaker_serve",
        "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
    }
}, {
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
xgboost:1.3-1-cpu-py3",
    "ModelDataUrl": "s3://DOC-EXAMPLE-BUCKET/output/model.tar.gz",
    "Environment": {
        "MAX_CONTENT_LENGTH": "20971520",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
        "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_SUPPORTED":
 "predicted_label,probability,probabilities"
    }
}, {
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-sklearn-
automl:2.5-1-cpu-py3", aws-region
    "ModelDataUrl": "s3://DOC-EXAMPLE-BUCKET/output/model.tar.gz",
    "Environment": {
        "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
        "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_SUPPORTED":
 "predicted_label,probability,labels,probabilities",
        "SAGEMAKER_PROGRAM": "sagemaker_serve",
        "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
    }
```

```
}]' \
--execution-role-arn 'arn:aws:iam::1234567890:role/sagemaker-execution-role' \
--region 'us-west-2'
```

Untuk detail tambahan, lihat membuat model.

`create model`Perintah akan mengembalikan respons dalam format berikut.

```
{
    "ModelArn": "arn:aws:sagemaker:us-west-2:1234567890:model/test-sagemaker-
model"
}
```

3. Buat konfigurasi endpoint dengan menggunakan contoh kode berikut.

```
aws sagemaker create-endpoint-config --endpoint-config-name 'test-endpoint-config'
 \
--production-variants '[{"VariantName": "variant1",
                         "ModelName": "test-sagemaker-model",
                         "InitialInstanceCount": 1,
                         "InstanceType": "ml.m5.2xlarge"
                        }]' \
--region us-west-2
```

Perintah `create endpoint` konfigurasi akan mengembalikan respons dalam format berikut.

```
{
    "EndpointConfigArn": "arn:aws:sagemaker:us-west-2:1234567890:endpoint-config/
test-endpoint-config"
}
```

4. Buat endpoint dengan menggunakan contoh kode berikut.

```
aws sagemaker create-endpoint --endpoint-name 'test-endpoint' \
--endpoint-config-name 'test-endpoint-config' \
--region us-west-2
```

`create endpoint`Perintah akan mengembalikan respons dalam format berikut.

```
{
    "EndpointArn": "arn:aws:sagemaker:us-west-2:1234567890:endpoint/test-endpoint"
```

```
}
```

Periksa kemajuan penerapan titik akhir dengan menggunakan contoh kode CLI describe-endpoint berikut.

```
aws sagemaker describe-endpoint --endpoint-name 'test-endpoint' --region us-west-2
```

Pemeriksaan kemajuan sebelumnya akan mengembalikan respons dalam format berikut.

```
{
    "EndpointName": "test-endpoint",
    "EndpointArn": "arn:aws:sagemaker:us-west-2:1234567890:endpoint/test-
endpoint",
    "EndpointConfigName": "test-endpoint-config",
    "EndpointStatus": "Creating",
    "CreationTime": 1660251167.595,
    "LastModifiedTime": 1660251167.595
}
```

Setelah `EndpointStatus` perubahan`InService`, titik akhir siap digunakan dalam inferensi waktu nyata.

5. Memanggil endpoint untuk inferensi real-time dengan menggunakan struktur perintah berikut.

```
aws sagemaker-runtime invoke-endpoint --endpoint-name 'test-endpoint' \
--region 'us-west-2' \
--body '1,51,3.5,1.4,0.2' \
--content-type 'text/csv' \
'/tmp/inference_output'
```

Untuk opsi lainnya, lihat menjalankan titik akhir.

Terapkan model dari akun yang berbeda

Anda dapat menerapkan model Autopilot dari akun yang berbeda dari akun asli tempat model dibuat. Untuk menerapkan penerapan model lintas akun, bagian ini menunjukkan cara melakukan hal berikut:

1. Berikan izin ke akun penerapan

Untuk mengambil peran dalam akun pembangkit, Anda harus memberikan izin ke akun penyebaran. Ini memungkinkan akun penyebaran untuk menjelaskan pekerjaan Autopilot di akun pembangkit.

Contoh berikut menggunakan akun penghasil dengan `sagemaker-role` entitas tepercaya. Contoh menunjukkan cara memberikan akun penerapan dengan izin ID 111122223333 untuk mengambil peran akun pembangkit.

```
"Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com"
                ],
                "AWS": [ "111122223333"]
            },
            "Action": "sts:AssumeRole"
        }
```

Akun baru dengan ID 111122223333 sekarang dapat mengambil peran untuk akun pembangkit.

Selanjutnya, panggil `DescribeAutoMLJob` API dari akun penerapan untuk mendapatkan deskripsi pekerjaan yang dibuat oleh akun pembuat.

Contoh kode berikut menjelaskan model dari akun deploying.

```
import sagemaker
import boto3
session = sagemaker.session.Session()

sts_client = boto3.client('sts')
sts_client.assume_role

role = 'arn:aws:iam::111122223333:role/sagemaker-role'
role_session_name = "role-session-name"
_assumed_role = sts_client.assume_role(RoleArn=role,
 RoleSessionName=role_session_name)

credentials = _assumed_role["Credentials"]
access_key = credentials["AccessKeyId"]
```

```
secret_key = credentials["SecretAccessKey"]
session_token = credentials["SessionToken"]

session = boto3.session.Session()

sm_client = session.client('sagemaker', region_name='us-west-2',
                           aws_access_key_id=access_key,
                            aws_secret_access_key=secret_key,
                            aws_session_token=session_token)

# now you can call describe automl job created in account A

job_name = "test-job"
response= sm_client.describe_auto_ml_job(AutoMLJobName=job_name)
```

2. Berikan akses ke akun penyebaran ke artefak model di akun pembangkit.

   Akun penerapan hanya memerlukan akses ke artefak model di akun pembuat untuk menerapkannya. Ini terletak di S3 OutputPath yang ditentukan dalam panggilan `CreateAutoMLJob` API asli selama pembuatan model.

   Untuk memberikan akses akun penyebaran ke artefak model, pilih salah satu opsi berikut:

   a. Berikan akses ke `ModelDataUrl` dari akun pembangkit ke akun penyebaran.

      Selanjutnya, Anda perlu memberikan izin akun penerapan untuk mengambil peran tersebut. ikuti langkah-langkah inferensi waktu nyata untuk menerapkan.

   b. Salin artefak model dari S3 asli akun pembuat OutputPath ke akun pembuat.

      Untuk memberikan akses ke artefak model, Anda harus menentukan `best_candidate` model dan menetapkan ulang wadah model ke akun baru.

      Contoh berikut menunjukkan bagaimana mendefinisikan `best_candidate` model dan menetapkan kembali. `ModelDataUrl`

      ```
      best_candidate = automl.describe_auto_ml_job()['BestCandidate']

      # reassigning ModelDataUrl for best_candidate containers below
      new_model_locations = ['new-container-1-ModelDataUrl', 'new-container-2-
      ModelDataUrl', 'new-container-3-ModelDataUrl']
      new_model_locations_index = 0
      for container in best_candidate['InferenceContainers']:
      ```

```
    container['ModelDataUrl'] = new_model_locations[new_model_locations_index++]
```

Setelah penugasan kontainer ini, ikuti langkah-langkah Terapkan menggunakan API SageMaker untuk menerapkan.

Untuk membuat payload dalam inferensi real-time, lihat contoh notebook untuk menentukan payload pengujian. Untuk membuat payload dari file CSV dan menjalankan endpoint, lihat bagian Predict with your model di Create a machine learning model secara otomatis.

Inferensi Batch

Batch inferencing, juga dikenal sebagai inferensi offline, menghasilkan prediksi model pada batch pengamatan. Inferensi Batch adalah pilihan yang baik untuk kumpulan data besar atau jika Anda tidak memerlukan respons langsung terhadap permintaan prediksi model.

Sebaliknya, inferensi online (inferensi waktu nyata) menghasilkan prediksi secara real time.

Anda dapat membuat inferensi batch dari model Autopilot menggunakan SageMaker Python SDK, antarmuka pengguna Autopilot (UI), SDK AWS untuk Python (boto3), atau (). AWS Command Line Interface AWS CLI

Tab berikut menampilkan tiga opsi untuk menerapkan model Anda: Menggunakan API, UI Autopilot, atau menggunakan API untuk menerapkan dari akun yang berbeda. Instruksi ini mengasumsikan bahwa Anda telah membuat model di Autopilot. Jika Anda tidak memiliki model, lihatMembuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API. Untuk melihat contoh untuk setiap opsi, buka setiap tab.

Menerapkan model menggunakan Autopilot UI

UI Autopilot berisi menu tarik-turun yang bermanfaat, sakelar, tooltips, dan lainnya untuk membantu Anda menavigasi penerapan model.

Langkah-langkah berikut menunjukkan cara menerapkan model dari eksperimen Autopilot untuk prediksi batch.

1. Masuk di https://console.aws.amazon.com/sagemaker/ dan pilih Studio dari panel navigasi.
2. Di panel navigasi kiri, pilih Studio.
3. Di bawah Memulai, pilih Domain tempat Anda ingin meluncurkan aplikasi Studio. Jika profil pengguna Anda hanya milik satu Domain, Anda tidak melihat opsi untuk memilih Domain.

4. Pilih profil pengguna yang ingin Anda luncurkan aplikasi Studio Classic. Jika tidak ada profil pengguna di Domain, pilih Buat profil pengguna. Untuk informasi selengkapnya, lihat Menambahkan dan Menghapus Profil Pengguna.
5. Pilih Launch Studio. Jika profil pengguna milik ruang bersama, pilih Open Spaces.
6. Saat konsol SageMaker Studio Classic terbuka, pilih tombol Launch SageMaker Studio.
7. Pilih AutoML dari panel navigasi kiri.
8. Di bawah Nama, pilih eksperimen Autopilot yang sesuai dengan model yang ingin Anda gunakan. Ini membuka tab AUTOPILOT JOB baru.
9. Di bagian Nama model, pilih model yang ingin Anda terapkan.
10. Pilih model Deploy. Ini membuka tab baru.
11. Pilih Buat prediksi batch di bagian atas halaman.
12. Untuk konfigurasi pekerjaan transformasi Batch, masukkan tipe Instance, jumlah Instance, dan informasi opsional lainnya.
13. Di bagian Input data configuration, buka menu dropdown.
    a. Untuk tipe data S3, pilih ManifestFileatau S3Prefix.
    b. Untuk tipe Split, pilih Line, Recordio, TFRecord atau None.
    c. Untuk Kompresi, pilih Gzip atau Tidak Ada.
14. Untuk lokasi S3, masukkan lokasi bucket Amazon S3 dari data input dan informasi opsional lainnya.
15. Di bawah Konfigurasi data keluaran, masukkan bucket S3 untuk data keluaran, dan pilih cara merakit output pekerjaan Anda.
    a. Untuk konfigurasi Tambahan (opsional), Anda dapat memasukkan tipe MIME dan kunci Enkripsi S3.
16. Untuk penyaringan input/output dan gabungan data (opsional), Anda memasukkan ekspresi JSONPath untuk memfilter data input Anda, menggabungkan data sumber input dengan data keluaran Anda, dan memasukkan ekspresi JSONPath untuk memfilter data keluaran Anda.
    a. Untuk contoh untuk setiap jenis filter, lihat DataProcessing API.
17. Untuk melakukan prediksi batch pada kumpulan data input Anda, pilih Buat pekerjaan transformasi batch. Tab Batch Transform Jobs baru muncul.
18. Di tab Batch Transform Jobs: Temukan nama pekerjaan Anda di bagian Status. Kemudian periksa kemajuan pekerjaan.

Terapkan menggunakan API SageMaker

Untuk menggunakan SageMaker API untuk inferensi batch, ada tiga langkah:

1. Dapatkan definisi kandidat

   Definisi kandidat dari InferenceContainersdigunakan untuk membuat SageMaker model.

   Contoh berikut menunjukkan cara menggunakan DescribeAutoMLJob API untuk mendapatkan definisi kandidat untuk kandidat model terbaik. Lihat AWS CLI perintah berikut sebagai contoh.

   ```
   aws sagemaker describe-auto-ml-job --auto-ml-job-name <job-name> --region <region>
   ```

   Gunakan ListCandidatesForAutoMLJob API untuk mencantumkan semua kandidat. Lihat AWS CLI perintah berikut sebagai contoh.

   ```
   aws sagemaker list-candidates-for-auto-ml-job --auto-ml-job-name <job-name> --
   region <region>
   ```

2. Buat SageMaker model

   Untuk membuat SageMaker model menggunakan CreateModelAPI, gunakan definisi container dari langkah sebelumnya. Lihat AWS CLI perintah berikut sebagai contoh.

   ```
   aws sagemaker create-model --model-name '<your-custom-model-name>' \
                       --containers ['<container-definition1>, <container-
   definition2>, <container-definition3>]' \
                       --execution-role-arn '<execution-role-arn>' --region '<region>
   ```

3. Buat pekerjaan SageMaker transformasi

   Contoh berikut membuat pekerjaan SageMaker transformasi dengan CreateTransformJobAPI. Lihat AWS CLI perintah berikut sebagai contoh.

   ```
   aws sagemaker create-transform-job --transform-job-name '<your-custom-transform-job-
   name>' --model-name '<your-custom-model-name-from-last-step>'\
   --transform-input '{
           "DataSource": {
               "S3DataSource": {
                   "S3DataType": "S3Prefix",
                   "S3Uri": "<your-input-data>"
               }
   ```

```
        },
        "ContentType": "text/csv",
        "SplitType": "Line"
    }'\
--transform-output '{
        "S3OutputPath": "<your-output-path>",
        "AssembleWith": "Line"
    }'\
--transform-resources '{
        "InstanceType": "<instance-type>",
        "InstanceCount": 1
    }' --region '<region>'
```

Periksa kemajuan pekerjaan transformasi Anda menggunakan DescribeTransformJobAPI. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-transform-job --transform-job-name '<your-custom-transform-job-name>' --region <region>
```

Setelah pekerjaan selesai, hasil yang diprediksi akan tersedia di`<your-output-path>`.

Nama file output memiliki format berikut:`<input_data_file_name>.out`. Sebagai contoh, jika file input Anda`text_x.csv`, nama output akan menjadi`text_x.csv.out`.

Tab berikut menunjukkan contoh kode untuk SageMaker Python SDK, AWS SDK untuk Python (boto3), dan file. AWS CLI

SageMaker Python SDK

Contoh berikut menggunakan SageMaker Python SDK untuk membuat prediksi dalam batch.

```
from sagemaker import AutoML

sagemaker_session= sagemaker.session.Session()

job_name = 'test-auto-ml-job' # your autopilot job name
automl = AutoML.attach(auto_ml_job_name=job_name)
output_path = 's3://test-auto-ml-job/output'
input_data = 's3://test-auto-ml-job/test_X.csv'

# call DescribeAutoMLJob API to get the best candidate definition
```

```
best_candidate = automl.describe_auto_ml_job()['BestCandidate']
best_candidate_name = best_candidate['CandidateName']

# create model
model = automl.create_model(name=best_candidate_name,
               candidate=best_candidate)

# create transformer
transformer = model.transformer(instance_count=1,
    instance_type='ml.m5.2xlarge',
    assemble_with='Line',
    output_path=output_path)

# do batch transform
transformer.transform(data=input_data,
                      split_type='Line',
                       content_type='text/csv',
                       wait=True)
```

AWS SDK for Python (boto3)

Contoh berikut menggunakan AWSSDK untuk Python (boto3) untuk membuat prediksi dalam batch.

```
import sagemaker
import boto3

session = sagemaker.session.Session()

sm_client = boto3.client('sagemaker', region_name='us-west-2')
role = 'arn:aws:iam::1234567890:role/sagemaker-execution-role'
output_path = 's3://test-auto-ml-job/output'
input_data = 's3://test-auto-ml-job/test_X.csv'

best_candidate = sm_client.describe_auto_ml_job(AutoMLJobName=job_name)
['BestCandidate']
best_candidate_containers = best_candidate['InferenceContainers']
best_candidate_name = best_candidate['CandidateName']

# create model
reponse = sm_client.create_model(
    ModelName = best_candidate_name,
    ExecutionRoleArn = role,
```

```
    Containers = best_candidate_containers
)

# Lauch Transform Job
response = sm_client.create_transform_job(
    TransformJobName=f'{best_candidate_name}-transform-job',
    ModelName=model_name,
    TransformInput={
        'DataSource': {
            'S3DataSource': {
                'S3DataType': 'S3Prefix',
                'S3Uri': input_data
            }
        },
        'ContentType': "text/csv",
        'SplitType': 'Line'
    },
    TransformOutput={
        'S3OutputPath': output_path,
        'AssembleWith': 'Line',
    },
    TransformResources={
        'InstanceType': 'ml.m5.2xlarge',
        'InstanceCount': 1,
    },
)
```

Pekerjaan inferensi batch mengembalikan respons dalam format berikut.

```
{'TransformJobArn': 'arn:aws:sagemaker:us-west-2:1234567890:transform-job/test-
transform-job',
 'ResponseMetadata': {'RequestId': '659f97fc-28c4-440b-b957-a49733f7c2f2',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '659f97fc-28c4-440b-b957-a49733f7c2f2',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '96',
   'date': 'Thu, 11 Aug 2022 22:23:49 GMT'},
  'RetryAttempts': 0}}
```

AWS Command Line Interface (AWS CLI)

1. Dapatkan definisi kandidat dengan menggunakan contoh kode berikut.

```
aws sagemaker describe-auto-ml-job --auto-ml-job-name 'test-automl-job' --
region us-west-2
```

2. Buat model dengan menggunakan contoh kode berikut.

```
aws sagemaker create-model --model-name 'test-sagemaker-model'
--containers '[{
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-sklearn-
automl:2.5-1-cpu-py3",
    "ModelDataUrl": "s3://test-bucket/out/test-job1/data-processor-models/test-
job1-dpp0-1-e569ff7ad77f4e55a7e549a/output/model.tar.gz",
    "Environment": {
        "AUTOML_SPARSE_ENCODE_RECORDIO_PROTOBUF": "1",
        "AUTOML_TRANSFORM_MODE": "feature-transform",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
        "SAGEMAKER_PROGRAM": "sagemaker_serve",
        "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
    }
}, {
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
xgboost:1.3-1-cpu-py3",
    "ModelDataUrl": "s3://test-bucket/out/test-job1/tuning/flicdf10v2-dpp0-xgb/
test-job1E9-244-7490a1c0/output/model.tar.gz",
    "Environment": {
        "MAX_CONTENT_LENGTH": "20971520",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
        "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_SUPPORTED":
 "predicted_label,probability,probabilities"
    }
}, {
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-sklearn-
automl:2.5-1-cpu-py3",
    "ModelDataUrl": "s3://test-bucket/out/test-job1/data-processor-models/test-
job1-dpp0-1-e569ff7ad77f4e55a7e549a/output/model.tar.gz",
    "Environment": {
        "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
        "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_SUPPORTED":
 "predicted_label,probability,labels,probabilities",
```

```
        "SAGEMAKER_PROGRAM": "sagemaker_serve",
        "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
    }
}]' \
--execution-role-arn 'arn:aws:iam::1234567890:role/sagemaker-execution-role' \
--region 'us-west-2'
```

3. Buat pekerjaan transformasi dengan menggunakan contoh kode berikut.

```
aws sagemaker create-transform-job --transform-job-name 'test-tranform-job'\
 --model-name 'test-sagemaker-model'\
--transform-input '{
        "DataSource": {
            "S3DataSource": {
                "S3DataType": "S3Prefix",
                "S3Uri": "s3://test-bucket/data.csv"
            }
        },
        "ContentType": "text/csv",
        "SplitType": "Line"
    }'\
--transform-output '{
        "S3OutputPath": "s3://test-bucket/output/",
        "AssembleWith": "Line"
    }'\
--transform-resources '{
        "InstanceType": "ml.m5.2xlarge",
        "InstanceCount": 1
    }'\
--region 'us-west-2'
```

4. Periksa kemajuan pekerjaan transformasi dengan menggunakan contoh kode berikut.

```
aws sagemaker describe-transform-job --transform-job-name  'test-tranform-job' --
region us-west-2
```

Berikut ini adalah respons dari pekerjaan transformasi.

```
{
    "TransformJobName": "test-tranform-job",
    "TransformJobArn": "arn:aws:sagemaker:us-west-2:1234567890:transform-job/test-
tranform-job",
    "TransformJobStatus": "InProgress",
```

```
    "ModelName": "test-model",
    "TransformInput": {
        "DataSource": {
            "S3DataSource": {
                "S3DataType": "S3Prefix",
                "S3Uri": "s3://test-bucket/data.csv"
            }
        },
        "ContentType": "text/csv",
        "CompressionType": "None",
        "SplitType": "Line"
    },
    "TransformOutput": {
        "S3OutputPath": "s3://test-bucket/output/",
        "AssembleWith": "Line",
        "KmsKeyId": ""
    },
    "TransformResources": {
        "InstanceType": "ml.m5.2xlarge",
        "InstanceCount": 1
    },
    "CreationTime": 1662495635.679,
    "TransformStartTime": 1662495847.496,
    "DataProcessing": {
        "InputFilter": "$",
        "OutputFilter": "$",
        "JoinSource": "None"
    }
}
```

Setelah `TransformJobStatus` perubahan`Completed`, Anda dapat memeriksa hasil inferensi di. `S3OutputPath`

Terapkan model dari akun yang berbeda

Untuk membuat pekerjaan inferensi batch di akun yang berbeda dari yang dibuat model, ikuti instruksi di[Terapkan model dari akun yang berbeda](). Kemudian Anda dapat membuat model dan mengubah pekerjaan dengan mengikuti[Terapkan menggunakan API SageMaker]() .

Model yang dihasilkan oleh Amazon SageMaker Autopilot

Prosedur ini menjelaskan cara membagikan model yang Anda buat di Amazon SageMaker Autopilot dengan pengguna lain di Canvas. SageMaker Ini juga menunjukkan cara melihat detail tentang pekerjaan yang telah Anda jalankan.

Prasyarat

Sebelum Anda memulai prosedur ini, Anda harus telah membuat dan menjalankan eksperimen Autopilot. Untuk petunjuk, lihat Membuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API.

Bagikan model Autopilot Anda

Anda dapat membagikan model Autopilot Anda dengan pengguna lain di Canvas. SageMaker Pengguna lain kemudian dapat mengimpor model Anda dan menggunakannya untuk menghasilkan prediksi.

Untuk berbagi model di antarmuka pengguna Autopilot menggunakan tombol, lihat bagian berikut Lihat detail model. Tombol Bagikan Model dibahas pada Langkah 6.

Untuk informasi selengkapnya tentang cara berbagi model, lihat Membawa Model Anda Sendiri Ke Kanvas.

Lihat detail model

Autopilot menghasilkan detail tentang model kandidat yang dapat Anda peroleh. Rincian ini meliputi yang berikut:

- Plot nilai SHAP agregat yang menunjukkan pentingnya setiap fitur. Ini membantu menjelaskan prediksi model Anda.
- Ringkasan statistik untuk berbagai metrik pelatihan dan validasi, termasuk metrik objektif.
- Daftar hyperparameter yang digunakan untuk melatih dan menyetel model.

Untuk melihat detail model setelah menjalankan pekerjaan Autopilot, ikuti langkah-langkah berikut:

1. Pilih ikon Beranda

   dari panel navigasi kiri untuk melihat menu navigasi Amazon SageMaker Studio Classic tingkat atas.

2. Pilih kartu AutoML dari area kerja utama. Ini membuka tab Autopilot baru.
3. Di bagian Nama, pilih pekerjaan Autopilot yang memiliki detail yang ingin Anda periksa. Ini membuka tab pekerjaan Autopilot baru.
4. Panel pekerjaan Autopilot mencantumkan nilai metrik termasuk metrik Objektif untuk setiap model di bawah nama Model. Model Terbaik tercantum di bagian atas daftar di bawah nama Model dan juga disorot di tab Model.

   - Untuk meninjau detail model, pilih model yang Anda minati dan pilih Lihat detail model. Ini membuka tab Detail Model baru.
5. Tab Detail Model dibagi menjadi empat subbagian.

   1. Bagian atas tab Explainability berisi plot nilai SHAP agregat yang menunjukkan pentingnya setiap fitur. Berikut itu adalah metrik dan nilai hyperparameter untuk model ini.
   2. Tab Performance berisi statistik metrik matriks kebingungan.
   3. Tab Artefak berisi informasi tentang input model, output, dan hasil antara.
   4. Tab Jaringan merangkum pilihan isolasi dan enkripsi jaringan Anda.

   > **Note**
   >
   > Kepentingan fitur dan informasi di tab Performance hanya dihasilkan untuk model Terbaik.

   Untuk informasi lebih lanjut tentang bagaimana nilai SHAP membantu menjelaskan prediksi berdasarkan kepentingan fitur, lihat whitepaper Memahami penjelasan model. Informasi tambahan juga tersedia dalam Penjelasan Model topik di Panduan SageMaker Pengembang.
6. Untuk membagikan model Autopilot Anda dengan pengguna SageMaker Canvas lain, pilih Bagikan Model. Tombol itu terletak di kanan atas tab Detail Model.

   - Di bagian Tambahkan pengguna Canvas, gunakan panah bawah untuk memilih pengguna SageMaker Canvas.

Lihat Laporan Kinerja Model Autopilot

Laporan kualitas SageMaker model Amazon (juga disebut sebagai laporan kinerja) memberikan wawasan dan informasi kualitas untuk kandidat model terbaik yang dihasilkan oleh pekerjaan

AutoML. Ini termasuk informasi tentang detail pekerjaan, jenis masalah model, fungsi tujuan, dan informasi lain yang terkait dengan jenis masalah. Panduan ini menunjukkan cara melihat metrik kinerja Amazon SageMaker Autopilot secara grafis, atau melihat metrik sebagai data mentah dalam file JSON.

Misalnya, dalam masalah klasifikasi, laporan kualitas model meliputi yang berikut:

- Matriks kebingungan
- Area di bawah kurva karakteristik operasi penerima (AUC)
- Informasi untuk memahami positif palsu dan negatif palsu
- Pengorbanan antara positif benar dan positif palsu
- Pengorbanan antara presisi dan penarikan

Autopilot juga menyediakan metrik kinerja untuk semua model kandidat Anda. Metrik ini dihitung menggunakan semua data pelatihan dan digunakan untuk memperkirakan kinerja model. Area kerja utama mencakup metrik ini secara default. Jenis metrik ditentukan oleh jenis masalah yang ditangani.

Metrik kinerja berikut dikaitkan dengan jenis masalah yang sesuai:

- Regresi:`MAE`,,, `MSE` `R2` `RMSE`
- Klasifikasi biner: `Accuracy` `AUC`2`BalancedAccuracy`,`F1`,,`LogLoss`,`Precision`, `Recall`
- Klasifikasi multiclass:`Accuracy`,,,,`BalancedAccuracy`, `F1macro` `LogLoss` `PrecisionMacro` `RecallMacro`

Anda dapat mengurutkan kandidat model Anda dengan metrik yang relevan untuk membantu Anda memilih dan menerapkan model yang memenuhi kebutuhan bisnis Anda. Untuk definisi metrik ini, lihat topik metrik kandidat Autopilot.

Untuk melihat laporan kinerja dari pekerjaan Autopilot, ikuti langkah-langkah berikut:

1. Pilih ikon Beranda

   dari panel navigasi kiri untuk melihat menu navigasi Amazon SageMaker Studio Classic tingkat atas.
2. Pilih kartu AutoML dari area kerja utama. Ini membuka tab Autopilot baru.

3. Di bagian Nama, pilih pekerjaan Autopilot yang memiliki detail yang ingin Anda periksa. Ini membuka tab pekerjaan Autopilot baru.
4. Panel pekerjaan Autopilot mencantumkan nilai metrik termasuk metrik Objektif untuk setiap model di bawah nama Model. Model Terbaik tercantum di bagian atas daftar di bawah nama Model dan disorot di tab Model.

   - Untuk meninjau detail model, pilih model yang Anda minati dan pilih Lihat dalam detail model. Ini membuka tab Detail Model baru.
5. Pilih tab Performance antara tab Explainability dan Artefacts.

   a. Di bagian kanan atas tab, pilih panah bawah pada tombol Unduh Laporan Kinerja.

   b. Panah bawah menyediakan dua opsi untuk melihat metrik kinerja Autopilot:

      i. Anda dapat mengunduh PDF laporan kinerja untuk melihat metrik secara grafis.

      ii. Anda dapat melihat metrik sebagai data mentah dan mengunduhnya sebagai file JSON.

Untuk petunjuk tentang cara membuat dan menjalankan pekerjaan AutoML di SageMaker Studio Classic, lihat. Membuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API

Laporan kinerja berisi dua bagian. Yang pertama berisi detail tentang pekerjaan Autopilot yang menghasilkan model. Bagian kedua berisi laporan kualitas model.

Autopilot Job Detail

Bagian pertama dari laporan ini memberikan beberapa informasi umum tentang pekerjaan Autopilot yang menghasilkan model. Rincian pekerjaan ini mencakup informasi berikut:

- Nama kandidat autopilot
- Nama pekerjaan Autopilot
- Jenis masalah
- Metrik obyektif
- Arah optimasi

Laporan kualitas model

Informasi kualitas model dihasilkan oleh wawasan model Autopilot. Konten laporan yang dihasilkan bergantung pada jenis masalah yang ditangani: regresi, klasifikasi biner, atau klasifikasi multikelas.

Laporan tersebut menentukan jumlah baris yang termasuk dalam dataset evaluasi dan waktu evaluasi terjadi.

Tabel metrik

Bagian pertama dari laporan kualitas model berisi tabel metrik. Ini sesuai untuk jenis masalah yang ditangani model.

Gambar berikut adalah contoh tabel metrik yang dihasilkan Autopilot untuk masalah regresi. Ini menunjukkan nama metrik, nilai, dan standar deviasi.

Metrics table

Metric Name	Value	Standard Deviation
mae	5.347324	0.118636
mse	87.874017	4.346468
rmse	9.374114	0.232349
r2	0.924700	0.003710

Gambar berikut adalah contoh tabel metrik yang dihasilkan oleh Autopilot untuk masalah klasifikasi multiclass. Ini menunjukkan nama metrik, nilai, dan standar deviasi.

Metrics table

Metric Name	Value	Standard Deviation
weighted_recall	0.597104	0.005410
weighted_precision	0.591693	0.005729
accuracy	0.597104	0.005410
weighted_f0_5	0.592155	0.005659
weighted_f1	0.593423	0.005554
weighted_f2	0.595392	0.005456
accuracy_best_constant_classifier	0.200699	0.004422
weighted_recall_best_constant_classifier	0.200699	0.004422
weighted_precision_best_constant_classifier	0.040280	0.001753
weighted_f0_5_best_constant_classifier	0.047944	0.002039
weighted_f1_best_constant_classifier	0.067094	0.002684
weighted_f2_best_constant_classifier	0.111716	0.003808

Informasi kinerja model grafis

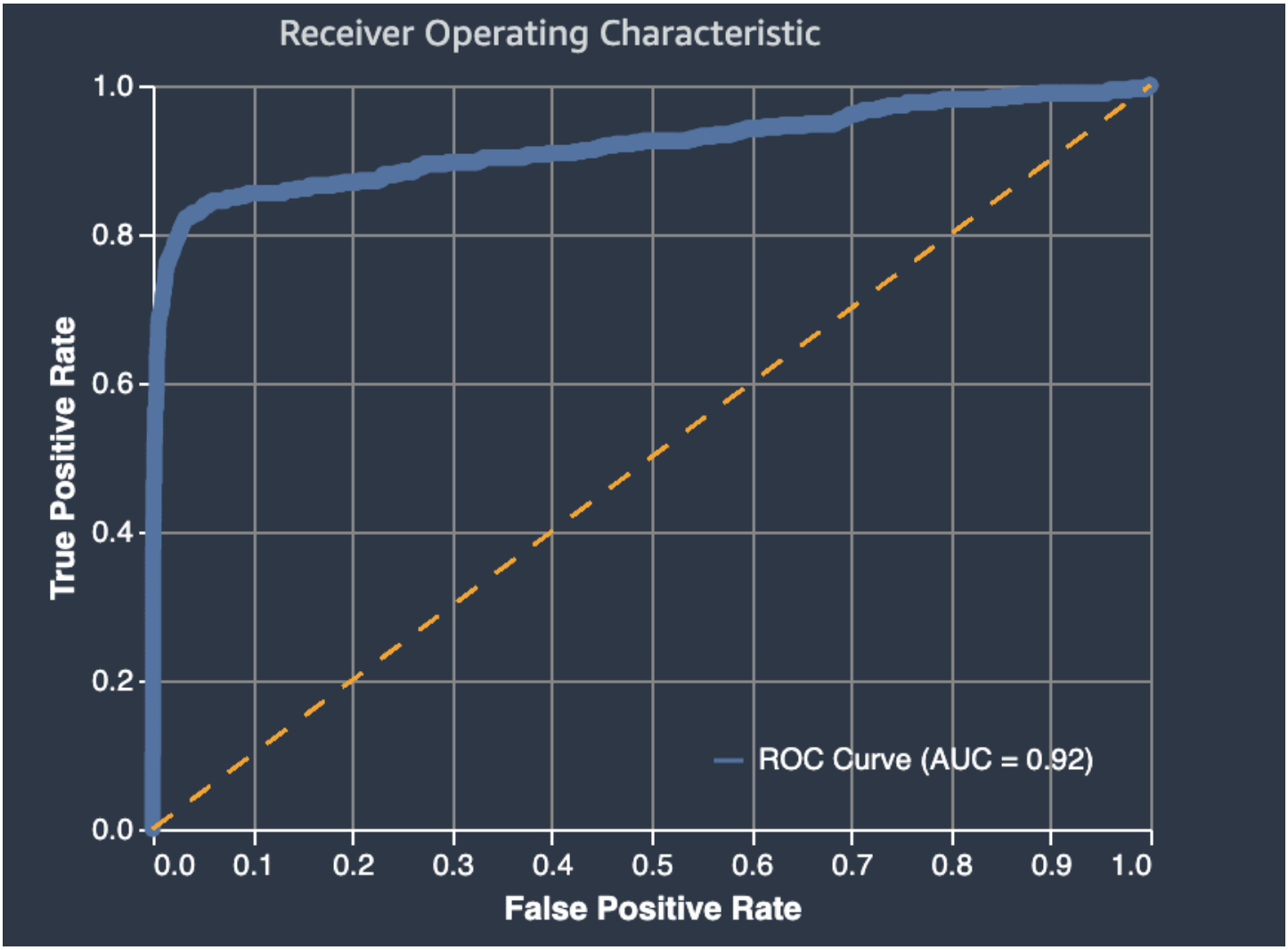
Bagian kedua dari laporan kualitas model berisi informasi grafis untuk membantu Anda mengevaluasi kinerja model. Isi bagian ini tergantung pada jenis masalah yang digunakan dalam pemodelan.

Area di bawah kurva karakteristik operasi penerima

Area di bawah kurva karakteristik operasi penerima mewakili trade-off antara tingkat positif benar dan positif palsu. Ini adalah metrik akurasi standar industri yang digunakan untuk model klasifikasi biner. AUC (area di bawah kurva) mengukur kemampuan model untuk memprediksi skor yang lebih tinggi untuk contoh positif, dibandingkan dengan contoh negatif. Metrik AUC memberikan ukuran agregat dari kinerja model di semua ambang batas klasifikasi yang mungkin.

Metrik AUC mengembalikan nilai desimal dari 0 menjadi 1. Nilai AUC mendekati 1 menunjukkan bahwa model pembelajaran mesin sangat akurat. Nilai mendekati 0,5 menunjukkan bahwa model berkinerja tidak lebih baik daripada menebak secara acak. Nilai AUC mendekati 0 menunjukkan bahwa model telah mempelajari pola yang benar, tetapi membuat prediksi yang seakurat mungkin. Nilai mendekati nol dapat menunjukkan masalah dengan data. Untuk informasi selengkapnya tentang metrik AUC, lihat artikel karakteristik pengoperasian Penerima di Wikipedia.

Berikut ini adalah contoh area di bawah grafik kurva karakteristik operasi penerima untuk mengevaluasi prediksi yang dibuat oleh model klasifikasi biner. Garis tipis putus-putus mewakili area di bawah kurva karakteristik operasi penerima yang akan dinilai oleh model yang mengklasifikasikan no-better-than-random tebakan, dengan skor AUC 0,5. Kurva model klasifikasi yang lebih akurat terletak di atas garis dasar acak ini, di mana tingkat positif sejati melebihi tingkat positif palsu. Area di bawah kurva karakteristik operasi penerima yang mewakili kinerja model klasifikasi biner adalah garis padat yang lebih tebal.



Ringkasan komponen grafik tingkat positif palsu (FPR) dan tingkat positif sejati (TPR) didefinisikan sebagai berikut.

- Prediksi yang benar
  - True positive (TP): Nilai yang diprediksi adalah 1, dan nilai sebenarnya adalah 1.
  - Benar negatif (TN): Nilai yang diprediksi adalah 0, dan nilai sebenarnya adalah 0.
- Prediksi yang salah
  - Positif palsu (FP): Nilai yang diprediksi adalah 1, tetapi nilai sebenarnya adalah 0.
  - False negative (FN): Nilai yang diprediksi adalah 0, tetapi nilai sebenarnya adalah 1.

Tingkat positif palsu (FPR) mengukur fraksi negatif sejati (TN) yang diprediksi secara salah sebagai positif (FP), atas jumlah FP dan TN. Kisarannya adalah 0 hingga 1. Nilai yang lebih kecil menunjukkan akurasi prediksi yang lebih baik.

- FPR = FP/ (FP+TN)

Tingkat positif sejati (TPR) mengukur fraksi positif sejati yang diprediksi dengan benar sebagai positif (TP) atas jumlah TP dan negatif palsu (FN). Kisarannya adalah 0 hingga 1. Nilai yang lebih besar menunjukkan akurasi prediksi yang lebih baik.

- TPR = TP/ (TP+FN)

Matriks kebingungan

Matriks kebingungan menyediakan cara untuk memvisualisasikan keakuratan prediksi yang dibuat oleh model untuk klasifikasi biner dan multikelas untuk masalah yang berbeda. Matriks kebingungan dalam laporan kualitas model berisi yang berikut ini.

- Jumlah dan persentase prediksi yang benar dan salah untuk label yang sebenarnya
- Jumlah dan persentase prediksi akurat pada diagonal dari kiri atas ke pojok kanan bawah
- Jumlah dan persentase prediksi yang tidak akurat pada diagonal dari kanan atas ke sudut kiri bawah

Prediksi yang salah pada matriks kebingungan adalah nilai kebingungan.

Diagram berikut adalah contoh matriks kebingungan untuk masalah klasifikasi biner. Itu berisi informasi berikut:

- Sumbu vertikal dibagi menjadi dua baris yang berisi label aktual benar dan salah.
- Sumbu horizontal dibagi menjadi dua kolom yang berisi label benar dan salah yang diprediksi oleh model.
- Bilah warna memberikan nada yang lebih gelap ke sejumlah besar sampel untuk secara visual menunjukkan jumlah nilai yang diklasifikasikan dalam setiap kategori.

Dalam contoh ini, model memprediksi 2817 nilai palsu aktual dengan benar, dan 353 nilai sebenarnya sebenarnya dengan benar. Model salah memprediksi 130 nilai sebenarnya sebenarnya menjadi salah dan 33 nilai palsu aktual menjadi benar. Perbedaan nada menunjukkan bahwa dataset

tidak seimbang. Ketidakseimbangan ini karena ada lebih banyak label palsu yang sebenarnya daripada label sebenarnya.

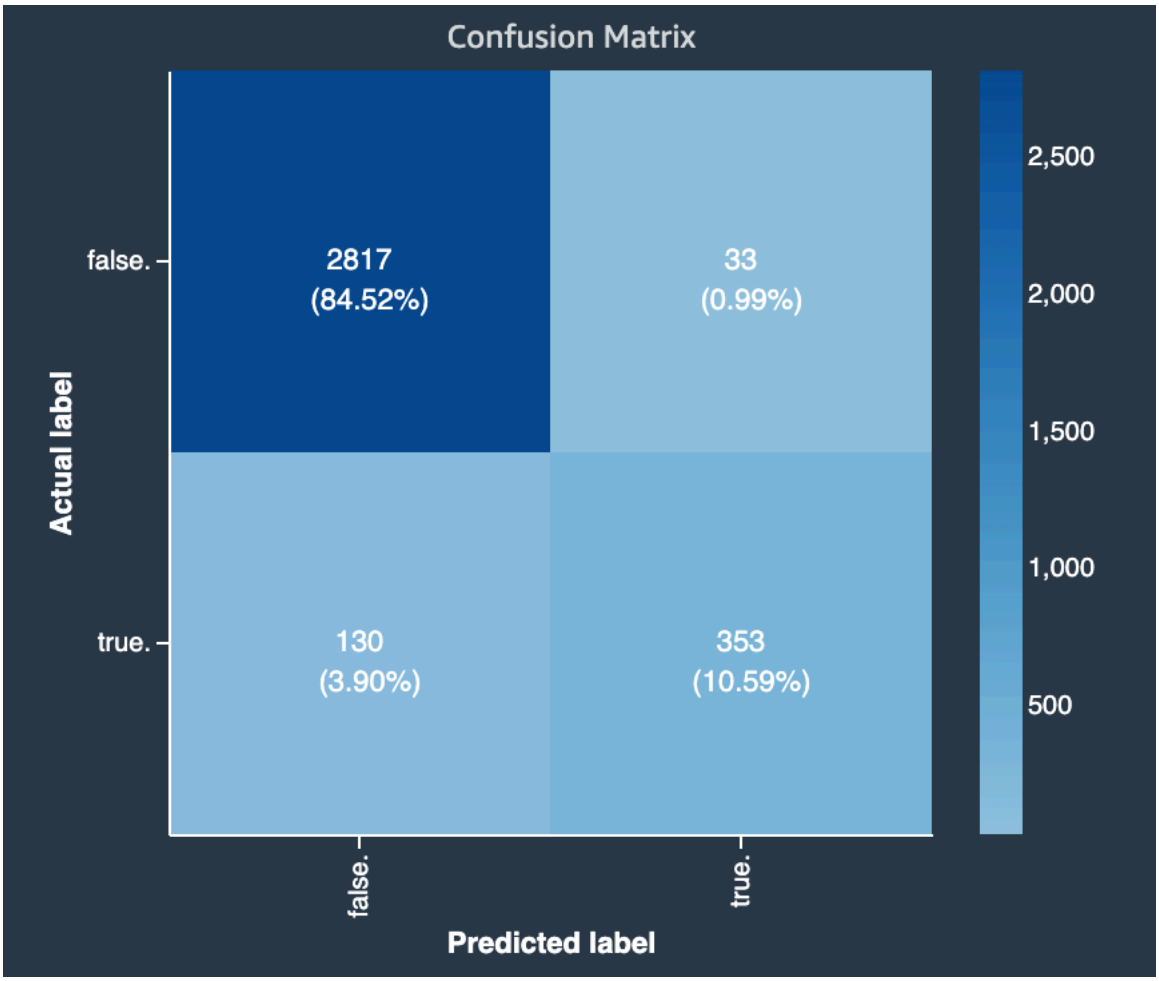
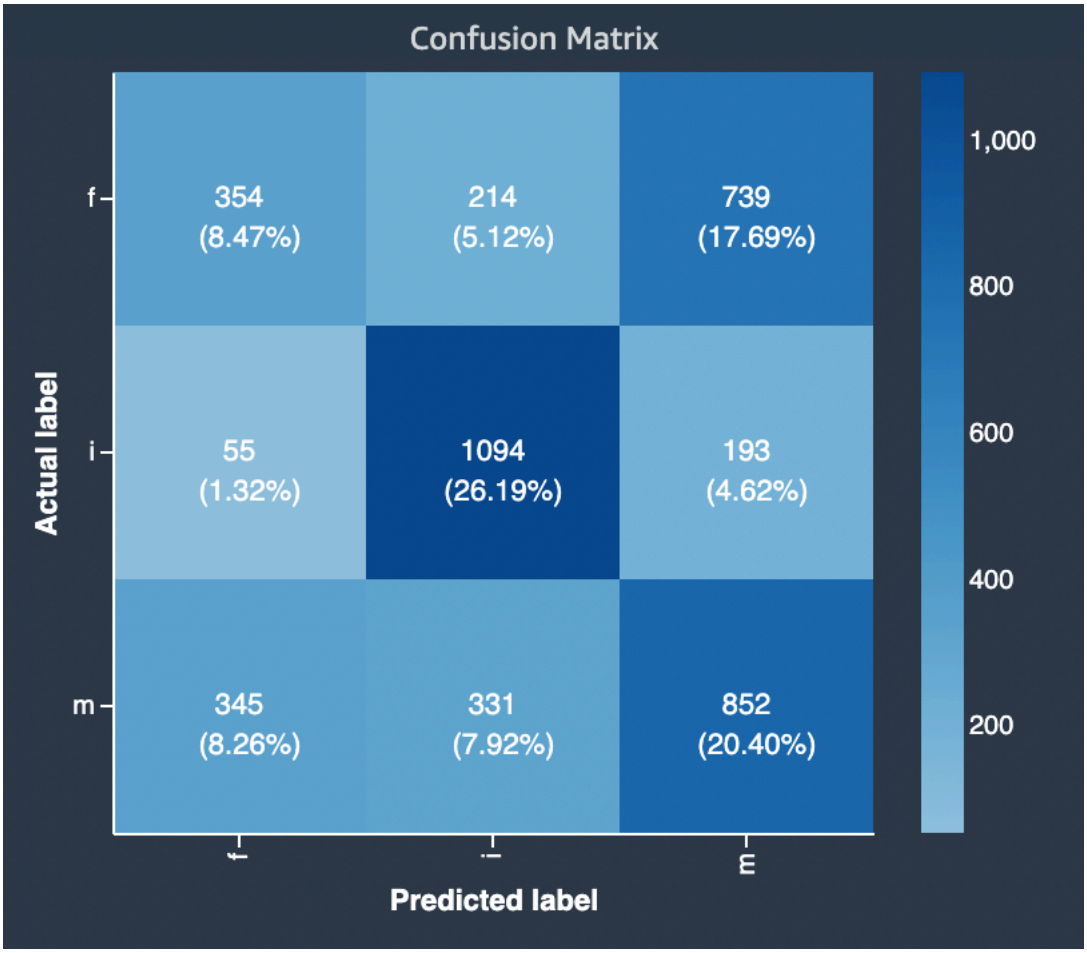


Diagram berikut adalah contoh matriks kebingungan untuk masalah klasifikasi multi-kelas. Matriks kebingungan dalam laporan kualitas model berisi yang berikut ini.

- Sumbu vertikal dibagi menjadi tiga baris yang berisi tiga label aktual yang berbeda.
- Sumbu horizontal dibagi menjadi tiga kolom yang berisi label yang diprediksi oleh model.
- Bilah warna memberikan nada yang lebih gelap ke sejumlah besar sampel untuk secara visual menunjukkan jumlah nilai yang diklasifikasikan dalam setiap kategori.

Dalam contoh di bawah ini, model dengan benar memprediksi 354 nilai aktual untuk label f, 1094 nilai untuk label i dan 852 nilai untuk label m. Perbedaan nada menunjukkan bahwa kumpulan data tidak seimbang karena ada lebih banyak label untuk nilai i daripada untuk f atau m.



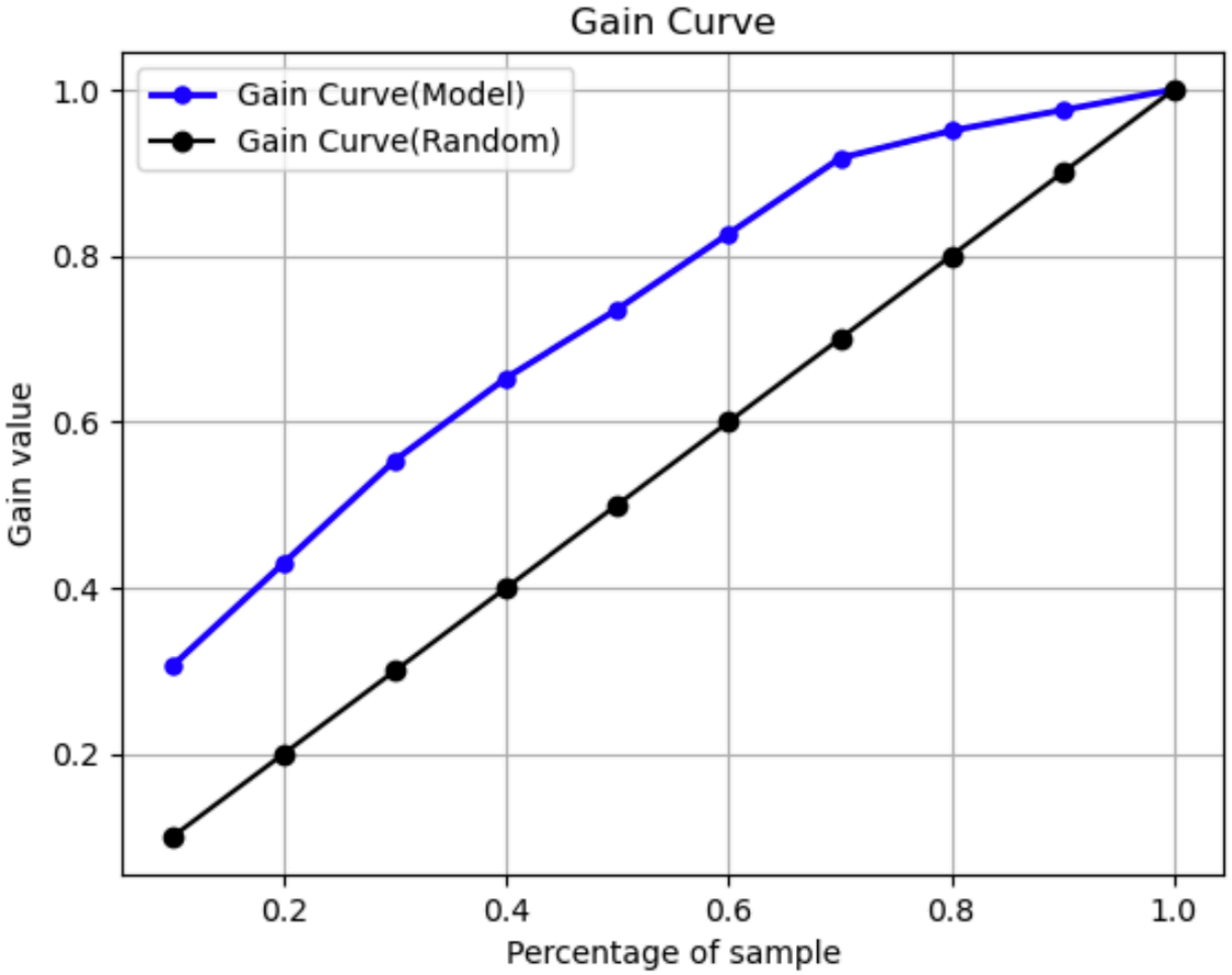
Matriks kebingungan dalam laporan kualitas model yang disediakan dapat mengakomodasi maksimum 15 label untuk jenis masalah klasifikasi multikelas. Jika baris yang sesuai dengan label menunjukkan Nan nilai, itu berarti kumpulan data validasi yang digunakan untuk memeriksa prediksi model tidak berisi data dengan label tersebut.

Kurva keuntungan

Dalam klasifikasi biner, kurva penguatan memprediksi manfaat kumulatif menggunakan persentase kumpulan data untuk menemukan label positif. Nilai gain dihitung selama pelatihan dengan membagi jumlah kumulatif pengamatan positif dengan jumlah total pengamatan positif dalam data, pada setiap desil. Jika model klasifikasi yang dibuat selama pelatihan mewakili data yang tidak terlihat, Anda dapat menggunakan kurva penguatan untuk memprediksi persentase data yang harus Anda targetkan untuk mendapatkan persentase label positif. Semakin besar persentase dataset yang digunakan, semakin tinggi persentase label positif yang ditemukan.

Dalam contoh grafik berikut, kurva penguatan adalah garis dengan kemiringan yang berubah. Garis lurus adalah persentase label positif yang ditemukan dengan memilih persentase data dari kumpulan

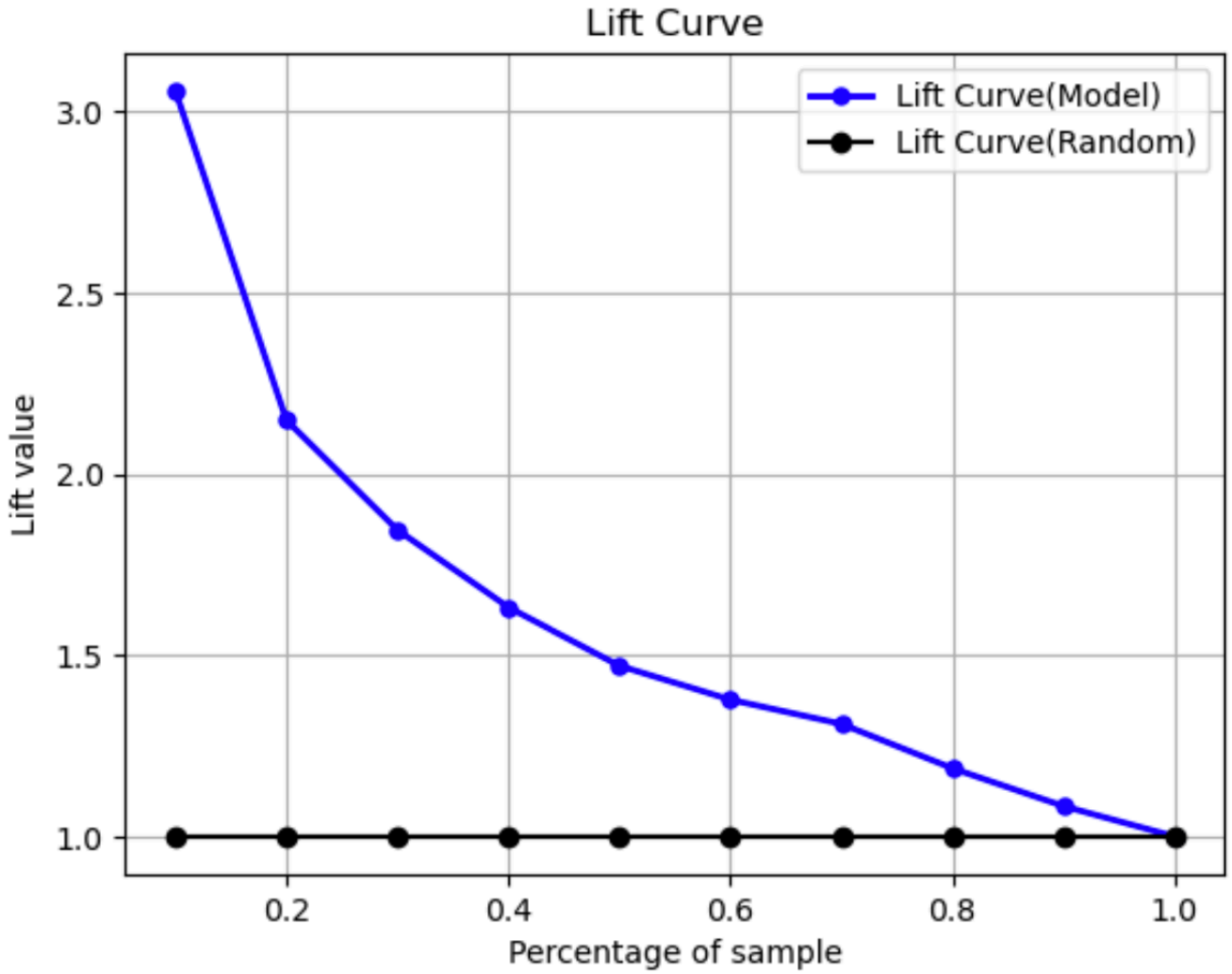
data secara acak. Setelah menargetkan 20% dari kumpulan data, Anda akan menemukan lebih besar dari 40% label positif. Misalnya, Anda dapat mempertimbangkan untuk menggunakan kurva keuntungan untuk menentukan upaya Anda dalam kampanye pemasaran. Menggunakan contoh kurva keuntungan kami, untuk 83% orang di lingkungan untuk membeli cookie, Anda akan mengirim iklan ke sekitar 60% dari lingkungan sekitar.



Kurva angkat

Dalam klasifikasi biner, kurva angkat menggambarkan peningkatan menggunakan model terlatih untuk memprediksi kemungkinan menemukan label positif dibandingkan dengan tebakan acak. Nilai angkat dihitung selama pelatihan menggunakan rasio kenaikan persentase dengan rasio label positif pada setiap desil. Jika model yang dibuat selama pelatihan mewakili data yang tidak terlihat, gunakan kurva angkat untuk memprediksi manfaat menggunakan model daripada menebak secara acak.

Pada contoh grafik berikut, kurva angkat adalah garis dengan kemiringan yang berubah. Garis lurus adalah kurva angkat yang terkait dengan pemilihan persentase yang sesuai secara acak dari kumpulan data. Setelah menargetkan 40% kumpulan data dengan label klasifikasi model Anda, Anda akan menemukan sekitar 1,7 kali jumlah label positif yang akan Anda temukan dengan memilih secara acak 40% dari data yang tidak terlihat.



Kurva penarikan presisi

Kurva recall presisi mewakili tradeoff antara presisi dan recall untuk masalah klasifikasi biner.

Presisi mengukur fraksi positif aktual yang diprediksi positif (TP) dari semua prediksi positif (TP dan positif palsu). Kisarannya adalah 0 hingga 1. Nilai yang lebih besar menunjukkan akurasi yang lebih baik dalam nilai yang diprediksi.

- Presisi = TP/ (TP+FP)

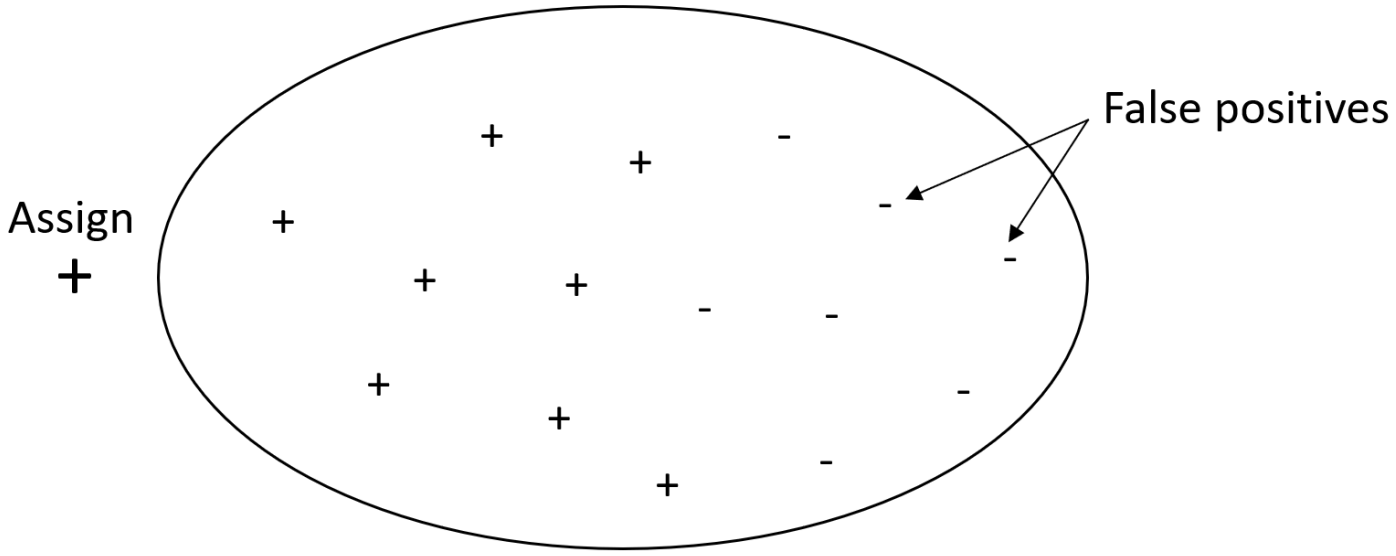
Ingat mengukur fraksi positif aktual (TP) yang diprediksi positif dari semua prediksi positif (TP dan negatif palsu). Ini juga dikenal sebagai sensitivitas dan sebagai tingkat positif sejati. Kisarannya adalah 0 hingga 1. Nilai yang lebih besar menunjukkan deteksi nilai positif yang lebih baik dari sampel.

- Ingat = TP/ (TP+FN)

Tujuan dari masalah klasifikasi adalah untuk memberi label dengan benar sebanyak mungkin elemen. Sebuah sistem dengan recall tinggi tetapi presisi rendah mengembalikan persentase positif palsu yang tinggi.

Grafik berikut menggambarkan filter spam yang menandai setiap email sebagai spam. Ini memiliki daya ingat tinggi, tetapi presisi rendah, karena mengingat tidak mengukur positif palsu.

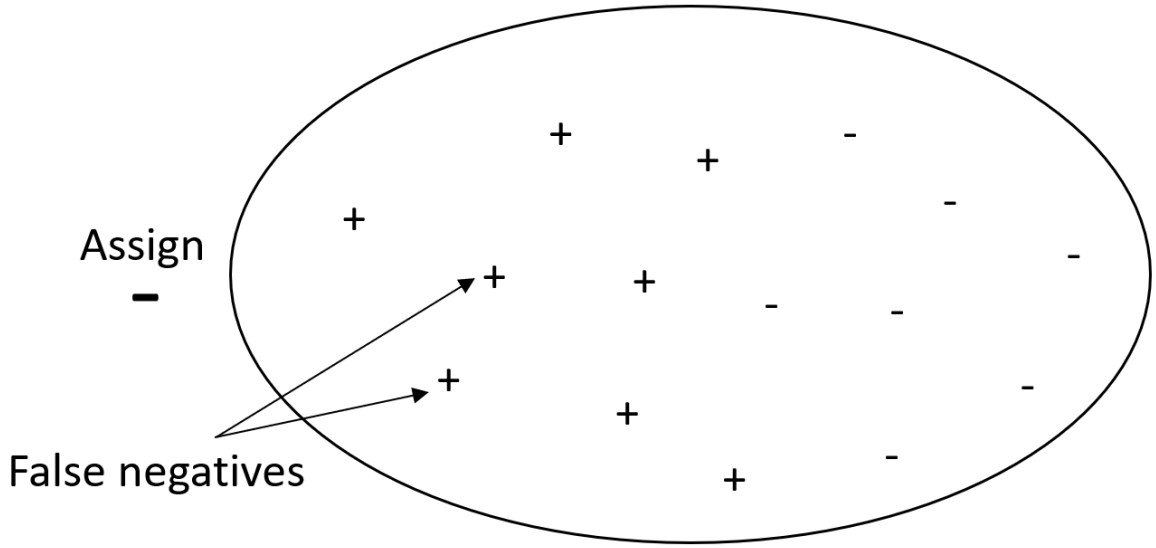
Berikan lebih banyak bobot untuk mengingat lebih presisi jika masalah Anda memiliki penalti rendah untuk nilai positif palsu, tetapi penalti tinggi karena kehilangan hasil positif yang sebenarnya. Misalnya, mendeteksi tabrakan yang akan datang di kendaraan self-driving.



Sebaliknya, sistem dengan presisi tinggi, tetapi penarikan rendah, mengembalikan persentase negatif palsu yang tinggi. Filter spam yang menandai setiap email sebagai diinginkan (bukan spam) memiliki presisi tinggi tetapi penarikan rendah karena presisi tidak mengukur negatif palsu.

Jika masalah Anda memiliki penalti rendah untuk nilai negatif palsu, tetapi penalti tinggi karena kehilangan hasil negatif yang sebenarnya, berikan bobot lebih pada presisi daripada mengingat. Misalnya, menandai filter mencurigakan untuk audit pajak.

Grafik berikut menggambarkan filter spam yang memiliki presisi tinggi tetapi daya ingat rendah, karena presisi tidak mengukur negatif palsu.

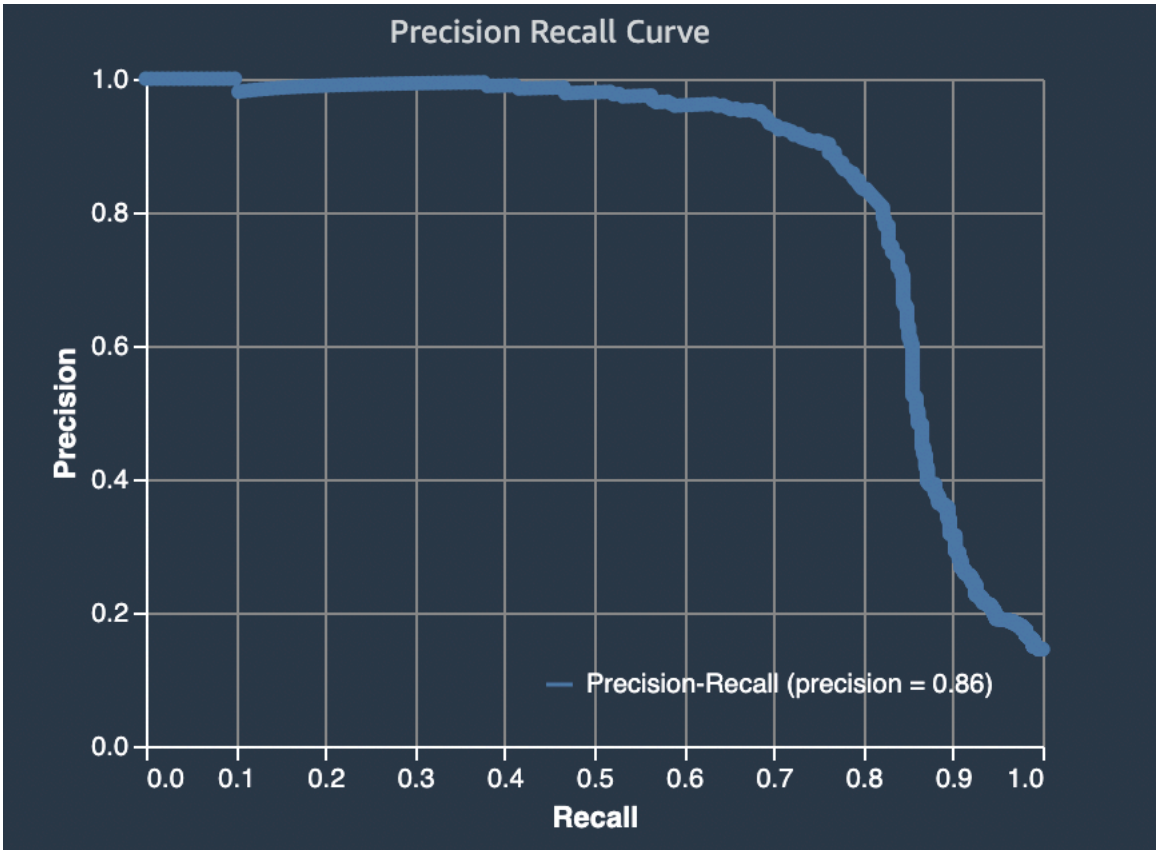


Model yang membuat prediksi dengan presisi tinggi dan ingatan tinggi menghasilkan sejumlah besar hasil yang diberi label dengan benar. Untuk informasi lebih lanjut, lihat artikel Presisi dan ingat di Wikipedia.

Area di bawah kurva penarikan presisi (AUPRC)

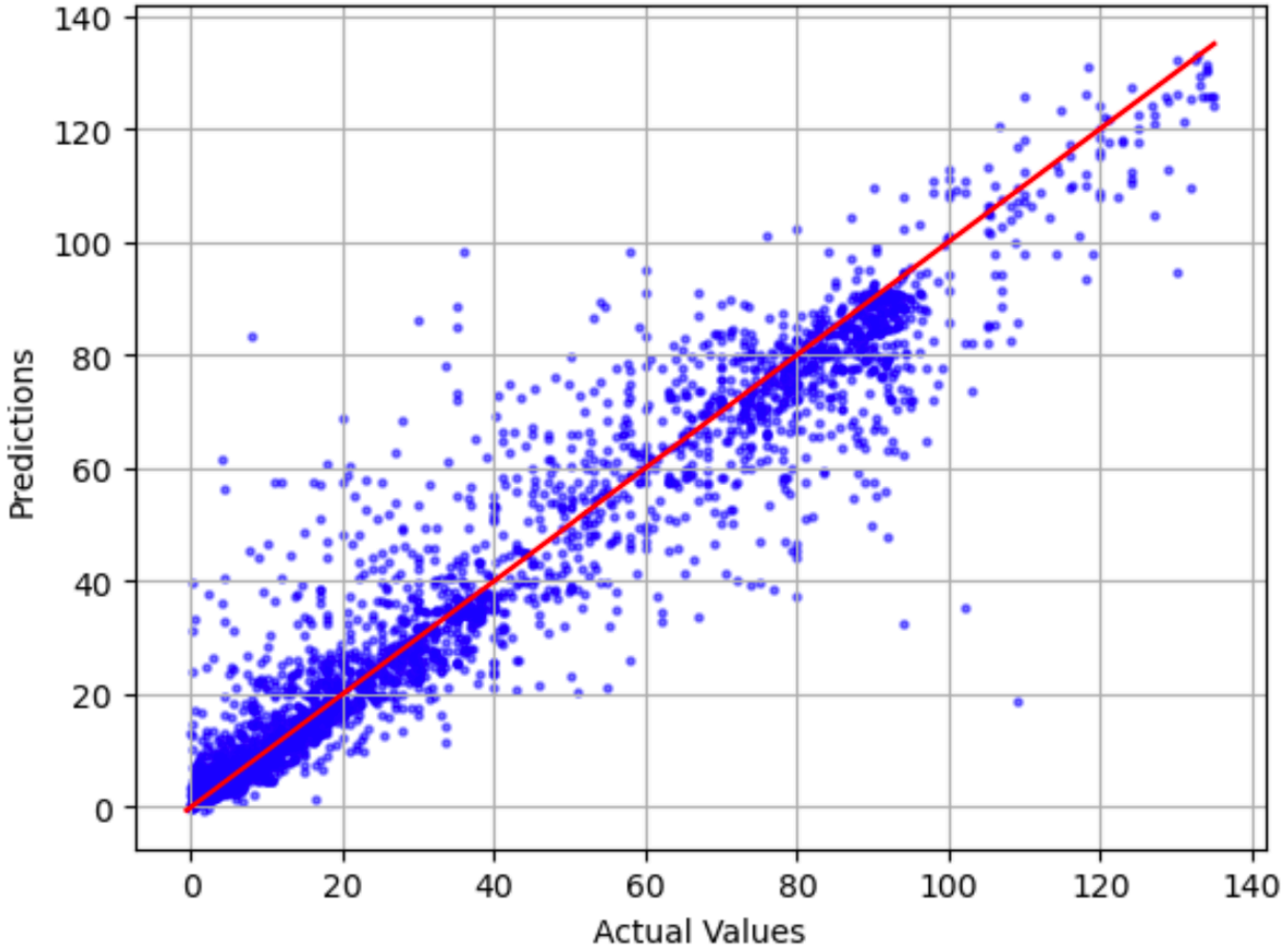
Untuk masalah klasifikasi biner, Amazon SageMaker Autopilot menyertakan grafik area di bawah kurva penarikan presisi (AUPRC). Metrik AUPRC memberikan ukuran agregat dari kinerja model di semua ambang klasifikasi yang mungkin dan menggunakan presisi dan penarikan. AUPRC tidak memperhitungkan jumlah negatif sejati. Oleh karena itu, dapat berguna untuk mengevaluasi kinerja model dalam kasus di mana ada sejumlah besar negatif sejati dalam data. Misalnya, untuk memodelkan gen yang mengandung mutasi langka.

Grafik berikut adalah contoh grafik AUPRC. Presisi pada nilai tertinggi adalah 1, dan recall pada 0. Di sudut kanan bawah grafik, recall adalah nilai tertinggi (1) dan presisi adalah 0. Di antara dua titik ini, kurva AUPRC menggambarkan tradeoff antara presisi dan recall pada ambang batas yang berbeda.



Aktual terhadap plot yang diprediksi

Plot aktual terhadap prediksi menunjukkan perbedaan antara nilai model aktual dan prediksi. Dalam contoh grafik berikut, garis padat adalah garis linier yang paling cocok. Jika modelnya 100% akurat, setiap titik yang diprediksi akan sama dengan titik aktual yang sesuai dan terletak pada garis yang paling cocok ini. Jarak jauh dari garis yang paling cocok adalah indikasi visual kesalahan model. Semakin besar jarak dari garis yang paling cocok, semakin tinggi kesalahan model.



Plot residu standar

Plot residu standar menggabungkan istilah statistik berikut:

`residual`

Sisa (mentah) menunjukkan perbedaan antara aktual dan nilai yang diprediksi oleh model Anda. Semakin besar perbedaannya, semakin besar nilai residu.

`standard deviation`

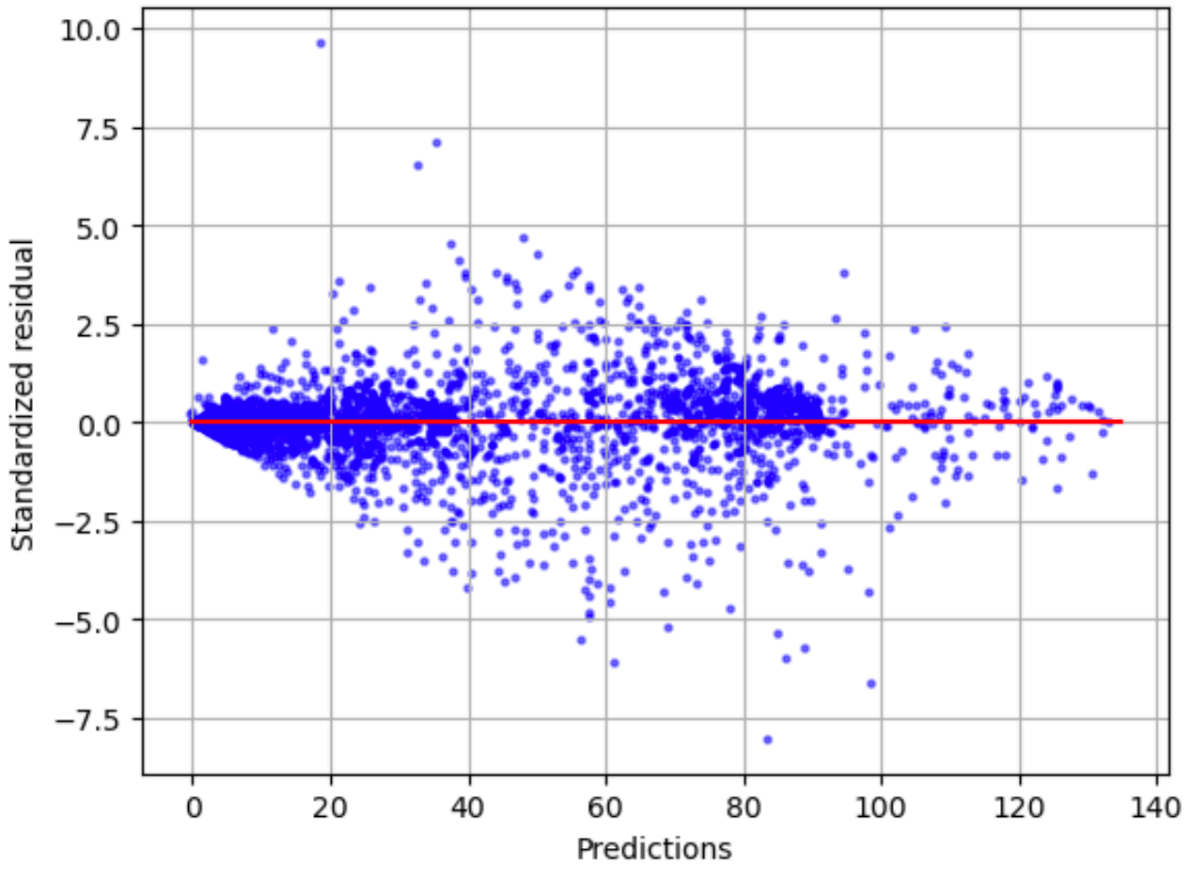
Standar deviasi adalah ukuran bagaimana nilai bervariasi dari nilai rata-rata. Standar deviasi yang tinggi menunjukkan bahwa banyak nilai sangat berbeda dari nilai rata-ratanya. Standar deviasi yang rendah menunjukkan bahwa banyak nilai mendekati nilai rata-ratanya.

`standardized residual`

Residu standar membagi residu mentah dengan standar deviasi mereka. Residu standar memiliki satuan standar deviasi dan berguna dalam mengidentifikasi outlier dalam data terlepas dari perbedaan skala residu mentah. Jika residu standar jauh lebih kecil atau lebih besar daripada residu standar lainnya, ini menunjukkan bahwa model tersebut tidak sesuai dengan pengamatan ini dengan baik.

Plot residu standar mengukur kekuatan perbedaan antara nilai yang diamati dan yang diharapkan. Nilai prediksi aktual ditampilkan pada sumbu x. Titik dengan nilai lebih besar dari nilai absolut 3 umumnya dianggap sebagai outlier.

Contoh grafik berikut menunjukkan bahwa sejumlah besar residu standar dikelompokkan sekitar 0 pada sumbu horizontal. Nilai mendekati nol menunjukkan bahwa model cocok dengan titik-titik ini dengan baik. Titik-titik ke arah atas dan bawah plot tidak diprediksi dengan baik oleh model.



Histogram sisa

Histogram residual menggabungkan istilah statistik berikut:

`residual`

Sisa (mentah) menunjukkan perbedaan antara aktual dan nilai yang diprediksi oleh model Anda. Semakin besar perbedaannya, semakin besar nilai residu.

`standard deviation`

Standar deviasi adalah ukuran seberapa banyak nilai bervariasi dari nilai rata-rata. Standar deviasi yang tinggi menunjukkan bahwa banyak nilai sangat berbeda dari nilai rata-ratanya. Standar deviasi yang rendah menunjukkan bahwa banyak nilai mendekati nilai rata-ratanya.

`standardized residual`

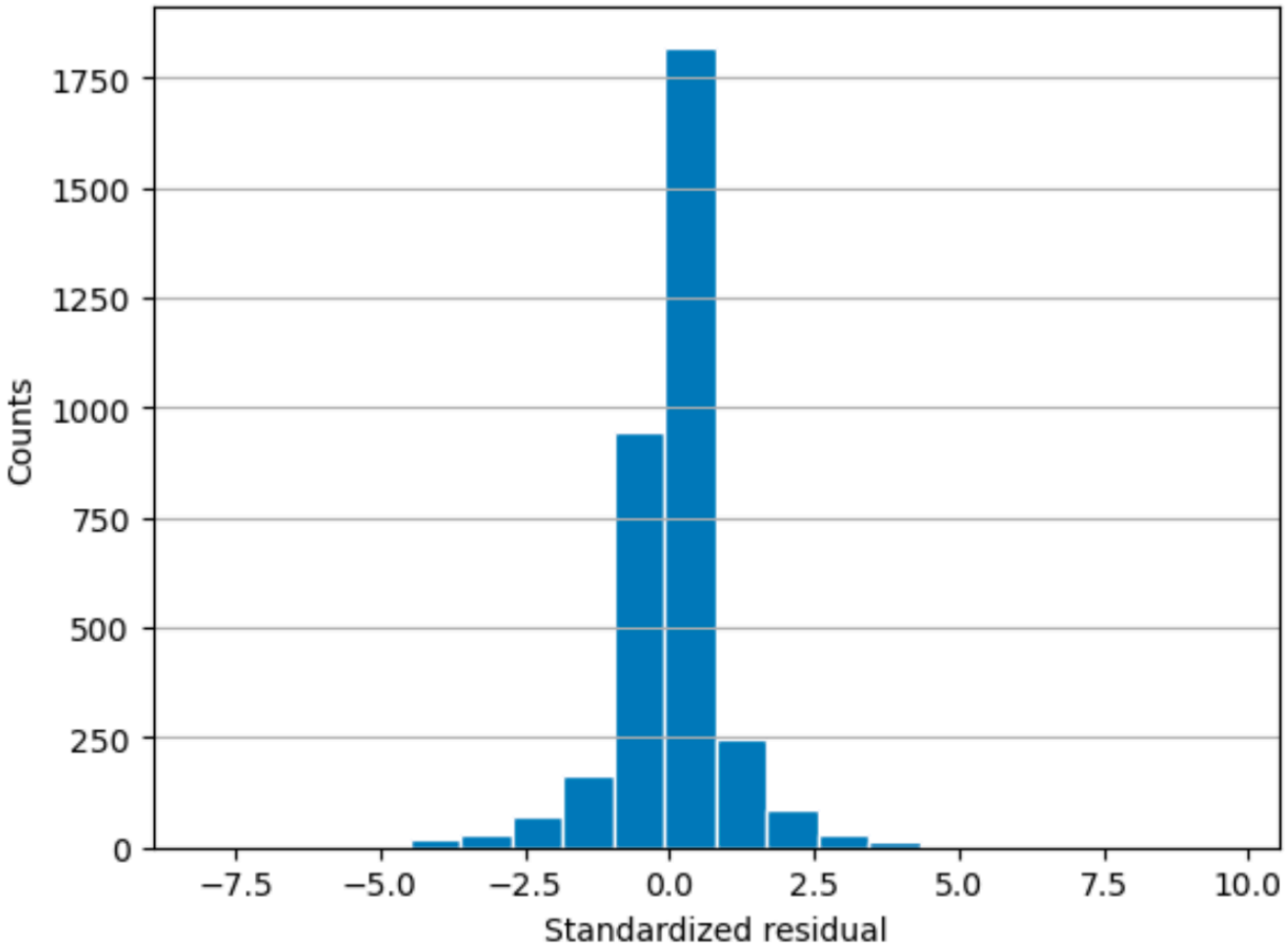
Residu standar membagi residu mentah dengan standar deviasi mereka. Residu standar memiliki satuan standar deviasi. Ini berguna dalam mengidentifikasi outlier dalam data terlepas dari perbedaan skala residu mentah. Jika residu standar jauh lebih kecil atau lebih besar daripada residu standar lainnya, itu akan menunjukkan bahwa model tersebut tidak sesuai dengan pengamatan ini dengan baik.

`histogram`

Histogram adalah grafik yang menunjukkan seberapa sering suatu nilai terjadi.

Histogram residu menunjukkan distribusi nilai residu standar. Histogram yang didistribusikan dalam bentuk lonceng dan berpusat pada nol menunjukkan bahwa model tidak secara sistematis memprediksi atau meremehkan rentang nilai target tertentu.

Dalam grafik berikut, nilai residu standar menunjukkan bahwa model tersebut sesuai dengan data dengan baik. Jika grafik menunjukkan nilai yang jauh dari nilai pusat, itu akan menunjukkan bahwa nilai-nilai tersebut tidak sesuai dengan model dengan baik.



Notebook Amazon SageMaker Autopilot dihasilkan untuk mengelola tugas AutoML

Amazon SageMaker Autopilot mengelola tugas-tugas utama dalam proses pembelajaran mesin otomatis (AutoML) menggunakan pekerjaan AutoML.

Pekerjaan AutoML membuat tiga laporan berbasis notebook yang menggambarkan rencana yang diikuti Autopilot untuk menghasilkan model kandidat. Model kandidat terdiri dari pasangan (pipeline, algoritma). Pertama, ada buku catatan eksplorasi data yang menjelaskan apa yang dipelajari Autopilot tentang data yang Anda berikan. Kedua, ada buku catatan definisi kandidat, yang menggunakan informasi tentang data untuk menghasilkan kandidat. Ketiga, laporan wawasan model yang dapat membantu merinci karakteristik kinerja model terbaik di papan peringkat eksperimen Autopilot.

Topik

- Amazon SageMaker Autopilot Laporan eksplorasi data

- Notebook definisi kandidat

Anda dapat menjalankan notebook ini di Amazon SageMaker, atau secara lokal, jika Anda telah menginstal Amazon Python SageMaker SDK. Anda dapat berbagi notebook seperti notebook SageMaker Studio Classic lainnya. Notebook dibuat untuk Anda melakukan eksperimen. Misalnya, Anda dapat mengedit item berikut di buku catatan:

- Preprocessors yang digunakan pada data
- Jumlah optimasi hyperparameter (HPO) berjalan dan paralelismenya
- Algoritma untuk dicoba
- Jenis instans yang digunakan untuk pekerjaan HPO
- Rentang hiperparameter

Modifikasi buku catatan definisi kandidat didorong sebagai alat pembelajaran. Dengan kemampuan ini, Anda mempelajari bagaimana keputusan yang dibuat selama proses pembelajaran mesin memengaruhi hasil Anda.

> **Note**
>
> Ketika Anda menjalankan notebook dalam instance default Anda, Anda dikenakan biaya dasar. Namun, ketika Anda menjalankan pekerjaan HPO dari notebook kandidat, pekerjaan ini menggunakan sumber daya komputasi tambahan yang menimbulkan biaya tambahan.

Amazon SageMaker Autopilot Laporan eksplorasi data

Amazon SageMaker Autopilot membersihkan dan memproses kumpulan data Anda secara otomatis. Data berkualitas tinggi meningkatkan efisiensi pembelajaran mesin dan menghasilkan model yang membuat prediksi yang lebih akurat.

Ada masalah dengan kumpulan data yang disediakan pelanggan yang tidak dapat diperbaiki secara otomatis tanpa manfaat dari beberapa pengetahuan domain. Nilai outlier besar di kolom target untuk masalah regresi, misalnya, dapat menyebabkan prediksi suboptimal untuk nilai non-outlier. Outlier mungkin perlu dihapus tergantung pada tujuan pemodelan. Jika kolom target dimasukkan secara tidak sengaja sebagai salah satu fitur input, model akhir akan memvalidasi dengan baik, tetapi memiliki nilai kecil untuk prediksi masa depan.

Untuk membantu pelanggan menemukan masalah semacam ini, Autopilot menyediakan laporan eksplorasi data yang berisi wawasan tentang potensi masalah dengan data mereka. Laporan ini juga menyarankan bagaimana menangani masalah.

Buku catatan eksplorasi data yang berisi laporan dibuat untuk setiap pekerjaan Autopilot. Laporan disimpan dalam bucket Amazon S3 dan dapat diakses dari jalur keluaran Anda. Jalur laporan eksplorasi data biasanya mengikuti pola berikut.

```
[s3 output path]/[name of the automl job]/sagemaker-automl-
candidates/[name of processing job used for data analysis]/notebooks/
SageMakerAutopilotDataExplorationNotebook.ipynb
```

Lokasi notebook eksplorasi data dapat diperoleh dari Autopilot API menggunakan respons `DescribeAutoMLJob`operasi, yang disimpan di. DataExplorationNotebookLocation

Saat menjalankan Autopilot dari SageMaker Studio Classic, Anda dapat membuka laporan eksplorasi data menggunakan langkah-langkah berikut:

1. Pilih ikon Beranda

   dari panel navigasi kiri untuk melihat menu navigasi Amazon SageMaker Studio Classic tingkat atas.
2. Pilih kartu AutoML dari area kerja utama. Ini membuka tab Autopilot baru.
3. Di bagian Nama, pilih pekerjaan Autopilot yang memiliki buku catatan eksplorasi data yang ingin Anda periksa. Ini membuka tab pekerjaan Autopilot baru.
4. Pilih Buka buku catatan eksplorasi data dari bagian kanan atas tab pekerjaan Autopilot.

Laporan eksplorasi data dihasilkan dari data Anda sebelum proses pelatihan dimulai. Ini memungkinkan Anda untuk menghentikan pekerjaan Autopilot yang mungkin mengarah pada hasil yang tidak berarti. Demikian juga, Anda dapat mengatasi masalah atau peningkatan apa pun dengan kumpulan data Anda sebelum menjalankan ulang Autopilot. Dengan cara ini, Anda dapat menggunakan keahlian domain Anda untuk meningkatkan kualitas data secara manual, sebelum Anda melatih model pada kumpulan data yang dikuratori dengan lebih baik.

Laporan data hanya berisi penurunan harga statis dan dapat dibuka di lingkungan Jupyter apa pun. Notebook yang berisi laporan dapat dikonversi ke format lain, seperti PDF atau HTML. Untuk informasi selengkapnya tentang konversi, lihat Menggunakan skrip nbconvert untuk mengonversi buku catatan Jupyter ke format lain. .

Topik

Ringkasan Dataset

Ringkasan Set Data ini menyediakan statistik utama yang mengkarakterisasi kumpulan data Anda termasuk jumlah baris, kolom, persen baris duplikat, dan nilai target yang hilang. Ini dimaksudkan untuk memberi Anda peringatan cepat ketika ada masalah dengan kumpulan data Anda yang terdeteksi Amazon SageMaker Autopilot dan yang kemungkinan memerlukan intervensi Anda. Wawasan muncul sebagai peringatan yang diklasifikasikan sebagai tingkat keparahan “tinggi” atau “rendah”. Klasifikasi tergantung pada tingkat kepercayaan bahwa masalah tersebut akan berdampak buruk pada kinerja model.

Wawasan tingkat keparahan tinggi dan rendah muncul dalam ringkasan sebagai pop-up. Untuk sebagian besar wawasan, rekomendasi ditawarkan untuk cara mengonfirmasi bahwa ada masalah dengan kumpulan data yang memerlukan perhatian Anda. Proposal juga disediakan untuk cara menyelesaikan masalah.

Autopilot memberikan statistik tambahan tentang nilai target yang hilang atau tidak valid dalam kumpulan data kami untuk membantu Anda mendeteksi masalah lain yang mungkin tidak ditangkap oleh wawasan tingkat keparahan tinggi. Jumlah kolom yang tidak terduga dari jenis tertentu mungkin menunjukkan bahwa beberapa kolom yang ingin Anda gunakan mungkin hilang dari kumpulan data. Ini juga bisa menunjukkan bahwa ada masalah dengan bagaimana data disiapkan atau disimpan. Memperbaiki masalah data yang dibawa ke perhatian Anda oleh Autopilot dapat meningkatkan kinerja model pembelajaran mesin yang dilatih pada data Anda.

Wawasan tingkat keparahan tinggi ditampilkan di bagian ringkasan dan di bagian lain yang relevan dalam laporan. Contoh wawasan tingkat keparahan tinggi dan rendah biasanya diberikan tergantung pada bagian laporan data.

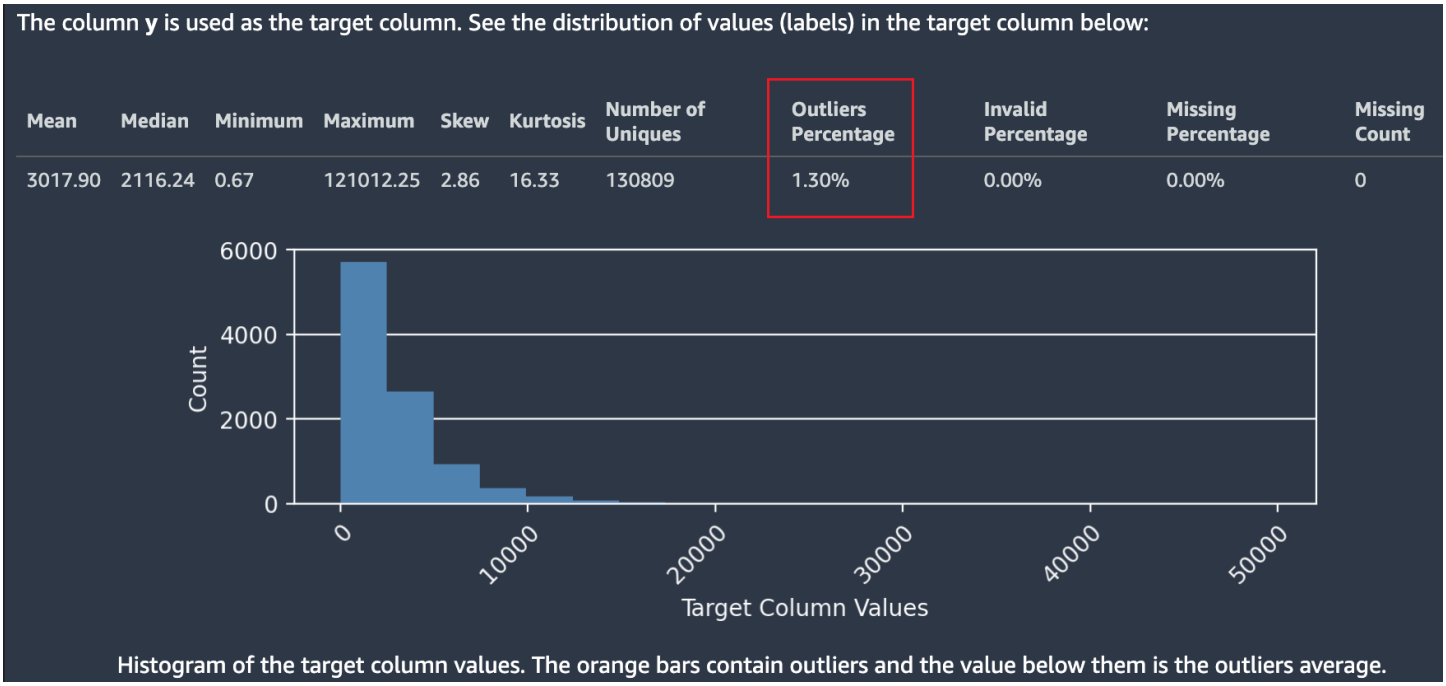
Analisis Target

Berbagai wawasan tingkat keparahan tinggi dan rendah ditampilkan di bagian ini terkait dengan distribusi nilai di kolom target. Periksa apakah kolom target berisi nilai yang benar. Nilai yang salah di kolom target kemungkinan akan menghasilkan model pembelajaran mesin yang tidak melayani tujuan bisnis yang dimaksudkan. Beberapa wawasan data tingkat keparahan tinggi dan rendah hadir di bagian ini. Berikut adalah beberapa contoh tanda.

- Nilai target outlier - Distribusi target miring atau tidak biasa untuk regresi, seperti target berekor berat.
- Kardinalitas target tinggi atau rendah - Jumlah label kelas yang jarang atau sejumlah besar kelas unik untuk klasifikasi.

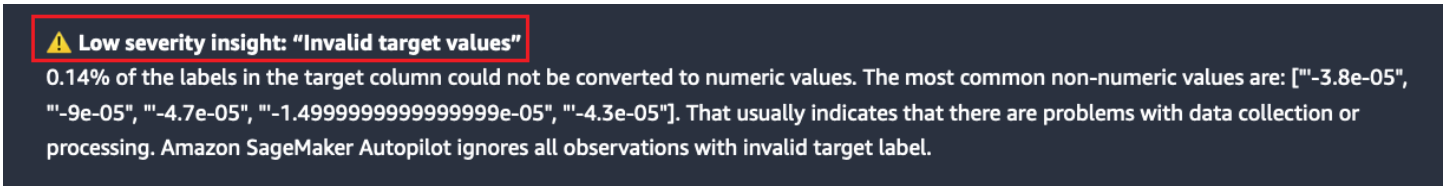
Untuk jenis masalah regresi dan klasifikasi, nilai yang tidak valid seperti tak terhingga numerik, `NaN` atau ruang kosong di kolom target muncul. Tergantung pada jenis masalah, statistik dataset yang berbeda disajikan. Distribusi nilai kolom target untuk masalah regresi memungkinkan Anda memverifikasi apakah distribusi sesuai dengan yang Anda harapkan.

Tangkapan layar berikut menunjukkan laporan data Autopilot, yang mencakup statistik seperti rata-rata, median, minimum, maksimum, persentase outlier dalam kumpulan data Anda. Tangkapan layar juga menyertakan histogram yang menunjukkan distribusi label di kolom target. Histogram menunjukkan Nilai Kolom Target pada sumbu horizontal dan Hitung pada sumbu vertikal. Sebuah kotak menyoroti bagian Persentase Outliers pada tangkapan layar untuk menunjukkan di mana statistik ini muncul.



Beberapa statistik ditampilkan mengenai nilai target dan distribusinya. Jika salah satu outlier, bukan nilai yang valid, atau persentase yang hilang lebih besar dari nol, nilai ini muncul sehingga Anda dapat menyelidiki mengapa data Anda berisi nilai target yang tidak dapat digunakan. Beberapa nilai target yang tidak dapat digunakan disorot sebagai peringatan wawasan tingkat keparahan rendah.

Pada tangkapan layar berikut, simbol `ditambahkan secara tidak sengaja ke kolom target, yang mencegah nilai numerik target diurai. Wawasan tingkat keparahan rendah: Peringatan “Nilai target tidak valid” muncul. Peringatan dalam contoh ini menyatakan "0,14% label di kolom target tidak dapat dikonversi ke nilai numerik. Nilai non-numerik yang paling umum adalah: [” -3.8e-05",” -9-05",” -4.7e-05",” -1.4999999999999999e-05",” -4.3e-05"]. Itu biasanya menunjukkan bahwa ada masalah dengan pengumpulan atau pemrosesan data. Amazon SageMaker Autopilot mengabaikan semua pengamatan dengan label target yang tidak valid.

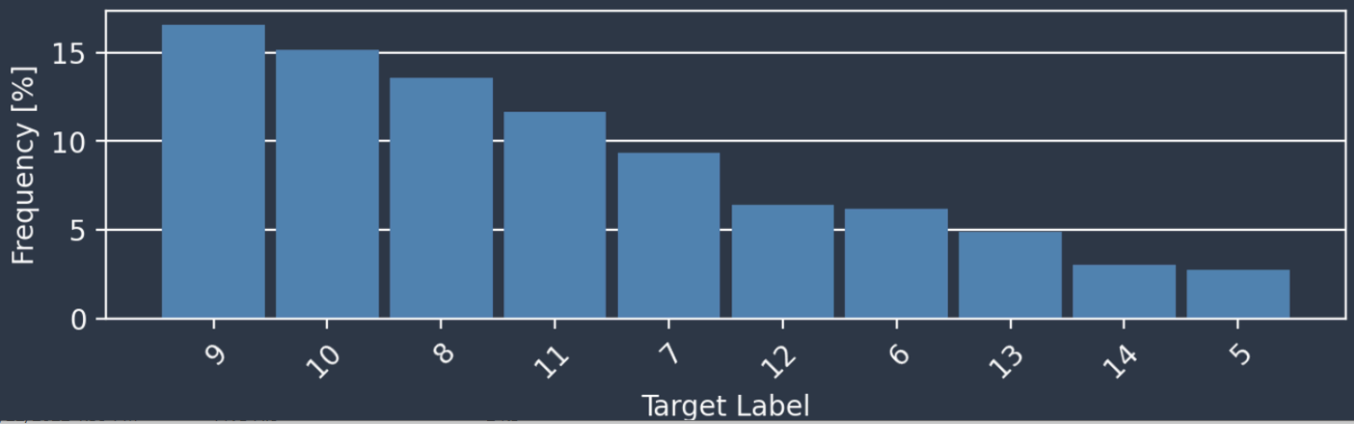


Autopilot juga menyediakan histogram yang menunjukkan distribusi label untuk klasifikasi.

Tangkapan layar berikut menunjukkan contoh statistik yang diberikan untuk kolom target Anda termasuk jumlah kelas, nilai yang hilang atau tidak valid. Histogram dengan Label Target pada sumbu horizontal dan Frekuensi pada sumbu vertikal menunjukkan distribusi setiap kategori label.

Number of Classes	Invalid Percentage	Missing Percentage
23	0.00%	0.00%

Target Label	Frequency Percentage	Label Count
9	16.55%	230
10	15.18%	211
8	13.60%	189
11	11.65%	162
7	9.35%	130
12	6.40%	89
6	6.19%	86
13	4.89%	68
14	3.02%	42
5	2.73%	38



> **Note**
>
> Anda dapat menemukan definisi dari semua istilah yang disajikan dalam bagian ini dan lainnya di bagian Definisi di bagian bawah buku catatan laporan.

Sampel Data

Autopilot menyajikan sampel aktual data Anda untuk membantu Anda menemukan masalah dengan kumpulan data Anda. Tabel sampel bergulir secara horizontal. Periksa data sampel untuk memverifikasi bahwa semua kolom yang diperlukan ada dalam kumpulan data.

Autopilot juga menghitung ukuran daya prediksi, yang dapat digunakan untuk mengidentifikasi hubungan linier atau nonlinier antara fitur dan variabel target. Nilai `0` menunjukkan bahwa fitur tersebut tidak memiliki nilai prediktif dalam memprediksi variabel target. Nilai 1 menunjukkan daya prediksi tertinggi untuk variabel target. Untuk informasi lebih lanjut tentang kekuatan prediksi, lihat bagian Definisi.

> **Note**
>
> Anda tidak disarankan untuk menggunakan kekuatan prediksi sebagai pengganti kepentingan fitur. Gunakan hanya jika Anda yakin bahwa kekuatan prediksi adalah ukuran yang tepat untuk kasus penggunaan Anda.

Tangkapan layar berikut menunjukkan contoh sampel data. Baris atas berisi kekuatan prediksi setiap kolom dalam kumpulan data Anda. Baris kedua berisi tipe data kolom. Baris berikutnya berisi label. Kolom berisi kolom target diikuti oleh setiap kolom fitur. Setiap kolom fitur memiliki kekuatan prediksi terkait, disorot dalam tangkapan layar ini, dengan sebuah kotak. Dalam contoh ini, kolom yang berisi fitur `x51` memiliki kekuatan prediksi `0.68` untuk variabel y target. Fitur `x55` ini sedikit kurang prediktif dengan kekuatan prediksi. `0.59`

	y	x51	x55	x54	x52	x20	x56	x15
Prediction Power	-	0.680107	0.594356	0.580346	0.548662	0.543034	0.480431	0.448701
Column Types	-	numeric	numeric	numeric	numeric	numeric	numeric	numeric
0	0.0	0.0	2.0	1.4280000000000002	0.0	0.0	10.0	0.0
1	1.0	0.152	19.0	1.357	0.0	1.18	148.0	0.0
2	1.0	0.0	46.0	4.8180000000000005	0.0	2.63	106.0	1.31
3	0.0	0.134	121.0	3.08	0.0	1.56	693.0	0.0
4	0.0	0.377	1.0	1.0	0.0	0.0	33.0	0.0
5	0.0	0.0	1.0	1.0	0.0	0.0	10.0	0.0
6	0.0	0.327	2.0	1.068	0.0	0.61	47.0	0.0
7	0.0	0.039	6.0	1.2919999999999998	0.0	0.42	106.0	0.21

Baris duplikat

Jika baris duplikat ada dalam kumpulan data, Amazon SageMaker Autopilot menampilkan sampelnya.

> **Note**
>
> Tidak disarankan untuk menyeimbangkan kumpulan data dengan up-sampling sebelum memberikannya ke Autopilot. Hal ini dapat mengakibatkan skor validasi yang tidak akurat untuk model yang dilatih oleh Autopilot, dan model yang diproduksi mungkin tidak dapat digunakan.

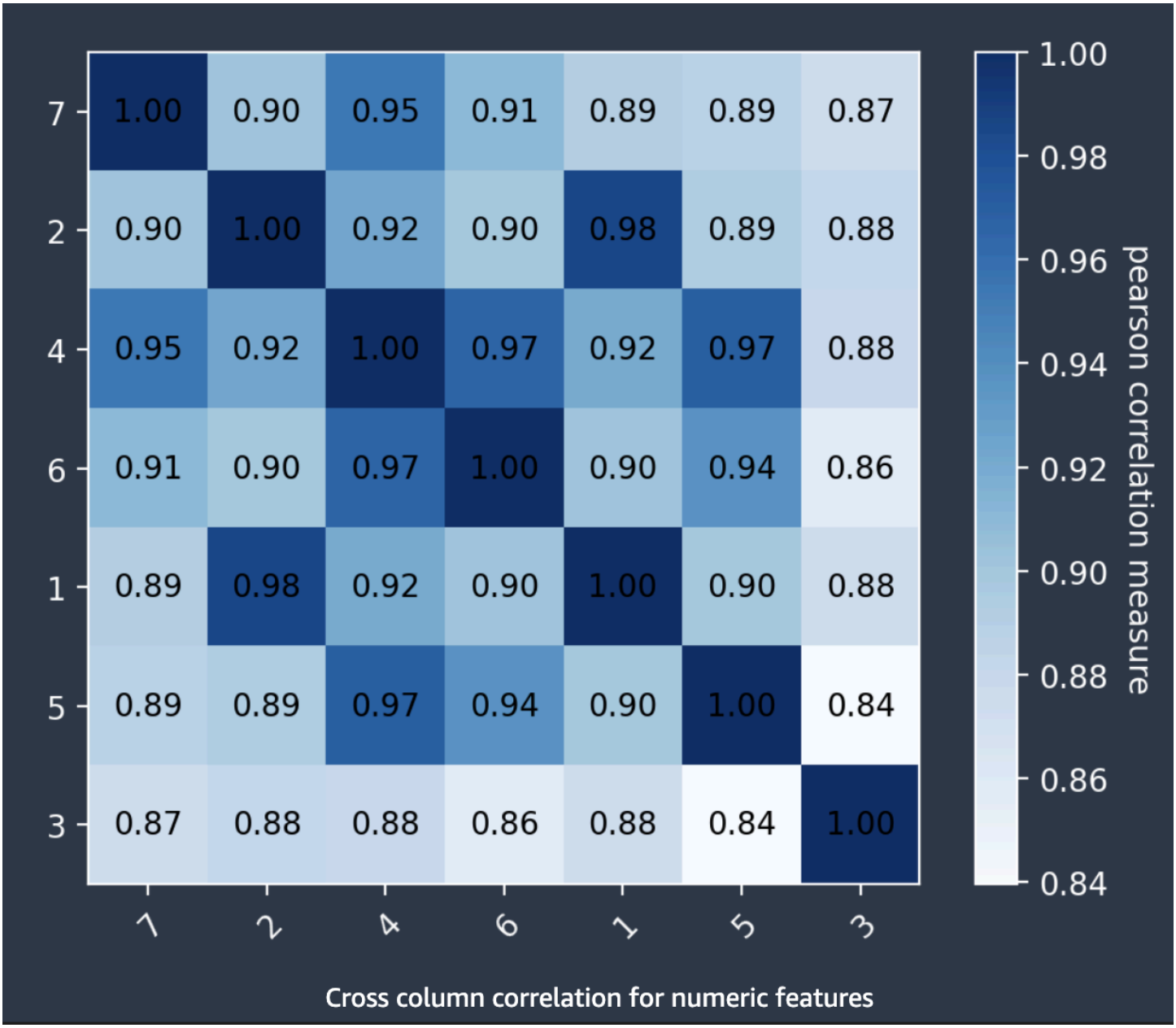
Korelasi kolom silang

Autopilot menggunakan koefisien korelasi Pearson, ukuran korelasi linier antara dua fitur, untuk mengisi matriks korelasi. Dalam matriks korelasi, fitur numerik diplot pada sumbu horizontal dan vertikal, dengan koefisien korelasi Pearson diplot di persimpangan mereka. Semakin tinggi korelasi antara dua fitur, semakin tinggi koefisiennya, dengan nilai maksimum. `|1|`

- Nilai `-1` menunjukkan bahwa fitur berkorelasi negatif sempurna.
- Nilai1, yang terjadi ketika suatu fitur berkorelasi dengan dirinya sendiri, menunjukkan korelasi positif yang sempurna.

Anda dapat menggunakan informasi dalam matriks korelasi untuk menghapus fitur yang sangat berkorelasi. Sejumlah kecil fitur mengurangi kemungkinan overfitting model dan dapat mengurangi biaya produksi dengan dua cara. Ini mengurangi runtime Autopilot yang dibutuhkan dan, untuk beberapa aplikasi, dapat membuat prosedur pengumpulan data lebih murah.

Tangkapan layar berikut menunjukkan contoh matriks korelasi antar 7 fitur. Setiap fitur ditampilkan dalam matriks pada sumbu horizontal dan vertikal. Koefisien korelasi Pearson ditampilkan di persimpangan antara dua fitur. Setiap persimpangan fitur memiliki nada warna yang terkait dengannya. Semakin tinggi korelasinya, semakin gelap nadanya. Nada paling gelap menempati diagonal matriks, di mana setiap fitur berkorelasi dengan dirinya sendiri, mewakili korelasi sempurna.



Baris Anomali

Amazon SageMaker Autopilot mendeteksi baris mana dalam kumpulan data Anda yang mungkin anomali. Kemudian memberikan skor anomali untuk setiap baris. Baris dengan skor anomali negatif dianggap anomali.

Tangkapan layar berikut menunjukkan output dari analisis Autopilot untuk baris yang berisi anomali. Kolom yang berisi skor anomali muncul di sebelah kolom dataset untuk setiap baris.

	Anomaly Scores	0	1	2	3	4	5	6	7
1237	-0.215202	F	0.8	0.63	0.195	2.526	0.933	0.59	0.62
405	-0.200257	F	0.815	0.65	0.25	2.255	0.8905	0.42	0.7975
861	-0.194832	F	0.75	0.61	0.235	2.5085	1.232	0.519	0.612
1319	-0.193176	M	0.73	0.595	0.23	2.8255	1.1465	0.419	0.897
403	-0.184558	M	0.77	0.62	0.195	2.5155	1.1155	0.6415	0.642
229	-0.182169	F	0.735	0.6	0.22	2.555	1.1335	0.44	0.6
989	-0.171010	I	0.11	0.09	0.03	0.008	0.0025	0.002	0.003
1066	-0.160921	M	0.665	0.535	0.225	2.1835	0.7535	0.391	0.885
1056	-0.155347	I	0.14	0.105	0.035	0.014	0.0055	0.0025	0.004
637	-0.154234	M	0.175	0.125	0.04	0.024	0.0095	0.006	0.005

Nilai yang hilang, kardinalitas, dan statistik deskriptif

Amazon SageMaker Autopilot memeriksa dan melaporkan properti masing-masing kolom kumpulan data Anda. Di setiap bagian laporan data yang menyajikan analisis ini, konten disusun secara berurutan. Ini agar Anda dapat memeriksa nilai yang paling “mencurigakan” terlebih dahulu. Dengan menggunakan statistik ini Anda dapat meningkatkan konten kolom individual, dan meningkatkan kualitas model yang dihasilkan oleh Autopilot.

Autopilot menghitung beberapa statistik pada nilai kategoris dalam kolom yang berisi mereka. Ini termasuk jumlah entri unik dan, untuk teks, jumlah kata unik.

Autopilot menghitung beberapa statistik standar pada nilai numerik dalam kolom yang berisi mereka. Gambar berikut menggambarkan statistik ini, termasuk nilai rata-rata, median, minimum dan maksimum, dan persentase jenis numerik dan nilai outlier.

	% of Numerical Values	Mean	Median	Min	Max	% of Outlier Values
y	100.0%	9.93957	9.0	3.0	27.0	nan
1	100.0%	0.523612	0.545	0.11	0.815	0.0
2	100.0%	0.407799	0.425	0.09	0.65	0.0
3	100.0%	0.13995	0.145	0.015	0.515	0.1
4	100.0%	0.828266	0.81	0.008	2.8255	0.0
5	100.0%	0.358844	0.339	0.0025	1.2395	0.0
6	100.0%	0.180348	0.1725	0.002	0.6415	0.0
7	100.0%	0.238783	0.235	0.003	1.005	0.2

Notebook definisi kandidat

Notebook definisi kandidat berisi setiap langkah preprocessing yang disarankan, algoritma, dan rentang hyperparameter.

Anda dapat memilih kandidat mana yang akan dilatih dan disetel dengan dua cara. Yang pertama, dengan menjalankan bagian notebook. Kedua, dengan menjalankan seluruh notebook untuk mengoptimalkan semua kandidat untuk mengidentifikasi kandidat terbaik. Jika Anda menjalankan seluruh buku catatan, hanya kandidat terbaik yang ditampilkan setelah pekerjaan selesai.

Untuk menjalankan Autopilot dari SageMaker Studio Classic, buka notebook definisi kandidat dengan mengikuti langkah-langkah berikut:

1. Pilih ikon Beranda

   dari panel navigasi kiri untuk melihat menu navigasi Amazon SageMaker Studio Classic tingkat atas.
2. Pilih kartu AutoML dari area kerja utama. Ini membuka tab Autopilot baru.
3. Di bagian Nama, pilih pekerjaan Autopilot yang memiliki buku catatan definisi kandidat yang ingin Anda periksa. Ini membuka tab pekerjaan Autopilot baru.

4. Pilih Buka buku catatan generasi kandidat dari bagian kanan atas tab pekerjaan Autopilot. Ini membuka pratinjau hanya-baca baru dari Notebook Definisi Kandidat Amazon SageMaker Autopilot.

Untuk menjalankan buku catatan definisi kandidat, ikuti langkah-langkah berikut:

1. Pilih Impor buku catatan di kanan atas tab Notebook Definisi Kandidat Amazon SageMaker Autopilot. Ini membuka tab untuk mengatur lingkungan notebook baru untuk menjalankan notebook.
2. Pilih SageMaker Gambar yang ada atau gunakan Gambar Kustom.
3. Pilih Kernel, tipe Instance, dan skrip Start-up opsional.

Anda sekarang dapat menjalankan notebook di lingkungan baru ini.

Konfigurasikan output inferensi dalam wadah yang dihasilkan

Autopilot menghasilkan daftar yang diurutkan. `ContainerDefinition` Ini dapat digunakan untuk membangun model untuk diterapkan dalam pipa pembelajaran mesin. Model ini dapat digunakan untuk hosting online dan inferensi.

Pelanggan dapat membuat daftar definisi kontainer inferensi dengan `ListCandidateForAutoMLJob`API. Daftar definisi wadah inferensi yang mewakili kandidat terbaik juga tersedia dalam `DescribeAutoMLJob`tanggapan.

Definisi wadah inferensi untuk jenis masalah regresi dan klasifikasi

Autopilot menghasilkan wadah inferensi khusus untuk mode pelatihan dan jenis masalah pekerjaan.

Definisi wadah untuk mode optimasi hyperparameter (HPO)

- Regresi: HPO menghasilkan dua kontainer:
  1. Wadah rekayasa fitur yang mengubah fitur asli menjadi fitur yang dapat dilatih oleh algoritma regresi.
  2. Wadah algoritme yang mengubah fitur dan menghasilkan skor regresi untuk kumpulan data.
- Klasifikasi: HPO menghasilkan tiga kontainer:
  1. Wadah rekayasa fitur yang mengubah fitur asli menjadi fitur yang dapat dilatih oleh algoritma klasifikasi.

2. Sebuah wadah algoritma yang menghasilkan `predicted_label` dengan probabilitas tertinggi. Wadah ini juga dapat menghasilkan berbagai probabilitas yang terkait dengan hasil klasifikasi dalam respons inferensi.
3. Sebuah wadah rekayasa fitur yang melakukan pasca-pemrosesan prediksi algoritma. Misalnya, ia dapat melakukan transformasi terbalik pada label yang diprediksi dan mengubahnya ke label asli.

Definisi wadah untuk mode ansambel

Dalam mode ansambel, tipe masalah regresi dan klasifikasi hanya memiliki satu wadah inferensi. Wadah inferensi ini mengubah fitur dan menghasilkan prediksi berdasarkan jenis masalah.

Respons inferensi per jenis masalah

Respons inferensi untuk model klasifikasi

Untuk wadah inferensi klasifikasi, Anda dapat memilih konten respons inferensi dengan menggunakan empat kunci yang telah ditentukan:

- `predicted_label`: Label dengan probabilitas tertinggi untuk memprediksi label yang benar, sebagaimana ditentukan oleh Autopilot.
- `probability`:
  - Model HPO: Probabilitas `True` kelas untuk klasifikasi biner. Probabilitas `predicted_label` untuk klasifikasi multiclass.
  - Model ansambel: Probabilitas klasifikasi biner dan multikelas. `predicted_label`
- `probabilities`: Daftar probabilitas untuk semua kelas yang sesuai.
- `labels`: Daftar semua label.

Misalnya, untuk masalah klasifikasi biner, jika Anda melewatkan kunci respons inferensi `['predicted_label', 'probability', 'probabilities', 'labels']` dan respons keluaran muncul sebagai`[1, 0.1, "[0.9, 0.1]", "['1', '0']"]`, Anda harus menafsirkannya sebagai berikut:

1. `predicted_label`sama 1 karena label “1" memiliki probabilitas yang lebih tinggi (`0.9`dalam hal ini).
2. Untuk model HPO, `probability` sama dengan `0.1` probabilitas `positive_class` (`0`dalam hal ini) yang dipilih oleh Autopilot.

Untuk model Ensemble, `probability` sama dengan `0.9` yang merupakan probabilitas dari. `predicted_label`

3. `probabilities`daftar `probability` setiap label di`labels`.
4. `labels`adalah label unik dalam kumpulan data, di mana label kedua (“0" dalam kasus ini) `positive_class` dipilih oleh Autopilot.

Secara default, kontainer inferensi dikonfigurasi untuk menghasilkan hanya file. `predicted_label` Untuk memilih konten inferensi tambahan, Anda dapat memperbarui `inference_response_keys` parameter untuk menyertakan hingga tiga variabel lingkungan ini:

- `SAGEMAKER_INFERENCE_SUPPORTED`: Ini diatur untuk memberikan petunjuk kepada Anda tentang konten apa yang didukung setiap wadah.
- `SAGEMAKER_INFERENCE_INPUT`: Ini harus diatur ke kunci yang diharapkan kontainer dalam muatan input.
- `SAGEMAKER_INFERENCE_OUTPUT`: Ini harus diisi dengan set kunci yang dikeluarkan kontainer.

Respons inferensi untuk model klasifikasi dalam mode HPO

Bagian ini menunjukkan cara mengonfigurasi respons inferensi dari model klasifikasi menggunakan mode optimasi hyperparameter (HPO).

Untuk memilih konten respons inferensi dalam mode HPO: Tambahkan `SAGEMAKER_INFERENCE_OUTPUT` variabel `SAGEMAKER_INFERENCE_INPUT` dan ke wadah kedua dan ketiga yang dihasilkan dalam mode HPO untuk masalah klasifikasi.

Kunci yang didukung oleh wadah kedua (algoritma) adalah predicted_label, probabilitas, dan probabilitas. Perhatikan `labels` bahwa sengaja tidak ditambahkan ke`SAGEMAKER_INFERENCE_SUPPORTED`.

Kunci yang didukung oleh wadah model klasifikasi ketiga adalah`predicted_label`,`labels`,`probability`, dan`probabilities`. Oleh karena itu, `SAGEMAKER_INFERENCE_SUPPORTED` lingkungan menyertakan nama-nama kunci ini.

Untuk memperbarui definisi wadah inferensi untuk menerima `predicted_label` dan`probability`, gunakan contoh kode berikut.

```
containers[1]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label,
 probability'})
containers[2]['Environment'].update({'SAGEMAKER_INFERENCE_INPUT': 'predicted_label,
 probability'})
containers[2]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label,
 probability'})
```

Contoh kode berikut memperbarui definisi wadah inferensi untuk menerima`predicted_label`,`probabilities`, dan`labels`. Jangan meneruskan `labels` ke wadah kedua (wadah algoritma), karena dihasilkan oleh wadah ketiga secara independen.

```
containers[1]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT':
 'predicted_label,probabilities'})
containers[2]['Environment'].update({'SAGEMAKER_INFERENCE_INPUT':
 'predicted_label,probabilities'})
containers[2]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label,
 probabilities,labels'})
```

Bagian yang dapat dilipat berikut memberikan contoh kode untuk AWS SDK for Python (Boto3) dan untuk SageMaker SDK untuk Python. Setiap bagian menunjukkan cara memilih konten tanggapan inferensi dalam mode HPO untuk contoh kode masing-masing.

AWS SDK for Python (Boto3)

```
import boto3

sm_client = boto3.client('sagemaker', region_name='<Region>')

role = '<IAM role>'
input_data = '<S3 input uri>'
output_path = '<S3 output uri>'

best_candidate = sm_client.describe_auto_ml_job(AutoMLJobName='<AutoML Job Name>')
['BestCandidate']
best_candidate_containers = best_candidate['InferenceContainers']
best_candidate_name = best_candidate['CandidateName']

best_candidate_containers[1]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT':
 'predicted_label, probability'})
best_candidate_containers[2]['Environment'].update({'SAGEMAKER_INFERENCE_INPUT':
 'predicted_label, probability'})
```

```
best_candidate_containers[2]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT':
 'predicted_label, probability'})

# create model
reponse = sm_client.create_model(
    ModelName = '<Model Name>',
    ExecutionRoleArn = role,
    Containers = best_candidate_containers
)

# Lauch Transform Job
response = sm_client.create_transform_job(
    TransformJobName='<Transform Job Name>',
    ModelName='<Model Name>',
    TransformInput={
        'DataSource': {
            'S3DataSource': {
                'S3DataType': 'S3Prefix',
                'S3Uri': input_data
            }
        },
        'ContentType': "text/CSV",
        'SplitType': 'Line'
    },
    TransformOutput={
        'S3OutputPath': output_path,
        'AssembleWith': 'Line',
    },
    TransformResources={
        'InstanceType': 'ml.m4.xlarge',
        'InstanceCount': 1,
    },
)
```

SageMaker SDK untuk Python

```
from sagemaker import AutoML

aml = AutoML.attach(auto_ml_job_name='<AutoML Job Name>')
aml_best_model = aml.create_model(name='<Model Name>',
                                  candidate=None,
                                  inference_response_keys**=['probabilities',
 'labels'])
```

```
aml_transformer = aml_best_model.transformer(accept='text/csv',
                                             assemble_with='Line',
                                             instance_type='ml.m5.xlarge',
                                             instance_count=1,)

aml_transformer.transform('<S3 input uri>',
                          content_type='text/csv',
                          split_type='Line',
                          job_name='<Transform Job Name>',
                          wait=True)
```

Respons inferensi untuk model klasifikasi dalam mode ansambel

Bagian ini menunjukkan cara mengonfigurasi respons inferensi dari model klasifikasi menggunakan mode ansambel.

Dalam mode ansambel, untuk memilih konten respons inferensi, perbarui variabel lingkungan. `SAGEMAKER_INFERENCE_OUTPUT`

Kunci yang didukung oleh wadah model klasifikasi adalah`predicted_label`,`labels`,`probability`, dan`probabilities`. Kunci-kunci ini termasuk dalam `SAGEMAKER_INFERENCE_SUPPORTED` lingkungan.

Untuk memperbarui definisi kontainer inferensi untuk menerima `predicted_label` dan`probability`, lihat contoh kode berikut.

```
containers[0]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT': 'predicted_label,
 probability'})
```

Bagian yang dapat dilipat berikut memberikan contoh kode untuk memilih konten respons inferensi dalam mode ansambel. Contoh menggunakan AWS SDK for Python (Boto3).

AWS SDK for Python (Boto3)

```
import boto3
sm_client = boto3.client('sagemaker', region_name='<Region>')

role = '<IAM role>'
input_data = '<S3 input uri>'
output_path = '<S3 output uri>'
```

```
best_candidate = sm_client.describe_auto_ml_job(AutoMLJobName='<AutoML Job Name>')
['BestCandidate']
best_candidate_containers = best_candidate['InferenceContainers']
best_candidate_name = best_candidate['CandidateName']

*best_candidate_containers[0]['Environment'].update({'SAGEMAKER_INFERENCE_OUTPUT':
 'predicted_label, probability'})
*
# create model
reponse = sm_client.create_model(
    ModelName = '<Model Name>',
    ExecutionRoleArn = role,
    Containers = best_candidate_containers
)

# Lauch Transform Job
response = sm_client.create_transform_job(
    TransformJobName='<Transform Job Name>',
    ModelName='<Model Name>',
    TransformInput={
        'DataSource': {
            'S3DataSource': {
                'S3DataType': 'S3Prefix',
                'S3Uri': input_data
            }
        },
        'ContentType': "text/CSV",
        'SplitType': 'Line'
    },
    TransformOutput={
        'S3OutputPath': output_path,
        'AssembleWith': 'Line',
    },
    TransformResources={
        'InstanceType': 'ml.m4.xlarge',
        'InstanceCount': 1,
    },
)
```

Bagian collapsible berikut memberikan contoh kode yang identik dengan SageMaker SDK untuk contoh Python untuk HPO. Ini termasuk untuk kenyamanan Anda.

SageMaker SDK untuk Python

Contoh kode HPO berikut menggunakan SageMaker SDK untuk Python.

```
from sagemaker import AutoML

aml = AutoML.attach(auto_ml_job_name='<AutoML Job Name>')
aml_best_model = aml.create_model(name='<Model Name>',
                                  candidate=None,
                                  *inference_response_keys**=['probabilities',
 'labels'])*

aml_transformer = aml_best_model.transformer(accept='text/csv',
                                             assemble_with='Line',
                                             instance_type='ml.m5.xlarge',
                                             instance_count=1,)

aml_transformer.transform('<S3 input uri>',
                          content_type='text/csv',
                          split_type='Line',
                          job_name='<Transform Job Name>',
                          wait=True)
```

Tutorial dan contoh notebook

Contoh notebook, video tutorial, dan panduan untuk memulai dengan Amazon Autopilot. SageMaker

Topik

- Contoh notebook: Jelajahi pemodelan dengan Amazon SageMaker Autopilot
- Video: Gunakan Autopilot untuk mengotomatisasi dan menjelajahi proses pembelajaran mesin
- Tutorial: Memulai dengan Amazon SageMaker Autopilot

Contoh notebook: Jelajahi pemodelan dengan Amazon SageMaker Autopilot

Amazon SageMaker Autopilot menyediakan contoh notebook berikut.

- Pemasaran langsung dengan Amazon SageMaker Autopilot: Notebook ini menunjukkan bagaimana menggunakan Kumpulan Data Pemasaran Bank untuk memprediksi apakah pelanggan akan mendaftar untuk deposito berjangka di bank. Anda dapat menggunakan Autopilot pada kumpulan data ini untuk mendapatkan pipeline ML paling akurat dengan menjelajahi opsi yang

terdapat dalam berbagai jalur pipa kandidat. Autopilot menghasilkan setiap kandidat dalam prosedur dua langkah. Langkah pertama melakukan rekayasa fitur otomatis pada kumpulan data. Langkah kedua melatih dan menyetel algoritma untuk menghasilkan model. Notebook berisi instruksi tentang cara melatih model dan cara menerapkan model untuk melakukan inferensi batch menggunakan kandidat terbaik.

- Prediksi Churn Pelanggan dengan Amazon SageMaker Autopilot: Notebook ini menjelaskan penggunaan pembelajaran mesin untuk identifikasi otomatis pelanggan yang tidak bahagia, juga dikenal sebagai prediksi churn pelanggan. Contoh ini menunjukkan bagaimana menganalisis kumpulan data yang tersedia untuk umum dan melakukan rekayasa fitur di atasnya. Selanjutnya ini menunjukkan cara menyetel model dengan memilih pipeline berkinerja terbaik bersama dengan hyperparameters optimal untuk algoritma pelatihan. Akhirnya, ini menunjukkan bagaimana menerapkan model ke titik akhir yang dihosting dan bagaimana mengevaluasi prediksinya terhadap kebenaran dasar. Namun, model ML jarang memberikan prediksi yang sempurna. Itu sebabnya notebook ini juga menunjukkan bagaimana menggabungkan biaya relatif kesalahan prediksi saat menentukan hasil keuangan menggunakan ML.
- Prediksi Churn Pelanggan Kandidat Teratas dengan Amazon SageMaker Autopilot dan Batch Transform (Python SDK): Notebook ini juga menjelaskan penggunaan pembelajaran mesin untuk identifikasi otomatis pelanggan yang tidak puas, juga dikenal sebagai prediksi churn pelanggan. Notebook ini mendemonstrasikan cara mengonfigurasi model untuk mendapatkan probabilitas inferensi, memilih model N teratas, dan membuat Batch Transform pada set pengujian penahanan untuk evaluasi.

  > **Note**
  >
  > Notebook ini bekerja dengan SageMaker Python SDK >= 1.65.1 dirilis pada 6/19/2020.

- Membawa kode pemrosesan data Anda sendiri ke Amazon SageMaker Autopilot: Notebook ini menunjukkan cara menggabungkan dan menerapkan kode pemrosesan data khusus saat menggunakan Amazon Autopilot. SageMaker Ini menambahkan langkah pemilihan fitur khusus untuk menghapus variabel yang tidak relevan ke pekerjaan Autopilot. Ini kemudian menunjukkan bagaimana menerapkan kode pemrosesan khusus dan model yang dihasilkan oleh Autopilot pada titik akhir waktu nyata dan, sebagai alternatif, untuk pemrosesan batch.

Video: Gunakan Autopilot untuk mengotomatisasi dan menjelajahi proses pembelajaran mesin

Berikut adalah seri video yang menyediakan tur kemampuan Amazon SageMaker Autopilot menggunakan Studio Classic. Mereka menunjukkan bagaimana memulai pekerjaan AutoML,

menganalisis dan memproses data sebelumnya, bagaimana melakukan rekayasa fitur dan optimasi hyperparameter pada model kandidat, dan bagaimana memvisualisasikan dan membandingkan metrik model yang dihasilkan.

Topik

- Memulai pekerjaan AutoML dengan Amazon Autopilot SageMaker
- Tinjau eksplorasi data dan rekayasa fitur otomatis di Autopilot.
- Tune model untuk mengoptimalkan kinerja
- Pilih dan gunakan model terbaik
- Amazon SageMaker Autopilot tutorial

Memulai pekerjaan AutoML dengan Amazon Autopilot SageMaker

Video ini menunjukkan kepada Anda cara memulai pekerjaan AutoML dengan Autopilot. (Panjangnya: 8:41)

Amazon SageMaker Studio - AutoML dengan Amazon SageMaker Autopilot (bagian 1)

Tinjau eksplorasi data dan rekayasa fitur otomatis di Autopilot.

Video ini menunjukkan kepada Anda cara meninjau eksplorasi data dan notebook definisi kandidat yang dihasilkan oleh Amazon SageMaker Autopilot. (Panjangnya: 10:04)

Amazon SageMaker Studio - AutoML dengan Amazon SageMaker Autopilot (bagian 2)

Tune model untuk mengoptimalkan kinerja

Video ini menunjukkan kepada Anda cara mengoptimalkan kinerja model selama pelatihan menggunakan penyetelan hyperparameter. (Panjangnya: 4:59)

SageMaker Studio - AutoML dengan Amazon SageMaker Autopilot (bagian 3)

Pilih dan gunakan model terbaik

Video ini menunjukkan kepada Anda cara menggunakan metrik pekerjaan untuk memilih model terbaik dan kemudian cara menerapkannya. (Panjangnya: 5:20)

SageMaker Studio - AutoML dengan Amazon SageMaker Autopilot (bagian 4)

Amazon SageMaker Autopilot tutorial

Video ini memandu Anda melalui demo ujung ke ujung di mana kami pertama kali membangun model klasifikasi biner secara otomatis dengan Amazon SageMaker Autopilot. Kami melihat bagaimana model kandidat telah dibangun dan dioptimalkan menggunakan notebook yang dibuat secara otomatis. Kami juga melihat kandidat teratas dengan SageMaker Eksperimen Amazon. Akhirnya, kami menerapkan kandidat teratas (berdasarkan XGBoost), dan mengonfigurasi pengambilan data dengan SageMaker Model Monitor.

Demo ujung ke ujung dengan AutoML aktif SageMaker

Tutorial: Memulai dengan Amazon SageMaker Autopilot

Memulai tutorial untuk Autopilot menunjukkan cara membuat model pembelajaran mesin secara otomatis tanpa menulis kode. Mereka menunjukkan kepada Anda bagaimana Autopilot menyederhanakan pengalaman pembelajaran mesin dengan membantu Anda menjelajahi data Anda dan mencoba algoritme yang berbeda. Autopilot membangun model pembelajaran mesin terbaik untuk jenis masalah menggunakan kemampuan AutoML sambil memungkinkan kontrol dan visibilitas penuh.

- Buat model pembelajaran mesin secara otomatis dengan Autopilot: Anda mengambil peran pengembang yang bekerja di bank dalam tutorial ini. Anda telah diminta untuk mengembangkan model pembelajaran mesin untuk memprediksi apakah pelanggan akan mendaftar untuk sertifikat setoran (CD). Ini adalah masalah klasifikasi biner. Model ini dilatih pada dataset pemasaran yang berisi informasi tentang demografi pelanggan, respons terhadap peristiwa pemasaran, dan faktor eksternal.

Membuat pekerjaan AutoML untuk klasifikasi gambar menggunakan API

Petunjuk berikut menunjukkan cara membuat pekerjaan Amazon SageMaker Autopilot sebagai percobaan percontohan untuk jenis masalah klasifikasi gambar menggunakan SageMaker Referensi API.

> **Note**
>
> Tugas seperti klasifikasi teks dan gambar, peramalan deret waktu, dan penyempurnaan model bahasa besar tersedia secara eksklusif melalui API Autopilot versi 2. Untuk pengguna Python, sebaiknya gunakan AWS SDK for Python (Boto3)sebagai Amazon SageMaker Python SDK saat ini tidak didukung untuk Autopilot API versi 2.

> Pengguna yang lebih menyukai kenyamanan antarmuka pengguna dapat menggunakan Amazon SageMaker Canvas untuk mengakses model pra-terlatih dan model dasar AI generatif, atau membuat model khusus yang disesuaikan untuk teks tertentu, klasifikasi gambar, kebutuhan peramalan, atau AI generatif.

Anda dapat membuat eksperimen klasifikasi gambar Autopilot secara terprogram dengan memanggil tindakan `CreateAutoMLJobV2`API dalam bahasa apa pun yang didukung oleh Amazon Autopilot atau. SageMaker AWS CLI

Untuk informasi tentang cara tindakan API ini diterjemahkan ke dalam fungsi dalam bahasa pilihan Anda, lihat bagian Lihat Juga `CreateAutoMLJobV2` dan pilih SDK. Sebagai contoh, untuk pengguna Python, lihat sintaks permintaan lengkap dari in. `create_auto_ml_job_v2` AWS SDK for Python (Boto3)

Berikut ini adalah kumpulan parameter permintaan input wajib dan opsional untuk tindakan `CreateAutoMLJobV2` API yang digunakan dalam klasifikasi gambar.

Parameter yang diperlukan

Saat menelepon `CreateAutoMLJobV2` untuk membuat eksperimen Autopilot untuk klasifikasi gambar, Anda harus memberikan nilai berikut:

- An `AutoMLJobName` untuk menentukan nama pekerjaan Anda.
- Setidaknya satu `AutoMLJobChannel` `AutoMLJobInputDataConfig` untuk menentukan sumber data Anda.
- Sebuah `AutoMLProblemTypeConfig` tipe`ImageClassificationJobConfig`.
- `OutputDataConfig`Untuk menentukan jalur keluaran Amazon S3 untuk menyimpan artefak pekerjaan AutoML Anda.
- A `RoleArn` untuk menentukan ARN dari peran yang digunakan untuk mengakses data Anda.

Semua parameter lainnya adalah opsional.

Parameter opsional

Bagian berikut memberikan rincian beberapa parameter opsional yang dapat Anda berikan ke pekerjaan AutoML klasifikasi gambar Anda.

Cara menentukan kumpulan data pelatihan dan validasi pekerjaan AutoML

Anda dapat memberikan kumpulan data validasi dan rasio pemisahan data khusus Anda sendiri, atau membiarkan Autopilot membagi kumpulan data secara otomatis.

Setiap `AutoMLJobChannel` objek (lihat parameter AutoML yang diperlukanJobInputDataConfig) memilikiChannelType, yang dapat diatur ke salah satu `training` atau `validation` nilai yang menentukan bagaimana data akan digunakan saat membangun model pembelajaran mesin.

Setidaknya satu sumber data harus disediakan dan maksimal dua sumber data diperbolehkan: satu untuk data pelatihan dan satu untuk data validasi. Bagaimana Anda membagi data menjadi kumpulan data pelatihan dan validasi tergantung pada apakah Anda memiliki satu atau dua sumber data.

Bagaimana Anda membagi data menjadi kumpulan data pelatihan dan validasi tergantung pada apakah Anda memiliki satu atau dua sumber data.

- Jika Anda hanya memiliki satu sumber data, `ChannelType` diatur ke secara `training` default dan harus memiliki nilai ini.
  - Jika `ValidationFraction` nilai dalam tidak `AutoMLDataSplitConfig`disetel, 0.2 (20%) data dari sumber ini digunakan untuk validasi secara default.
  - Jika `ValidationFraction` diatur ke nilai antara 0 dan 1, dataset dibagi berdasarkan nilai yang ditentukan, di mana nilai menentukan fraksi dari dataset yang digunakan untuk validasi.
- Jika Anda memiliki dua sumber data, `ChannelType` salah satu `AutoMLJobChannel` objek harus diatur ke`training`, nilai default. Sumber data lainnya harus diatur ke`validation`. `ChannelType` Kedua sumber data harus memiliki format yang sama, baik CSV atau Parket, dan skema yang sama. Anda tidak boleh menetapkan nilai untuk `ValidationFraction` dalam kasus ini karena semua data dari setiap sumber digunakan untuk pelatihan atau validasi. Menyetel nilai ini menyebabkan kesalahan.

Cara menentukan konfigurasi penerapan model otomatis untuk pekerjaan AutoML

Untuk mengaktifkan penerapan otomatis untuk kandidat model terbaik dari pekerjaan AutoML, sertakan `ModelDeployConfig` a dalam permintaan pekerjaan AutoML. Ini akan memungkinkan penerapan model terbaik ke titik SageMaker akhir. Di bawah ini adalah konfigurasi yang tersedia untuk kustomisasi.

- Untuk membiarkan Autopilot menghasilkan nama titik akhir, setel ke. `AutoGenerateEndpointName` `True`

- Untuk memberikan nama Anda sendiri untuk titik akhir, aturAutoGenerateEndpointName to False and provide a name of your choice in EndpointName.

Format kumpulan data dan metrik objektif untuk klasifikasi gambar

Pada bagian ini kita belajar tentang format yang tersedia untuk kumpulan data yang digunakan dalam klasifikasi gambar serta metrik objektif yang digunakan untuk mengevaluasi kualitas prediktif kandidat model pembelajaran mesin. Metrik yang dihitung untuk kandidat ditentukan menggunakan array MetricDatumtipe.

Format kumpulan data

Autopilot mendukung format gambar.png, .jpg, dan .jpeg. Jika dataset Anda berisi semua gambar.png yang digunakanimage/png, jika berisi semua gambar.jpg atau .jpeg yang digunakanimage/jpeg, dan jika dataset Anda berisi campuran format gambar, gunakan. image/*

Metrik objektif

Daftar berikut berisi nama-nama metrik yang saat ini tersedia untuk mengukur kinerja model untuk klasifikasi gambar.

Accuracy

Rasio jumlah item yang diklasifikasikan dengan benar dengan jumlah total item yang diklasifikasikan (benar dan salah). Akurasi mengukur seberapa dekat nilai kelas yang diprediksi dengan nilai aktual. Nilai untuk metrik akurasi bervariasi antara nol (0) dan satu (1). Nilai 1 menunjukkan akurasi sempurna, dan 0 menunjukkan ketidakakuratan sempurna.

Penyebaran dan prediksi model autopilot

Panduan Autopilot ini mencakup langkah-langkah untuk penerapan model dan pengaturan inferensi waktu nyata.

Setelah Anda melatih model Autopilot Anda, Anda dapat mengatur titik akhir dan mendapatkan prediksi secara interaktif.

Inferensi waktu nyata

Inferensi real-time sangat ideal untuk beban kerja inferensi di mana Anda memiliki persyaratan real-time, interaktif, latensi rendah. Bagian ini menunjukkan bagaimana Anda dapat menggunakan inferensi waktu nyata untuk mendapatkan prediksi secara interaktif dari model Anda.

Anda dapat menggunakan SageMaker API untuk menerapkan model secara manual yang menghasilkan metrik validasi terbaik dalam eksperimen Autopilot sebagai berikut.

Atau, Anda dapat memilih opsi penerapan otomatis saat membuat eksperimen Autopilot Anda. Untuk informasi tentang pengaturan penerapan otomatis model, lihat `ModelDeployConfig` di parameter permintaan. `CreateAutoMLJobV2` Ini menciptakan titik akhir secara otomatis.

> **Note**
>
> Untuk menghindari biaya yang tidak perlu, Anda dapat menghapus titik akhir yang tidak diperlukan dan sumber daya yang dibuat dari penerapan model. Untuk informasi tentang penetapan harga instans menurut Wilayah, lihat SageMaker Harga Amazon.

1. Dapatkan definisi wadah kandidat

   Dapatkan definisi wadah kandidat dari InferenceContainers. Definisi kontainer untuk inferensi mengacu pada lingkungan kontainer yang dirancang untuk menerapkan dan menjalankan SageMaker model terlatih Anda untuk membuat prediksi.

   Contoh AWS CLI perintah berikut menggunakan API DescribeAutoMLJobv2 untuk mendapatkan definisi kandidat untuk kandidat model terbaik.

   ```
   aws sagemaker describe-auto-ml-job-v2 --auto-ml-job-name job-name --region region
   ```

2. Daftar kandidat

   Contoh AWS CLI perintah berikut menggunakan ListCandidatesForAutoMLJob API untuk daftar semua kandidat model.

   ```
   aws sagemaker list-candidates-for-auto-ml-job --auto-ml-job-name <job-name> --
   region <region>
   ```

3. Buat SageMaker model

Gunakan definisi kontainer dari langkah sebelumnya dan kandidat pilihan Anda untuk membuat SageMaker model dengan menggunakan CreateModelAPI. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker create-model --model-name '<your-candidate-name>' \
                    --containers ['<container-definition1>, <container-
definition2>, <container-definition3>]' \
                    --execution-role-arn '<execution-role-arn>' --region '<region>
```

4. Buat konfigurasi titik akhir

   Contoh AWS CLI perintah berikut menggunakan CreateEndpointConfigAPI untuk membuat konfigurasi endpoint.

```
aws sagemaker create-endpoint-config --endpoint-config-name '<your-endpoint-config-
name>' \
                    --production-variants '<list-of-production-variants>' \
                    --region '<region>'
```

5. Buat titik akhir

   AWS CLIContoh berikut menggunakan CreateEndpointAPI untuk membuat titik akhir.

```
aws sagemaker create-endpoint --endpoint-name '<your-endpoint-name>' \
                    --endpoint-config-name '<endpoint-config-name-you-just-created>'
 \
                    --region '<region>'
```

   Periksa kemajuan penerapan titik akhir Anda dengan menggunakan API. DescribeEndpoint Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-endpoint –endpoint-name '<endpoint-name>' –region <region>
```

   Setelah `EndpointStatus` perubahan`InService`, titik akhir siap digunakan untuk inferensi waktu nyata.

6. Memanggil titik akhir

   Struktur perintah berikut memanggil titik akhir untuk inferensi real-time.

```
aws sagemaker invoke-endpoint --endpoint-name '<endpoint-name>' \
                  --region '<region>' --body '<your-data>' [--content-type]
 '<content-type>' <outfile>
```

Laporan penjelasan

Amazon SageMaker Autopilot menyediakan laporan penjelasan untuk membantu menjelaskan bagaimana kandidat model terbaik membuat prediksi untuk masalah klasifikasi gambar. Laporan ini dapat membantu insinyur, manajer produk, dan pemangku kepentingan internal lainnya dalam memahami karakteristik model. Baik konsumen maupun regulator mengandalkan transparansi dalam pembelajaran mesin untuk mempercayai dan menafsirkan keputusan yang dibuat berdasarkan prediksi model. Anda dapat menggunakan penjelasan ini untuk mengaudit dan memenuhi persyaratan peraturan, membangun kepercayaan pada model, mendukung pengambilan keputusan manusia, dan men-debug dan meningkatkan kinerja model.

Fungsionalitas penjelasan Autopilot untuk klasifikasi gambar menggunakan pendekatan peta aktivasi kelas visual (CAM) yang menghasilkan peta panas di mana distribusi dan intensitas setiap warna menyoroti area gambar yang paling berkontribusi pada prediksi tertentu. Pendekatan ini bergantung pada komponen utama yang berasal dari implementasi EIGEN-CAM.

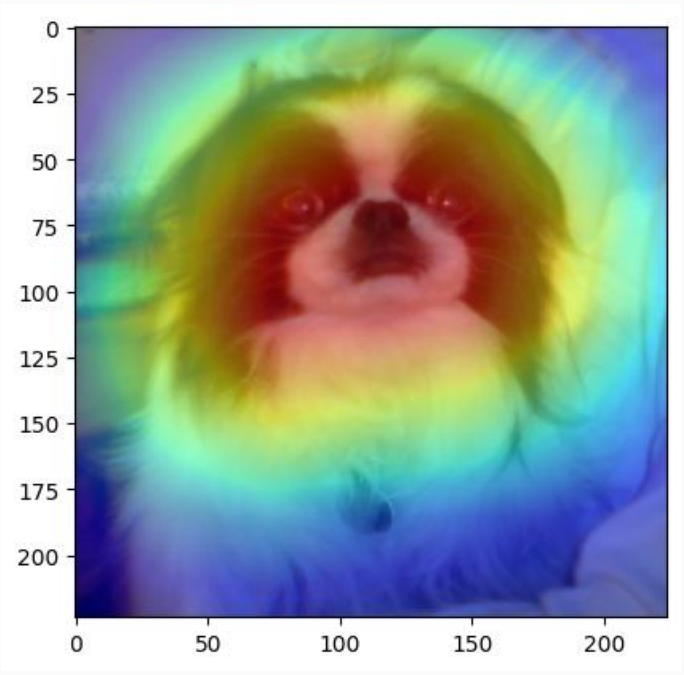
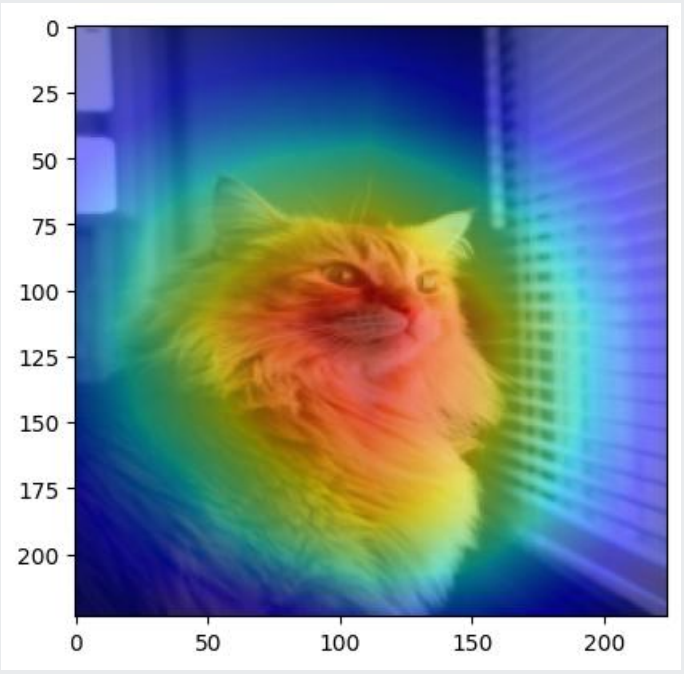
Autopilot menghasilkan laporan penjelasan sebagai file JSON. Laporan tersebut mencakup rincian analisis yang didasarkan pada dataset validasi. Setiap gambar yang digunakan untuk menghasilkan laporan berisi informasi berikut:

- `input_image_uri`: URI Amazon S3 ke gambar input yang diambil sebagai input untuk peta panas.
- `heatmap_image_uri`: URI Amazon S3 ke gambar peta panas yang dihasilkan oleh Autopilot.
- `predicted_label`: Kelas label diprediksi oleh model terbaik yang dilatih oleh Autopilot.
- `probability`: Keyakinan yang `predicted_label` diprediksi.

Anda dapat menemukan awalan Amazon S3 untuk artefak penjelasan yang dihasilkan untuk kandidat terbaik dalam menanggapi at. `DescribeAutoMLJobV2` `BestCandidate.CandidateProperties.CandidateArtifactLocations.Explainability`

Contoh berikut menggambarkan seperti apa peta panas pada beberapa sampel dari Oxford-IIit Pet Dataset. Gambar peta panas menampilkan gradien warna yang menunjukkan kepentingan relatif dari

berbagai fitur dalam gambar. Warna merah mewakili daerah dengan kepentingan yang lebih besar dalam memprediksi “predicted_label” dari gambar input dibandingkan dengan fitur yang diwakili oleh warna biru.

Input Gambar	Gambar Heatmap
	
	

Laporan kinerja model

Laporan kualitas SageMaker model Amazon (juga disebut sebagai laporan kinerja) memberikan wawasan dan informasi kualitas untuk kandidat model terbaik yang dihasilkan oleh pekerjaan AutoML. Ini termasuk informasi tentang detail pekerjaan, jenis masalah model, fungsi objektif, dan berbagai metrik. Bagian ini merinci konten laporan kinerja untuk masalah klasifikasi gambar dan menjelaskan cara mengakses metrik sebagai data mentah dalam file JSON.

Anda dapat menemukan awalan Amazon S3 untuk artefak laporan kualitas model yang dihasilkan untuk kandidat terbaik dalam menanggapi at. `DescribeAutoMLJobV2` `BestCandidate.CandidateProperties.CandidateArtifactLocations.ModelInsights`

Laporan kinerja berisi dua bagian:

- Bagian pertama berisi rincian tentang pekerjaan Autopilot yang menghasilkan model.
- Bagian kedua berisi laporan kualitas model dengan berbagai metrik kinerja.

Detail pekerjaan Autopilot

Bagian pertama dari laporan ini memberikan beberapa informasi umum tentang pekerjaan Autopilot yang menghasilkan model. Rincian ini mencakup informasi berikut:

- Nama kandidat autopilot: Nama kandidat model terbaik.
- Nama pekerjaan autopilot: Nama pekerjaan.
- Jenis masalah: Jenis masalah. Dalam kasus kami, klasifikasi gambar.
- Metrik objektif: Metrik objektif yang digunakan untuk mengoptimalkan kinerja model. Dalam kasus kami, Akurasi.
- Arah optimasi: Menunjukkan apakah akan meminimalkan atau memaksimalkan metrik objektif.

Laporan kualitas model

Informasi kualitas model dihasilkan oleh wawasan model Autopilot. Konten laporan yang dihasilkan bergantung pada jenis masalah yang ditangani. Laporan tersebut menentukan jumlah baris yang termasuk dalam dataset evaluasi dan waktu evaluasi terjadi.

Tabel metrik

Bagian pertama dari laporan kualitas model berisi tabel metrik. Ini sesuai untuk jenis masalah yang ditangani model.

Gambar berikut adalah contoh tabel metrik yang dihasilkan oleh Autopilot untuk masalah klasifikasi gambar atau teks. Ini menunjukkan nama metrik, nilai, dan standar deviasi.

Metrics table

Metric Name	Value	Standard Deviation
weighted_recall	0.597104	0.005410
weighted_precision	0.591693	0.005729
accuracy	0.597104	0.005410
weighted_f0_5	0.592155	0.005659
weighted_f1	0.593423	0.005554
weighted_f2	0.595392	0.005456
accuracy_best_constant_classifier	0.200699	0.004422
weighted_recall_best_constant_classifier	0.200699	0.004422
weighted_precision_best_constant_classifier	0.040280	0.001753
weighted_f0_5_best_constant_classifier	0.047944	0.002039
weighted_f1_best_constant_classifier	0.067094	0.002684
weighted_f2_best_constant_classifier	0.111716	0.003808

Informasi kinerja model grafis

Bagian kedua dari laporan kualitas model berisi informasi grafis untuk membantu Anda mengevaluasi kinerja model. Isi bagian ini tergantung pada jenis masalah yang dipilih.

Matriks kebingungan

Matriks kebingungan menyediakan cara untuk memvisualisasikan keakuratan prediksi yang dibuat oleh model untuk klasifikasi biner dan multikelas untuk masalah yang berbeda.

Ringkasan komponen grafik tingkat positif palsu (FPR) dan tingkat positif sejati (TPR) didefinisikan sebagai berikut.

- Prediksi yang benar
  - True positive (TP): Nilai yang diprediksi adalah 1, dan nilai sebenarnya adalah 1.
  - Benar negatif (TN): Nilai yang diprediksi adalah 0, dan nilai sebenarnya adalah 0.
- Prediksi yang salah
  - Positif palsu (FP): Nilai yang diprediksi adalah 1, tetapi nilai sebenarnya adalah 0.
  - False negative (FN): Nilai yang diprediksi adalah 0, tetapi nilai sebenarnya adalah 1.

Matriks kebingungan dalam laporan kualitas model berisi yang berikut ini.

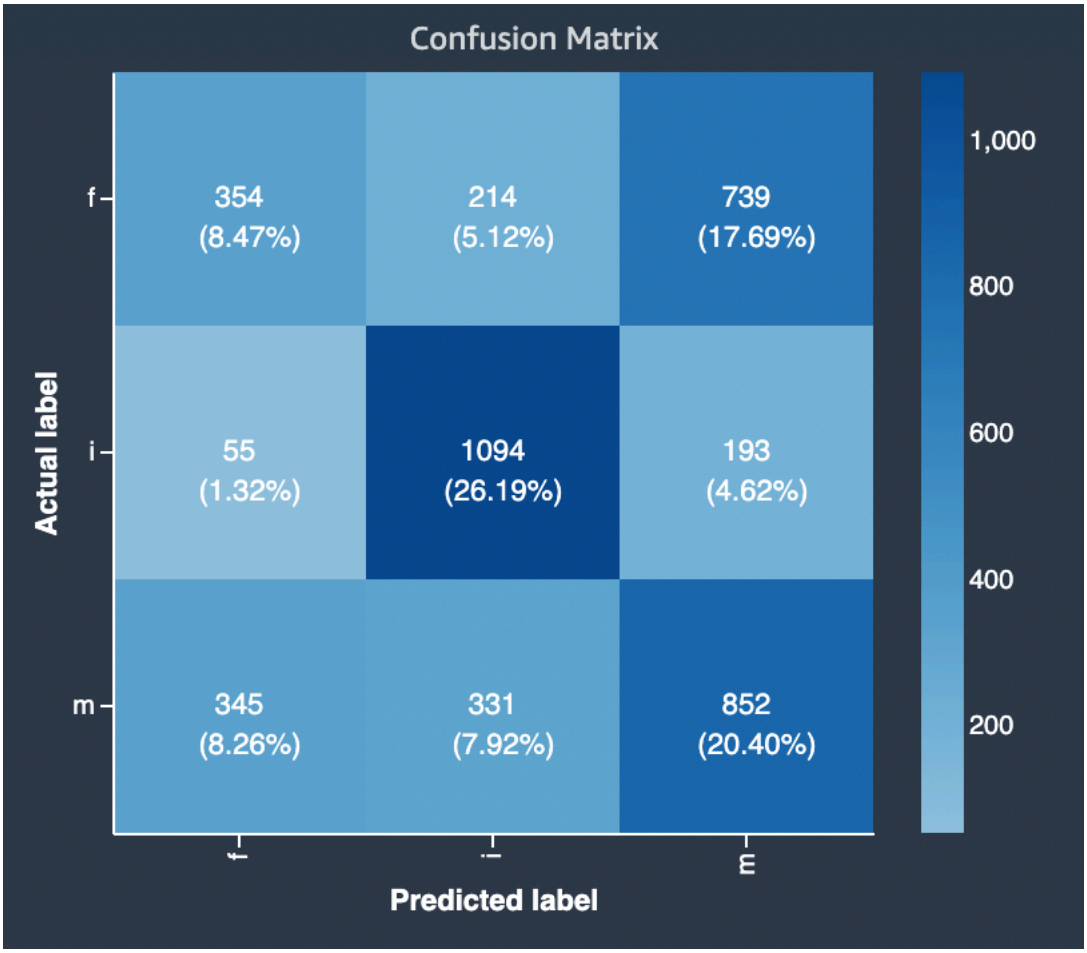
- Jumlah dan persentase prediksi yang benar dan salah untuk label yang sebenarnya
- Jumlah dan persentase prediksi akurat pada diagonal dari kiri atas ke pojok kanan bawah
- Jumlah dan persentase prediksi yang tidak akurat pada diagonal dari kanan atas ke sudut kiri bawah

Prediksi yang salah pada matriks kebingungan adalah nilai kebingungan.

Diagram berikut adalah contoh matriks kebingungan untuk masalah klasifikasi multi-kelas. Matriks kebingungan dalam laporan kualitas model berisi yang berikut ini.

- Sumbu vertikal dibagi menjadi tiga baris yang berisi tiga label aktual yang berbeda.
- Sumbu horizontal dibagi menjadi tiga kolom yang berisi label yang diprediksi oleh model.
- Bilah warna memberikan nada yang lebih gelap ke sejumlah besar sampel untuk secara visual menunjukkan jumlah nilai yang diklasifikasikan dalam setiap kategori.

Dalam contoh di bawah ini, model dengan benar memprediksi 354 nilai aktual untuk label f, 1094 nilai untuk label i dan 852 nilai untuk label m. Perbedaan nada menunjukkan bahwa kumpulan data tidak seimbang karena ada lebih banyak label untuk nilai i daripada untuk f atau m.



Matriks kebingungan dalam laporan kualitas model yang disediakan dapat mengakomodasi maksimum 15 label untuk jenis masalah klasifikasi multikelas. Jika baris yang sesuai dengan label menunjukkan Nan nilai, itu berarti kumpulan data validasi yang digunakan untuk memeriksa prediksi model tidak berisi data dengan label tersebut.

Membuat pekerjaan AutoML untuk klasifikasi teks menggunakan API

Petunjuk berikut menunjukkan cara membuat pekerjaan Amazon SageMaker Autopilot sebagai percobaan percobaan untuk jenis masalah klasifikasi teks menggunakan SageMaker Referensi API.

> **Note**
>
> Tugas seperti klasifikasi teks dan gambar, peramalan deret waktu, dan penyempurnaan model bahasa besar tersedia secara eksklusif melalui API Autopilot versi 2. Untuk pengguna Python, sebaiknya gunakan AWS SDK for Python (Boto3)sebagai Amazon SageMaker Python SDK saat ini tidak didukung untuk Autopilot API versi 2.

> Pengguna yang lebih menyukai kenyamanan antarmuka pengguna dapat menggunakan Amazon SageMaker Canvas untuk mengakses model pra-terlatih dan model dasar AI generatif, atau membuat model khusus yang disesuaikan untuk teks tertentu, klasifikasi gambar, kebutuhan peramalan, atau AI generatif.

Anda dapat membuat eksperimen klasifikasi teks Autopilot secara terprogram dengan memanggil tindakan `CreateAutoMLJobV2`API dalam bahasa apa pun yang didukung oleh Amazon Autopilot atau. SageMaker AWS CLI

Untuk informasi tentang cara tindakan API ini diterjemahkan ke dalam fungsi dalam bahasa pilihan Anda, lihat bagian Lihat Juga `CreateAutoMLJobV2` dan pilih SDK. Sebagai contoh, untuk pengguna Python, lihat sintaks permintaan lengkap dari in. `create_auto_ml_job_v2` AWS SDK for Python (Boto3)

Berikut ini adalah kumpulan parameter permintaan input wajib dan opsional untuk tindakan `CreateAutoMLJobV2` API yang digunakan dalam klasifikasi teks.

Parameter yang diperlukan

Saat menelepon `CreateAutoMLJobV2` untuk membuat eksperimen Autopilot untuk klasifikasi teks, Anda harus memberikan nilai berikut:

- An `AutoMLJobName` untuk menentukan nama pekerjaan Anda.
- Setidaknya satu `AutoMLJobChannel` `AutoMLJobInputDataConfig` untuk menentukan sumber data Anda.
- Sebuah `AutoMLProblemTypeConfig` tipe`TextClassificationJobConfig`.
- `OutputDataConfig`Untuk menentukan jalur keluaran Amazon S3 untuk menyimpan artefak pekerjaan AutoML Anda.
- A `RoleArn` untuk menentukan ARN dari peran yang digunakan untuk mengakses data Anda.

Semua parameter lainnya adalah opsional.

Parameter opsional

Bagian berikut memberikan rincian beberapa parameter opsional yang dapat Anda teruskan ke tugas AutoML klasifikasi teks Anda.

Cara menentukan kumpulan data pelatihan dan validasi pekerjaan AutoML

Anda dapat memberikan kumpulan data validasi dan rasio pemisahan data khusus Anda sendiri, atau membiarkan Autopilot membagi kumpulan data secara otomatis.

Setiap `AutoMLJobChannel`objek (lihat parameter AutoML yang diperlukanJobInputDataConfig) memilikiChannelType, yang dapat diatur ke salah satu `training` atau `validation` nilai yang menentukan bagaimana data akan digunakan saat membangun model pembelajaran mesin.

Setidaknya satu sumber data harus disediakan dan maksimal dua sumber data diperbolehkan: satu untuk data pelatihan dan satu untuk data validasi. Bagaimana Anda membagi data menjadi kumpulan data pelatihan dan validasi tergantung pada apakah Anda memiliki satu atau dua sumber data.

Bagaimana Anda membagi data menjadi kumpulan data pelatihan dan validasi tergantung pada apakah Anda memiliki satu atau dua sumber data.

- Jika Anda hanya memiliki satu sumber data, `ChannelType` diatur ke secara `training` default dan harus memiliki nilai ini.
  - Jika `ValidationFraction` nilai dalam tidak `AutoMLDataSplitConfig`disetel, 0.2 (20%) data dari sumber ini digunakan untuk validasi secara default.
  - Jika `ValidationFraction` diatur ke nilai antara 0 dan 1, dataset dibagi berdasarkan nilai yang ditentukan, di mana nilai menentukan fraksi dari dataset yang digunakan untuk validasi.
- Jika Anda memiliki dua sumber data, `ChannelType` salah satu `AutoMLJobChannel` objek harus diatur ke`training`, nilai default. Sumber data lainnya harus diatur ke`validation`. `ChannelType` Kedua sumber data harus memiliki format yang sama, baik CSV atau Parket, dan skema yang sama. Anda tidak boleh menetapkan nilai untuk `ValidationFraction` dalam kasus ini karena semua data dari setiap sumber digunakan untuk pelatihan atau validasi. Menyetel nilai ini menyebabkan kesalahan.

Cara menentukan konfigurasi penerapan model otomatis untuk pekerjaan AutoML

Untuk mengaktifkan penerapan otomatis untuk kandidat model terbaik dari pekerjaan AutoML, sertakan `ModelDeployConfig` a dalam permintaan pekerjaan AutoML. Ini akan memungkinkan penerapan model terbaik ke titik SageMaker akhir. Di bawah ini adalah konfigurasi yang tersedia untuk kustomisasi.

- Untuk membiarkan Autopilotmenghasilkan nama titik akhir, setel ke. `AutoGenerateEndpointName` `True`

- Untuk memberikan nama Anda sendiri untuk titik akhir, atur`AutoGenerateEndpointName` `to False and provide a name of your choice in` `EndpointName`.

Format kumpulan data dan metrik objektif untuk klasifikasi teks

Pada bagian ini kita belajar tentang format yang tersedia untuk kumpulan data yang digunakan dalam klasifikasi teks serta metrik yang digunakan untuk mengevaluasi kualitas prediktif kandidat model pembelajaran mesin. Metrik yang dihitung untuk kandidat ditentukan menggunakan array MetricDatumtipe.

Format kumpulan data

Autopilot mendukung data tabular yang diformat sebagai file CSV atau sebagai file Parket. Untuk data tabular, setiap kolom berisi fitur dengan tipe data tertentu dan setiap baris berisi pengamatan. Properti dari dua format file ini sangat berbeda.

- CSV (comma-separated-values) adalah format file berbasis baris yang menyimpan data dalam teks biasa yang dapat dibaca manusia yang merupakan pilihan populer untuk pertukaran data karena didukung oleh berbagai aplikasi.
- Parket adalah format file berbasis kolom di mana data disimpan dan diproses lebih efisien daripada format file berbasis baris. Ini membuat mereka menjadi pilihan yang lebih baik untuk masalah data besar.

Tipe data yang diterima untuk kolom termasuk numerik, kategoris, teks.

Autopilot mendukung pembuatan model pembelajaran mesin pada kumpulan data besar hingga ratusan GB. Untuk detail tentang batas sumber daya default untuk kumpulan data input dan cara meningkatkannya, lihat kuota Amazon SageMaker Autopilot.

Metrik obyektif

Daftar berikut berisi nama-nama metrik yang saat ini tersedia untuk mengukur kinerja model untuk klasifikasi teks.

`Accuracy`

Rasio jumlah item yang diklasifikasikan dengan benar dengan jumlah total item yang diklasifikasikan (benar dan salah). Akurasi mengukur seberapa dekat nilai kelas yang diprediksi

dengan nilai aktual. Nilai untuk metrik akurasi bervariasi antara nol (0) dan satu (1). Nilai 1 menunjukkan akurasi sempurna, dan 0 menunjukkan ketidakakuratan sempurna.

Penyebaran dan prediksi model autopilot

Panduan Autopilot ini mencakup langkah-langkah untuk penerapan model dan pengaturan inferensi waktu nyata.

Setelah Anda melatih model Autopilot Anda, Anda dapat mengatur titik akhir dan mendapatkan prediksi secara interaktif.

Inferensi waktu nyata

Inferensi real-time sangat ideal untuk beban kerja inferensi di mana Anda memiliki persyaratan real-time, interaktif, latensi rendah. Bagian ini menunjukkan bagaimana Anda dapat menggunakan inferensi real-time untuk mendapatkan prediksi secara interaktif dari model Anda.

Anda dapat menggunakan SageMaker API untuk menerapkan model secara manual yang menghasilkan metrik validasi terbaik dalam eksperimen Autopilot sebagai berikut.

Atau, Anda dapat memilih opsi penerapan otomatis saat membuat eksperimen Autopilot Anda. Untuk informasi tentang pengaturan penerapan otomatis model, lihat `ModelDeployConfig` di parameter permintaan. `CreateAutoMLJobV2` Ini menciptakan titik akhir secara otomatis.

> **Note**
>
> Untuk menghindari biaya yang tidak perlu, Anda dapat menghapus titik akhir yang tidak diperlukan dan sumber daya yang dibuat dari penerapan model. Untuk informasi tentang penetapan harga instans menurut Wilayah, lihat SageMaker Harga Amazon.

1. Dapatkan definisi wadah kandidat

   Dapatkan definisi wadah kandidat dari InferenceContainers. Definisi kontainer untuk inferensi mengacu pada lingkungan kontainer yang dirancang untuk menerapkan dan menjalankan SageMaker model terlatih Anda untuk membuat prediksi.

   Contoh AWS CLI perintah berikut menggunakan API DescribeAutoMLJobv2 untuk mendapatkan definisi kandidat untuk kandidat model terbaik.

```
aws sagemaker describe-auto-ml-job-v2 --auto-ml-job-name job-name --region region
```

2. Daftar kandidat

   Contoh AWS CLI perintah berikut menggunakan ListCandidatesForAutoMLJob API untuk daftar semua kandidat model.

```
aws sagemaker list-candidates-for-auto-ml-job --auto-ml-job-name <job-name> --
region <region>
```

3. Buat SageMaker model

   Gunakan definisi kontainer dari langkah sebelumnya dan kandidat pilihan Anda untuk membuat SageMaker model dengan menggunakan CreateModelAPI. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker create-model --model-name '<your-candidate-name>' \
                    --containers ['<container-definition1>, <container-
definition2>, <container-definition3>]' \
                    --execution-role-arn '<execution-role-arn>' --region '<region>
```

4. Buat konfigurasi titik akhir

   Contoh AWS CLI perintah berikut menggunakan CreateEndpointConfigAPI untuk membuat konfigurasi endpoint.

```
aws sagemaker create-endpoint-config --endpoint-config-name '<your-endpoint-config-
name>' \
                    --production-variants '<list-of-production-variants>' \
                    --region '<region>'
```

5. Buat titik akhir

   AWS CLIContoh berikut menggunakan CreateEndpointAPI untuk membuat titik akhir.

```
aws sagemaker create-endpoint --endpoint-name '<your-endpoint-name>' \
                    --endpoint-config-name '<endpoint-config-name-you-just-created>'
 \
                    --region '<region>'
```

Periksa kemajuan penerapan titik akhir Anda dengan menggunakan API. DescribeEndpoint Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-endpoint –endpoint-name '<endpoint-name>' –region <region>
```

Setelah `EndpointStatus` perubahan`InService`, titik akhir siap digunakan untuk inferensi waktu nyata.

6. Memanggil titik akhir

   Struktur perintah berikut memanggil titik akhir untuk inferensi real-time.

```
aws sagemaker invoke-endpoint --endpoint-name '<endpoint-name>' \
                  --region '<region>' --body '<your-data>' [--content-type]
 '<content-type>' <outfile>
```

Laporan penjelasan

Amazon SageMaker Autopilot menyediakan laporan penjelasan untuk membantu menjelaskan bagaimana kandidat model terbaik membuat prediksi untuk masalah klasifikasi teks. Laporan ini dapat membantu insinyur, manajer produk, dan pemangku kepentingan internal lainnya dalam memahami karakteristik model. Baik konsumen maupun regulator mengandalkan transparansi dalam pembelajaran mesin untuk mempercayai dan menafsirkan keputusan yang dibuat berdasarkan prediksi model. Anda dapat menggunakan penjelasan ini untuk mengaudit dan memenuhi persyaratan peraturan, membangun kepercayaan pada model, mendukung pengambilan keputusan manusia, dan men-debug dan meningkatkan kinerja model.

Fungsionalitas penjelasan Autopilot untuk klasifikasi teks menggunakan metode atribusi aksiomatik Gradien Terpadu. Pendekatan ini bergantung pada implementasi Atribusi Aksiomatik untuk Jaringan Dalam.

Autopilot menghasilkan laporan penjelasan sebagai file JSON. Laporan tersebut mencakup rincian analisis yang didasarkan pada dataset validasi. Setiap sampel yang digunakan untuk menghasilkan laporan berisi informasi berikut:

- `text`: Isi teks masukan dijelaskan.
- `token_scores`: Daftar skor untuk setiap token dalam teks.
- • `attribution`: Skor yang menggambarkan pentingnya token.

  - `description.partial_text`: Substring sebagian yang mewakili token.
- `predicted_label`: Kelas label diprediksi oleh kandidat model terbaik.
- `probability`: Keyakinan yang `predicted_label` diprediksi.

Anda dapat menemukan awalan Amazon S3 untuk artefak penjelasan yang dihasilkan untuk kandidat terbaik dalam menanggapi at. `DescribeAutoMLJobV2` `BestCandidate.CandidateProperties.CandidateArtifactLocations.Explainability`

Berikut ini adalah contoh konten analisis yang dapat Anda temukan di artefak penjelasan.


```
{
    "text": "It was a fantastic movie!",
    "predicted_label": 2,
    "probability": 0.9984835,
    "token_scores": [
        {
            "attribution": 0,
            "description": {
                "partial_text": "It"
            }
        },
        {
            "attribution": -0.022447118861679088,
            "description": {
                "partial_text": "was"
            }
        },
        {
            "attribution": -0.2164326456817965,
            "description": {
                "partial_text": "a"
            }
        },
        {
            "attribution": 0.675,
            "description": {
                "partial_text": "fantastic"
            }
        },
        {
            "attribution": 0.416,
```

```
            "description": {
                "partial_text": "movie!"
            }
        }
    ]
}
```

Dalam sampel laporan JSON ini, fungsionalitas penjelasan mengevaluasi teks `It was a fantastic movie!` dan menilai kontribusi masing-masing tokennya ke label prediksi keseluruhan. Label yang diprediksi adalah2, yang merupakan sentimen positif yang kuat, dengan probabilitas 99,85%. Sampel JSON kemudian merinci kontribusi masing-masing token individu terhadap prediksi ini. Misalnya, token `fantastic` memiliki atribusi yang lebih kuat daripada token`was`. Ini adalah token yang berkontribusi paling besar pada prediksi akhir.

Laporan kinerja model

Laporan kualitas SageMaker model Amazon (juga disebut sebagai laporan kinerja) memberikan wawasan dan informasi kualitas untuk kandidat model terbaik yang dihasilkan oleh pekerjaan AutoML. Ini termasuk informasi tentang detail pekerjaan, jenis masalah model, fungsi objektif, dan berbagai metrik. Bagian ini merinci konten laporan kinerja untuk masalah klasifikasi teks dan menjelaskan cara mengakses metrik sebagai data mentah dalam file JSON.

Anda dapat menemukan awalan Amazon S3 untuk artefak laporan kualitas model yang dihasilkan untuk kandidat terbaik dalam menanggapi at. `DescribeAutoMLJobV2 BestCandidate.CandidateProperties.CandidateArtifactLocations.ModelInsights`

Laporan kinerja berisi dua bagian:

- Bagian pertama berisi rincian tentang pekerjaan Autopilot yang menghasilkan model.
- Bagian kedua berisi laporan kualitas model dengan berbagai metrik kinerja.

Detail pekerjaan Autopilot

Bagian pertama dari laporan ini memberikan beberapa informasi umum tentang pekerjaan Autopilot yang menghasilkan model. Rincian ini mencakup informasi berikut:

- Nama kandidat autopilot: Nama kandidat model terbaik.
- Nama pekerjaan autopilot: Nama pekerjaan.
- Jenis masalah: Jenis masalah. Dalam kasus kami, klasifikasi teks.

- Metrik objektif: Metrik objektif yang digunakan untuk mengoptimalkan kinerja model. Dalam kasus kami, Akurasi.
- Arah optimasi: Menunjukkan apakah akan meminimalkan atau memaksimalkan metrik objektif.

Laporan kualitas model

Informasi kualitas model dihasilkan oleh wawasan model Autopilot. Konten laporan yang dihasilkan bergantung pada jenis masalah yang ditangani. Laporan tersebut menentukan jumlah baris yang termasuk dalam dataset evaluasi dan waktu evaluasi terjadi.

Tabel metrik

Bagian pertama dari laporan kualitas model berisi tabel metrik. Ini sesuai untuk jenis masalah yang ditangani model.

Gambar berikut adalah contoh tabel metrik yang dihasilkan oleh Autopilot untuk masalah klasifikasi gambar atau teks. Ini menunjukkan nama metrik, nilai, dan standar deviasi.

Metrics table

Metric Name	Value	Standard Deviation
weighted_recall	0.597104	0.005410
weighted_precision	0.591693	0.005729
accuracy	0.597104	0.005410
weighted_f0_5	0.592155	0.005659
weighted_f1	0.593423	0.005554
weighted_f2	0.595392	0.005456
accuracy_best_constant_classifier	0.200699	0.004422
weighted_recall_best_constant_classifier	0.200699	0.004422
weighted_precision_best_constant_classifier	0.040280	0.001753
weighted_f0_5_best_constant_classifier	0.047944	0.002039
weighted_f1_best_constant_classifier	0.067094	0.002684
weighted_f2_best_constant_classifier	0.111716	0.003808

Informasi kinerja model grafis

Bagian kedua dari laporan kualitas model berisi informasi grafis untuk membantu Anda mengevaluasi kinerja model. Isi bagian ini tergantung pada jenis masalah yang dipilih.

Matriks kebingungan

Matriks kebingungan menyediakan cara untuk memvisualisasikan keakuratan prediksi yang dibuat oleh model untuk klasifikasi biner dan multikelas untuk masalah yang berbeda.

Ringkasan komponen grafik tingkat positif palsu (FPR) dan tingkat positif sejati (TPR) didefinisikan sebagai berikut.

- Prediksi yang benar
  - True positive (TP): Nilai yang diprediksi adalah 1, dan nilai sebenarnya adalah 1.
  - Benar negatif (TN): Nilai yang diprediksi adalah 0, dan nilai sebenarnya adalah 0.
- Prediksi yang salah
  - Positif palsu (FP): Nilai yang diprediksi adalah 1, tetapi nilai sebenarnya adalah 0.
  - False negative (FN): Nilai yang diprediksi adalah 0, tetapi nilai sebenarnya adalah 1.

Matriks kebingungan dalam laporan kualitas model berisi yang berikut ini.

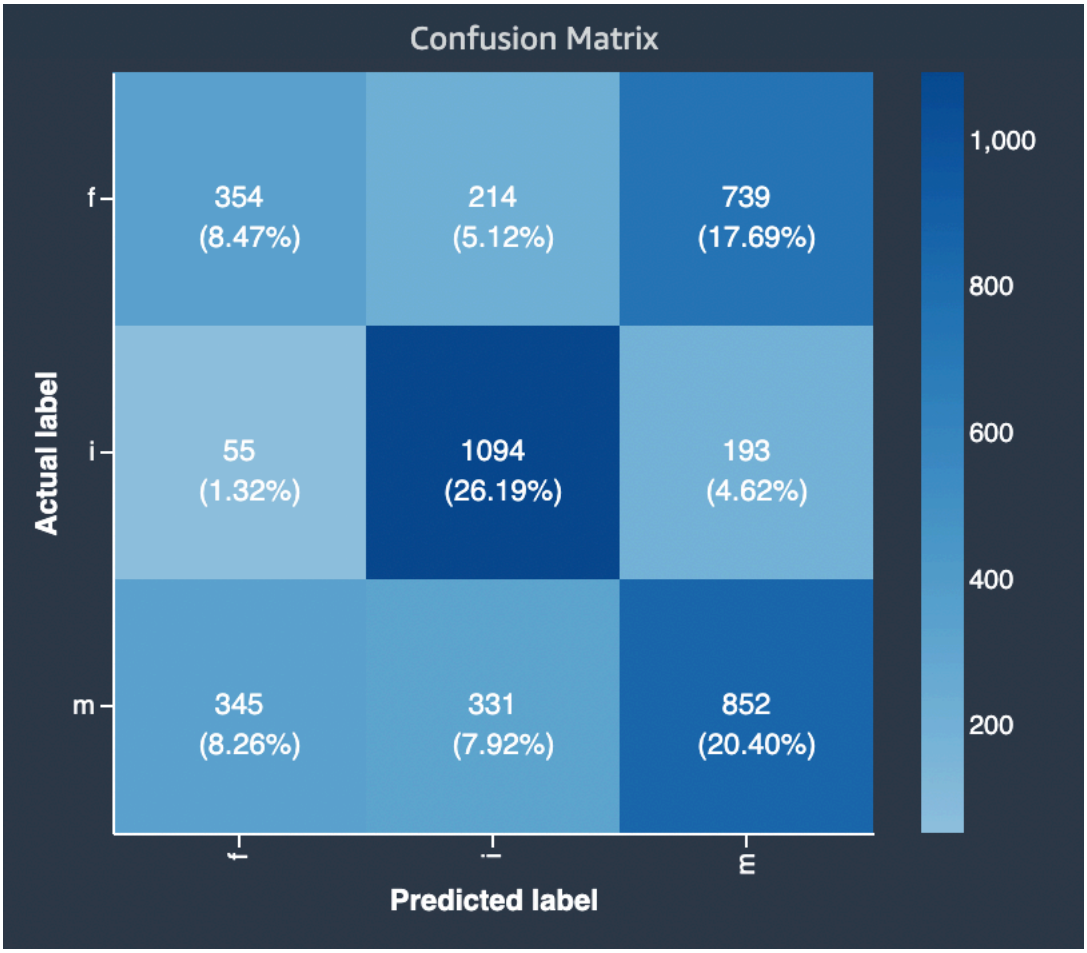
- Jumlah dan persentase prediksi yang benar dan salah untuk label yang sebenarnya
- Jumlah dan persentase prediksi akurat pada diagonal dari kiri atas ke pojok kanan bawah
- Jumlah dan persentase prediksi yang tidak akurat pada diagonal dari kanan atas ke sudut kiri bawah

Prediksi yang salah pada matriks kebingungan adalah nilai kebingungan.

Diagram berikut adalah contoh matriks kebingungan untuk masalah klasifikasi multi-kelas. Matriks kebingungan dalam laporan kualitas model berisi yang berikut ini.

- Sumbu vertikal dibagi menjadi tiga baris yang berisi tiga label aktual yang berbeda.
- Sumbu horizontal dibagi menjadi tiga kolom yang berisi label yang diprediksi oleh model.
- Bilah warna memberikan nada yang lebih gelap ke sejumlah besar sampel untuk secara visual menunjukkan jumlah nilai yang diklasifikasikan dalam setiap kategori.

Dalam contoh di bawah ini, model dengan benar memprediksi 354 nilai aktual untuk label f, 1094 nilai untuk label i dan 852 nilai untuk label m. Perbedaan nada menunjukkan bahwa kumpulan data tidak seimbang karena ada lebih banyak label untuk nilai i daripada untuk f atau m.



Matriks kebingungan dalam laporan kualitas model yang disediakan dapat mengakomodasi maksimum 15 label untuk jenis masalah klasifikasi multikelas. Jika baris yang sesuai dengan label menunjukkan `Nan` nilai, itu berarti kumpulan data validasi yang digunakan untuk memeriksa prediksi model tidak berisi data dengan label tersebut.

Buat pekerjaan AutoML untuk peramalan deret waktu menggunakan API

Peramalan dalam pembelajaran mesin mengacu pada proses memprediksi hasil atau tren masa depan berdasarkan data dan pola historis. Dengan menganalisis data deret waktu lalu dan mengidentifikasi pola yang mendasarinya, algoritma pembelajaran mesin dapat membuat prediksi dan memberikan wawasan berharga tentang perilaku masa depan. Dalam peramalan, tujuannya adalah untuk mengembangkan model yang dapat secara akurat menangkap hubungan antara variabel input dan variabel target dari waktu ke waktu. Ini melibatkan pemeriksaan berbagai faktor seperti tren, musiman, dan pola relevan lainnya dalam data. Informasi yang dikumpulkan kemudian digunakan untuk melatih model pembelajaran mesin. Model terlatih mampu menghasilkan prediksi dengan mengambil data input baru dan menerapkan pola dan hubungan yang dipelajari. Ini dapat

memberikan perkiraan untuk berbagai kasus penggunaan, seperti proyeksi penjualan, tren pasar saham, prakiraan cuaca, perkiraan permintaan, dan banyak lagi.

Petunjuk berikut menunjukkan cara membuat pekerjaan Amazon SageMaker Autopilot sebagai percobaan percontohan untuk jenis masalah peramalan deret waktu menggunakan Referensi API. SageMaker

> **Note**
>
> Tugas seperti klasifikasi teks dan gambar, peramalan deret waktu, dan penyempurnaan model bahasa besar tersedia secara eksklusif melalui API Autopilot versi 2. Untuk pengguna Python, sebaiknya gunakan AWS SDK for Python (Boto3)sebagai Amazon SageMaker Python SDK saat ini tidak didukung untuk Autopilot API versi 2.
> Pengguna yang lebih menyukai kenyamanan antarmuka pengguna dapat menggunakan Amazon SageMaker Canvas untuk mengakses model pra-terlatih dan model dasar AI generatif, atau membuat model khusus yang disesuaikan untuk teks tertentu, klasifikasi gambar, kebutuhan peramalan, atau AI generatif.

Anda dapat membuat eksperimen peramalan deret waktu Autopilot secara terprogram dengan memanggil API `CreateAutoMLJobV2`dalam bahasa apa pun yang didukung oleh Amazon Autopilot atau. SageMaker AWS CLI

Untuk informasi tentang cara tindakan API ini diterjemahkan ke dalam fungsi dalam bahasa pilihan Anda, lihat bagian Lihat Juga `CreateAutoMLJobV2` dan pilih SDK. Sebagai contoh, untuk pengguna Python, lihat sintaks permintaan lengkap dari in. `create_auto_ml_job_v2` AWS SDK for Python (Boto3)

Autopilot melatih beberapa kandidat model dengan deret waktu target Anda, lalu memilih model peramalan optimal untuk metrik objektif tertentu. Ketika kandidat model Anda telah dilatih, Anda dapat menemukan metrik kandidat terbaik dalam menanggapi `DescribeAutoMLJobV2` at`BestCandidate`.

Bagian berikut menentukan parameter permintaan input wajib dan opsional untuk `CreateAutoMLJobV2` API yang digunakan dalam peramalan deret waktu.

> **Note**
>
> Lihat Peramalan Seri Waktu notebook dengan Amazon SageMaker Autopilot untuk contoh prakiraan deret waktu yang praktis dan praktis. Di buku catatan ini, Anda menggunakan Amazon SageMaker Autopilot untuk melatih model deret waktu dan menghasilkan prediksi menggunakan model terlatih. Notebook ini memberikan instruksi untuk mengambil kumpulan data data historis tabular yang sudah jadi di Amazon S3.

Prasyarat

Sebelum menggunakan Autopilot untuk membuat eksperimen peramalan deret waktu, pastikan untuk: SageMaker

- Siapkan kumpulan data deret waktu Anda. Persiapan dataset melibatkan pengumpulan data yang relevan dari berbagai sumber, membersihkan dan menyaringnya untuk menghilangkan noise dan inkonsistensi, dan mengaturnya ke dalam format terstruktur. Lihat Format kumpulan data deret waktu dan metode pengisian nilai yang hilang untuk mempelajari lebih lanjut tentang persyaratan format deret waktu di Autopilot. Secara opsional, Anda dapat melengkapi kumpulan data Anda dengan kalender hari libur nasional negara pilihan Anda untuk menangkap pola terkait. Untuk informasi lebih lanjut tentang kalender liburan, lihatKalender hari libur nasional.
- Tempatkan data deret waktu Anda di bucket Amazon S3.
- Berikan akses penuh ke bucket Amazon S3 yang berisi data masukan untuk peran SageMaker eksekusi yang digunakan untuk menjalankan eksperimen. Setelah ini selesai, Anda dapat menggunakan ARN dari peran eksekusi ini dalam permintaan API Autopilot.
  - Untuk informasi tentang mengambil peran SageMaker eksekusi Anda, lihatDapatkan peran eksekusi.
  - Untuk informasi tentang pemberian izin peran SageMaker eksekusi untuk mengakses satu atau beberapa bucket tertentu di Amazon S3, lihat Menambahkan Izin Amazon S3 Tambahan ke Peran Eksekusi di. SageMaker Buat peran eksekusi

Parameter yang diperlukan

Saat menelepon `CreateAutoMLJobV2` untuk membuat eksperimen Autopilot untuk peramalan deret waktu, Anda harus memberikan nilai berikut:

- An `AutoMLJobName` untuk menentukan nama pekerjaan Anda. Nama harus bertipe`string`, dan harus memiliki panjang minimal 1 karakter dan panjang maksimum 32.
- Setidaknya satu `AutoMLJobChannel` `AutoMLJobInputDataConfig` di mana Anda menentukan nama bucket Amazon S3 yang berisi data Anda. Secara opsional, Anda dapat menentukan jenis konten (file CSV atau Parket) dan kompresi (GZip).
- `AutoMLProblemTypeConfig`Jenis `TimeSeriesForecastingJobConfig` untuk mengonfigurasi pengaturan pekerjaan peramalan deret waktu Anda. Secara khusus, Anda harus menentukan:
  - Frekuensi prediksi, yang mengacu pada perincian yang diinginkan (per jam, harian, bulanan, dll) dari perkiraan Anda.

    Interval yang valid adalah bilangan bulat diikuti oleh `Y` (Tahun), `M` (Bulan), `W` (Minggu), `D` (Hari), `H` (Jam), dan `min` (Menit). Misalnya, `1D` menunjukkan setiap hari dan `15min` menunjukkan setiap 15 menit. Nilai frekuensi tidak boleh tumpang tindih dengan frekuensi yang lebih besar berikutnya. Misalnya, Anda harus menggunakan frekuensi `1H` alih-alih`60min`.

    Nilai yang valid untuk setiap frekuensi adalah sebagai berikut:

    - Menit - 1-59
    - Jam - 1-23
    - Hari - 1-6
    - Minggu - 1-4
    - Bulan - 1-11
    - Tahun - 1
  - Cakrawala prediksi dalam perkiraan Anda, yang mengacu pada jumlah langkah waktu yang diprediksi model. Cakrawala ramalan juga disebut panjang prediksi. Cakrawala perkiraan maksimum adalah kurang dari 500 langkah waktu atau 1/4 dari langkah waktu dalam kumpulan data.
  - A TimeSeriesConfigdi mana Anda menentukan skema kumpulan data Anda untuk memetakan header kolom ke perkiraan Anda dengan menentukan:
    - A`TargetAttributeName`: Kolom yang berisi data historis bidang target yang akan diramalkan.
    - A`TimestampAttributeName`: Kolom yang berisi titik waktu di mana nilai target dari item tertentu dicatat.

- `AItemIdentifierAttributeName`: Kolom yang berisi pengidentifikasi item yang ingin Anda prediksi nilai targetnya.

Berikut ini adalah contoh parameter permintaan tersebut. Dalam contoh ini, Anda menyiapkan perkiraan harian untuk jumlah yang diharapkan atau tingkat permintaan item tertentu selama periode 20 hari.

```
"AutoMLProblemTypeConfig": {
        "ForecastFrequency": "D",
        "ForecastHorizon": 20,
        "TimeSeriesConfig": {
            "TargetAttributeName": "demand",
            "TimestampAttributeName": "timestamp",
            "ItemIdentifierAttributeName": "item_id"
        },
```

- `OutputDataConfig`Untuk menentukan jalur keluaran Amazon S3 untuk menyimpan artefak pekerjaan AutoML Anda.
- A `RoleArn` untuk menentukan ARN dari peran yang digunakan untuk mengakses data Anda. Anda dapat menggunakan ARN dari peran eksekusi yang telah Anda berikan akses ke data Anda.

Semua parameter lainnya adalah opsional. Misalnya, Anda dapat mengatur kuantil perkiraan tertentu, memilih metode pengisian untuk nilai yang hilang dalam kumpulan data, atau menentukan cara menggabungkan data yang tidak sejajar dengan frekuensi perkiraan. Untuk mempelajari cara mengatur parameter tambahan tersebut, lihatParameter opsional.

Parameter opsional

Bagian berikut memberikan rincian beberapa parameter opsional yang dapat Anda berikan ke pekerjaan AutoML perkiraan deret waktu Anda.

Cara menentukan kuantil kustom

Autopilot melatih 6 kandidat model dengan deret waktu target Anda, kemudian menggabungkan model ini menggunakan metode ansambel susun untuk membuat model peramalan optimal untuk metrik objektif tertentu. Setiap model peramalan Autopilot menghasilkan perkiraan probabilistik dengan menghasilkan perkiraan pada kuantil antara P1 dan P99. Kuantil ini digunakan untuk menjelaskan ketidakpastian perkiraan. Secara default, perkiraan akan dihasilkan untuk 0.1 (`p10`), 0.5 (`p50`), dan 0.9 (`p90`). Anda dapat memilih untuk menentukan kuantil Anda sendiri.

Di Autopilot, Anda dapat menentukan hingga lima kuantil perkiraan dari 0,01 (p1) hingga 0,99 (p99), dengan penambahan 0,01 atau lebih tinggi dalam atribut. `ForecastQuantiles` TimeSeriesForecastingJobConfig

Dalam contoh berikut, Anda menyiapkan perkiraan persentil 10, 25, 50, 75, dan 90 harian untuk jumlah yang diharapkan atau tingkat permintaan barang tertentu selama periode 20 hari.

```
"AutoMLProblemTypeConfig": {
        "ForecastFrequency": "D",
        "ForecastHorizon": 20,
        "ForecastQuantiles": ["p10", "p25", "p50", "p75", "p90"],
        "TimeSeriesConfig": {
            "TargetAttributeName": "demand",
            "TimestampAttributeName": "timestamp",
            "ItemIdentifierAttributeName": "item_id"
        },
```

Cara mengumpulkan data untuk frekuensi perkiraan yang berbeda

Untuk membuat model perkiraan (juga disebut sebagai kandidat model terbaik dari eksperimen Anda), Anda harus menentukan frekuensi perkiraan. Frekuensi perkiraan menentukan frekuensi prediksi dalam perkiraan Anda. Misalnya, perkiraan penjualan bulanan. Model terbaik autopilot dapat menghasilkan perkiraan untuk frekuensi data yang lebih tinggi dari frekuensi di mana data Anda direkam.

Selama pelatihan, Autopilot mengumpulkan data apa pun yang tidak selaras dengan frekuensi perkiraan yang Anda tentukan. Misalnya, Anda mungkin memiliki beberapa data harian tetapi tentukan frekuensi perkiraan mingguan. Autopilot menyelaraskan data harian berdasarkan minggu di mana ia berada. Autopilot kemudian menggabungkannya menjadi rekor tunggal untuk setiap minggu.

Selama agregasi, metode transformasi default adalah menjumlahkan data. Anda dapat mengonfigurasi agregasi saat membuat pekerjaan AutoML Anda `Transformations` di atribut. TimeSeriesForecastingJobConfig Metode agregasi yang didukung adalah `sum` (default),,`avg`, `firstmin`,`max`. Agregasi hanya didukung untuk kolom target.

Dalam contoh berikut, Anda mengonfigurasi agregasi untuk menghitung rata-rata perkiraan promo individu untuk memberikan nilai perkiraan agregat akhir.

```
"Transformations": {
            "Aggregation": {
                "promo": "avg"
```

```
            }
        }
```

Cara menangani nilai yang hilang dalam kumpulan data input Anda

Autopilot menyediakan sejumlah metode pengisian untuk menangani nilai yang hilang di target dan kolom numerik lainnya dari kumpulan data deret waktu Anda. Untuk informasi tentang daftar metode pengisian yang didukung dan logika pengisiannya yang tersedia, lihatTangani nilai yang hilang.

Anda mengonfigurasi strategi pengisian Anda dalam `Transformations` atribut TimeSeriesForecastingJobConfigsaat membuat pekerjaan AutoML Anda.

Untuk mengatur metode pengisian, Anda perlu memberikan pasangan kunci-nilai:

- Kuncinya adalah nama kolom yang ingin Anda tentukan metode pengisiannya.
- Nilai yang terkait dengan kunci adalah objek yang mendefinisikan strategi pengisian untuk kolom itu.

Anda dapat menentukan beberapa metode pengisian untuk satu kolom.

Untuk menetapkan nilai tertentu untuk metode pengisian, Anda harus mengatur parameter isian ke nilai metode pengisian yang diinginkan (misalnya`"backfill" : "value"`), dan menentukan nilai pengisian aktual dalam parameter tambahan yang diakhiran dengan “_value”. Misalnya, untuk mengatur `backfill` ke nilai2, Anda harus menyertakan dua parameter: `"backfill": "value"` dan`"backfill_value":"2"`.

Dalam contoh berikut, Anda menentukan strategi pengisian untuk kolom data yang tidak lengkap, “harga” sebagai berikut: Semua nilai yang hilang antara titik data pertama suatu item dan yang terakhir diatur ke `0` setelah itu semua nilai yang hilang diisi dengan nilai 2 hingga tanggal akhir kumpulan data.

```
"Transformations": {
            "Filling": {
                "price": {
                        "middlefill" : "zero",
                        "backfill" : "value",
                        "backfill_value": "2"
                }
            }
        }
```

Cara menentukan metrik objektif

Autopilot menghasilkan metrik akurasi untuk mengevaluasi kandidat model dan membantu Anda memilih mana yang akan digunakan untuk menghasilkan perkiraan. Saat Anda menjalankan eksperimen peramalan deret waktu, Anda dapat memilih AutoML untuk membiarkan Autopilot mengoptimalkan prediktor untuk Anda, atau Anda dapat secara manual memilih algoritme untuk prediktor Anda.

Secara default, Autopilot menggunakan Average Weighted Quantile Loss. Namun, Anda dapat mengonfigurasi metrik objektif saat membuat pekerjaan AutoML Anda di `MetricName` atribut AutoML. JobObjective

Untuk daftar algoritma yang tersedia, lihatDukungan algoritma untuk peramalan deret waktu.

Cara memasukkan informasi hari libur nasional ke kumpulan data Anda

Di Autopilot, Anda dapat menggabungkan kumpulan data informasi hari libur nasional yang direkayasa fitur ke deret waktu Anda. Autopilot memberikan dukungan asli untuk kalender liburan lebih dari 250 negara. Setelah Anda memilih negara, Autopilot menerapkan kalender liburan negara tersebut ke setiap item dalam kumpulan data Anda selama pelatihan. Ini memungkinkan model untuk mengidentifikasi pola yang terkait dengan hari libur tertentu.

Anda dapat mengaktifkan featurisasi liburan saat Anda membuat pekerjaan AutoML Anda dengan meneruskan HolidayConfigAttributesobjek ke atribut. `HolidayConfig` TimeSeriesForecastingJobConfig `HolidayConfigAttributes`Objek berisi `CountryCode` atribut dua huruf yang menentukan negara kalender hari libur nasional yang digunakan untuk menambah kumpulan data deret waktu Anda.

Lihat daftar kalender yang didukung dan kode negara yang sesuai. Kode Negara

Cara mengaktifkan penyebaran otomatis

Autopilot memungkinkan Anda untuk secara otomatis menerapkan model perkiraan Anda ke titik akhir. Untuk mengaktifkan penerapan otomatis untuk kandidat model terbaik dari pekerjaan AutoML, sertakan `ModelDeployConfig` a dalam permintaan pekerjaan AutoML. Ini memungkinkan penerapan model terbaik ke titik SageMaker akhir. Di bawah ini adalah konfigurasi yang tersedia untuk kustomisasi.

- Untuk membiarkan Autopilotmenghasilkan nama titik akhir, setel ke. `AutoGenerateEndpointName` `True`

- Untuk memberikan nama Anda sendiri untuk titik akhir, atur`AutoGenerateEndpointName` to `False and provide a name of your choice in` `EndpointName`.

Format kumpulan data deret waktu dan metode pengisian nilai yang hilang

Data deret waktu mengacu pada kumpulan pengamatan atau pengukuran yang direkam selama interval waktu yang teratur. Dalam jenis data ini, setiap pengamatan dikaitkan dengan stempel waktu atau periode waktu tertentu, menciptakan urutan titik data yang diurutkan secara kronologis.

Kolom spesifik yang Anda sertakan dalam kumpulan data deret waktu bergantung pada tujuan analisis Anda dan data yang tersedia untuk Anda. Minimal, data deret waktu terdiri dari tabel 3 kolom di mana:

- Satu kolom berisi pengidentifikasi unik yang ditugaskan ke item individual untuk merujuk nilainya pada saat tertentu.
- Kolom lain mewakili point-in-time nilai atau target untuk mencatat nilai item tertentu pada saat tertentu. Setelah model dilatih pada nilai-nilai target tersebut, kolom target ini berisi nilai-nilai yang diprediksi model pada frekuensi tertentu dalam cakrawala yang ditentukan.
- Dan kolom stempel waktu disertakan untuk mencatat tanggal dan waktu ketika nilai diukur.
- Kolom tambahan dapat berisi faktor-faktor lain yang dapat mempengaruhi kinerja perkiraan. Misalnya, dalam kumpulan data deret waktu untuk ritel di mana targetnya adalah penjualan atau pendapatan, Anda mungkin menyertakan fitur yang memberikan informasi tentang unit yang terjual, ID produk, lokasi toko, jumlah pelanggan, tingkat inventaris, serta indikator kovariat seperti data cuaca atau informasi demografis.

> **Note**
>
> Anda dapat menambahkan kumpulan data informasi hari libur nasional yang direkayasa fitur ke deret waktu Anda. Dengan memasukkan liburan dalam model deret waktu Anda, Anda dapat menangkap pola periodik yang dibuat liburan. Ini membantu perkiraan Anda mencerminkan musim yang mendasari data Anda dengan lebih baik. Untuk informasi tentang kalender yang tersedia per negara, lihat Kalender hari libur nasional

Format kumpulan data untuk peramalan deret waktu

Autopilot mendukung tipe data numerik, kategoris, teks, dan datetime. Tipe data kolom target harus numerik.

Autopilot mendukung data deret waktu yang diformat sebagai file CSV (default) atau sebagai file Parket.

- CSV (comma-separated-values) adalah format file berbasis baris yang menyimpan data dalam teks biasa yang dapat dibaca manusia yang merupakan pilihan populer untuk pertukaran data karena didukung oleh berbagai aplikasi.
- Parket adalah format file berbasis kolom di mana data disimpan dan diproses lebih efisien daripada format file berbasis baris. Ini membuat mereka menjadi pilihan yang lebih baik untuk masalah data besar.

Untuk informasi selengkapnya tentang batas sumber daya pada kumpulan data deret waktu untuk peramalan di Autopilot, lihat. Batas sumber daya peramalan deret waktu Amazon SageMaker Autopilot

Tangani nilai yang hilang

Masalah umum dalam data peramalan deret waktu adalah adanya nilai yang hilang. Data Anda mungkin berisi nilai yang hilang karena sejumlah alasan, termasuk kegagalan pengukuran, masalah pemformatan, kesalahan manusia, atau kurangnya informasi untuk direkam. Misalnya, jika Anda memperkirakan permintaan produk untuk toko ritel dan barang terjual habis atau tidak tersedia, tidak akan ada data penjualan untuk dicatat saat barang itu kehabisan stok. Jika cukup umum, nilai yang hilang dapat secara signifikan memengaruhi akurasi model.

Autopilot menyediakan sejumlah metode pengisian untuk menangani nilai yang hilang, dengan pendekatan berbeda untuk kolom target dan kolom tambahan lainnya. Pengisian adalah proses menambahkan nilai standar ke entri yang hilang dalam kumpulan data Anda.

Lihat Cara menangani nilai yang hilang dalam kumpulan data input Anda untuk mempelajari cara mengatur metode untuk mengisi nilai yang hilang dalam kumpulan data deret waktu Anda.

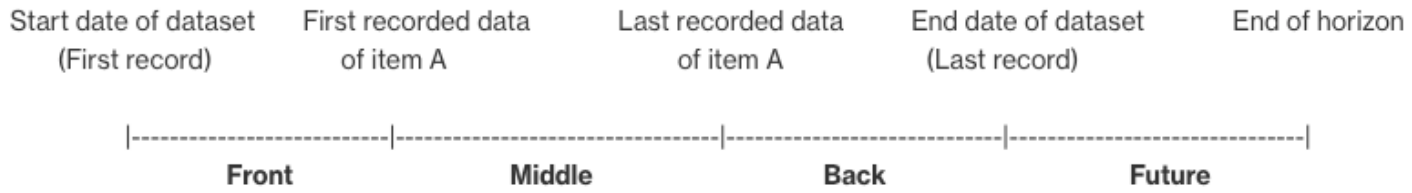
Autopilot mendukung metode pengisian berikut:

- Pengisian depan: Mengisi nilai yang hilang antara titik data tercatat paling awal di antara semua item dan titik awal setiap item (setiap item dapat dimulai pada waktu yang berbeda). Ini

memastikan bahwa data untuk setiap item lengkap dan mencakup dari titik data yang direkam paling awal ke titik awal masing-masing.

- Pengisian tengah: Mengisi nilai yang hilang antara tanggal mulai dan akhir item dalam kumpulan data.
- Pengisian kembali: Mengisi nilai yang hilang antara titik data terakhir dari setiap item (setiap item dapat berhenti pada waktu yang berbeda) dan titik data terakhir yang direkam di antara semua item.
- Pengisian di masa depan: Mengisi nilai yang hilang antara titik data terakhir yang direkam di antara semua item dan akhir cakrawala perkiraan.

Gambar berikut memberikan representasi visual dari metode pengisian yang berbeda.



Pilih logika pengisian

Saat memilih logika pengisian, Anda harus mempertimbangkan bagaimana logika akan ditafsirkan oleh model Anda. Misalnya, dalam skenario ritel, mencatat 0 penjualan barang yang tersedia berbeda dengan mencatat 0 penjualan barang yang tidak tersedia, karena yang terakhir tidak menyiratkan kurangnya minat pelanggan pada item tersebut. Karena itu, `0` mengisi kolom target deret waktu dapat menyebabkan prediktor menjadi kurang bias dalam prediksinya, sementara `NaN` pengisian mungkin mengabaikan kejadian aktual dari 0 item yang tersedia yang dijual dan menyebabkan prediktor menjadi terlalu bias.

Mengisi logika

Anda dapat melakukan pengisian pada kolom target dan kolom numerik lainnya dalam kumpulan data Anda. Kolom target memiliki pedoman dan batasan pengisian yang berbeda dari kolom numerik lainnya.

Pedoman Pengisian

Tipe kolom	Mengisi secara default?	Metode pengisian yang didukung	Logika pengisian default	Logika pengisian yang diterima
Kolom target	Ya	Isi tengah dan belakang	0	• `zero`- 0 mengisi. • `value`- bilangan bulat atau nomor float. • `nan`- Bukan angka. • `mean`- nilai rata-rata dari seri data. • `median`- nilai median dari seri data. • `min`- nilai minimum dari seri data. • `max`- nilai maksimum dari seri data.
Kolom numerik lainnya	Tidak	Pengisian tengah, belakang, dan future	Tidak ada default	• `zero`- 0 mengisi. • `value`- nilai integer atau float. • `mean`- nilai rata-rata dari seri data.

Tipe kolom	Mengisi secara default?	Metode pengisian yang didukung	Logika pengisian default	Logika pengisian yang diterima
				• `median`- nilai median dari seri data. • `min`- nilai minimum dari seri data. • `max`- nilai maksimum dari seri data.

> **Note**
>
> Untuk target dan kolom numerik lainnya,, `meanmedian`,`min`, dan `max` dihitung berdasarkan jendela bergulir dari 64 entri data terbaru sebelum nilai yang hilang.

Kalender hari libur nasional

Autopilot mendukung kumpulan data informasi hari libur nasional yang direkayasa fitur yang menyediakan akses ke kalender liburan lebih dari 250 negara.

Fitur kalender liburan sangat berguna dalam domain ritel, di mana hari libur nasional dapat secara signifikan mempengaruhi permintaan.

Lihat Cara memasukkan informasi hari libur nasional ke kumpulan data Anda untuk mempelajari cara menambahkan kalender ke kumpulan data Anda.

Kode Negara

Autopilot memberikan dukungan asli untuk kalender hari libur umum dari negara-negara berikut. Gunakan Kode Negara saat menentukan negara dengan API.

Negara yang Didukung

Negara	Kode Negara
Afghanistan	AF
Kepulauan Åland	AX
Albania	AL
Aljazair	DZ
Samoa Amerika	AS
Andorra	AD
Angola	AO
Anguilla	AI
Antartika	AQ
Antigua dan Barbuda	AG
Argentina	AR
Armenia	AM
Aruba	AW
Australia	AU
Austria	AT
Azerbaijan	AZ
Bahama	BS
Bahrain	BH
Bangladesh	BD
Barbados	BB

Negara	Kode Negara
Belarus	BY
Belgium	BE
Belize	BZ
Benin	BJ
Bermuda	BM
Bhutan	BT
Bolivia	BO
Bosnia dan Herzegovina	BA
Botswana	BW
Pulau Bouvet	BV
Brazil	BR
Wilayah Samudra Hindia Britania	IO
Kepulauan Virgin Inggris	VG
Brunei Darussalam	BN
Bulgaria	BG
Burkina Faso	BF
Burundi	BI
Kamboja	KH
Kamerun	CM
Canada	CA

Negara	Kode Negara
Tanjung Verde	CV
Karibia Belanda	BQ
Kepulauan Cayman	KY
Republik Afrika Tengah	CF
Chad	TD
Chili	CL
Tiongkok	CN
Pulau Natal	CX
Kepulauan Cocos (Keeling)	CC
Kolombia	CO
Komoro	KM
Kepulauan Cook	CK
Kosta Rika	CR
Kroasia	HR
Kuba	CU
Curacao	CW
Cyprus	CY
Ceko	CZ
Republik Demokrasi Kongo	CD
Denmark	DK

Negara	Kode Negara
Djibouti	DJ
Dominika	DM
Republik Dominika	DO
Ekuador	EC
Mesir	EG
El Salvador	SV
Guinea Khatulistiwa	GQ
Eritrea	ER
Estonia	EE
Eswatini	SZ
Etiopia	ET
Kepulauan Falkland	FK
Kepulauan Faroe	FO
Fiji	FJ
Finland	FI
France	FR
Guyana Prancis	GF
Polinesia Prancis	PF
Wilayah Selatan Prancis	TF
Gabon	GA

Negara	Kode Negara
Gambia	GM
Georgia	GE
Germany	DE
Ghana	GH
Gibraltar	GI
Greece	GR
Greenland	GL
Grenada	GD
Guadeloupe	GP
Guam	GU
Guatemala	GT
Guernsey	GG
Guinea	GN
Guinea-Bissau	GW
Guyana	GY
Haiti	HT
Pulau Heard dan McDonald Kepulauan	HM
Honduras	HN
Hong Kong	HK
Hungary	HU

Negara	Kode Negara
Islandia	IS
India	IN
Indonesia	ID
Iran	IR
Irak	IQ
Irlandia	IE
Pulau Man	IM
Israel	IL
Italy	IT
Pantai Gading	CI
Jamaika	JM
Jepang	JP
Jersey	JE
Yordania	JO
Kazakstan	KZ
Kenya	KE
Kiribati	KI
Kosovo	XK
Kuwait	KW
Kirgistan	KG

Negara	Kode Negara
Laos	LA
Latvia	LV
Libanon	LB
Lesotho	LS
Liberia	LR
Libya	LY
Liechtenstein	LI
Lithuania	LT
Luxembourg	LU
Makau	MO
Madagaskar	MG
Malawi	MW
Malaysia	MY
Maladewa	MV
Mali	ML
Malta	MT
Kepulauan Marshall	MH
Martinik	MQ
Mauritania	MR
Mauritius	MU

Negara	Kode Negara
Mayotte	YT
Meksiko	MX
Mikronesia	FM
Moldova	MD
Monako	MC
Mongolia	MN
Montenegro	ME
Montserrat	MS
Maroko	MA
Mozambik	MZ
Myanmar	MM
Namibia	NA
Nauru	NR
Nepal	NP
Belanda	NL
Kaledonia Baru	NC
Selandia Baru	NZ
Nikaragua	NI
Niger	NE
Nigeria	NG

Negara	Kode Negara
Niue	NU
Pulau Norfolk	NF
Korea Utara	KP
Makedonia Utara	MK
Kepulauan Mariana Utara	MP
Norwegia	NO
Oman	OM
Pakistan	PK
Palau	PW
Palestina	PS
Panama	PA
Papua Nugini	PG
Paraguay	PY
Peru	PE
Filipina	PH
Kepulauan Pitcairn	PN
Poland	PL
Portugal	PT
Puerto Riko	PR
Qatar	QA

Negara	Kode Negara
Republik Kongo	CG
Reuni	RE
Romania	RO
Federasi Rusia	RU
Rwanda	RW
Santo Barthélemy	BL
“Santo Helena, Kenaikan dan Tristan da Cunha”	SH
Saint Kitts dan Nevis	KN
Saint Lucia	LC
Santo Martin	MF
Saint Pierre dan Miquelon	PM
Saint Vincent dan Grenadines	VC
Samoa	WS
San Marino	SM
Sao Tome dan Principe	ST
Arab Saudi	SA
Senegal	SN
Serbia	RS
Seychelles	SC
Sierra Leone	SL

Negara	Kode Negara
Singapura	SG
Sint Maarten	SX
Slovakia	SK
Slovenia	SI
Kepulauan Solomon	SB
Somalia	SO
Afrika Selatan	ZA
Georgia Selatan dan Kepulauan Sandwich Selatan	GS
Korea Selatan	KR
Sudan Selatan	SS
Spain	ES
Sri Lanka	LK
Sudan	SD
Suriname	SR
Svalbard dan Jan Mayen	SJ
Sweden	SE
Swiss	CH
Republik Arab Suriah	SY
Taiwan	TW
Tajikistan	TJ

Negara	Kode Negara
Tanzania	TZ
Thailand	TH
Timor-Leste	TL
Togo	TG
Tokelau	TK
Tonga	TO
Trinidad dan Tobago	TT
Tunisia	TN
Turki	TR
Turkmenistan	TM
Kepulauan Turks dan Caicos	TC
Tuvalu	TV
Uganda	UG
Ukraina	UA
Uni Emirat Arab	AE
Britania Raya	UK
Perserikatan Bangsa-Bangsa	UN
Amerika Serikat	US
Kepulauan Terluar Kecil Amerika Serikat	UM
Kepulauan Virgin Amerika Serikat	VI

Negara	Kode Negara
Uruguay	UY
Uzbekistan	UZ
Vanuatu	VU
Kota Vatikan	VA
Venezuela	VE
Vietnam	VN
Wallis dan Futuna	WF
Sahara Barat	EH
Yaman	YE
Zambia	ZM
Zimbabwe	ZW

Metrik obyektif

Autopilot menghasilkan metrik akurasi untuk mengevaluasi kandidat model dan membantu Anda memilih mana yang akan digunakan untuk menghasilkan perkiraan. Anda dapat membiarkan Autopilot mengoptimalkan prediktor untuk Anda, atau Anda dapat secara manual memilih algoritme untuk prediktor Anda. Secara default, Autopilot menggunakan Average Weighted Quantile Loss.

Daftar berikut berisi nama-nama metrik yang saat ini tersedia untuk mengukur kinerja model untuk peramalan deret waktu.

`RMSE`

Kesalahan kuadrat rata-rata akar (RMSE) - Mengukur akar kuadrat dari perbedaan kuadrat antara nilai prediksi dan aktual, dan dirata-ratakan pada semua nilai. Ini adalah metrik penting untuk menunjukkan adanya kesalahan model besar dan outlier. Nilai berkisar dari nol (0) hingga tak terhingga, dengan angka yang lebih kecil menunjukkan kecocokan model yang lebih baik dengan

data. RMSE tergantung pada skala, dan tidak boleh digunakan untuk membandingkan kumpulan data dengan ukuran yang berbeda.

wQL

Weighted Quantile Loss (wQL) — Menilai keakuratan perkiraan dengan mengukur perbedaan absolut tertimbang antara kuantil P10, P50, dan P90 yang diprediksi dan aktual dengan nilai yang lebih rendah yang menunjukkan kinerja yang lebih baik.

Average wQL (default)

Rata-rata Kehilangan Kuantil Tertimbang (Rata-rata wQL) - Mengevaluasi perkiraan dengan rata-rata akurasi pada kuantil P10, P50, dan P90. Nilai yang lebih rendah menunjukkan model yang lebih akurat.

MASE

Mean Absolute Scaled Error (MASE) — Kesalahan absolut rata-rata dari perkiraan dinormalisasi oleh kesalahan absolut rata-rata dari metode peramalan dasar sederhana. Nilai yang lebih rendah menunjukkan model yang lebih akurat, di mana MASE < 1 is estimated to be better than the baseline and MASE > 1 diperkirakan lebih buruk daripada baseline.

MAPE

Mean Absolute Percent Error (MAPE) — Persentase kesalahan (perbedaan persen dari nilai perkiraan rata-rata versus nilai aktual) dirata-ratakan pada semua titik waktu. Nilai yang lebih rendah menunjukkan model yang lebih akurat, di mana MAPE = 0 adalah model tanpa kesalahan.

WAPE

Weighted Absolute Percent Error (WAPE) — Jumlah kesalahan absolut dinormalisasi dengan jumlah target absolut, yang mengukur deviasi keseluruhan nilai yang diperkirakan dari nilai yang diamati. Nilai yang lebih rendah menunjukkan model yang lebih akurat.

Dukungan algoritma untuk peramalan deret waktu

Autopilot melatih enam algoritma bawaan berikut dengan deret waktu target Anda. Kemudian, dengan menggunakan metode ansambel susun, ini menggabungkan kandidat model ini untuk membuat model peramalan yang optimal untuk metrik objektif tertentu.

- Convolutional Neural Network - Quantile Regression (CNN-QR) - CNN-QR adalah algoritma pembelajaran mesin eksklusif untuk meramalkan deret waktu menggunakan jaringan saraf

konvolusional kausal (CNN). CNN-QR bekerja paling baik dengan kumpulan data besar yang berisi ratusan deret waktu.

- DeepAR+ — DeepAR+ adalah algoritma pembelajaran mesin eksklusif untuk memperkirakan deret waktu menggunakan jaringan saraf berulang (RNN). DeepAR+ bekerja paling baik dengan kumpulan data besar yang berisi ratusan deret waktu fitur.
- Nabi adalah model deret waktu struktural Bayesian lokal yang populer berdasarkan model aditif di mana tren non-linier cocok dengan musim tahunan, mingguan, dan harian. Algoritma Autopilot Prophet menggunakan kelas Nabi dari implementasi Python Nabi. Ini bekerja paling baik dengan deret waktu dengan efek musiman yang kuat dan beberapa musim data historis.
- Non-Parametric Time Series (NPTS) — Algoritma kepemilikan NPTS adalah peramal dasar probabilistik yang dapat diskalakan. Ini memprediksi distribusi nilai future dari deret waktu tertentu dengan mengambil sampel dari pengamatan sebelumnya. NPTS sangat berguna saat bekerja dengan deret waktu yang jarang atau intermiten.
- Autoregressive Integrated Moving Average (ARIMA) — ARIMA adalah algoritma statistik yang umum digunakan untuk peramalan deret waktu. Algoritma menangkap struktur temporal standar (organisasi berpola waktu) dalam dataset input. Ini sangat berguna untuk kumpulan data sederhana dengan deret waktu di bawah 100.
- Exponential Smoothing (ETS) — ETS adalah algoritma statistik yang umum digunakan untuk peramalan deret waktu. Algoritma ini sangat berguna untuk kumpulan data sederhana dengan deret waktu di bawah 100, dan kumpulan data dengan pola musiman. ETS menghitung rata-rata tertimbang atas semua pengamatan dalam kumpulan data deret waktu sebagai prediksinya, dengan bobot yang menurun secara eksponensial dari waktu ke waktu.

Penyebaran dan prakiraan model autopilot

Setelah Anda melatih prediktor Autopilot Anda (model terbaik), Anda dapat menerapkan model untuk mendapatkan prediksi dengan salah satu dari dua cara:

1. Gunakan Peramalan waktu nyata untuk mengatur titik akhir dan mendapatkan prediksi secara interaktif.
2. Gunakan Peramalan Batch untuk membuat prediksi secara paralel pada batch pengamatan pada seluruh kumpulan data.

Saat memberikan data input untuk peramalan, skema data Anda harus tetap sama dengan yang digunakan untuk melatih model Anda, termasuk jumlah kolom, header kolom, dan tipe data. Anda

dapat memperkirakan ID item yang ada atau baru dalam rentang waktu yang sama atau berbeda untuk memprediksi periode waktu yang berbeda.

Model peramalan memprediksi titik horizon perkiraan di masa depan yang ditentukan dalam permintaan input saat pelatihan, yaitu dari tanggal akhir target hingga tanggal akhir target+cakrawala perkiraan. Untuk menggunakan model untuk memprediksi tanggal tertentu, Anda harus memberikan data dalam format yang sama dengan data input asli, memperpanjang hingga tanggal akhir target yang ditentukan. Dalam skenario ini, model akan mulai memprediksi dari tanggal akhir target baru.

Misalnya, jika dataset Anda memiliki data bulanan dari Januari hingga Juni dengan horizon Forecast 2, maka model akan memprediksi nilai target untuk 2 bulan ke depan, yaitu Juli dan Agustus. Jika pada bulan Agustus, Anda ingin memprediksi untuk 2 bulan ke depan, kali ini data input Anda harus dari Januari hingga Agustus dan model akan memprediksi untuk 2 bulan ke depan (September, Oktober).

> **Note**
>
> Sebaiknya gunakan jenis contoh berikut untuk peramalan:
>
> - Untuk peramalan real-time, gunakan instance m5.12xlarge.
> - Untuk peramalan batch, gunakan instans m5.12xlarge untuk beban kerja tujuan umum dan instans m5.24xlarge untuk tugas peramalan data besar.

Peramalan waktu nyata

Anda dapat menggunakan peramalan real-time untuk beban kerja inferensi di mana Anda memiliki persyaratan real-time, interaktif, latensi rendah.

> **Note**
>
> Untuk peramalan waktu nyata, kumpulan data harus menjadi bagian dari kumpulan data input. Titik akhir waktu nyata memiliki ukuran data input sekitar 6MB dan batasan batas waktu respons 60 detik. Kami merekomendasikan membawa satu atau beberapa item sekaligus.

Anda dapat menggunakan SageMaker API untuk menerapkan model secara manual yang menghasilkan metrik validasi terbaik dalam eksperimen Autopilot sebagai berikut.

Atau, Anda dapat memilih opsi penerapan otomatis saat membuat eksperimen Autopilot Anda. Untuk informasi tentang pengaturan penerapan otomatis model, lihatCara mengaktifkan penyebaran otomatis.

1. Dapatkan definisi wadah kandidat

   Dapatkan definisi wadah kandidat dari InferenceContainers. Definisi kontainer untuk inferensi mengacu pada lingkungan kontainer yang dirancang untuk menerapkan dan menjalankan SageMaker model terlatih Anda untuk membuat prediksi.

   Contoh AWS CLI perintah berikut menggunakan API DescribeAutoMLJobv2 untuk mendapatkan definisi kandidat untuk kandidat model terbaik.

   ```
   aws sagemaker describe-auto-ml-job-v2 --auto-ml-job-name job-name --region region
   ```

2. Daftar kandidat

   Contoh AWS CLI perintah berikut menggunakan ListCandidatesForAutoMLJob API untuk daftar semua kandidat model.

   ```
   aws sagemaker list-candidates-for-auto-ml-job --auto-ml-job-name <job-name> --
   region <region>
   ```

3. Buat SageMaker model

   Gunakan definisi container dari langkah sebelumnya dan kandidat pilihan Anda untuk membuat SageMaker model dengan menggunakan CreateModelAPI. Lihat AWS CLI perintah berikut sebagai contoh.

   ```
   aws sagemaker create-model --model-name '<your-candidate-name>' \
                       --containers ['<container-definition1>, <container-
   definition2>, <container-definition3>]' \
                       --execution-role-arn '<execution-role-arn>' --region '<region>
   ```

4. Buat konfigurasi titik akhir

   Contoh AWS CLI perintah berikut menggunakan CreateEndpointConfigAPI untuk membuat konfigurasi endpoint.

   ```
   aws sagemaker create-endpoint-config --endpoint-config-name '<your-endpoint-config-
   name>' \
   ```

```
                    --production-variants '<list-of-production-variants>' \
                    --region '<region>'
```

5. Buat titik akhir

   AWS CLIContoh berikut menggunakan CreateEndpointAPI untuk membuat titik akhir.

```
aws sagemaker create-endpoint --endpoint-name '<your-endpoint-name>' \
                    --endpoint-config-name '<endpoint-config-name-you-just-created>'
 \
                    --region '<region>'
```

   Periksa kemajuan penerapan titik akhir Anda dengan menggunakan API. DescribeEndpoint Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-endpoint –endpoint-name '<endpoint-name>' –region <region>
```

   Setelah `EndpointStatus` perubahan`InService`, titik akhir siap digunakan untuk inferensi waktu nyata.

6. Memanggil titik akhir

   Struktur perintah berikut memanggil titik akhir untuk inferensi real-time.

```
aws sagemaker invoke-endpoint --endpoint-name '<endpoint-name>' \
                  --region '<region>' --body '<your-data-in-bytes>' [--content-type]
 '<content-type>' <outfile>
```

Peramalan Batch

Peramalan Batch, juga dikenal sebagai inferensi offline, menghasilkan prediksi model pada batch pengamatan. Inferensi Batch adalah pilihan yang baik untuk kumpulan data besar atau jika Anda tidak memerlukan respons langsung terhadap permintaan prediksi model

Sebaliknya, inferensi online (inferensi waktu nyata) menghasilkan prediksi secara real time.

Anda dapat membuat inferensi batch dari model Autopilot menggunakan referensi API.

Untuk menggunakan SageMaker API untuk inferensi batch:

1. Dapatkan definisi kandidat

Definisi kandidat dari InferenceContainersdigunakan untuk membuat SageMaker model.

Contoh berikut menunjukkan cara menggunakan API DescribeAutoMLJobv2 untuk mendapatkan definisi kandidat untuk kandidat model terbaik. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-auto-ml-job-v2 --auto-ml-job-name <job-name> --region <region>
```

Gunakan ListCandidatesForAutoMLJob API untuk mencantumkan semua kandidat. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker list-candidates-for-auto-ml-job --auto-ml-job-name <job-name> --
region <region>
```

2. Buat SageMaker model

Untuk membuat SageMaker model menggunakan CreateModelAPI, gunakan definisi container dari langkah sebelumnya. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker create-model --model-name '<your-custom-model-name>' \
                    --containers ['<container-definition1>, <container-
definition2>, <container-definition3>]' \
                    --execution-role-arn '<execution-role-arn>' --region '<region>
```

3. Buat pekerjaan SageMaker transformasi

Contoh berikut membuat pekerjaan SageMaker transformasi dengan CreateTransformJobAPI. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker create-transform-job --transform-job-name '<your-custom-transform-job-
name>' --model-name '<your-custom-model-name-from-last-step>'\
--transform-input '{
        "DataSource": {
            "S3DataSource": {
                "S3DataType": "S3Prefix",
                "S3Uri": "<your-input-data>"
            }
        },
        "ContentType": "text/csv",
        "SplitType": "None"
    }'\
--transform-output '{
```

```
        "S3OutputPath": "<your-output-path>",
        "AssembleWith": "Line"
    }'\
--transform-resources '{
        "InstanceType": "<instance-type>",
        "InstanceCount": 1
    }' --region '<region>'
```

Periksa kemajuan pekerjaan transformasi Anda menggunakan DescribeTransformJobAPI. Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-transform-job --transform-job-name '<your-custom-transform-job-name>' --region <region>
```

Setelah pekerjaan selesai, hasil yang diprediksi tersedia di`<your-output-path>`.

Nama file output memiliki format berikut:`<input_data_file_name>.out`. Sebagai contoh, jika file input Anda`text_x.csv`, nama output akan menjadi`text_x.csv.out`.

Tab berikut menunjukkan contoh kode untuk AWS SDK untuk Python (boto3), dan file. AWS CLI

AWS SDK for Python (boto3)

Contoh berikut menggunakan AWSSDK untuk Python (boto3) untuk membuat prediksi dalam batch.

```
import sagemaker
import boto3

session = sagemaker.session.Session()

sm_client = boto3.client('sagemaker', region_name='us-west-2')
role = 'arn:aws:iam::1234567890:role/sagemaker-execution-role'
output_path = 's3://test-auto-ml-job/output'
input_data = 's3://test-auto-ml-job/test_X.csv'

best_candidate = sm_client.describe_auto_ml_job_v2(AutoMLJobName=job_name)
['BestCandidate']
best_candidate_containers = best_candidate['InferenceContainers']
best_candidate_name = best_candidate['CandidateName']
```

```
# create model
reponse = sm_client.create_model(
    ModelName = best_candidate_name,
    ExecutionRoleArn = role,
    Containers = best_candidate_containers
)

# Lauch Transform Job
response = sm_client.create_transform_job(
    TransformJobName=f'{best_candidate_name}-transform-job',
    ModelName=model_name,
    TransformInput={
        'DataSource': {
            'S3DataSource': {
                'S3DataType': 'S3Prefix',
                'S3Uri': input_data
            }
        },
        'ContentType': "text/csv",
        'SplitType': 'None'
    },
    TransformOutput={
        'S3OutputPath': output_path,
        'AssembleWith': 'Line',
    },
    TransformResources={
        'InstanceType': 'ml.m5.2xlarge',
        'InstanceCount': 1,
    },
)
```

Pekerjaan inferensi batch mengembalikan respons dalam format berikut.

```
{'TransformJobArn': 'arn:aws:sagemaker:us-west-2:1234567890:transform-job/test-
transform-job',
 'ResponseMetadata': {'RequestId': '659f97fc-28c4-440b-b957-a49733f7c2f2',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '659f97fc-28c4-440b-b957-a49733f7c2f2',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '96',
   'date': 'Thu, 11 Aug 2022 22:23:49 GMT'},
  'RetryAttempts': 0}}
```

AWS Command Line Interface (AWS CLI)

1. Dapatkan definisi kandidat dengan menggunakan contoh kode berikut.

```
aws sagemaker describe-auto-ml-job-v2 --auto-ml-job-name 'test-automl-job' --
region us-west-2
```

2. Buat model dengan menggunakan contoh kode berikut.

```
aws sagemaker create-model --model-name 'test-sagemaker-model'
--containers '[{
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-sklearn-
automl:2.5-1-cpu-py3",
    "ModelDataUrl": "s3://test-bucket/out/test-job1/data-processor-models/test-
job1-dpp0-1-e569ff7ad77f4e55a7e549a/output/model.tar.gz",
    "Environment": {
        "AUTOML_SPARSE_ENCODE_RECORDIO_PROTOBUF": "1",
        "AUTOML_TRANSFORM_MODE": "feature-transform",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "application/x-recordio-protobuf",
        "SAGEMAKER_PROGRAM": "sagemaker_serve",
        "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
    }
}, {
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
xgboost:1.3-1-cpu-py3",
    "ModelDataUrl": "s3://test-bucket/out/test-job1/tuning/flicdf10v2-dpp0-xgb/
test-job1E9-244-7490a1c0/output/model.tar.gz",
    "Environment": {
        "MAX_CONTENT_LENGTH": "20971520",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
        "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_SUPPORTED":
 "predicted_label,probability,probabilities"
    }
}, {
    "Image": "348316444620.dkr.ecr.us-west-2.amazonaws.com/sagemaker-sklearn-
automl:2.5-1-cpu-py3",
    "ModelDataUrl": "s3://test-bucket/out/test-job1/data-processor-models/test-
job1-dpp0-1-e569ff7ad77f4e55a7e549a/output/model.tar.gz",
    "Environment": {
        "AUTOML_TRANSFORM_MODE": "inverse-label-transform",
        "SAGEMAKER_DEFAULT_INVOCATIONS_ACCEPT": "text/csv",
        "SAGEMAKER_INFERENCE_INPUT": "predicted_label",
```

```
        "SAGEMAKER_INFERENCE_OUTPUT": "predicted_label",
        "SAGEMAKER_INFERENCE_SUPPORTED":
 "predicted_label,probability,labels,probabilities",
        "SAGEMAKER_PROGRAM": "sagemaker_serve",
        "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code"
    }
}]' \
--execution-role-arn 'arn:aws:iam::1234567890:role/sagemaker-execution-role' \
--region 'us-west-2'
```

3. Buat pekerjaan transformasi dengan menggunakan contoh kode berikut.

```
aws sagemaker create-transform-job --transform-job-name 'test-tranform-job'\
 --model-name 'test-sagemaker-model'\
--transform-input '{
        "DataSource": {
            "S3DataSource": {
                "S3DataType": "S3Prefix",
                "S3Uri": "s3://test-bucket/data.csv"
            }
        },
        "ContentType": "text/csv",
        "SplitType": "None"
    }'\
--transform-output '{
        "S3OutputPath": "s3://test-bucket/output/",
        "AssembleWith": "Line"
    }'\
--transform-resources '{
        "InstanceType": "ml.m5.2xlarge",
        "InstanceCount": 1
    }'\
--region 'us-west-2'
```

4. Periksa kemajuan pekerjaan transformasi dengan menggunakan contoh kode berikut.

```
aws sagemaker describe-transform-job --transform-job-name  'test-tranform-job' --
region us-west-2
```

Berikut ini adalah respons dari pekerjaan transformasi.

```
{
    "TransformJobName": "test-tranform-job",
```

```
    "TransformJobArn": "arn:aws:sagemaker:us-west-2:1234567890:transform-job/test-
tranform-job",
    "TransformJobStatus": "InProgress",
    "ModelName": "test-model",
    "TransformInput": {
        "DataSource": {
            "S3DataSource": {
                "S3DataType": "S3Prefix",
                "S3Uri": "s3://test-bucket/data.csv"
            }
        },
        "ContentType": "text/csv",
        "CompressionType": "None",
        "SplitType": "None"
    },
    "TransformOutput": {
        "S3OutputPath": "s3://test-bucket/output/",
        "AssembleWith": "Line",
        "KmsKeyId": ""
    },
    "TransformResources": {
        "InstanceType": "ml.m5.2xlarge",
        "InstanceCount": 1
    },
    "CreationTime": 1662495635.679,
    "TransformStartTime": 1662495847.496,
    "DataProcessing": {
        "InputFilter": "$",
        "OutputFilter": "$",
        "JoinSource": "None"
    }
}
```

Setelah `TransformJobStatus` perubahan`Completed`, Anda dapat memeriksa hasil inferensi di. `S3OutputPath`

Notebook eksplorasi data Amazon SageMaker Autopilot

Amazon SageMaker Autopilot membersihkan dan memproses kumpulan data Anda secara otomatis. Untuk membantu pengguna memahami data mereka, mengungkap pola, hubungan, dan anomali tentang deret waktu, SageMaker Amazon Autopilot menghasilkan laporan statis eksplorasi data dalam bentuk buku catatan untuk referensi pengguna.

Notebook eksplorasi data dihasilkan untuk setiap pekerjaan Autopilot. Laporan disimpan dalam bucket Amazon S3 dan dapat diakses dari jalur output pekerjaan.

Anda dapat menemukan awalan Amazon S3 ke notebook eksplorasi data dalam respons terhadap at. `DescribeAutoMLJobV2` `AutoMLJobArtifacts.DataExplorationNotebookLocation`

Laporan yang dihasilkan oleh Amazon SageMaker Autopilot

Selain notebook eksplorasi data, Autopilot menghasilkan berbagai laporan untuk kandidat model terbaik dari setiap percobaan.

- Laporan penjelasan memberikan wawasan tentang bagaimana model membuat prakiraan.
- Laporan kinerja memberikan penilaian kuantitatif kemampuan peramalan model.
- Laporan hasil backtest dihasilkan setelah menguji kinerja model pada data historis.

Laporan penjelasan

Laporan penjelasan autopilot membantu Anda lebih memahami bagaimana atribut dalam kumpulan data memengaruhi perkiraan untuk deret waktu tertentu (kombinasi item dan dimensi) dan titik waktu. Autopilot menggunakan metrik yang disebut Skor dampak untuk mengukur dampak relatif dari setiap atribut dan menentukan apakah mereka meningkatkan atau menurunkan nilai perkiraan.

Misalnya, pertimbangkan skenario peramalan di mana target berada `sales` dan ada dua atribut terkait: `price` dan`color`. Autopilot mungkin menemukan bahwa warna item memiliki dampak tinggi pada penjualan untuk item tertentu, tetapi efek yang dapat diabaikan untuk item lainnya. Mungkin juga menemukan bahwa promosi di musim panas memiliki dampak tinggi pada penjualan, tetapi promosi di musim dingin memiliki sedikit efek.

Laporan penjelasan dibuat hanya jika:

- Kumpulan data deret waktu mencakup kolom fitur tambahan atau dikaitkan dengan kalender liburan.
- Model dasar CNN-QR dan DeepAR+ termasuk dalam ansambel akhir.

Menafsirkan skor Dampak

Skor dampak mengukur atribut dampak relatif terhadap nilai perkiraan. Misalnya, jika `price` atribut memiliki skor dampak yang dua kali lebih besar dari `store location` atribut, Anda dapat

menyimpulkan bahwa harga item memiliki dampak dua kali lipat pada nilai perkiraan daripada lokasi penyimpanan.

Skor dampak juga memberikan informasi tentang apakah atribut meningkatkan atau menurunkan nilai perkiraan.

Skor Dampak berkisar dari -1 hingga 1, di mana tanda menunjukkan arah tumbukan. Skor 0 menunjukkan tidak ada dampak, sedangkan skor mendekati 1 atau -1 menunjukkan dampak yang signifikan.

Penting untuk dicatat bahwa Skor dampak mengukur dampak relatif atribut, bukan dampak absolut. Oleh karena itu, skor Dampak tidak dapat digunakan untuk menentukan apakah atribut tertentu meningkatkan akurasi model. Jika atribut memiliki skor Dampak rendah, itu tidak berarti bahwa ia memiliki dampak rendah pada nilai perkiraan; itu berarti bahwa ia memiliki dampak yang lebih rendah pada nilai perkiraan daripada atribut lain yang digunakan oleh prediktor.

Temukan laporan penjelasan

Anda dapat menemukan awalan Amazon S3 untuk artefak penjelasan yang dihasilkan untuk kandidat terbaik dalam menanggapi at. `DescribeAutoMLJobV2` `BestCandidate.CandidateProperties.CandidateArtifactLocations.Explainability`

Laporan kinerja model

Laporan kualitas model autopilot (juga disebut sebagai laporan kinerja) memberikan wawasan dan informasi kualitas untuk kandidat model terbaik (prediktor terbaik) yang dihasilkan oleh pekerjaan AutoML. Ini termasuk informasi tentang detail pekerjaan, fungsi objektif, dan metrik akurasi (`wQL`,,`MAPE`, `WAPERMSE`,`MASE`).

Anda dapat menemukan awalan Amazon S3 untuk artefak laporan kualitas model yang dihasilkan untuk kandidat terbaik dalam menanggapi at. `DescribeAutoMLJobV2` `BestCandidate.CandidateProperties.CandidateArtifactLocations.ModelInsights`

Laporan hasil backtests

Hasil backtests memberikan wawasan tentang kinerja model peramalan deret waktu dengan mengevaluasi akurasi dan keandalan prediktifnya. Ini membantu analis dan ilmuwan data menilai kinerjanya pada data historis dan membantu memahami potensi kinerjanya pada data masa depan yang tidak terlihat.

Autopilot menggunakan backtesting untuk menyetel parameter dan menghasilkan metrik akurasi. Selama backtesting, Autopilot secara otomatis membagi data deret waktu Anda menjadi dua set, satu set pelatihan dan satu set pengujian. Set pelatihan digunakan untuk melatih model yang kemudian digunakan untuk menghasilkan perkiraan untuk titik data dalam set pengujian. Autopilot menggunakan dataset pengujian ini untuk mengevaluasi akurasi model dengan membandingkan nilai yang diperkirakan dengan nilai yang diamati dalam set pengujian.

Anda dapat menemukan awalan Amazon S3 untuk artefak laporan kualitas model yang dihasilkan untuk kandidat terbaik dalam menanggapi at. `DescribeAutoMLJobV2` `BestCandidate.CandidateProperties.CandidateArtifactLocations.BacktestResults`

Batas sumber daya peramalan deret waktu Amazon SageMaker Autopilot

Batas sumber daya	Batas default	Dapat disesuaikan
Ukuran dataset masukan	30 GB	Ya
Ukuran satu file Parket	2 GB	Tidak
Jumlah maksimum baris dalam set data	3 miliar	Ya
Jumlah maksimum kolom pengelompokan	5	Tidak
Jumlah maksimum fitur numerik	13	Tidak
Jumlah maksimum fitur kategoris	10	Tidak
Jumlah maksimum deret waktu (kombinasi unik item dan kolom pengelompokan) per kumpulan data	5.000.000	Ya
Cakrawala Forecast Maksimum	500	Ya

Buat pekerjaan AutoML untuk menyempurnakan model pembuatan teks menggunakan API

Model bahasa besar (LLM) unggul dalam beberapa tugas generatif, termasuk pembuatan teks, ringkasan, penyelesaian, menjawab pertanyaan, dan banyak lagi. Kinerja mereka dapat dikaitkan dengan ukuran signifikan dan pelatihan ekstensif pada kumpulan data yang beragam dan berbagai tugas. Namun, domain tertentu, seperti layanan kesehatan dan keuangan, mungkin memerlukan penyesuaian khusus untuk beradaptasi dengan data unik dan kasus penggunaan. Dengan menyesuaikan pelatihan mereka dengan domain khusus mereka, LLM dapat meningkatkan kinerja mereka dan memberikan output yang lebih akurat untuk aplikasi yang ditargetkan.

Autopilot menawarkan kemampuan untuk menyempurnakan pilihan model teks generatif yang telah dilatih sebelumnya. Secara khusus, Autopilot mendukung fine tuning berbasis instruksi dari pilihan model bahasa besar tujuan umum (LLM) yang didukung oleh. SageMaker JumpStart

> **Note**
>
> Model pembuatan teks yang mendukung fine-tuning di Autopilot saat ini dapat diakses secara eksklusif di Wilayah yang didukung oleh Canvas. SageMaker Lihat dokumentasi SageMaker Canvas untuk daftar lengkap Wilayah yang didukung.

Penyetelan model yang telah dilatih sebelumnya memerlukan kumpulan data spesifik dari instruksi yang jelas yang memandu model tentang cara menghasilkan output atau berperilaku untuk tugas itu. Model belajar dari kumpulan data, menyesuaikan parameternya agar sesuai dengan instruksi yang diberikan. Penyetelan berbasis instruksi melibatkan penggunaan contoh berlabel yang diformat sebagai pasangan respons prompt dan diungkapkan sebagai instruksi. Untuk informasi selengkapnya tentang fine-tuning, lihat Menyempurnakan model foundation.

Pedoman berikut menguraikan proses pembuatan pekerjaan Amazon SageMaker Autopilot sebagai percobaan percontohan untuk menyempurnakan LLM pembuatan teks menggunakan Referensi API. SageMaker

> **Note**
>
> Tugas seperti klasifikasi teks dan gambar, peramalan deret waktu, dan penyempurnaan model bahasa besar tersedia secara eksklusif melalui API Autopilot versi 2. Untuk pengguna

Python, sebaiknya gunakan AWS SDK for Python (Boto3)sebagai Amazon SageMaker Python SDK saat ini tidak didukung untuk Autopilot API versi 2.
Pengguna yang lebih menyukai kenyamanan antarmuka pengguna dapat menggunakan Amazon SageMaker Canvas untuk mengakses model pra-terlatih dan model dasar AI generatif, atau membuat model khusus yang disesuaikan untuk teks tertentu, klasifikasi gambar, kebutuhan peramalan, atau AI generatif.

Untuk membuat eksperimen Autopilot secara terprogram untuk menyempurnakan LLM, Anda dapat memanggil API `CreateAutoMLJobV2`dalam bahasa apa pun yang didukung oleh Amazon Autopilot atau. SageMaker AWS CLI

Untuk informasi tentang cara tindakan API ini diterjemahkan ke dalam fungsi dalam bahasa pilihan Anda, lihat bagian Lihat Juga `CreateAutoMLJobV2` dan pilih SDK. Sebagai contoh, untuk pengguna Python, lihat sintaks permintaan lengkap dari in. `create_auto_ml_job_v2` AWS SDK for Python (Boto3)

> **Note**
>
> Autopilot menyempurnakan model bahasa besar tanpa memerlukan banyak kandidat untuk dilatih dan dievaluasi. Alih-alih, menggunakan kumpulan data Anda, Autopilot secara langsung menyempurnakan model target Anda untuk meningkatkan metrik objektif default, kehilangan lintas entropi. Model bahasa fine-tuning di Autopilot tidak memerlukan pengaturan bidang. `AutoMLJobObjective`

Setelah LLM Anda disetel dengan baik, Anda dapat mengevaluasi kinerjanya dengan mengakses berbagai skor ROUGE melalui saat melakukan panggilan API. `BestCandidate` `DescribeAutoMLJobV2` Model ini juga memberikan informasi tentang pelatihan dan kehilangan validasi serta kebingungannya. Untuk daftar lengkap metrik untuk mengevaluasi kualitas teks yang dihasilkan oleh model yang disetel dengan baik, lihat. Metrik untuk menyempurnakan model bahasa besar di Autopilot

Prasyarat

Sebelum menggunakan Autopilot untuk membuat eksperimen fine-tuning SageMaker, pastikan untuk mengambil langkah-langkah berikut:

- (Opsional) Pilih model pra-terlatih yang ingin Anda sesuaikan.

Untuk daftar model pra-terlatih yang tersedia untuk fine-tuning di SageMaker Amazon Autopilot, lihat. Model bahasa besar yang didukung untuk fine-tuning Pemilihan model tidak wajib; jika tidak ada model yang ditentukan, Autopilot secara otomatis default ke model Falcon7bInstruct.

- Buat kumpulan data instruksi. Lihat Jenis file dataset dan format data input untuk mempelajari tentang persyaratan format untuk kumpulan data berbasis instruksi Anda.
- Tempatkan kumpulan data Anda di bucket Amazon S3.
- Berikan akses penuh ke bucket Amazon S3 yang berisi data masukan untuk peran SageMaker eksekusi yang digunakan untuk menjalankan eksperimen.
  - Untuk informasi tentang mengambil peran SageMaker eksekusi Anda, lihatDapatkan peran eksekusi.
  - Untuk informasi tentang pemberian izin peran SageMaker eksekusi untuk mengakses satu atau beberapa bucket tertentu di Amazon S3, lihat Menambahkan Izin Amazon S3 Tambahan ke Peran Eksekusi di. SageMaker Buat peran eksekusi
- Selain itu, Anda harus memberi peran eksekusi izin yang diperlukan untuk mengakses bucket Amazon S3 penyimpanan default yang digunakan oleh. SageMaker JumpStart Akses ini diperlukan untuk menyimpan dan mengambil artefak model pra-terlatih di. SageMaker JumpStart Untuk memberikan akses ke bucket Amazon S3 ini, Anda harus membuat kebijakan kustom inline baru pada peran eksekusi Anda.

  Berikut adalah contoh kebijakan yang dapat Anda gunakan di editor JSON Anda saat mengonfigurasi pekerjaan fine-tuning AutoML di: `us-west-2`

  SageMaker JumpStartnama bucket mengikuti pola yang telah ditentukan yang bergantung pada. Wilayah AWS Anda harus menyesuaikan nama ember yang sesuai.

```
{
    "Sid": "Statement1",
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::jumpstart-cache-prod-us-west-2",
        "arn:aws:s3:::jumpstart-cache-prod-us-west-2/*"
    ]
```

```
}
```

Setelah ini selesai, Anda dapat menggunakan ARN dari peran eksekusi ini dalam permintaan API Autopilot.

Parameter yang diperlukan

Saat menelepon `CreateAutoMLJobV2` untuk membuat eksperimen Autopilot untuk fine-tuning LLM, Anda harus memberikan nilai berikut:

- An `AutoMLJobName` untuk menentukan nama pekerjaan Anda. Nama harus bertipe`string`, dan harus memiliki panjang minimal 1 karakter dan panjang maksimum 32.
- Setidaknya `AutoMLJobChannel` salah satu `training` tipe di dalam`AutoMLJobInputDataConfig`. Saluran ini menentukan nama bucket Amazon S3 tempat dataset fine-tuning Anda berada. Anda memiliki opsi untuk menentukan `validation` saluran. Jika tidak ada saluran validasi yang disediakan, dan a `ValidationFraction` dikonfigurasi di `AutoMLDataSplitConfig`, fraksi ini digunakan untuk membagi kumpulan data pelatihan secara acak menjadi set pelatihan dan validasi. Selain itu, Anda dapat menentukan jenis konten (file CSV atau Parket) untuk kumpulan data.
- `AutoMLProblemTypeConfig`Jenis `TextGenerationJobConfig` untuk mengonfigurasi pengaturan pekerjaan pelatihan Anda.

  Secara khusus, Anda dapat menentukan nama model dasar untuk menyempurnakan bidang. `BaseModelName` Untuk daftar model pra-terlatih yang tersedia untuk fine-tuning di SageMaker Amazon Autopilot, lihat. Model bahasa besar yang didukung untuk fine-tuning
- `OutputDataConfig`Untuk menentukan jalur keluaran Amazon S3 untuk menyimpan artefak pekerjaan AutoML Anda.
- A `RoleArn` untuk menentukan ARN dari peran yang digunakan untuk mengakses data Anda.

Berikut ini adalah contoh format permintaan lengkap yang digunakan saat membuat panggilan API `CreateAutoMLJobV2` untuk menyempurnakan model ()`Falcon7BInstruct`.

```
{
   "AutoMLJobName": "<job_name>",
   "AutoMLJobInputDataConfig": [
      {
         "ChannelType": "training",
```

```
         "CompressionType": "None",
         "ContentType": "text/csv",
         "DataSource": {
            "S3DataSource": {
               "S3DataType": "S3Prefix",
               "S3Uri": "s3://<bucket_name>/<input_data>.csv"
            }
         }
      }
   ],
  "OutputDataConfig": {
      "S3OutputPath": "s3://<bucket_name>/output",
      "KmsKeyId": "arn:aws:kms:<region>:<account_id>:key/<key_value>"
   },
   "RoleArn":"arn:aws:iam::<account_id>:role/<sagemaker_execution_role_name>",
   "AutoMLProblemTypeConfig": {
        "TextGenerationJobConfig": {
            "BaseModelName": "Falcon7BInstruct"
        }
   }
}
```

Semua parameter lainnya adalah opsional.

Parameter opsional

Bagian berikut memberikan rincian beberapa parameter opsional yang dapat Anda berikan ke pekerjaan AutoML fine-tuning Anda.

Cara menentukan kumpulan data pelatihan dan validasi pekerjaan AutoML

Anda dapat memberikan kumpulan data validasi dan rasio pemisahan data khusus Anda sendiri, atau membiarkan Autopilot membagi kumpulan data secara otomatis.

Setiap `AutoMLJobChannel` objek (lihat parameter AutoML yang diperlukanJobInputDataConfig) memilikiChannelType, yang dapat diatur ke salah satu `training` atau `validation` nilai yang menentukan bagaimana data akan digunakan saat membangun model pembelajaran mesin.

Setidaknya satu sumber data harus disediakan dan maksimal dua sumber data diperbolehkan: satu untuk data pelatihan dan satu untuk data validasi. Bagaimana Anda membagi data menjadi kumpulan data pelatihan dan validasi tergantung pada apakah Anda memiliki satu atau dua sumber data.

- Jika Anda hanya memiliki satu sumber data, `ChannelType` diatur ke secara `training` default dan harus memiliki nilai ini.
  - Jika `ValidationFraction` nilai dalam tidak `AutoMLDataSplitConfig`disetel, 0.2 (20%) data dari sumber ini digunakan untuk validasi secara default.
  - Jika `ValidationFraction` diatur ke nilai antara 0 dan 1, dataset dibagi berdasarkan nilai yang ditentukan, di mana nilai menentukan fraksi dari dataset yang digunakan untuk validasi.
- Jika Anda memiliki dua sumber data, `ChannelType` salah satu `AutoMLJobChannel` objek harus diatur ke`training`, nilai default. Sumber data lainnya harus diatur ke`validation`. `ChannelType` Kedua sumber data harus memiliki format yang sama, baik CSV atau Parket, dan skema yang sama. Anda tidak boleh menetapkan nilai untuk `ValidationFraction` dalam kasus ini karena semua data dari setiap sumber digunakan untuk pelatihan atau validasi. Menyetel nilai ini menyebabkan kesalahan.

Cara mengaktifkan penyebaran otomatis

Dengan Autopilot, Anda dapat secara otomatis menerapkan model fine-tuned Anda ke titik akhir. Untuk mengaktifkan penerapan otomatis untuk model yang disetel dengan baik, sertakan a dalam permintaan pekerjaan `ModelDeployConfig` AutoML. Hal ini memungkinkan penerapan model fine-tuned Anda ke titik akhir. SageMaker Di bawah ini adalah konfigurasi yang tersedia untuk kustomisasi.

- Untuk membiarkan Autopilot menghasilkan nama titik akhir, setel ke. `AutoGenerateEndpointName` `True`
- Untuk memberikan nama Anda sendiri untuk titik akhir, atur`AutoGenerateEndpointName` `to False and provide a name of your choice in` `EndpointName`.

Cara mengatur penerimaan EULA saat menyempurnakan model menggunakan AutoML API

Untuk model yang memerlukan penerimaan perjanjian lisensi pengguna akhir sebelum fine-tuning, Anda dapat menerima EULA dengan menyetel `AcceptEula` atribut to in saat mengonfigurasi. `ModelAccessConfig` `True` `TextGenerationJobConfig` `AutoMLProblemTypeConfig`

Cara mengatur hyperparameters untuk mengoptimalkan proses pembelajaran suatu model

Anda dapat mengoptimalkan proses pembelajaran model pembuatan teks Anda dengan menyetel nilai hyperparameter dalam `TextGenerationHyperParameters` atribut `TextGenerationJobConfig` saat mengonfigurasi. `AutoMLProblemTypeConfig`

Autopilot memungkinkan pengaturan empat hiperparameter umum di semua model.

- `epochCount`: Nilainya harus berupa string yang berisi nilai integer dalam kisaran `1` to`10`.
- `batchSize`: Nilainya harus berupa string yang berisi nilai integer dalam kisaran `1` to`64`.
- `learningRate`: Nilainya harus berupa string yang berisi nilai floating-point dalam kisaran to. `0` `1`
- `learningRateWarmupSteps`: Nilainya harus berupa string yang berisi nilai integer dalam kisaran `0` to`250`.

Untuk detail lebih lanjut tentang setiap hyperparameter, lihatOptimalkan proses pembelajaran model pembuatan teks Anda dengan hyperparameters.

Contoh JSON berikut menunjukkan `TextGenerationHyperParameters` bidang yang diteruskan ke TextGenerationJobConfig tempat keempat hyperparameters dikonfigurasi.

```
"AutoMLProblemTypeConfig": {
  "TextGenerationJobConfig": {
    "BaseModelName": "Falcon7B",
    "TextGenerationHyperParameters": {"epochCount":"5", "learningRate":"0.000001",
 "batchSize": "32", "learningRateWarmupSteps": "10"}
  }
}
```

Model bahasa besar yang didukung untuk fine-tuning

Menggunakan Autopilot API, pengguna dapat menyempurnakan model bahasa besar (LLM) berikut. Model-model tersebut didukung oleh Amazon SageMaker JumpStart.

> **Note**
>
> Untuk model fine-tuning yang memerlukan penerimaan perjanjian lisensi pengguna akhir, Anda harus secara eksplisit menyatakan penerimaan EULA saat membuat pekerjaan AutoML Anda. Perhatikan bahwa setelah menyempurnakan model yang telah dilatih sebelumnya, bobot model asli diubah, sehingga Anda tidak perlu menerima EULA nanti saat menerapkan model yang disetel dengan baik.
> Untuk informasi tentang cara menerima EULA saat membuat pekerjaan fine-tuning menggunakan AutoML API, lihat. the section called “Set EULA”

Anda dapat menemukan detail lengkap setiap model dengan mencari ID SageMaker JumpStart Model Anda di tabel model berikut, lalu mengikuti tautan di kolom Sumber. Detail ini mungkin termasuk bahasa yang didukung oleh model, bias yang mungkin ditunjukkannya, kumpulan data yang digunakan untuk fine-tuning, dan banyak lagi.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-textgeneration-dolly-v2-3b-bf16	`Dolly3B`	Dolly 3B adalah 2,8 miliar parameter instruksi mengikuti model bahasa besar berdasarkan pythia-2.8b. Ini dilatih pada petunjuk/respons fine tuning dataset databricks-dolly-15k dan dapat melakukan tugas-tugas termasuk brainstorming, klasifikasi, pertanyaan dan jawaban, pembuatan teks, ekstraksi informasi, dan peringkasan.
huggingface-textgeneration-dolly-v2-7b-bf16	`Dolly7B`	Dolly 7B adalah model bahasa besar mengikuti instruksi parameter 6,9 miliar berdasarkan pythia-6.9b. Ini dilatih pada petunjuk/respons fine tuning dataset databricks-dolly-15k dan dapat melakukan tugas-tugas termasuk brainstorming, klasifikasi, pertanyaan dan jawaban, pembuatan teks, ekstraksi informasi, dan peringkasan.
huggingface-textgeneration-dolly-v2-12b-bf16	`Dolly12B`	Dolly 12B adalah 12 miliar parameter instruksi mengikuti

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
		model bahasa besar berdasarkan pythia-12b. Ini dilatih pada petunjuk/ respons fine tuning dataset databricks-dolly-15k dan dapat melakukan tugas-tugas termasuk brainstorming, klasifikasi, pertanyaan dan jawaban, pembuatan teks, ekstraksi informasi, dan peringkasan.
huggingface-llm-falcon-7b-bf16	`Falcon7B`	Falcon 7B adalah model bahasa besar kausal parameter 7 miliar yang dilatih pada 1.500 miliar token yang ditingkatkan dengan corpora yang dikuratori. Falcon-7B dilatih hanya pada data bahasa Inggris dan Prancis, dan tidak menggeneralisasi dengan tepat ke bahasa lain. Karena model ini dilatih pada sejumlah besar data web, ia membawa stereotip dan bias yang biasa ditemukan secara online.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-llm-falcon-7b-instruct-bf16	`Falcon7BInstruct`	Falcon 7B Instruct adalah model bahasa besar kausal parameter 7 miliar yang dibangun di atas Falcon 7B dan disetel dengan baik pada campuran 250 juta token dari kumpulan data obrolan/instruksi. Falcon 7B Instruct sebagian besar dilatih pada data bahasa Inggris, dan tidak menggeneralisasi dengan tepat ke bahasa lain. Selain itu, karena dilatih pada perwakilan korpora skala besar dari web, ia membawa stereotip dan bias yang biasa ditemui secara online.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-llm-falcon-40b-bf16	`Falcon40B`	Falcon 40B adalah model bahasa besar kausal parameter 40 miliar yang dilatih pada 1.000 miliar token yang ditingkatkan dengan corpora yang dikuratori. Ini dilatih sebagian besar dalam bahasa Inggris, Jerman, Spanyol, dan Prancis, dengan kemampuan terbatas dalam bahasa Italia, Portugis, Polandia, Belanda, Rumania, Ceko, dan Swedia. Itu tidak menggeneralisasi dengan tepat ke bahasa lain. Selain itu, karena dilatih pada perwakilan korpora skala besar dari web, ia membawa stereotip dan bias yang biasa ditemui secara online.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-llm-falcon-40b-instruct-bf16	`Falcon40BInstruct`	Falcon 40B Instruct adalah model bahasa besar kausal parameter 40 miliar yang dibangun di atas Falcon40B dan disetel dengan baik pada campuran Baize. Ini sebagian besar dilatih pada data bahasa Inggris dan Prancis, dan tidak menggeneralisasi dengan tepat ke bahasa lain. Selain itu, karena dilatih pada perwakilan korpora skala besar dari web, ia membawa stereotip dan bias yang biasa ditemui secara online.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-text2text-flan-t5-large	`FlanT5L`	Keluarga model Flan-T5 adalah seperangkat model bahasa besar yang disetel dengan baik pada banyak tugas dan dapat dilatih lebih lanjut. Model-model ini sangat cocok untuk tugas-tugas seperti terjemahan bahasa, pembuatan teks, penyelesaian kalimat, disambiguasi pengertian kata, ringkasan, atau menjawab pertanyaan. Flan T5 L adalah 780 juta parameter model bahasa besar yang dilatih pada berbagai bahasa. Anda dapat menemukan daftar bahasa yang didukung oleh Flan T5 L dalam detail model yang diambil dari pencarian Anda berdasarkan ID model di SageMaker JumpStart tabel model.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-text2text-flan-t5-xl	`FlanT5XL`	Keluarga model Flan-T5 adalah seperangkat model bahasa besar yang disetel dengan baik pada banyak tugas dan dapat dilatih lebih lanjut. Model-model ini sangat cocok untuk tugas-tugas seperti terjemahan bahasa, pembuatan teks, penyelesaian kalimat, disambiguasi pengertian kata, ringkasan, atau menjawab pertanyaan. Flan T5 XL adalah model bahasa besar 3 miliar parameter yang dilatih pada berbagai bahasa. Anda dapat menemukan daftar bahasa yang didukung oleh Flan T5 XL dalam detail model yang diambil dari pencarian Anda berdasarkan ID model di SageMaker JumpStart tabel model.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-text2text-flan-t5-xxll	`FlanT5XXL`	Keluarga model Flan-T5 adalah seperangkat model bahasa besar yang disetel dengan baik pada banyak tugas dan dapat dilatih lebih lanjut. Model-model ini sangat cocok untuk tugas-tugas seperti terjemahan bahasa, pembuatan teks, penyelesaian kalimat, disambiguasi pengertian kata, ringkasan, atau menjawab pertanyaan. Flan T5 XXL adalah model parameter 11 miliar. Anda dapat menemukan daftar bahasa yang didukung oleh Flan T5 XXL dalam detail model yang diambil dari pencarian Anda berdasarkan ID model di SageMaker JumpStart tabel model.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
meta-textgeneration-llama-2-7b	`Llama2-7B`	Llama 2 adalah kumpulan model teks generatif yang telah dilatih dan disetel dengan baik, mulai dari 7 miliar hingga 70 miliar parameter. Llama2-7B adalah model parameter 7 miliar yang ditujukan untuk penggunaan bahasa Inggris dan dapat disesuaikan untuk berbagai tugas pembuatan bahasa alami.
meta-textgeneration-llama-2-7b-f	`Llama2-7BChat`	Llama 2 adalah kumpulan model teks generatif yang telah dilatih dan disetel dengan baik, mulai dari 7 miliar hingga 70 miliar parameter. Llama2-7B adalah model obrolan parameter 7 miliar yang dioptimalkan untuk kasus penggunaan dialog.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
meta-textgeneration-llama-2-13b	`Llama2-13B`	Llama 2 adalah kumpulan model teks generatif yang telah dilatih dan disetel dengan baik, mulai dari 7 miliar hingga 70 miliar parameter. Llama2-13B adalah model parameter 13 miliar yang ditujukan untuk penggunaan bahasa Inggris dan dapat disesuaikan untuk berbagai tugas pembuatan bahasa alami.
meta-textgeneration-llama-2-13b-f	`Llama2-13BChat`	Llama 2 adalah kumpulan model teks generatif yang telah dilatih dan disetel dengan baik, mulai dari 7 miliar hingga 70 miliar parameter. Llama2-13 B adalah model obrolan parameter 13 miliar yang dioptimalkan untuk kasus penggunaan dialog.
huggingface-llm-mistral-7b	`Mistral7B`	Mistral 7B adalah kode tujuh miliar parameter dan model pembuatan teks bahasa Inggris tujuan umum. Ini dapat digunakan dalam berbagai kasus penggunaan termasuk ringkasan teks, klasifika si, penyelesaian teks, atau penyelesaian kode.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Deskripsi
huggingface-llm-mistral-7b-instruksikan	`Mistral7BInstruct`	Mistral 7B Instruct adalah versi Mistral 7B yang disetel dengan baik untuk kasus penggunaan percakapan. Itu khusus menggunakan berbagai kumpulan data percakapan yang tersedia untuk umum dalam bahasa Inggris.
huggingface-textgeneration1-mpt-7b-bf16	`MPT7B`	MPT 7B adalah model bahasa besar transformator gaya decoder dengan 6,7 miliar parameter, pra-dilatih dari awal pada 1 triliun token teks dan kode bahasa Inggris. Ini disiapkan untuk menangani panjang konteks yang panjang.
huggingface-textgeneration1-mpt-7b-instruct-bf16	`MPT7BInstruct`	MPT 7B Instruct adalah model untuk instruksi bentuk pendek berikut tugas-tugas. Ini dibangun dengan menyempurnakan MPT 7B pada kumpulan data yang berasal dari dataset yang berasal dari dataset databricks-dolly-15k dan Anthropic Helpful and Harmware (HH-RLHF).

Jenis file dataset dan format data input

Penyetelan berbasis instruksi menggunakan kumpulan data berlabel untuk meningkatkan kinerja LLM yang telah dilatih sebelumnya pada tugas pemrosesan bahasa alami (NLP) tertentu. Contoh berlabel diformat sebagai pasangan prompt respons dan diungkapkan sebagai instruksi.

Untuk mempelajari jenis file kumpulan data yang didukung, lihatJenis file dataset yang didukung.

Untuk mempelajari tentang format data input, lihatFormat data input untuk fine-tuning berbasis instruksi.

Jenis file dataset yang didukung

Autopilot mendukung kumpulan data fine-tuning berbasis instruksi yang diformat sebagai file CSV (default) atau sebagai file Parket.

- CSV (nilai dipisahkan koma) adalah format file berbasis baris yang menyimpan data dalam teks biasa yang dapat dibaca manusia, yang merupakan pilihan populer untuk pertukaran data karena didukung oleh berbagai aplikasi.
- Parket adalah format file biner berbasis kolom di mana data disimpan dan diproses lebih efisien daripada dalam format file yang dapat dibaca manusia seperti CSV. Ini menjadikannya pilihan yang lebih baik untuk masalah data besar.

> **Note**
>
> Dataset dapat terdiri dari beberapa file, yang masing-masing harus mematuhi template tertentu. Untuk informasi tentang cara memformat data input Anda, lihatFormat data input untuk fine-tuning berbasis instruksi.

Format data input untuk fine-tuning berbasis instruksi

Setiap file dalam kumpulan data harus mematuhi format berikut:

- Dataset harus berisi tepat dua kolom yang dipisahkan koma dan diberi nama, dan. `input` `output` Autopilot tidak mengizinkan kolom tambahan.
- `input`Kolom berisi petunjuk, dan yang sesuai `output` berisi jawaban yang diharapkan. Keduanya `input` dan `output` dalam format string.

Contoh berikut menggambarkan format data input untuk fine-tuning berbasis instruksi di Autopilot.

```
input,output
"<prompt text>","<expected generated text>"
```

> **Note**
>
> Kami merekomendasikan penggunaan kumpulan data dengan minimal 1000 baris untuk memastikan pembelajaran dan kinerja model yang optimal.

Selain itu, Autopilot menetapkan batas maksimum jumlah baris dalam kumpulan data dan panjang konteks berdasarkan jenis model yang digunakan.

- Batas jumlah baris dalam kumpulan data berlaku untuk jumlah kumulatif baris di semua file dalam kumpulan data, termasuk beberapa file. Jika ada dua jenis saluran yang ditentukan (satu untuk pelatihan dan satu untuk validasi), batas tersebut berlaku untuk jumlah total baris di semua kumpulan data dalam kedua saluran. Ketika jumlah baris melebihi ambang batas, pekerjaan gagal dengan kesalahan validasi.
- Ketika panjang input atau output baris dalam kumpulan data melebihi batas yang ditetapkan pada konteks model bahasa, maka secara otomatis terpotong. Jika lebih dari 60% baris dalam kumpulan data terpotong, baik dalam input atau outputnya, Autopilot gagal dalam pekerjaan dengan kesalahan validasi.

Tabel berikut menyajikan batas-batas untuk setiap model.

SageMaker JumpStart ID Model	**`BaseModelName`** dalam permintaan API	Batas Baris	Batas Panjang Konteks
huggingface-textgeneration-dolly-v2-3b-bf16	`Dolly3B`	10.000 baris	1024 token
huggingface-textgeneration-dolly-v2-7b-bf16	`Dolly7B`	10.000 baris	1024 token

SageMaker JumpStart ID Model	**`BaseModel Name`** dalam permintaan API	Batas Baris	Batas Panjang Konteks
huggingface-textgeneration-dolly-v2-12b-bf16	`Dolly12B`	10.000 baris	1024 token
huggingface-llm-falcon-7b-bf16	`Falcon7B`	1.000 baris	1024 token
huggingface-llm-falcon-7b-instruct-bf16	`Falcon7BInstruct`	1.000 baris	1024 token
huggingface-llm-falcon-40b-bf16	`Falcon40B`	10.000 baris	1024 token
huggingface-llm-falcon-40b-instruct-bf16	`Falcon40BInstruct`	10.000 baris	1024 token
huggingface-text2text-flan-t5-large	`FlanT5L`	10.000 baris	1024 token
huggingface-text2text-flan-t5-xl	`FlanT5XL`	10.000 baris	1024 token
huggingface-text2text-flan-t5-xxll	`FlanT5XXL`	10.000 baris	1024 token
meta-textgeneration-llama-2-7b	`Llama2-7B`	10.000 baris	2048 token
meta-textgeneration-llama-2-7b-f	`Llama2-7BChat`	10.000 baris	2048 token
meta-textgeneration-llama-2-13b	`Llama2-13B`	7.000 baris	2048 token

SageMaker JumpStart ID Model	**`BaseModel Name`** dalam permintaan API	Batas Baris	Batas Panjang Konteks
meta-textgeneration-llama-2-13b-f	`Llama2-13BChat`	7.000 baris	2048 token
huggingface-llm-mistral-7b	`Mistral7B`	10.000 baris	2048 token
huggingface-llm-mistral-7b-instruct	`Mistral7B Instruct`	10.000 baris	2048 token
huggingface-textgeneration1-mpt-7b-bf16	`MPT7B`	10.000 baris	1024 token
huggingface-textgeneration1-mpt-7b-instruct-bf16	`MPT7BInstruct`	10.000 baris	1024 token

Optimalkan proses pembelajaran model pembuatan teks Anda dengan hyperparameters

Anda dapat mengoptimalkan proses pembelajaran model dasar Anda dengan menyesuaikan kombinasi apa pun dari hiperparameter berikut. Parameter ini tersedia untuk semua model.

- Hitungan Epoch: `epochCount` Hyperparameter menentukan berapa kali model melewati seluruh kumpulan data pelatihan. Ini mempengaruhi durasi pelatihan dan dapat mencegah overfitting ketika diatur dengan tepat. Sejumlah besar zaman dapat meningkatkan runtime keseluruhan pekerjaan fine-tuning. Kami merekomendasikan pengaturan besar `MaxAutoMLJobRuntimeInSeconds` di dalam `TextGenerationJobConfig` untuk menghindari pekerjaan fine-tuning berhenti sebelum waktunya. `CompletionCriteria`
- Ukuran Batch: `batchSize` Hyperparameter mendefinisikan jumlah sampel data yang digunakan dalam setiap iterasi pelatihan. Hal ini dapat mempengaruhi kecepatan konvergensi dan penggunaan memori. Dengan ukuran batch yang besar, risiko kesalahan out of memory (OOM) meningkat, yang mungkin muncul sebagai kesalahan server internal di Autopilot. Untuk memeriksa kesalahan tersebut, periksa grup `/aws/sagemaker/TrainingJobs` log untuk pekerjaan

pelatihan yang diluncurkan oleh pekerjaan Autopilot Anda. Anda dapat mengakses log masuk tersebut CloudWatch dari konsol AWS manajemen. Pilih Log, lalu pilih grup **`/aws/sagemaker/TrainingJobs`** log. Untuk memperbaiki kesalahan OOM, kurangi ukuran batch.

Kami merekomendasikan memulai dengan ukuran batch 1, kemudian secara bertahap meningkatkannya sampai terjadi kesalahan di luar memori. Sebagai referensi, 10 zaman biasanya membutuhkan waktu hingga 72 jam untuk diselesaikan.

- Tingkat Pembelajaran: `learningRate` Hyperparameter mengontrol ukuran langkah di mana parameter model diperbarui selama pelatihan. Ini menentukan seberapa cepat atau lambat parameter model diperbarui selama pelatihan. Tingkat pembelajaran yang tinggi berarti bahwa parameter diperbarui dengan ukuran langkah yang besar, yang dapat menyebabkan konvergensi lebih cepat tetapi juga dapat menyebabkan proses pengoptimalan melampaui solusi optimal dan menjadi tidak stabil. Tingkat pembelajaran yang rendah berarti bahwa parameter diperbarui dengan ukuran langkah kecil, yang dapat menyebabkan konvergensi yang lebih stabil tetapi dengan mengorbankan pembelajaran yang lebih lambat.
- Learning Rate Warmup Steps: `learningRateWarmupSteps` Hyperparameter menentukan jumlah langkah pelatihan di mana tingkat pembelajaran meningkat secara bertahap sebelum mencapai target atau nilai maksimumnya. Ini membantu model bertemu lebih efektif dan menghindari masalah seperti divergensi atau konvergensi lambat yang dapat terjadi dengan tingkat pembelajaran yang awalnya tinggi.

Untuk mempelajari cara menyesuaikan hyperparameters untuk eksperimen fine-tuning Anda di Autopilot dan menemukan kemungkinan nilainya, lihat. Cara mengatur hyperparameters untuk mengoptimalkan proses pembelajaran suatu model

Metrik untuk menyempurnakan model bahasa besar di Autopilot

Dengan menggunakan kumpulan data Anda, Autopilot secara langsung menyempurnakan model bahasa target (LLM) untuk meningkatkan metrik objektif default, kehilangan lintas entropi.

Kehilangan entropi silang adalah metrik yang banyak digunakan untuk menilai perbedaan antara distribusi probabilitas yang diprediksi dan distribusi kata yang sebenarnya dalam data pelatihan. Dengan meminimalkan kehilangan lintas entropi, model belajar untuk membuat prediksi yang lebih akurat dan relevan secara kontekstual, terutama dalam tugas-tugas yang berkaitan dengan pembuatan teks.

Setelah menyempurnakan LLM Anda dapat mengevaluasi kualitas teks yang dihasilkan menggunakan berbagai skor ROUGE. Selain itu, Anda dapat menganalisis kebingungan dan pelatihan lintas entropi dan kerugian validasi sebagai bagian dari proses evaluasi.

- Kehilangan kebingungan mengukur seberapa baik model dapat memprediksi kata berikutnya dalam urutan teks, dengan nilai yang lebih rendah menunjukkan pemahaman yang lebih baik tentang bahasa dan konteks.
- Recall-Oriented Understudy for Gisting Evaluation (ROUGE) adalah seperangkat metrik yang digunakan di bidang pemrosesan bahasa alami (NLP) dan pembelajaran mesin untuk mengevaluasi kualitas teks yang dihasilkan mesin, seperti ringkasan teks atau pembuatan teks. Ini terutama menilai kesamaan antara teks yang dihasilkan dan teks referensi kebenaran dasar (ditulis manusia) dari kumpulan data validasi. Ukuran ROUGE dirancang untuk menilai berbagai aspek kesamaan teks, termasuk presisi dan ingatan n-gram (urutan kata yang berdekatan) dalam teks yang dihasilkan sistem dan referensi. Tujuannya adalah untuk menilai seberapa baik model menangkap informasi yang ada dalam teks referensi.

  Ada beberapa varian metrik ROUGE, tergantung pada jenis n-gram yang digunakan dan aspek spesifik dari kualitas teks yang dievaluasi.

  Daftar berikut berisi nama dan deskripsi metrik ROUGE yang tersedia setelah penyempurnaan model bahasa besar di Autopilot.

  `ROUGE-1`, **`ROUGE-2`**

  ROUGE-N, metrik ROUGE utama, mengukur tumpang tindih n-gram antara teks yang dihasilkan sistem dan teks referensi. ROUGE-N dapat disesuaikan dengan nilai yang berbeda dari n (di sini 1 atau2) untuk mengevaluasi seberapa baik teks yang dihasilkan sistem menangkap n-gram dari teks referensi.

  `ROUGE-L`

  ROUGE-L (Rouge-longest Common Subjurence) menghitung urutan umum terpanjang antara teks yang dihasilkan sistem dan teks referensi. Varian ini mempertimbangkan urutan kata selain konten tumpang tindih.

  `ROUGE-L-Sum`

  ROUGE-L-SUM (Urutan Umum Terpanjang untuk Ringkasan) dirancang untuk evaluasi sistem ringkasan teks. Ini berfokus pada pengukuran urutan umum terpanjang antara ringkasan yang dihasilkan mesin dan ringkasan referensi. ROUGE-L-SUM memperhitungkan urutan kata dalam teks, yang penting dalam tugas ringkasan teks.

Penyebaran dan prediksi model autopilot

Setelah menyempurnakan model bahasa besar (LLM), Anda dapat menerapkan model untuk pembuatan teks waktu nyata dengan menyiapkan titik akhir untuk mendapatkan prediksi interaktif.

> **Note**
>
> Kami merekomendasikan menjalankan pekerjaan inferensi waktu nyata `ml.g5.12xlarge` untuk kinerja yang lebih baik. Atau, `ml.g5.8xlarge` instance cocok untuk tugas pembuatan teks Falcon-7B-Instruct dan MPT-7B-Instruct.
> Anda dapat menemukan spesifikasi instans ini dalam kategori Komputasi Akselerasi dalam pemilihan jenis instans yang disediakan oleh Amazon EC2.

Pembuatan teks waktu nyata

Anda dapat menggunakan SageMaker API untuk menerapkan model fine-tuned secara manual ke titik akhir inferensi real-time SageMaker Hosting, lalu mulai membuat prediksi dengan menjalankan endpoint sebagai berikut.

> **Note**
>
> Atau, Anda dapat memilih opsi penerapan otomatis saat membuat eksperimen fine-tuning Anda di Autopilot. Untuk informasi tentang pengaturan penerapan otomatis model, lihatCara mengaktifkan penyebaran otomatis.
> Anda juga dapat menggunakan SageMaker Python SDK dan `JumpStartModel` kelas untuk melakukan inferensi dengan model yang disetel dengan baik oleh Autopilot. Ini dapat dilakukan dengan menentukan lokasi khusus untuk artefak model di Amazon S3. Untuk informasi tentang mendefinisikan model Anda sebagai model dan menerapkan SageMaker JumpStart model Anda untuk inferensi, lihat Penerapan kode rendah dengan kelas. JumpStartModel

1. Dapatkan definisi wadah inferensi kandidat

   Anda dapat menemukan bagian `InferenceContainerDefinitions` dalam `BestCandidate` objek yang diambil dari respons terhadap panggilan API DescribeAutoMLJobv2. Definisi kontainer untuk inferensi mengacu pada lingkungan kontainer yang dirancang untuk menerapkan dan menjalankan model terlatih Anda untuk membuat prediksi.

Contoh AWS CLI perintah berikut menggunakan DescribeAutoMLJobv2 API untuk mendapatkan definisi kontainer yang direkomendasikan untuk nama pekerjaan Anda.

```
aws sagemaker describe-auto-ml-job-v2 --auto-ml-job-name job-name --region region
```

2. Buat SageMaker model

   Gunakan definisi container dari langkah sebelumnya untuk membuat SageMaker model dengan menggunakan CreateModelAPI. Lihat AWS CLI perintah berikut sebagai contoh. Gunakan `CandidateName` untuk nama model Anda.

```
aws sagemaker create-model --model-name '<your-candidate-name>' \
                    --primary-container '<container-definition' \
                    --execution-role-arn '<execution-role-arn>' --region '<region>
```

3. Buat konfigurasi titik akhir

   Contoh AWS CLI perintah berikut menggunakan CreateEndpointConfigAPI untuk membuat konfigurasi endpoint.

> **Note**
>
> Untuk mencegah pembuatan titik akhir dari kehabisan waktu karena unduhan model yang panjang, kami sarankan pengaturan `ModelDataDownloadTimeoutInSeconds = 3600` dan. `ContainerStartupHealthCheckTimeoutInSeconds = 3600`

```
aws sagemaker create-endpoint-config --endpoint-config-name '<your-endpoint-config-
name>' \
                    --production-variants '<list-of-
production-variants>' ModelDataDownloadTimeoutInSeconds=3600
 ContainerStartupHealthCheckTimeoutInSeconds=3600 \
                    --region '<region>'
```

4. Buat titik akhir

   AWS CLIContoh berikut menggunakan CreateEndpointAPI untuk membuat titik akhir.

```
aws sagemaker create-endpoint --endpoint-name '<your-endpoint-name>' \
```

```
                    --endpoint-config-name '<endpoint-config-name-you-just-created>'
 \
                    --region '<region>'
```

Periksa kemajuan penerapan titik akhir Anda dengan menggunakan API. DescribeEndpoint Lihat AWS CLI perintah berikut sebagai contoh.

```
aws sagemaker describe-endpoint –endpoint-name '<endpoint-name>' –region <region>
```

Setelah `EndpointStatus` perubahan`InService`, titik akhir siap digunakan untuk inferensi waktu nyata.

5. Memanggil titik akhir

   Perintah berikut memanggil titik akhir untuk inferensi real-time. Prompt Anda perlu dikodekan dalam byte.

   > **Note**
   >
   > Format prompt input Anda tergantung pada model bahasa. Untuk informasi selengkapnya tentang format prompt pembuatan teks, lihatFormat permintaan untuk model pembuatan teks inferensi waktu nyata.

```
aws sagemaker invoke-endpoint --endpoint-name '<endpoint-name>' \
                  --region '<region>' --body '<your-promt-in-bytes>' [--content-type]
 'application/json' <outfile>
```

Format permintaan untuk model pembuatan teks inferensi waktu nyata

Model bahasa besar (LLM) yang berbeda mungkin memiliki dependensi perangkat lunak tertentu, lingkungan runtime, dan persyaratan perangkat keras yang memengaruhi wadah yang direkomendasikan Autopilot untuk meng-host model untuk inferensi. Selain itu, setiap model menentukan format data input yang diperlukan dan format yang diharapkan untuk prediksi dan output.

Berikut adalah contoh input untuk beberapa model dan wadah yang direkomendasikan.

- Untuk model Falcon dengan wadah `huggingface-pytorch-tgi-inference:2.0.1-tgi1.0.3-gpu-py39-cu118-ubuntu20.04` yang direkomendasikan:

```
payload = {
    "inputs": "Large language model fine-tuning is defined as",
    "parameters": {
        "do_sample": false,
        "top_p": 0.9,
        "temperature": 0.1,
        "max_new_tokens": 128,
        "stop": ["<|endoftext|>", "</s>"]
    }
}
```

- Untuk semua model lain dengan wadah yang direkomendasikan`djl-inference:0.22.1-fastertransformer5.3.0-cu118`:

```
payload= {
    "text_inputs": "Large language model fine-tuning is defined as"
}
```

Membuat eksperimen Autopilot Regresi atau Klasifikasi untuk data tabular menggunakan UI Studio Classic

> **Note**
>
> Semua instruksi terkait UI dalam panduan ini berkaitan dengan fitur mandiri Autopilot sebelum bermigrasi ke Amazon Canvas. SageMaker Pengguna yang mengikuti petunjuk ini harus menggunakan Studio Classic.

Anda dapat menggunakan Amazon SageMaker Studio Classic UI untuk membuat eksperimen Autopilot untuk masalah klasifikasi atau regresi pada data tabular. UI membantu Anda menentukan nama eksperimen, menyediakan lokasi untuk data input dan output, dan menentukan data target mana yang akan diprediksi. Secara opsional, Anda juga dapat menentukan jenis masalah yang ingin Anda selesaikan (regresi, klasifikasi, klasifikasi multiclass), pilih strategi pemodelan Anda (ansambel bertumpuk atau optimasi hiperparameter), pilih daftar algoritma yang digunakan oleh pekerjaan Autopilot untuk melatih data, dan banyak lagi.

UI memiliki deskripsi, sakelar sakelar, menu tarik-turun, tombol radio, dan lainnya untuk membantu Anda menavigasi pembuatan kandidat model Anda. Setelah eksperimen berjalan, Anda dapat membandingkan uji coba dan mempelajari detail langkah pra-pemrosesan, algoritme, dan rentang hiperparameter dari setiap model. Secara opsional, Anda dapat mengunduh laporan penjelasan dan kinerjanya. Gunakan buku catatan yang disediakan untuk melihat hasil eksplorasi data otomatis atau definisi model kandidat.

Atau, Anda dapat menggunakan Autopilot AutoML API di. Membuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API

Konfigurasikan parameter default eksperimen Autopilot (untuk administrator)

Autopilot mendukung pengaturan nilai default untuk menyederhanakan konfigurasi Amazon SageMaker Autopilot saat Anda membuat eksperimen Autopilot menggunakan UI Studio Classic. Administrator dapat menggunakan konfigurasi siklus hidup Studio Classic (LCC) untuk mengatur infrastruktur, jaringan, dan nilai keamanan dalam file konfigurasi dan mengisi pengaturan lanjutan pekerjaan. `AutoML`

Dengan demikian, mereka dapat sepenuhnya mengontrol konektivitas jaringan dan izin akses untuk sumber daya yang terkait dengan Amazon SageMaker Studio Classic, termasuk SageMaker instance, sumber data, data keluaran, dan layanan terkait lainnya. Secara khusus, administrator dapat mengonfigurasi arsitektur jaringan yang diinginkan, seperti Amazon VPC, subnet, dan grup keamanan, untuk Domain Studio Classic atau profil pengguna individu. Ilmuwan data dapat fokus pada parameter spesifik ilmu data saat membuat eksperimen Autopilot mereka menggunakan UI Studio Classic. Selanjutnya, administrator dapat mengelola enkripsi data pada instance di mana eksperimen Autopilot dijalankan dengan menetapkan kunci enkripsi default.

> **Note**
>
> Fitur ini saat ini tidak tersedia di Wilayah keikutsertaan Asia Pasifik (Hong Kong) dan Timur Tengah (Bahrain).

Di bagian berikut, Anda dapat menemukan daftar lengkap parameter yang mendukung pengaturan default saat membuat eksperimen Autopilot menggunakan UI Studio Classic, dan mempelajari cara mengatur nilai default tersebut.

Topik

- Daftar parameter default yang didukung

- Tetapkan parameter eksperimen Autopilot default

Daftar parameter default yang didukung

Parameter berikut mendukung pengaturan nilai default dengan file konfigurasi untuk membuat eksperimen Autopilot menggunakan UI Studio Classic. Setelah disetel, nilai secara otomatis mengisi bidang yang sesuai di tab Create Experiment Autopilot di UI Studio Classic. Lihat Pengaturan lanjutan (opsional) untuk deskripsi lengkap dari setiap bidang.

- Keamanan: Amazon VPC, subnet, dan grup keamanan.
- Akses: ARN peran AWS IAM.
- Enkripsi: ID AWS KMS kunci.
- Tag: Pasangan nilai kunci yang digunakan untuk memberi label dan mengatur SageMaker sumber daya.

Tetapkan parameter eksperimen Autopilot default

Administrator dapat menetapkan nilai default dalam file konfigurasi, lalu menempatkan file secara manual di lokasi yang direkomendasikan dalam lingkungan Studio Classic pengguna tertentu, atau mereka dapat meneruskan file ke skrip konfigurasi siklus hidup (LCC) untuk mengotomatiskan penyesuaian lingkungan Studio Classic untuk Domain atau profil pengguna tertentu.

- Untuk mengatur file konfigurasi, mulailah dengan mengisi parameter defaultnya.

  Untuk mengkonfigurasi salah satu atau semua nilai default yang tercantum dalamDaftar parameter default yang didukung, administrator dapat membuat file konfigurasi bernama`config.yaml`, struktur yang harus mematuhi file konfigurasi sampel ini. Cuplikan berikut menunjukkan contoh file konfigurasi dengan semua parameter yang didukung`AutoML`. Untuk informasi lebih lanjut tentang format file ini, lihat skema lengkap.

```
SchemaVersion: '1.0'
SageMaker:
  AutoMLJob:
    # https://docs.aws.amazon.com/sagemaker/latest/APIReference/
API_CreateAutoMLJob.html
    AutoMLJobConfig:
      SecurityConfig:
        EnableInterContainerTrafficEncryption: true
        VolumeKmsKeyId: 'kms-key-id'
```

```
        VpcConfig:
          SecurityGroupIds:
            - 'security-group-id-1'
            - 'security-group-id-2'
          Subnets:
            - 'subnet-1'
            - 'subnet-2'
    OutputDataConfig:
      KmsKeyId: 'kms-key-id'
    RoleArn: 'arn:aws:iam::111222333444:role/Admin'
    Tags:
    - Key: 'tag_key'
      Value: 'tag_value'
```

- Kemudian, tempatkan file konfigurasi di lokasi yang direkomendasikan dengan menyalin file secara manual ke jalur yang direkomendasikan atau menggunakan konfigurasi siklus hidup (LCC).

  File konfigurasi harus ada setidaknya di salah satu lokasi berikut di lingkungan Studio Classic pengguna. Secara default, SageMaker mencari file konfigurasi di dua lokasi:

  - Pertama, di`/etc/xdg/sagemaker/config.yaml`. Kami menyebut file ini sebagai file konfigurasi administrator.
  - Kemudian, di`/root/.config/sagemaker/config.yaml`. Kami menyebut file ini sebagai file konfigurasi pengguna.

  Menggunakan file konfigurasi administrator, administrator dapat menentukan satu set nilai default. Secara opsional, mereka dapat menggunakan file konfigurasi pengguna untuk mengganti nilai yang ditetapkan dalam file konfigurasi administrator, atau menetapkan nilai parameter default tambahan.

  Cuplikan berikut menunjukkan skrip contoh yang menulis file konfigurasi parameter default ke lokasi administrator di lingkungan Studio Classic pengguna. Anda dapat mengganti `/etc/xdg/sagemaker` dengan `/root/.config/sagemaker` untuk menulis file ke lokasi pengguna.

```
## Sample script with AutoML intelligent defaults
#!/bin/bash

sudo mkdir -p /etc/xdg/sagemaker

echo "SchemaVersion: '1.0'
CustomParameters:
  AnyStringKey: 'AnyStringValue'
```

```
SageMaker:
  AutoMLJob:
    # https://docs.aws.amazon.com/sagemaker/latest/APIReference/
API_CreateAutoMLJob.html
    AutoMLJobConfig:
      SecurityConfig:
        EnableInterContainerTrafficEncryption: true
        VolumeKmsKeyId: 'kms-key-id'
        VpcConfig:
          SecurityGroupIds:
            - 'security-group-id-1'
            - 'security-group-id-2'
          Subnets:
            - 'subnet-1'
            - 'subnet-2'
    OutputDataConfig:
      KmsKeyId: 'kms-key-id'
    RoleArn: 'arn:aws:iam::111222333444:role/Admin'
    Tags:
    - Key: 'tag_key'
      Value: 'tag_value'
" | sudo tee /etc/xdg/sagemaker/config.yaml
```

- Salin file secara manual — Untuk menyalin file konfigurasi secara manual, jalankan skrip yang dibuat pada langkah sebelumnya dari terminal Studio Classic. Dalam hal ini, profil pengguna yang mengeksekusi skrip dapat membuat eksperimen Autopilot dengan nilai default yang hanya berlaku untuk mereka.
- Buat konfigurasi SageMaker siklus hidup — Atau, Anda dapat menggunakan konfigurasi siklus hidup (LCC) untuk mengotomatiskan penyesuaian lingkungan Studio Classic Anda. LCC adalah skrip shell yang dipicu oleh peristiwa siklus hidup Amazon SageMaker Studio Classic seperti memulai aplikasi Studio Classic. Kustomisasi ini termasuk menginstal paket khusus, mengonfigurasi ekstensi notebook, pra-pemuatan kumpulan data, menyiapkan repositori kode sumber, atau, dalam kasus kami, pra-mengisi parameter default. Administrator dapat melampirkan LCC ke Domain Studio Classic untuk mengotomatiskan konfigurasi nilai default untuk setiap profil pengguna dalam Domain tersebut.

  Bagian berikut merinci cara membuat konfigurasi siklus hidup sehingga pengguna dapat memuat parameter default Autopilot secara otomatis saat meluncurkan Studio Classic. Anda dapat memilih untuk membuat LCC menggunakan SageMaker Konsol atau. AWS CLI

Create a LCC from the SageMaker Console

Gunakan langkah-langkah berikut untuk membuat LCC yang berisi parameter default Anda, lampirkan LCC ke Domain atau profil pengguna, lalu luncurkan aplikasi Studio Classic yang telah diisi sebelumnya dengan parameter default yang ditetapkan oleh LCC menggunakan Konsol. SageMaker

- Untuk membuat konfigurasi siklus hidup yang menjalankan skrip yang berisi nilai default Anda menggunakan Konsol SageMaker
  - Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
  - Di sisi kiri, arahkan ke konfigurasi Admin, lalu konfigurasi Siklus Hidup.
  - Dari halaman Konfigurasi Siklus Hidup, navigasikan ke tab Studio Classic, lalu pilih Buat konfigurasi.
  - Untuk Nama, ketikkan nama menggunakan karakter alfanumerik dan “-”, tetapi tidak ada spasi. Nama dapat memiliki maksimal 63 karakter.
  - Tempel skrip Anda di bagian Skrip.
  - Pilih Buat konfigurasi untuk membuat konfigurasi siklus hidup. Ini menciptakan tipe `Kernel gateway app` LCC.
- Untuk melampirkan konfigurasi siklus hidup ke Domain Studio Classic, spasi, atau profil pengguna

  Ikuti langkah-langkah di Lampirkan konfigurasi siklus hidup ke Domain Studio Classic atau profil pengguna untuk melampirkan LCC Anda ke Domain Studio Classic atau profil pengguna tertentu.
- Untuk meluncurkan aplikasi Studio Classic Anda dengan konfigurasi siklus hidup

  Setelah LCC dilampirkan ke Domain atau profil pengguna, pengguna yang terkena dampak dapat memulai aplikasi Studio Classic dari halaman landing Studio Classic di Studio untuk mengambil default yang ditetapkan oleh LCC secara otomatis. Ini secara otomatis mengisi UI Studio Classic saat membuat eksperimen Autopilot.

Create a LCC from the AWS CLI

Gunakan cuplikan berikut untuk meluncurkan aplikasi Studio Classic yang menjalankan skrip Anda menggunakan. AWS CLI Perhatikan bahwa `lifecycle_config.sh` adalah nama yang diberikan untuk skrip Anda dalam contoh ini.

Sebelum memulai:

- Pastikan Anda telah memperbarui dan mengonfigurasi AWS CLI dengan menyelesaikan prasyarat yang dijelaskan dalam Membuat konfigurasi siklus hidup dari konfigurasi. AWS CLI
- Instal dokumentasi OpenSSL. AWS CLIPerintah ini menggunakan pustaka open source OpenSSL untuk menyandikan skrip Anda dalam format Base64. Persyaratan ini mencegah kesalahan yang terjadi dari spasi dan pengkodean jeda baris.

Anda sekarang dapat mengikuti tiga langkah ini:

- Buat konfigurasi siklus hidup baru yang mereferensikan skrip konfigurasi **`lifecycle_config.sh`**

```
LCC_CONTENT=`openssl base64 -A -in lifecycle_config.sh`

## Create a new lifecycle config
aws sagemaker create-studio-lifecycle-config --region region \
--studio-lifecycle-config-name lcc-name \
--studio-lifecycle-config-content $LCC_CONTENT \
--studio-lifecycle-config-app-type default
```

  Perhatikan ARN dari konfigurasi siklus hidup yang baru dibuat yang dikembalikan. ARN ini diperlukan untuk melampirkan konfigurasi siklus hidup ke aplikasi Anda.

- Lampirkan konfigurasi siklus hidup ke **`JupyterServerApp`**

  Contoh berikut menunjukkan cara membuat profil pengguna baru dengan konfigurasi siklus hidup terlampir. Untuk memperbarui profil pengguna yang ada, gunakan AWS CLI update-user-profileperintah. Untuk membuat atau memperbarui Domain, lihat create-domain dan update-domain. Tambahkan ARN konfigurasi siklus hidup dari langkah sebelumnya ke pengaturan jenis aplikasi. `JupyterServerAppSettings` Anda dapat menambahkan beberapa konfigurasi siklus hidup secara bersamaan dengan menggunakan daftar konfigurasi siklus hidup.

```
# Create a new UserProfile
aws sagemaker create-user-profile --domain-id domain-id \
--user-profile-name user-profile-name \
--region region \
--user-settings '{
"JupyterServerAppSettings": {
  "LifecycleConfigArns":
    ["lifecycle-configuration-arn"]
```

```
  }
}'
```

Setelah LCC dilampirkan ke Domain atau profil pengguna, pengguna yang terkena dampak dapat mematikan dan memperbarui aplikasi Studio Classic yang ada dengan mengikuti langkah-langkah di Matikan dan Perbarui Amazon SageMaker Studio Classic, atau memulai aplikasi Studio Classic baru dari AWS Konsol untuk mengambil default yang ditetapkan oleh LCC secara otomatis. Ini secara otomatis mengisi UI Studio Classic saat membuat eksperimen Autopilot. Atau, mereka dapat meluncurkan aplikasi Studio Classic baru menggunakan AWS CLI sebagai berikut.

- Luncurkan aplikasi Studio Classic Anda dengan konfigurasi siklus hidup menggunakan AWS CLI

```
# Create a Jupyter Server application
aws sagemaker create-app --domain-id domain-id \
--user-profile-name user-profile-name \
--region region \
--app-type JupyterServer \
--resource-spec LifecycleConfigArn=lifecycle-configuration-arn \
--app-name default
```

Untuk informasi selengkapnya tentang membuat konfigurasi siklus hidup menggunakanAWS CLI, lihat Membuat Konfigurasi Siklus Hidup dari konfigurasi. AWS CLI

Untuk membuat eksperimen Autopilot menggunakan Studio Classic UI

1. Masuk di https://console.aws.amazon.com/sagemaker/, pilih Studio dari panel navigasi kiri, pilih Domain dan profil pengguna Anda, lalu Buka Studio.
2. Di Studio, pilih ikon Studio Classic di panel navigasi kiri atas. Ini membuka aplikasi Studio Classic.
3. Jalankan atau buka aplikasi Studio Classic dari ruang pilihan Anda, atau Buat ruang Studio Classic. . Pada tab Beranda, pilih kartu AutoML. Ini membuka tab AutoML baru.
4. Pilih Buat eksperimen AutoML. Ini membuka tab Buat eksperimen baru.
5. Di bagian Eksperimen dan detail data, masukkan informasi berikut:

   a. Nama eksperimen — Harus unik untuk akun Anda saat ini Wilayah AWS dan berisi maksimal 63 karakter alfanumerik. Dapat menyertakan tanda hubung (-) tetapi bukan spasi.

b. Input data — Menyediakan lokasi bucket Amazon Simple Storage Service (Amazon S3) untuk data masukan Anda. Bucket S3 ini harus ada di ember Anda saat iniWilayah AWS. URL harus dalam `s3://` format di mana Amazon SageMaker memiliki izin menulis. File harus dalam format CSV atau Parket dan berisi setidaknya 500 baris. Pilih Jelajahi untuk menggulir jalur yang tersedia dan Pratinjau untuk melihat sampel data masukan Anda.

c. Apakah input S3 Anda file manifes? — File manifes menyertakan metadata dengan data masukan Anda. Metadata menentukan lokasi data Anda di Amazon S3. Ini juga menentukan bagaimana data diformat dan atribut mana dari kumpulan data yang akan digunakan saat melatih model Anda. Anda dapat menggunakan file manifes sebagai alternatif untuk pra-pemrosesan saat data berlabel sedang dialirkan dalam mode. `Pipe`

d. Pisahkan data secara otomatis? Autopilot dapat membagi data Anda menjadi 80-20% split untuk data pelatihan dan validasi. Jika Anda lebih suka pemisahan khusus, Anda dapat memilih Tentukan rasio pemisahan. Untuk menggunakan kumpulan data kustom untuk validasi, pilih Menyediakan kumpulan validasi.

e. Lokasi data keluaran (bucket S3) - Nama lokasi bucket S3 tempat Anda ingin menyimpan data keluaran. URL untuk bucket ini harus dalam format Amazon S3 di mana Amazon SageMaker memiliki izin menulis. Bucket S3 harus dalam arusWilayah AWS. Autopilot juga dapat membuat ini untuk Anda di lokasi yang sama dengan data input Anda.

6. Pilih Berikutnya: Target dan fitur. Tab Target dan fitur terbuka.
7. Di bagian Target dan fitur:

   - Pilih kolom untuk ditetapkan sebagai target untuk prediksi model.
   - Secara opsional, Anda dapat meneruskan nama kolom bobot sampel di bagian Bobot sampel untuk meminta baris kumpulan data Anda diberi bobot selama pelatihan dan evaluasi. Untuk informasi selengkapnya tentang metrik objektif yang tersedia, lihatMetrik tertimbang autopilot.

     > **Note**
     >
     > Support untuk bobot sampel hanya tersedia dalam mode ensembling.

   - Anda juga dapat memilih fitur untuk pelatihan dan mengubah tipe datanya. Tipe data berikut tersedia:`Text`,`Numerical`,`Categorical`,`Datetime`,`Sequence`, dan`Auto`. Semua fitur dipilih secara default.

8. Pilih Berikutnya: Metode pelatihan. Tab Metode pelatihan terbuka.

9. Di bagian Metode pelatihan, pilih opsi pelatihan Anda: Ensembling, Hyperparameter optimization (HPO), atau Auto untuk membiarkan Autopilot memilih metode pelatihan secara otomatis berdasarkan ukuran dataset. Setiap mode pelatihan menjalankan serangkaian algoritme yang telah ditentukan sebelumnya pada kumpulan data Anda untuk melatih kandidat model. Secara default, Autopilot pra-memilih semua algoritma yang tersedia untuk mode pelatihan yang diberikan. Anda dapat menjalankan eksperimen pelatihan Autopilot dengan semua algoritme atau memilih subset Anda sendiri.

   Untuk informasi lebih lanjut tentang mode pelatihan dan algoritme yang tersedia, lihat bagian Mode pelatihan Autopilot di halaman Mode pelatihan dan algoritme.

10. Pilih Berikutnya: Deployment dan pengaturan lanjutan untuk membuka tab Deployment dan advanced settings. Pengaturan mencakup nama titik akhir tampilan otomatis, jenis masalah pembelajaran mesin, dan pilihan tambahan untuk menjalankan eksperimen Anda.

    a. Pengaturan penyebaran — Autopilot dapat secara otomatis membuat titik akhir dan menerapkan model Anda untuk Anda.

       Untuk menerapkan otomatis ke titik akhir yang dibuat secara otomatis, atau untuk memberikan nama titik akhir untuk penerapan khusus, setel sakelar ke Ya di bawah Penerapan otomatis? Jika Anda mengimpor data dari Amazon SageMaker Data Wrangler, Anda memiliki opsi tambahan untuk menerapkan model terbaik secara otomatis dengan atau tanpa transformasi dari Data Wrangler.

       > **Note**
       >
       > Jika alur Data Wrangler berisi operasi multi-baris seperti`groupby`,, atau `joinconcatenate`, Anda tidak dapat menerapkan otomatis dengan transformasi ini. Untuk informasi selengkapnya, lihat Melatih Model Secara Otomatis pada Alur Data Anda.

    b. Pengaturan lanjutan (opsional) - Autopilot menyediakan kontrol tambahan untuk mengatur parameter eksperimental secara manual seperti menentukan jenis masalah Anda, batasan waktu pada pekerjaan dan uji coba Autopilot Anda, keamanan, dan pengaturan enkripsi.

       > **Note**
       >
       > Autopilot mendukung pengaturan nilai default untuk menyederhanakan konfigurasi eksperimen Autopilot menggunakan Studio Classic UI. Administrator dapat

menggunakan konfigurasi siklus hidup Studio Classic (LCC) untuk mengatur infrastruktur, jaringan, dan nilai keamanan dalam file konfigurasi dan mengisi pengaturan lanjutan pekerjaan. `AutoML`
Untuk mempelajari tentang bagaimana administrator dapat mengotomatiskan kustomisasi eksperimen Autopilot, lihat. Konfigurasikan parameter default eksperimen Autopilot (untuk administrator)

i. Jenis masalah pembelajaran mesin — Autopilot dapat secara otomatis menyimpulkan jenis masalah pembelajaran yang diawasi dari kumpulan data Anda. Jika Anda lebih suka memilihnya secara manual, Anda dapat menggunakan menu tarik-turun Pilih jenis masalah pembelajaran mesin. Perhatikan bahwa defaultnya ke Auto. Dalam beberapa kasus, SageMaker tidak dapat menyimpulkan secara akurat. Ketika itu terjadi, Anda harus memberikan nilai agar pekerjaan berhasil. Secara khusus, Anda dapat memilih dari jenis berikut:

   - Klasifikasi biner — Klasifikasi biner memberikan data input ke salah satu dari dua kelas yang telah ditentukan dan saling eksklusif, berdasarkan atributnya, seperti diagnosis medis berdasarkan hasil tes diagnostik yang menentukan apakah seseorang memiliki penyakit.
   - Regresi menetapkan hubungan antara variabel input (juga dikenal sebagai variabel independen atau fitur) dan variabel target (juga dikenal sebagai variabel dependen). Hubungan ini ditangkap melalui fungsi matematika atau model yang memetakan variabel input ke output kontinu. Ini biasanya digunakan untuk tugas-tugas seperti memprediksi harga rumah berdasarkan fitur seperti luas persegi dan jumlah kamar mandi, tren pasar saham, atau memperkirakan angka penjualan.
   - Klasifikasi multiclass — Klasifikasi multiclass memberikan data input ke salah satu dari beberapa kelas berdasarkan atributnya, seperti prediksi topik yang paling relevan dengan dokumen teks, seperti politik, keuangan, atau filsafat.

ii. Runtime — Anda dapat menentukan batas waktu maksimum. Setelah mencapai batas waktu, uji coba dan pekerjaan yang melebihi batasan waktu secara otomatis berhenti.

iii. Akses - Anda dapat memilih peran yang diasumsikan Amazon SageMaker Studio Classic untuk mendapatkan akses sementara Layanan AWS (khususnya, SageMaker dan Amazon S3) atas nama Anda. Jika tidak ada peran yang didefinisikan secara eksplisit, Studio Classic secara otomatis menggunakan peran SageMaker eksekusi default yang dilampirkan ke profil pengguna Anda.

iv. Enkripsi — Untuk meningkatkan keamanan data Anda saat istirahat dan melindunginya dari akses yang tidak sah, Anda dapat menentukan kunci enkripsi untuk mengenkripsi data di bucket Amazon S3 dan di volume Amazon Elastic Block Store (Amazon EBS) yang dilampirkan ke Domain Studio Classic Anda.

v. Keamanan — Anda dapat memilih cloud pribadi virtual (Amazon VPC) tempat SageMaker pekerjaan Anda berjalan. Pastikan Amazon VPC memiliki akses ke bucket Amazon S3 input dan output Anda.

vi. Proyek — Tentukan nama SageMaker proyek untuk dikaitkan dengan eksperimen Autopilot ini dan keluaran model. Saat Anda menentukan proyek, Autopilot menandai proyek tersebut ke eksperimen. Ini memungkinkan Anda mengetahui keluaran model mana yang terkait dengan proyek ini.

vii. Tag — Tag adalah array pasangan kunci-nilai. Gunakan tag untuk mengkategorikan sumber daya AndaLayanan AWS, seperti tujuan, pemilik, atau lingkungannya.

c. Pilih Berikutnya: Tinjau dan buat untuk mendapatkan ringkasan eksperimen Autopilot Anda sebelum Anda membuatnya.

11. Pilih Buat eksperimen. Pembuatan eksperimen memulai pekerjaan Autopilot di. SageMaker Autopilot memberikan status eksperimen, informasi tentang proses eksplorasi data dan kandidat model di notebook, daftar model yang dihasilkan dan laporannya, dan profil pekerjaan yang digunakan untuk membuatnya.

    Untuk informasi tentang buku catatan yang dihasilkan oleh pekerjaan Autopilot, lihat. Notebook Amazon SageMaker Autopilot dihasilkan untuk mengelola tugas AutoML Untuk informasi tentang detail setiap kandidat model dan laporannya, lihatModel yang dihasilkan oleh Amazon SageMaker Autopilot .

> **Note**
>
> Untuk menghindari biaya yang tidak perlu: Jika Anda menerapkan model yang tidak lagi diperlukan, hapus titik akhir dan sumber daya yang dibuat selama penerapan tersebut. Informasi tentang instans harga menurut Wilayah tersedia di Amazon SageMaker Pricing.

Notebook SageMaker contoh Amazon Autopilot

Notebook berikut berfungsi sebagai contoh praktis dan langsung yang membahas berbagai kasus penggunaan Autopilot.

Anda dapat menemukan semua notebook Autopilot di `autopilot`direktori repositori contoh. SageMaker GitHub

Kami merekomendasikan kloning repositori Git lengkap dalam Studio Classic untuk mengakses dan menjalankan notebook secara langsung. Untuk informasi tentang cara mengkloning repositori Git di Studio Classic, lihat. Mengkloning Repositori Git di Studio Classic SageMaker

Kasus penggunaan	Deskripsi
Inferensi tanpa server	Secara default, Autopilot memungkinkan penerapan model yang dihasilkan ke titik akhir inferensi waktu nyata. Dalam repositori ini, notebook mengilustrasikan cara menerapkan model Autopilot yang dilatih dan mode ke titik akhir tanpa server. `ENSEMBLING HYPERPARAMETER OPTIMIZATION (HPO)` Titik akhir tanpa server secara otomatis meluncurkan sumber daya komputasi dan menskalakannya masuk dan keluar tergantung pada lalu lintas, sehingga tidak perlu memilih jenis instance atau mengelola kebijakan penskalaan.
Pemilihan fitur kustom	Autopilot memeriksa kumpulan data Anda, dan menjalankan sejumlah kandidat untuk mengetahui kombinasi optimal dari langkah-langkah pra-pemrosesan data, algoritme pembelajaran mesin, dan hiperparameter. Anda dapat dengan mudah menerapkan baik pada titik akhir real-time atau untuk pemrosesan batch. Dalam beberapa kasus, Anda mungkin ingin memiliki fleksibilitas untuk membawa kode pemrosesan data khusus ke Autopilot. Misalnya, kumpulan data Anda mungkin berisi sejumlah besar variabel independen, dan Anda mungkin ingin memasukkan langkah pemilihan fitur khusus untuk menghapus variabel yang

Kasus penggunaan	Deskripsi
	tidak relevan terlebih dahulu. Dataset yang lebih kecil yang dihasilkan kemudian dapat digunakan untuk meluncurkan pekerjaan Autopilot. Pada akhirnya, Anda juga ingin menyertakan kode pemrosesan khusus dan model dari Autopilot untuk pemrosesan real-time atau batch.

Kasus penggunaan	Deskripsi
Contoh pipa	Sementara Autopilot merampingkan proses pembuatan model MLOP, insinyur MLOPs masih bertanggung jawab untuk membuat, mengotomatisasi, dan mengelola alur kerja ML dalam produksi. end-to-end SageMaker Pipelines dapat membantu mengotoma tisasi berbagai langkah siklus hidup ML, seperti prapemrosesan data, pelatihan model, penyetelan hiperparameter, evaluasi model, dan penerapan. Notebook ini berfungsi sebagai demonstrasi bagaimana menggabungkan Autopilot ke dalam alur kerja pelatihan AutoML Pipelines SageMaker . end-to-end Untuk meluncurkan eksperimen Autopilot dalam Pipelines, Anda harus membuat alur kerja pembuatan model dengan menulis kode integrasi kustom menggunakan Pipelines Lambda atau langkah-langkah Pemrosesan. Untuk informasi selengkapnya, lihat Pindahkan model Amazon SageMaker Autopilot MLdari eksperimen ke produksi menggunakan Amazon Pipelines. SageMaker Atau, saat menggunakan Autopilot dalam mode Ensembling, Anda dapat merujuk ke contoh notebook yang menunjukkan cara menggunak an langkah AutoML asli dalam langkah AutoML asli Pipeline. SageMaker Dengan Autopilot didukung sebagai langkah asli dalam Pipelines , Anda sekarang dapat menambahkan langkah pelatihan otomatis (AutoMLStep) ke Pipelines Anda dan menjalankan eksperimen Autopilot dalam mode Ensembling.

Kasus penggunaan	Deskripsi
Lebih banyak notebook	Anda dapat menemukan lebih banyak buku catatan yang menggambarkan kasus penggunaan lain seperti transformasi batch, peramalan deret waktu, dan lainnya di direktori root.

Kuota Amazon SageMaker Autopilot

Ada kuota yang membatasi sumber daya yang tersedia untuk Anda saat menggunakan Amazon SageMaker Autopilot. Beberapa dari batasan ini dapat ditingkatkan dan beberapa tidak.

> **Note**
>
> Kuota sumber daya yang didokumentasikan di bagian berikut berlaku untuk versi Amazon SageMaker Studio Classic 3.22.2 dan yang lebih tinggi. Untuk informasi tentang memperbarui versi SageMaker Studio Classic, lihatMatikan dan Perbarui Aplikasi SageMaker Studio Classic dan Studio Classic.

Topik

- Kuota yang dapat Anda tingkatkan
- Kuota sumber daya

Kuota yang dapat Anda tingkatkan

Batas sumber daya

Sumber daya	Wilayah	Batas default	Dapat ditingkatkan hingga
Ukuran dataset masukan	Semua	100 GB	Ratusan GB
Ukuran file Parket tunggal*	Semua	2 GB	N/A

Sumber daya	Wilayah	Batas default	Dapat ditingkatkan hingga
Ukuran set data target untuk subsampling**	Semua	5 GB	Ratusan GB
Jumlah pekerjaan Autopilot bersamaan	us-timur-1, us-timur-2, us-barat-2, ap-timur laut-1, eu-barat-1, eu-sentral-1	4	Ratusan
	ap-timur laut-2, ap-tenggara 2, eu-barat-2, ap-tenggara	2	Ratusan
	Wilayah Lainnya	1	Puluhan

> **Note**
>
> * Batas ukuran 2 GB ini untuk satu file Parket terkompresi. Anda dapat menyediakan kumpulan data Parket yang mencakup beberapa file Parket terkompresi hingga ukuran maksimum kumpulan data input. Setelah file didekompresi, masing-masing dapat diperluas ke ukuran yang lebih besar.
> ** Autopilot secara otomatis mensubsampel kumpulan data input yang lebih besar dari ukuran set data target sambil memperhitungkan ketidakseimbangan kelas dan mempertahankan label kelas langka.

Untuk meminta peningkatan kuota:

1. Buka konsol Service Quotas.
2. Pilih peningkatan kuota Anda, lalu pilih Permintaan kenaikan di tingkat akun.
3. Dalam nilai kuota Naikkan, masukkan nilai limit baru yang Anda minta.
4. Pilih Minta.

Kuota sumber daya

Tabel berikut berisi batas sumber daya runtime untuk pekerjaan Amazon SageMaker Autopilot di file. Wilayah AWS

Batas sumber daya per pekerjaan Autopilot

Sumber daya	Batas per pekerjaan Autopilot
Runtime maksimum untuk pekerjaan Autopilot	30 hari

Panduan Referensi API untuk Amazon SageMaker Autopilot

Bagian ini menyediakan subset dari API layanan HTTP untuk membuat dan mengelola SageMaker sumber daya Amazon Autopilot (pekerjaan AutoML) secara terprogram.

Untuk informasi tentang seluruh SageMaker REST API dan SDK yang tersedia, lihat Referensi API dan SDK.

Jika bahasa pilihan Anda adalah Python, Anda dapat merujuk AWS SDK for Python (Boto3)langsung.

Tindakan

Daftar ini merinci operasi yang tersedia di API Referensi untuk mengelola pekerjaan AutoML secara terprogram.

- `CreateAutoMLJob`
- `CreateAutoMLJobV2`
- `DescribeAutoMLJob`
- `DescribeAutoMLJobV2`
- `ListAutoMLJobs`
- `ListCandidatesForAutoMLJob`
- `StopAutoMLJob`

> **Note**
>
> CreateAutoMLJobv2 dan DescribeAutoMLJobv2 adalah versi baru CreateAutoMLJob dan DescribeAutoMLJob yang menawarkan kompatibilitas mundur.

Kami merekomendasikan menggunakan`CreateAutoMLJobV2`. `CreateAutoMLJobV2`dapat mengelola jenis masalah tabular yang identik dengan versi sebelumnya`CreateAutoMLJob`, serta jenis masalah non-tabular seperti klasifikasi gambar atau teks, atau peramalan deret waktu.
Temukan panduan tentang cara memigrasikan ke `CreateAutoMLJobV2` dalam Migrasi CreateAuto MLJob `CreateAutoMLJob` ke MLJobv2. CreateAuto

Tipe Data

Daftar ini merinci objek API AutoML yang digunakan oleh tindakan di atas sebagai permintaan masuk atau respons keluar.

- `AutoMLAlgorithmConfig`
- `AutoMLCandidate`
- `AutoMLCandidateGenerationConfig`
- `AutoMLCandidateStep`
- `AutoMLChannel`
- `AutoMLContainerDefinition`
- `AutoMLDataSource`
- `AutoMLDataSplitConfig`
- `AutoMLInferenceContainerDefinitions`
- `AutoMLJobArtifacts`
- `AutoMLJobChannel`
- `AutoMLJobCompletionCriteria`
- `AutoMLJobInputDataConfig`
- `AutoMLJobConfig`
- `AutoMLJobObjective`
- `AutoMLJobStepMetadata`
- `AutoMLJobSummary`
- `AutoMLOutputDataConfig`
- `AutoMLProblemTypeConfig`
- `AutoMLJobCompletionCriteria`

- AutoMLJobSummary
- AutoMLOutputDataConfig
- AutoMLPartialFailureReason
- AutoMLProblemTypeConfig
- AutoMLProblemTypeResolvedAttributes
- AutoMLResolvedAttributes
- AutoMLSecurityConfig
- AutoMLS3DataSource
- CandidateArtifactLocations
- CandidateGenerationConfig
- CandidateProperties
- FinalAutoMLJobObjectiveMetric
- HolidayConfigAttributes
- ImageClassificationJobConfig
- MetricDatum
- ModelDeployConfig
- ModelDeployResult
- ResolvedAttributes
- TabularJobConfig
- TabularResolvedAttributes
- TextGenerationJobConfig
- TextGenerationResolvedAttribute
- TimeSeriesConfig
- TimeSeriesForecastingJobConfig
- TimeSeriesTransformations
- TuningJobCompletionCriteria

SageMaker JumpStart

SageMaker JumpStart menyediakan model sumber terbuka yang sudah terlatih untuk berbagai jenis masalah untuk membantu Anda memulai pembelajaran mesin. Anda dapat melatih dan

menyetel model ini secara bertahap sebelum penerapan. JumpStart juga menyediakan templat solusi yang menyiapkan infrastruktur untuk kasus penggunaan umum, dan contoh notebook yang dapat dieksekusi untuk pembelajaran mesin. SageMaker

Anda dapat menerapkan, menyempurnakan, dan mengevaluasi model yang telah dilatih sebelumnya dari hub model populer melalui halaman JumpStart arahan dalam pengalaman Studio yang diperbarui.

Anda juga dapat mengakses model terlatih, templat solusi, dan contoh melalui halaman JumpStart arahan di Amazon SageMaker Studio Classic.

Langkah-langkah berikut menunjukkan cara mengakses JumpStart model menggunakan Amazon SageMaker Studio dan Amazon SageMaker Studio Classic.

Anda juga dapat mengakses JumpStart model menggunakan SageMaker Python SDK. Untuk informasi tentang cara menggunakan JumpStart model secara terprogram, lihat Menggunakan SageMaker JumpStart Algoritma dengan Model Terlatih.

Buka dan gunakan JumpStart di Studio

Bagian berikut memberikan informasi tentang cara membuka, menggunakan, dan mengelola JumpStart dari UI Studio.

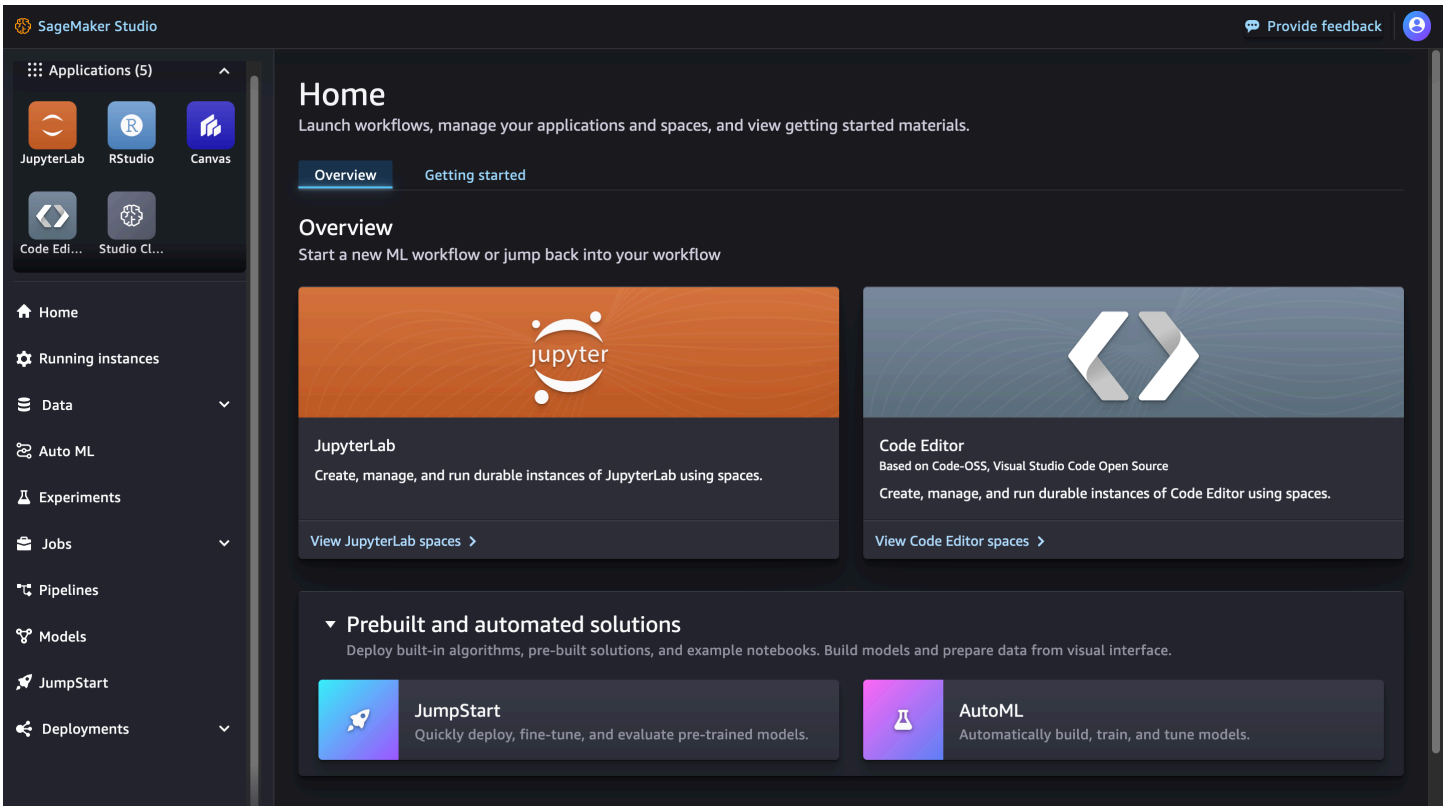
> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Buka JumpStart di Studio

Di Amazon SageMaker Studio, buka halaman JumpStart arahan baik melalui halaman Beranda atau menu Beranda di panel sisi kiri. Ini membuka halaman SageMaker JumpStartarahan tempat Anda dapat menjelajahi hub model dan mencari model.

- Dari halaman Beranda, pilih JumpStartdi panel solusi bawaan dan otomatis.
- Dari menu Beranda di panel kiri, arahkan ke SageMaker JumpStartnode.

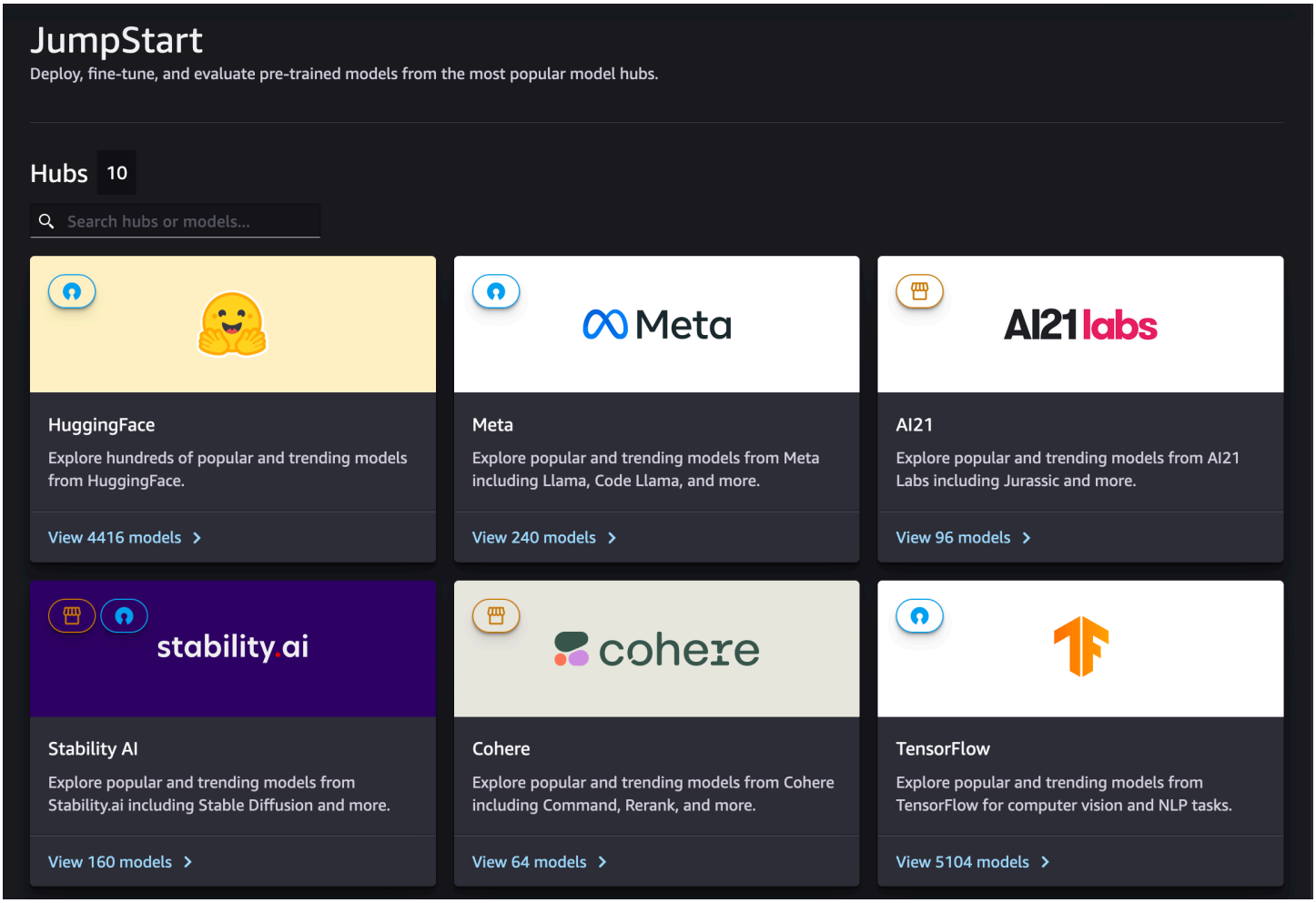
Untuk informasi selengkapnya tentang memulai Amazon SageMaker Studio, lihat SageMaker Studio Amazon.



> **Important**
>
> Sebelum mengunduh atau menggunakan konten pihak ketiga: Anda bertanggung jawab untuk meninjau dan mematuhi persyaratan lisensi yang berlaku dan memastikan bahwa mereka dapat diterima untuk kasus penggunaan Anda.

Gunakan JumpStart di Studio

Dari halaman SageMaker JumpStartarahan di Studio, Anda dapat menjelajahi hub model dari penyedia model eksklusif dan tersedia untuk umum.



Anda dapat menemukan hub atau model tertentu menggunakan bilah pencarian. Dalam setiap hub model, Anda dapat mencari model secara langsung, mengurutkan berdasarkan atribut yang disediakan, atau memfilter berdasarkan daftar tugas model yang disediakan.

Kelola JumpStart di Studio

Pilih model untuk melihat kartu detail modelnya. Di sudut kanan atas kartu detail model, pilih Fine-tune, Deploy, atau Evaluate untuk mulai mengerjakan alur kerja fine-tuning, deployment, atau evaluasi. Perhatikan bahwa tidak semua model tersedia untuk fine-tuning atau evaluasi. Untuk informasi lebih lanjut tentang masing-masing opsi ini, lihatGunakan model pondasi di Studio.

Buka dan gunakan JumpStart di Studio Classic

Bagian berikut memberikan informasi tentang cara membuka, menggunakan, dan mengelola JumpStart dari Amazon SageMaker Studio Classic UI.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Buka JumpStart di Studio Classic

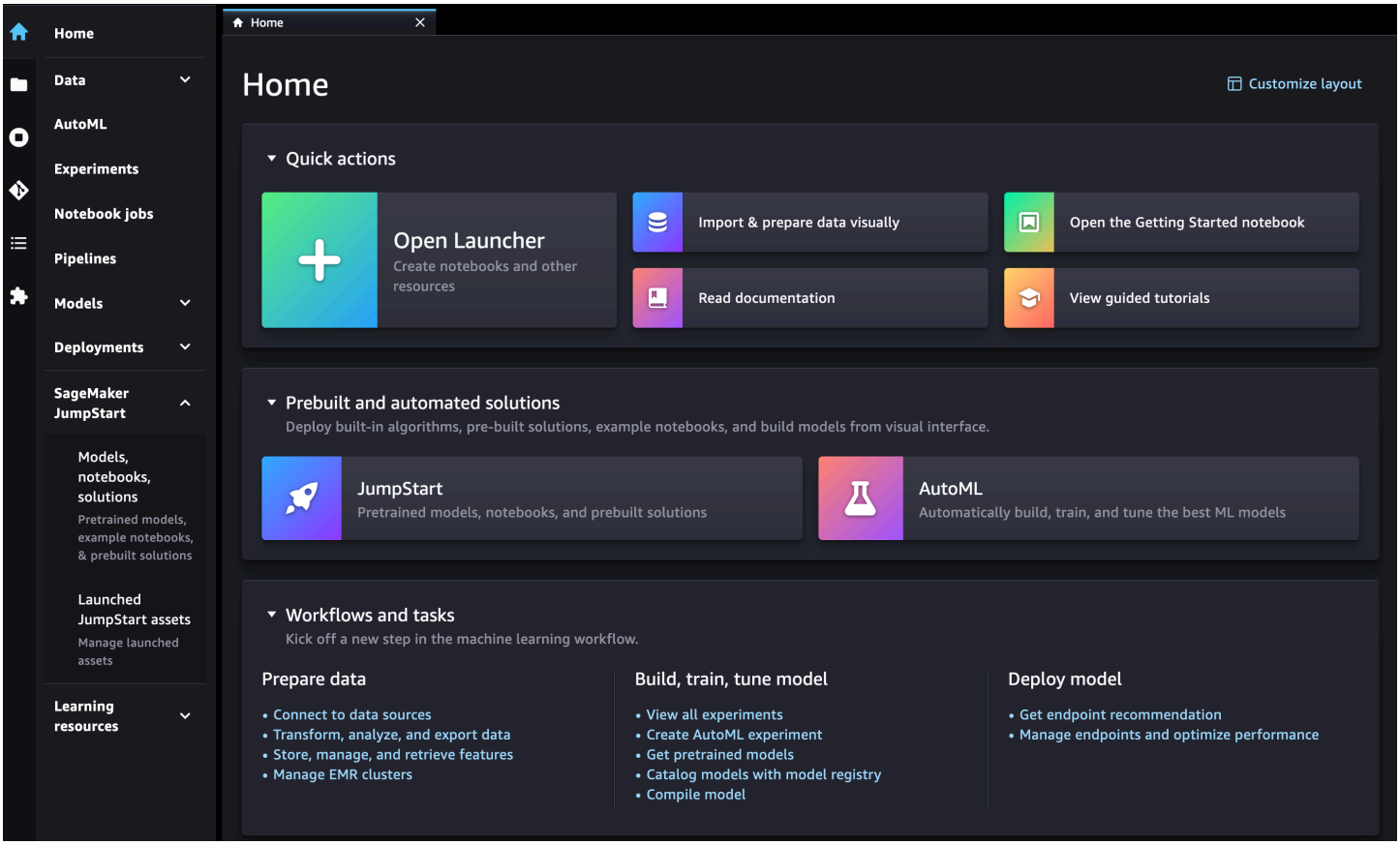
Di Amazon SageMaker Studio Classic, buka halaman JumpStart landing baik melalui halaman Beranda atau menu Beranda di panel sisi kiri.

- Dari halaman Beranda Anda dapat:
  - Pilih JumpStartdi panel solusi bawaan dan otomatis. Ini membuka halaman SageMaker JumpStartarahan.
  - Pilih model langsung di halaman SageMaker JumpStartarahan, atau pilih opsi Jelajahi Semua untuk melihat solusi atau model yang tersedia dari jenis tertentu.
- Dari menu Beranda di panel kiri Anda dapat:
  - Arahkan ke SageMaker JumpStartnode, lalu pilih Model, notebook, solusi. Ini membuka halaman SageMaker JumpStartarahan.
  - Arahkan ke JumpStartnode, lalu pilih JumpStart Aset yang diluncurkan.

    Halaman JumpStart Aset yang Diluncurkan mencantumkan solusi yang saat ini diluncurkan, titik akhir model yang diterapkan, dan pekerjaan pelatihan yang dibuat dengan. JumpStart Anda dapat mengakses halaman JumpStart arahan dari tab ini dengan mengklik JumpStart tombol Browse di kanan atas tab.

Halaman JumpStart arahan mencantumkan solusi pembelajaran end-to-end mesin yang tersedia, model yang telah dilatih sebelumnya, dan contoh buku catatan. Dari setiap solusi individu atau halaman model, Anda dapat memilih JumpStart tombol Browse (

) di kanan atas tab untuk kembali ke SageMaker JumpStarthalaman.

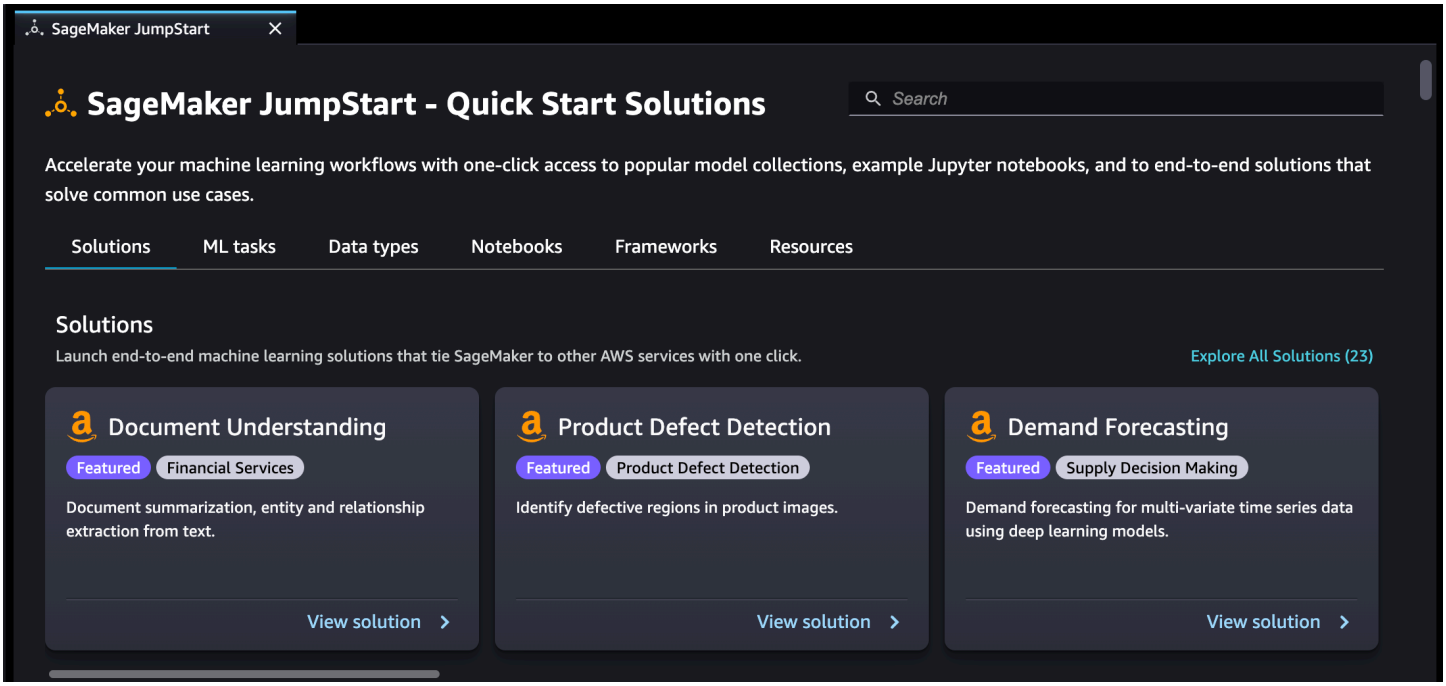


Important

Sebelum mengunduh atau menggunakan konten pihak ketiga: Anda bertanggung jawab untuk meninjau dan mematuhi persyaratan lisensi yang berlaku dan memastikan bahwa mereka dapat diterima untuk kasus penggunaan Anda.

Gunakan JumpStart di Studio Classic

Dari halaman SageMaker JumpStartarahan, Anda dapat menelusuri solusi, model, notebook, dan sumber daya lainnya.



Anda dapat menemukan JumpStart sumber daya dengan menggunakan bilah pencarian, atau dengan menelusuri setiap kategori. Gunakan tab untuk memfilter solusi yang tersedia berdasarkan kategori:

- Solusi — Dalam satu langkah, luncurkan solusi pembelajaran mesin komprehensif yang SageMaker terkait dengan AWS layanan lain. Pilih Jelajahi Semua Solusi untuk melihat semua solusi yang tersedia.
- Sumber Daya — Gunakan contoh buku catatan, blog, dan tutorial video untuk mempelajari dan memulai jenis masalah Anda.
  - Blog — Baca detail dan solusi dari pakar pembelajaran mesin.
  - Tutorial video — Tonton tutorial video untuk SageMaker fitur dan kasus penggunaan pembelajaran mesin dari pakar pembelajaran mesin.
  - Contoh notebook — Jalankan contoh notebook yang menggunakan SageMaker fitur seperti pelatihan dan eksperimen Spot Instance pada berbagai jenis model dan kasus penggunaan.
- Tipe data — Temukan model berdasarkan tipe data (misalnya, Visi, Teks, Tabular, Audio, Pembuatan Teks). Pilih Jelajahi Semua Model untuk melihat semua model yang tersedia.
- Tugas ML — Temukan model berdasarkan jenis masalah (misalnya, Klasifikasi Gambar, Penyematan Gambar, Deteksi Objek, Pembuatan Teks). Pilih Jelajahi Semua Model untuk melihat semua model yang tersedia.

- Notebook — Temukan contoh buku catatan yang menggunakan SageMaker fitur di beberapa jenis model dan kasus penggunaan. Pilih Jelajahi Semua Buku Catatan untuk melihat semua contoh buku catatan yang tersedia.
- Frameworks — Temukan model berdasarkan kerangka kerja (mis.,, PyTorch TensorFlow, Hugging Face).

Kelola JumpStart di Studio Classic

Dari menu Beranda di panel kiri, navigasikan ke SageMaker JumpStart, lalu pilih JumpStart Aset yang diluncurkan untuk mencantumkan solusi yang saat ini diluncurkan, titik akhir model yang diterapkan, dan pekerjaan pelatihan yang dibuat dengan. JumpStart

Topik

- Template Solusi
- JumpStart Model Yayasan
- Model Khusus Tugas
- Model dan Notebook Bersama
- SageMaker JumpStart Industri Amazon: Keuangan

Template Solusi

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

> **Note**
>
> JumpStart Solusi hanya tersedia di Studio Classic.

SageMaker JumpStart menyediakan satu-klik, end-to-end solusi untuk banyak kasus penggunaan pembelajaran mesin yang umum. Jelajahi kasus penggunaan berikut untuk informasi lebih lanjut tentang templat solusi yang tersedia.

- Peramalan permintaan
- Prediksi peringkat kredit
- Deteksi penipuan
- Visi komputer
- Ekstrak dan analisis data dari dokumen
- Pemeliharaan prediktif
- Prediksi Churn
- Rekomendasi yang dipersonalisasi
- Pembelajaran penguatan
- Kesehatan dan ilmu kehidupan
- Harga keuangan
- Inferensi kausal

Pilih template solusi yang paling sesuai dengan kasus penggunaan Anda dari halaman JumpStart arahan. Saat Anda memilih template solusi, JumpStart buka tab baru yang menampilkan deskripsi solusi dan tombol Luncurkan. Saat Anda memilih Launch, JumpStart buat semua sumber daya yang Anda butuhkan untuk menjalankan solusi, termasuk pelatihan dan contoh hosting model. Untuk informasi lebih lanjut tentang meluncurkan JumpStart solusi, lihatthe section called “Luncurkan Solusi”.

Setelah meluncurkan solusi, Anda dapat menjelajahi fitur solusi dan artefak apa pun yang dihasilkan. JumpStart Gunakan menu JumpStart Aset yang diluncurkan untuk menemukan solusi Anda. Di tab solusi, pilih Buka Notebook untuk menggunakan buku catatan yang disediakan dan jelajahi fitur solusinya. Saat artefak dihasilkan selama peluncuran atau setelah menjalankan buku catatan yang disediakan, artefak tersebut tercantum dalam tabel Artefak yang Dihasilkan. Anda dapat menghapus artefak individual dengan ikon sampah
() .
Anda dapat menghapus semua sumber daya solusi dengan memilih Hapus sumber daya solusi.

Peramalan permintaan

Peramalan permintaan menggunakan data deret waktu historis untuk membuat estimasi masa depan dalam kaitannya dengan permintaan pelanggan selama periode tertentu dan merampingkan proses pengambilan keputusan permintaan-penawaran di seluruh bisnis.

Kasus penggunaan perkiraan permintaan termasuk memprediksi penjualan tiket di industri transportasi, harga saham, jumlah kunjungan rumah sakit, jumlah perwakilan pelanggan yang akan dipekerjakan untuk beberapa lokasi di bulan depan, penjualan produk di beberapa wilayah pada kuartal berikutnya, penggunaan server cloud untuk hari berikutnya untuk layanan streaming video, konsumsi listrik untuk beberapa wilayah selama minggu depan, jumlah perangkat dan sensor IoT seperti konsumsi energi, dan banyak lagi.

Data deret waktu dikategorikan sebagai univariat dan multi-variat. Misalnya, total konsumsi listrik untuk satu rumah tangga adalah deret waktu univariat selama periode waktu tertentu. Ketika beberapa deret waktu univariat ditumpuk satu sama lain, itu disebut deret waktu multi-variat. Misalnya, total konsumsi listrik dari 10 rumah tangga yang berbeda (tetapi berkorelasi) dalam satu lingkungan membentuk kumpulan data deret waktu multi-variat.

Nama solusi	Deskripsi	Memulai
Peramalan permintaan	Peramalan permintaan untuk data deret waktu multivariat menggunakan tiga algoritma peramalan deret state-of-the-art waktu: LSTNet, Prophet, dan DeepAR. SageMaker	GitHub »

Prediksi peringkat kredit

Solusi prediksi peringkat kredit Use JumpStart untuk memprediksi peringkat kredit perusahaan atau untuk menjelaskan keputusan prediksi kredit yang dibuat oleh model pembelajaran mesin. Dibandingkan dengan metode pemodelan peringkat kredit tradisional, model pembelajaran mesin dapat mengotomatiskan dan meningkatkan akurasi prediksi kredit.

Nama solusi	Deskripsi	Memulai
Prediksi peringkat kredit perusahaan	Pembelajaran mesin multimodal (teks panjang dan tabular) untuk prediksi kredit berkualitas menggunakan Tabular. AWS AutoGluon	GitHub »
Penilaian kredit berbasis grafik	Memprediksi peringkat kredit perusahaan menggunakan data tabular dan jaringan perusahaan dengan melatih Graph Neural Network GraphSage dan model AWS AutoGluon Tabular.	Temukan di Amazon SageMaker Studio Classic.
Jelaskan keputusan kredit	Memprediksi default kredit dalam aplikasi kredit dan memberikan penjelasan menggunakan LightGBM dan SHAP (Shapley Additive Explanations).	GitHub »

Deteksi penipuan

Banyak bisnis kehilangan miliaran per tahun karena penipuan. Model deteksi penipuan berbasis pembelajaran mesin dapat membantu secara sistematis mengidentifikasi kemungkinan aktivitas penipuan dari sejumlah besar data. Solusi berikut menggunakan kumpulan data transaksi dan identitas pengguna untuk mengidentifikasi transaksi penipuan.

Nama solusi	Deskripsi	Memulai
Mendeteksi pengguna dan transaksi jahat	Secara otomatis mendeteksi aktivitas yang berpotensi penipuan dalam transaksi menggunakan SageMaker XGBoost dengan teknik over-	GitHub »

Nama solusi	Deskripsi	Memulai
	sampling Synthetic Minority Over-sampling (SMOTE).	
Deteksi penipuan dalam transaksi keuangan menggunakan Deep Graph Library	Deteksi penipuan dalam transaksi keuangan dengan melatih jaringan convolutional grafik dengan pustaka grafik dalam dan model XGBoost. SageMaker	GitHub »
Klasifikasi pembayaran keuangan	Klasifikasi pembayaran keuangan berdasarkan informasi transaksi menggunakan SageMaker XGBoost. Gunakan template solusi ini sebagai langkah perantara dalam deteksi penipuan, personalisasi, atau deteksi anomali.	Temukan di Amazon SageMaker Studio Classic.

Visi komputer

Dengan munculnya kasus penggunaan bisnis seperti kendaraan otonom, pengawasan video pintar, pemantauan perawatan kesehatan dan berbagai tugas penghitungan objek, sistem deteksi objek yang cepat dan akurat meningkat dalam permintaan. Sistem ini melibatkan tidak hanya mengenali dan mengklasifikasikan setiap objek dalam gambar, tetapi melokalisasi masing-masing dengan menggambar kotak pembatas yang sesuai di sekitarnya. Dalam dekade terakhir, kemajuan pesat teknik pembelajaran mendalam sangat mempercepat momentum deteksi objek.

Nama solusi	Deskripsi	Memulai
Deteksi cacat produk visual	Identifikasi daerah yang rusak dalam gambar produk baik dengan melatih model deteksi objek dari awal atau	GitHub »

Nama solusi	Deskripsi	Memulai
	menyempurnakan model yang telah dilatih sebelumnya. SageMaker	
Pengenalan tulisan tangan	Kenali teks tulisan tangan dalam gambar dengan melatih model deteksi objek dan model pengenalan tulisan tangan. Beri label data Anda sendiri menggunakan SageMaker Ground Truth.	GitHub »
Deteksi objek untuk spesies burung	Identifikasi spesies burung dalam sebuah adegan menggunakan model deteksi SageMaker objek.	Temukan di Amazon SageMaker Studio Classic.

Ekstrak dan analisis data dari dokumen

JumpStart memberikan solusi bagi Anda untuk mengungkap wawasan dan koneksi berharga dalam dokumen penting bisnis. Kasus penggunaan termasuk klasifikasi teks, ringkasan dokumen, pengenalan tulisan tangan, ekstraksi hubungan, pertanyaan dan jawaban, dan mengisi nilai yang hilang dalam catatan tabel.

Nama solusi	Deskripsi	Memulai
Privasi untuk klasifikasi sentimen	Menganonimkan teks untuk menjaga privasi pengguna dalam klasifikasi sentimen dengan lebih baik.	GitHub »
Pemahaman dokumen	Ringkasan dokumen, entitas, dan ekstraksi hubungan menggunakan perpustakaan transformer di. PyTorch	GitHub »

Nama solusi	Deskripsi	Memulai
Pengenalan tulisan tangan	Kenali teks tulisan tangan dalam gambar dengan melatih model deteksi objek dan model pengenalan tulisan tangan. Beri label data Anda sendiri menggunakan SageMaker Ground Truth.	GitHub »
Mengisi nilai yang hilang dalam catatan tabel	Isi nilai yang hilang dalam catatan tabel dengan melatih SageMaker AutoPilotmodel.	GitHub »

Pemeliharaan prediktif

Pemeliharaan prediktif bertujuan untuk mengoptimalkan keseimbangan antara pemeliharaan korektif dan preventif dengan memfasilitasi penggantian komponen secara tepat waktu. Solusi berikut menggunakan data sensor dari aset industri untuk memprediksi kegagalan alat berat, waktu henti yang tidak direncanakan, dan biaya perbaikan.

Nama solusi	Deskripsi	Memulai
Perawatan prediktif untuk armada kendaraan	Memprediksi kegagalan armada kendaraan menggunakan sensor kendaraan dan informasi pemeliharaan dengan model jaringan saraf convolutional.	GitHub »
Pemeliharaan prediktif untuk manufaktur	Memprediksi masa manfaat yang tersisa untuk setiap sensor dengan melatih model jaringan saraf LSTM Dua Arah yang ditumpuk menggunakan pembacaan sensor historis.	GitHub »

Prediksi Churn

Churn pelanggan, atau tingkat gesekan, adalah masalah mahal yang dihadapi oleh berbagai perusahaan. Dalam upaya untuk mengurangi churn, perusahaan dapat mengidentifikasi pelanggan yang cenderung meninggalkan layanan mereka untuk memfokuskan upaya mereka pada retensi pelanggan. Gunakan solusi prediksi JumpStart churn untuk menganalisis sumber data seperti perilaku pengguna dan log obrolan dukungan pelanggan untuk mengidentifikasi pelanggan yang berisiko tinggi membatalkan langganan atau layanan.

Nama solusi	Deskripsi	Memulai
Prediksi Churn dengan teks	Memprediksi churn menggunakan fitur numerik, kategoris, dan tekstual dengan encoder BERT dan. RandomForestClassifier	GitHub »
Prediksi Churn untuk pelanggan ponsel	Identifikasi pelanggan ponsel yang tidak senang menggunakan SageMaker XGBoost.	Temukan di Amazon SageMaker Studio Classic.

Rekomendasi yang dipersonalisasi

Anda dapat menggunakan JumpStart solusi untuk menganalisis grafik identitas pelanggan atau sesi pengguna untuk lebih memahami dan memprediksi perilaku pelanggan. Gunakan solusi berikut untuk rekomendasi yang dipersonalisasi untuk memodelkan identitas pelanggan di beberapa perangkat, untuk menentukan kemungkinan pelanggan melakukan pembelian, atau untuk membuat rekomendasi film khusus berdasarkan perilaku pelanggan sebelumnya.

Nama solusi	Deskripsi	Memulai
Resolusi entitas dalam grafik identitas dengan pustaka grafik dalam	Lakukan penautan entitas lintas perangkat untuk iklan online dengan melatih jaringan convolutional grafik dengan pustaka grafik dalam.	GitHub »

Nama solusi	Deskripsi	Memulai
Pemodelan pembelian	Memprediksi apakah pelanggan akan melakukan pembelian dengan melatih model SageMaker XGBoost.	GitHub »
Sistem rekomendasi yang disesuaikan	Latih dan terapkan sistem pemberi rekomendasi khusus yang menghasilkan saran film untuk pelanggan berdasarkan perilaku masa lalu menggunakan Neural Collaborative Filtering in. SageMaker	Temukan di Amazon SageMaker Studio Classic.

Pembelajaran penguatan

Pembelajaran penguatan (RL) adalah jenis pembelajaran yang didasarkan pada interaksi dengan lingkungan. Jenis pembelajaran ini digunakan oleh agen yang harus mempelajari perilaku melalui trial-and-error interaksi dengan lingkungan yang dinamis di mana tujuannya adalah untuk memaksimalkan imbalan jangka panjang yang diterima agen sebagai hasil dari tindakannya. Hadiah dimaksimalkan dengan menukar tindakan penjelajahan yang memiliki imbalan yang tidak pasti dengan mengeksploitasi tindakan yang memiliki imbalan yang diketahui.

RL sangat cocok untuk memecahkan masalah besar dan kompleks, seperti manajemen rantai pasokan, sistem HVAC, robotika industri, kecerdasan buatan game, sistem dialog, dan kendaraan otonom.

Nama solusi	Deskripsi	Memulai
Pembelajaran penguatan untuk kompetisi AI Battlesnake	Berikan alur kerja pembelajaran penguatan untuk pelatihan dan inferensi dengan kompetisi BattleSnakeAI.	GitHub »

Nama solusi	Deskripsi	Memulai
Pembelajaran penguatan terdistribusi untuk tantangan Procgen	Kit starter pembelajaran penguatan terdistribusi untuk tantangan pembelajaran NeurIPs 2020 Procgen Reinforcement.	GitHub »

Kesehatan dan ilmu kehidupan

Dokter dan peneliti dapat menggunakan JumpStart solusi untuk menganalisis citra medis, informasi genom, dan catatan kesehatan klinis.

Nama solusi	Deskripsi	Memulai
Prediksi kelangsungan hidup kanker paru-paru	Memprediksi status kelangsungan hidup pasien kanker paru-paru non-sel kecil dengan pemindaian tomografi terkomputerisasi paru (CT) 3 dimensi, data genom, dan catatan kesehatan klinis menggunakan XGBoost. SageMaker	GitHub »

Harga keuangan

Banyak bisnis secara dinamis menyesuaikan harga secara teratur untuk memaksimalkan pengembalian mereka. Gunakan JumpStart solusi berikut untuk pengoptimalan harga, penetapan harga dinamis, penetapan harga opsi, atau kasus penggunaan pengoptimalan portofolio.

Nama solusi	Deskripsi	Memulai
Optimalisasi harga	Estimasi elastisitas harga menggunakan Double Machine Learning (ML) untuk	Temukan di Amazon SageMaker Studio Classic.

Nama solusi	Deskripsi	Memulai
	inferensi kausal dan prosedur peramalan Nabi. Gunakan perkiraan ini untuk mengoptim alkan harga harian.	

Inferensi kausal

Peneliti dapat menggunakan model pembelajaran mesin seperti jaringan Bayesian untuk mewakili ketergantungan kausal dan menarik kesimpulan kausal berdasarkan data. Gunakan JumpStart solusi berikut untuk memahami hubungan sebab akibat antara aplikasi pupuk berbasis nitrogen dan hasil tanaman jagung.

Nama solusi	Deskripsi	Memulai
Kontrafaktual hasil panen	Hasilkan analisis kontrafak tual respons jagung terhadap nitrogen. Solusi ini mempelajari siklus fenologi tanaman secara keseluruh an menggunakan citra satelit multi-spektral dan pengamata n di permukaan tanah.	Temukan di Amazon SageMaker Studio Classic.

Luncurkan Solusi

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

> **Note**
>
> JumpStart Solusi hanya tersedia di Studio Classic.

Pertama, pilih solusi melalui halaman SageMaker JumpStart arahan di Amazon SageMaker Studio Classic UI. Untuk informasi tentang langkah-langkah orientasi untuk masuk ke Amazon SageMaker Studio Classic, lihat Onboard to Amazon SageMaker Domain. Untuk detail tentang membuka halaman SageMaker JumpStart arahan, lihatBuka dan gunakan JumpStart di Studio Classic.

Setelah Anda memilih solusi, tab solusi terbuka menunjukkan deskripsi solusi dan `Launch` tombol. Untuk meluncurkan solusi, pilih `Launch` di bagian Launch Solution. JumpStart kemudian membuat semua sumber daya yang dibutuhkan untuk menjalankan solusi. Ini termasuk pelatihan dan contoh hosting model.

Parameter lanjutan

Solusi yang Anda pilih mungkin memiliki parameter lanjutan yang dapat Anda pilih. Pilih Parameter Lanjutan untuk menentukan AWS Identity and Access Management peran solusi.

Solusi dapat meluncurkan sumber daya di 9 AWS layanan yang berinteraksi satu sama lain. Agar solusi berfungsi seperti yang diharapkan, komponen yang baru dibuat dari satu layanan harus dapat bertindak pada komponen yang baru dibuat dari layanan lain. Kami menyarankan Anda menggunakan peran IAM default untuk memastikan bahwa semua izin yang diperlukan ditambahkan. Untuk informasi selengkapnya tentang peran IAM, lihatIdentity and Access Management untuk Amazon SageMaker.

Peran IAM default

Jika Anda memilih opsi ini, peran IAM default yang diperlukan oleh solusi ini akan digunakan. Setiap solusi membutuhkan sumber daya yang berbeda. Daftar berikut menjelaskan peran default yang digunakan untuk solusi berdasarkan layanan yang dibutuhkan. Untuk deskripsi tentang izin yang diperlukan untuk setiap layanan, lihatAWSKebijakan Terkelola untuk SageMaker proyek dan JumpStart.

- API Gateway — AmazonSageMakerServiceCatalogProductsApiGatewayRole
- CloudFormation— AmazonSageMakerServiceCatalogProductsCloudformation Peran
- CodeBuild – AmazonSageMakerServiceCatalogProductsCodeBuildRole
- CodePipeline – AmazonSageMakerServiceCatalogProductsCodePipelineRole

- Acara - AmazonSageMakerServiceCatalogProductsEvents Peran
- Firehose - Peran AmazonSageMakerServiceCatalogProductsFirehose
- Glue - AmazonSageMakerServiceCatalogProductsGlue Peran
- Lambda — Peran AmazonSageMakerServiceCatalogProductsLambda
- SageMaker— AmazonSageMakerServiceCatalogProductsExecution Peran

Jika Anda menggunakan SageMaker Domain baru dengan templat JumpStart proyek diaktifkan, peran ini secara otomatis dibuat di akun Anda.

Jika Anda menggunakan SageMaker domain yang ada, peran ini mungkin tidak ada di akun Anda. Jika ini masalahnya, Anda akan menerima kesalahan berikut saat meluncurkan solusi.

```
Unable to locate the updated roles required to launch this solution, a general role '/
service-role/AmazonSageMakerServiceCatalogProductsUseRole' will be used. Please update
 your studio domain to generate these roles.
```

Anda masih dapat meluncurkan solusi tanpa peran yang diperlukan, tetapi peran default lama `AmazonSageMakerServiceCatalogProductsUseRole` digunakan sebagai pengganti peran yang diperlukan. Peran default lama memiliki hubungan kepercayaan dengan semua layanan yang perlu berinteraksi dengan JumpStart solusi. Untuk keamanan terbaik, sebaiknya Anda memperbarui domain agar memiliki peran default yang baru dibuat untuk setiap AWS layanan.

Jika Anda sudah onboard ke SageMaker domain, Anda dapat memperbarui domain Anda untuk menghasilkan peran default menggunakan prosedur berikut.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih Control Panel di kiri atas halaman.
3. Dari halaman Domain, pilih ikon Pengaturan
   (⚙)
   untuk mengedit pengaturan domain.
4. Pada Pengaturan Umum pilih Berikutnya.
5. Di bawah SageMaker Proyek dan JumpStart, pilih Aktifkan templat SageMaker proyek Amazon dan Amazon SageMaker JumpStart untuk akun ini dan Aktifkan templat SageMaker proyek Amazon dan Amazon SageMaker JumpStart untuk pengguna Studio Classic, pilih Berikutnya.
6. Pilih Kirim.

Anda harus dapat melihat peran default yang tercantum dalam Proyek - Templat SageMaker proyek Amazon diaktifkan untuk akun ini di bawah tab Apps - Studio.

Temukan peran IAM

Jika Anda memilih opsi ini, Anda harus memilih peran IAM yang ada dari daftar tarik-turun untuk setiap layanan yang diperlukan. Peran yang dipilih harus memiliki setidaknya izin minimum yang diperlukan untuk layanan terkait. Untuk deskripsi tentang izin yang diperlukan untuk setiap layanan, lihatAWSKebijakan Terkelola untuk SageMaker proyek dan JumpStart.

Masukan peran IAM

Jika Anda memilih opsi ini, Anda harus memasukkan ARN secara manual untuk peran IAM yang ada. Peran yang dipilih harus memiliki setidaknya izin minimum yang diperlukan untuk layanan terkait. Untuk deskripsi tentang izin yang diperlukan untuk setiap layanan, lihatAWSKebijakan Terkelola untuk SageMaker proyek dan JumpStart.

JumpStart Model Yayasan

Amazon SageMaker JumpStart menawarkan model state-of-the-art dasar untuk kasus penggunaan seperti penulisan konten, pembuatan kode, penjawab pertanyaan, copywriting, ringkasan, klasifikasi, pengambilan informasi, dan banyak lagi. Gunakan model JumpStart foundation untuk membuat solusi AI generatif Anda sendiri dan mengintegrasikan solusi khusus dengan SageMaker fitur tambahan. Untuk informasi selengkapnya, lihat Memulai Amazon SageMaker JumpStart.

Model pondasi adalah model pra-terlatih besar yang dapat beradaptasi dengan banyak tugas hilir dan sering berfungsi sebagai titik awal untuk mengembangkan model yang lebih khusus. Contoh model pondasi termasuk L LaMa -2-7b, BLOOM 176B, FLAN-T5 XL, atau GPT-J 6B, yang telah dilatih sebelumnya pada sejumlah besar data teks dan dapat disetel dengan baik untuk tugas bahasa tertentu.

SageMaker JumpStart Amazon menggunakan dan mengelola model dasar yang tersedia untuk umum agar Anda dapat mengakses, menyesuaikan, dan mengintegrasikan ke dalam siklus hidup pembelajaran mesin Anda. Untuk informasi selengkapnya, lihat Model pondasi yang tersedia untuk umum. Amazon SageMaker JumpStart juga menyertakan model yayasan berpemilik dari penyedia pihak ketiga. Untuk informasi selengkapnya, lihat Model pondasi berpemilik.

Untuk mulai menjelajahi dan bereksperimen dengan model yang tersedia, lihatCara menggunakan model JumpStart pondasi. Semua model foundation tersedia untuk digunakan secara terprogram

dengan Python SageMaker SDK. Untuk informasi selengkapnya, lihat Gunakan model foundation dengan SageMaker Python SDK.

Untuk informasi lebih lanjut tentang pertimbangan yang harus dibuat saat memilih model, lihatPilih model pondasi.

Untuk informasi spesifik tentang model pondasi kustomisasi dan fine-tuning, lihat. Sesuaikan model pondasi

Untuk informasi lebih umum tentang model pondasi, lihat paper On the Opportunities and Risks of Foundation Models.

Jelajahi model foundation terbaru

Amazon SageMaker JumpStart menawarkan state-of-the-art, model dasar berpemilik bawaan dan tersedia untuk umum untuk menyesuaikan dan mengintegrasikan ke dalam alur kerja AI generatif Anda.

Model pondasi berpemilik

Amazon SageMaker JumpStart menyediakan akses ke model yayasan berpemilik dari penyedia pihak ketiga seperti AI21 Labs, Cohere, dan. LightOn

Untuk memulai dengan salah satu model eksklusif ini, lihatCara menggunakan model JumpStart pondasi. Untuk menggunakan model pondasi berpemilik, Anda harus terlebih dahulu berlangganan model diAWS Marketplace. Setelah berlangganan model, cari model pondasi di Studio atau SageMaker Studio Classic. Untuk informasi selengkapnya, lihat SageMaker JumpStart.

Untuk menjelajahi model foundation berpemilik terbaru untuk berbagai kasus penggunaan, lihat Memulai Amazon SageMaker JumpStart.

Model pondasi yang tersedia untuk umum

Amazon melakukan SageMaker JumpStart onboard dan memelihara model foundation open source dari sumber pihak ketiga. Untuk memulai dengan salah satu model yang tersedia untuk umum ini, lihat Cara menggunakan model JumpStart pondasi atau jelajahi salah satu yang tersediaNotebook contoh. Dalam contoh notebook tertentu untuk model yang tersedia untuk umum, coba ganti ID model untuk bereksperimen dengan model yang berbeda dalam keluarga model yang sama.

Untuk informasi selengkapnya tentang ID model dan sumber daya tentang penerapan model JumpStart dasar yang tersedia untuk umum dengan SageMaker Python SDK, lihat. Gunakan model foundation yang tersedia untuk umum dengan SageMaker Python SDK

Notebook contoh

Untuk step-by-step contoh tentang cara menggunakan model JumpStart dasar yang tersedia untuk umum dengan SageMaker Python SDK, lihat buku catatan berikut tentang pembuatan teks, pembuatan gambar, dan penyesuaian model.

> **Note**
>
> Model JumpStart foundation eksklusif dan tersedia untuk umum memiliki alur kerja penerapan SageMaker Python SDK yang berbeda. Temukan contoh notebook model foundation berpemilik melalui Amazon SageMaker Studio Classic atau konsol. SageMaker Untuk informasi selengkapnya, lihat Cara menggunakan model JumpStart pondasi.

Anda dapat mengkloning repositori SageMaker contoh Amazon untuk menjalankan contoh model JumpStart dasar yang tersedia di lingkungan Jupyter pilihan Anda dalam Studio. Untuk informasi selengkapnya tentang aplikasi yang dapat Anda gunakan untuk membuat dan mengakses Jupyter SageMaker, lihat. Aplikasi yang didukung di Amazon SageMaker Studio

Pembuatan teks

Jelajahi contoh buku catatan pembuatan teks, termasuk panduan tentang alur kerja pembuatan teks umum, klasifikasi teks multibahasa, inferensi batch waktu nyata, pembelajaran beberapa bidikan, interaksi chatbot, dan banyak lagi.

- SageMaker JumpStart Model Foundation - HuggingFace Text2Text Generation dengan FLAN-T5 XL sebagai contoh
- SageMaker JumpStart Model Foundation - BloomZ: Klasifikasi Teks Multilingual, Pertanyaan dan Jawaban, Pembuatan Kode, Pengungkapan ulang Paragraf, dan Lainnya
- SageMaker JumpStart Model Foundation - Transformasi Batch Generasi HuggingFace Teks Text2Text dan Inferensi Batch Real-Time
- SageMaker JumpStart Model Foundation - Pembelajaran GPT-J, GPT-neo Few-shot
- SageMaker JumpStart Model Yayasan - Chatbots

Pembuatan gambar

Mulailah dengan model Difusi text-to-image Stabil, pelajari cara menerapkan model inpainting, dan bereksperimen dengan alur kerja sederhana untuk menghasilkan gambar anjingmu.

- Pengantar JumpStart - Teks ke Gambar
- Pengantar Pengeditan JumpStart Gambar - Inpainting Difusi Stabil
- Hasilkan gambar menyenangkan dari anjingmu

Kustomisasi model

Terkadang kasus penggunaan Anda memerlukan kustomisasi model fondasi yang lebih besar untuk tugas-tugas tertentu. Untuk informasi selengkapnya tentang pendekatan penyesuaian model, lihat Sesuaikan model pondasi atau jelajahi salah satu contoh buku catatan berikut.

- SageMaker JumpStart Model Foundation - Model GPT-J 6B generasi teks penyetelan halus pada kumpulan data khusus domain
- SageMaker JumpStart Model Foundation - HuggingFace Text2Text Instruksi Fine-Tuning
- Retrieval-Augmented Generation: Menjawab Pertanyaan berdasarkan Dataset Kustom
- Retrieval-Augmented Generation: Menjawab Pertanyaan berdasarkan Dataset Kustom dengan Perpustakaan Sumber Terbuka LangChain

Cara menggunakan model JumpStart pondasi

Pilih, latih, atau terapkan model foundation melalui Amazon SageMaker Studio atau Amazon SageMaker Studio Classic, gunakan model JumpStart foundation secara terprogram dengan SageMaker Python SDK, atau temukan model JumpStart foundation langsung melalui konsol. SageMaker

Gunakan model pondasi di Studio

Anda dapat menyempurnakan, menerapkan, dan mengevaluasi model JumpStart foundation eksklusif dan tersedia untuk umum secara langsung melalui Amazon Studio UI. SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Di Amazon SageMaker Studio, buka halaman JumpStart arahan baik melalui halaman Beranda atau menu Beranda di panel sisi kiri. Ini membuka halaman SageMaker JumpStartarahan tempat Anda dapat menjelajahi hub model dan mencari model.

- Dari halaman Beranda, pilih JumpStartdi panel solusi bawaan dan otomatis.
- Dari menu Beranda di panel kiri, arahkan ke SageMaker JumpStartnode.

Untuk informasi selengkapnya tentang memulai Amazon SageMaker Studio, lihat SageMaker Studio Amazon.

Dari halaman SageMaker JumpStartarahan di Studio, Anda dapat menjelajahi hub model dari penyedia model eksklusif dan tersedia untuk umum. Anda dapat menemukan hub atau model tertentu menggunakan bilah pencarian. Dalam setiap hub model, Anda dapat mencari model secara langsung, mengurutkan berdasarkan Paling suka, Paling Banyak unduhan, atau Baru diperbarui, atau memfilter berdasarkan daftar tugas model yang disediakan. Pilih model untuk melihat kartu detail modelnya. Di sudut kanan atas kartu detail model, pilih Fine-tune, Deploy, atau Evaluate untuk mulai mengerjakan alur kerja fine-tuning, deployment, atau evaluasi. Perhatikan bahwa tidak semua model tersedia untuk fine-tuning atau evaluasi. Jika Anda membutuhkan panduan dalam memilih model, lihatPilih model pondasi.

Sempurnakan model pondasi di Studio

Fine-tuning melatih model yang telah dilatih sebelumnya pada dataset baru tanpa pelatihan dari awal. Proses ini, juga dikenal sebagai pembelajaran transfer, dapat menghasilkan model yang akurat dengan kumpulan data yang lebih kecil dan waktu pelatihan yang lebih sedikit. Untuk menyempurnakan model JumpStart foundation, navigasikan ke kartu detail model di UI Studio. Untuk informasi selengkapnya tentang cara membuka JumpStart di Studio, lihatBuka dan gunakan JumpStart di Studio. Setelah menavigasi ke kartu detail model pilihan Anda, pilih Fine-tune di sudut kanan atas. Perhatikan bahwa tidak semua model memiliki fine-tuning yang tersedia.

> **Important**
>
> Beberapa model foundation memerlukan penerimaan eksplisit dari perjanjian lisensi pengguna akhir (EULA) sebelum fine-tuning. Untuk informasi selengkapnya, lihat Penerimaan EULA di Amazon Studio SageMaker .

Pengaturan model

Saat menggunakan model JumpStart pondasi terlatih di Amazon SageMaker Studio, lokasi artefak Model (URI S3) diisi secara default. Untuk mengedit URI S3 default, pilih Masukkan lokasi artefak model.

Pengaturan data

Di bidang Data, berikan titik URI S3 ke lokasi kumpulan data pelatihan Anda. Untuk mengedit URI S3 default, pilih Masukkan lokasi kumpulan data pelatihan model. Pastikan untuk meninjau kartu detail model di Amazon SageMaker Studio untuk informasi tentang memformat data pelatihan.

Hyperparameter

Anda dapat menyesuaikan hyperparameters dari pekerjaan pelatihan yang digunakan untuk menyempurnakan model. Hiperparameter yang tersedia untuk setiap model yang dapat disetel berbeda tergantung pada modelnya.

Hyperparameter berikut umum di antara model:

- Epoch — Satu epoch adalah satu siklus melalui seluruh dataset. Beberapa interval menyelesaikan batch, dan beberapa batch akhirnya menyelesaikan sebuah epoch. Beberapa zaman dijalankan hingga keakuratan model mencapai tingkat yang dapat diterima, atau ketika tingkat kesalahan turun di bawah tingkat yang dapat diterima.
- Laju belajar — Jumlah nilai yang harus diubah antar zaman. Saat model disempurnakan, bobot internalnya didorong dan tingkat kesalahan diperiksa untuk melihat apakah model membaik. Tingkat pembelajaran tipikal adalah 0,1 atau 0,01, di mana 0,01 adalah penyesuaian yang jauh lebih kecil dan dapat menyebabkan pelatihan memakan waktu lama untuk bertemu, sedangkan 0,1 jauh lebih besar dan dapat menyebabkan pelatihan melampaui batas. Ini adalah salah satu hiperparameter utama yang mungkin Anda sesuaikan untuk melatih model Anda. Perhatikan bahwa untuk model teks, tingkat pembelajaran yang jauh lebih kecil (5e-5 untuk BERT) dapat menghasilkan model yang lebih akurat.
- Ukuran Batch — Jumlah catatan dari kumpulan data yang akan dipilih untuk setiap interval untuk dikirim ke GPU untuk pelatihan.

Tinjau petunjuk tip alat dan informasi tambahan dalam kartu detail model di UI Studio untuk mempelajari lebih lanjut tentang hiperparameter khusus untuk model pilihan Anda.

Pengaturan instans

Tentukan jenis instans pelatihan dan lokasi artefak keluaran untuk pekerjaan pelatihan Anda. Anda hanya dapat memilih dari instance yang kompatibel dengan model pilihan Anda dalam menyempurnakan UI Studio.

Informasi tambahan

Di bidang Informasi Tambahan, Anda dapat mengedit nama pekerjaan pelatihan. Anda juga dapat menambah dan menghapus tag dalam bentuk pasangan nilai kunci untuk membantu mengatur dan mengkategorikan pekerjaan pelatihan fine-tuning Anda.

Setelah memberikan informasi untuk konfigurasi fine-tuning Anda, pilih Kirim. Jika model foundation terlatih yang Anda pilih untuk disempurnakan memerlukan persetujuan eksplisit dari perjanjian lisensi pengguna akhir (EULA) sebelum pelatihan, EULA disediakan di jendela pop-up. Untuk menerima ketentuan EULA, pilih Terima. Anda bertanggung jawab untuk meninjau dan mematuhi persyaratan lisensi yang berlaku dan memastikannya dapat diterima untuk kasus penggunaan Anda sebelum mengunduh atau menggunakan model.

Menyebarkan model pondasi di Studio

Untuk menerapkan model JumpStart foundation, navigasikan ke kartu detail model di UI Studio. Untuk informasi selengkapnya tentang cara membuka JumpStart di Studio, lihatBuka dan gunakan JumpStart di Studio. Setelah menavigasi ke halaman detail model pilihan Anda, pilih Deploy di sudut kanan atas UI Studio. Kemudian, ikuti langkah-langkah di Deploy model dengan SageMaker Studio.

> **Important**
>
> Beberapa model dasar memerlukan penerimaan eksplisit dari perjanjian lisensi pengguna akhir (EULA) sebelum penerapan. Untuk informasi selengkapnya, lihat Penerimaan EULA di Amazon Studio SageMaker .

Evaluasi model pondasi di Studio

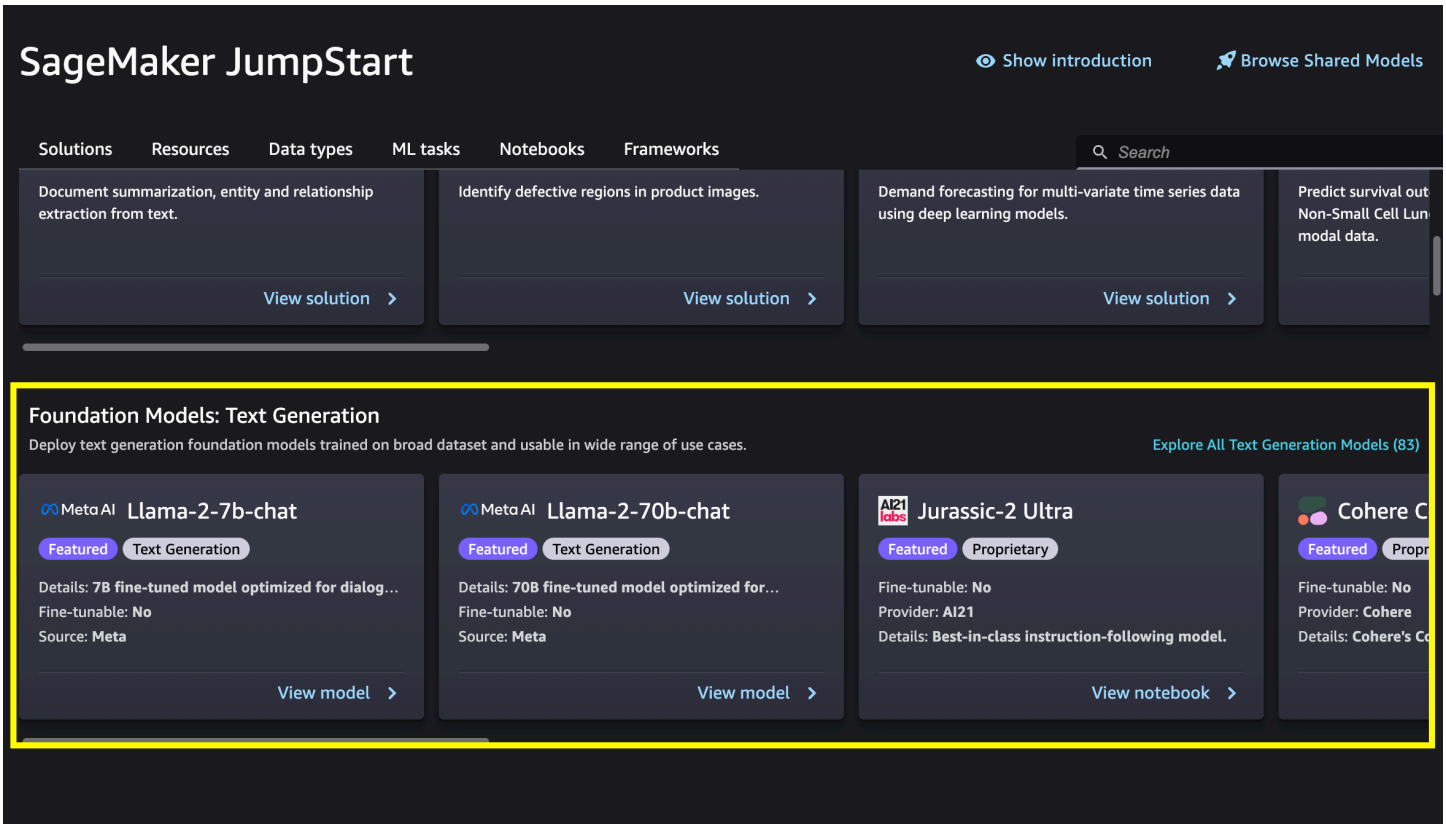
Amazon SageMaker JumpStart memiliki integrasi dengan SageMaker Clarify foundation model evaluations (FME) di Studio. Jika JumpStart model memiliki kemampuan evaluasi bawaan yang tersedia, Anda dapat memilih Evaluasi di sudut kanan atas halaman detail model di UI JumpStart Studio. Untuk informasi lebih lanjut, lihat Mengevaluasi model pondasi.

Gunakan model foundation di Amazon SageMaker Studio Classic

Anda dapat menerapkan, melatih, dan menyempurnakan model JumpStart foundation milik dan tersedia untuk umum melalui UI Studio Classic.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Untuk memulai dengan Studio Classic, lihatLuncurkan Amazon SageMaker Studio Classic.



Setelah membuka Amazon SageMaker Studio Classic, pilih Model, notebook, solusi di SageMaker JumpStart bagian panel navigasi. Kemudian, gulir ke bawah untuk menemukan bagian Foundation Models: Text Generation atau Foundation Models: Image Generation tergantung pada kasus penggunaan Anda.

Anda dapat memilih Lihat model pada kartu model foundation yang disarankan, atau pilih Jelajahi Semua Model untuk melihat semua model dasar yang tersedia baik untuk pembuatan teks atau pembuatan gambar. Jika Anda memilih untuk melihat semua model yang tersedia, Anda dapat memfilter model yang tersedia lebih lanjut berdasarkan tugas, tipe data, jenis konten, atau kerangka kerja. Anda juga dapat mencari nama model langsung di bilah Pencarian. Jika Anda membutuhkan panduan dalam memilih model, lihatPilih model pondasi.

> **Important**
>
> Beberapa model dasar memerlukan penerimaan eksplisit dari perjanjian lisensi pengguna akhir (EULA). Untuk informasi selengkapnya, lihat Penerimaan EULA di Amazon Studio SageMaker .

Setelah Anda memilih model View untuk model dasar pilihan Anda di Studio Classic, Anda dapat menerapkan model. Untuk informasi selengkapnya, lihat Menyebarkan Model.

Anda juga dapat memilih Buka notebook di bagian Run in notebook untuk menjalankan contoh notebook untuk model foundation langsung di Studio Classic.

> **Note**
>
> Untuk menerapkan model pondasi berpemilik di Studio Classic, Anda harus terlebih dahulu berlangganan model di. AWS Marketplace AWS MarketplaceTautan disediakan di notebook contoh terkait dalam Studio Classic.

Jika modelnya dapat disetel dengan baik, Anda juga dapat menyempurnakan modelnya. Untuk informasi selengkapnya, lihat Sempurnakan Model. Untuk daftar model JumpStart pondasi mana yang dapat disetel dengan baik, lihat. Sempurnakan model pondasi

Gunakan model foundation dengan SageMaker Python SDK

Semua model JumpStart foundation tersedia untuk diterapkan secara terprogram menggunakan Python SageMaker SDK. Model kepemilikan harus digunakan menggunakan informasi paket model setelah berlangganan model diAWS Marketplace, sementara model dasar yang tersedia untuk umum dapat digunakan menggunakan ID model di Algoritma Bawaan dengan Tabel Model yang telah dilatih sebelumnya. Untuk memulai dengan model JumpStart foundation menggunakan SageMaker Python SDK, pilih model foundation di Studio Classic, lalu jalankan dan jelajahi contoh notebook terkait.

> **Important**
>
> Beberapa model dasar memerlukan penerimaan eksplisit dari perjanjian lisensi pengguna akhir (EULA). Untuk informasi selengkapnya, lihat Penerimaan EULA dengan SageMaker Python SDK.

Gunakan model foundation berpemilik dengan SageMaker Python SDK

Model berpemilik harus digunakan menggunakan informasi paket model setelah berlangganan model di. AWS Marketplace Untuk informasi selengkapnya tentang SageMaker danAWS Marketplace, lihat SageMakerAlgoritma dan Model Beli dan Jual Amazon di AWS Marketplace. Untuk menemukan AWS Marketplace tautan untuk model kepemilikan terbaru, lihat Memulai Amazon SageMaker JumpStart.

Setelah berlangganan model pilihan AndaAWS Marketplace, Anda dapat menerapkan model foundation menggunakan SageMaker Python SDK dan SDK yang terkait dengan penyedia model. Misalnya, AI21 Labs, Cohere, dan LightOn gunakan`"ai21[SM]"`,, dan `lightonsage` paket`cohere-sagemaker`, masing-masing.

Misalnya step-by-step , temukan dan jalankan notebook yang terkait dengan model pondasi berpemilik pilihan Anda di SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Gunakan model foundation di Amazon SageMaker Studio Classic.

Gunakan model foundation yang tersedia untuk umum dengan SageMaker Python SDK

Untuk informasi selengkapnya tentang penerapan model JumpStart dasar yang tersedia untuk umum secara terprogram, lihat Menerapkan Model Pra-Terlatih Langsung ke Titik Akhir. SageMaker Untuk informasi tentang menyempurnakan model JumpStart foundation secara terprogram, lihat Menyempurnakan Model dan Menerapkan ke Titik Akhir. SageMaker

Untuk mereferensikan ID model yang tersedia untuk model dasar yang tersedia untuk umum, lihat Algoritma Bawaan dengan Tabel Model yang telah dilatih sebelumnya. Cari nama model dasar pilihan Anda di bilah Pencarian, ubah jumlah entri yang ditampilkan menggunakan menu tarik-turun Tampilkan entri, atau pilih teks Berikutnya yang disorot dengan warna biru di sisi kiri halaman untuk menavigasi melalui model yang tersedia.

Misalnya notebook dengan langkah-langkah rinci tentang menggunakan model JumpStart dasar yang tersedia untuk umum dengan SageMaker Python SDK, lihat. Notebook contoh Anda juga dapat menjelajahi contoh notebook yang tersedia di UI SageMaker JumpStart Studio Classic.

Temukan model fondasi di SageMaker Konsol

Anda dapat menjelajahi model JumpStart fondasi langsung melalui SageMaker Konsol Amazon.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Temukan JumpStartdi panel navigasi kiri dan pilih model Foundation.
3. Jelajahi model atau cari model tertentu. Jika Anda membutuhkan panduan untuk pemilihan model, lihatPilih model pondasi. Pilih Lihat model untuk melihat halaman detail model untuk model dasar pilihan Anda.
4. Jika model adalah model berpemilik, pilih Berlangganan di sudut kanan atas halaman detail model untuk berlangganan model diAWS Marketplace. Anda harus menerima email yang mengonfirmasi langganan Anda ke model pilihan Anda. Untuk informasi selengkapnya tentang SageMaker danAWS Marketplace, lihat SageMaker Algoritma dan Model Beli dan Jual Amazon di AWS Marketplace. Model foundation yang tersedia untuk umum tidak memerlukan langganan.

   > **Note**
   >
   > Kemampuan penemuan model pondasi berpemilik saat ini sedang dalam pratinjau di konsol. SageMaker Akses pratinjau untuk model pondasi berpemilik di konsol termasuk ketersediaan taman bermain. Di halaman detail model, pilih Playground untuk menguji model berpemilik tanpa perlu menyiapkan atau menerapkan paket model apa pun.

5. Untuk melihat contoh buku catatan GitHub, pilih Lihat kode di sudut kanan atas halaman detail model.
6. Untuk melihat dan menjalankan contoh notebook secara langsung di Amazon SageMaker Studio Classic, pilih Buka notebook di Studio Classic di sudut kanan atas halaman detail model.

Pilih model pondasi

Amazon SageMaker JumpStart menyediakan akses ke ratusan model yayasan yang tersedia untuk umum dan eksklusif dari sumber dan mitra pihak ketiga. Anda dapat menjelajahi pemilihan model JumpStart foundation langsung di SageMaker konsol, Studio, atau Studio Classic. Saat memilih model pondasi, kami sarankan memulai dengan tugas tertentu dalam pikiran dan mengeksplorasi model potensial dari sana.

Tugas model pondasi

Menurut definisi, model pondasi dapat beradaptasi dengan banyak tugas hilir. Model dasar dilatih pada sejumlah besar data domain umum dan model yang sama dapat diimplementasikan atau disesuaikan untuk beberapa kasus penggunaan. Saat memilih model pondasi Anda, mulailah dengan mendefinisikan tugas tertentu.

Pilih model pembuatan teks

Model dasar pembuatan teks dapat digunakan untuk berbagai tugas hilir, termasuk ringkasan teks, klasifikasi teks, penjawab pertanyaan, pembuatan konten bentuk panjang, copywriting bentuk pendek, ekstraksi informasi, dan banyak lagi.

Untuk menjelajahi model JumpStart dasar pembuatan teks terbaru, gunakan filter Pembuatan Teks di halaman Deskripsi SageMaker JumpStart produk Memulai dengan Amazon. Anda juga dapat menjelajahi model foundation berdasarkan tugas langsung di Amazon SageMaker Studio UI atau SageMaker Studio Classic UI. Untuk informasi selengkapnya, lihat Gunakan model foundation di Amazon SageMaker Studio Classic.

Pilih model pembuatan gambar

JumpStart menyediakan berbagai macam model pondasi pembuatan gambar Stable Diffusion termasuk model dasar dari Stability AI serta model pra-terlatih untuk text-to-image tugas-tugas tertentu dari Hugging Face. Jika Anda perlu menyempurnakan model text-to-image foundation Anda, Anda dapat menggunakan basis Stable Diffusion 2.1 dari Stability AI. Jika Anda ingin menjelajahi model yang sudah dilatih tentang gaya seni tertentu, Anda dapat menjelajahi salah satu dari banyak model pihak ketiga dari Hugging Face langsung di SageMaker Amazon Studio UI SageMaker atau Studio Classic UI.

Untuk menjelajahi model JumpStart dasar pembuatan gambar terbaru, gunakan filter Teks ke Gambar di halaman Memulai dengan deskripsi SageMaker JumpStart produk Amazon. Untuk memulai dengan model text-to-image pondasi pilihan Anda, lihatGunakan model foundation di Amazon SageMaker Studio Classic.

Lisensi dan sumber model

Amazon SageMaker JumpStart menyediakan akses ke model yayasan yang tersedia untuk umum dan eksklusif. Model Foundation dionboard dan dikelola dari penyedia open source dan proprietary pihak ketiga. Dengan demikian, mereka dirilis di bawah lisensi yang berbeda seperti yang ditunjuk

oleh sumber model. Pastikan untuk meninjau lisensi untuk model yayasan apa pun yang Anda gunakan. Anda bertanggung jawab untuk meninjau dan mematuhi persyaratan lisensi yang berlaku dan memastikannya dapat diterima untuk kasus penggunaan Anda sebelum mengunduh atau menggunakan konten. Beberapa contoh lisensi model pondasi umum meliputi:

- Alexa Guru Model
- Apache 2.0
- BigScience Lisensi AI yang Bertanggung Jawab v1.0
- Lisensi CreativeML Open RAIL++-M

Demikian pula, untuk model yayasan berpemilik apa pun, pastikan untuk meninjau dan mematuhi ketentuan penggunaan dan pedoman penggunaan apa pun dari penyedia model. Jika Anda memiliki pertanyaan tentang informasi lisensi untuk model kepemilikan tertentu, hubungi penyedia model secara langsung. Anda dapat menemukan informasi kontak penyedia model di tab Support di setiap halaman modelAWS Marketplace. Untuk menjelajahi model yayasan berpemilik terbaru, lihat Memulai dengan Amazon SageMaker JumpStart.

Perjanjian lisensi pengguna akhir

Beberapa model JumpStart dasar memerlukan penerimaan eksplisit dari perjanjian lisensi pengguna akhir (EULA) sebelum penerapan.

Penerimaan EULA di Amazon Studio SageMaker

Anda mungkin diminta untuk menerima perjanjian lisensi pengguna akhir sebelum menyempurnakan, menerapkan, atau mengevaluasi model dasar di Studio. JumpStart Untuk memulai dengan model JumpStart foundation di Studio, lihatGunakan model pondasi di Studio.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Beberapa model JumpStart dasar memerlukan penerimaan perjanjian lisensi pengguna akhir sebelum penerapan. Jika ini berlaku untuk model dasar yang Anda pilih untuk digunakan, Studio

akan meminta Anda dengan jendela yang berisi konten EULA. Anda bertanggung jawab untuk meninjau dan mematuhi persyaratan lisensi yang berlaku dan memastikannya dapat diterima untuk kasus penggunaan Anda sebelum mengunduh atau menggunakan model.

Penerimaan EULA di Amazon SageMaker Studio Classic

Anda mungkin diminta untuk menerima perjanjian lisensi pengguna akhir sebelum menerapkan model foundation atau membuka notebook model JumpStart JumpStart foundation di Studio Classic. Untuk memulai dengan model JumpStart foundation di Studio Classic, lihatGunakan model foundation di Amazon SageMaker Studio Classic.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Beberapa model JumpStart dasar memerlukan penerimaan perjanjian lisensi pengguna akhir sebelum penerapan. Jika ini berlaku untuk model dasar yang Anda pilih untuk digunakan, Studio Classic akan meminta Anda dengan jendela berjudul Tinjau Perjanjian Lisensi Pengguna Akhir (EULA) dan Kebijakan Penggunaan yang Dapat Diterima (AUP) di bawah ini setelah Anda memilih Deploy atau Buka buku catatan. Anda bertanggung jawab untuk meninjau dan mematuhi persyaratan lisensi yang berlaku dan memastikannya dapat diterima untuk kasus penggunaan Anda sebelum mengunduh atau menggunakan model.

Penerimaan EULA dengan SageMaker Python SDK

Untuk informasi lebih lanjut tentang memulai dengan model JumpStart foundation menggunakan SageMaker Python SDK, lihat. Gunakan model foundation dengan SageMaker Python SDK

> **Important**
>
> Pastikan untuk meng-upgrade ke versi terbaru dari model yang Anda gunakan dan versi terbaru dari SageMaker Python SDK sebelum mengikuti alur kerja ini. Untuk menggunakan alur kerja berikut, Anda harus memiliki v2.198.0 atau yang lebih baru atau SDK Python diinstal. SageMaker

Penerimaan EULA saat menerapkan model JumpStart

Untuk model yang memerlukan penerimaan perjanjian lisensi pengguna akhir, Anda harus secara eksplisit menyatakan penerimaan EULA saat menerapkan model Anda. JumpStart

```
from sagemaker.jumpstart.model import JumpStartModel
model_id = "meta-textgeneration-llama-2-13b"
my_model = JumpStartModel(model_id=model_id)

# Declare EULA acceptance when deploying your JumpStart model
predictor = my_model.deploy(accept_eula=True)
```

`accept_eula`Nilai secara `None` default dan harus didefinisikan ulang secara eksplisit untuk menerima `True` perjanjian lisensi pengguna akhir. Untuk informasi selengkapnya, lihat JumpStartModelreferensi JumpStart Python SDK API.

Penerimaan EULA saat menyempurnakan model JumpStart

Untuk model fine-tuning yang memerlukan penerimaan perjanjian lisensi pengguna akhir, Anda harus secara eksplisit menyatakan penerimaan EULA saat mendefinisikan estimator Anda. JumpStart Perhatikan bahwa setelah menyempurnakan model yang telah dilatih sebelumnya, bobot model asli diubah, sehingga Anda tidak perlu menerima EULA nanti saat menerapkan model yang disetel dengan baik.

```
from sagemaker.jumpstart.estimator import JumpStartEstimator
model_id = "meta-textgeneration-llama-2-13b"

# Declare EULA acceptance when defining your JumpStart estimator
estimator = JumpStartEstimator(model_id=model_id, environment={"accept_eula": "true"})
estimator.fit(
{"train": training_dataset_s3_path, "validation": validation_dataset_s3_path}
)
```

`accept_eula`Nilai secara `None` default dan harus didefinisikan ulang secara eksplisit seperti `"true"` dalam lingkungan estimator untuk menerima perjanjian lisensi pengguna akhir. Untuk informasi selengkapnya, lihat JumpStartEstimatorreferensi JumpStart Python SDK API.

Penerimaan EULA versi SageMaker Python SDK yang lebih lama dari 2.198.0

> **Important**
>
> Saat menggunakan versi yang lebih lama dari 2.198.0 dari SageMaker Python SDK, Anda harus menggunakan SageMaker `Predictor` kelas untuk menerima model EULA.

Setelah menerapkan model JumpStart foundation secara terprogram menggunakan SageMaker Python SDK, Anda dapat menjalankan inferensi terhadap titik akhir yang diterapkan dengan kelas. SageMaker `Predictor` Untuk model yang memerlukan penerimaan perjanjian lisensi pengguna akhir, Anda harus secara eksplisit menyatakan penerimaan EULA dalam panggilan Anda ke kelas. `Predictor`

```
predictor.predict(payload, custom_attributes="accept_eula=true")
```

`accept_eula`Nilai secara `false` default dan harus didefinisikan ulang secara eksplisit untuk menerima `true` perjanjian lisensi pengguna akhir. Prediktor mengembalikan kesalahan jika Anda mencoba menjalankan inferensi saat `accept_eula` disetel ke. `false` Untuk informasi lebih lanjut tentang memulai dengan model JumpStart foundation menggunakan SageMaker Python SDK, lihat. Gunakan model foundation dengan SageMaker Python SDK

> **Important**
>
> `custom_attributes`Parameter menerima pasangan nilai kunci dalam format`"key1=value1;key2=value2"`. Jika Anda menggunakan kunci yang sama beberapa kali, server inferensi menggunakan nilai terakhir yang terkait dengan kunci. Misalnya, jika Anda meneruskan `"accept_eula=false;accept_eula=true"` ke `custom_attributes` parameter, maka server inferensi mengaitkan nilai `true` dengan kunci. `accept_eula`

Sesuaikan model pondasi

Model pondasi adalah model yang sangat kuat yang mampu menyelesaikan beragam tugas. Untuk menyelesaikan sebagian besar tugas secara efektif, model ini memerlukan beberapa bentuk penyesuaian.

Cara yang disarankan untuk terlebih dahulu menyesuaikan model pondasi ke kasus penggunaan tertentu adalah melalui rekayasa yang cepat. Menyediakan model fondasi Anda dengan petunjuk yang direkayasa dengan baik dan kaya konteks dapat membantu mencapai hasil yang diinginkan tanpa penyempurnaan atau perubahan bobot model. Untuk informasi selengkapnya, lihat Rekayasa cepat untuk model pondasi.

Jika rekayasa cepat saja tidak cukup untuk menyesuaikan model pondasi Anda dengan tugas tertentu, Anda dapat menyempurnakan model fondasi pada data spesifik domain tambahan. Untuk informasi selengkapnya, lihat Sempurnakan model pondasi. Proses fine-tuning melibatkan perubahan bobot model.

Jika Anda ingin menyesuaikan model Anda dengan informasi dari perpustakaan pengetahuan tanpa pelatihan ulang, lihatRetrieval Augmented Generation (RAG).

Rekayasa cepat untuk model pondasi

Rekayasa cepat adalah proses merancang dan menyempurnakan petunjuk atau rangsangan input untuk model bahasa untuk menghasilkan jenis output tertentu. Rekayasa cepat melibatkan pemilihan kata kunci yang tepat, memberikan konteks, dan membentuk input dengan cara yang mendorong model untuk menghasilkan respons yang diinginkan dan merupakan teknik penting untuk secara aktif membentuk perilaku dan output model pondasi.

Rekayasa cepat yang efektif sangat penting untuk mengarahkan perilaku model dan mencapai respons yang diinginkan. Melalui teknik yang cepat, Anda dapat mengontrol nada, gaya, dan keahlian domain model tanpa lebih melibatkan langkah-langkah penyesuaian seperti fine-tuning. Kami merekomendasikan untuk mendedikasikan waktu untuk mempercepat rekayasa sebelum Anda mempertimbangkan untuk menyempurnakan model pada data tambahan. Tujuannya adalah untuk memberikan konteks dan panduan yang memadai untuk model sehingga dapat menggeneralisasi dan berkinerja baik pada skenario data yang tidak terlihat atau terbatas.

Pembelajaran zero-shot

Pembelajaran zero-shot melibatkan pelatihan model untuk menggeneralisasi dan membuat prediksi pada kelas atau tugas yang tidak terlihat. Untuk melakukan rekayasa cepat di lingkungan pembelajaran zero-shot, kami merekomendasikan pembuatan prompt yang secara eksplisit memberikan informasi tentang tugas target dan format keluaran yang diinginkan. Misalnya, jika Anda ingin menggunakan model dasar untuk klasifikasi teks zero-shot pada serangkaian kelas yang tidak dilihat model selama pelatihan, prompt yang direkayasa dengan baik dapat berupa: `"Classify the following text as either sports, politics, or entertainment: [input text]."`

Dengan secara eksplisit menentukan kelas target dan format keluaran yang diharapkan, Anda dapat memandu model untuk membuat prediksi yang akurat bahkan pada kelas yang tidak terlihat.

Pembelajaran beberapa tembakan

Pembelajaran few-shot melibatkan pelatihan model dengan jumlah data terbatas untuk kelas atau tugas baru. Rekayasa cepat di lingkungan belajar yang sedikit berfokus pada perancangan petunjuk yang secara efektif menggunakan data pelatihan terbatas yang tersedia. Misalnya, jika Anda menggunakan model dasar untuk tugas klasifikasi gambar dan hanya memiliki beberapa contoh kelas gambar baru, Anda dapat merekayasa prompt yang menyertakan contoh berlabel yang tersedia dengan placeholder untuk kelas target. Misalnya, prompt dapat berupa:"`[image 1], [image 2], and [image 3] are examples of [target class]. Classify the following image as [target class]`". Dengan memasukkan contoh berlabel terbatas dan secara eksplisit menentukan kelas target, Anda dapat memandu model untuk menggeneralisasi dan membuat prediksi yang akurat bahkan dengan data pelatihan minimal.

Jika rekayasa cepat tidak cukup untuk menyesuaikan model fondasi Anda dengan kebutuhan bisnis tertentu, bahasa khusus domain, tugas target, atau persyaratan lainnya, Anda dapat mempertimbangkan untuk menyempurnakan model Anda pada data tambahan atau menggunakan Retrieval Augmented Generation (RAG) untuk menambah arsitektur model Anda dengan konteks yang ditingkatkan dari sumber pengetahuan yang diarsipkan. Untuk informasi selengkapnya, lihat Sempurnakan model pondasi atau Retrieval Augmented Generation (RAG).

Sempurnakan model pondasi

Model pondasi mahal secara komputasi dan dilatih pada korpus besar yang tidak berlabel. Menyesuaikan model pondasi pra-terlatih adalah cara yang terjangkau untuk memanfaatkan kemampuan mereka yang luas sambil menyesuaikan model pada korpus kecil Anda sendiri. Fine-tuning adalah metode penyesuaian yang melibatkan pelatihan lebih lanjut dan mengubah bobot model Anda.

Fine-tuning mungkin berguna bagi Anda jika Anda membutuhkan:

- untuk menyesuaikan model Anda dengan kebutuhan bisnis tertentu
- model Anda untuk berhasil bekerja dengan bahasa khusus domain, seperti jargon industri, istilah teknis, atau kosakata khusus lainnya
- peningkatan kinerja untuk tugas-tugas tertentu
- tanggapan akurat, relatif, dan sadar konteks dalam aplikasi

- tanggapan yang lebih faktual, kurang beracun, dan lebih selaras dengan persyaratan tertentu

Ada dua pendekatan utama yang dapat Anda ambil untuk fine-tuning tergantung pada kasus penggunaan Anda dan model pondasi yang dipilih. Jika Anda tertarik untuk menyempurnakan model Anda pada data spesifik domain, lihat. Penyetelan adaptasi domain Jika Anda tertarik dengan fine-tuning berbasis instruksi menggunakan contoh prompt dan respons, lihat. Penyetelan berbasis instruksi

Penyetelan adaptasi domain

Penyetelan penyesuaian domain memungkinkan Anda memanfaatkan model fondasi yang telah dilatih sebelumnya dan menyesuaikannya dengan tugas tertentu menggunakan data spesifik domain terbatas. Jika upaya rekayasa yang cepat tidak memberikan penyesuaian yang cukup, Anda dapat menggunakan penyesuaian penyesuaian domain untuk membuat model Anda bekerja dengan bahasa khusus domain, seperti jargon industri, istilah teknis, atau data khusus lainnya. Proses fine-tuning ini memodifikasi bobot model.

Untuk informasi selengkapnya, lihat Model SageMaker JumpStart Foundation - Model Fine-tuning model GPT-J 6B generasi teks pada notebook contoh kumpulan data khusus domain. Anda juga dapat mengikuti langkah-langkah format dataset adaptasi domain dalam model Fine-tune lLama 2 pada contoh notebook. SageMaker JumpStart

Penyetelan penyesuaian domain tersedia dengan model fondasi berikut:

> **Note**
>
> Beberapa model JumpStart dasar, seperti L LaMa -2-7B, memerlukan penerimaan perjanjian lisensi pengguna akhir sebelum menyempurnakan dan melakukan inferensi. Untuk informasi selengkapnya, lihat Perjanjian lisensi pengguna akhir.

- BloomZ 7b1
- GPT-J 6B
- GPT Neo 2.7B
- L LaMa -2-7b
- L LaMa -2-13b

Penyetelan berbasis instruksi

Penyetelan berbasis instruksi menggunakan contoh berlabel untuk meningkatkan kinerja model pondasi yang telah dilatih sebelumnya pada tugas tertentu. Contoh berlabel diformat sebagai prompt, pasangan respons dan diungkapkan sebagai instruksi. Proses fine-tuning ini memodifikasi bobot model. Untuk informasi lebih lanjut tentang fine-tuning berbasis instruksi, lihat makalah Memperkenalkan FLAN: Model Bahasa yang Lebih Dapat Digeneralisasikan dengan Instruksi Fine-Tuning dan Scaling Instruction-Finetuned Language Models.

Model Fine-tuned Language Net (FLAN) menggunakan penyetelan instruksi untuk membuat model lebih setuju untuk menyelesaikan tugas NLP hilir umum. Amazon SageMaker JumpStart menyediakan sejumlah model pondasi dalam keluarga model FLAN. Misalnya, model FLAN-T5 adalah instruksi yang disetel dengan baik pada berbagai tugas untuk meningkatkan kinerja zero-shot untuk berbagai kasus penggunaan umum. Dengan data tambahan dan fine-tuning, model berbasis instruksi dapat lebih lanjut disesuaikan dengan tugas yang lebih spesifik yang tidak dipertimbangkan selama pra-pelatihan.

Penyetelan berbasis instruksi tersedia dengan model pondasi berikut:

> **Note**
>
> Beberapa model JumpStart dasar, seperti L LaMa -2-7B, memerlukan penerimaan perjanjian lisensi pengguna akhir sebelum menyempurnakan dan melakukan inferensi. Untuk informasi selengkapnya, lihat Perjanjian lisensi pengguna akhir.

- FLAN-T5 XL
- FLAN-T5 Besar
- FLAN-T5 Kecil
- Dasar FLAN-T5
- L LaMa -2-7b
- L LaMa -2-13b

Notebook contoh

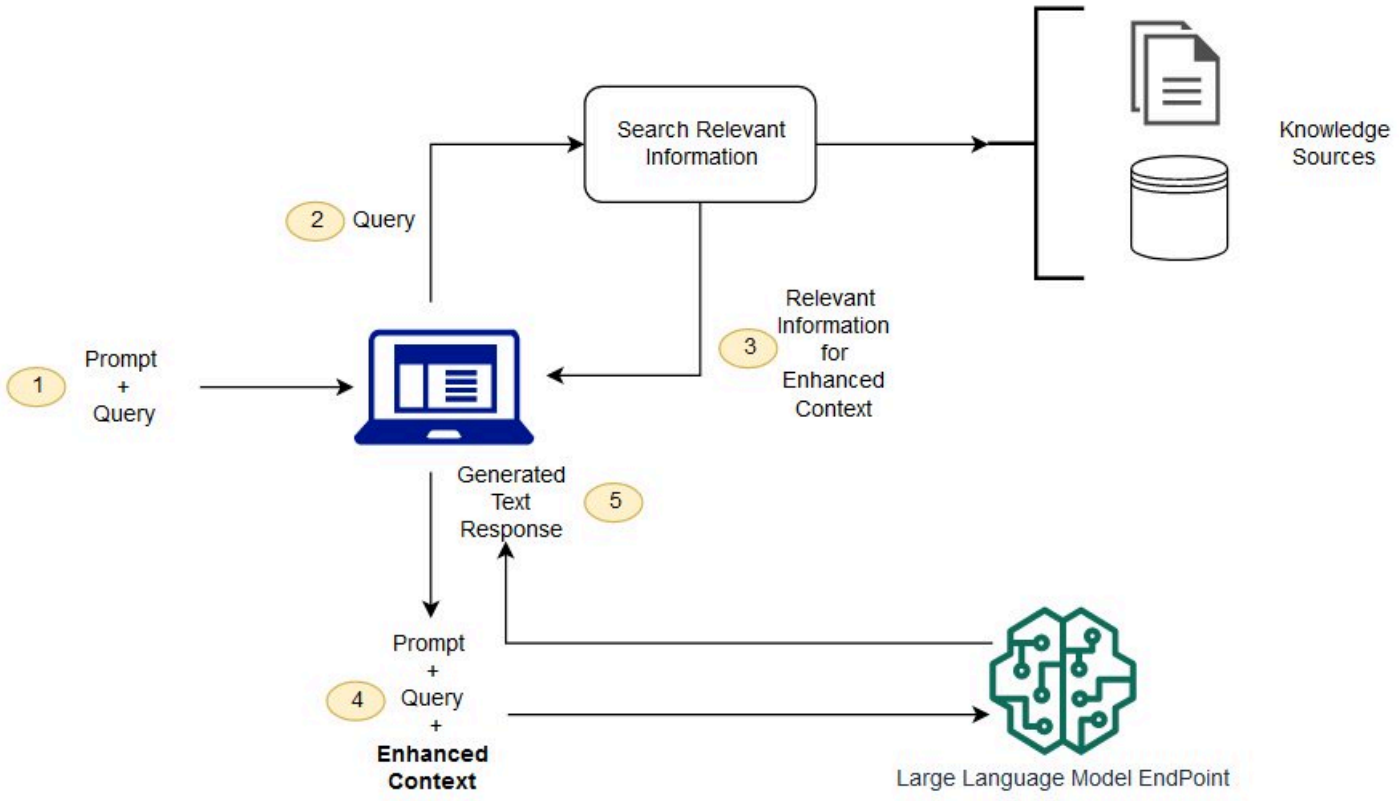
Untuk informasi selengkapnya, lihat contoh buku catatan berikut:

- Sempurnakan model lLAMA 2 pada SageMaker JumpStart
- SageMaker JumpStart Model Foundation - HuggingFace Text2Text Instruksi Fine-Tuning

Retrieval Augmented Generation (RAG)

Model foundation biasanya dilatih secara offline, membuat model agnostik terhadap data apa pun yang dibuat setelah model dilatih. Selain itu, model foundation dilatih pada corpora domain yang sangat umum, membuatnya kurang efektif untuk tugas-tugas khusus domain. Anda dapat menggunakan Retrieval Augmented Generation (RAG) untuk mengambil data dari luar model foundation dan menambah prompt Anda dengan menambahkan data yang diambil yang relevan dalam konteks. Untuk informasi selengkapnya tentang arsitektur model RAG, lihat Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.

Dengan RAG, data eksternal yang digunakan untuk menambah prompt Anda dapat berasal dari beberapa sumber data, seperti repositori dokumen, database, atau API. Langkah pertama adalah mengonversi dokumen Anda dan kueri pengguna apa pun menjadi format yang kompatibel untuk melakukan pencarian relevansi. Untuk membuat format yang kompatibel, koleksi dokumen, atau pustaka pengetahuan, dan kueri yang dikirimkan pengguna dikonversi ke representasi numerik menggunakan model bahasa penyematan. Embedding adalah proses dimana teks diberikan representasi numerik dalam ruang vektor. Arsitektur model RAG membandingkan penyematan kueri pengguna dalam vektor perpustakaan pengetahuan. Prompt pengguna asli kemudian ditambahkan dengan konteks yang relevan dari dokumen serupa dalam perpustakaan pengetahuan. Prompt tambahan ini kemudian dikirim ke model pondasi. Anda dapat memperbarui pustaka pengetahuan dan penyematannya yang relevan secara asinkron.



Notebook contoh

Untuk informasi selengkapnya, lihat contoh buku catatan berikut:

- Retrieval-Augmented Generation: Menjawab Pertanyaan menggunakan LangChain dan Menghasilkan dan Menyematkan Model Cohere dari SageMaker JumpStart
- Retrieval-Augmented Generation: Menjawab Pertanyaan menggunakan LLAMA-2, Pinecone dan Custom Dataset
- Retrieval-Augmented Generation: Menjawab Pertanyaan berdasarkan Dataset Kustom dengan Perpustakaan Sumber Terbuka LangChain
- Retrieval-Augmented Generation: Menjawab Pertanyaan berdasarkan Dataset Kustom

Mengevaluasi model fondasi pembuatan teks di Studio

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

> **Important**
>
> Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Amazon SageMaker JumpStart memiliki integrasi dengan SageMaker Clarify Foundation Model Evaluations (FMEval) di Studio. Jika JumpStart model memiliki kemampuan evaluasi bawaan yang tersedia, Anda dapat memilih Evaluasi di sudut kanan atas halaman detail model di UI JumpStart Studio. Untuk informasi selengkapnya tentang menavigasi UI JumpStart Studio, lihat, Buka dan gunakan JumpStart di Studio

Gunakan Amazon SageMaker JumpStart untuk mengevaluasi model foundation berbasis teks dengan FMEval. Anda dapat menggunakan evaluasi model ini untuk membandingkan metrik kualitas dan tanggung jawab model untuk satu model, antara dua model, atau antara versi berbeda dari model yang sama, untuk membantu Anda mengukur risiko model. FMEval dapat mengevaluasi model berbasis teks yang melakukan tugas-tugas berikut:

- Generasi terbuka — Produksi respons manusia alami terhadap teks yang tidak memiliki struktur yang telah ditentukan sebelumnya.
- Ringkasan teks — Pembuatan ringkasan ringkas dan ringkas sambil mempertahankan makna dan informasi kunci yang terkandung dalam teks yang lebih besar.
- Question Answering — Generasi jawaban dalam bahasa alami untuk sebuah pertanyaan.
- Klasifikasi — Penugasan kelas, seperti `positive` versus `negative` ke bagian teks berdasarkan isinya.

Anda dapat menggunakan FMEval untuk secara otomatis mengevaluasi respons model berdasarkan tolok ukur tertentu. Anda juga dapat mengevaluasi respons model terhadap kriteria Anda sendiri dengan membawa kumpulan data prompt Anda sendiri. FMEval menyediakan antarmuka pengguna (UI) yang memandu Anda melalui pengaturan dan konfigurasi pekerjaan evaluasi. Anda juga dapat menggunakan perpustakaan FMEval di dalam kode Anda sendiri.

Setiap evaluasi membutuhkan kuota untuk dua contoh:

- Hosting instance — Instance yang menghosting dan menyebarkan LLM.
- Contoh evaluasi — Sebuah contoh yang digunakan untuk meminta dan melakukan evaluasi LLM pada instance hosting.

Jika LLM Anda sudah digunakan, berikan titik akhir, dan SageMaker akan menggunakan instance hosting Anda untuk meng-host dan menyebarkan LLM.

Jika Anda mengevaluasi SageMaker JumpStart model yang belum diterapkan ke akun Anda, FMEval membuat instance hosting sementara untuk Anda di akun Anda, dan menyimpannya hanya untuk durasi evaluasi Anda. FMEval menggunakan instance default yang SageMaker JumpStart merekomendasikan LLM yang dipilih sebagai instance hosting Anda. Anda harus memiliki kuota yang cukup untuk contoh yang direkomendasikan ini.

Setiap evaluasi juga menggunakan contoh evaluasi untuk memberikan petunjuk dan menilai tanggapan dari LLM. Anda juga harus memiliki kuota dan memori yang cukup untuk menjalankan algoritma evaluasi. Persyaratan kuota dan memori dari instance evaluasi umumnya lebih kecil daripada yang diperlukan untuk instance hosting. Kami merekomendasikan memilih `ml.m5.2xlarge` instance. Untuk informasi lebih lanjut tentang kuota dan memori, lihatPanduan Pemecahan Masalah FMEval.

Evaluasi otomatis dapat digunakan untuk mencetak LLM di seluruh dimensi berikut:

- Akurasi — Untuk ringkasan teks, penjawab pertanyaan, dan klasifikasi teks
- Kekokohan semantik — Untuk tugas generasi terbuka, ringkasan teks, dan klasifikasi teks
- Pengetahuan faktual — Untuk generasi terbuka
- Stereotip cepat — Untuk generasi terbuka
- Toksisitas — Untuk generasi terbuka, ringkasan teks, dan menjawab pertanyaan

Anda juga dapat menggunakan evaluasi manusia untuk mengevaluasi respons model secara manual. UI FMEval memandu Anda melalui alur kerja memilih satu atau beberapa model, menyediakan sumber daya, dan menulis instruksi untuk dan menghubungi tenaga kerja manusia Anda. Setelah evaluasi manusia selesai, hasilnya ditampilkan di FMEval.

Anda dapat mengakses evaluasi model melalui halaman JumpStart arahan di Studio dengan memilih model untuk dievaluasi dan kemudian memilih Evaluasi. Perhatikan bahwa tidak semua JumpStart

model memiliki kemampuan evaluasi yang tersedia. Untuk informasi selengkapnya tentang cara mengonfigurasi, menyediakan, dan menjalankan FMEval, lihat Apa itu Evaluasi Model Foundation?

Model Khusus Tugas

JumpStart mendukung model khusus tugas di lima belas jenis masalah paling populer. Dari jenis masalah yang didukung, tipe terkait Visi dan NLP berjumlah tiga belas. Ada delapan jenis masalah yang mendukung pelatihan tambahan dan fine-tuning. Untuk informasi selengkapnya tentang pelatihan inkremental dan penyetelan hiper-parameter, lihat SageMaker Penyetelan Model Otomatis. JumpStart juga mendukung empat algoritma populer untuk pemodelan data tabular.

Anda dapat mencari dan menelusuri model dari halaman JumpStart arahan di Studio atau Studio Classic. Saat Anda memilih model, halaman detail model memberikan informasi tentang model, dan Anda dapat melatih dan menerapkan model Anda dalam beberapa langkah. Bagian deskripsi menjelaskan apa yang dapat Anda lakukan dengan model, jenis input dan output yang diharapkan, dan tipe data yang diperlukan untuk menyempurnakan model Anda.

Anda juga dapat menggunakan model secara terprogram dengan Python SageMaker SDK. Untuk daftar semua model yang tersedia, lihat Tabel Model yang JumpStart Tersedia.

Daftar jenis masalah dan tautan ke contoh notebook Jupyter mereka dirangkum dalam tabel berikut.

Jenis masalah	Mendukung inferensi dengan model pra-terla tih	Dapat dilatih pada kumpulan data khusus	Kerangka kerja yang didukung	Contoh Notebook
Klasifikasi gambar	Ya	Ya	PyTorch, TensorFlow	Pengantar JumpStart - Klasifikasi Gambar
Deteksi objek	Ya	Ya	PyTorch, TensorFlow, MXNet	Pengantar JumpStart - Deteksi Objek
Segmentasi semantik	Ya	Ya	MXNet	Pengantar JumpStart -

Jenis masalah	Mendukung inferensi dengan model pra-terla tih	Dapat dilatih pada kumpulan data khusus	Kerangka kerja yang didukung	Contoh Notebook
				Segmentasi Semantik
Segmentasi contoh	Ya	Ya	MXNet	Pengantar JumpStart - Segmentasi Instance
Penyematan gambar	Ya	Tidak	TensorFlow, MxNet	Pengantar JumpStart - Penyematan Gambar
Klasifikasi teks	Ya	Ya	TensorFlow	Pengantar JumpStart - Klasifikasi Teks
Klasifikasi pasangan kalimat	Ya	Ya	TensorFlow, Hugging Face	Pengantar JumpStart - Klasifikasi Pasangan Kalimat
Menjawab pertanyaan	Ya	Ya	PyTorch, Hugging Face	Pengantar JumpStart — Menjawab Pertanyaan
Pengakuan entitas bernama	Ya	Tidak	Hugging Face	Pengantar JumpStart - Pengakuan Entitas Bernama

Jenis masalah	Mendukung inferensi dengan model pra-terla tih	Dapat dilatih pada kumpulan data khusus	Kerangka kerja yang didukung	Contoh Notebook
Ringkasan teks	Ya	Tidak	Hugging Face	Pengantar JumpStart - Ringkasan Teks
Pembuatan teks	Ya	Tidak	Hugging Face	Pengantar JumpStart - Pembuatan Teks
Terjemahan mesin	Ya	Tidak	Hugging Face	Pengantar JumpStart - Terjemahan Mesin
Penyematan teks	Ya	Tidak	TensorFlow, MxNet	Pengantar JumpStart - Penyematan Teks

Jenis masalah	Mendukung inferensi dengan model pra-terla tih	Dapat dilatih pada kumpulan data khusus	Kerangka kerja yang didukung	Contoh Notebook
Klasifikasi tabel	Ya	Ya	LightGBM, , XGBoost CatBoost, AutoGluon -Tabular,, Pembelajar Linier TabTransf ormer	Pengantar JumpStart - Klasifika si Tabular - LightGBM, CatBoost Pengantar JumpStart - Klasifika si Tabular - XGBoost, Linear Learner Pengantar JumpStart - Klasifika si Tabular - Peserta AutoGluon Pengantar JumpStart - Klasifika si Tabular - Peserta TabTransformer

Jenis masalah	Mendukung inferensi dengan model pra-terla tih	Dapat dilatih pada kumpulan data khusus	Kerangka kerja yang didukung	Contoh Notebook
Regresi tabel	Ya	Ya	LightGBM, , XGBoost CatBoost, AutoGluon -Tabular,, Pembelajar Linier TabTransf ormer	Pengantar JumpStart - Regresi Tabular - LightGBM, CatBoost Pengantar - Regresi Tabular JumpStart - XGBoost, Linear Learner Pengantar JumpStart — Regresi Tabular - Peserta AutoGluon Pengantar JumpStart — Regresi Tabular - Peserta TabTransformer

Menyebarkan Model

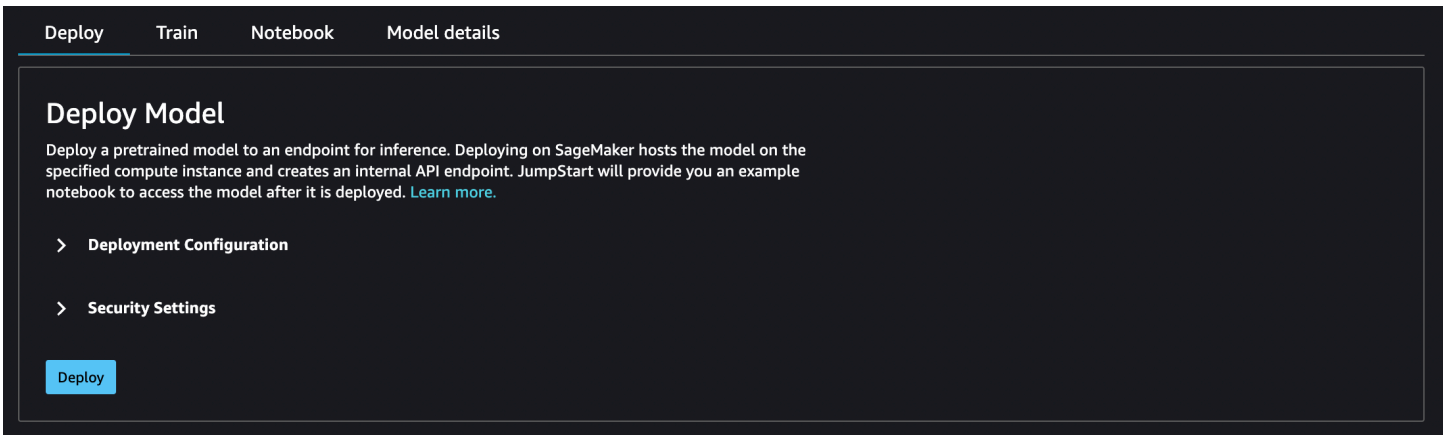
Saat Anda menerapkan model dari JumpStart, SageMaker host model dan menerapkan titik akhir yang dapat Anda gunakan untuk inferensi. JumpStart juga menyediakan contoh notebook yang dapat Anda gunakan untuk mengakses model setelah diterapkan.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

> **Note**
>
> Untuk informasi lebih lanjut tentang penerapan JumpStart model di Studio, lihat Menyebarkan model pondasi di Studio

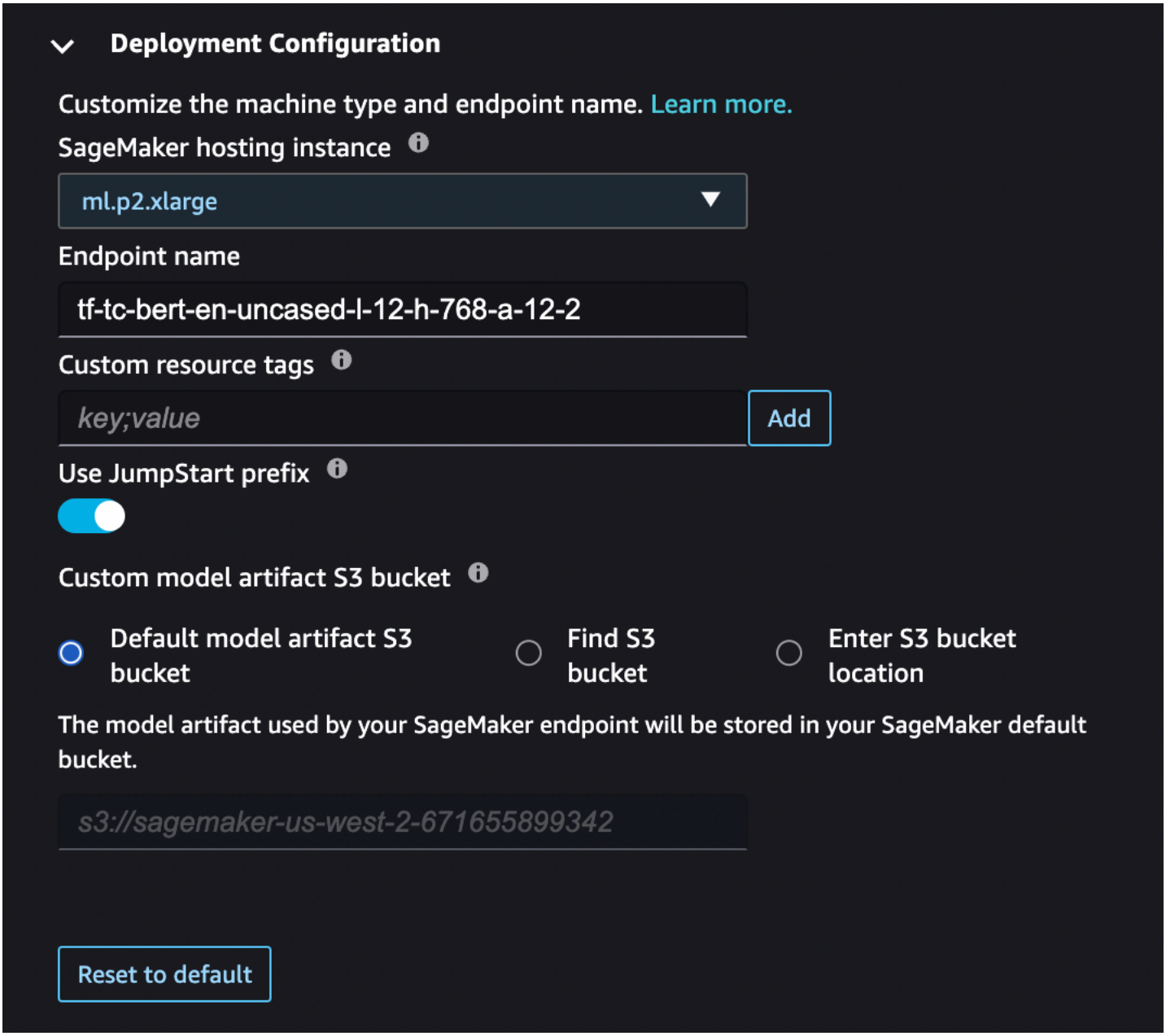
Konfigurasi penerapan model

Setelah Anda memilih model, tab model terbuka. Di panel Deploy Model, pilih Deployment Configuration untuk mengonfigurasi penerapan model Anda.

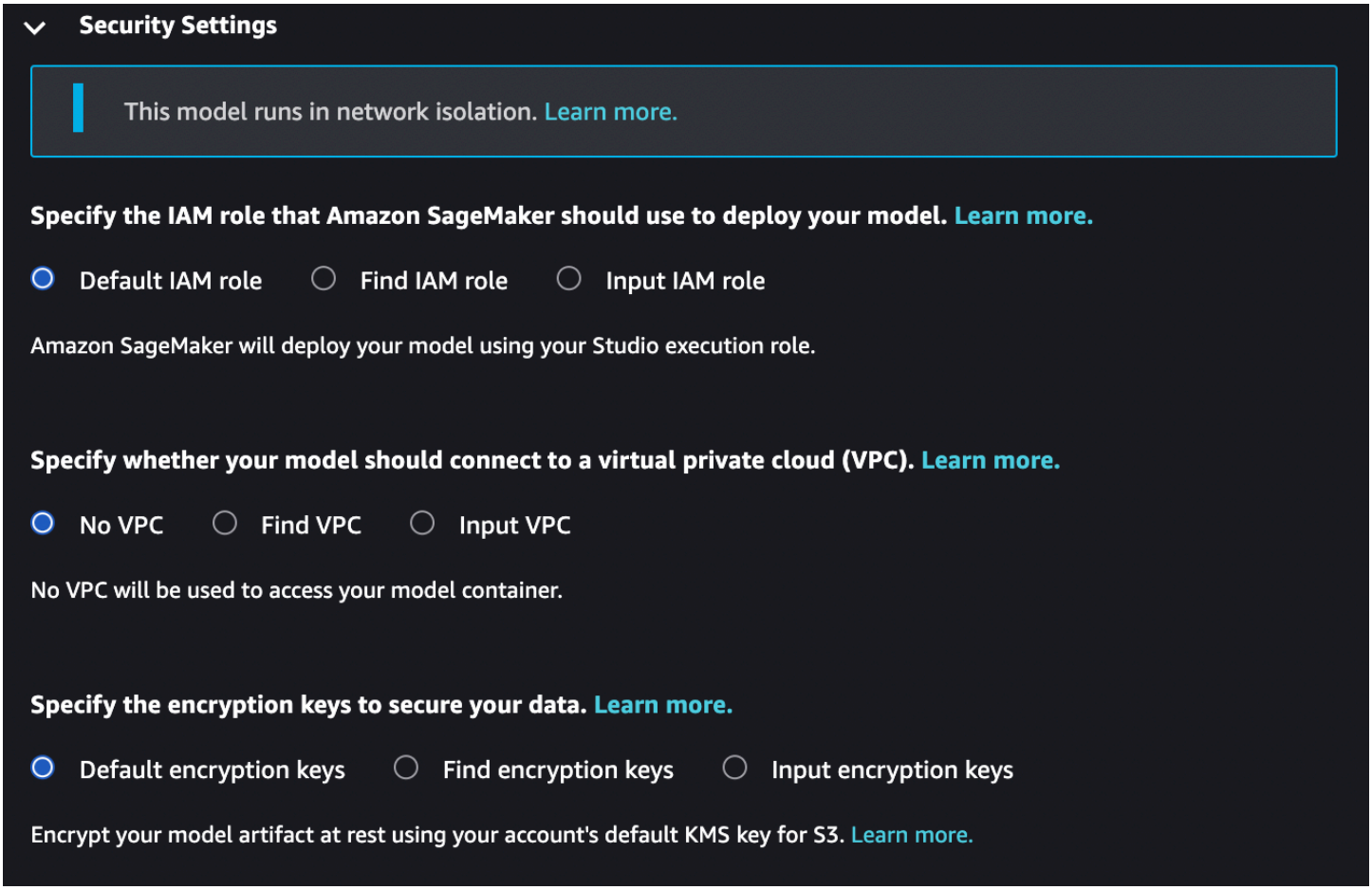


Jenis instans default untuk menerapkan model bergantung pada model. Jenis instans adalah perangkat keras tempat pekerjaan pelatihan berjalan. Dalam contoh berikut, `ml.p2.xlarge` instance adalah default untuk model BERT khusus ini.

Anda juga dapat mengubah nama titik akhir, menambahkan tag `key;value` sumber daya, mengaktifkan atau menonaktifkan `jumpstart-` awalan untuk sumber JumpStart daya apa pun yang terkait dengan model, dan menentukan bucket Amazon S3 untuk menyimpan artefak model yang digunakan oleh titik akhir Anda. SageMaker



Pilih Pengaturan Keamanan untuk menentukan peran AWS Identity and Access Management (IAM), Amazon Virtual Private Cloud (Amazon VPC), dan kunci enkripsi untuk model.



Keamanan penerapan model

Saat menerapkan model JumpStart, Anda dapat menentukan peran IAM, Amazon VPC, dan kunci enkripsi untuk model tersebut. Jika Anda tidak menentukan nilai apa pun untuk entri ini: Peran IAM default adalah peran runtime Studio Classic Anda; enkripsi default digunakan; tidak ada VPC Amazon yang digunakan.

Peran IAM

Anda dapat memilih peran IAM yang disahkan sebagai bagian dari pekerjaan pelatihan dan pekerjaan hosting. SageMaker menggunakan peran ini untuk mengakses data pelatihan dan artefak model. Jika Anda tidak memilih peran IAM, SageMaker gunakan model menggunakan peran runtime Studio Classic. Untuk informasi selengkapnya tentang peran IAM, lihatIdentity and Access Management untuk Amazon SageMaker.

Peran yang Anda lewati harus memiliki akses ke sumber daya yang dibutuhkan model, dan harus mencakup semua hal berikut.

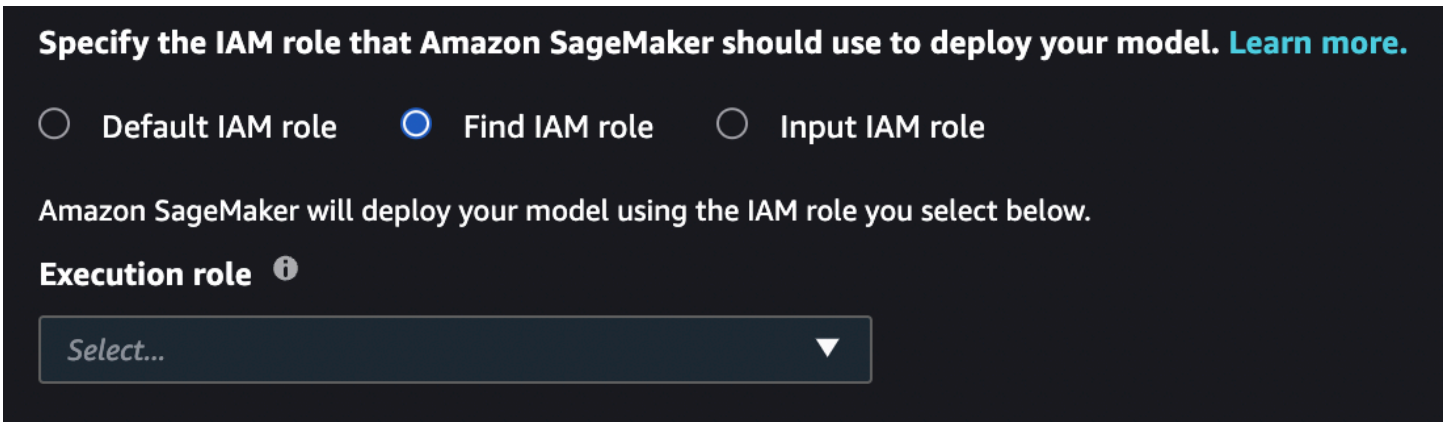
- Untuk pekerjaan pelatihan: CreateTrainingJob API: Izin Peran Eksekusi.
- Untuk pekerjaan hosting: CreateModel API: Izin Peran Eksekusi.

> **Note**
>
> Anda dapat mencatat izin Amazon S3 yang diberikan di setiap peran berikut. Lakukan ini dengan menggunakan ARN bucket Amazon Simple Storage Service (Amazon S3) dan bucket Amazon S3. JumpStart

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:PutObject",
    "s3:ListMultipartUploadParts",
    "s3:ListBucket"
  ],
  "Resources": [
    "arn:aws:s3:::jumpstart-cache-prod-<region>/*",
    "arn:aws:s3:::jumpstart-cache-prod-<region>",
    "arn:aws:s3:::bucket/*"
  ]
}
```

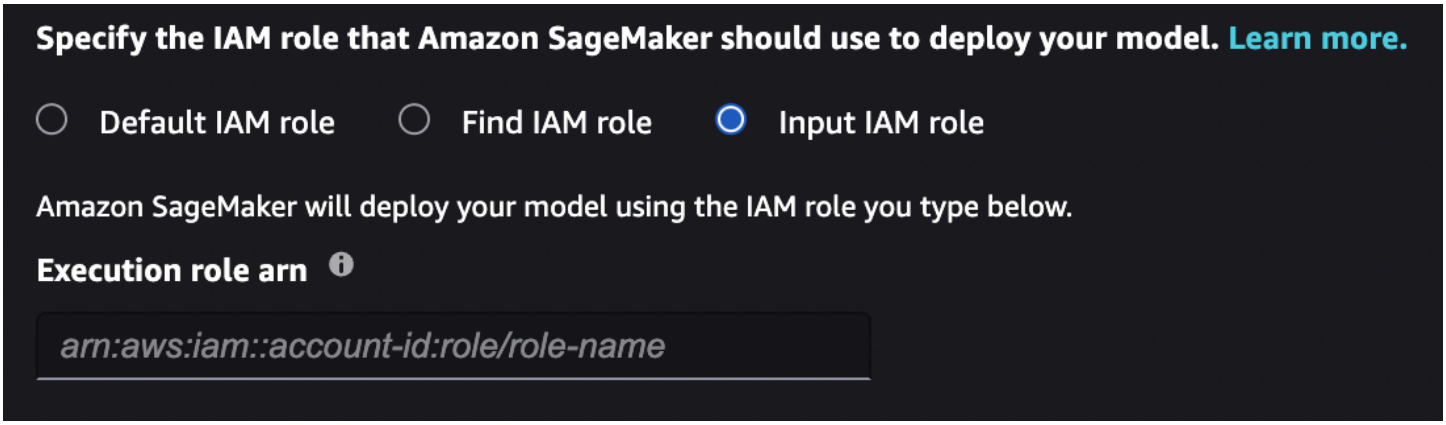
Temukan peran IAM

Jika Anda memilih opsi ini, Anda harus memilih peran IAM yang ada dari daftar dropdown.



Masukan peran IAM

Jika Anda memilih opsi ini, Anda harus memasukkan ARN secara manual untuk peran IAM yang ada. Jika peran runtime Studio Classic atau Amazon VPC memblokir `iam:list*` panggilan, Anda harus menggunakan opsi ini untuk menggunakan peran IAM yang ada.



Amazon VPC

Semua JumpStart model berjalan dalam mode isolasi jaringan. Setelah wadah model dibuat, tidak ada lagi panggilan yang dapat dilakukan. Anda dapat memilih VPC Amazon yang lulus sebagai bagian dari pekerjaan pelatihan dan pekerjaan hosting. SageMaker menggunakan VPC Amazon ini untuk mendorong dan menarik sumber daya dari bucket Amazon S3 Anda. VPC Amazon ini berbeda dengan VPC Amazon yang membatasi akses ke internet publik dari instans Studio Classic Anda. Untuk informasi selengkapnya tentang Studio Classic Amazon VPC, lihat. Connect SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal

VPC Amazon yang Anda lewati tidak memerlukan akses ke internet publik, tetapi memang membutuhkan akses ke Amazon S3. Titik akhir Amazon VPC untuk Amazon S3 harus memungkinkan akses ke setidaknya sumber daya berikut yang dibutuhkan model.

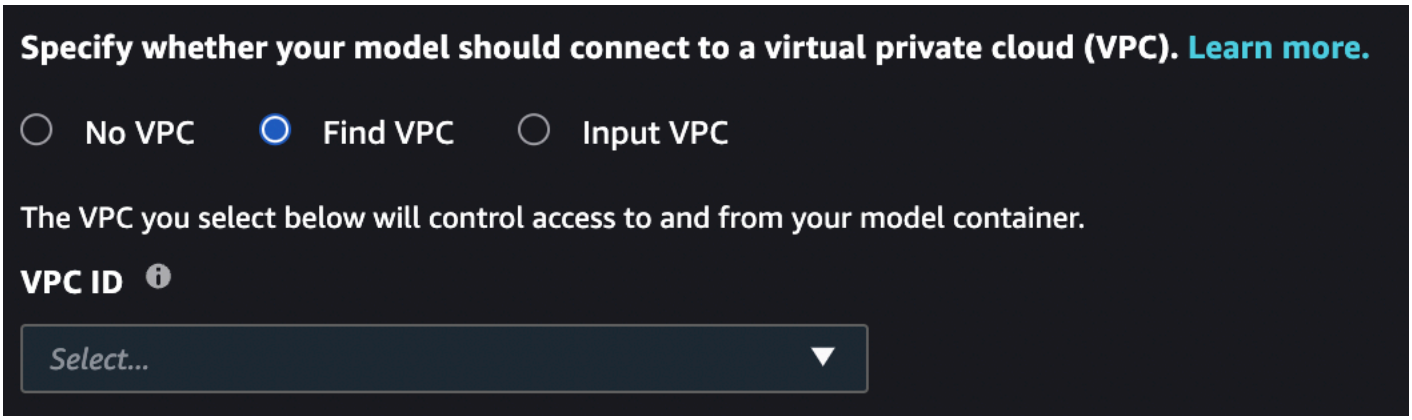
```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:PutObject",
    "s3:ListMultipartUploadParts",
    "s3:ListBucket"
  ],
  "Resources": [
    "arn:aws:s3:::jumpstart-cache-prod-<region>/*",
```

```
    "arn:aws:s3:::jumpstart-cache-prod-<region>",
    "arn:aws:s3:::bucket/*"
  ]
}
```

Jika Anda tidak memilih VPC Amazon, tidak ada VPC Amazon yang digunakan.

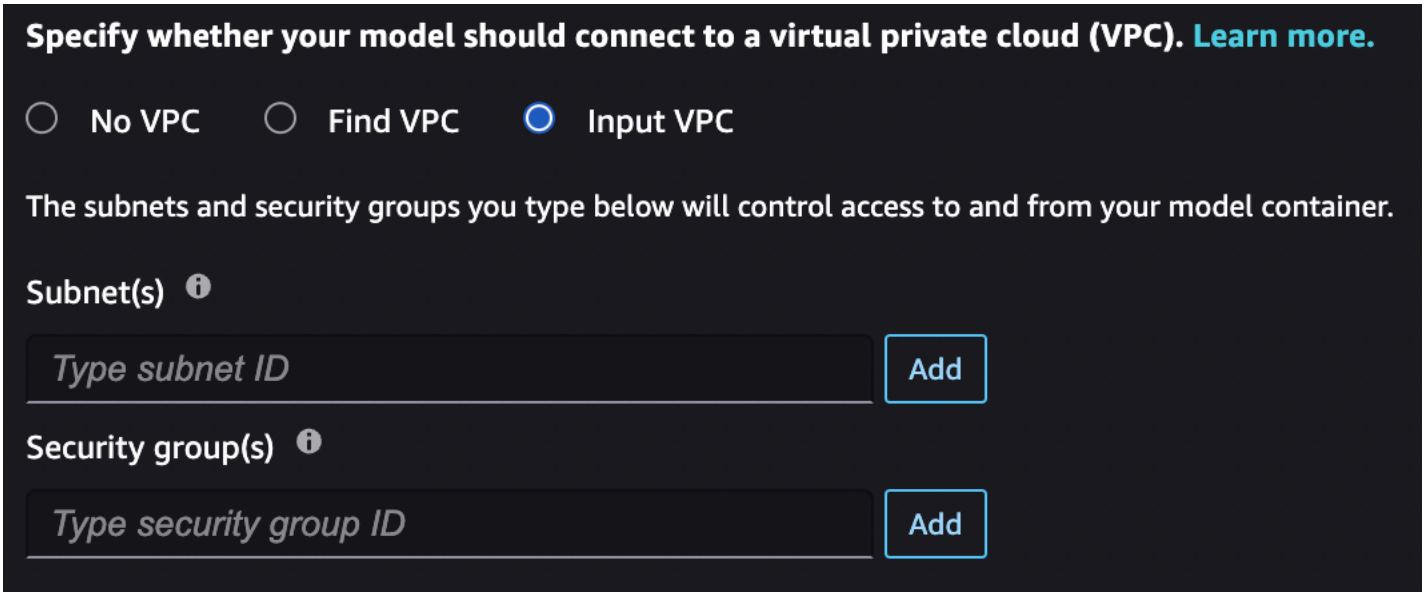
Temukan VPC

Jika Anda memilih opsi ini, Anda harus memilih VPC Amazon yang ada dari daftar tarik-turun. Setelah memilih VPC Amazon, Anda harus memilih subnet dan grup keamanan untuk VPC Amazon Anda. Untuk informasi selengkapnya tentang subnet dan grup keamanan, lihat Ikhtisar VPC dan subnet.



Masukan VPC

Jika Anda memilih opsi ini, Anda harus secara manual memilih subnet dan grup keamanan yang menyusun VPC Amazon Anda. Jika peran runtime Studio Classic atau Amazon VPC memblokir `ec2:list*` panggilan, Anda harus menggunakan opsi ini untuk memilih subnet dan grup keamanan.



Kunci enkripsi

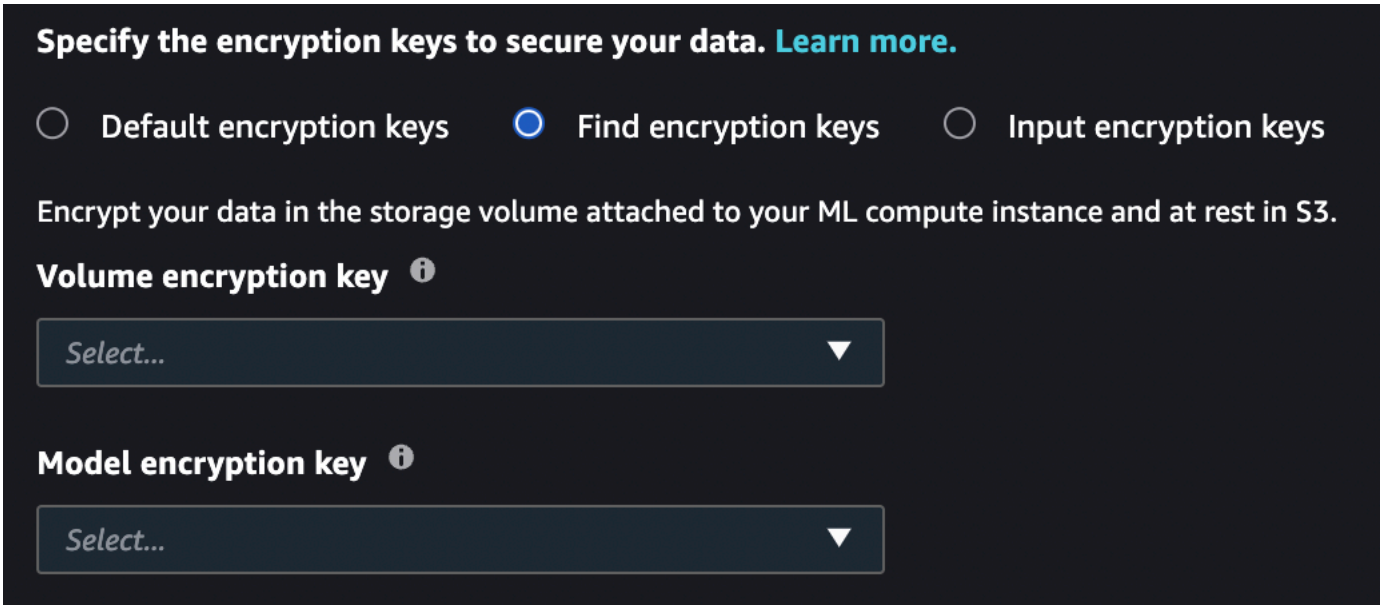
Anda dapat memilih AWS KMS kunci yang diteruskan sebagai bagian dari pekerjaan pelatihan dan pekerjaan hosting. SageMaker menggunakan kunci ini untuk mengenkripsi volume Amazon EBS untuk wadah, dan model yang dikemas ulang di Amazon S3 untuk pekerjaan hosting dan output untuk pekerjaan pelatihan. Untuk informasi selengkapnya tentang AWS KMS kunci, lihat AWS KMSkunci.

Kunci yang Anda lewati harus mempercayai peran IAM yang Anda lewati. Jika Anda tidak menentukan peran IAM, AWS KMS kunci harus mempercayai peran runtime Studio Classic Anda.

Jika Anda tidak memilih AWS KMS kunci, SageMaker berikan enkripsi default untuk data dalam volume Amazon EBS dan artefak Amazon S3.

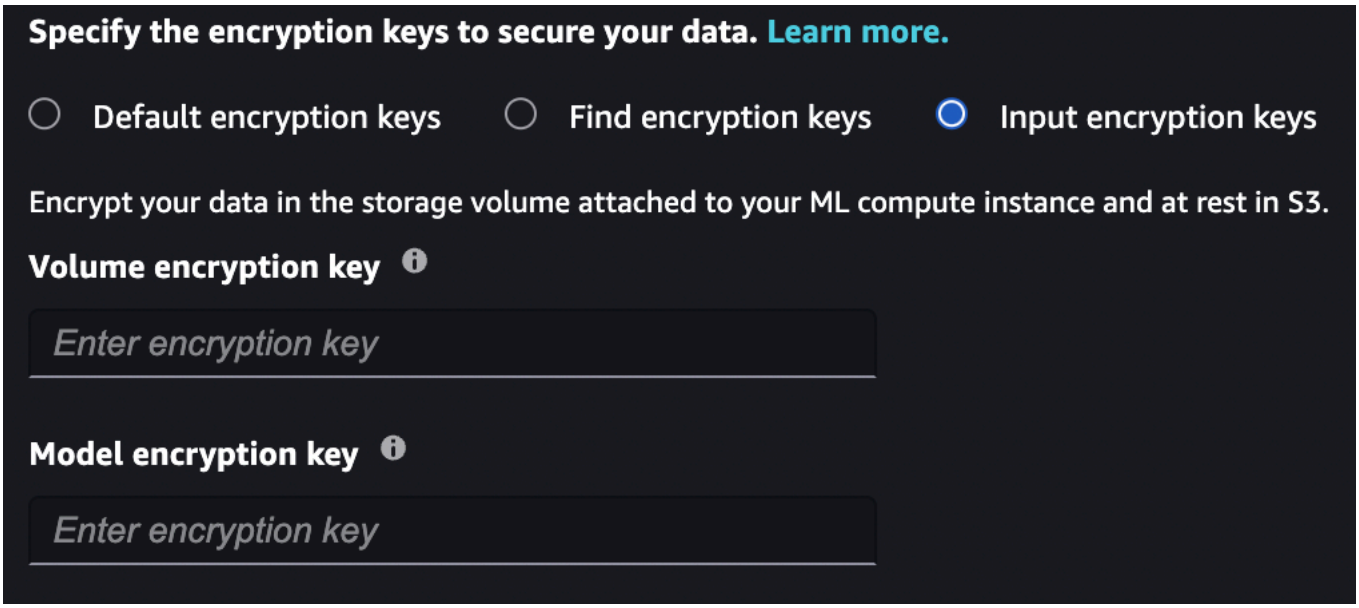
Temukan kunci enkripsi

Jika Anda memilih opsi ini, Anda harus memilih AWS KMS kunci yang ada dari daftar dropdown.



Kunci enkripsi masukan

Jika Anda memilih opsi ini, Anda harus memasukkan AWS KMS tombol secara manual. Jika peran eksekusi Studio Classic atau Amazon VPC memblokir `kms:list*` panggilan, Anda harus menggunakan opsi ini untuk memilih kunci yang adaAWS KMS.



Konfigurasikan nilai default untuk JumpStart model

Anda dapat mengonfigurasi nilai default untuk parameter seperti peran IAM, VPC, dan kunci KMS untuk diisi sebelumnya untuk penerapan dan pelatihan model. JumpStart Setelah mengonfigurasi nilai default, UI Studio Classic secara otomatis menyediakan setelan keamanan dan tag yang

Anda tentukan ke SageMaker JumpStart model untuk menyederhanakan alur kerja penerapan dan pelatihan. Administrator dan pengguna akhir dapat menginisialisasi nilai default yang ditentukan dalam file konfigurasi dalam format YAMM.

Secara default, SageMaker Python SDK menggunakan dua file konfigurasi: satu untuk administrator dan satu untuk pengguna. Menggunakan file konfigurasi admininistrator, administrator dapat menentukan satu set nilai default. Pengguna akhir dapat mengganti nilai yang ditetapkan dalam file konfigurasi administrator dan menetapkan nilai default tambahan menggunakan file konfigurasi pengguna akhir. Untuk informasi selengkapnya, lihat Lokasi file konfigurasi default.

Contoh kode berikut mencantumkan lokasi default file konfigurasi saat menggunakan SageMaker Python SDK di Amazon SageMaker Studio Classic.

```
# Location of the admin config file
/etc/xdg/sagemaker/config.yaml

# Location of the user config file
/root/.config/sagemaker/config.yaml
```

Nilai yang ditentukan dalam file konfigurasi pengguna mengganti nilai yang ditetapkan dalam file konfigurasi administrator. File konfigurasi unik untuk setiap profil pengguna dalam SageMaker Domain Amazon. Aplikasi Studio Classic profil pengguna secara langsung terkait dengan profil pengguna. Untuk informasi selengkapnya, lihat Profil Pengguna Domain.

Administrator dapat secara opsional mengatur default konfigurasi untuk pelatihan JumpStart model dan penerapan melalui konfigurasi siklus hidup. `JupyterServer` Untuk informasi selengkapnya, lihat Membuat dan mengaitkan konfigurasi siklus hidup.

Konfigurasi nilai default file YAMM

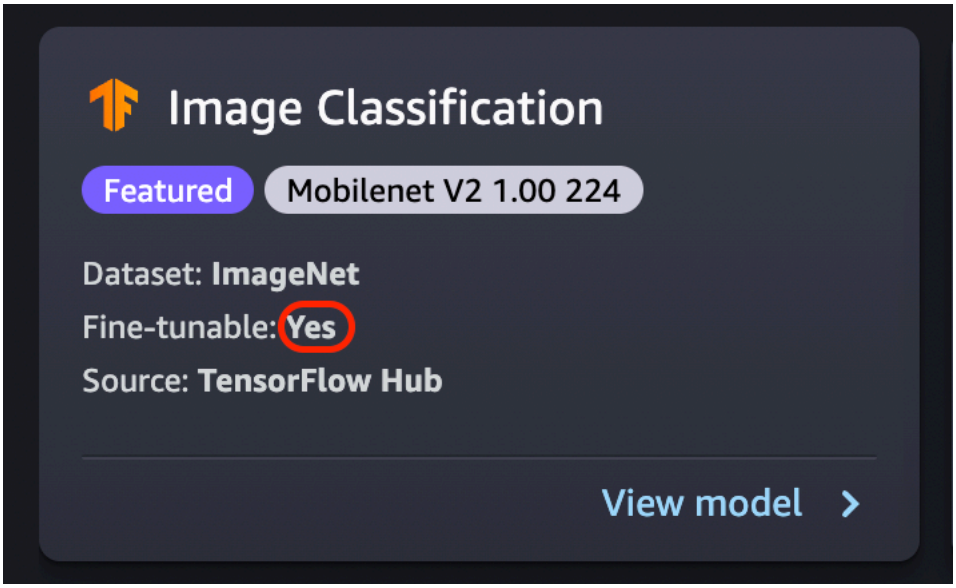
File konfigurasi Anda harus mematuhi struktur file konfigurasi SageMaker Python SDK. Perhatikan bahwa bidang tertentu dalam`TrainingJob`,`Model`, dan `EndpointConfig` konfigurasi berlaku untuk pelatihan JumpStart model dan nilai default penerapan.

```
SchemaVersion: '1.0'
SageMaker:
  TrainingJob:
    OutputDataConfig:
      KmsKeyId: example-key-id
    ResourceConfig:
      # Training configuration - Volume encryption key
```

```
      VolumeKmsKeyId: example-key-id
    # Training configuration form - IAM role
    RoleArn: arn:aws:iam::123456789012:role/SageMakerExecutionRole
    VpcConfig:
      # Training configuration - Security groups
      SecurityGroupIds:
      - sg-1
      - sg-2
      # Training configuration - Subnets
      Subnets:
      - subnet-1
      - subnet-2
    # Training configuration - Custom resource tags
    Tags:
    - Key: Example-key
      Value: Example-value
  Model:
    EnableNetworkIsolation: true
    # Deployment configuration - IAM role
    ExecutionRoleArn: arn:aws:iam::123456789012:role/SageMakerExecutionRole
    VpcConfig:
      # Deployment configuration - Security groups
      SecurityGroupIds:
      - sg-1
      - sg-2
      # Deployment configuration - Subnets
      Subnets:
      - subnet-1
      - subnet-2
  EndpointConfig:
    AsyncInferenceConfig:
      OutputConfig:
        KmsKeyId: example-key-id
    DataCaptureConfig:
      # Deployment configuration - Volume encryption key
      KmsKeyId: example-key-id
    KmsKeyId: example-key-id
    # Deployment configuration - Custom resource tags
    Tags:
    - Key: Example-key
      Value: Example-value
```

Sempurnakan Model

Fine-tuning melatih model yang telah dilatih sebelumnya pada dataset baru tanpa pelatihan dari awal. Proses ini, juga dikenal sebagai pembelajaran transfer, dapat menghasilkan model yang akurat dengan kumpulan data yang lebih kecil dan waktu pelatihan yang lebih sedikit. Anda dapat menyempurnakan model jika kartunya menunjukkan atribut fine-tunable yang disetel ke Yes.



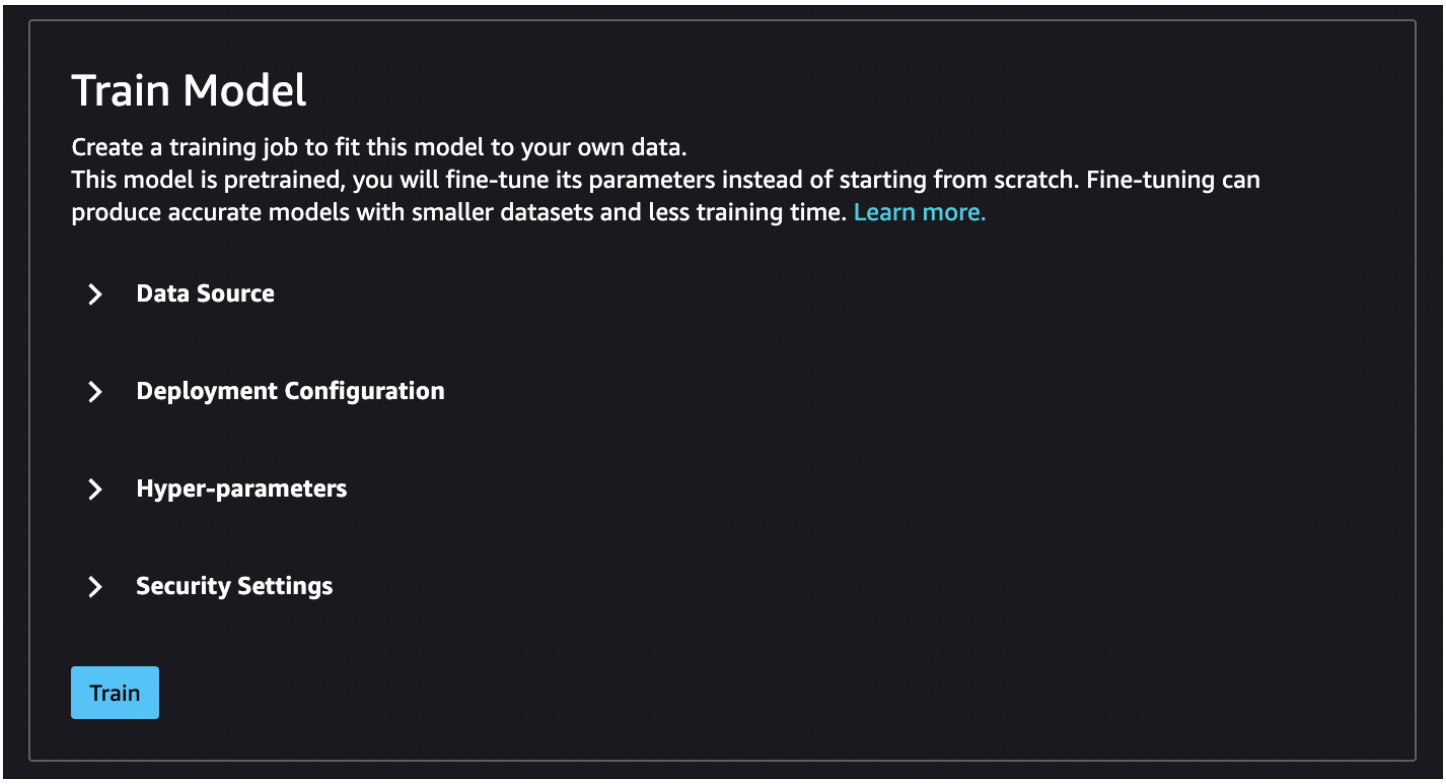
> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

> **Note**
>
> Untuk informasi selengkapnya tentang fine-tuning JumpStart model di Studio, lihat Sempurnakan model pondasi di Studio

Sumber data Fine-Tuning

Saat menyempurnakan model, Anda dapat menggunakan kumpulan data default atau memilih data Anda sendiri, yang terletak di bucket Amazon S3.

Untuk menelusuri bucket yang tersedia untuk Anda, pilih bucket Find S3. Bucket ini dibatasi oleh izin yang digunakan untuk menyiapkan akun Studio Classic Anda. Anda juga dapat menentukan URI Amazon S3 dengan memilih Masukkan lokasi bucket Amazon S3.



> **Tip**
>
> Untuk mengetahui cara memformat data di bucket, pilih Pelajari lebih lanjut. Bagian deskripsi untuk model memiliki informasi rinci tentang input dan output.

Untuk model teks:

- Bucket harus memiliki file data.csv.
- Kolom pertama harus berupa bilangan bulat unik untuk label kelas. Misalnya:1,2,3,4, n
- Kolom kedua harus berupa string.
- Kolom kedua harus memiliki teks yang sesuai yang cocok dengan jenis dan bahasa untuk model.

Untuk model visi:

- Bucket harus memiliki subdirektori sebanyak jumlah kelas.

- Setiap subdirektori harus berisi gambar yang termasuk dalam kelas itu dalam format.jpg.

> **Note**
>
> Bucket Amazon S3 harus sama dengan Wilayah AWS tempat Anda menjalankan SageMaker Studio Classic karena SageMaker tidak mengizinkan permintaan Lintas wilayah.

Konfigurasi penyebaran Fine-Tuning

Keluarga p3 direkomendasikan sebagai yang tercepat untuk pelatihan pembelajaran mendalam, dan ini direkomendasikan untuk menyempurnakan model. Bagan berikut menunjukkan jumlah GPU di setiap jenis instans. Ada opsi lain yang tersedia yang dapat Anda pilih, termasuk jenis instans p2 dan g4.

Jenis instans	GPU
p3.2xlarge	1
p3.8xlarge	4
p3.16xlarge	8
p3dn.24xlarge	8

Hyperparameter

Anda dapat menyesuaikan hyperparameters dari pekerjaan pelatihan yang digunakan untuk menyempurnakan model. Hiperparameter yang tersedia untuk setiap model yang dapat disetel berbeda tergantung pada modelnya. Untuk informasi tentang setiap hyperparameter yang tersedia, rujuk dokumentasi hyperparameters untuk model yang Anda pilih. Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih Misalnya, lihat Klasifikasi Gambar - TensorFlow Hyperparameters detail tentang Klasifikasi Gambar yang dapat disetel dengan baik - hiperparameter. TensorFlow

Jika Anda menggunakan dataset default untuk model teks tanpa mengubah hyperparameters, Anda mendapatkan model yang hampir identik sebagai hasilnya. Untuk model visi, kumpulan data default

berbeda dari kumpulan data yang digunakan untuk melatih model yang telah dilatih sebelumnya, sehingga model Anda berbeda sebagai hasilnya.

Hyperparameter berikut umum di antara model:

- Epoch — Satu epoch adalah satu siklus melalui seluruh dataset. Beberapa interval menyelesaikan batch, dan beberapa batch akhirnya menyelesaikan sebuah epoch. Beberapa zaman dijalankan hingga keakuratan model mencapai tingkat yang dapat diterima, atau ketika tingkat kesalahan turun di bawah tingkat yang dapat diterima.
- Laju belajar — Jumlah nilai yang harus diubah antar zaman. Saat model disempurnakan, bobot internalnya didorong dan tingkat kesalahan diperiksa untuk melihat apakah model membaik. Tingkat pembelajaran tipikal adalah 0,1 atau 0,01, di mana 0,01 adalah penyesuaian yang jauh lebih kecil dan dapat menyebabkan pelatihan memakan waktu lama untuk bertemu, sedangkan 0,1 jauh lebih besar dan dapat menyebabkan pelatihan melampaui batas. Ini adalah salah satu hiperparameter utama yang mungkin Anda sesuaikan untuk melatih model Anda. Perhatikan bahwa untuk model teks, tingkat pembelajaran yang jauh lebih kecil (5e-5 untuk BERT) dapat menghasilkan model yang lebih akurat.
- Ukuran Batch — Jumlah catatan dari kumpulan data yang akan dipilih untuk setiap interval untuk dikirim ke GPU untuk pelatihan.

  Dalam contoh gambar, Anda mungkin mengirimkan 32 gambar per GPU, jadi 32 akan menjadi ukuran batch Anda. Jika Anda memilih jenis instans dengan lebih dari satu GPU, batch dibagi dengan jumlah GPU. Ukuran batch yang disarankan bervariasi tergantung pada data dan model yang Anda gunakan. Misalnya, cara Anda mengoptimalkan data gambar berbeda dari cara Anda menangani data bahasa.

  Pada bagan tipe instans di bagian konfigurasi penerapan, Anda dapat melihat jumlah GPU per jenis instans. Mulailah dengan ukuran batch standar yang direkomendasikan (misalnya, 32 untuk model visi). Kemudian, kalikan ini dengan jumlah GPU dalam jenis instance yang Anda pilih. Misalnya, jika Anda menggunakan `ap3.8xlarge`, ini akan menjadi 32 (ukuran batch) dikalikan dengan 4 (GPU), dengan total 128, karena ukuran batch Anda menyesuaikan dengan jumlah GPU. Untuk model teks seperti BERT, coba mulai dengan ukuran batch 64, dan kemudian kurangi sesuai kebutuhan.

Output pelatihan

Ketika proses fine-tuning selesai, JumpStart berikan informasi tentang model: model induk, nama pekerjaan pelatihan, ARN pekerjaan pelatihan, waktu pelatihan, dan jalur keluaran. Jalur keluaran

adalah tempat Anda dapat menemukan model baru Anda di bucket Amazon S3. Struktur folder menggunakan nama model yang Anda berikan dan file model ada di `/output` subfolder dan selalu diberi nama`model.tar.gz`.

Contoh: `s3://bucket/model-name/output/model.tar.gz`

Konfigurasikan nilai default untuk pelatihan model

Anda dapat mengonfigurasi nilai default untuk parameter seperti peran IAM, VPC, dan kunci KMS untuk diisi sebelumnya untuk penerapan dan pelatihan model. JumpStart Untuk informasi selengkapnya, lihat, Konfigurasikan nilai default untuk JumpStart model.

Bagikan Model

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat berbagi JumpStart model melalui UI Studio Classic langsung dari halaman JumpStart Aset yang diluncurkan menggunakan prosedur berikut:

1. Buka Amazon SageMaker Studio Classic dan pilih JumpStart Aset yang diluncurkan di JumpStartbagian panel navigasi kiri.
2. Pilih tab Pekerjaan pelatihan untuk melihat daftar pekerjaan pelatihan model Anda.
3. Di bawah daftar Pekerjaan pelatihan, pilih pekerjaan pelatihan yang ingin Anda bagikan. Ini membuka halaman detail pekerjaan pelatihan. Anda tidak dapat berbagi lebih dari satu pekerjaan pelatihan dalam satu waktu.
4. Di header untuk pekerjaan pelatihan, pilih Bagikan, dan pilih Bagikan ke Kanvas atau Bagikan dengan organisasi saya.

Untuk informasi selengkapnya tentang cara berbagi model dengan pengguna SageMaker Canvas, lihat Membawa Model Anda Sendiri Ke Kanvas.

> **Note**
>
> Hanya model tabular yang dapat dibagikan ke SageMaker Canvas. Mencoba membagikan model non-tabular ke SageMaker Canvas memunculkan kesalahan Tipe Data yang Tidak Didukung.

Untuk informasi selengkapnya tentang berbagi model dengan organisasi Anda, lihatModel dan Notebook Bersama.

Model dan Notebook Bersama

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Bagikan model dan buku catatan Anda untuk memusatkan artefak model, memfasilitasi kemampuan ditemukan, dan meningkatkan penggunaan kembali model dalam organisasi Anda. Saat membagikan model Anda, Anda dapat memberikan informasi lingkungan pelatihan dan inferensi dan memungkinkan kolaborator menggunakan lingkungan ini untuk pekerjaan pelatihan dan inferensi mereka sendiri.

Semua model yang Anda bagikan dan model yang dibagikan dengan Anda dapat dicari di lokasi terpusat langsung di Amazon SageMaker Studio Classic. Untuk informasi tentang langkah-langkah orientasi untuk masuk ke Amazon SageMaker Studio Classic, lihat Onboard to Amazon SageMaker Domain.

Akses model dan notebook bersama

Untuk mengakses konten bersama, pilih Model bersama di panel navigasi kiri Amazon SageMaker Studio Classic UI.

Tambahkan konten bersama

Anda dapat berbagi model atau buku catatan melalui bagian Model bersama di UI Studio Classic. Untuk detail tentang setiap langkah, lihatBagikan model dan notebook melalui UI Studio Classic.

Filter konten bersama

Ada tiga opsi utama untuk memfilter model dan notebook bersama:

1. Dibagikan oleh saya - Model dan buku catatan yang Anda bagikan ke JumpStart SageMaker Canvas.
2. Berbagi dengan saya - Model dan notebook dibagikan dengan Anda
3. Dibagikan oleh organisasi saya — Semua model dan buku catatan yang dibagikan kepada siapa pun di organisasi Anda

Anda juga dapat mengurutkan model dan buku catatan Anda berdasarkan waktu terakhir diperbarui atau dengan urutan abjad naik atau turun. Pilih ikon filter () untuk mengurutkan pilihan Anda lebih lanjut.

Bagikan model tabular dengan pengguna SageMaker Canvas

Selain berbagi model dengan organisasi Anda, Anda juga dapat berbagi model dengan kolaborator yang menggunakan SageMaker Canvas. Jika Anda membagikan model ke SageMaker Canvas, kolaborator Anda dapat mengimpor model tersebut ke SageMaker Canvas dan menggunakannya untuk menghasilkan prediksi.

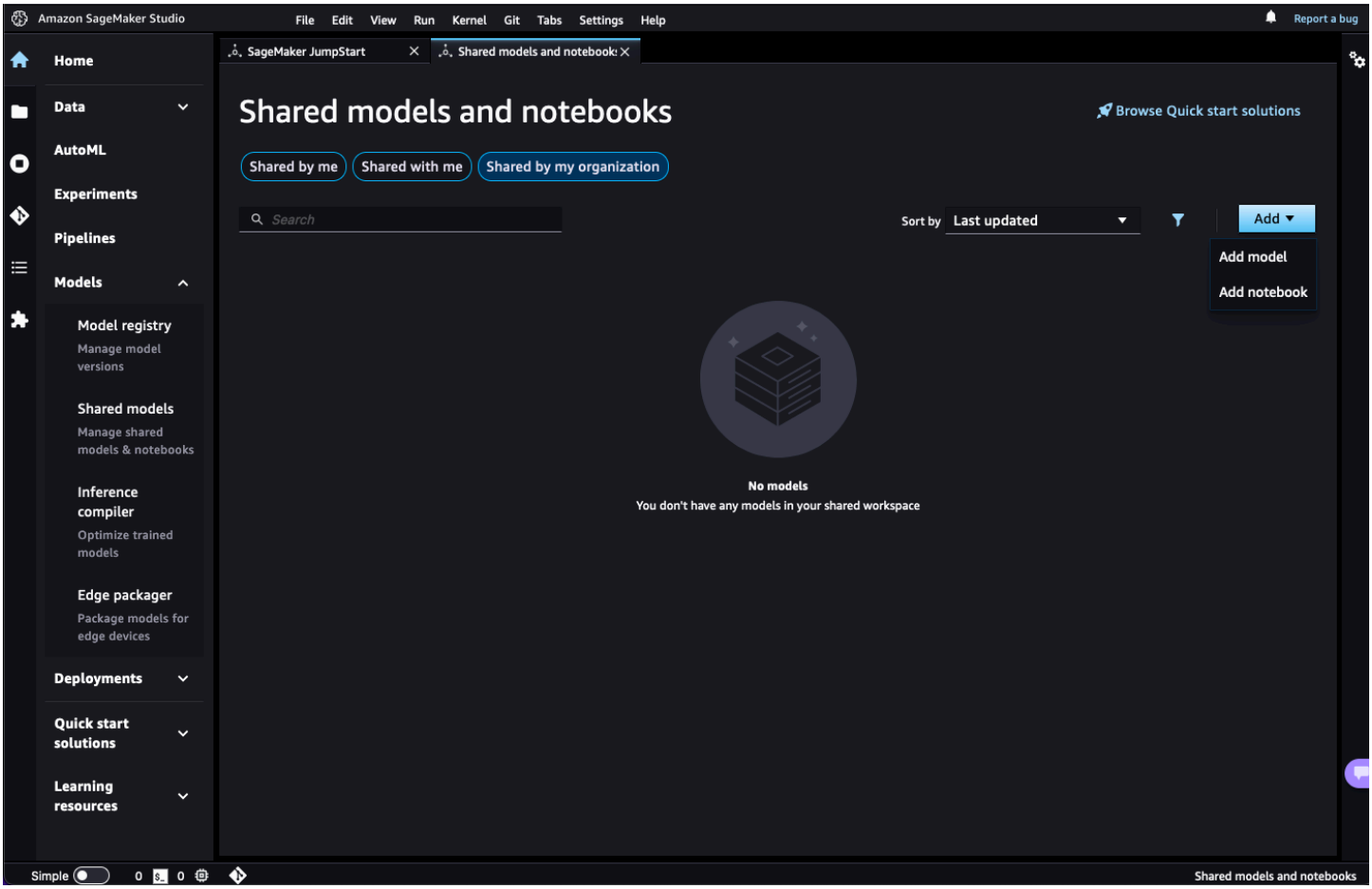
> **Important**
>
> Penting: Anda hanya dapat berbagi model tabular ke SageMaker Canvas.

Anda dapat memfilter untuk model dan buku catatan yang dibagikan ke dan dari SageMaker Canvas dengan memilih ikon filter () di tab Shared by me atau Shared with me. Untuk informasi selengkapnya tentang cara membagikan model ke SageMaker Canvas, lihat Membawa Model Anda Sendiri Ke Kanvas.

Bagikan model dan notebook melalui UI Studio Classic

Untuk berbagi model dan buku catatan, navigasikan ke bagian Model bersama di Amazon SageMaker Studio Classic, pilih Dibagikan oleh organisasi saya, lalu pilih daftar tarik-

turun Tambah. Pilih untuk menambahkan model atau menambahkan buku catatan.



Tambahkan model

Untuk menambahkan model, pilih Dibagikan oleh organisasi saya, lalu pilih Tambah model dari daftar tarik-turun Tambah. Masukkan informasi dasar untuk model Anda, dan tambahkan informasi pelatihan atau inferensi apa pun yang ingin Anda bagikan dengan kolaborator untuk melatih atau menerapkan model Anda. Setelah Anda memasukkan semua informasi yang diperlukan, pilih Tambahkan model di sudut kanan bawah.

Informasi dasar

Pertama, tambahkan informasi deskriptif dasar tentang model Anda. Informasi ini digunakan untuk meningkatkan kemampuan pencarian model Anda.

1. Tambahkan judul untuk model ini. Menambahkan judul secara otomatis mengisi pengenal unik di bidang ID berdasarkan judul model.
2. Tambahkan deskripsi model.

3. Pilih tipe data dari opsi: teks, visi, tabular, atau audio.
4. Pilih tugas pembelajaran mesin dari daftar tugas yang tersedia, seperti klasifikasi gambar atau pembuatan teks.
5. Pilih kerangka pembelajaran mesin.
6. Tambahkan informasi metadata dengan kata kunci atau frasa untuk digunakan saat mencari model. Gunakan koma untuk memisahkan kata kunci. Setiap spasi secara otomatis diganti dengan koma.

Aktifkan pelatihan

Saat menambahkan model untuk dibagikan, Anda dapat menyediakan lingkungan pelatihan secara opsional dan memungkinkan kolaborator di organisasi Anda untuk melatih model bersama.

> **Note**
>
> Jika Anda menambahkan model tabel, Anda juga perlu menentukan format kolom dan kolom target untuk mengaktifkan pelatihan. Untuk informasi selengkapnya, lihat Amazon SageMaker Canvas di Panduan SageMaker Pengembang Amazon.

1. Tambahkan wadah untuk digunakan untuk pelatihan model. Anda dapat memilih wadah yang digunakan untuk pekerjaan pelatihan yang ada, membawa wadah Anda sendiri di Amazon ECR, atau menggunakan Amazon SageMaker Deep Learning Container.
2. Tambahkan variabel lingkungan.
3. Berikan lokasi skrip pelatihan.
4. Berikan titik masuk mode skrip.
5. Berikan URI Amazon S3 untuk artefak model yang dihasilkan selama pelatihan.
6. Berikan URI Amazon S3 ke kumpulan data pelatihan default.
7. Berikan jalur keluaran model. Jalur keluaran model harus berupa jalur URI Amazon S3 untuk artefak model apa pun yang dihasilkan dari pelatihan. SageMaker menyimpan artefak model sebagai file TAR terkompresi tunggal di Amazon S3.
8. Berikan kumpulan data validasi yang akan digunakan untuk mengevaluasi model Anda selama pelatihan. Kumpulan data validasi harus berisi jumlah kolom yang sama dan header fitur yang sama dengan kumpulan data pelatihan.

9. Nyalakan isolasi jaringan. Isolasi jaringan mengisolasi wadah model sehingga tidak ada panggilan jaringan masuk atau keluar yang dapat dilakukan ke atau dari wadah model.
10. Menyediakan saluran pelatihan yang SageMaker dapat mengakses data Anda. Misalnya, Anda dapat menentukan saluran input bernama `train` atau`test`. Untuk setiap saluran, tentukan nama saluran dan URI ke lokasi data Anda. Pilih Jelajahi untuk mencari lokasi Amazon S3.
11. Berikan hiperparameter. Tambahkan hiperparameter apa pun yang harus dilakukan kolaborator selama pelatihan. Berikan berbagai nilai yang valid untuk hiperparameter ini. Rentang ini digunakan untuk melatih validasi hyperparameter pekerjaan. Anda dapat menentukan rentang berdasarkan tipe data hyperparameter.
12. Pilih jenis instance. Kami merekomendasikan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Untuk daftar lengkap instans SageMaker pelatihan di seluruh AWS Wilayah, lihat tabel Harga Sesuai Permintaan di Harga Amazon SageMaker .
13. Berikan metrik. Tentukan metrik untuk pekerjaan pelatihan dengan menentukan nama dan ekspresi reguler untuk setiap metrik yang dipantau pelatihan Anda. Rancang ekspresi reguler untuk menangkap nilai metrik yang dipancarkan algoritme Anda. Misalnya, metrik `loss` mungkin memiliki ekspresi reguler`"Loss =(.*?);"`.

Aktifkan penerapan

Saat menambahkan model untuk dibagikan, Anda dapat secara opsional menyediakan lingkungan inferensi di mana kolaborator di organisasi Anda dapat menerapkan model bersama untuk inferensi.

1. Tambahkan wadah untuk digunakan untuk inferensi. Anda dapat membawa wadah Anda sendiri di Amazon ECR atau menggunakan Amazon SageMaker Deep Learning Container.
2. Berikan URI Amazon S3 ke skrip inferensi. Skrip inferensi kustom berjalan di dalam wadah pilihan Anda. Skrip inferensi Anda harus menyertakan fungsi untuk pemuatan model, dan fungsi opsional menghasilkan prediksi, serta pemrosesan input dan output. Untuk informasi selengkapnya tentang membuat skrip inferensi untuk kerangka kerja pilihan Anda, lihat Kerangka kerja dalam dokumentasi Python SageMaker SDK. Misalnya, untuk TensorFlow, lihat Cara mengimplementasikan handler sebelum dan/atau pasca-pemrosesan.
3. Menyediakan URI Amazon S3 untuk artefak model. Artefak model adalah output yang dihasilkan dari pelatihan model, dan biasanya terdiri dari parameter terlatih, definisi model yang menjelaskan cara menghitung kesimpulan, dan metadata lainnya. Jika Anda melatih model Anda SageMaker, artefak model disimpan sebagai file TAR terkompresi tunggal di Amazon S3. Jika Anda melatih model Anda di luar SageMaker, Anda perlu membuat file TAR terkompresi tunggal ini dan menyimpannya di lokasi Amazon S3.

4. Pilih jenis instance. Kami merekomendasikan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Untuk daftar lengkap instans SageMaker pelatihan di seluruh AWS Wilayah, lihat tabel Harga Sesuai Permintaan di Harga Amazon SageMaker.

Tambahkan buku catatan

Untuk menambahkan buku catatan, pilih Dibagikan oleh organisasi saya, lalu pilih Tambahkan buku catatan dari daftar tarik-turun Tambah. Masukkan informasi dasar untuk buku catatan Anda dan berikan URI Amazon S3 untuk lokasi buku catatan tersebut.

Informasi dasar

Pertama, tambahkan informasi deskriptif dasar tentang buku catatan Anda. Informasi ini digunakan untuk meningkatkan kemampuan pencarian notebook Anda.

1. Tambahkan judul untuk buku catatan ini. Menambahkan judul secara otomatis mengisi pengenal unik di bidang ID berdasarkan judul buku catatan.
2. Tambahkan deskripsi notebook.
3. Pilih tipe data dari opsi: teks, visi, tabular, atau audio.
4. Pilih tugas ML dari daftar tugas yang tersedia, seperti klasifikasi gambar atau pembuatan teks.
5. Pilih kerangka kerja ML.
6. Tambahkan informasi metadata dengan kata kunci atau frasa untuk digunakan saat mencari buku catatan. Gunakan koma untuk memisahkan kata kunci. Setiap spasi secara otomatis diganti dengan koma.

Tambahkan buku catatan

Berikan URI Amazon S3 untuk lokasi notebook itu. Anda dapat memilih Browse untuk mencari melalui bucket Amazon S3 Anda untuk lokasi file notebook Anda. Setelah menemukan buku catatan, salin URI Amazon S3, pilih Batal, lalu tambahkan URI Amazon S3 ke bidang Lokasi Notebook.

Setelah Anda memasukkan semua informasi yang diperlukan, pilih Tambahkan buku catatan di sudut kanan bawah.

SageMaker JumpStart Industri Amazon: Keuangan

Gunakan SageMaker JumpStart Industri: Solusi keuangan, model, dan contoh notebook untuk mempelajari tentang SageMaker fitur dan kemampuan melalui solusi satu langkah yang dikuratori

dan contoh notebook masalah pembelajaran mesin (ML) yang berfokus pada industri. Notebook juga berjalan melalui cara menggunakan SDK Python SageMaker JumpStart Industri untuk meningkatkan data teks industri dan menyempurnakan model yang telah dilatih sebelumnya.

Topik

SDK Python SageMaker JumpStart Industri Amazon

SageMaker Runtime JumpStart menyediakan alat pemrosesan untuk mengkurasi kumpulan data industri dan menyempurnakan model yang telah dilatih sebelumnya melalui pustaka kliennya yang disebut Industry Python SDK. SageMaker JumpStart Untuk dokumentasi API SDK yang mendetail, dan untuk mempelajari selengkapnya tentang pemrosesan dan penyempurnaan kumpulan data teks industri untuk meningkatkan kinerja state-of-the-art model SageMaker JumpStart, lihat dokumentasi sumber terbuka SDK SageMaker JumpStartPython Industri.

SageMaker JumpStart Industri Amazon: Solusi Keuangan

SageMaker JumpStart Industri: Keuangan menyediakan notebook solusi berikut:

- Prediksi Peringkat Kredit Perusahaan

SageMaker JumpStart Industri ini: Solusi keuangan menyediakan template untuk model peringkat kredit perusahaan yang ditingkatkan teks. Ini menunjukkan bagaimana mengambil model berdasarkan fitur numerik (dalam hal ini, 5 rasio keuangan Altman yang terkenal) dikombinasikan dengan teks dari pengajuan SEC untuk mencapai peningkatan dalam prediksi peringkat kredit. Selain rasio 5 Altman, Anda dapat menambahkan lebih banyak variabel sesuai kebutuhan atau mengatur variabel khusus. Notebook solusi ini menunjukkan bagaimana SDK Python SageMaker JumpStart Industri membantu memproses penilaian Natural Language Processing (NLP) teks dari pengajuan SEC. Selain itu, solusinya menunjukkan cara melatih model menggunakan dataset yang

disempurnakan untuk mencapai best-in-class model, menyebarkan model ke SageMaker titik akhir untuk produksi, dan menerima prediksi yang ditingkatkan secara real time.

- Penilaian Kredit Berbasis Grafik

Peringkat kredit secara tradisional dihasilkan menggunakan model yang menggunakan data laporan keuangan dan data pasar, yang hanya tabular (numerik dan kategoris). Solusi ini membangun jaringan perusahaan menggunakan pengarsipan SEC dan menunjukkan bagaimana menggunakan jaringan hubungan perusahaan dengan data tabular untuk menghasilkan prediksi peringkat yang akurat. Solusi ini menunjukkan metodologi untuk menggunakan data tentang hubungan perusahaan untuk memperluas model penilaian kredit berbasis tabular tradisional, yang telah digunakan oleh industri peringkat selama beberapa dekade, ke kelas model pembelajaran mesin di jaringan.

> **Note**
>
> Notebook solusi hanya untuk tujuan demonstrasi. Mereka tidak boleh diandalkan sebagai nasihat keuangan atau investasi.

Anda dapat menemukan solusi layanan keuangan ini melalui SageMaker JumpStart halaman di Studio Classic.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

> **Note**
>
> SageMaker JumpStart Industri: Solusi keuangan, kartu model, dan contoh notebook dihosting dan dijalankan hanya melalui SageMaker Studio Classic. Masuk ke SageMaker konsol, dan luncurkan SageMaker Studio Classic. Untuk informasi selengkapnya tentang cara menemukan kartu solusi, lihat topik sebelumnya di SageMaker JumpStart.

SageMaker JumpStart Industri Amazon: Model Keuangan

SageMaker JumpStart Industri: Keuangan menyediakan model pendekatan ROBUSTLY OPTIMIZED BERT (RoberTA) yang telah dilatih sebelumnya:

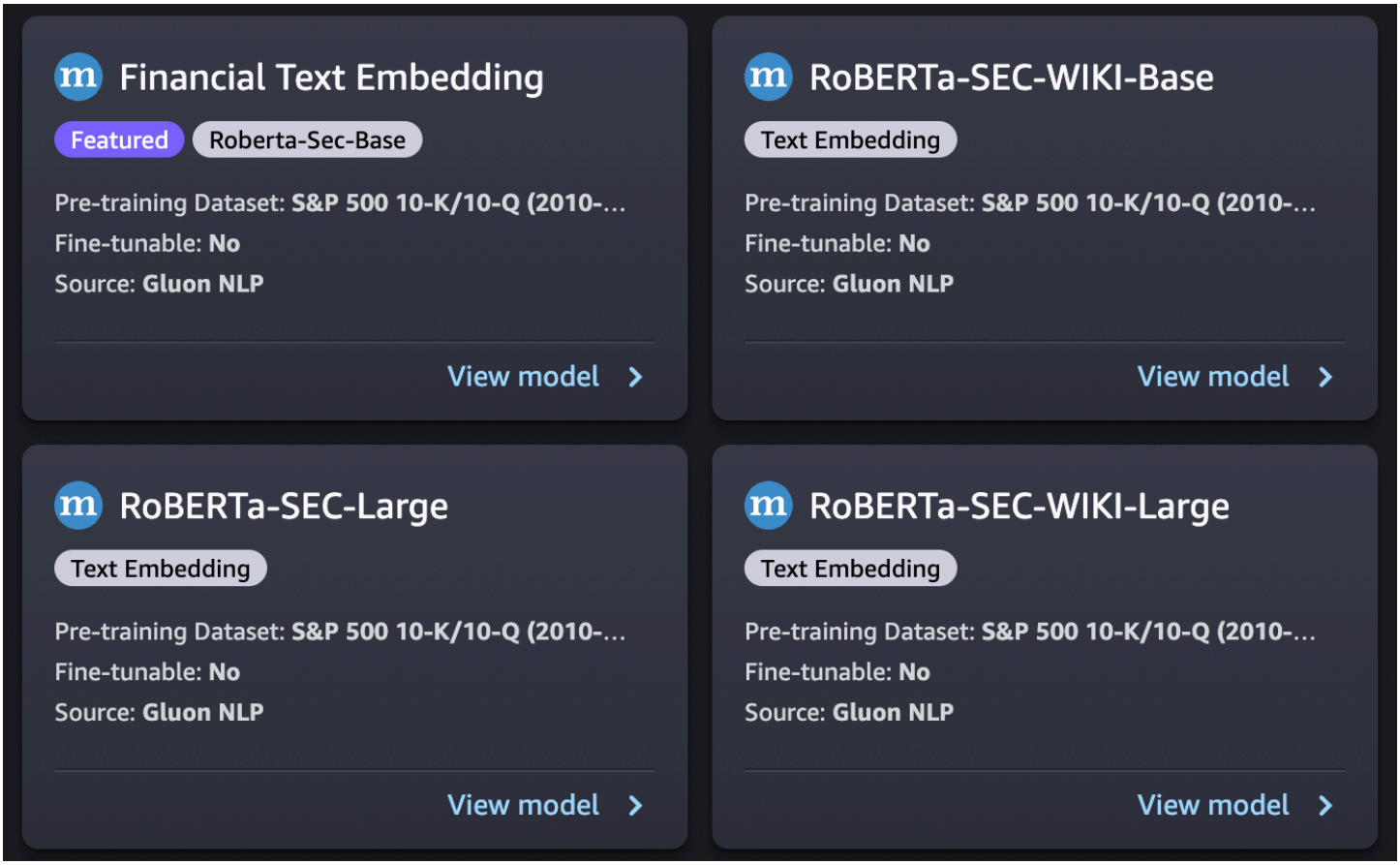
- Penyematan Teks Keuangan (Roberta-SEC-Base)
- Roberta-sec-Wiki-basis
- Roberta-sec-besar
- Roberta-sec-wiki besar

Model Roberta-SEC-Base dan Roberta-SEC-Large adalah model penyematan teks berdasarkan model RoberTA GluonNLP dan dilatih sebelumnya pada laporan S&P 500 SEC 10-K/10-Q dekade 2010 (dari 2010 hingga 2019). Selain itu, SageMaker JumpStart Industry: Financial menyediakan dua variasi Roberta lagi, Roberta-sec-Wiki-base dan Roberta-sec-Wiki-Large, yang dilatih sebelumnya pada pengajuan SEC dan teks umum Wikipedia.

Anda dapat menemukan model ini SageMaker JumpStart dengan menavigasi ke node Model Teks, memilih Jelajahi Semua Model Teks, dan kemudian memfilter untuk Penyematan Teks Tugas ML. Anda dapat mengakses notebook yang sesuai setelah memilih model pilihan Anda. Notebook yang dipasangkan akan memandu Anda melalui bagaimana model yang telah dilatih sebelumnya dapat disetel dengan baik untuk tugas klasifikasi tertentu pada kumpulan data multimodal, yang ditingkatkan oleh SDK Python Industri. SageMaker JumpStart

> **Note**
>
> Notebook model hanya untuk tujuan demonstrasi. Mereka tidak boleh diandalkan sebagai nasihat keuangan atau investasi.

Tangkapan layar berikut menunjukkan kartu model terlatih yang disediakan melalui SageMaker JumpStart halaman di Studio Classic.



> **Note**
>
> SageMaker JumpStart Industri: Solusi keuangan, kartu model, dan contoh notebook dihosting dan dijalankan hanya melalui SageMaker Studio Classic. Masuk ke SageMaker konsol, dan luncurkan SageMaker Studio Classic. Untuk informasi lebih lanjut tentang cara menemukan kartu model, lihat topik sebelumnya di SageMaker JumpStart.

SageMaker JumpStart Industri Amazon: Notebook Contoh Keuangan

SageMaker JumpStart Industri: Financial menyediakan contoh notebook berikut untuk menunjukkan solusi untuk masalah ML yang berfokus pada industri:

- Konstruksi TabText Data Keuangan — Contoh ini memperkenalkan cara menggunakan SDK Python SageMaker JumpStart Industri untuk memproses pengajuan SEC, seperti ringkasan teks dan teks penilaian berdasarkan jenis skor NLP dan daftar kata yang sesuai. Untuk melihat pratinjau konten buku catatan ini, lihat Konstruksi Sederhana Dataset Multimodal dari Pengarsipan SEC dan Skor NLP.

- Multimodal ML on TabText Data - Contoh ini menunjukkan bagaimana menggabungkan berbagai jenis dataset ke dalam kerangka data tunggal yang disebut dan melakukan multimodal. TabText Untuk melihat pratinjau konten buku catatan ini, lihat Machine Learning on a TabText Dataframe — Contoh Berdasarkan Program Perlindungan Gaji.
- Multi-kategori MLpada data pengarsipan SEC — Contoh ini menunjukkan cara melatih model AutoGluon NLP melalui kumpulan data multimodal (TabText) yang dikuratori dari pengajuan SEC untuk tugas klasifikasi multiclass. Klasifikasi Pengajuan SEC 10K/Q ke Kode Industri Berdasarkan Kolom Teks MDNA.

> **Note**
>
> Contoh notebook hanya untuk tujuan demonstratif. Mereka tidak boleh diandalkan sebagai nasihat keuangan atau investasi.

> **Note**
>
> SageMaker JumpStart Industri: Solusi keuangan, kartu model, dan contoh notebook dihosting dan dijalankan hanya melalui SageMaker Studio Classic. Masuk ke SageMaker konsol, dan luncurkan SageMaker Studio Classic. Untuk informasi selengkapnya tentang cara menemukan contoh buku catatan, lihat topik sebelumnya di SageMaker JumpStart.

Untuk melihat pratinjau konten notebook contoh, lihat Tutorial — Dokumentasi SDK Python Keuangan dalam SageMaker JumpStart Industri.

SageMaker JumpStart Industri Amazon: Postingan Blog Keuangan

Untuk aplikasi menyeluruh menggunakan SageMaker JumpStart Industri: Solusi keuangan, model, contoh, dan SDK, lihat posting blog berikut:

- Gunakan model bahasa keuangan pra-terlatih untuk pembelajaran transfer di Amazon SageMaker JumpStart
- Gunakan teks SEC untuk klasifikasi peringkat menggunakan multimodal MLdi Amazon SageMaker JumpStart
- Buat dasbor dengan teks SEC untuk NLP keuangan di Amazon SageMaker JumpStart

- Buat pengklasifikasi peringkat kredit perusahaan menggunakan pembelajaran mesin grafik di Amazon SageMaker JumpStart
- Penyesuaian Adaptasi Domain Model Foundation di Amazon pada data Keuangan SageMaker JumpStart

SageMaker JumpStart Industri Amazon: Penelitian Terkait Keuangan

Untuk penelitian yang berkaitan dengan SageMaker JumpStart Industri: Solusi keuangan, lihat makalah berikut:

- Konteks, Pemodelan Bahasa, dan Data Multimodal di bidang Keuangan
- Machine Learning Multimodal untuk Pemodelan Kredit
- Tentang Kurangnya Interpretasi yang Kuat dari Pengklasifikasi Teks Saraf
- FinLex: Penggunaan Penyematan Kata yang Efektif untuk Generasi Leksikon Keuangan

SageMaker JumpStart Industri Amazon: Sumber Daya Tambahan Keuangan

Untuk dokumentasi dan tutorial tambahan, lihat sumber daya berikut:

- SageMaker JumpStart Industri: SDK Python Keuangan
- SageMaker JumpStart Industri: Tutorial SDK Python Keuangan
- SageMaker JumpStart Industri: GitHub Repositori keuangan
- Memulai dengan Amazon SageMaker - Tutorial Machine Learning

Gunakan lingkungan pembelajaran mesin yang ditawarkan oleh SageMaker

Amazon SageMaker mendukung lingkungan pembelajaran mesin berikut.

- Amazon SageMaker Studio: Pengalaman berbasis web terbaru untuk menjalankan alur kerja ML dengan serangkaian IDE. Studio mendukung aplikasi berikut.
  - Amazon SageMaker Studio Klasik
  - Editor Kode, berdasarkan Kode-OSS, Kode Visual Studio - Sumber Terbuka
  - JupyterLab
  - SageMaker Kanvas Amazon
  - RStudio
- Amazon SageMaker Studio Classic: Memungkinkan Anda membuat, melatih, men-debug, menerapkan, dan memantau model pembelajaran mesin Anda.
- Instans SageMaker Notebook Amazon: Memungkinkan Anda menyiapkan dan memproses data, serta melatih serta menerapkan model machine learning dari instance komputasi yang menjalankan aplikasi Jupyter Notebook.
- Amazon SageMaker Studio Lab: Studio Lab adalah layanan gratis yang memberi Anda akses ke sumber daya AWS komputasi, di lingkungan berbasis sumber terbuka JupyterLab, tanpa memerlukan akunAWS.
- Amazon SageMaker Canvas: Memberi Anda kemampuan untuk menggunakan pembelajaran mesin untuk menghasilkan prediksi tanpa perlu kode.
- Amazon SageMaker geospasial: Memberi Anda kemampuan untuk membangun, melatih, dan menerapkan model geospasial.
- RStudio di Amazon SageMaker: RStudio adalah IDE untuk R, dengan konsol, editor penyorotan sintaks yang mendukung eksekusi kode langsung, dan alat untuk merencanakan, riwayat, debugging, dan manajemen ruang kerja.
- SageMaker HyperPod: SageMaker HyperPod memungkinkan Anda menyediakan cluster tangguh untuk menjalankan beban kerja machine learning (MLM) dan mengembangkan state-of-the-art model seperti model bahasa besar (LLM), model difusi, dan model dasar (FM).

Untuk menggunakan lingkungan pembelajaran mesin ini, kecuali Studio Lab, Instans SageMaker Notebook SageMaker HyperPod, dan, Anda atau administrator organisasi harus membuat SageMaker Domain Amazon. Studio Lab memiliki proses orientasi terpisah.

Topik

- SageMaker Studio Amazon
- Amazon SageMaker Studio Klasik
- SageMaker JupyterLab
- Instans SageMaker Notebook Amazon
- Lab SageMaker Studio Amazon
- SageMaker Kanvas Amazon
- Kemampuan SageMaker geospasial Amazon
- RStudio di Amazon SageMaker
- Memulai Editor Kode di Amazon SageMaker Studio
- SageMaker HyperPod
- Gunakan AI generatif di lingkungan SageMaker notebook

SageMaker Studio Amazon

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Amazon SageMaker Studio adalah pengalaman berbasis web terbaru untuk menjalankan alur kerja ML. Studio menawarkan serangkaian lingkungan pengembangan terintegrasi (IDE). Ini termasuk Code Editor, berdasarkan Code-OSS, Visual Studio Code - Open Source, JupyterLab aplikasi baru, RStudio, dan Amazon Studio Classic. SageMaker Untuk informasi selengkapnya, lihat Aplikasi yang didukung di Amazon SageMaker Studio.

UI berbasis web baru di Studio lebih cepat dan menyediakan akses ke semua SageMaker sumber daya, termasuk pekerjaan dan titik akhir, dalam satu antarmuka. Praktisi ML juga dapat

memilih IDE pilihan mereka untuk mempercepat pengembangan ML. Seorang ilmuwan data dapat menggunakannya JupyterLab untuk mengeksplorasi data dan menyetel model. Selain itu, insinyur operasi pembelajaran mesin (MLOPs) dapat menggunakan Editor Kode dengan alat pipa di Studio untuk menyebarkan dan memantau model dalam produksi.

Pengalaman Studio sebelumnya masih didukung sebagai Amazon SageMaker Studio Classic. Studio Classic adalah pengalaman default untuk pelanggan yang sudah ada, dan tersedia sebagai aplikasi di Studio. Untuk informasi selengkapnya tentang Studio Classic, lihatAmazon SageMaker Studio Klasik. Untuk informasi tentang cara bermigrasi dari Studio Classic ke Studio, lihatMigrasi dari Amazon SageMaker Studio Classic.

Studio menawarkan manfaat berikut:

- JupyterLab Aplikasi baru yang memiliki waktu start-up lebih cepat dan lebih dapat diandalkan daripada aplikasi Studio Classic yang ada. Untuk informasi selengkapnya, lihat SageMaker JupyterLab.
- Serangkaian IDE yang terbuka di tab terpisah, termasuk Editor Kode baru, berdasarkan Code-OSS, Visual Studio Code - aplikasi Open Source. Pengguna dapat berinteraksi dengan IDE yang didukung dalam pengalaman layar penuh. Untuk informasi selengkapnya, lihat Aplikasi yang didukung di Amazon SageMaker Studio.
- Akses ke semua sumber SageMaker daya Anda di satu tempat. Studio menampilkan instance yang berjalan di semua aplikasi Anda.
- Akses ke semua pekerjaan pelatihan dalam satu tampilan, terlepas dari apakah mereka dijadwalkan dari notebook atau dimulai dari Amazon. SageMaker JumpStart
- Alur kerja penerapan model yang disederhanakan dan manajemen dan pemantauan titik akhir langsung dari Studio. Anda tidak perlu mengakses SageMaker konsol.
- Pembuatan otomatis semua aplikasi yang dikonfigurasi saat Anda onboard ke domain. Untuk informasi tentang orientasi ke domain, lihatIkhtisar SageMaker Domain Amazon.
- SageMaker JumpStart Pengalaman yang lebih baik di mana Anda dapat menemukan, mengimpor, mendaftar, menyempurnakan, dan menerapkan model pondasi. Lihat informasi yang lebih lengkap di SageMaker JumpStart.

Topik

- Migrasi dari Amazon SageMaker Studio Classic
- Luncurkan Amazon SageMaker Studio

- Ikhtisar Amazon SageMaker Studio UI
- Aplikasi yang didukung di Amazon SageMaker Studio
- Ruang Amazon SageMaker Studio
- Lakukan tugas-tugas umum
- Dukungan mode lokal di Amazon SageMaker Studio
- Lihat dan hentikan instance yang sedang berjalan
- Harga Amazon SageMaker Studio
- Memecahkan masalah

Migrasi dari Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Amazon SageMaker saat ini mendukung dua pengalaman default yang berbeda: pengalaman Amazon SageMaker Studio dan pengalaman Amazon SageMaker Studio Classic. Untuk informasi tentang manfaat pengalaman Studio, lihat SageMaker Studio Amazon.

- Untuk pelanggan yang sudah ada, Amazon SageMaker Studio Classic adalah pengalaman default. Anda dapat bermigrasi ke Amazon SageMaker Studio sebagai pengalaman default menggunakan. AWS Command Line Interface Untuk informasi selengkapnya tentang Studio Classic, lihatAmazon SageMaker Studio Klasik.
- Untuk pelanggan baru, Studio adalah pengalaman default.

> **Note**
>
> Jika pengalaman Studio Classic ditetapkan sebagai default, Anda tidak memiliki akses ke fitur Studio baru.

Migrasi ke Studio dari Studio Classic dikelola oleh administrator domain. Jika Anda tidak memiliki izin untuk menetapkan Studio sebagai pengalaman default untuk domain Anda, hubungi administrator Anda.

Halaman berikut menguraikan langkah-langkah utama yang diperlukan saat bermigrasi ke Studio dari Studio Classic dengan menjadikan Studio sebagai pengalaman default untuk domain Anda. Ini juga menunjukkan cara untuk kembali ke pengalaman Studio Classic sebagai default.

Topik

Prasyarat

Prasyarat berikut diperlukan untuk bermigrasi dari Amazon Studio SageMaker Classic ke pengalaman Amazon SageMaker Studio menggunakan (). AWS Command Line Interface AWS CLI

- Perbarui AWS CLI dengan mengikuti langkah-langkah di Instal atau perbarui versi terbaru AWS CLI.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensyal, lihat Memahami dan mendapatkan kredensyal Anda AWS.

Tetapkan Studio sebagai pengalaman default

Administrator dapat menjadikan Studio sebagai pengalaman default saat membuat domain dari SageMaker konsol atau. AWS CLI Untuk informasi selengkapnya tentang menyetel pengalaman default saat membuat domain, lihatIkhtisar SageMaker Domain Amazon. Saat Anda menjadikan Studio sebagai pengalaman default untuk domain, Studio adalah pengalaman default untuk semua pengguna di domain.

Administrator hanya dapat mengubah pengalaman default saat memperbarui domain menggunakan AWS CLI atauAWS CloudFormation. Untuk informasi tentang sumber daya domain yang didukung olehAWS CloudFormation, lihat AWS::SageMaker: :Domain.

Anda dapat mengatur Studio sebagai pengalaman default saat membuat atau memperbarui domain dariAWS CLI. Untuk melakukannya, teruskan `StudioWebPortal` nilai ke panggilan create-domain atau update-domain sebagai bagian dari parameter. `default-user-settings`

`StudioWebPortal`menunjukkan apakah pengalaman Studio baru tersedia untuk pengguna. Jika `StudioWebPortal` disetel ke`DISABLED`, pengguna tidak dapat mengakses Studio di domain. Mereka hanya akan dapat menggunakan Studio Classic.

Contoh kode berikut menunjukkan cara menggunakan `StudioWebPortal` nilai:

```
aws sagemaker update-domain \
--domain-id domain-id \
--region region \
--default-user-settings '
{
    "StudioWebPortal": "ENABLED",
    "DefaultLandingUri": "studio::"
}
'
```

> **Note**
>
> Rilis Studio mencakup pembaruan pada kebijakan AWS terkelola. Untuk informasi selengkapnya, lihat SageMaker Pembaruan Kebijakan AWS Terkelola.

Batasi akses ke aplikasi default di Studio

Karena Studio menampilkan serangkaian aplikasi yang diperluas, pengguna mungkin memiliki akses ke aplikasi yang tidak ditampilkan sebelumnya. Administrator dapat membatasi akses ke aplikasi default ini dengan membuat kebijakan AWS Identity and Access Management (IAM) yang menolak akses ke aplikasi tersebut. Kebijakan kemudian harus dilampirkan pada peran eksekusi domain atau profil pengguna, seperti yang ditunjukkan dalam prosedur berikut.

1. Buat kebijakan dengan konten berikut dengan mengikuti langkah-langkah dalam Membuat kebijakan IAM:

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
   ```

```
    {
      "Sid": "AllowSageMakerCreateAppOperations",
      "Effect": "Allow",
      "Action": "sagemaker:CreateApp",
      "Resource": "*"
    },
    {
      "Sid": "DenySageMakerCreateApp",
      "Effect": "Deny",
      "Action": "sagemaker:CreateApp",
      "Resource": "arn:aws:sagemaker:region:555555555555:app/domain-id/user-
profile-name/app-type/*"
    }
  ]
}
```

2. Lampirkan kebijakan ke peran eksekusi domain atau profil pengguna yang ingin Anda tolak aksesnya. Untuk petunjuk, ikuti langkah-langkah dalam Menambahkan izin identitas IAM (konsol).

Perbarui kebijakan CORS Anda untuk mengakses bucket Amazon S3

Di Studio Classic, pengguna dapat membuat, membuat daftar, dan mengunggah file ke bucket Amazon Simple Storage Service (Amazon S3). Untuk mendukung pengalaman yang sama di Studio, administrator harus melampirkan konfigurasi Cross-Origin Resource Sharing (CORS) ke bucket Amazon S3. Ini diperlukan karena Studio membuat panggilan Amazon S3 dari browser internet. Browser memanggil CORS atas nama pengguna. Akibatnya, semua permintaan ke bucket Amazon S3 gagal kecuali kebijakan CORS dilampirkan ke Amazon S3.

Ini hanya diperlukan jika sudah ada bucket default Amazon S3 yang tidak memiliki kebijakan CORS yang benar. Jika bucket default Amazon S3 tidak ada, SageMaker buat bucket Amazon S3 dengan kebijakan CORS yang benar terlampir.

Prosedur berikut menunjukkan cara menambahkan konfigurasi CORS ke bucket Amazon S3.

Untuk menambahkan konfigurasi CORS ke bucket Amazon S3

1. Verifikasi bahwa ada bucket Amazon S3 di Wilayah yang sama dengan domain Anda dengan nama berikut. Untuk petunjuknya, lihat Melihat properti untuk bucket Amazon S3.

```
sagemaker-region-accountId
```

2. Tambahkan konfigurasi CORS dengan konten berikut ke bucket Amazon S3 default. Untuk petunjuk, lihat Mengonfigurasi berbagi sumber daya lintas asal (CORS).

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "POST",
            "PUT",
            "GET",
            "HEAD",
            "DELETE"
        ],
        "AllowedOrigins": [
            "https://*.sagemaker.aws"
        ],
        "ExposeHeaders": [
            "ETag",
            "x-amz-delete-marker",
            "x-amz-id-2",
            "x-amz-request-id",
            "x-amz-server-side-encryption",
            "x-amz-version-id"
        ]
    }
]
```

Migrasikan alur kerja dari Studio Classic ke JupyterLab

Migrasi alur kerja Anda dari aplikasi Studio Classic ke JupyterLab aplikasi melibatkan dua proses yang berbeda:

1. Mengatur Studio sebagai pengalaman default.
2. Memindahkan data dan alur kerja dari Studio Classic ke JupyterLab aplikasi.

Studio Classic dan Studio menggunakan dua jenis volume penyimpanan yang berbeda. Studio Classic menggunakan satu volume Amazon Elastic File System (Amazon EFS) untuk semua data

dalam aplikasi. Di Studio, setiap aplikasi mendapatkan volume Amazon Elastic Block Store (Amazon EBS) sendiri.

SageMaker tidak secara otomatis mentransfer data antara dua jenis volume ini saat Anda memperbarui pengalaman default domain yang ada. Akibatnya, data yang disimpan dalam volume Amazon EBS atau Amazon EFS tetap dalam volume tersebut. Jika pengguna dengan data di Studio Classic mengakses Studio, ketika pengalaman default berubah, mereka tidak akan melihat data tersebut di JupyterLab Amazon EBS.

Untuk informasi selengkapnya tentang migrasi alur kerja dari Studio Classic ke Studio JupyterLab, lihat. Migrasi dari SageMaker Studio Classic ke SageMaker Studio

Kembali ke pengalaman Studio Classic

Anda dapat kembali ke Studio Classic sebagai pengalaman default saat membuat atau memperbarui domain. Untuk melakukannya, teruskan `DISABLED` sebagai nilai untuk `StudioWebPortal` ke create-domain atau pembaruan-domain panggilan sebagai bagian dari parameter. `default-user-settings`

`StudioWebPortal`menunjukkan apakah pengalaman Studio baru tersedia untuk pengguna. Jika `StudioWebPortal` disetel ke`DISABLED`, pengguna tidak dapat mengakses Studio di domain. Mereka hanya akan dapat menggunakan Studio Classic.

Luncurkan Amazon SageMaker Studio

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Topik halaman ini menunjukkan cara meluncurkan Amazon SageMaker Studio dari SageMaker konsol Amazon dan AWS Command Line Interface (AWS CLI).

Topik

- Prasyarat

- Luncurkan dari SageMaker konsol Amazon
- Luncurkan menggunakan AWS CLI

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Onboard ke SageMaker domain dengan akses Studio. Jika Anda tidak memiliki izin untuk menetapkan Studio sebagai pengalaman default untuk domain Anda, hubungi administrator Anda. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensi Anda. Untuk informasi tentang AWS kredensi, lihat Memahami dan mendapatkan kredensi Anda AWS.

Luncurkan dari SageMaker konsol Amazon

Selesaikan prosedur berikut untuk meluncurkan Studio dari SageMaker konsol Amazon.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Dari panel navigasi kiri, pilih Studio.
3. Dari halaman landing Studio, pilih domain dan profil pengguna untuk meluncurkan Studio.
4. Pilih Open Studio.
5. Untuk meluncurkan Studio, pilih Luncurkan Studio pribadi.

Luncurkan menggunakan AWS CLI

Bagian ini menunjukkan cara meluncurkan Studio menggunakan file. AWS CLI Prosedur untuk mengakses Studio menggunakan AWS CLI tergantung jika domain menggunakan otentikasi atau AWS IAM Identity Center otentikasi AWS Identity and Access Management (IAM). Anda dapat menggunakan AWS CLI to launch Studio dengan membuat URL domain presigned saat domain Anda menggunakan autentikasi IAM. Untuk informasi tentang meluncurkan Studio dengan autentikasi IAM Identity Center, lihat. Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center

Luncurkan jika Studio adalah pengalaman default

Cuplikan kode berikut menunjukkan cara meluncurkan Studio dari AWS CLI menggunakan URL domain presigned jika Studio adalah pengalaman default. Untuk informasi lebih lanjut, lihat create-presigned-domain-url.

```
aws sagemaker create-presigned-domain-url \
--region region \
--domain-id domain-id \
--user-profile-name user-profile-name \
--session-expiration-duration-in-seconds 43200
```

Luncurkan jika Amazon SageMaker Studio Classic adalah pengalaman default Anda

Cuplikan kode berikut menunjukkan cara meluncurkan Studio dari AWS CLI menggunakan URL domain presigned jika Studio Classic adalah pengalaman default. Untuk informasi lebih lanjut, lihat create-presigned-domain-url.

```
aws sagemaker create-presigned-domain-url \
--region region \
--domain-id domain-id \
--user-profile-name user-profile-name \
--session-expiration-duration-in-seconds 43200 \
--landing-uri studio::
```

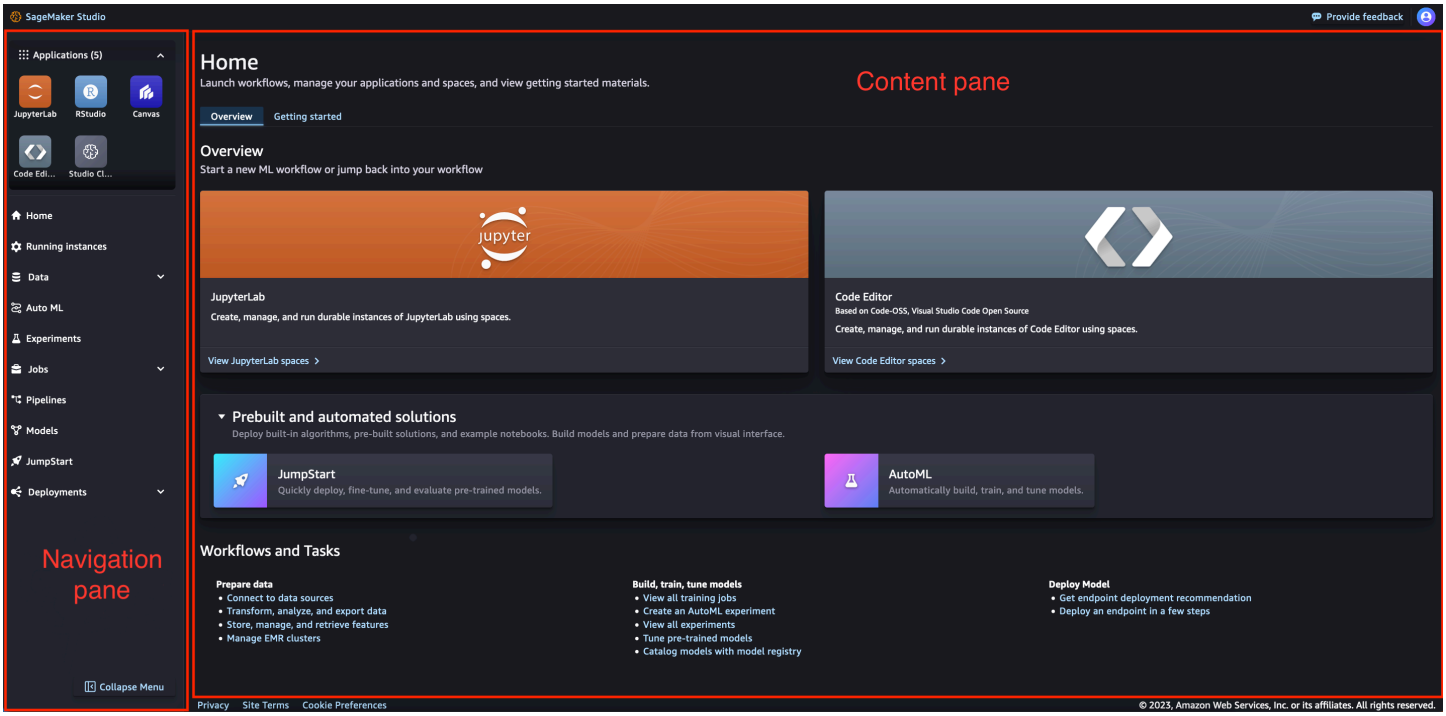
Ikhtisar Amazon SageMaker Studio UI

> **Important**
>
> Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Antarmuka pengguna Amazon SageMaker Studio dibagi menjadi tiga bagian yang berbeda.

- Bilah navigasi - Bagian UI ini mencakup URL, remah roti, notifikasi, dan opsi pengguna.
- Panel navigasi — Bagian UI ini mencakup daftar aplikasi yang didukung di Studio dan opsi untuk alur kerja utama di Studio.

- Panel konten - Area kerja utama yang menampilkan halaman UI Studio saat ini yang telah Anda buka.



Topik

- Bilah navigasi Amazon SageMaker Studio
- Panel navigasi Amazon SageMaker Studio
- Panel konten studio

Bilah navigasi Amazon SageMaker Studio

Bilah navigasi UI Studio mencakup URL, remah roti, notifikasi, dan opsi pengguna.

Struktur URL

URL Studio berubah saat Anda menavigasi UI. Saat Anda menavigasi ke halaman lain di UI, URL akan berubah untuk mencerminkan halaman tersebut. Dengan URL yang diperbarui, Anda membuka halaman apa pun di UI Studio secara langsung tanpa menavigasi ke halaman arahan terlebih dahulu.

Remah roti

Saat Anda menavigasi UI Studio, remah roti melacak halaman induk dari halaman saat ini. Dengan memilih salah satu remah roti ini, Anda dapat menavigasi ke halaman induk di UI.

Pemberitahuan

Bagian notifikasi UI memberikan informasi tentang perubahan penting pada Studio, pembaruan aplikasi, dan masalah yang harus diselesaikan.

Opsi pengguna

Opsi pengguna
()
memberikan informasi tentang profil pengguna yang saat ini menggunakan Studio, dan memberikan opsi untuk keluar dari Studio.

Panel navigasi Amazon SageMaker Studio

Panel navigasi

Panel navigasi UI menyertakan daftar aplikasi yang didukung di Studio. Ini juga menyediakan opsi untuk alur kerja utama di Studio.

Bagian UI ini dapat digunakan dalam keadaan diperluas atau diciutkan.
Untuk mengubah apakah bagian diperluas atau diciutkan, pilih ikon Ciutkan
().

Aplikasi

Bagian aplikasi mencantumkan aplikasi yang tersedia di Studio. Jika Anda memilih salah satu jenis aplikasi, Anda diarahkan ke halaman arahan untuk aplikasi itu.

Alur kerja

Daftar alur kerja mencakup semua tindakan yang tersedia yang dapat Anda lakukan di Studio. Pilih salah satu opsi untuk menavigasi ke halaman arahan untuk alur kerja tersebut. Jika ada beberapa alur kerja yang tersedia untuk opsi itu, memilih opsi membuka menu tarik-turun di mana Anda dapat memilih halaman arahan yang diinginkan.

Daftar berikut menjelaskan opsi dan menyediakan tautan untuk informasi lebih lanjut.

- Beranda — Halaman arahan utama dengan ikhtisar, memulai, dan apa yang baru.
- Menjalankan instans — Semua instance yang sedang berjalan di Studio. Untuk informasi selengkapnya, lihat Lihat dan hentikan instance yang sedang berjalan.
- Data — Opsi persiapan data tempat Anda dapat berkolaborasi untuk menyimpan, menjelajahi, menyiapkan, mengubah, dan membagikan data Anda.
  - Untuk informasi selengkapnya tentang Amazon SageMaker Data Wrangler, lihat. Siapkan data
  - Untuk informasi selengkapnya tentang Amazon SageMaker Feature Store, lihatBuat, simpan, dan bagikan fitur dengan Amazon SageMaker Feature Store.
  - Untuk informasi selengkapnya tentang klaster EMR Amazon, lihat. Siapkan data menggunakan Amazon EMR
- Auto ML - Secara otomatis membangun, melatih, menyetel, dan menerapkan model machine learning (ML). Untuk informasi selengkapnya, lihat SageMaker Kanvas Amazon.
- Eksperimen — Buat, kelola, analisis, dan bandingkan eksperimen pembelajaran mesin Anda menggunakanAmazon SageMaker Experiments. Untuk informasi lebih lanjut, lihatKelola Machine Learning dengan Amazon SageMaker Experiments.
- Pekerjaan — Lihat pekerjaan yang dibuat di Studio.
  - Untuk informasi lebih lanjut tentang pelatihan, lihatMelatih model pembelajaran mesin.
  - Untuk informasi lebih lanjut tentang evaluasi model, lihatGunakan SageMaker Clarify untuk mengevaluasi model pondasi.
- Pipelines — Otomatiskan alur kerja ML Anda dengan Amazon SageMaker Model Building Pipelines, yang menyediakan sumber daya untuk membantu Anda membangun, melacak, dan mengelola sumber daya pipeline. Untuk informasi selengkapnya, lihat Pipa Bangunan SageMaker Model Amazon.
- Model — Atur model Anda ke dalam grup dan koleksi di registri model, tempat Anda dapat mengelola versi model, melihat metadata, dan menerapkan model ke produksi. Untuk informasi selengkapnya, lihat Daftarkan dan Terapkan Model dengan Registri Model.
- JumpStartAmazon SageMaker JumpStart menyediakan model sumber terbuka yang sudah terlatih untuk berbagai jenis masalah untuk membantu Anda memulai pembelajaran mesin. Untuk informasi lebih lanjut, lihatSageMaker JumpStart.
- Deployment - Menerapkan model machine learning (ML) Anda untuk inferensi.
  - Untuk informasi selengkapnya tentang Amazon SageMaker Inference Recommender, lihat. Rekomendasi SageMaker Inferensi Amazon
  - Untuk informasi selengkapnya tentang titik akhir, lihatMenyebarkan model untuk inferensi.

Panel konten studio

Area kerja utama juga disebut panel konten. Ini menampilkan halaman UI Studio saat ini yang telah Anda buka.

Halaman rumah studio

Halaman beranda Studio adalah halaman arahan utama di area kerja utama. Halaman beranda mencakup dua tab yang berbeda. Ada tab Ikhtisar dan tab Memulai.

Ikhtisar

Tab Ikhtisar mencakup opsi untuk memulai spasi untuk jenis aplikasi populer, memulai dengan solusi pra-bangun dan otomatis untuk alur kerja ML, dan menautkan ke tugas umum di UI Studio.

Memulai

Tab Memulai mencakup informasi, panduan, dan sumber daya tentang cara memulai dengan Studio. Ini termasuk tur berpemandu ke UI Studio, tautan ke dokumentasi tentang Studio, dan pilihan kiat cepat.

Aplikasi yang didukung di Amazon SageMaker Studio

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Amazon SageMaker Studio mendukung aplikasi berikut:

- Code Editor, berdasarkan Code-OSS, Visual Studio Code - Open Source - Code Editor menawarkan lingkungan pengembangan terintegrasi (IDE) yang ringan dan kuat dengan pintasan, terminal, dan kemampuan debugging canggih dan alat refactoring yang sudah dikenal. Ini adalah aplikasi berbasis browser yang dikelola sepenuhnya di Studio. Untuk informasi selengkapnya, lihat Memulai Editor Kode di Amazon SageMaker Studio.
- Amazon SageMaker Studio Classic — Amazon SageMaker Studio Classic adalah IDE berbasis web untuk pembelajaran mesin. Dengan Studio Classic, Anda dapat membuat, melatih,

men-debug, menerapkan, dan memantau model pembelajaran mesin Anda. Untuk informasi selengkapnya, lihat Amazon SageMaker Studio Klasik.

- JupyterLab— JupyterLab menawarkan serangkaian kemampuan yang menambah penawaran notebook yang dikelola sepenuhnya. Ini mencakup kernel yang dimulai dalam hitungan detik, runtime yang telah dikonfigurasi sebelumnya dengan ilmu data populer, kerangka kerja pembelajaran mesin, dan penyimpanan blok kinerja tinggi. Untuk informasi selengkapnya, lihat SageMaker JupyterLab.
- Amazon SageMaker Canvas — Dengan SageMaker Canvas, Anda dapat menggunakan pembelajaran mesin untuk menghasilkan prediksi tanpa menulis kode. Dengan Canvas, Anda dapat mengobrol dengan model bahasa besar (LLM) populer, mengakses ready-to-use model, atau membuat model khusus yang dilatih pada data Anda. Untuk informasi selengkapnya, lihat SageMaker Kanvas Amazon.
- RStudio — RStudio adalah lingkungan pengembangan terintegrasi untuk R. Ini termasuk konsol dan editor penyorotan sintaks yang mendukung menjalankan kode secara langsung. Ini juga mencakup alat untuk merencanakan, riwayat, debugging, dan manajemen ruang kerja. Lihat informasi yang lebih lengkap di RStudio di Amazon SageMaker.

Ruang Amazon SageMaker Studio

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Spasi digunakan untuk mengelola penyimpanan dan kebutuhan sumber daya dari beberapa aplikasi Amazon SageMaker Studio. Setiap ruang memiliki hubungan 1:1 dengan instance aplikasi. Setiap aplikasi yang didukung yang dibuat mendapat ruangnya sendiri. Aplikasi berikut di Studio berjalan pada spasi:

- Memulai Editor Kode di Amazon SageMaker Studio
- SageMaker JupyterLab
- Amazon SageMaker Studio Klasik

Sebuah ruang terdiri dari sumber daya berikut:

- Volume penyimpanan.
  - Untuk Studio Classic, ruang terhubung ke volume Amazon Elastic File System (Amazon EFS) bersama untuk domain.
  - Untuk aplikasi lain, volume Amazon Elastic Block Store (Amazon EBS) yang berbeda dilampirkan ke ruang. Semua aplikasi diberi volume Amazon EBS mereka sendiri. Aplikasi tidak memiliki akses ke volume Amazon EBS dari aplikasi lain. Untuk informasi selengkapnya tentang volume Amazon EBS, lihat Amazon Elastic Block Store (Amazon EBS).
- Jenis aplikasi ruang.
- Gambar yang menjadi dasar aplikasi.

Spasi dapat bersifat pribadi atau bersama:

- Private: Ruang pribadi dicakup untuk satu pengguna dalam domain. Ruang pribadi tidak dapat dibagikan dengan pengguna lain. Semua aplikasi yang mendukung ruang juga mendukung ruang pribadi.
- Bersama: Ruang bersama dapat diakses oleh semua pengguna di domain. Hanya Studio Classic yang mendukung ruang bersama. Untuk informasi selengkapnya tentang spasi bersama, lihatBerkolaborasi dengan ruang bersama.

Spasi dapat dibuat dalam domain yang menggunakan salah satu AWS IAM Identity Center atau AWS Identity and Access Management (IAM) otentikasi. Bagian berikut memberikan informasi umum tentang cara mengakses spasi. Untuk informasi spesifik tentang membuat dan mengakses spasi, lihat dokumentasi untuk jenis aplikasi masing-masing ruang yang Anda buat.

Topik

- Ruang akses

Ruang akses

Bagian berikut menunjukkan cara mengakses daftar spasi yang terkait dengan profil pengguna di domain.

Mengakses spasi dari konsol Amazon SageMaker

Untuk mengakses spasi dari SageMaker konsol Amazon

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di bawah konfigurasi Admin, pilih Domain.
3. Dari daftar domain, pilih domain yang berisi spasi.
4. Pada halaman Detail domain, pilih tab Manajemen ruang. Untuk informasi selengkapnya tentang mengelola ruang, lihatBerkolaborasi dengan ruang bersama.
5. Dari daftar spasi untuk domain itu, pilih ruang yang akan diluncurkan.
6. Pilih Launch Studio untuk ruang yang ingin Anda luncurkan.

Mengakses ruang dari Studio

Ikuti langkah-langkah ini untuk mengakses spasi dari Studio untuk jenis aplikasi tertentu.

Untuk mengakses ruang dari Studio

1. Buka Studio dengan mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Pilih jenis aplikasi dengan spasi yang ingin Anda akses.

Mengakses spasi menggunakan AWS CLI

Bagian berikut menunjukkan cara mengakses spasi dari AWS Command Line Interface (AWS CLI). Prosedurnya adalah untuk domain yang menggunakan AWS Identity and Access Management (IAM) atau AWS IAM Identity Center otentikasi.

Autentikasi IAM

Prosedur berikut menguraikan secara umum cara mengakses ruang menggunakan otentikasi IAM dari. AWS CLI

1. Buat URL domain presigned yang menentukan nama ruang yang ingin Anda akses.

```
aws \
    --region region \
    sagemaker \
    create-presigned-domain-url \
```

```
    --domain-id domain-id \
    --user-profile-name user-profile-name \
    --space-name space-name
```

2. Arahkan ke URL.

Mengakses spasi dalam autentikasi IAM Identity Center

Prosedur berikut menguraikan cara mengakses ruang menggunakan autentikasi IAM Identity Center dari. AWS CLI

1. Gunakan perintah berikut untuk mengembalikan URL yang terkait dengan spasi.

```
aws \
    --region region \
    sagemaker \
    describe-space \
    --domain-id domain-id \
    --space-name space-name
```

2. Tambahkan parameter pengalihan masing-masing untuk jenis aplikasi ke URL yang akan difederasi melalui IAM Identity Center. Untuk informasi selengkapnya tentang parameter pengalihan, lihat deskripsi-spasi.
3. Arahkan ke URL untuk difederasi melalui IAM Identity Center.

Lakukan tugas-tugas umum

> **Important**
>
> Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Bagian berikut menjelaskan cara melakukan tugas umum di Amazon SageMaker Studio. Untuk gambaran umum tentang antarmuka pengguna Studio, lihatIkhtisar Amazon SageMaker Studio UI.

Tetapkan preferensi cookie

1. Luncurkan Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Di bagian bawah antarmuka pengguna Studio, pilih Preferensi Cookie.
3. Pilih kotak centang untuk setiap jenis cookie yang Anda SageMaker ingin Amazon gunakan.
4. Pilih Simpan preferensi.

Kelola notifikasi

Pemberitahuan memberikan informasi tentang perubahan penting pada Studio, pembaruan aplikasi, dan masalah yang harus diselesaikan.

1. Luncurkan Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Di bilah navigasi atas, pilih ikon Pemberitahuan (

).
3. Dari daftar notifikasi, pilih notifikasi untuk mendapatkan informasi tentangnya.

Tinggalkan umpan balik

Kami menanggapi tanggapan Anda dengan serius. Kami mendorong Anda untuk memberikan umpan balik.

Di bagian atas navigasi Studio, pilih Berikan umpan balik.

Keluar

Keluar dari UI Studio berbeda dengan menutup jendela browser. Keluar menghapus data sesi dari browser dan menghapus perubahan yang belum disimpan.

Perilaku yang sama ini juga terjadi ketika waktu sesi Studio habis. Ini terjadi setelah 5 menit.

1. Luncurkan Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Pilih ikon Opsi pengguna (
).
3. Pilih Keluar.
4. Di jendela pop-up, pilih Keluar.

Dukungan mode lokal di Amazon SageMaker Studio

Aplikasi Amazon SageMaker Studio mendukung penggunaan mode lokal untuk membuat estimator, prosesor, dan pipeline, lalu menerapkannya ke lingkungan lokal. Mode lokal memungkinkan pengguna menguji skrip pembelajaran mesin sebelum menjalankannya di lingkungan pelatihan atau hosting SageMaker terkelola Amazon. Studio mendukung mode lokal dalam aplikasi berikut:

- Amazon SageMaker Studio Klasik
- JupyterLab
- Editor Kode, berdasarkan Kode-OSS, Kode Visual Studio - Sumber Terbuka

Mode lokal dalam aplikasi Studio dipanggil menggunakan SageMaker Python SDK. Mode lokal di aplikasi Studio berfungsi mirip dengan mode lokal di instance SageMaker Notebook Amazon, dengan beberapa perbedaan. Untuk informasi selengkapnya tentang penggunaan mode lokal dengan SageMaker Python SDK, lihat Mode Lokal.

> **Note**
>
> Aplikasi studio tidak mendukung pekerjaan multi-kontainer dalam mode lokal. Pekerjaan mode lokal terbatas pada satu contoh untuk pelatihan, inferensi, dan pekerjaan pemrosesan. Saat membuat pekerjaan mode lokal, konfigurasi hitungan instance harus1.

Sebagai bagian dari dukungan mode lokal, aplikasi Studio mendukung kemampuan akses Docker terbatas. Dengan dukungan ini, pengguna dapat berinteraksi dengan Docker API dari notebook Jupyter atau terminal gambar aplikasi. Pelanggan dapat berinteraksi dengan Docker menggunakan salah satu dari berikut ini:

- CLI Docker
- CLI Tulis Docker
- klien SDK Docker khusus bahasa

Prasyarat

Anda harus menyelesaikan prasyarat berikut untuk menggunakan mode lokal di aplikasi Studio.

- Untuk menarik gambar dari repositori Amazon Elastic Container Registry, akun yang menghosting image Amazon ECR harus memberikan izin akses untuk peran eksekusi pengguna. Peran eksekusi domain juga harus memungkinkan akses Amazon ECR.
- Verifikasi bahwa Anda menggunakan versi terbaru Studio Python SDK menggunakan perintah berikut:

```
pip install -U sagemaker
```

- Untuk menggunakan mode lokal, serta kemampuan Docker, Anda harus mengatur parameter domain berikut `DockerSettings` menggunakan: AWS CLI

```
EnableDockerAccess : ENABLED
```

- Dengan menggunakan`EnableDockerAccess`, Anda juga dapat mengontrol apakah pengguna di domain dapat menggunakan mode lokal. Secara default, mode lokal dan kemampuan Docker tidak diizinkan dalam aplikasi Studio. Untuk informasi selengkapnya, lihat [Pengaturan `EnableDockerAccess`](#).
- Instal CLI Docker di aplikasi Studio mengikuti langkah-langkahnya. [Instalasi Docker](#)

Pengaturan **`EnableDockerAccess`**

Bagian berikut menunjukkan cara mengatur `EnableDockerAccess` kapan domain memiliki akses internet publik atau dalam `VPC-only` mode.

> **Note**
>
> Perubahan `EnableDockerAccess` hanya berlaku untuk aplikasi yang dibuat setelah domain diperbarui. Anda harus membuat aplikasi baru setelah memperbarui domain.

Akses internet publik

Contoh perintah berikut menunjukkan cara mengatur `EnableDockerAccess` saat membuat domain baru atau memperbarui domain yang ada dengan akses internet publik.

```
# create new domain
aws --region region \
    sagemaker create-domain --domain-name domain-name \
```

```
    --vpc-id vpc-id \
    --subnet-ids subnet-ids \
    --auth-mode IAM \
    --default-user-settings "ExecutionRole=execution-role" \
    --domain-settings '{"DockerSettings": {"EnableDockerAccess": "ENABLED"}}' \
    --query DomainArn \
    --output text

# update domain
aws --region region \
    sagemaker update-domain --domain-id domain-id \
    --domain-settings-for-update '{"DockerSettings": {"EnableDockerAccess":
 "ENABLED"}}'
```

`VPC-only`modus

Saat menggunakan domain dalam `VPC-only` mode, permintaan push dan pull image Docker dirutekan melalui VPC layanan alih-alih VPC yang dikonfigurasi oleh pelanggan. Karena fungsi ini, administrator dapat mengonfigurasi daftar AWS akun tepercaya yang dapat digunakan pengguna untuk menarik dan mendorong permintaan operasi Amazon ECR Docker. Jika permintaan push atau pull image Docker dibuat ke AWS akun yang tidak ada dalam daftar AWS akun tepercaya, permintaan gagal. Operasi tarik dan dorong Docker di luar Amazon ECR tidak didukung dalam `VPC-only` mode.

AWSAkun berikut dipercaya secara default:

- Akun hosting SageMaker domain.
- SageMaker akun yang menampung SageMaker gambar-gambar berikut:
- • Gambar kerangka kerja DLC
  - Sklearn, Spark, XGBoost memproses gambar

Untuk mengonfigurasi daftar AWS akun tepercaya tambahan, Anda harus menentukan `VpcOnlyTrustedAccounts` nilainya sebagai berikut:

```
aws --region region \
    sagemaker update-domain --domain-id domain-id \
    --domain-settings-for-update '{"DockerSettings": {"EnableDockerAccess": "ENABLED",
 "VpcOnlyTrustedAccounts": ["account-list"]}}'
```

Dukungan Docker

Studio juga mendukung kemampuan akses Docker terbatas dengan batasan berikut:

- Penggunaan jaringan Docker tidak didukung.
- Penggunaan volume Docker tidak didukung selama penampung dijalankan. Hanya input volume bind mount yang diizinkan selama orkestrasi kontainer. Input pemasangan pengikat volume harus ditempatkan pada volume Amazon Elastic File System untuk Studio Classic atau volume Amazon Elastic Block Store untuk JupyterLab dan aplikasi Editor Kode.
- Operasi pemeriksaan kontainer diperbolehkan.
- Port kontainer ke pemetaan host tidak diperbolehkan. Namun, Anda dapat menentukan port untuk hosting. Titik akhir kemudian dapat diakses dari Studio menggunakan URL berikut.

```
http://localhost:port
```

Operasi Docker didukung

Tabel berikut mencantumkan semua titik akhir Docker API yang didukung di Studio, serta batasan dukungan apa pun. Studio tidak mendukung titik akhir API yang tidak tercantum dalam tabel.

Dokumentasi API	Batasan
SystemAuth	
SystemEvents	
SystemVersion	
SystemPing	
SystemPingHead	
ContainerCreate	• Kontainer tidak dapat dijalankan di jembatan default Docker atau jaringan Docker khusus. Kontainer dijalankan di jaringan yang sama dengan wadah aplikasi Studio.

Dokumentasi API	Batasan
	• Pengguna hanya dapat menggunakan nilai berikut untuk nama jaringan:`sagemaker` . Sebagai contoh: `docker run --net sagemaker parameter-values` • Hanya pengikat pengikat yang diizinkan untuk penggunaan volume. Direktori host harus ada di Amazon EFS untuk KernelGateway aplikasi atau Amazon EBS untuk aplikasi lain. • Kontainer tidak dapat berjalan dalam mode priviledged atau dengan priviledge seccomp yang ditinggikan.
ContainerStart	
ContainerStop	
ContainerKill	
ContainerDelete	
ContainerList	
ContainerLogs	
ContainerInspect	
ContainerWait	
ContainerAttach	
ContainerPrune	
ContainerResize	

Dokumentasi API	Batasan
ImageCreate	`VPC-only`dukungan terbatas pada gambar Amazon ECR di akun yang diizinkan.
ImagePrune	
ImagePush	`VPC-only`dukungan terbatas pada gambar Amazon ECR di akun yang diizinkan.
ImageList	
ImageInspect	
ImageGet	
ImageDelete	
ImageBuild	• `VPC-only`dukungan terbatas pada gambar Amazon ECR di akun yang diizinkan. • Pengguna hanya dapat menggunakan nilai berikut untuk nama jaringan:`sagemaker` . Sebagai contoh: `docker build --network sagemaker parameter-values`

Instalasi Docker

Untuk menggunakan Docker, Anda harus menginstal Docker secara manual dari terminal aplikasi Studio Anda.

1. Arahkan ke terminal aplikasi Studio tempat Anda ingin menginstal Docker.
2. Instal Docker mengikuti petunjuk di https://docs.docker.com/engine/install/ubuntu/ dengan pertimbangan berikut:
   - Instal Docker menggunakan repositori apt. Menggunakan metode lain untuk menginstal Docker akan gagal.

- Saat menggunakan Docker dari Studio Classic, hapus `sudo` saat menjalankan perintah karena terminal sudah berjalan di bawah`superuser`.
- Jika perintah dirantai gagal, jalankan perintah satu per satu.
- Studio hanya mendukung versi `20.10.X.` Docker.
- Paket-paket berikut tidak diperlukan untuk menggunakan CLI Docker di Studio dan instalasinya dapat dilewati:
  - `containerd.io`
  - `docker-ce`

> **Note**
>
> Anda tidak perlu memulai Layanan Docker di aplikasi Anda. Instance yang meng-host aplikasi Studio menjalankan Docker Service secara default. Semua panggilan API Docker dirutekan melalui Layanan Docker secara otomatis.

3. Gunakan soket Docker yang terbuka untuk interaksi Docker dalam aplikasi Studio. Secara default, soket berikut terbuka.

```
unix:///docker/proxy.sock
```

Variabel lingkungan aplikasi Studio berikut untuk default `USER` menggunakan soket terbuka ini.

```
DOCKER_HOST
```

Lihat dan hentikan instance yang sedang berjalan

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Halaman instans yang sedang berjalan memberikan informasi tentang semua instance aplikasi yang sedang berjalan yang dibuat di Amazon SageMaker Studio oleh pengguna, atau dibagikan dengan pengguna.

Pada halaman ini, Anda dapat melihat dan berhenti menjalankan instance untuk semua aplikasi dan spasi Anda. Jika sebuah instance dihentikan, itu tidak muncul di halaman ini. Instans yang dihentikan dapat dilihat dari halaman arahan untuk jenis aplikasi masing-masing.

Topik

Lihat detail aplikasi

Anda dapat melihat daftar aplikasi yang sedang berjalan dan detailnya di Studio.

Untuk melihat instance yang sedang berjalan

1. Luncurkan Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Menjalankan instance.
3. Dari halaman Running instance, Anda dapat melihat daftar aplikasi yang sedang berjalan dan detail tentang aplikasi tersebut.

Hentikan aplikasi yang sedang berjalan

Anda dapat menghentikan aplikasi yang sedang berjalan di Studio.

Untuk menghentikan aplikasi yang sedang berjalan

1. Luncurkan Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Dari daftar instance yang sedang berjalan, pilih aplikasi yang ingin Anda hentikan.
3. Pilih tombol Stop untuk aplikasi.

Harga Amazon SageMaker Studio

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Tidak ada biaya tambahan untuk menggunakan Amazon SageMaker Studio UI.

Berikut ini menimbulkan biaya:

- Amazon Elastic Block Store atau Amazon Elastic File System volume yang dipasang dengan aplikasi Anda.
- Pekerjaan dan sumber daya apa pun yang diluncurkan pengguna dari aplikasi Studio.
- Meluncurkan JupyterLab aplikasi, bahkan jika tidak ada sumber daya atau pekerjaan yang diluncurkan dalam aplikasi.

Untuk informasi tentang cara Amazon SageMaker Studio Classic ditagih, lihatHarga Amazon SageMaker Studio Classic.

Untuk informasi selengkapnya tentang penagihan beserta contoh harga, lihat SageMaker Harga Amazon.

Memecahkan masalah

> **Important**
>
> Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Bagian ini menunjukkan cara memecahkan masalah umum di Amazon SageMaker Studio.

Tidak dapat menghapus Editor Kode, berdasarkan Code-OSS, Visual Studio Code - Open Source atau aplikasi JupyterLab

Masalah ini terjadi ketika pengguna membuat aplikasi dari Amazon SageMaker Studio yang hanya tersedia di Studio, lalu kembali ke pengalaman Studio Classic sebagai default mereka. Akibatnya, pengguna tidak dapat menghapus aplikasi untuk Editor Kode, berdasarkan Code-OSS, Visual Studio Code - Open Source atau JupyterLab karena mereka tidak dapat mengakses UI Studio.

Untuk mengatasi masalah ini, beri tahu administrator Anda sehingga mereka dapat menghapus aplikasi secara manual menggunakan AWS Command Line Interface (AWS CLI).

Amazon SageMaker Studio Klasik

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Amazon SageMaker Studio Classic adalah lingkungan pengembangan terintegrasi (IDE) berbasis web untuk pembelajaran mesin yang memungkinkan Anda membuat, melatih, men-debug, menerapkan, dan memantau model pembelajaran mesin Anda. Studio Classic menyediakan semua alat yang Anda butuhkan untuk mengambil model Anda dari persiapan data hingga eksperimen hingga produksi sambil meningkatkan produktivitas Anda. Dalam satu antarmuka visual terpadu, pelanggan dapat melakukan tugas-tugas berikut:

- Tulis dan jalankan kode di notebook Jupyter
- Mempersiapkan data untuk pembelajaran mesin
- Membangun dan melatih model pembelajaran mesin
- Menyebarkan model dan memantau kinerja prediksi mereka
- Lacak dan debug eksperimen pembelajaran mesin

Untuk informasi tentang langkah-langkah orientasi untuk masuk ke Studio Classic, lihatIkhtisar SageMaker Domain Amazon.

Untuk AWS Wilayah yang didukung oleh Studio Classic, lihatWilayah dan kuota yang didukung.

Topik

Fitur Studio Klasik

Studio Classic mencakup fitur-fitur berikut:

- SageMaker Autopilot
- SageMaker Klarifikasi
- SageMaker Data Wrangler
- SageMaker Debugger
- SageMaker Eksperimen
- SageMaker Toko Fitur
- SageMaker JumpStart
- Pipa Bangunan SageMaker Model Amazon
- SageMaker Registri Model
- SageMaker Proyek
- SageMakerNotebook Studio Klasik
- SageMaker Buku Catatan Universal Studio

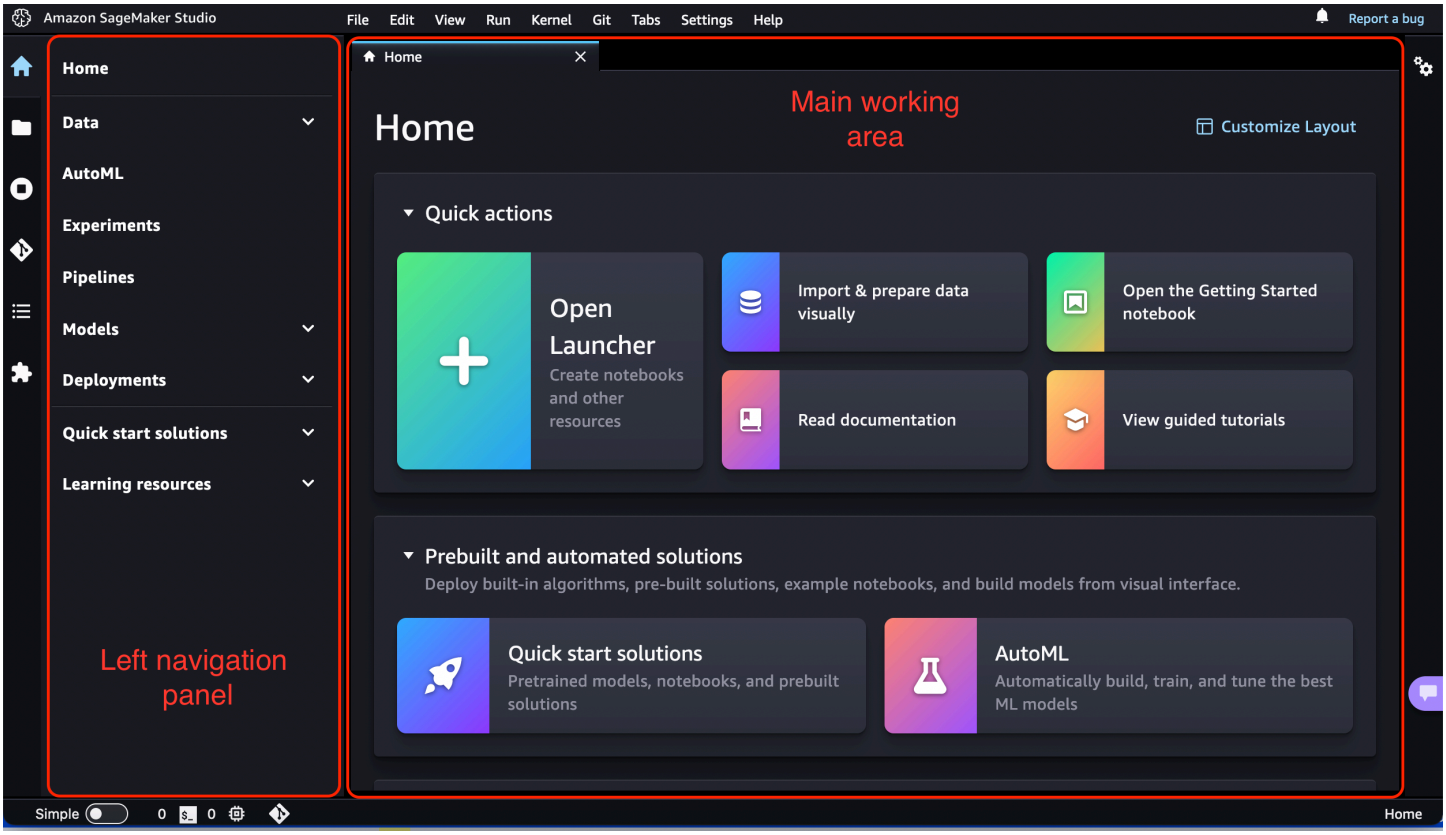
Ikhtisar UI Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Amazon SageMaker Studio Classic memperluas kemampuan JupyterLab dengan sumber daya khusus yang dapat mempercepat proses Machine Learning (ML) Anda dengan memanfaatkan kekuatan komputasi. AWS Pengguna sebelumnya JupyterLab akan melihat kesamaan antarmuka pengguna. Penambahan yang paling menonjol dirinci di bagian berikut. Untuk ikhtisar JupyterLab antarmuka asli, lihat JupyterLabAntarmuka.

Gambar berikut menunjukkan tampilan default saat meluncurkan Amazon SageMaker Studio Classic. Panel navigasi kiri menampilkan semua kategori fitur tingkat atas, dan a Halaman Beranda Studio Klasikterbuka di area kerja utama. Kembalilah ke titik pusat orientasi ini dengan memilih ikon Home () kapan saja, lalu pilih simpul Beranda di menu navigasi.

Coba buku catatan Memulai untuk panduan langsung dalam produk tentang cara mengatur dan membiasakan diri dengan fitur Amazon SageMaker Studio Classic. Pada bagian Tindakan cepat di halaman Beranda Studio Klasik, pilih Buka buku catatan Memulai.



> **Note**
>
> Bab ini didasarkan pada antarmuka pengguna (UI) Studio Classic yang diperbarui yang tersedia pada versi `v5.38.x` dan di atasnya pada JupyterLab 3.
>
> - Untuk mengambil versi Studio Classic UI Anda, dari Studio Classic Launcher, buka Terminal Sistem, lalu
>
>   1. Jalankan `conda activate studio`
>   2. Jalankan `jupyter labextension list`
>   3. Cari versi yang ditampilkan setelah `@amzn/sagemaker-ui version` di output.
> - Untuk informasi tentang memperbarui Amazon SageMaker Studio Classic, lihatMatikan dan Perbarui SageMaker Studio Classic.

Topik

- Halaman Beranda Studio Klasik
- Tata letak Studio Klasik

Halaman Beranda Studio Klasik

Halaman Beranda menyediakan akses ke tugas umum dan alur kerja. Secara khusus, ini mencakup daftar tindakan Cepat untuk tugas-tugas umum seperti Open Launcher untuk membuat notebook dan sumber daya lainnya dan Impor & menyiapkan data secara visual untuk membuat aliran baru di Data Wrangler. Halaman Beranda juga menawarkan tooltips tentang kontrol utama di UI.

Solusi bawaan dan otomatis membantu Anda memulai dengan cepat dengan solusi kode rendah seperti Amazon SageMaker JumpStart dan SageMaker Autopilot.

Di Alur Kerja dan tugas, Anda dapat menemukan daftar tugas yang relevan untuk setiap langkah alur kerja ML Anda yang membawa Anda ke alat yang tepat untuk pekerjaan itu. Misalnya, Transform, analisis, dan ekspor data akan membawa Anda ke Amazon SageMaker Data Wrangler dan membuka alur kerja untuk membuat alur data baru, atau Lihat semua eksperimen akan membawa Anda ke Eksperimen dan membuka tampilan daftar SageMaker eksperimen.

Setelah peluncuran Studio Classic, halaman Beranda terbuka di area kerja utama.
Anda dapat menyesuaikan halaman SageMaker Beranda Anda dengan memilih

Sesuai
Tata Letak di kanan atas tab Beranda.

Tata letak Studio Klasik

Antarmuka Amazon SageMaker Studio Classic terdiri dari bilah menu di bagian atas, bilah sisi kiri yang dapat dilipat menampilkan berbagai ikon seperti ikon Beranda dan Browser File, bilah status di bagian bawah layar, dan area pusat dibagi secara horizontal menjadi dua panel. Panel kiri adalah panel navigasi yang dapat dilipat. Panel kanan, atau area kerja utama, berisi satu atau lebih tab untuk sumber daya seperti peluncur, notebook, terminal, metrik, dan grafik, dan dapat dibagi lebih lanjut.

Laporkan bug di Studio Classic atau pilih ikon notifikasi
(
)
untuk melihat notifikasi dari Studio Classic, seperti versi Studio Classic baru dan SageMaker fitur baru, di sudut kanan bilah menu. Untuk memperbarui ke versi baru Studio Classic, lihatMatikan dan Perbarui Aplikasi SageMaker Studio Classic dan Studio Classic.

Bagian berikut menjelaskan area antarmuka pengguna utama Studio Classic.

Sidebar kiri

Sidebar kiri mencakup ikon berikut. Saat mengarahkan kursor ke ikon, tooltip menampilkan nama ikon. Satu klik pada ikon membuka panel navigasi kiri dengan fungsionalitas yang dijelaskan. Klik dua kali meminimalkan panel navigasi kiri.

Ikon	Deskripsi
	Rumah Pilih ikon Beranda untuk membuka menu navigasi tingkat atas di panel navigasi kiri. Dengan menggunakan menu navigasi Beranda, Anda dapat menemukan dan menavigasi ke alat yang tepat untuk setiap langkah alur kerja ML Anda. Menu ini juga menyediakan pintasan untuk solusi mulai cepat dan sumber belajar seperti dokumentasi dan tutorial terpandu. Kategori menu mengelompokkan fitur yang relevan bersama-s ama. Memilih Data, misalnya, memperluas SageMaker kemampuan yang relevan untuk tugas persiapan data Anda. Dari sini, Anda dapat menyiapkan data Anda dengan Data Wrangler, membuat dan menyimpan fitur ML dengan Amazon SageMaker Feature Store, dan mengelola klaster EMR Amazon untuk pemrosesan data skala besar. Kategori diurutkan mengikuti alur kerja ML yang khas mulai dari menyiapkan data, hingga membangun, melatih, dan menerapkan model ML (data, pipeline, model, dan penerapan). Ketika Anda memilih node tertentu (seperti Data Wrangler), halaman yang sesuai terbuka di area kerja utama. Pilih Beranda di menu navigasi untuk membuka Halaman Beranda Studio Klasik
	Browser Berkas File Browser menampilkan daftar buku catatan, eksperimen, uji coba, komponen uji coba, titik akhir, dan solusi kode rendah.

Ikon	Deskripsi
	Apakah Anda berada di ruang pribadi atau bersama menentukan siapa yang memiliki akses ke file Anda. Anda dapat mengidentifikasi jenis ruang yang Anda masuki dengan melihat sudut kanan atas. ***Jika Anda berada di aplikasi pribadi, Anda melihat ikon pengguna diikuti oleh [user_name] /Personal Studio dan jika Anda berada di ruang kolaboratif, Anda melihat ikon globe diikuti oleh "[user_name]/[space_name] ."*** • Aplikasi Personal Studio Classic: Direktori Amazon EFS pribadi yang hanya dapat Anda akses. • Ruang kolaboratif: Direktori Amazon EFS bersama dengan anggota tim Anda yang lain untuk akses grup ke buku catatan dan sumber daya. Bekerja di ruang bersama memungkinkan kolaborasi tim real-time di notebook. • Peluncur Studio Classic: Pilih tanda plus (+) pada menu di bagian atas browser file untuk membuka Amazon SageMaker Studio Classic Launcher. • Unggah file: Pilih ikon Unggah File () untuk menambahkan file ke Studio Classic atau seret dan jatuhkan dari desktop Anda. • Buka file: Klik dua kali file untuk membuka file di tab baru atau klik kanan dan pilih Buka. • Manajemen panel: Untuk bekerja di file yang berdekatan, pilih tab yang berisi buku catatan, Python, atau file teks, lalu pilih Tampilan Baru untuk File.

Ikon	Deskripsi
	Untuk entri hierarkis, breadcrumb yang dapat dipilih di bagian atas browser menunjukkan lokasi Anda dalam hierarki.
	Inspektur Properti Property Inspector adalah inspektur alat sel notebook yang menampilkan pengaturan properti kontekstual saat dibuka.
	Menjalankan Terminal dan Kernel Anda dapat memeriksa daftar semua kernel dan terminal yang saat ini berjalan di semua notebook, konsol kode, dan direktori. Anda dapat mematikan sumber daya individual, termasuk notebook, terminal, kernel, aplikasi, dan instance. Anda juga dapat mematikan semua sumber daya di salah satu kategori ini secara bersamaan. Untuk informasi selengkapnya, lihat Matikan Sumber Daya.
	Git Anda dapat terhubung ke repositori Git dan kemudian mengakses berbagai alat dan operasi Git. Untuk informasi selengkapnya, lihat Mengkloning Repositori Git di Studio Classic SageMaker .
	Daftar Isi Anda dapat menavigasi struktur dokumen saat file notebook atau Python terbuka. Daftar isi dibuat secara otomatis di panel navigasi kiri saat Anda membuka buku catatan, file Markdown, atau file Python. Entri dapat diklik dan gulir dokumen ke judul yang dimaksud.

Ikon	Deskripsi
	Ekstensi Anda dapat mengaktifkan dan mengelola JupyterLab ekstensi pihak ketiga. Anda dapat memeriksa ekstensi yang sudah diinstal dan mencari ekstensi dengan mengetikkan nama di bilah pencarian. Ketika Anda telah menemukan ekstensi yang ingin Anda instal, pilih Instal. Setelah menginstal ekstensi baru Anda, pastikan untuk memulai ulang JupyterLab dengan menyegarkan browser Anda. Untuk informasi selengkapnya, lihat dokumentasi JupyterLab Ekstensi.

Panel navigasi kiri

Konten panel navigasi kiri bervariasi dengan Ikon yang dipilih di sidebar kiri.

Misalnya, memilih ikon Beranda menampilkan menu navigasi. Memilih File browser mencantumkan semua file dan direktori yang tersedia di ruang kerja Anda (buku catatan, eksperimen, alur data, uji coba, komponen uji coba, titik akhir, atau solusi kode rendah).

Di menu navigasi, memilih node menampilkan halaman fitur yang sesuai di area kerja utama. Misalnya, memilih Data Wrangler di menu Data membuka tab Data Wrangler yang mencantumkan semua alur yang ada.

Wilayah kerja utama

Area kerja utama terdiri dari beberapa tab yang berisi buku catatan terbuka, terminal, dan informasi terperinci tentang eksperimen dan titik akhir Anda. Di area kerja utama, Anda dapat mengatur dokumen (seperti buku catatan dan file teks) dan aktivitas lainnya (seperti terminal dan konsol kode) ke dalam panel tab yang dapat Anda ubah ukurannya atau bagi lagi. Seret tab ke tengah panel tab untuk memindahkan tab ke panel. Bagilah panel tab dengan menyeret tab ke kiri, kanan, atas, atau bawah panel. Tab untuk aktivitas saat ini ditandai dengan batas atas berwarna (biru secara default).

> **Note**
>
> Semua halaman fitur memberikan bantuan kontekstual dalam produk. Untuk mengakses bantuan, pilih Tampilkan informasi. Antarmuka bantuan memberikan pengantar singkat tentang alat dan tautan ke sumber daya tambahan, seperti video, tutorial, atau blog.

Luncurkan Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Setelah Anda onboard ke SageMaker Domain Amazon, Anda dapat meluncurkan aplikasi Amazon SageMaker Studio Classic baik dari SageMaker konsol atau. AWS CLI Untuk informasi selengkapnya tentang orientasi ke Domain, lihatIkhtisar SageMaker Domain Amazon.

Topik

- Luncurkan Studio Classic Menggunakan SageMaker Konsol Amazon
- Luncurkan Studio Classic Menggunakan AWS CLI

Luncurkan Studio Classic Menggunakan SageMaker Konsol Amazon

Proses untuk menavigasi ke Studio Classic dari Amazon SageMaker Console berbeda tergantung pada apakah Studio Classic atau Amazon SageMaker Studio ditetapkan sebagai pengalaman default untuk domain Anda. Untuk informasi selengkapnya tentang menyetel pengalaman default untuk domain Anda, lihatMigrasi dari Amazon SageMaker Studio Classic.

Topik

- Prasyarat

Prasyarat

Untuk menyelesaikan prosedur ini, Anda harus melakukan onboard ke Domain dengan mengikuti langkah-langkah di Onboard to Amazon SageMaker Domain.

Luncurkan Studio Classic jika Studio adalah pengalaman default Anda

1. Arahkan ke Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Dari UI Studio, cari panel aplikasi di sisi kiri.

3. Dari panel aplikasi, pilih Studio Classic.
4. Dari halaman landing Studio Classic, pilih instance Studio Classic yang akan dibuka.
5. Pilih “Buka”.

Luncurkan Studio Classic Menggunakan AWS CLI

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) untuk meluncurkan Amazon SageMaker Studio Classic dengan membuat URL Domain presigned.

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Onboard ke SageMaker Domain Amazon. Untuk informasi selengkapnya, lihat Onboard to Amazon SageMaker Domain.
- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensil Anda. Untuk informasi tentang AWS kredensyal, lihat Memahami dan mendapatkan kredensyal Anda AWS.

Cuplikan kode berikut menunjukkan cara meluncurkan Amazon SageMaker Studio Classic dari AWS CLI menggunakan URL Domain yang telah ditetapkan sebelumnya. Untuk informasi lebih lanjut, lihat create-presigned-domain-url.

```
aws sagemaker create-presigned-domain-url \
--region region \
--domain-id domain-id \
--space-name space-name \
--user-profile-name user-profile-name \
--session-expiration-duration-in-seconds 43200
```

JupyterLab Pembuatan Versi

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan

aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Antarmuka Amazon SageMaker Studio Classic didasarkan pada JupyterLab, yang merupakan lingkungan pengembangan interaktif berbasis web untuk notebook, kode, dan data. Studio Classic sekarang mendukung penggunaan JupyterLab 1 dan JupyterLab 3. Versi default JupyterLab di Studio Classic adalah JupyterLab 3. Jika Anda membuat SageMaker Domain Amazon dan profil pengguna menggunakan AWS Management Console sebelum 31/08/2022 atau menggunakan 22/02/23 AWS Command Line Interface sebelum, instans Studio Classic Anda default ke 1. JupyterLab Setelah 08/31/2022, versi JupyterLab 1 di Amazon SageMaker Studio Classic hanya menerima perbaikan keamanan. Anda dapat memilih versi yang ingin Anda jalankan. Namun, Anda hanya dapat menjalankan satu instance JupyterLab sekaligus per profil pengguna. Anda tidak dapat menjalankan beberapa versi secara JupyterLab bersamaan.

Setelah 03/31/23, Studio Classic hanya mendukung pembuatan 3 aplikasi. JupyterLab Setelah tanggal tersebut, Studio Classic berhenti mendukung JupyterLab 1 pembuatan aplikasi. Pada 04/30/2023, Studio Classic menghapus semua aplikasi yang ada yang menjalankan 1. JupyterLab Perbarui JupyterLab 1 aplikasi Anda yang ada ke JupyterLab 3 sebelum 04/30/2023 mengikuti langkah-langkahnya. Lihat dan perbarui JupyterLab versi aplikasi dari konsol

Topik

JupyterLab 3

JupyterLab 3 termasuk fitur berikut yang tidak tersedia di versi sebelumnya. Untuk informasi selengkapnya tentang fitur-fitur ini, lihat JupyterLab 3.0 dirilis! .

- Visual debugger saat menggunakan kernel Base Python 2.0 dan Data Science 2.0.
- Filter peramban file
- Daftar Isi (TOC)

- Dukungan multi-bahasa
- Modus sederhana
- Mode antarmuka tunggal

Perubahan penting ke JupyterLab 3

Pertimbangkan hal berikut saat menggunakan JupyterLab 3:

- Saat mengatur JupyterLab versi menggunakanAWS CLI, pilih gambar yang sesuai untuk Wilayah Anda dan JupyterLab versi dari daftar gambar diDari AWS CLI.
- Di JupyterLab 3, Anda harus mengaktifkan lingkungan `studio` conda sebelum menginstal ekstensi. Untuk informasi selengkapnya, lihat Instalasi JupyterLab dan ekstensi Server Jupyter.
- Debugger hanya didukung saat menggunakan gambar berikut:
  - Dasar Python 2.0
  - Ilmu Data 2.0
  - Dasar Python 3.0
  - Ilmu Data 3.0

Membatasi JupyterLab versi default menggunakan kunci kondisi kebijakan IAM

Anda dapat menggunakan kunci kondisi kebijakan IAM untuk membatasi versi JupyterLab yang dapat diluncurkan pengguna Anda.

Kebijakan berikut menunjukkan cara membatasi JupyterLab versi di tingkat Domain.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Block users from creating JupyterLab 3 apps at the domain level",
            "Effect": "Deny",
            "Action": [
                "sagemaker:CreateDomain",
                "sagemaker:UpdateDomain"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "ForAnyValue:StringLike": {
                    "sagemaker:ImageArns": "*image/jupyter-server-3"
                }
            }
        }
    ]
}
```

Kebijakan berikut menunjukkan cara membatasi JupyterLab versi di tingkat profil pengguna.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Block users from creating JupyterLab 3 apps at the user profile
 level",
            "Effect": "Deny",
            "Action": [
                "sagemaker:CreateUserProfile",
                "sagemaker:UpdateUserProfile"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "sagemaker:ImageArns": "*image/jupyter-server-3"
                }
            }
        }
    ]
}
```

Kebijakan berikut menunjukkan cara membatasi JupyterLab versi di tingkat aplikasi. `CreateApp`Permintaan harus menyertakan gambar ARN agar kebijakan ini berlaku.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Block users from creating JupyterLab 3 apps at the application
 level",
            "Effect": "Deny",
            "Action": "sagemaker:CreateApp",
            "Resource": "*",
```

```
            "Condition": {
                "ForAnyValue:StringLike": {
                    "sagemaker:ImageArns": "*image/jupyter-server-3"
                }
            }
        }
    ]
}
```

Menyetel JupyterLab versi default

Bagian berikut menunjukkan cara mengatur JupyterLab versi default untuk Studio Classic menggunakan konsol atauAWS CLI.

Dari konsol

Anda dapat memilih JupyterLab versi default yang akan digunakan pada tingkat Domain atau profil pengguna selama pembuatan sumber daya. Untuk mengatur JupyterLab versi default menggunakan konsol, lihatIkhtisar SageMaker Domain Amazon.

Dari AWS CLI

Anda dapat memilih JupyterLab versi default yang akan digunakan pada tingkat Domain atau profil pengguna menggunakanAWS CLI.

Untuk mengatur JupyterLab versi default menggunakanAWS CLI, Anda harus menyertakan ARN dari JupyterLab versi default yang diinginkan sebagai bagian dari perintahAWS CLI. ARN ini berbeda berdasarkan versi dan Wilayah Domain. SageMaker

Tabel berikut mencantumkan ARN dari JupyterLab versi yang tersedia untuk setiap Wilayah:

Wilayah	JL1	JL3
us-east-1	arn:aws:sagemaker:us-east-1:081325390199:image/jupyter-server	arn:aws:sagemaker:us-east-1:081325390199:image/jupyter-server-3
us-east-2	arn:aws:sagemaker:us-east-2:429704687514:image/jupyter-server	arn:aws:sagemaker:us-east-2:429704687514:image/jupyter-server-3

us-west-1	arn:aws:sagemaker:us-west-1:742091327244:image/jupyter-server	arn:aws:sagemaker:us-west-1:742091327244:image/jupyter-server-3
us-west-2	arn:aws:sagemaker:us-west-2:236514542706:image/jupyter-server	arn:aws:sagemaker:us-west-2:236514542706:image/jupyter-server-3
af-south-1	arn:aws:sagemaker:af-south-1:559312083959:image/jupyter-server	arn:aws:sagemaker:af-south-1:559312083959:image/jupyter-server-3
ap-east-1	arn:aws:sagemaker:ap-east-1:493642496378:image/jupyter-server	arn:aws:sagemaker:ap-east-1:493642496378:image/jupyter-server-3
ap-south-1	arn:aws:sagemaker:ap-south-1:394103062818:image/jupyter-server	arn:aws:sagemaker:ap-south-1:394103062818:image/jupyter-server-3
ap-northeast-2	arn:aws:sagemaker:ap-northeast-2:806072073708:image/jupyter-server	arn:aws:sagemaker:ap-northeast-2:806072073708:image/jupyter-server-3
ap-southeast-1	arn:aws:sagemaker:ap-southeast-1:492261229750:image/jupyter-server	arn:aws:sagemaker:ap-southeast-1:492261229750:image/jupyter-server-3
ap-southeast-2	arn:aws:sagemaker:ap-southeast-2:452832661640:image/jupyter-server	arn:aws:sagemaker:ap-southeast-2:452832661640:image/jupyter-server-3
ap-northeast-1	arn:aws:sagemaker:ap-northeast-1:102112518831:image/jupyter-server	arn:aws:sagemaker:ap-northeast-1:102112518831:image/jupyter-server-3
ca-central-1	arn:aws:sagemaker:ca-central-1:310906938811:image/jupyter-server	arn:aws:sagemaker:ca-central-1:310906938811:image/jupyter-server-3

eu-central-1	arn:aws:sagemaker:eu-central-1:936697816551:image/jupyter-server	arn:aws:sagemaker:eu-central-1:936697816551:image/jupyter-server-3
eu-west-1	arn:aws:sagemaker:eu-west-1:470317259841:image/jupyter-server	arn:aws:sagemaker:eu-west-1:470317259841:image/jupyter-server-3
eu-west-2	arn:aws:sagemaker:eu-west-2:712779665605:image/jupyter-server	arn:aws:sagemaker:eu-west-2:712779665605:image/jupyter-server-3
eu-west-3	arn:aws:sagemaker:eu-west-3:615547856133:image/jupyter-server	arn:aws:sagemaker:eu-west-3:615547856133:image/jupyter-server-3
eu-north-1	arn:aws:sagemaker:eu-north-1:243637512696:image/jupyter-server	arn:aws:sagemaker:eu-north-1:243637512696:image/jupyter-server-3
eu-south-1	arn:aws:sagemaker:eu-south-1:592751261982:image/jupyter-server	arn:aws:sagemaker:eu-south-1:592751261982:image/jupyter-server-3
sa-east-1	arn:aws:sagemaker:sa-east-1:782484402741:image/jupyter-server	arn:aws:sagemaker:sa-east-1:782484402741:image/jupyter-server-3
cn-north-1	arn:aws-cn:sagemaker:cn-north-1:390048526115:image/jupyter-server	arn:aws-cn:sagemaker:cn-north-1:390048526115:image/jupyter-server-3
cn-northwest-1	arn:aws-cn:sagemaker:cn-northwest-1:390780980154:image/jupyter-server	arn:aws-cn:sagemaker:cn-northwest-1:390780980154:image/jupyter-server-3

Buat atau perbarui Domain

Anda dapat mengatur JupyterServer versi default di tingkat Domain dengan memanggil CreateDomainatau UpdateDomaindan meneruskan `UserSettings.JupyterServerAppSettings.DefaultResourceSpec.SageMakerImageArn` bidang.

Berikut ini menunjukkan cara membuat Domain dengan JupyterLab 3 sebagai default, menggunakanAWS CLI:

```
aws --region <REGION> \
sagemaker create-domain \
--domain-name <NEW_DOMAIN_NAME> \
--auth-mode <AUTHENTICATION_MODE> \
--subnet-ids <SUBNET-IDS> \
--vpc-id <VPC-ID> \
--default-user-settings '{
  "JupyterServerAppSettings": {
    "DefaultResourceSpec": {
      "SageMakerImageArn": "arn:aws:sagemaker:<REGION>:<ACCOUNT_ID>:image/jupyter-
server-3",
      "InstanceType": "system"
    }
  }
}'
```

Berikut ini menunjukkan cara memperbarui Domain untuk menggunakan JupyterLab 3 sebagai default, menggunakanAWS CLI:

```
aws --region <REGION> \
sagemaker update-domain \
--domain-id <YOUR_DOMAIN_ID> \
--default-user-settings '{
  "JupyterServerAppSettings": {
    "DefaultResourceSpec": {
      "SageMakerImageArn": "arn:aws:sagemaker:<REGION>:<ACCOUNT_ID>:image/jupyter-
server-3",
      "InstanceType": "system"
    }
  }
}'
```

Membuat atau memperbarui profil pengguna

Anda dapat mengatur JupyterServer versi default di tingkat profil pengguna dengan memanggil CreateUserProfileatau UpdateUserProfiledan meneruskan `UserSettings.JupyterServerAppSettings.DefaultResourceSpec.SageMakerImageArn` bidang.

Berikut ini menunjukkan cara membuat profil pengguna dengan JupyterLab 3 sebagai default pada Domain yang ada, menggunakanAWS CLI:

```
aws --region <REGION> \
sagemaker create-user-profile \
--domain-id <YOUR_DOMAIN_ID> \
--user-profile-name <NEW_USERPROFILE_NAME> \
--query UserProfileArn --output text \
--user-settings '{
  "JupyterServerAppSettings": {
    "DefaultResourceSpec": {
      "SageMakerImageArn": "arn:aws:sagemaker:<REGION>:<ACCOUNT_ID>:image/jupyter-
server-3",
      "InstanceType": "system"
    }
  }
}'
```

Berikut ini menunjukkan cara memperbarui profil pengguna untuk menggunakan JupyterLab 3 sebagai default, menggunakanAWS CLI:

```
aws --region <REGION> \
sagemaker update-user-profile \
 --domain-id <YOUR_DOMAIN_ID> \
 --user-profile-name <EXISTING_USERPROFILE_NAME> \
--user-settings '{
  "JupyterServerAppSettings": {
    "DefaultResourceSpec": {
      "SageMakerImageArn": "arn:aws:sagemaker:<REGION>:<ACCOUNT_ID>:image/jupyter-
server-3",
      "InstanceType": "system"
    }
  }
}'
```

Lihat dan perbarui JupyterLab versi aplikasi dari konsol

Berikut ini menunjukkan cara melihat dan memperbarui JupyterLab versi aplikasi.

1. Arahkan ke halaman SageMaker Domain.
2. Pilih domain untuk melihat profil penggunanya.
3. Pilih pengguna untuk melihat aplikasi mereka.
4. Untuk melihat JupyterLab versi aplikasi, pilih nama aplikasi.
5. Untuk memperbarui JupyterLab versi, pilih Tindakan.
6. Dari menu tarik-turun, pilih Ubah JupyterLab versi.
7. Dari halaman pengaturan Studio Classic, pilih JupyterLab versi dari menu tarik-turun.
8. Setelah JupyterLab versi untuk profil pengguna berhasil diperbarui, restart JupyterServer aplikasi untuk membuat perubahan versi efektif. Untuk informasi selengkapnya tentang memulai ulang JupyterServer aplikasi, lihat. Matikan dan Perbarui SageMaker Studio Classic

Instalasi JupyterLab dan ekstensi Server Jupyter

Proses untuk menginstal JupyterLab dan ekstensi Jupyter Server berbeda tergantung pada JupyterLab versi instans Studio Classic Anda. Di JupyterLab 1, Anda dapat membuka terminal dan menginstal ekstensi tanpa mengaktifkan lingkungan conda apa pun. Di JupyterLab 3, Anda harus mengaktifkan lingkungan `studio` conda sebelum menginstal ekstensi. Metode untuk ini berbeda jika Anda menginstal ekstensi dari dalam Studio Classic atau menggunakan skrip konfigurasi siklus hidup.

Menginstal Ekstensi dari dalam Studio Classic

Untuk menginstal ekstensi dari dalam Studio Classic, Anda harus mengaktifkan `studio` lingkungan sebelum Anda menginstal ekstensi.

```
# Before installing extensions
conda activate studio

# Install your extensions
pip install <JUPYTER_EXTENSION>

# After installing extensions
conda deactivate
```

Menginstal Ekstensi menggunakan skrip konfigurasi siklus hidup

Jika Anda menginstal JupyterLab dan ekstensi Jupyter Server dalam skrip konfigurasi siklus hidup Anda, Anda harus memodifikasi skrip Anda sehingga berfungsi dengan 3. JupyterLab Bagian berikut menunjukkan kode yang diperlukan untuk skrip konfigurasi siklus hidup yang ada dan yang baru.

Skrip konfigurasi siklus hidup yang ada

Jika Anda menggunakan kembali skrip konfigurasi siklus hidup yang ada yang harus bekerja dengan kedua versi JupyterLab, gunakan kode berikut dalam skrip Anda:

```
# Before installing extension
export
 AWS_SAGEMAKER_JUPYTERSERVER_IMAGE="${AWS_SAGEMAKER_JUPYTERSERVER_IMAGE:-'jupyter-
server'}"
if [ "$AWS_SAGEMAKER_JUPYTERSERVER_IMAGE" = "jupyter-server-3" ] ; then
   eval "$(conda shell.bash hook)"
   conda activate studio
fi;

# Install your extensions
pip install <JUPYTER_EXTENSION>


# After installing extension
if [ "$AWS_SAGEMAKER_JUPYTERSERVER_IMAGE" = "jupyter-server-3" ]; then
   conda deactivate
fi;
```

Skrip konfigurasi siklus hidup baru

Jika Anda menulis skrip konfigurasi siklus hidup baru yang hanya menggunakan JupyterLab 3, Anda dapat menggunakan kode berikut dalam skrip Anda:

```
# Before installing extension
eval "$(conda shell.bash hook)"
conda activate studio


# Install your extensions
pip install <JUPYTER_EXTENSION>
```

```
conda deactivate
```

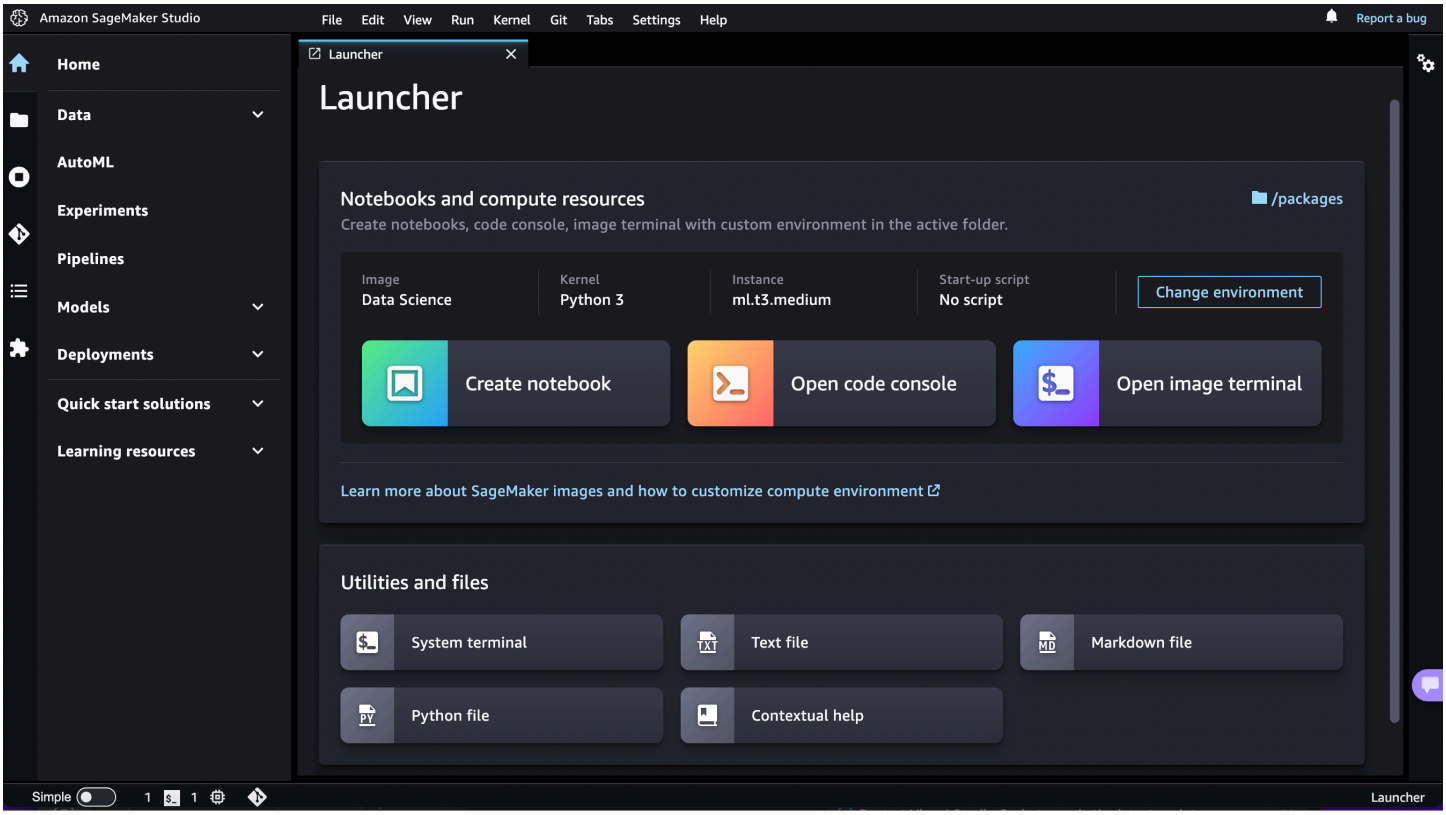
Gunakan Amazon SageMaker Studio Classic Launcher

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat menggunakan Amazon SageMaker Studio Classic Launcher untuk membuat notebook dan file teks, dan untuk meluncurkan terminal dan shell Python interaktif.

Anda dapat membuka Studio Classic Launcher dengan salah satu cara berikut:

- Pilih Amazon SageMaker Studio Classic di kiri atas antarmuka Studio Classic.
- Gunakan pintasan `Ctrl + Shift + L` keyboard.
- Dari menu Studio Classic, pilih File dan kemudian pilih Peluncur Baru.
- Jika browser SageMaker file terbuka, pilih tanda plus (+) di menu browser file Studio Classic.
- Di bagian Tindakan cepat pada tab Beranda, pilih Buka Peluncur. Peluncur terbuka di tab baru. Bagian Tindakan cepat terlihat secara default tetapi dapat dimatikan. Pilih Sesuaikan Tata Letak untuk mengaktifkan kembali bagian ini.



Peluncur terdiri dari dua bagian berikut:

Topik

- Notebook dan sumber daya komputasi
- Utilitas dan file

Notebook dan sumber daya komputasi

Di bagian ini, Anda dapat membuat buku catatan, membuka terminal gambar, atau membuka konsol Python.

Untuk membuat atau meluncurkan salah satu item tersebut:

1. Pilih Ubah lingkungan untuk memilih SageMaker gambar, kernel, jenis instance, dan, secara opsional, tambahkan skrip konfigurasi siklus hidup yang berjalan pada awal gambar. Untuk informasi selengkapnya tentang skrip konfigurasi siklus hidup, lihat. Menggunakan konfigurasi siklus hidup dengan Amazon Studio Classic SageMaker Untuk informasi selengkapnya tentang pembaruan kernel, lihatMengubah Gambar atau Kernel.
2. Pilih item.

> **Note**
>
> Ketika Anda memilih item dari bagian ini, Anda mungkin dikenakan biaya penggunaan tambahan. Untuk informasi selengkapnya, lihat Pengukuran Penggunaan.

Item berikut tersedia:

- Notebook

  Meluncurkan notebook dalam sesi kernel pada SageMaker gambar yang dipilih.

  Membuat buku catatan di folder yang saat ini Anda pilih di browser file. Untuk melihat browser file, di bilah sisi kiri Studio Classic, pilih ikon File Browser.

- Konsol

  Meluncurkan shell dalam sesi kernel pada SageMaker gambar yang dipilih.

  Membuka shell di folder yang saat ini Anda pilih di browser file.

- Terminal gambar

  Meluncurkan terminal dalam sesi terminal pada SageMaker gambar yang dipilih.

  Membuka terminal di folder root untuk pengguna (seperti yang ditunjukkan oleh folder Home di browser file).

> **Note**
>
> Secara default, instance CPU diluncurkan pada sebuah `ml.t3.medium` instance, sementara instance GPU diluncurkan pada sebuah instance. `ml.g4dn.xlarge`

Utilitas dan file

Di bagian ini, Anda dapat menambahkan bantuan kontekstual dalam buku catatan; membuat Python, Markdown dan file teks; dan membuka terminal sistem.

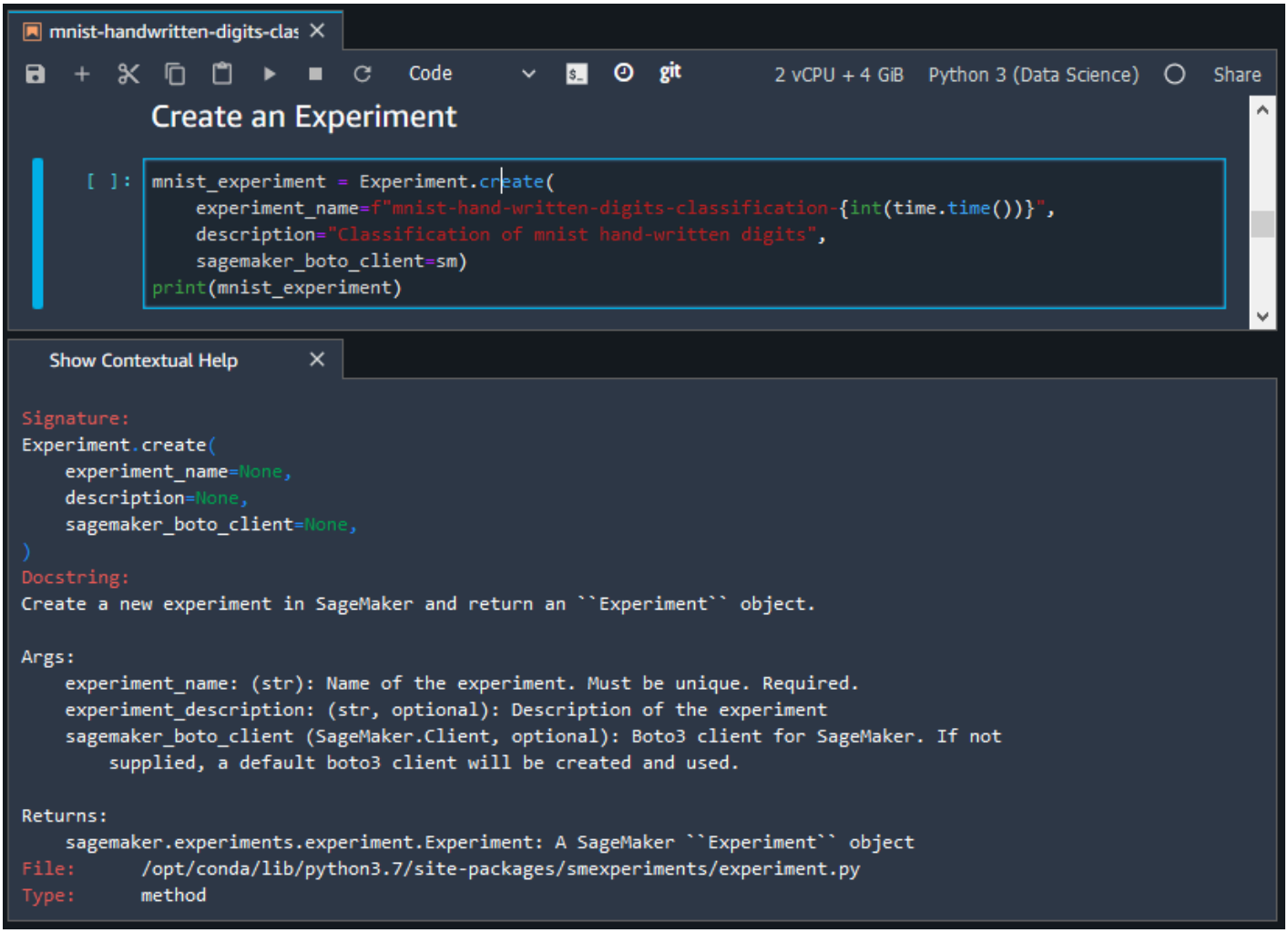
> **Note**
>
> Item di bagian ini berjalan dalam konteks Amazon SageMaker Studio Classic dan tidak dikenakan biaya penggunaan.

Item berikut tersedia:

- Tampilkan Bantuan Kontekstual

  Membuka tab baru yang menampilkan bantuan kontekstual untuk fungsi di notebook Studio Classic. Untuk menampilkan bantuan, pilih fungsi di notebook aktif. Untuk mempermudah melihat bantuan dalam konteks, seret tab bantuan sehingga berdekatan dengan tab notebook. Untuk membuka tab bantuan dari dalam buku catatan, tekan`Ctrl + I`.

  Tangkapan layar berikut menunjukkan bantuan kontekstual untuk metode ini`Experiment.create`.



- Terminal sistem

  Membuka bash shell di folder root untuk pengguna (seperti yang ditunjukkan oleh folder Home di browser file).

- File Teks dan File Markdown

  Membuat file dari jenis terkait di folder yang saat ini Anda pilih di browser file. Untuk melihat browser file, di sidebar kiri, pilih ikon File Browser ( ).

Berkolaborasi dengan ruang bersama

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Ruang bersama Amazon SageMaker Studio Classic terdiri dari JupyterServer aplikasi bersama dan direktori bersama. Semua profil pengguna dalam Domain memiliki akses ke semua ruang bersama di Domain. Amazon SageMaker secara otomatis mencakup sumber daya dalam ruang bersama dalam konteks aplikasi Amazon SageMaker Studio Classic yang Anda luncurkan di ruang bersama itu. Sumber daya dalam ruang bersama termasuk buku catatan, file, eksperimen, dan model.

Ruang bersama Studio Classic hanya mendukung Studio Classic dan KernelGateway aplikasi. Ruang bersama hanya mendukung penggunaan JupyterLab 3 image Amazon Resource Name (ARN). Untuk informasi selengkapnya, lihat JupyterLab Pembuatan Versi.

Amazon SageMaker secara otomatis menandai semua SageMaker sumber daya yang Anda buat dalam lingkup ruang bersama. Anda dapat menggunakan tag ini untuk memantau biaya dan merencanakan anggaran menggunakan alat, sepertiAWS Budgets.

Ruang bersama menggunakan pengaturan VPC yang sama dengan Domain yang dibuatnya.

> **Note**
>
> Ruang bersama tidak mendukung penggunaan klaster lintas akun Amazon SageMaker Data Wrangler atau Amazon EMR.

Penandaan otomatis

Semua sumber daya yang dibuat di ruang bersama secara otomatis ditandai dengan tag ARN Domain dan tag ARN ruang bersama. Tag ARN Domain didasarkan pada ID Domain, sedangkan tag ARN ruang bersama didasarkan pada nama spasi bersama.

Anda dapat menggunakan tag ini untuk memantau AWS CloudTrail penggunaan. Untuk informasi selengkapnya, lihat Log Panggilan SageMaker API Amazon dengan AWS CloudTrail.

Anda juga dapat menggunakan tag ini untuk memantau biayaAWS Billing and Cost Management. Untuk informasi selengkapnya, lihat Menggunakan tag alokasi AWS biaya.

Pengeditan bersama notebook secara real time

Manfaat utama dari ruang bersama adalah memfasilitasi kolaborasi antara anggota ruang bersama secara real time. Pengguna yang berkolaborasi di ruang kerja mendapatkan akses ke aplikasi Studio Classic bersama tempat mereka dapat mengakses, membaca, dan mengedit buku catatan mereka secara real time. Kolaborasi waktu nyata hanya didukung untuk JupyterServer aplikasi dalam ruang bersama.

Pengguna dengan akses ke ruang bersama dapat secara bersamaan membuka, melihat, mengedit, dan mengeksekusi notebook Jupyter di aplikasi Studio Classic bersama di ruang itu.

Notebook menunjukkan setiap pengguna co-editing dengan kursor berbeda yang menunjukkan nama profil pengguna. Meskipun beberapa pengguna dapat melihat buku catatan yang sama, pengeditan bersama paling cocok untuk kelompok kecil yang terdiri dari dua hingga lima pengguna.

Untuk melacak perubahan yang dilakukan oleh beberapa pengguna, kami sangat menyarankan untuk menggunakan kontrol versi berbasis Git bawaan Studio Classic.

JupyterServer 2

Untuk menggunakan spasi bersama, Jupyter Server versi 2 diperlukan. JupyterLab Ekstensi dan paket tertentu dapat secara paksa menurunkan versi Jupyter Server ke versi 1. Ini mencegah penggunaan ruang bersama. Jalankan yang berikut ini dari prompt perintah untuk mengubah nomor versi dan terus menggunakan spasi bersama.

```
conda activate studio
pip install jupyter-server==2.0.0rc3
```

Sesuaikan ruang bersama

Untuk melampirkan konfigurasi siklus hidup atau gambar kustom ke ruang bersama, Anda harus menggunakan. AWS CLI Untuk informasi selengkapnya tentang membuat dan melampirkan konfigurasi siklus hidup, lihat. Membuat dan mengaitkan konfigurasi siklus hidup Untuk informasi selengkapnya tentang membuat dan melampirkan gambar kustom, lihatBawa SageMaker gambar Anda sendiri.

Buat ruang bersama

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara membuat ruang bersama Amazon SageMaker Studio Classic di SageMaker Domain Amazon yang ada. Jika Anda membuat Domain tanpa dukungan untuk spasi bersama, Anda harus menambahkan dukungan untuk spasi bersama ke Domain yang ada sebelum Anda dapat membuat ruang bersama.

Topik

Menambahkan dukungan ruang bersama ke Domain yang ada

Anda dapat menggunakan SageMaker konsol atau menambahkan dukungan AWS CLI untuk spasi bersama ke Domain yang ada.

Konsol

Selesaikan prosedur berikut untuk menambahkan dukungan untuk spasi bersama ke Domain yang ada dari SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda buka halaman Pengaturan domain.
5. Pada halaman Detail domain, pilih tab Pengaturan domain.
6. Pilih Edit.
7. Untuk peran eksekusi default Space, tetapkan peran IAM yang digunakan secara default untuk semua spasi bersama yang dibuat di Domain.

8. Pilih Selanjutnya.
9. Pilih Selanjutnya.
10. Pilih Selanjutnya.
11. Pilih Kirim.

AWS CLI

Jalankan perintah berikut dari terminal mesin lokal Anda untuk menambahkan pengaturan ruang bersama default ke Domain dariAWS CLI. Jika Anda menambahkan pengaturan ruang bersama default ke Domain dalam VPC Amazon, Anda juga harus menyertakan daftar grup keamanan. spasi bersama hanya mendukung penggunaan JupyterLab 3 ARN gambar. Untuk informasi selengkapnya, lihat JupyterLab Pembuatan Versi.

```
# Public Internet domain
aws --region region \
sagemaker update-domain \
--domain-id domain-id \
--default-space-settings "ExecutionRole=execution-role-
arn,JupyterServerAppSettings={DefaultResourceSpec={InstanceType=system,SageMakerImageArn=sagema
image-arn}}"

# VPCOnly domain
aws --region region \
sagemaker update-domain \
--domain-id domain-id \
--default-space-settings "ExecutionRole=execution-role-
arn,JupyterServerAppSettings={DefaultResourceSpec={InstanceType=system,SageMakerImageArn=sagema
image-arn}},SecurityGroups=[security-groups]"
```

Verifikasi bahwa pengaturan ruang bersama default telah diperbarui.

```
aws --region region \
sagemaker describe-domain \
--domain-id domain-id
```

Buat ruang bersama

Bagian berikut menunjukkan cara membuat ruang bersama dari SageMaker konsol Amazon, Amazon SageMaker Studio, atauAWS CLI.

Buat dari Studio

Selesaikan prosedur berikut untuk membuat ruang bersama di Domain dari Studio.

1. Arahkan ke Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Dari UI Studio, cari panel aplikasi di sisi kiri.
3. Dari panel aplikasi, pilih Studio Classic.
4. Pilih Buat ruang Studio Classic
5. Di jendela pop up, masukkan nama untuk spasi.
6. Pilih Buat ruang.

Buat dari konsol

Selesaikan prosedur berikut untuk membuat ruang bersama di Domain dari SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda buat ruang bersama.
5. Pada halaman Detail domain, pilih tab Manajemen ruang.
6. Pilih Buat.
7. Masukkan nama untuk ruang bersama Anda. nama spasi bersama dalam Domain harus unik. Peran eksekusi untuk ruang bersama diatur ke peran eksekusi IAM Domain.

Buat dari AWS CLI

Bagian ini menunjukkan cara membuat ruang bersama dariAWS CLI.

Anda tidak dapat mengatur peran eksekusi ruang bersama saat membuat atau memperbaruinya. Hanya `DefaultDomainExecRole` dapat diatur saat membuat atau memperbarui Domain. spasi bersama hanya mendukung penggunaan JupyterLab 3 ARN gambar. Untuk informasi selengkapnya, lihat JupyterLab Pembuatan Versi.

Untuk membuat ruang bersama dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda.

```
aws --region region \
sagemaker create-space \
--domain-id domain-id \
--space-name space-name \
--space-settings '{
  "JupyterServerAppSettings": {
    "DefaultResourceSpec": {
      "SageMakerImageArn": "sagemaker-image-arn",
      "InstanceType": "system"
    }
  }
}'
```

Daftar dan Jelaskan ruang bersama

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Panduan ini menunjukkan cara mengakses daftar ruang bersama Amazon SageMaker Studio Classic di SageMaker Domain Amazon dengan SageMaker konsol Amazon, Amazon SageMaker Studio, atauAWS CLI. Ini juga menunjukkan cara melihat detail ruang bersama dariAWS CLI.

Topik

- Daftar spasi bersama
- Lihat detail ruang bersama

Daftar spasi bersama

Topik berikut menjelaskan cara melihat daftar spasi bersama dalam Domain dari SageMaker konsol atauAWS CLI.

Daftar spasi bersama dari Studio

Selesaikan prosedur berikut untuk melihat daftar spasi bersama di Domain dari Studio.

1. Arahkan ke Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Dari UI Studio, cari panel aplikasi di sisi kiri.
3. Dari panel aplikasi, pilih Studio Classic. Halaman ini mencantumkan semua ruang Studio Classic di domain yang dapat Anda akses.

Buat daftar spasi bersama dari konsol

Selesaikan prosedur berikut untuk melihat daftar spasi bersama di Domain dari SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda lihat daftar spasi bersama.
5. Pada halaman Detail domain, pilih tab Manajemen ruang.

Buat daftar spasi bersama dari AWS CLI

Untuk membuat daftar spasi bersama dalam Domain dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda.

```
aws --region region \
sagemaker list-spaces \
--domain-id domain-id
```

Lihat detail ruang bersama

Bagian berikut menjelaskan cara melihat detail ruang bersama dari SageMaker konsol, Studio, atauAWS CLI.

Lihat detail ruang bersama dari Studio

Selesaikan prosedur berikut untuk melihat detail spasi bersama di Domain dari Studio.

1. Arahkan ke Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Dari UI Studio, cari panel aplikasi di sisi kiri.

3. Dari panel aplikasi, pilih Studio Classic. Halaman ini mencantumkan semua ruang Studio Classic di domain yang dapat Anda akses.
4. Pilih nama spasi yang ingin Anda lihat detailnya.

Melihat detail ruang bersama dari konsol

Anda dapat melihat detail ruang bersama dari SageMaker konsol menggunakan prosedur berikut.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda lihat daftar spasi bersama.
5. Pada halaman Detail domain, pilih tab Manajemen ruang.
6. Pilih nama ruang untuk membuka halaman baru yang mencantumkan detail tentang ruang bersama.

Lihat detail ruang bersama dari AWS CLI

Untuk melihat detail ruang bersama dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda.

```
aws --region region \
sagemaker describe-space \
--domain-id domain-id \
--space-name space-name
```

Mengedit ruang bersama

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda hanya dapat mengedit detail untuk ruang bersama Amazon SageMaker Studio Classic menggunakan fileAWS CLI. Ini saat ini tidak didukung dari SageMaker konsol Amazon. Anda hanya dapat memperbarui atribut ruang kerja ketika tidak ada aplikasi yang berjalan di ruang bersama.

Untuk mengedit detail ruang bersama dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda. spasi bersama hanya mendukung penggunaan JupyterLab 3 ARN gambar. Untuk informasi selengkapnya, lihat JupyterLab Pembuatan Versi.

```
aws --region region \
sagemaker update-space \
--domain-id domain-id \
--space-name space-name \
--query SpaceArn --output text \
--space-settings '{
  "JupyterServerAppSettings": {
    "DefaultResourceSpec": {
      "SageMakerImageArn": "sagemaker-image-arn",
      "InstanceType": "system"
    }
  }
}'
```

Hapus ruang bersama

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara menghapus ruang bersama Amazon SageMaker Studio Classic dari SageMaker konsol Amazon atauAWS CLI. Ruang bersama hanya dapat dihapus jika tidak memiliki aplikasi yang berjalan.

Topik

- Konsol
- AWS CLI

Konsol

Selesaikan prosedur berikut untuk menghapus ruang bersama di SageMaker Domain Amazon dari SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda buat ruang bersama.
5. Pada halaman Detail domain, pilih tab Manajemen ruang.
6. Pilih ruang bersama yang ingin Anda hapus. Ruang bersama tidak boleh berisi aplikasi yang tidak gagal.
7. Pilih Hapus. Ini membuka jendela baru.
8. Pilih Ya, hapus spasi.
9. Masukkan hapus di bidang.
10. Pilih Hapus spasi.

AWS CLI

Untuk menghapus ruang bersama dariAWS CLI, jalankan perintah berikut dari terminal mesin lokal Anda.

```
aws --region region \
sagemaker delete-space \
--domain-id domain-id \
--space-name space-name
```

Gunakan Notebook Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Notebook Amazon SageMaker Studio Classic adalah notebook kolaboratif yang dapat Anda luncurkan dengan cepat karena Anda tidak perlu menyiapkan instans komputasi dan penyimpanan file terlebih dahulu. Satu set tipe instance, yang dikenal sebagai tipe peluncuran cepat dirancang untuk diluncurkan dalam waktu kurang dari dua menit. Notebook Studio Classic menyediakan penyimpanan persisten, yang memungkinkan Anda untuk melihat dan berbagi notebook meskipun instance yang dijalankan notebook dimatikan.

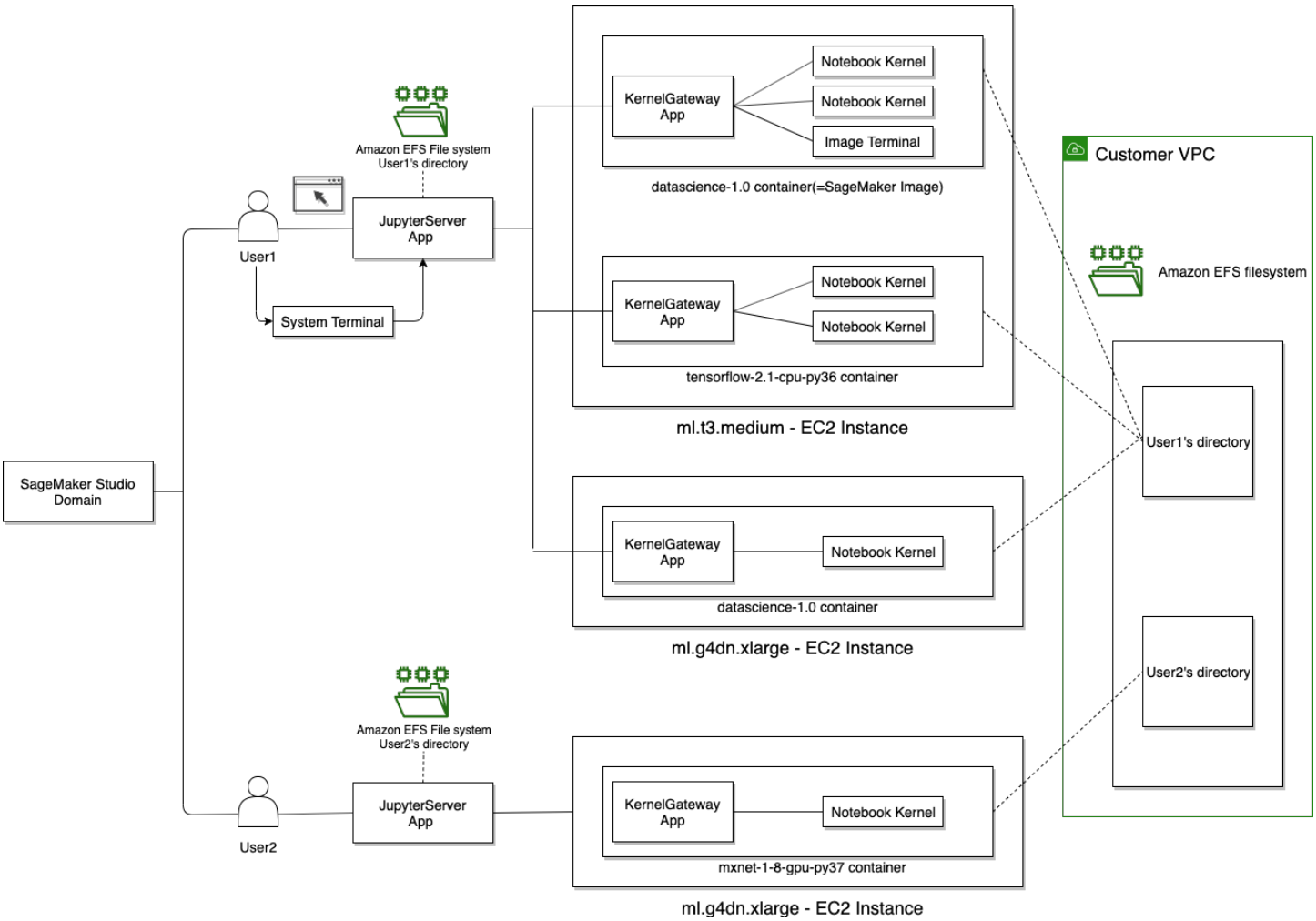
Anda dapat membagikan buku catatan Anda dengan orang lain, sehingga mereka dapat dengan mudah mereproduksi hasil Anda dan berkolaborasi sambil membuat model dan menjelajahi data Anda. Anda memberikan akses ke salinan buku catatan hanya-baca melalui URL aman. Dependensi untuk notebook Anda disertakan dalam metadata notebook. Ketika kolega Anda menyalin buku catatan, itu terbuka di lingkungan yang sama dengan notebook asli.

Notebook Studio Classic berjalan di lingkungan yang ditentukan oleh berikut ini:

- Jenis instans Amazon EC2 — Konfigurasi perangkat keras yang dijalankan notebook. Konfigurasi mencakup jumlah dan jenis prosesor (vCPU dan GPU), dan jumlah dan jenis memori. Jenis instans menentukan tingkat harga.
- SageMaker image - Gambar kontainer yang kompatibel dengan SageMaker Studio Classic. Gambar terdiri dari kernel, paket bahasa, dan file lain yang diperlukan untuk menjalankan notebook di Studio Classic. Mungkin ada beberapa gambar dalam sebuah contoh. Untuk informasi selengkapnya, lihat Bawa SageMaker gambar Anda sendiri.
- KernelGateway app — SageMaker Gambar berjalan sebagai KernelGateway aplikasi. Aplikasi ini menyediakan akses ke kernel dalam gambar. Ada one-to-one korespondensi antara SageMaker gambar dan KernelGateway aplikasi.
- Kernel — Proses yang memeriksa dan menjalankan kode yang terdapat dalam notebook. Kernel didefinisikan oleh spesifikasi kernel dalam gambar. Mungkin ada beberapa kernel dalam sebuah gambar.

Anda dapat mengubah salah satu sumber daya ini dari dalam buku catatan.

Diagram berikut menguraikan bagaimana kernel notebook berjalan dalam kaitannya dengan KernelGateway App, User, dan Domain.



Notebook Sample SageMaker Studio Classic tersedia di folder aws_sagemaker_studio dari repositori contoh Amazon. SageMaker GitHub Setiap notebook dilengkapi dengan SageMaker gambar yang diperlukan yang membuka notebook dengan kernel yang sesuai.

Sebaiknya Anda membiasakan diri dengan antarmuka SageMaker Studio Classic dan toolbar notebook Studio Classic sebelum membuat atau menggunakan notebook Studio Classic. Lihat informasi yang lebih lengkap di Ikhtisar UI Amazon SageMaker Studio Classic dan Menggunakan Toolbar Notebook Studio Classic.

Topik

- Bagaimana Notebook Amazon SageMaker Studio Classic Berbeda dengan Instans Notebook?
- Mulai
- Tur Klasik Amazon SageMaker Studio
- Membuat atau Membuka Notebook Amazon SageMaker Studio Classic

- Menggunakan Toolbar Notebook Studio Classic
- Instal Pustaka dan Kernel Eksternal di Amazon Studio Classic SageMaker
- Bagikan dan Gunakan Notebook Amazon SageMaker Studio Classic
- Dapatkan Notebook Studio Classic dan Metadata Aplikasi
- Dapatkan Perbedaan Notebook
- Kelola Sumber Daya
- Pengukuran Penggunaan
- Sumber Daya yang Tersedia

Bagaimana Notebook Amazon SageMaker Studio Classic Berbeda dengan Instans Notebook?

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Saat memulai notebook baru, sebaiknya buat notebook di Amazon SageMaker Studio Classic daripada meluncurkan instance notebook dari SageMaker konsol Amazon. Ada banyak manfaat menggunakan notebook Studio Classic, termasuk yang berikut ini:

- Lebih cepat: Memulai notebook Studio Classic lebih cepat daripada meluncurkan notebook berbasis instance. Biasanya, ini 5-10 kali lebih cepat dari notebook berbasis instance.
- Berbagi notebook yang mudah: Berbagi notebook adalah fitur terintegrasi di Studio Classic. Pengguna dapat membuat tautan yang dapat dibagikan yang mereproduksi kode notebook dan juga SageMaker gambar yang diperlukan untuk menjalankannya, hanya dalam beberapa klik.
- SDK Python Terbaru: Notebook Studio Classic sudah diinstal sebelumnya dengan SDK Amazon Python terbaru. SageMaker
- Akses semua fitur Studio Classic: Notebook Studio Classic diakses dari dalam Studio Classic. Ini memungkinkan Anda untuk membangun, melatih, men-debug, melacak, dan memantau model Anda tanpa meninggalkan Studio Classic.

- Direktori pengguna persisten: Setiap anggota tim Studio mendapatkan direktori home mereka sendiri untuk menyimpan notebook dan file lainnya. Direktori secara otomatis dipasang ke semua instance dan kernel saat dimulai, sehingga notebook dan file lainnya selalu tersedia. Direktori home disimpan di Amazon Elastic File System (Amazon EFS) sehingga Anda dapat mengaksesnya dari layanan lain.
- Akses langsung: Saat menggunakan IAM Identity Center, Anda menggunakan kredensyal IAM Identity Center melalui URL unik untuk mengakses Studio Classic secara langsung. Anda tidak perlu berinteraksi dengan AWS Management Console untuk menjalankan notebook Anda.
- Gambar yang dioptimalkan: Notebook Studio Classic dilengkapi dengan seperangkat pengaturan SageMaker gambar yang telah ditentukan untuk membantu Anda memulai lebih cepat.

> **Note**
>
> Notebook Studio Classic tidak mendukung mode lokal. Namun, Anda dapat menggunakan instance notebook untuk melatih sampel kumpulan data Anda secara lokal, lalu menggunakan kode yang sama di notebook Studio Classic untuk melatih kumpulan data lengkap.

Saat Anda membuka buku catatan di SageMaker Studio Classic, tampilan adalah perpanjangan dari JupyterLab antarmuka. Fitur utamanya sama, jadi Anda akan menemukan fitur khas notebook Jupyter dan. JupyterLab Untuk informasi selengkapnya tentang antarmuka Studio Classic, lihatIkhtisar UI Amazon SageMaker Studio Classic.

Mulai

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Untuk memulai, Anda atau administrator organisasi Anda harus menyelesaikan proses orientasi SageMaker domain. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.

Anda dapat mengakses notebook Studio Classic dengan salah satu cara berikut:

- Anda menerima undangan email untuk mengakses Studio Classic melalui Pusat Identitas IAM organisasi Anda, yang menyertakan tautan langsung untuk masuk ke Studio Classic tanpa harus menggunakan SageMaker konsol Amazon. Anda dapat melanjutkan kethe section called “Langkah Berikutnya”.
- Anda menerima tautan ke buku catatan Studio Classic bersama, yang menyertakan tautan langsung untuk masuk ke Studio Classic tanpa harus menggunakan SageMaker konsol. Anda dapat melanjutkan kethe section called “Langkah Berikutnya”.
- Anda onboard ke domain dan kemudian masuk ke SageMaker konsol. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.

Luncurkan Amazon SageMaker

Selesaikan langkah-langkah Luncurkan Amazon SageMaker Studio Classic untuk meluncurkan Studio Classic.

Langkah Berikutnya

Sekarang Anda berada di Studio Classic, Anda dapat mencoba salah satu opsi berikut:

- Untuk membuat notebook Studio Classic atau menjelajahi buku catatan end-to-end tutorial Studio Classic — Lihat Tur Klasik Amazon SageMaker Studio di bagian selanjutnya.
- Untuk membiasakan diri dengan antarmuka Studio Classic — Lihat Ikhtisar UI Amazon SageMaker Studio Classic atau coba buku catatan Memulai dengan memilih Buka buku catatan Memulai di bagian Tindakan cepat di halaman Beranda Studio Klasik.

Tur Klasik Amazon SageMaker Studio

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Untuk panduan yang membawa Anda dalam tur fitur utama Amazon SageMaker Studio Classic, lihat contoh notebook xgboost_customer_churn_studio.ipynb dari repositori aws/. amazon-sagemaker-

examples GitHub Kode di notebook melatih beberapa model dan mengatur SageMaker Debugger dan SageMaker Model Monitor. Panduan ini menunjukkan cara melihat uji coba, membandingkan model yang dihasilkan, menampilkan hasil debugger, dan menerapkan model terbaik menggunakan UI Studio Classic. Anda tidak perlu memahami kode untuk mengikuti panduan ini.

Prasyarat

Untuk menjalankan notebook untuk tur ini, Anda perlu:

- Akun IAM untuk masuk ke Studio. Untuk informasi, lihat Ikhtisar SageMaker Domain Amazon.
- Keakraban dasar dengan antarmuka pengguna Studio dan notebook Jupyter. Untuk informasi, lihat Ikhtisar UI Amazon SageMaker Studio Classic.
- Salinan aws/ amazon-sagemaker-examples repositori di lingkungan Studio Anda.

Untuk mengkloning repositori

1. Luncurkan Studio Classic mengikuti langkah-langkah di Luncurkan Amazon SageMaker Studio Classic Untuk pengguna di Pusat Identitas IAM, masuk menggunakan URL dari email undangan Anda.
2. Di menu atas, pilih File, lalu New, lalu Terminal.
3. Pada prompt perintah, jalankan perintah berikut untuk mengkloning amazon-sagemaker-examples GitHub aws/ repositori.

```
$ git clone https://github.com/aws/amazon-sagemaker-examples.git
```

Untuk menavigasi ke notebook sampel

1. Dari File Browser di menu sebelah kiri, pilih amazon-sagemaker-examples.
2. Arahkan ke notebook contoh dengan jalur berikut.

   ```
   ~/amazon-sagemaker-examples/aws_sagemaker_studio/getting_started/
   xgboost_customer_churn_studio.ipynb
   ```
3. Ikuti buku catatan untuk mempelajari fitur utama Studio Classic.

> **Note**
>
> Jika Anda mengalami kesalahan saat menjalankan contoh notebook, dan beberapa waktu telah berlalu sejak Anda mengkloning repositori, tinjau buku catatan di repositori jarak jauh untuk pembaruan.

Membuat atau Membuka Notebook Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Saat Anda Membuat Notebook dari Menu File berada di Amazon SageMaker Studio Classic atau Buka buku catatan di Studio Classic untuk pertama kalinya, Anda diminta untuk menyiapkan lingkungan dengan memilih SageMaker image, kernel, jenis instans, dan, secara opsional, skrip konfigurasi siklus hidup yang berjalan saat memulai gambar. SageMaker meluncurkan notebook pada instance dari jenis yang dipilih. Secara default, jenis instance diatur ke `ml.t3.medium` (tersedia sebagai bagian dari Tingkat AWS Gratis) untuk gambar berbasis CPU. Untuk gambar berbasis GPU, jenis instance default adalah. `ml.g4dn.xlarge`

Jika Anda membuat atau membuka notebook tambahan yang menggunakan jenis instans yang sama, terlepas dari apakah notebook menggunakan kernel yang sama atau tidak, notebook berjalan pada instance yang sama dari jenis instance tersebut.

Setelah meluncurkan notebook, Anda dapat mengubah jenis instans, SageMaker gambar, dan kernel dari dalam notebook. Lihat informasi yang lebih lengkap di Mengubah Tipe Instance dan Mengubah Gambar atau Kernel.

> **Note**
>
> Anda hanya dapat memiliki satu instance dari setiap jenis instance. Setiap instance dapat memiliki beberapa SageMaker gambar yang berjalan di atasnya. Setiap SageMaker gambar dapat menjalankan beberapa kernel atau instance terminal.

Penagihan terjadi per instance dan dimulai ketika instance pertama dari jenis instans tertentu diluncurkan. Jika Anda ingin membuat atau membuka buku catatan tanpa risiko menimbulkan biaya, buka buku catatan dari menu File dan pilih No Kernel dari dialog Select Kernel. Anda dapat membaca dan mengedit buku catatan tanpa kernel yang berjalan tetapi Anda tidak dapat menjalankan sel.

Penagihan berakhir ketika SageMaker gambar untuk instance dimatikan. Untuk informasi selengkapnya, lihat Pengukuran Penggunaan.

Untuk informasi tentang mematikan notebook, lihatMatikan Sumber Daya.

Topik

- Buka buku catatan di Studio Classic
- Membuat Notebook dari Menu File
- Buat Notebook dari Launcher
- Daftar jenis instance, gambar, dan kernel yang tersedia

Buka buku catatan di Studio Classic

Amazon SageMaker Studio Classic hanya dapat membuka notebook yang terdaftar di browser file Studio Classic. Untuk petunjuk tentang mengunggah buku catatan ke browser file, lihat Unggah File ke SageMaker Studio Classic ataumengkloning Repositori Git di Studio Classic SageMaker .

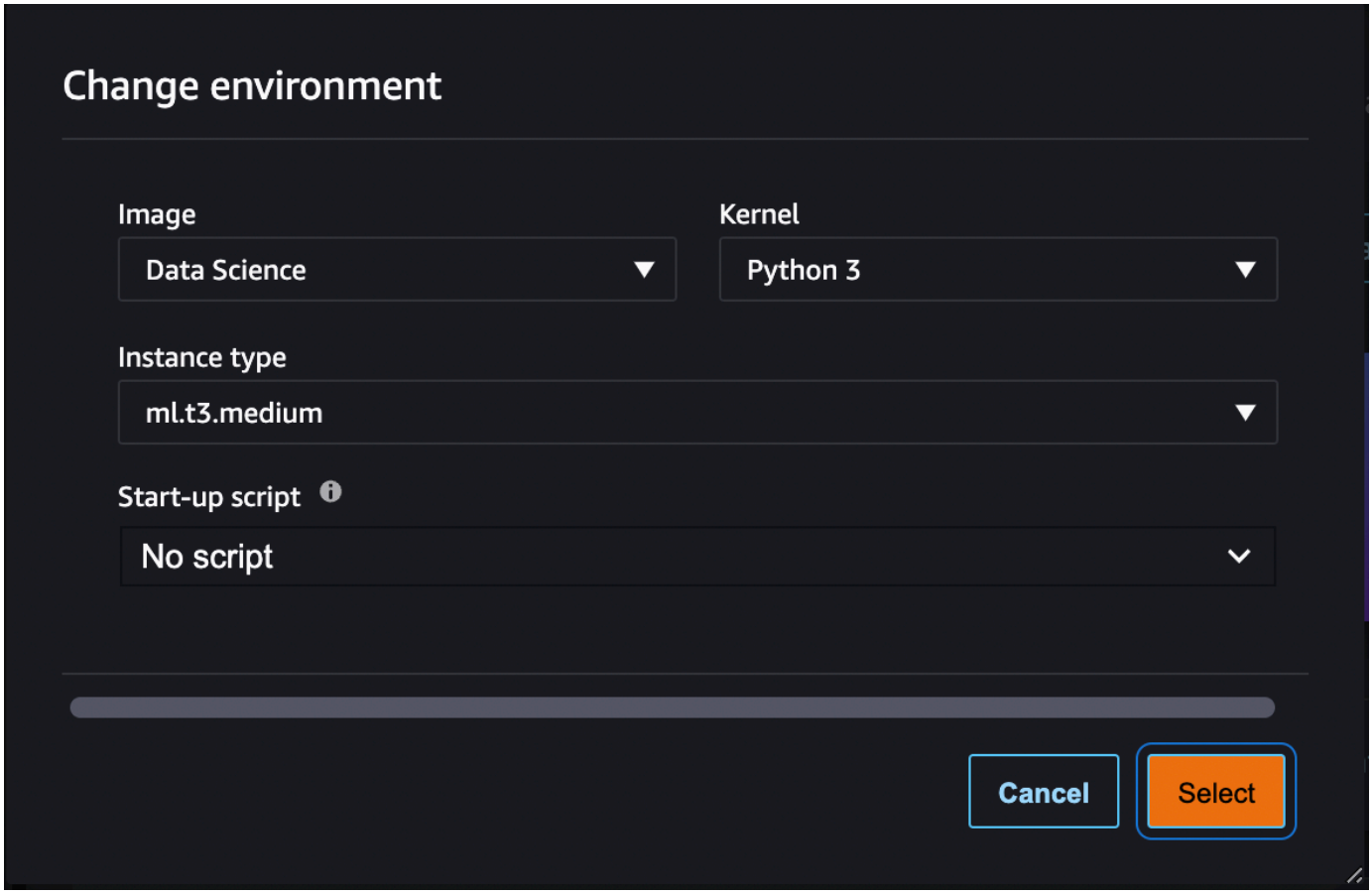
Untuk membuka buku catatan

1. Di bilah sisi kiri, pilih ikon File Browser
   ( )
   untuk menampilkan browser file.
2. Jelajahi file notebook dan klik dua kali untuk membuka buku catatan di tab baru.

Membuat Notebook dari Menu File

Untuk membuat buku catatan dari menu File

1. Dari menu Studio Classic, pilih File, pilih New, lalu pilih Notebook.
2. Dalam dialog Ubah lingkungan, gunakan menu tarik-turun untuk memilih skrip Image, Kernel, Instance type, dan Start-up, lalu pilih Select. Notebook Anda diluncurkan dan terbuka di tab Studio Classic baru.

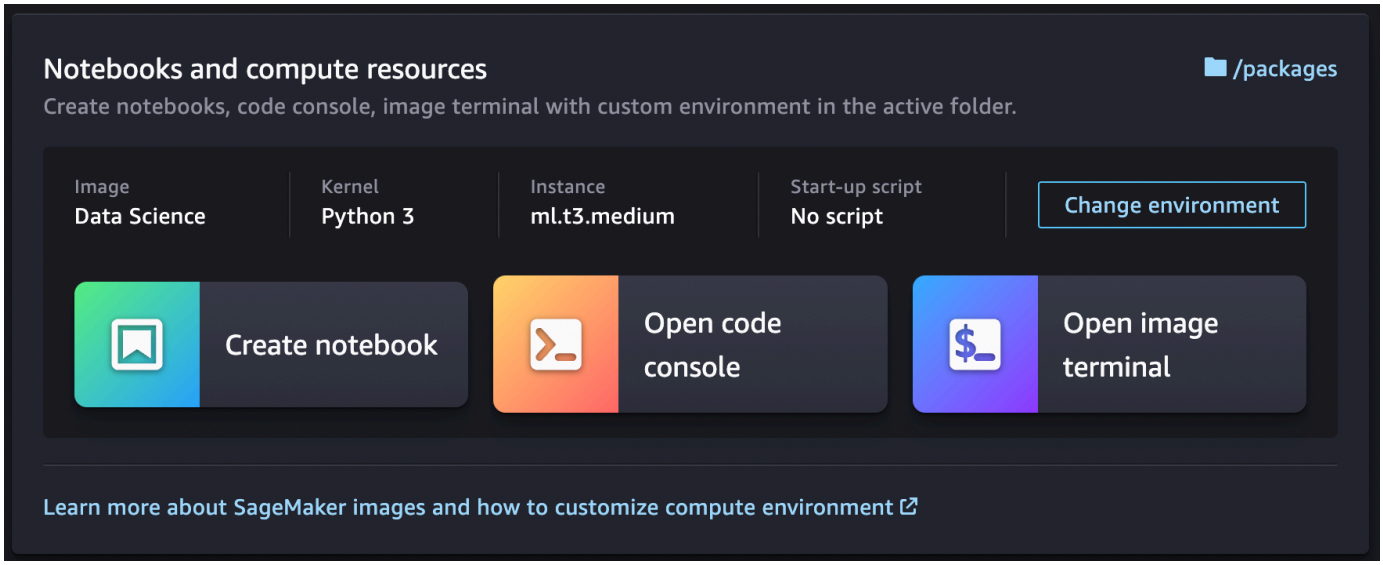


Buat Notebook dari Launcher

Untuk membuat notebook dari Launcher

1. Untuk membuka Launcher, pilih Amazon SageMaker Studio Classic di kiri atas antarmuka Studio Classic atau gunakan pintasan `Ctrl + Shift + L` keyboard.

   Untuk mempelajari semua cara yang tersedia untuk membuka Peluncur, lihat Gunakan Amazon SageMaker Studio Classic Launcher
2. Di Peluncur, di bagian Notebook dan sumber daya komputasi, pilih Ubah lingkungan.



3. Dalam dialog Ubah lingkungan, gunakan menu tarik-turun untuk memilih skrip Image, Kernel, Instance type, dan Start-up, lalu pilih Select.
4. Di Launcher, pilih Buat notebook. Notebook Anda diluncurkan dan terbuka di tab Studio Classic baru.

Untuk melihat sesi kernel notebook, di sidebar kiri, pilih ikon Running Terminals and Kernels
( ).
Anda dapat menghentikan sesi kernel notebook dari tampilan ini.

Daftar jenis instance, gambar, dan kernel yang tersedia

Untuk daftar semua sumber daya yang tersedia, lihat:

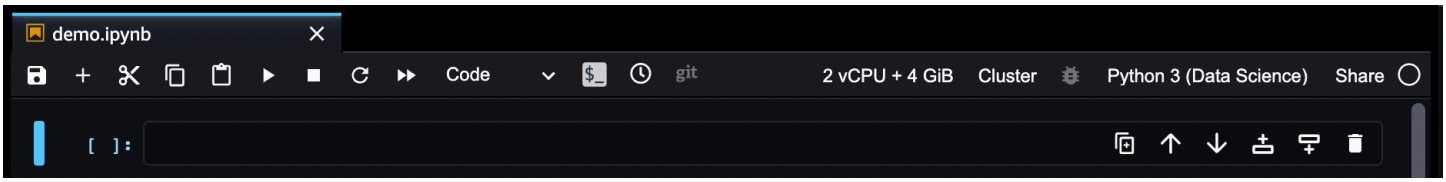
- Jenis Instans Studio Klasik yang Tersedia
- SageMaker Gambar Amazon yang Tersedia

Menggunakan Toolbar Notebook Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

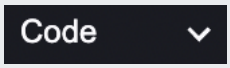
Notebook Amazon SageMaker Studio Classic memperluas JupyterLab antarmuka. Untuk ikhtisar JupyterLab antarmuka asli, lihat JupyterLabAntarmuka.

Gambar berikut menunjukkan toolbar dan sel kosong dari notebook Studio Classic.

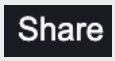


Saat Anda menjeda ikon toolbar, tooltip menampilkan fungsi ikon. Perintah notebook tambahan ditemukan di menu utama Studio Classic. Toolbar mencakup ikon-ikon berikut:

Ikon	Deskripsi
	Simpan dan pos pemeriksaan Menyimpan notebook dan memperbarui file pos pemeriksaan. Untuk informasi selengkapnya, lihat Dapatkan Perbedaan Antara Pos Pemeriksaan Terakhir.
	Masukkan sel Menyisipkan sel kode di bawah sel saat ini. Sel saat ini dicatat oleh penanda vertikal biru di margin kiri.
	Potong, salin, dan tempel sel Memotong, menyalin, dan menempelkan sel yang dipilih.
	Jalankan sel Menjalankan sel yang dipilih dan kemudian membuat sel yang mengikuti sel yang dipilih terakhir sel baru yang dipilih.
	Ganggu kernel Menginterupsi kernel, yang membatalkan operasi yang sedang berjalan. Kernel tetap aktif.
	Mulai ulang kernel

Ikon	Deskripsi
	Memulai ulang kernel. Variabel diatur ulang. Informasi yang belum disimpan tidak terpengaruh.
	Mulai ulang kernel dan jalankan semua sel Restart kernel, lalu jalankan semua sel notebook.
	Jenis sel Menampilkan atau mengubah jenis sel saat ini. Jenis selnya adalah: • Kode — Kode yang dijalankan kernel. • Markdown — Teks dirender sebagai penurunan harga. • Raw — Konten, termasuk markup Markdown, yang ditampilkan sebagai teks.
	Luncurkan terminal Meluncurkan terminal di SageMaker gambar yang menghosting notebook. Lihat contohnya di Dapatkan Metadata Aplikasi.
	Diff pos pemeriksaan Membuka tab baru yang menampilkan perbedaan antara notebook dan file pos pemeriksaan. Untuk informasi selengkapnya, lihat Dapatkan Perbedaan Antara Pos Pemeriksaan Terakhir.
	Git diff Hanya diaktifkan jika buku catatan dibuka dari repositori Git. Membuka tab baru yang menampilkan perbedaan antara buku catatan dan komit Git terakhir. Untuk informasi selengkapnya, lihat Dapatkan Perbedaan Antara Komit Terakhir.

Ikon	Deskripsi
2 vCPU+4 GiB	Jenis instans Menampilkan atau mengubah jenis instance yang dijalankan notebook. Formatnya adalah sebagai berikut: `number of vCPUs + amount of memory + number of GPUs` `Unknown`menunjukkan notebook dibuka tanpa menentukan kernel. Notebook berjalan pada instance SageMaker Studio dan tidak menambah biaya runtime. Anda tidak dapat menetapkan buku catatan ke jenis instans. Anda harus menentukan kernel dan kemudian Studio menetapkan notebook ke tipe default. Lihat informasi yang lebih lengkap di Membuat atau Membuka Notebook Amazon SageMaker Studio Classic dan Mengubah Tipe Instance.
Cluster	Kluster Hubungkan notebook Anda ke klaster EMR Amazon dan skala pekerjaan ETL Anda atau jalankan pelatihan model skala besar menggunakan Apache Spark, Hive, atau Presto. Untuk informasi selengkapnya, lihat Siapkan data menggunakan Amazon EMR.
Python 3 (Ilmu Data)	Kernel dan SageMaker Gambar Menampilkan atau mengubah kernel yang memproses sel di notebook. Formatnya adalah sebagai berikut: `Kernel (SageMaker Image)` `No Kernel`menunjukkan notebook dibuka tanpa menentukan kernel. Anda dapat mengedit buku catatan tetapi Anda tidak dapat menjalankan sel apa pun. Untuk informasi selengkapnya, lihat Mengubah Gambar atau Kernel.

Ikon	Deskripsi
	Status sibuk kernel Menampilkan status sibuk kernel. Ketika tepi lingkaran dan interiornya berwarna sama, kernel sibuk. Kernel sibuk ketika dimulai dan ketika sedang memproses sel. Status kernel tambahan ditampilkan di bilah status di sudut kiri bawah Studio. SageMaker
	Bagikan buku catatan Berbagi notebook. Untuk informasi selengkapnya, lihat Bagikan dan Gunakan Notebook Amazon SageMaker Studio Classic.

Untuk memilih beberapa sel, klik di margin kiri di luar sel. Tahan `Shift` tombol dan gunakan `K` atau `Up` tombol untuk memilih sel sebelumnya, atau gunakan `J` atau `Down` tombol untuk memilih sel berikut.

Instal Pustaka dan Kernel Eksternal di Amazon Studio Classic SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Notebook Amazon SageMaker Studio Classic hadir dengan beberapa gambar yang sudah diinstal. Gambar-gambar ini berisi kernel dan paket Python termasuk scikit-learn, NumPy Pandas,,,, dan MXNet. TensorFlow PyTorch Anda juga dapat menginstal gambar Anda sendiri yang berisi paket dan kernel pilihan Anda. Untuk informasi lebih lanjut tentang menginstal gambar Anda sendiri, lihatBawa SageMaker gambar Anda sendiri.

Kernel Jupyter yang berbeda di notebook Amazon SageMaker Studio Classic adalah lingkungan conda yang terpisah. Untuk informasi tentang lingkungan conda, lihat Mengelola lingkungan.

Alat instalasi Package

Metode yang Anda gunakan untuk menginstal paket Python dari terminal berbeda tergantung pada gambar. Studio Classic mendukung alat instalasi paket berikut:

- Notebook — Perintah berikut didukung. Jika salah satu dari berikut ini tidak berfungsi pada gambar Anda, coba yang lain.
  - `%conda install`
  - `%pip install`
- Terminal Jupyter — Anda dapat menginstal paket menggunakan pip dan conda secara langsung. Anda juga dapat menggunakan `apt-get install` untuk menginstal paket sistem dari terminal.

> **Note**
>
> Kami tidak menyarankan menggunakan `pip install -u` atau`pip install --user`, karena perintah tersebut menginstal paket pada volume Amazon EFS pengguna dan berpotensi memblokir restart JupyterServer aplikasi. Sebagai gantinya, gunakan konfigurasi siklus hidup untuk menginstal ulang paket yang diperlukan saat aplikasi dimulai ulang seperti yang ditunjukkan pada. Instal paket menggunakan konfigurasi siklus hidup

Kami merekomendasikan `%conda` untuk menggunakan `%pip` dan menginstal paket dari dalam buku catatan karena mereka dengan benar memperhitungkan lingkungan aktif atau juru bahasa yang digunakan. Untuk informasi selengkapnya, lihat Menambahkan %pip dan %conda magic functions. Anda juga dapat menggunakan sintaks perintah sistem (baris dimulai dengan!) untuk menginstal paket. Misalnya, `!pip install` dan `!conda install`.

Conda

Conda adalah sistem manajemen paket open source dan sistem manajemen lingkungan yang dapat menginstal paket dan dependensinya. SageMaker mendukung penggunaan conda dengan salah satu dari dua saluran utama ini: saluran default atau saluran conda-forge. Untuk informasi selengkapnya, lihat saluran Conda. Saluran conda-forge adalah saluran komunitas tempat kontributor dapat mengunggah paket.

> **Note**
>
> Menginstal paket dari conda-forge dapat memakan waktu hingga 10 menit. Pengaturan waktu berkaitan dengan bagaimana conda menyelesaikan grafik ketergantungan.

Semua lingkungan yang SageMaker disediakan fungsional. Paket yang diinstal pengguna mungkin tidak berfungsi dengan benar.

Conda memiliki dua metode untuk mengaktifkan lingkungan:`conda activate`, dan. `source activate` Untuk informasi selengkapnya, lihat Mengelola lingkungan.

Operasi conda yang didukung

- `conda install`dari sebuah paket dalam satu lingkungan
- `conda install`dari paket di semua lingkungan
- Menginstal paket dari repositori conda utama
- Menginstal paket dari conda-forge
- Mengubah lokasi pemasangan conda untuk menggunakan Amazon EBS
- Mendukung keduanya `conda activate` dan `source activate`

Pip

Pip adalah alat untuk menginstal dan mengelola paket Python. Pip mencari paket-paket pada Indeks Paket Python (PyPI) secara default. Tidak seperti conda, pip tidak memiliki dukungan lingkungan bawaan. Oleh karena itu, pip tidak selengkap conda dalam hal paket dengan dependensi asli atau pustaka sistem. Pip dapat digunakan untuk menginstal paket di lingkungan conda. Anda dapat menggunakan repositori paket alternatif dengan pip alih-alih PyPI.

Operasi pip yang didukung

- Menggunakan pip untuk menginstal paket tanpa lingkungan conda aktif
- Menggunakan pip untuk menginstal paket di lingkungan conda
- Menggunakan pip untuk menginstal paket di semua lingkungan conda
- Mengubah lokasi pemasangan pip untuk menggunakan Amazon EBS
- Menggunakan repositori alternatif untuk menginstal paket dengan pip

Tidak didukung

SageMaker bertujuan untuk mendukung sebanyak mungkin operasi instalasi paket. Namun, jika paket diinstal oleh SageMaker dan Anda menggunakan operasi berikut pada paket-paket ini, itu mungkin membuat lingkungan Anda tidak stabil:

- Menghapus instalasi
- Menurunkan
- Upgrading

Karena potensi masalah dengan kondisi jaringan atau konfigurasi, atau ketersediaan conda atau PyPi, paket mungkin tidak diinstal dalam jumlah waktu tetap atau deterministik.

> **Note**
>
> Mencoba menginstal paket di lingkungan dengan dependensi yang tidak kompatibel dapat mengakibatkan kegagalan. Jika terjadi masalah, Anda dapat menghubungi pengelola pustaka tentang memperbarui dependensi paket. Saat Anda memodifikasi lingkungan, seperti menghapus atau memperbarui paket yang ada, ini dapat mengakibatkan ketidakstabilan lingkungan tersebut.

Instal paket menggunakan konfigurasi siklus hidup

Instal gambar dan kernel khusus pada volume Amazon EBS instans Studio Classic sehingga tetap ada saat Anda menghentikan dan memulai ulang notebook, dan pustaka eksternal apa pun yang Anda instal tidak diperbarui oleh. SageMaker Untuk melakukannya, gunakan konfigurasi siklus hidup yang menyertakan skrip yang berjalan saat Anda membuat buku catatan (`on-create`)dan skrip yang berjalan setiap kali Anda memulai ulang notebook ()`on-start`. Untuk informasi selengkapnya tentang penggunaan konfigurasi siklus hidup dengan Studio Classic, lihat. Menggunakan konfigurasi siklus hidup dengan Amazon Studio Classic SageMaker Untuk contoh skrip konfigurasi siklus hidup, lihat Sampel Konfigurasi Siklus Hidup SageMaker Studio Classic.

Bagikan dan Gunakan Notebook Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

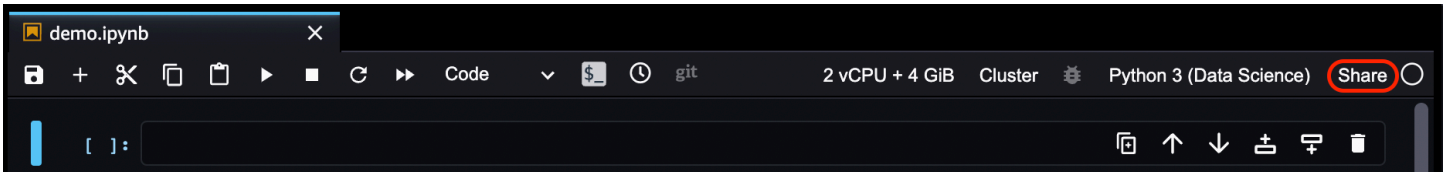
Anda dapat membagikan notebook Amazon SageMaker Studio Classic Anda dengan kolega Anda. Notebook bersama adalah salinan. Setelah membagikan buku catatan, perubahan apa pun yang Anda buat pada buku catatan asli tidak tercermin dalam buku catatan bersama dan perubahan apa pun yang dilakukan rekan kerja Anda dalam salinan buku catatan bersama mereka tidak tercermin di buku catatan asli Anda. Jika Anda ingin membagikan versi terbaru Anda, Anda harus membuat snapshot baru dan kemudian membagikannya.

Topik

- Bagikan Notebook
- Menggunakan Notebook Bersama
- Ruang bersama dan kolaborasi realtime

Bagikan Notebook

Tangkapan layar berikut menunjukkan menu dari notebook Studio Classic.



Untuk berbagi buku catatan

1. Di sudut kanan atas notebook, pilih Bagikan.
2. (Opsional) Di Buat snapshot yang dapat dibagikan, pilih salah satu item berikut:
   - Sertakan informasi repo Git - Termasuk tautan ke repositori Git yang berisi buku catatan. Ini memungkinkan Anda dan kolega Anda untuk berkolaborasi dan berkontribusi pada repositori Git yang sama.

- Sertakan output - Termasuk semua keluaran notebook yang telah disimpan.

> **Note**
>
> Jika Anda pengguna di Pusat Identitas IAM dan Anda tidak melihat opsi ini, administrator Pusat Identitas IAM Anda mungkin menonaktifkan fitur tersebut. Hubungi administrator Anda.

3. Pilih Buat.
4. Setelah snapshot dibuat, pilih Salin tautan lalu pilih Tutup.
5. Bagikan tautan dengan kolega Anda.

Setelah memilih opsi berbagi Anda, Anda diberikan URL. Anda dapat membagikan tautan ini dengan pengguna yang memiliki akses ke Amazon SageMaker Studio Classic. Saat pengguna membuka URL, mereka diminta untuk masuk menggunakan IAM Identity Center atau autentikasi IAM. Buku catatan bersama ini menjadi salinan, sehingga perubahan yang dilakukan oleh penerima tidak akan direproduksi di buku catatan asli Anda.

Menggunakan Notebook Bersama

Anda menggunakan buku catatan bersama dengan cara yang sama seperti yang Anda lakukan dengan buku catatan yang Anda buat sendiri. Anda harus terlebih dahulu masuk ke akun Anda, lalu buka tautan bersama. Jika Anda tidak memiliki sesi aktif, Anda menerima kesalahan.

Saat Anda memilih tautan ke buku catatan bersama untuk pertama kalinya, versi buku catatan hanya-baca akan terbuka. Untuk mengedit buku catatan bersama, pilih Buat Salinan. Ini menyalin buku catatan bersama ke penyimpanan pribadi Anda.

Notebook yang disalin diluncurkan pada instance dari jenis instans dan SageMaker gambar yang digunakan notebook saat pengirim membagikannya. Jika saat ini Anda tidak menjalankan instance dari jenis instance, instance baru dimulai. Kustomisasi pada SageMaker gambar tidak dibagikan. Anda juga dapat memeriksa snapshot notebook dengan memilih Detail Snapshot.

Berikut ini adalah beberapa pertimbangan penting tentang berbagi dan otentikasi:

- Jika Anda memiliki sesi aktif, Anda akan melihat tampilan buku catatan hanya-baca hingga Anda memilih Buat Salinan.
- Jika Anda tidak memiliki sesi aktif, Anda harus masuk.

- Jika Anda menggunakan IAM untuk login, setelah Anda login, pilih profil pengguna Anda lalu pilih Open Studio Classic. Maka Anda harus memilih tautan yang Anda kirim.
- Jika Anda menggunakan Pusat Identitas IAM untuk masuk, setelah Anda masuk buku catatan bersama dibuka secara otomatis di Studio.

Ruang bersama dan kolaborasi realtime

Ruang bersama terdiri dari JupyterServer aplikasi bersama dan direktori bersama. Manfaat utama dari ruang bersama adalah memfasilitasi kolaborasi antara anggota ruang bersama secara real time. Pengguna yang berkolaborasi di ruang kerja mendapatkan akses ke aplikasi Studio Classic bersama tempat mereka dapat mengakses, membaca, dan mengedit buku catatan mereka secara real time. Kolaborasi waktu nyata hanya didukung untuk JupyterServer aplikasi dalam ruang bersama. Pengguna dengan akses ke ruang bersama dapat secara bersamaan membuka, melihat, mengedit, dan mengeksekusi notebook Jupyter di aplikasi Studio Classic bersama di ruang itu. Untuk informasi selengkapnya tentang kolaborasi spasi bersama dan waktu nyata, lihatBerkolaborasi dengan ruang bersama.

Dapatkan Notebook Studio Classic dan Metadata Aplikasi

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat mengakses metadata notebook dan metadata Aplikasi menggunakan UI Amazon SageMaker Studio Classic.

Topik

- Dapatkan Metadata Notebook Studio Classic
- Dapatkan Metadata Aplikasi

Dapatkan Metadata Notebook Studio Classic

Notebook Jupyter berisi metadata opsional yang dapat Anda akses melalui Amazon Studio Classic UI. SageMaker

Untuk melihat metadata notebook:

1. Di sidebar kanan, pilih ikon Property Inspector
   ()  .
2. Buka bagian Advanced Tools.

Metadata akan terlihat mirip dengan yang berikut ini.

```
{
    "instance_type": "ml.t3.medium",
    "kernelspec": {
        "display_name": "Python 3 (Data Science)",
        "language": "python",
        "name": "python3__SAGEMAKER_INTERNAL__arn:aws:sagemaker:us-west-2:<acct-
id>:image/datascience-1.0"
    },
    "language_info": {
        "codemirror_mode": {
            "name": "ipython",
            "version": 3
        },
        "file_extension": ".py",
        "mimetype": "text/x-python",
        "name": "python",
        "nbconvert_exporter": "python",
        "pygments_lexer": "ipython3",
        "version": "3.7.10"
    }
}
```

Dapatkan Metadata Aplikasi

Saat Anda membuat buku catatan di Amazon SageMaker Studio Classic, metadata App ditulis ke file bernama `resource-metadata.json` di folder. `/opt/ml/metadata/` Anda bisa mendapatkan metadata App dengan membuka terminal Image dari dalam notebook. Metadata memberi Anda informasi berikut, yang mencakup SageMaker gambar dan jenis instance yang dijalankan notebook:

- AppType – `KernelGateway`
- DomainId— Sama seperti Studio ClassiCID
- UserProfileName— Nama profil pengguna saat ini

- ResourceArn— Nama Sumber Daya Amazon (ARN) Aplikasi, yang mencakup jenis instans
- ResourceName— Nama SageMaker gambar

Metadata tambahan mungkin disertakan untuk penggunaan internal oleh Studio Classic dan dapat berubah sewaktu-waktu.

Untuk mendapatkan metadata Aplikasi

1. Di tengah menu notebook, pilih ikon Launch Terminal
   ().
   Ini membuka terminal pada SageMaker gambar tempat notebook berjalan.
2. Jalankan perintah berikut untuk menampilkan isi `resource-metadata.json` file.

   ```
   $ cd /opt/ml/metadata/
   cat resource-metadata.json
   ```

   File akan terlihat mirip dengan yang berikut ini.

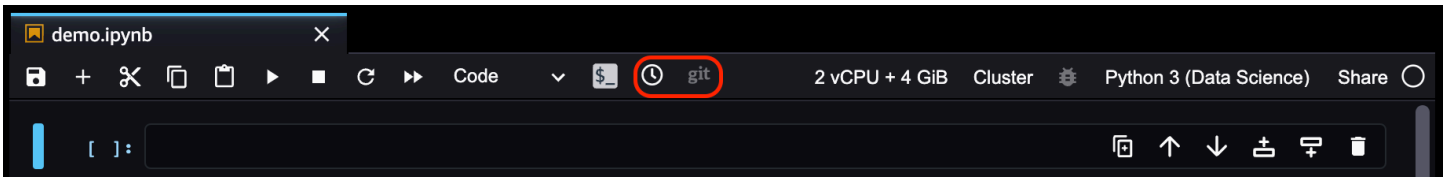
   ```
   {
       "AppType": "KernelGateway",
       "DomainId": "d-xxxxxxxxxxxx",
       "UserProfileName": "profile-name",
       "ResourceArn": "arn:aws:sagemaker:us-east-2:account-id:app/d-xxxxxxxxxxxx/
   profile-name/KernelGateway/datascience--1-0-ml-t3-medium",
       "ResourceName": "datascience--1-0-ml",
       "AppImageVersion":""
   }
   ```

Dapatkan Perbedaan Notebook

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat menampilkan perbedaan antara buku catatan saat ini dan pos pemeriksaan terakhir atau komit Git terakhir menggunakan SageMaker UI Amazon.

Tangkapan layar berikut menunjukkan menu dari notebook Studio Classic.



Topik

- Dapatkan Perbedaan Antara Pos Pemeriksaan Terakhir
- Dapatkan Perbedaan Antara Komit Terakhir

Dapatkan Perbedaan Antara Pos Pemeriksaan Terakhir

Saat Anda membuat buku catatan, file pos pemeriksaan tersembunyi yang cocok dengan buku catatan akan dibuat. Anda dapat melihat perubahan antara buku catatan dan file pos pemeriksaan atau mengembalikan buku catatan agar sesuai dengan file pos pemeriksaan.

Secara default, notebook disimpan secara otomatis setiap 120 detik dan juga saat Anda menutup notebook. Namun, file pos pemeriksaan tidak diperbarui agar sesuai dengan buku catatan. Untuk menyimpan buku catatan dan memperbarui file pos pemeriksaan agar cocok, Anda harus memilih ikon Simpan buku catatan dan buat pos pemeriksaan ( ) di sebelah kiri menu notebook atau gunakan pintasan `Ctrl + S` keyboard.

Untuk melihat perubahan antara buku catatan dan file pos pemeriksaan, pilih ikon Checkpoint diff ( ) di tengah menu notebook.

Untuk mengembalikan buku catatan ke file pos pemeriksaan, dari menu Studio Classic utama, pilih File lalu Kembalikan Notebook ke Checkpoint.

Dapatkan Perbedaan Antara Komit Terakhir

Jika buku catatan dibuka dari repositori Git, Anda dapat melihat perbedaan antara buku catatan dan komit Git terakhir.

Untuk melihat perubahan dalam buku catatan dari komit Git terakhir, pilih ikon diff Git
()
di tengah menu buku catatan.

Kelola Sumber Daya

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat mengubah jenis instans, serta SageMaker image dan kernel dari dalam notebook Amazon SageMaker Studio Classic. Untuk membuat kernel khusus untuk digunakan dengan notebook Anda, lihatBawa SageMaker gambar Anda sendiri.

Topik

- Mengubah Tipe Instance
- Mengubah Gambar atau Kernel
- Matikan Sumber Daya

Mengubah Tipe Instance

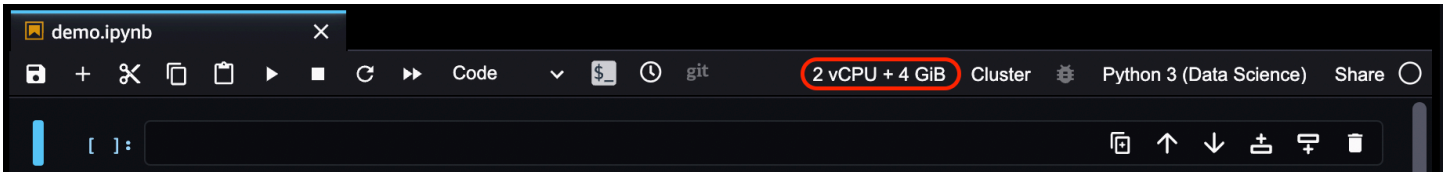
Saat membuka notebook Studio Classic baru untuk pertama kalinya, Anda akan diberi jenis instans Amazon Elastic Compute Cloud (Amazon EC2) default untuk menjalankan notebook. Saat Anda membuka notebook tambahan pada jenis instance yang sama, notebook berjalan pada instance yang sama dengan notebook pertama, meskipun notebook menggunakan kernel yang berbeda.

Anda dapat mengubah jenis instans yang dijalankan notebook Studio Classic dari dalam notebook.

Informasi berikut hanya berlaku untuk notebook Studio Classic. Untuk informasi tentang cara mengubah jenis instans instance SageMaker notebook Amazon, lihatMemperbarui Instance Notebook.

> **Important**
>
> Jika Anda mengubah jenis instans, informasi yang belum disimpan dan pengaturan yang ada untuk notebook hilang, dan paket yang diinstal harus diinstal ulang.
> Jenis instance sebelumnya terus berjalan meskipun tidak ada sesi kernel atau aplikasi yang aktif. Anda harus secara eksplisit menghentikan instance untuk berhenti menimbulkan biaya. Untuk menghentikan instance, lihatMatikan Sumber Daya.

Tangkapan layar berikut menunjukkan menu dari notebook Studio Classic. Prosesor dan memori tipe instance yang memberi daya pada notebook ditampilkan sebagai 2 vCpu+4 GiB.



Untuk mengubah jenis instance

1. Pilih prosesor dan memori dari jenis instance yang memberi daya pada notebook. Ini membuka jendela pop up.
2. Dari jendela pop up Siapkan lingkungan notebook, pilih menu tarik-turun tipe Instance.
3. Dari dropdown tipe Instance, pilih salah satu jenis instance yang terdaftar.
4. Setelah memilih jenis, pilih Pilih.
5. Tunggu instance baru diaktifkan, dan kemudian informasi tipe instance baru ditampilkan.

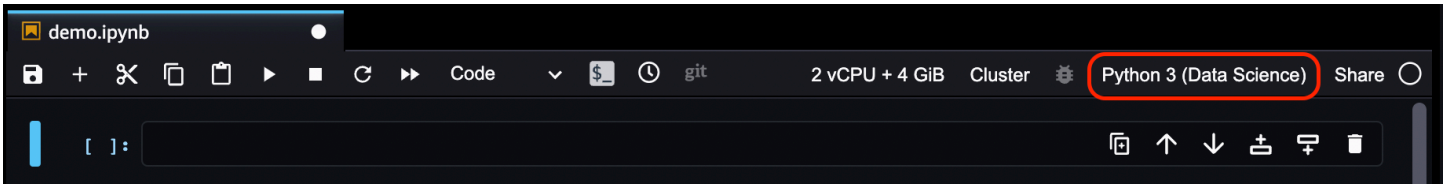
Untuk daftar jenis instance yang tersedia, lihatJenis Instans Studio Klasik yang Tersedia.

Mengubah Gambar atau Kernel

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Dengan notebook Amazon SageMaker Studio Classic, Anda dapat mengubah gambar atau kernel notebook dari dalam notebook.

Tangkapan layar berikut menunjukkan menu dari notebook Studio Classic. SageMakerKernel dan gambar saat ini ditampilkan sebagai Python 3 (Ilmu Data), di mana `Python 3` menunjukkan kernel dan `Data Science` menunjukkan SageMaker gambar yang berisi kernel. Warna lingkaran di sebelah kanan menunjukkan kernel sedang menganggur atau sibuk. Kernel sibuk ketika bagian tengah dan tepi lingkaran memiliki warna yang sama.



Untuk mengubah gambar atau kernel notebook

1. Pilih nama gambar/kernel di menu notebook.
2. Dari jendela pop up Siapkan lingkungan notebook, pilih menu dropdown Gambar atau Kernel.
3. Dari menu dropdown, pilih salah satu gambar atau kernel yang terdaftar.
4. Setelah memilih gambar atau kernel, pilih Pilih.
5. Tunggu status kernel ditampilkan sebagai idle, yang menunjukkan kernel telah dimulai.

Untuk daftar SageMaker gambar dan kernel yang tersedia, lihat SageMaker Gambar Amazon yang Tersedia.

Matikan Sumber Daya

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat mematikan sumber daya individual, termasuk notebook, terminal, kernel, aplikasi, dan instance. Anda juga dapat mematikan semua sumber daya di salah satu kategori ini secara bersamaan.

> **Note**
>
> Amazon SageMaker Studio Classic tidak mendukung mematikan sumber daya dari dalam notebook.

Topik

- Matikan Notebook Terbuka
- Matikan Sumber Daya

Matikan Notebook Terbuka

Anda dapat mematikan notebook yang terbuka dari menu File Amazon SageMaker Studio Classic atau dari panel Running Terminal dan Kernels.

> **Note**
>
> Saat Anda mematikan buku catatan, informasi apa pun yang belum disimpan di buku catatan akan hilang. Notebook tidak dihapus.

Untuk mematikan buku catatan yang terbuka dari menu File

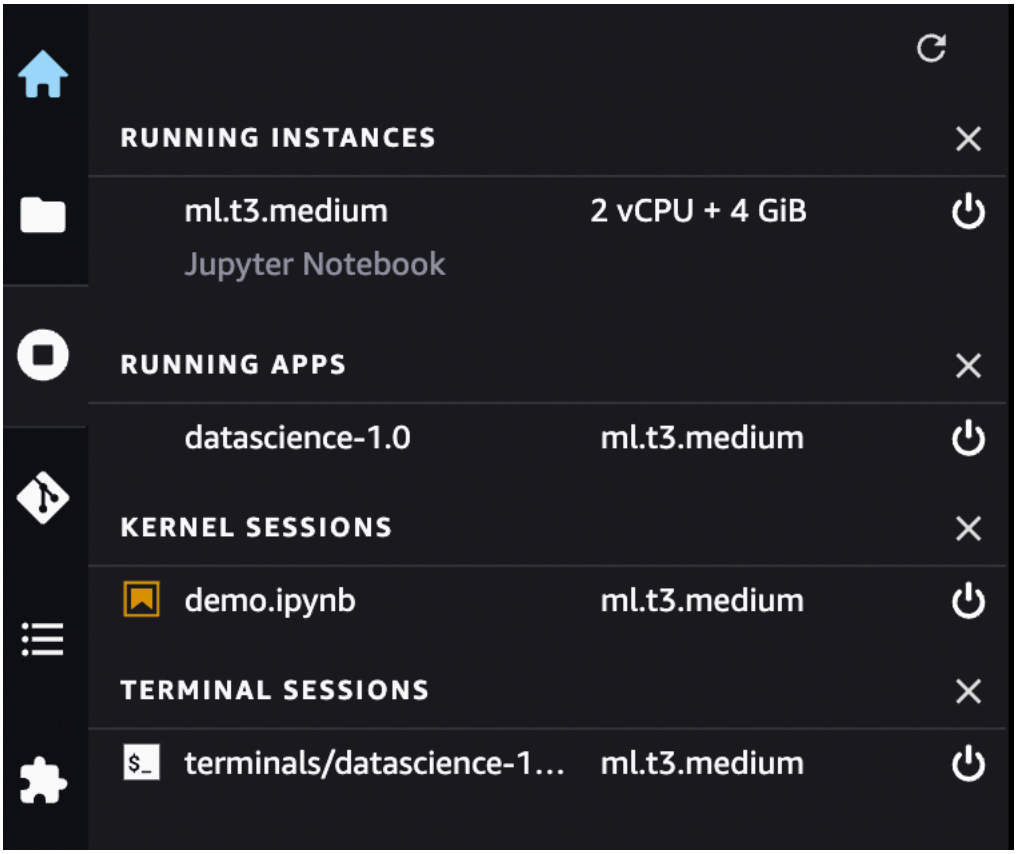
1. Secara opsional, simpan konten notebook dengan memilih ikon Disk di sebelah kiri menu notebook.
2. Pilih File lalu Tutup dan Matikan Notebook.
3. Pilih OK.

Matikan Sumber Daya

Anda dapat mencapai panel Running Terminal dan Kernels di sisi kiri Amazon SageMaker Studio Classic dengan

ikon 
.

Panel Running Terminal dan Kernels terdiri dari empat bagian. Setiap bagian mencantumkan semua sumber daya dari jenis itu. Anda dapat mematikan setiap sumber daya satu per satu atau mematikan semua sumber daya di bagian secara bersamaan.



Ketika Anda memilih untuk mematikan semua sumber daya di suatu bagian, hal berikut terjadi:

- MENJALANKAN INSTANCES/RUNNING APPS - Semua instance, aplikasi, notebook, sesi kernel, konsol/shell, dan terminal gambar dimatikan. Terminal sistem tidak dimatikan.

  > **Note**
  >
  > Saat Anda mematikan instans notebook Studio Classic, sumber daya tambahan apa pun, seperti titik SageMaker akhir, kluster EMR Amazon, dan bucket Amazon S3 yang dibuat dari Studio Classic tidak akan dihapus. Hapus sumber daya tersebut untuk menghentikan akrual biaya.

- KERNEL SESSIONS — Semua kernel, notebook dan konsol/shell dimatikan.
- SESI TERMINAL — Semua terminal gambar dan terminal sistem dimatikan.

Untuk mematikan sumber daya

1. Di bilah sisi kiri, pilih ikon Running Terminals dan Kernels
   ().
2. Lakukan salah satu dari langkah berikut:

   - Untuk mematikan sumber daya tertentu, pilih ikon Shut Down
     ()
     pada baris yang sama dengan sumber daya.

     Untuk menjalankan instance, dialog konfirmasi mencantumkan semua sumber daya yang akan dimatikan. Untuk menjalankan aplikasi, dialog konfirmasi ditampilkan. Pilih Matikan semua untuk melanjutkan.

     > **Note**
     >
     > Tidak ada dialog konfirmasi yang ditampilkan untuk sesi kernel atau sesi terminal.

   - Untuk mematikan semua sumber daya di bagian, pilih X di sebelah kanan label bagian. Dialog konfirmasi ditampilkan. Pilih Matikan semua untuk melanjutkan.

Pengukuran Penggunaan

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Tidak ada biaya tambahan untuk menggunakan Amazon SageMaker Studio Classic. Biaya yang dikeluarkan untuk menjalankan notebook Amazon SageMaker Studio Classic, shell interaktif, konsol, dan terminal didasarkan pada penggunaan instans Amazon Elastic Compute Cloud (Amazon EC2).

Ketika Anda menjalankan sumber daya berikut, Anda harus memilih SageMaker gambar dan kernel:

Dari Studio Classic Launcher

- Buku catatan
- Shell Interaktif
- Terminal Gambar

Dari menu File

- Buku catatan
- Konsol

Saat diluncurkan, sumber daya dijalankan pada instans Amazon EC2 dari jenis instans yang dipilih. Jika instance jenis itu sebelumnya diluncurkan dan tersedia, sumber daya dijalankan pada instance itu.

Untuk gambar berbasis CPU, jenis instans default yang disarankan adalah`ml.t3.medium`. Untuk gambar berbasis GPU, jenis instans default yang disarankan adalah`ml.g4dn.xlarge`.

Biaya yang dikeluarkan didasarkan pada jenis instans. Anda ditagih secara terpisah untuk setiap instance.

Pengukuran dimulai ketika sebuah instance dibuat. Pengukuran berakhir ketika semua aplikasi pada instance dimatikan, atau instance dimatikan. Untuk informasi tentang cara mematikan instance, lihatMatikan Sumber Daya.

> **Important**
>
> Anda harus mematikan instance untuk berhenti menimbulkan biaya. Jika Anda mematikan notebook yang berjalan pada instance tetapi tidak mematikan instance, Anda masih akan dikenakan biaya. Saat Anda mematikan instans notebook Studio Classic, sumber daya tambahan apa pun, seperti titik SageMaker akhir, kluster EMR Amazon, dan bucket Amazon S3 yang dibuat dari Studio Classic tidak akan dihapus. Hapus sumber daya tersebut untuk menghentikan akrual biaya.

Ketika Anda membuka beberapa notebook pada jenis instance yang sama, notebook berjalan pada instance yang sama meskipun mereka menggunakan kernel yang berbeda. Anda ditagih hanya untuk saat satu instance berjalan.

Anda dapat mengubah jenis instance dari dalam buku catatan setelah Anda membukanya. Untuk informasi selengkapnya, lihat Mengubah Tipe Instance.

Untuk informasi tentang penagihan beserta contoh harga, lihat SageMaker Harga Amazon.

Sumber Daya yang Tersedia

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Bagian berikut mencantumkan sumber daya yang tersedia untuk notebook Amazon SageMaker Studio Classic.

Topik

- Jenis Instans Studio Klasik yang Tersedia
- SageMaker Gambar Amazon yang Tersedia

Jenis Instans Studio Klasik yang Tersedia

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Notebook Amazon SageMaker Studio Classic berjalan di instans Amazon Elastic Compute Cloud (Amazon EC2). Jenis instans Amazon EC2 berikut tersedia untuk digunakan dengan notebook Studio Classic. Untuk informasi mendetail tentang jenis instans mana yang sesuai dengan kasus penggunaan Anda, dan kemampuan kinerjanya, lihat jenis Instans Cloud Amazon Elastic Compute. Untuk informasi tentang harga untuk jenis instans ini, lihat Harga Amazon EC2.

Untuk informasi tentang jenis Instans SageMaker Notebook Amazon yang tersedia, lihat CreateNotebookInstance.

> **Note**
>
> Untuk sebagian besar kasus penggunaan, Anda harus menggunakan`ml.t3.medium`. Ini adalah tipe instans default untuk SageMaker gambar berbasis CPU, dan tersedia sebagai bagian dari Tingkat AWSGratis.

Topik

- Instans CPU
- Instans dengan 1 atau lebih GPU

Instans CPU

Tabel berikut mencantumkan jenis instans CPU Amazon EC2 tanpa GPU terpasang yang tersedia untuk digunakan dengan notebook Studio Classic. Ini juga mencantumkan informasi tentang spesifikasi setiap jenis instance. Jenis instance default untuk gambar berbasis CPU adalah. `ml.t3.medium`

Untuk informasi mendetail tentang jenis instans mana yang sesuai dengan kasus penggunaan Anda, dan kemampuan kinerjanya, lihat jenis Instans Cloud Amazon Elastic Compute. Untuk informasi tentang harga untuk jenis instans ini, lihat Harga Amazon EC2.

> **Note**
>
> Jenis instans peluncuran cepat dioptimalkan untuk memulai dalam waktu kurang dari dua menit.

Instans CPU

Instans	Kasus penggunaan	Peluncuran cepat	vCPU	Memori (GiB)	Penyimpanan Instance (GB)
db.t3.medium	Tujuan umum	Ya	2	4	Hanya Amazon EBS
db.t3.large	Tujuan umum	Tidak	2	8	Hanya Amazon EBS
db.t3.xlarge	Tujuan umum	Tidak	4	16	Hanya Amazon EBS
ml.t3.2xlarge	Tujuan umum	Tidak	8	32	Hanya Amazon EBS
db.m5.large	Tujuan umum	Ya	2	8	Hanya Amazon EBS
db.m5.xlarge	Tujuan umum	Tidak	4	16	Hanya Amazon EBS
ml.m5.2xlarge	Tujuan umum	Tidak	8	32	Hanya Amazon EBS
ml.m5.4xlarge	Tujuan umum	Tidak	16	64	Hanya Amazon EBS

Instans	Kasus penggunaan	Peluncuran cepat	vCPU	Memori (GiB)	Penyimpanan Instance (GB)
ml.m5.8xlarge	Tujuan umum	Tidak	32	128	Hanya Amazon EBS
ml.m5.12xlarge	Tujuan umum	Tidak	48	192	Hanya Amazon EBS
ml.m5.16xlarge	Tujuan umum	Tidak	64	256	Hanya Amazon EBS
ml.m5.24xlarge	Tujuan umum	Tidak	96	384	Hanya Amazon EBS
ml.m5d.large	Tujuan umum	Tidak	2	8	1 x 75 NVMe SSD
ml.m5d.xlarge	Tujuan umum	Tidak	4	16	1 x 150 NVMe SSD
ml.m5d.2xlarge	Tujuan umum	Tidak	8	32	1 x 300 NVMe SSD
ml.m5d.4xlarge	Tujuan umum	Tidak	16	64	2 x 300 NVMe SSD

Instans	Kasus penggunaan	Peluncuran cepat	vCPU	Memori (GiB)	Penyimpanan Instance (GB)
ml.m5d.8xlarge	Tujuan umum	Tidak	32	128	2 x 600 NVMe SSD
ml.m5d.12xlarge	Tujuan umum	Tidak	48	192	2 x 900 NVMe SSD
ml.m5d.16xlarge	Tujuan umum	Tidak	64	256	4 x 600 NVMe SSD
ml.m5d.24xlarge	Tujuan umum	Tidak	96	384	4 x 900 NVMe SSD
ml.c5.large	Komputasi yang dioptimalkan	Ya	2	4	Hanya Amazon EBS
ml.c5.xlarge	Komputasi yang dioptimalkan	Tidak	4	8	Hanya Amazon EBS
ml.c5.2xlarge	Komputasi yang dioptimalkan	Tidak	8	16	Hanya Amazon EBS
ml.c5.4xlarge	Komputasi yang dioptimalkan	Tidak	16	32	Hanya Amazon EBS

Instans	Kasus penggunaan	Peluncuran cepat	vCPU	Memori (GiB)	Penyimpanan Instance (GB)
ml.c5.9xlarge	Komputasi yang dioptimalkan	Tidak	36	72	Hanya Amazon EBS
ml.c5.12xbesar	Komputasi yang dioptimalkan	Tidak	48	96	Hanya Amazon EBS
ml.c5.18xlarge	Komputasi yang dioptimalkan	Tidak	72	144	Hanya Amazon EBS
ml.c5.24xbesar	Komputasi yang dioptimalkan	Tidak	96	192	Hanya Amazon EBS
db.r5.large	Memori yang dioptimalkan	Tidak	2	16	Hanya Amazon EBS
db.r5.xlarge	Memori yang dioptimalkan	Tidak	4	32	Hanya Amazon EBS
ml.r5.2xlarge	Memori yang dioptimalkan	Tidak	8	64	Hanya Amazon EBS
ml.r5.4xlarge	Memori yang dioptimalkan	Tidak	16	128	Hanya Amazon EBS

Instans	Kasus penggunaan	Peluncuran cepat	vCPU	Memori (GiB)	Penyimpanan Instance (GB)
ml.r5.8xlarge	Memori yang dioptimalkan	Tidak	32	256	Hanya Amazon EBS
ml.r5.12xlarge	Memori yang dioptimalkan	Tidak	48	384	Hanya Amazon EBS
ml.r5.16xlarge	Memori yang dioptimalkan	Tidak	64	512	Hanya Amazon EBS
ml.r5.24xlarge	Memori yang dioptimalkan	Tidak	96	768	Hanya Amazon EBS

Instans dengan 1 atau lebih GPU

Tabel berikut mencantumkan jenis instans Amazon EC2 dengan 1 atau lebih GPU terpasang yang tersedia untuk digunakan dengan notebook Studio Classic. Ini juga mencantumkan informasi tentang spesifikasi setiap jenis instance. Jenis instance default untuk gambar berbasis GPU adalah. `ml.g4dn.xlarge`

Untuk informasi mendetail tentang jenis instans mana yang sesuai dengan kasus penggunaan Anda, dan kemampuan kinerjanya, lihat jenis Instans Cloud Amazon Elastic Compute. Untuk informasi tentang harga untuk jenis instans ini, lihat Harga Amazon EC2.

> **Note**
>
> Jenis instans peluncuran cepat dioptimalkan untuk memulai dalam waktu kurang dari dua menit.

Instans dengan 1 atau lebih GPU

Instans	Kasus penggunaan	Peluncuran cepat	GPU	vCPU	Memori (GiB)	Memori GPU (GiB)	Penyimpanan Instance (GB)
ml.p3.2xlarge	Komputasi yang dipercepat	Tidak	1	8	61	16	Hanya Amazon EBS
ml.p3.8xlarge	Komputasi yang dipercepat	Tidak	4	32	244	64	Hanya Amazon EBS
ml.p3.16xlarge	Komputasi yang dipercepat	Tidak	8	64	488	128	Hanya Amazon EBS
ml.p3dn.24xlarge	Komputasi yang dipercepat	Tidak	8	96	768	256	2 x 900 NVMe SSD
ml.p4d.24xlarge	Komputasi yang dipercepat	Tidak	8	96	1152	320 GB HBM2	8 x 1000 NVMe SSD
ml.p4de.24xlarge	Komputasi yang dipercepat	Tidak	8	96	1152	640 GB HBM2e	8 x 1000 NVMe SSD
ml.g4dn.xlarge	Komputasi yang dipercepat	Ya	1	4	16	16	1 x 125 NVMe SSD

Instans	Kasus penggunaan	Peluncuran cepat	GPU	vCPU	Memori (GiB)	Memori GPU (GiB)	Penyimpanan Instance (GB)
ml.g4dn.2xbesar	Komputasi yang dipercepat	Tidak	1	8	32	16	1 x 225 NVMe SSD
ml.g4dn.4xbesar	Komputasi yang dipercepat	Tidak	1	16	64	16	1 x 225 NVMe SSD
ml.g4dn.8xlarge	Komputasi yang dipercepat	Tidak	1	32	128	16	1 x 900 NVMe SSD
ml.g4dn.12xlarge	Komputasi yang dipercepat	Tidak	4	48	192	64	1 x 900 NVMe SSD
ml.g4dn.16xlarge	Komputasi yang dipercepat	Tidak	1	64	256	16	1 x 900 NVMe SSD
ml.g5.xlarge	Komputasi yang dipercepat	Tidak	1	4	16	24	1 x 250 NVMe SSD

Instans	Kasus penggunaan	Peluncuran cepat	GPU	vCPU	Memori (GiB)	Memori GPU (GiB)	Penyimpanan Instance (GB)
ml.g5.2xbesar	Komputasi yang dipercepat	Tidak	1	8	32	24	1 x 450 NVMe SSD
ml.g5.4xbesar	Komputasi yang dipercepat	Tidak	1	16	64	24	1 x 600 NVMe SSD
ml.g5.8xbesar	Komputasi yang dipercepat	Tidak	1	32	128	24	1 x 900 NVMe SSD
ml.g5.12xbesar	Komputasi yang dipercepat	Tidak	4	48	192	96	1 x 3800 NVMe SSD
ml.g5.16xbesar	Komputasi yang dipercepat	Tidak	1	64	256	24	1 x 1900 NVMe SSD
ml.g5.24xbesar	Komputasi yang dipercepat	Tidak	4	96	384	96	1 x 3800 NVMe SSD

Instans	Kasus penggunaan	Peluncuran cepat	GPU	vCPU	Memori (GiB)	Memori GPU (GiB)	Penyimpanan Instance (GB)
ml.g5.48x besar	Komputasi yang dipercepat	Tidak	8	192	768	192	2 x 3800 NVMe SSD

SageMaker Gambar Amazon yang Tersedia

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Halaman ini mencantumkan SageMaker gambar dan kernel terkait yang tersedia di Amazon SageMaker Studio Classic, serta format yang diperlukan untuk membuat ARN untuk setiap gambar. SageMaker gambar berisi SDK Amazon SageMaker Python terbaru dan versi terbaru dari kernel. Untuk informasi selengkapnya, lihat Gambar Deep Learning Containers.

Topik

- Format gambar ARN
- Tag URI yang didukung
- Gambar yang didukung
- Gambar dijadwalkan untuk penghentian

Format gambar ARN

Tabel berikut mencantumkan gambar ARN dan format URI untuk setiap Wilayah. Untuk membuat ARN lengkap untuk gambar, ganti placeholder pengenal *`sumber daya dengan pengenal`* sumber daya yang sesuai untuk gambar dari tabel gambar dan kernel. SageMaker Untuk membuat URI

lengkap untuk gambar, ganti placeholder *tag* dengan tag cpu atau gpu yang sesuai. Untuk daftar tag yang dapat Anda gunakan, lihatTag URI yang didukung.

> **Note**
>
> SageMaker Gambar distribusi menggunakan satu set ARN gambar yang berbeda.

Wilayah	Gambar ARN Format	SageMaker Format ARN Gambar Distribusi	SageMaker Format URI Gambar Distribusi
us-east-1	arn:aws:sagemaker:us-east-1:081325390199:image/*pengenal sumber daya*	arn:aws:sagemaker:us-east-1:885854791233:image/*pengenal sumber daya*	*885854791233.dkr.ecr.us-east-1.amazonaws.com/: menandai sagemaker-distribution-prod*
us-east-2	arn:aws:sagemaker:us-east-2:429704687514:image/*pengenal sumber daya*	arn:aws:sagemaker:us-east-2:137914896644:image/*pengenal sumber daya*	*137914896644.dkr.ecr.us-east-2.amazonaws.com/: menandai sagemaker-distribution-prod*
us-west-1	arn:aws:sagemaker:us-west-1:742091327244:image/*pengenal sumber daya*	arn:aws:sagemaker:us-west-1:053634841547:image/*pengenal sumber daya*	*053634841547.dkr.ecr.us-west-1.amazonaws.com*

			: menandai sagemaker-distribution-prod
us-west-2	arn:aws:sagemaker:us-west-2:236514542706:image/*pengenal sumber daya*	arn:aws:sagemaker:us-west-2:542918446943:image/*pengenal sumber daya*	*542918446943.dkr.ecr.us-west-2.amazonaws.com/: menandai sagemaker-distribution-prod*
af-south-1	arn:aws:sagemaker:af-south-1:559312083959:image/*pengenal sumber daya*	arn:aws:sagemaker:af-south-1:238384257742:image/*pengenal sumber daya*	*238384257742.dkr.ecr.af-south-1.amazonaws.com/: menandai sagemaker-distribution-prod*
ap-east-1	arn:aws:sagemaker:ap-east-1:493642496378:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-east-1:523751269255:image/*pengenal sumber daya*	*523751269255.dkr.ecr.ap-east-1.amazonaws.com/: menandai sagemaker-distribution-prod*

ap-south-1	arn:aws:sagemaker:ap-south-1:394103062818:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-south-1:245090515133:image/*pengenal sumber daya*	*245090515133.dkr.ecr.ap-south-1.amazonaws.com/: menandai sagemaker-distribution-prod*
ap-northeast-2	arn:aws:sagemaker:ap-northeast-2:806072073708:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-northeast-2:064688005998:image/*pengenal sumber daya*	*064688005998.dkr.ecr.ap-northeast-2.amazonaws.com/: menandai sagemaker-distribution-prod*
ap-southeast-1	arn:aws:sagemaker:ap-southeast-1:492261229750:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-southeast-1:022667117163:image/*pengenal sumber daya*	*022667117163.dkr.ecr.ap-southeast-1.amazonaws.com/: menandai sagemaker-distribution-prod*

ap-southeast-2	arn:aws:sagemaker:ap-southeast-2:452832661640:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-southeast-2:648430277019:image/*pengenal sumber daya*	*648430277019.dkr.ecr.ap-southeast-2.amazonaws.com/:menandai sagemaker-distribution-prod*
ap-northeast-1	arn:aws:sagemaker:ap-northeast-1:102112518831:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-northeast-1:010972774902:image/*pengenal sumber daya*	*010972774902.dkr.ecr.ap-northeast-1.amazonaws.com/:menandai sagemaker-distribution-prod*
ca-central-1	arn:aws:sagemaker:ca-central-1:310906938811:image/*pengenal sumber daya*	arn:aws:sagemaker:ca-central-1:481561238223:image/*pengenal sumber daya*	*481561238223.dkr.ecr.ca-central-1.amazonaws.com/: menandai sagemaker-distribution-prod*

eu-central-1	arn:aws:sagemaker:eu-central-1:936697816551:image/*pengenal sumber daya*	arn:aws:sagemaker:eu-central-1:545423591354:image/*pengenal sumber daya*	*545423591354.dkr.ecr.eu-central-1.amazonaws.com/: menandai sagemaker-distribution-prod*
eu-west-1	arn:aws:sagemaker:eu-west-1:470317259841:image/*pengenal sumber daya*	arn:aws:sagemaker:eu-west-1:819792524951:image/*pengenal sumber daya*	*819792524951.dkr.ecr.eu-west-1.amazonaws.com/: menandai sagemaker-distribution-prod*
eu-west-2	arn:aws:sagemaker:eu-west-2:712779665605:image/*pengenal sumber daya*	arn:aws:sagemaker:eu-west-2:021081402939:image/*pengenal sumber daya*	*021081402939.dkr.ecr.eu-west-2.amazonaws.com/: tag sagemaker-distribution-prod*

eu-west-3	arn:aws:sagemaker:eu-west-3:615547856133:image/*pengenal sumber daya*	arn:aws:sagemaker:eu-west-3:856416204555:image/*pengenal sumber daya*	*856416204555.dkr.ecr.eu-west-3.amazonaws.com/: menandai sagemaker-distribution-prod*
eu-north-1	arn:aws:sagemaker:eu-north-1:243637512696:image/*pengenal sumber daya*	arn:aws:sagemaker:eu-north-1:175620155138:image/*pengenal sumber daya*	*175620155138.dkr.ecr.eu-north-1.amazonaws.com/: menandai sagemaker-distribution-prod*
eu-south-1	arn:aws:sagemaker:eu-south-1:592751261982:image/*pengenal sumber daya*	arn:aws:sagemaker:eu-south-1:810671768855:image/*pengenal sumber daya*	*810671768855.dkr.ecr.eu-south-1.amazonaws.com/: menandai sagemaker-distribution-prod*

sa-east-1	arn:aws:sagemaker:sa-east-1:782484402741:image/*pengenal sumber daya*	arn:aws:sagemaker:sa-east-1:567556641782:image/*pengenal sumber daya*	*567556641782.dkr.ecr.sa-east-1.amazonaws.com/: menandai sagemaker-distribution-prod*
ap-northeast-3	arn:aws:sagemaker:ap-northeast-3:792733760839:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-northeast-3:564864627153:image/*pengenal sumber daya*	*564864627153.dkr.ecr.ap-northeast-3.amazonaws.com/: menandai sagemaker-distribution-prod*
ap-southeast-3	arn:aws:sagemaker:ap-southeast-3:276181064229:image/*pengenal sumber daya*	arn:aws:sagemaker:ap-southeast-3:370607712162:image/*pengenal sumber daya*	*370607712162.dkr.ecr.ap-southeast-3.amazonaws.com/: menandai sagemaker-distribution-prod*

me-south-1	arn:aws:sagemaker:me-south-1:117516905037:image/*pengenal sumber daya*	arn:aws:sagemaker:me-south-1:523774347010:image/*pengenal sumber daya*	*523774347010.dkr.ecr.me-south-1.amazonaws.com/: menandai sagemaker-distribution-prod*
me-central-1	arn:aws:sagemaker:me-central-1:103105715889:image/*pengenal sumber daya*	arn:aws:sagemaker:me-central-1:358593528301:image/*pengenal sumber daya*	*358593528301.dkr.ecr.me-central-1.amazonaws.com/: menandai sagemaker-distribution-prod*

Tag URI yang didukung

Daftar berikut menunjukkan tag yang dapat Anda sertakan dalam URI gambar Anda.

- 1-cpu
- 1-gpu
- 0-cpu
- 0-gpu

Contoh berikut menunjukkan URI dengan berbagai format tag:

- 542918446943.dkr. ecr.us-west-2.amazonaws.com /:1-cpu sagemaker-distribution-prod
- 542918446943.dkr. ecr.us-west-2.amazonaws.com /:0-gpu sagemaker-distribution-prod

Gambar yang didukung

Tabel berikut memberikan informasi tentang SageMaker gambar dan kernel terkait yang tersedia di Amazon SageMaker Studio Classic, serta pengenal sumber daya dan versi Python yang disertakan dalam gambar.

SageMaker gambar dan kernel

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
SageMaker Distribusi v0 CPU	`SageMaker Distribution v0 CPU`adalah gambar Python 3.8 yang mencakup kerangka kerja populer untuk pembelajaran mesin, ilmu data, dan visualisasi pada CPU. Ini termasuk kerangka pembelajaran mendalam seperti PyTorch, TensorFlow dan Keras; paket Python populer seperti numpy, scikit-learn dan panda; dan IDE seperti Jupyter Lab. Untuk informasi selengkapnya, lihat repo SageMaker Distribusi Amazon.	sagemaker-distribution-cpu-v0	Python 3 (python3)	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
SageMaker Distribusi v0 GPU	`SageMaker Distribution v0 GPU`adalah gambar Python 3.8 yang mencakup kerangka kerja populer untuk pembelajaran mesin, ilmu data, dan visualisasi pada GPU. Ini termasuk kerangka pembelajaran mendalam seperti PyTorch, TensorFlow dan Keras; paket Python populer seperti numpy, scikit-le arn dan panda; dan IDE seperti Jupyter Lab. Untuk informasi selengkapnya, lihat repo SageMaker Distribusi Amazon.	sagemaker-distribu tion-gpu-v0	Python 3 (python3)	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
SageMaker Distribusi v1 CPU	`SageMaker Distribution v1` CPUadalah gambar Python 3.10 yang mencakup kerangka kerja populer untuk pembelajaran mesin, ilmu data, dan analitik data pada CPU. Ini termasuk kerangka pembelajaran mendalam seperti PyTorch, TensorFlow dan Keras; paket Python populer seperti numpy, scikit-learn dan panda; dan IDE seperti Jupyter Lab. Untuk informasi selengkapnya, lihat repo SageMaker Distribusi Amazon.	sagemaker-distribution-cpu-v1	Python 3 (python3)	Python 3.10

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
SageMaker Distribusi v1 GPU	`SageMaker Distribution v1 GPU`adalah gambar Python 3.10 yang mencakup kerangka kerja populer untuk pembelajaran mesin, ilmu data, dan analitik data pada GPU. Ini termasuk kerangka pembelajaran mendalam seperti PyTorch, TensorFlow dan Keras; paket Python populer seperti numpy, scikit-learn dan panda; dan IDE seperti Jupyter Lab. Untuk informasi selengkapnya, lihat repo SageMaker Distribusi Amazon.	sagemaker-distribution-gpu-v1	Python 3 (python3)	Python 3.10

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
Dasar Python 3.0	Gambar resmi Python 3.10 dari DockerHub dengan boto3 dan disertakan. AWS CLI	sagemaker-base-python-310-v1	Python 3 (python3)	Python 3.10
Dasar Python 2.0	Gambar resmi Python 3.8 dari DockerHub dengan boto3 dan disertakan. AWS CLI	sagemaker-base-python-38	Python 3 (python3)	Python 3.8
Ilmu Data 3.0	`Data Science 3.0`adalah gambar conda Python 3.10 berdasarkan anaconda versi 2022.10 dengan paket dan pustaka Python yang paling umum digunakan, seperti dan Belajar. NumPy SciKit	sagemaker-data-science-310-v1	Python 3 (python3)	Python 3.10

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
Ilmu Data 2.0	`Data Science 2.0`adalah gambar conda Python 3.8 berdasarkan anaconda versi 2021.11 dengan paket dan pustaka Python yang paling umum digunakan, seperti dan Belajar. NumPy SciKit	sagemaker-data-science-38	Python 3 (python3)	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
Geospasial 1.0	Amazon SageMaker geospasial adalah gambar Python yang terdiri dari perpustakaan geospasial yang umum digunakan seperti GDAL, Fiona,, Shapely, dan Rasterio, dan GeoPandas memungkin kan Anda untuk memvisualisasikan data geospasia l di dalamnya. SageMaker Untuk informasi selengkapnya, lihat SDK Notebook SageMaker geospasial Amazon	pembuat sagemer-geospasial-1.0	Python 3 (python3)	Python 3.10

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
PyTorch 2.0.0 Python 3.10 CPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 2.0.0 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	pytorch-2.0.0-cpu-py310	Python 3 (python3)	Python 3.10
PyTorch 2.0.0 Python 3.10 GPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 2.0.0 dengan CUDA 11.8 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	pytorch-2.0.0-gpu-py310	Python 3 (python3)	Python 3.10

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
PyTorch 1.13 Python 3.9 CPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 1.13 dengan CUDA 11.3 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	pytorch-1.13-cpu-py39	Python 3 (python3)	Python 3.9
PyTorch 1.13 Python 3.9 GPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 1.13 dengan CUDA 11.7 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	pytorch-1.13-gpu-py39	Python 3 (python3)	Python 3.9

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
PyTorch 1.12 Python 3.8 CPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 1.12 dengan CUDA 11.3 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat AWSDeep Learning Containers untuk PyTorch 1.12.0.	pytorch-1.12-cpu-py38	Python 3 (python3)	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
PyTorch 1.12 Python 3.8 GPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 1.12 dengan CUDA 11.3 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat AWSDeep Learning Containers untuk PyTorch 1.12.0.	pytorch-1.12-gpu-py38	Python 3 (python3)	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
PyTorch 1.10 Python 3.8 CPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 1.10 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skalaAWS. Untuk informasi selengkapnya, lihat AWSDeep Learning Containers untuk PyTorch 1.10.2 di. SageMaker	pytorch-1.10-cpu-py38	Python 3 (python3)	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
PyTorch 1.10 Python 3.8 GPU Dioptimalkan	AWSDeep Learning Containers untuk PyTorch 1.10 dengan CUDA 11.3 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat AWSDeep Learning Containers untuk PyTorch 1.10.2 di. SageMaker	pytorch-1.10-gpu-py38	Python 3 (python3)	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
SparkAnalytics 2.0	Anaconda Individual Edition dengan PySpark dan Spark kernel. Untuk informasi lebih lanjut, lihat sparkmagic.	pembuat sagemer-sparkanalitik-310-v1	• SparkMagic Spark (conda-env-sm_sparkmagic-sparkkernel) • SparkMagic PySpark (conda-env-sm_sparkmagic-pysparkkernel) • Glue Spark (conda-env-sm_glue_is-glue_spark) • Glue Python [PySpark dan Ray] (_glue_is-glue_pyspark) conda-env-sm	Python 3.10

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
SparkAnalytics 1.0	Anaconda Individual Edition dengan PySpark dan Spark kernel. Untuk informasi lebih lanjut, lihat sparkmagic.	sagemaker-sparkanalytics-v1	• SparkMagic Spark (conda-env-sm_sparkmagic-sparkkernel) • SparkMagic PySpark (conda-env-sm_sparkmagic-pysparkkernel) • Glue Spark (conda-env-sm_glue_is-glue_spark) • Glue Python [PySpark dan Ray] (_glue_is-glue_pyspark) conda-env-sm	Python 3.8

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
TensorFlow 2.12.0 Python 3.10 CPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.12.0 dengan CUDA 11.2 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	tensorflow-2.12.0-cpu-py310-ubuntu20.04-sagemaker-v1.0	Python 3 (python3)	Python 3.10
TensorFlow 2.12.0 Python 3.10 GPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.12.0 dengan CUDA 11.8 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	tensorflow-2.12.0-gpu-py310-cu118-ubuntu20.04-sagemaker-v1	Python 3 (python3)	Python 3.10

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
TensorFlow 2.11.0 Python 3.9 CPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.11.0 dengan CUDA 11.2 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	tensorflow-2.11.0-cpu-py39-ubuntu20.04-sagemaker-v1.1	Python 3 (python3)	Python 3.9
TensorFlow 2.11.0 Python 3.9 GPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.11.0 dengan CUDA 11.2 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	tensorflow-2.11.0-gpu-py39-cu112-ubuntu20.04-sagemaker-v1.1	Python 3 (python3)	Python 3.9

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
TensorFlow 2.10 Python 3.9 CPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.10 dengan CUDA 11.2 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	tensorflow-2.10.1-cpu-py39-ubuntu20.04-sagemaker-v1.2	Python 3 (python3)	Python 3.9
TensorFlow 2.10 Python 3.9 GPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.10 dengan CUDA 11.2 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat Catatan Rilis untuk Deep Learning Containers.	tensorflow-2.10.1-gpu-py39-ubuntu20.04-sagemaker-v1.2	Python 3 (python3)	Python 3.9

SageMaker Gambar	Deskripsi	Pengidentifikasi Sumber Daya	Kernel (dan Identifier)	Versi Python
TensorFlow 2.6 Python 3.8 CPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.6 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat AWSDeep Learning Containers untuk TensorFlow 2.6.	tensorflow-2.6-cpu-py38-ubuntu20.04-v1	Python 3 (python3)	Python 3.8
TensorFlow 2.6 Python 3.8 GPU Dioptimalkan	AWSDeep Learning Containers untuk TensorFlow 2.6 dengan CUDA 11.2 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat AWSDeep Learning Containers untuk TensorFlow 2.6.	tensorflow-2.6-gpu-py38-cu112-ubuntu20.04-v1	Python 3 (python3)	Python 3.8

Gambar dijadwalkan untuk penghentian

SageMaker mengakhiri dukungan untuk gambar sehari setelah salah satu paket dalam gambar mencapai akhir kehidupan oleh penerbit mereka.

SageMaker Gambar-gambar berikut dijadwalkan untuk penghentian. Gambar-gambar ini didasarkan pada Python 3.7, yang dicapai end-of-lifepada 27 Juni 2023. Mulai 30 Oktober 2023, SageMaker akan menghentikan dukungan untuk gambar-gambar ini dan mereka tidak akan dapat dipilih dari UI Studio Classic. Untuk menghindari masalah ketidakpatuhan, jika Anda menggunakan salah satu gambar ini, sebaiknya Anda pindah ke gambar dengan versi yang lebih baru.

SageMaker gambar dijadwalkan untuk penghentian

SageMaker Gambar	Tanggal pengusangan	Deskripsi	Pengidentifikasi Sumber Daya	Kernel	Versi Python
Ilmu Data	30 Oktober 2023	`Data Science`adalah gambar conda Python 3.7 dengan paket dan pustaka Python yang paling umum digunakan, seperti dan Belajar. NumPy SciKit	ilmu data-1.0	Python 3	Python 3.7
SageMaker JumpStart Ilmu Data 1.0	30 Oktober 2023	`SageMaker JumpStart Data Science 1.0`adalah SageMaker JumpStart gambar yang mencakup paket dan pustaka	sagemaker-jumpstart-data-science-1,0	Python 3	Python 3.7

SageMaker Gambar	Tanggal pengusangan	Deskripsi	Pengidentifikasi Sumber Daya	Kernel	Versi Python
		yang umum digunakan.			
SageMaker JumpStart MxNet 1.0	30 Oktober 2023	`SageMaker JumpStart MXNet 1.0`adalah SageMaker JumpStart gambar yang mencakup MXNet.	sagemaker -jumpstart-mxnet-1,0	Python 3	Python 3.7
SageMaker JumpStart PyTorch 1.0	30 Oktober 2023	`SageMaker JumpStart PyTorch 1.0`adalah SageMaker JumpStart gambar yang mencakup PyTorch.	sagemaker -jumpstart-pytorch-1,0	Python 3	Python 3.7
SageMaker JumpStart TensorFlow 1.0	30 Oktober 2023	`SageMaker JumpStart TensorFlo w 1.0` adalah SageMaker JumpStart gambar yang mencakup TensorFlow.	sagemaker -jumpstart-tensorflow-1,0	Python 3	Python 3.7

SageMaker Gambar	Tanggal pengusangan	Deskripsi	Pengidentifikasi Sumber Daya	Kernel	Versi Python
SparkMagic	30 Oktober 2023	Anaconda Individual Edition dengan PySpark dan Spark kernel. Untuk informasi lebih lanjut, lihat sparkmagic.	sagemaker-sparkmagic	• PySpar • Spark	Python 3.7
TensorFlow 2.3 Python 3.7 CPU Dioptimalkan	30 Oktober 2023	AWSDeep Learning Containers untuk TensorFlow 2.3 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skalaAWS. Untuk informasi selengkapnya, lihat AWSDeep Learning Containers dengan TensorFlow 2.3.0.	tensorflow-2.3-cpu-py37-ubuntu18.04-v1	Python 3	Python 3.7

SageMaker Gambar	Tanggal pengusangan	Deskripsi	Pengidentifikasi Sumber Daya	Kernel	Versi Python
TensorFlow 2.3 Python 3.7 GPU Dioptimalkan	30 Oktober 2023	AWSDeep Learning Containers untuk TensorFlow 2.3 dengan CUDA 11.0 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat AWSDeep Learning Containers untuk TensorFlow 2.3.1 dengan CUDA 11.0.	tensorflow-2.3-gpu-py37-cu110-ubuntu18.04-v3	Python 3	Python 3.7

SageMaker Gambar	Tanggal pengusangan	Deskripsi	Pengidentifikasi Sumber Daya	Kernel	Versi Python
TensorFlow 1.15 Python 3.7 CPU Dioptimalkan	30 Oktober 2023	AWSDeep Learning Containers untuk TensorFlow 1.15 mencakup wadah untuk pelatihan tentang CPU, dioptimalkan untuk kinerja dan skalaAWS. Untuk informasi selengkapnya, lihat AWSDeep Learning Containers v7.0 untuk. TensorFlow	tensorflow-1.15-cpu-py37-ubuntu18.04-v7	Python 3	Python 3.7

SageMaker Gambar	Tanggal pengusangan	Deskripsi	Pengidentifikasi Sumber Daya	Kernel	Versi Python
TensorFlow 1.15 Python 3.7 GPU Dioptimalkan	30 Oktober 2023	AWSDeep Learning Containers untuk TensorFlow 1.15 dengan CUDA 11.0 mencakup wadah untuk pelatihan tentang GPU, dioptimalkan untuk kinerja dan skala. AWS Untuk informasi selengkapnya, lihat AWSDeep Learning Containers v7.0 untuk. TensorFlow	tensorflow-1.15-gpu-py37-cu110-ubuntu18.04-v8	Python 3	Python 3.7

Kustomisasi Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Ada empat opsi untuk menyesuaikan lingkungan Amazon SageMaker Studio Classic Anda. Anda membawa SageMaker gambar Anda sendiri, menggunakan skrip konfigurasi siklus hidup, melampirkan repo Git yang disarankan ke Studio Classic, atau membuat kernel menggunakan

lingkungan Conda persisten di Amazon EFS. Gunakan setiap opsi secara individual, atau bersama-sama.

- Bawa SageMaker gambar Anda sendiri: SageMaker Gambar adalah file yang mengidentifikasi kernel, paket bahasa, dan dependensi lain yang diperlukan untuk menjalankan notebook Jupyter di Amazon Studio Classic. SageMaker Amazon SageMaker menyediakan banyak gambar bawaan untuk Anda gunakan. Jika Anda memerlukan fungsionalitas yang berbeda, Anda dapat membawa gambar kustom Anda sendiri ke Studio Classic.
- Gunakan konfigurasi siklus hidup dengan Amazon SageMaker Studio Classic: Konfigurasi siklus hidup adalah skrip shell yang dipicu oleh peristiwa siklus hidup Amazon SageMaker Studio Classic, seperti memulai notebook Studio Classic baru. Anda dapat menggunakan konfigurasi siklus hidup untuk mengotomatiskan penyesuaian lingkungan Studio Classic Anda. Misalnya, Anda dapat menginstal paket kustom, mengkonfigurasi ekstensi notebook, preload dataset, dan mengatur repositori kode sumber.
- Lampirkan repo Git yang disarankan ke Studio Classic: Anda dapat melampirkan URL repositori Git yang disarankan di tingkat SageMaker Domain Amazon atau profil pengguna. Kemudian, Anda dapat memilih URL repo dari daftar saran dan mengkloningnya ke lingkungan Anda menggunakan ekstensi Git di Studio Classic.
- Pertahankan lingkungan Conda ke volume Studio Classic Amazon EFS: Studio Classic menggunakan volume Amazon EFS sebagai lapisan penyimpanan persisten. Anda dapat menyimpan lingkungan Conda di volume Amazon EFS ini, lalu menggunakan lingkungan yang disimpan untuk membuat kernel. Studio Classic secara otomatis mengambil semua lingkungan valid yang disimpan di Amazon EFS sebagai KernelGateway kernel. Kernel ini bertahan melalui restart kernel, aplikasi, dan Studio Classic. Untuk informasi selengkapnya, lihat lingkungan Persist Conda ke bagian volume Studio Classic EFS dalam Empat pendekatan untuk mengelola paket Python di notebook Amazon SageMaker Studio Classic.

Topik berikut menunjukkan cara menggunakan tiga opsi ini untuk menyesuaikan lingkungan Amazon SageMaker Studio Classic Anda.

Topik

Bawa SageMaker gambar Anda sendiri

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

SageMaker Gambar adalah file yang mengidentifikasi kernel, paket bahasa, dan dependensi lain yang diperlukan untuk menjalankan notebook Jupyter di Amazon Studio Classic. SageMaker Gambar-gambar ini digunakan untuk membuat lingkungan tempat Anda menjalankan notebook Jupyter. Amazon SageMaker menyediakan banyak gambar bawaan untuk Anda gunakan. Untuk daftar gambar bawaan, lihat SageMaker Gambar Amazon yang Tersedia.

Jika Anda memerlukan fungsionalitas yang berbeda, Anda dapat membawa gambar kustom Anda sendiri ke Studio Classic. Anda dapat membuat gambar dan versi gambar, dan melampirkan versi gambar ke domain atau ruang bersama, menggunakan panel SageMaker kontrol AWS SDK for Python (Boto3), dan AWS Command Line Interface(AWS CLI). Anda juga dapat membuat gambar dan versi gambar menggunakan SageMaker konsol, bahkan jika Anda belum onboard ke domain. SageMaker SageMakermenyediakan contoh Dockerfiles untuk digunakan sebagai titik awal untuk SageMaker gambar kustom Anda di repositori SageMaker Studio Classic Custom Image Sampel.

Topik berikut menjelaskan cara membawa gambar Anda sendiri menggunakan SageMaker konsol atauAWS CLI, lalu luncurkan gambar di Studio Classic. Untuk artikel blog serupa, lihat Membawa lingkungan R Anda sendiri ke Amazon SageMaker Studio Classic. Untuk buku catatan yang menunjukkan cara membawa gambar Anda sendiri untuk digunakan dalam pelatihan dan inferensi, lihat Amazon SageMaker Studio Classic Container Build CLI.

Terminologi kunci

Bagian berikut mendefinisikan istilah kunci untuk membawa gambar Anda sendiri untuk digunakan dengan Studio Classic.

- Dockerfile: Dockerfile adalah file yang mengidentifikasi paket bahasa dan dependensi lain untuk image Docker Anda.
- Gambar Docker: Gambar Docker adalah Dockerfile yang dibangun. Gambar ini diperiksa ke Amazon ECR dan berfungsi sebagai dasar SageMaker gambar.

- SageMaker image: SageMaker Gambar adalah dudukan untuk satu set versi SageMaker gambar berdasarkan gambar Docker. Setiap versi gambar tidak dapat diubah.
- Versi gambar: Versi gambar dari gambar mewakili SageMaker gambar Docker dan disimpan dalam repositori Amazon ECR. Setiap versi gambar tidak dapat diubah. Versi gambar ini dapat dilampirkan ke domain atau ruang bersama dan digunakan dengan Studio Classic.

Topik

Spesifikasi SageMaker gambar kustom

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Spesifikasi berikut berlaku untuk gambar kontainer yang diwakili oleh versi SageMaker gambar.

Menjalankan gambar

`ENTRYPOINT`dan `CMD` instruksi diganti untuk mengaktifkan gambar berjalan sebagai aplikasi. KernelGateway

Port 8888 pada gambar dicadangkan untuk menjalankan server KernelGateway web.

Menghentikan gambar

`DeleteApp`API mengeluarkan `docker stop` perintah yang setara. Proses lain dalam wadah tidak akan mendapatkan sinyal SIGKILL/SIGTERM.

Penemuan kernel

SageMaker mengenali kernel seperti yang didefinisikan oleh spesifikasi kernel Jupyter.

Anda dapat menentukan daftar kernel yang akan ditampilkan sebelum menjalankan gambar. Jika tidak ditentukan, python3 ditampilkan. Gunakan DescribeAppImageConfigAPI untuk melihat daftar kernel.

Lingkungan Conda diakui sebagai spesifikasi kernel secara default.

Sistem file

`/opt/ml`Direktori `/opt/.sagemakerinternal` dan dicadangkan. Data apa pun di direktori ini mungkin tidak terlihat saat runtime.

Data pengguna

Setiap pengguna dalam domain mendapatkan direktori pengguna pada volume Amazon Elastic File System bersama dalam gambar. Lokasi direktori pengguna saat ini pada volume Amazon EFS dapat dikonfigurasi. Secara default, lokasi direktori adalah`/home/sagemaker-user`.

SageMaker mengkonfigurasi pemetaan POSIX UID/GID antara gambar dan host. Ini default untuk memetakan UID/GID pengguna root (0/0) ke UID/GID pada host.

Anda dapat menentukan nilai-nilai ini menggunakan CreateAppImageConfigAPI.

Batas GID/UID

Amazon SageMaker Studio Classic hanya mendukung yang berikut `DefaultUID` dan `DefaultGID` kombinasi:

- DefaultUid: 1000 dan defaultGid: 100, yang sesuai dengan pengguna non-priveleged.
- DefaultUid: 0 dan defaultGid: 0, yang sesuai dengan akses root.

Metadata

File metadata terletak di. `/opt/ml/metadata/resource-metadata.json` Tidak ada variabel lingkungan tambahan yang ditambahkan ke variabel yang ditentukan dalam gambar. Untuk informasi selengkapnya, lihat Dapatkan Metadata Aplikasi.

GPU

Pada instance GPU, gambar dijalankan dengan `--gpus` opsi. Hanya toolkit CUDA yang harus disertakan dalam gambar bukan driver NVIDIA. Untuk informasi selengkapnya, lihat Panduan Pengguna NVIDIA.

Metrik dan pencatatan

Log dari KernelGateway proses dikirim ke Amazon CloudWatch di akun pelanggan. Nama grup log adalah`/aws/sagemaker/studio`. Nama aliran log adalah`$domainID/$userProfileName/KernelGateway/$appName`.

Ukuran gambar

Terbatas hingga 25 GB. Untuk melihat ukuran gambar Anda, jalankan`docker image ls`.

Contoh Dockerfile

Contoh berikut Dockerfile membuat Amazon Linux 2 berbasis gambar, menginstal paket pihak ketiga dan `python3` kernel, dan menetapkan cakupan ke pengguna yang tidak memiliki hak istimewa.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2

ARG NB_USER="sagemaker-user"
ARG NB_UID="1000"
ARG NB_GID="100"

RUN \
    yum install --assumeyes python3 shadow-utils && \
    useradd --create-home --shell /bin/bash --gid "${NB_GID}" --uid ${NB_UID}
 ${NB_USER} && \
    yum clean all && \
    python3 -m pip install ipykernel && \
    python3 -m ipykernel install

USER ${NB_UID}
```

Prasyarat

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda harus memenuhi prasyarat berikut untuk membawa wadah Anda sendiri untuk digunakan dengan Amazon Studio Classic. SageMaker

- Aplikasi Docker. Untuk informasi tentang pengaturan Docker, lihat Orientasi dan penyiapan.
- Instal AWS CLI dengan mengikuti langkah-langkah di Memulai dengan AWS CLI.
- Salinan lokal dari Dockerfile apa pun untuk membuat gambar yang kompatibel dengan Studio Classic. Untuk contoh gambar kustom, lihat repositori sampel gambar kustom SageMaker Studio Classic.
- Izin untuk mengakses layanan Amazon Elastic Container Registry (Amazon ECR). Untuk informasi selengkapnya, lihat Kebijakan Terkelola Amazon ECR.
- Peran AWS Identity and Access Management eksekusi yang memiliki AmazonSageMakerFullAccesskebijakan terlampir. Jika Anda telah onboard ke SageMaker domain Amazon, Anda bisa mendapatkan peran dari bagian Ringkasan Domain pada panel SageMaker kontrol.
- Instal CLI build image Studio Classic dengan mengikuti langkah-langkah di SageMaker Docker Build. CLI ini memungkinkan Anda untuk membangun Dockerfile menggunakan. AWS CodeBuild

Tambahkan gambar Docker yang kompatibel dengan Studio Classic ke Amazon ECR

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda melakukan langkah-langkah berikut untuk menambahkan gambar kontainer ke Amazon ECR:

- Buat repositori Amazon ECR.
- Otentikasi ke Amazon ECR.
- Buat gambar Docker yang kompatibel dengan Studio Classic.
- Dorong gambar ke repositori Amazon ECR.

> **Note**
>
> Repositori Amazon ECR harus sama Wilayah AWS dengan Studio Classic.

Untuk membuat dan menambahkan gambar kontainer ke Amazon ECR

1. Buat repositori Amazon ECR menggunakan file. AWS CLI Untuk membuat repositori menggunakan konsol Amazon ECR, lihat Membuat repositori.

   ```
   aws ecr create-repository \
       --repository-name smstudio-custom \
       --image-scanning-configuration scanOnPush=true
   ```

   Responsnya akan terlihat mirip dengan yang berikut ini.

   ```
   {
       "repository": {
           "repositoryArn": "arn:aws:ecr:us-east-2:acct-id:repository/smstudio-
   custom",
           "registryId": "acct-id",
           "repositoryName": "smstudio-custom",
           "repositoryUri": "acct-id.dkr.ecr.us-east-2.amazonaws.com/smstudio-custom",
           ...
       }
   }
   ```

2. Bangun `Dockerfile` menggunakan CLI build image Studio Classic. Periode (.) menentukan bahwa Dockerfile harus dalam konteks perintah build. Perintah ini membangun gambar dan mengunggah gambar yang dibangun ke repo ECR. Kemudian menampilkan URI gambar.

```
sm-docker build . --repository smstudio-custom:custom
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
Image URI: <acct-id>.dkr.ecr.<region>.amazonaws.com/<image_name>
```

Buat SageMaker gambar kustom

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik ini menjelaskan bagaimana Anda dapat membuat SageMaker gambar kustom menggunakan SageMaker konsol atauAWS CLI.

Saat Anda membuat gambar dari konsol, SageMaker juga membuat versi gambar awal. Versi gambar mewakili gambar kontainer di Amazon Elastic Container Registry (ECR). Gambar kontainer harus memenuhi persyaratan yang akan digunakan di Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Spesifikasi SageMaker gambar kustom. Untuk informasi tentang pengujian gambar Anda secara lokal dan menyelesaikan masalah umum, lihat repo Sampel Gambar Kustom SageMaker Studio Classic.

Setelah Anda membuat SageMaker gambar kustom, Anda harus melampirkannya ke domain atau ruang bersama untuk menggunakannya dengan Studio Classic. Untuk informasi selengkapnya, lihat Lampirkan SageMaker gambar khusus.

Buat SageMaker gambar dari konsol

Bagian berikut menunjukkan cara membuat SageMaker gambar khusus dari SageMaker konsol.

Untuk membuat gambar

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.

2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Gambar.
4. Pada halaman Custom images, pilih Create image.
5. Untuk Sumber gambar, masukkan jalur registri ke gambar kontainer di Amazon ECR. Path dalam format berikut:

   ```
   acct-id.dkr.ecr.region.amazonaws.com/repo-name[:tag] or [@digest]
   ```

6. Pilih Berikutnya.
7. Di bawah properti Gambar, masukkan yang berikut ini:

   - Nama gambar — Nama harus unik untuk akun Anda saat iniWilayah AWS.
   - (Opsional) Nama tampilan - Nama yang ditampilkan di antarmuka pengguna Studio Classic. Ketika tidak disediakan, `Image name` ditampilkan.
   - (Opsional) Deskripsi — Deskripsi gambar.
   - Peran IAM — Peran harus memiliki AmazonSageMakerFullAccesskebijakan yang dilampirkan. Gunakan menu tarik-turun untuk memilih salah satu opsi berikut:
     - Buat peran baru — Tentukan bucket Amazon Simple Storage Service (Amazon S3) tambahan yang dapat diakses oleh pengguna notebook. Jika Anda tidak ingin mengizinkan akses ke bucket tambahan, pilih Tidak Ada.

       SageMaker melekatkan `AmazonSageMakerFullAccess` kebijakan pada peran. Peran ini memungkinkan pengguna notebook Anda mengakses bucket S3 yang tercantum di sebelah tanda centang.
     - Masukkan peran IAM khusus ARN — Masukkan Nama Sumber Daya Amazon (ARN) peran IAM Anda.
     - Gunakan peran yang ada — Pilih salah satu peran yang ada dari daftar.
   - (Opsional) Tag gambar - Pilih Tambahkan tag baru. Anda dapat menambahkan hingga 50 tanda. Tag dapat dicari menggunakan antarmuka pengguna Studio Classic, SageMaker konsol, atau API. SageMaker `Search`
8. Pilih Kirim.

Gambar baru ditampilkan dalam daftar gambar Kustom dan disorot secara singkat. Setelah gambar berhasil dibuat, Anda dapat memilih nama gambar untuk melihat propertinya atau memilih Buat versi untuk membuat versi lain.

Untuk membuat versi gambar lain

1. Pilih Buat versi pada baris yang sama dengan gambar.
2. Untuk Sumber gambar, masukkan jalur registri ke gambar kontainer Amazon ECR. Gambar kontainer tidak boleh menjadi gambar yang sama seperti yang digunakan dalam versi SageMaker gambar sebelumnya.

Buat SageMaker gambar dari AWS CLI

Anda melakukan langkah-langkah berikut untuk membuat SageMaker gambar dari gambar kontainer menggunakan fileAWS CLI.

- Buat `Image`.
- Buat `ImageVersion`.
- Buat file konfigurasi.
- Buat `AppImageConfig`.

Untuk membuat entitas SageMaker gambar

1. Buat SageMaker gambar.

```
aws sagemaker create-image \
    --image-name custom-image \
    --role-arn arn:aws:iam::<acct-id>:role/service-role/<execution-role>
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "ImageArn": "arn:aws:sagemaker:us-east-2:acct-id:image/custom-image"
}
```

2. Buat versi SageMaker gambar dari gambar kontainer.

```
aws sagemaker create-image-version \
    --image-name custom-image \
    --base-image <acct-id>.dkr.ecr.<region>.amazonaws.com/smstudio-custom:custom-
image
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "ImageVersionArn": "arn:aws:sagemaker:us-east-2:acct-id:image-version/custom-image/1"
}
```

3. Periksa apakah versi gambar berhasil dibuat.

```
aws sagemaker describe-image-version \
    --image-name custom-image \
    --version-number 1
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "ImageVersionArn": "arn:aws:sagemaker:us-east-2:acct-id:image-version/custom-image/1",
    "ImageVersionStatus": "CREATED"
}
```

> **Note**
>
> Jika responsnya`"ImageVersionStatus": "CREATED_FAILED"`, responsnya juga mencakup alasan kegagalan. Masalah izin adalah penyebab umum kegagalan. Anda juga dapat memeriksa CloudWatch log Amazon jika mengalami kegagalan saat memulai atau menjalankan KernelGateway aplikasi untuk gambar khusus. Nama grup log adalah`/aws/sagemaker/studio`. Nama aliran log adalah`$domainID/$userProfileName/KernelGateway/$appName`.

4. Buat file konfigurasi, bernama`app-image-config-input.json`. `NameNilai KernelSpecs` harus cocok dengan nama KernelSpec yang tersedia di Gambar yang terkait dengan ini. `AppImageConfig` Nilai ini peka huruf besar kecil. Anda dapat menemukan KernelSpecs yang tersedia dalam gambar dengan menjalankan `jupyter-kernelspec list` dari shell di dalam wadah. `MountPath`adalah jalur dalam gambar untuk memasang direktori home Amazon Elastic File System (Amazon EFS) Anda. Itu harus berbeda dari jalur yang Anda gunakan di dalam wadah karena jalur itu akan diganti ketika direktori home Amazon EFS Anda dipasang.

> **Note**
>
> Berikut `DefaultUID` dan `DefaultGID` kombinasi adalah satu-satunya nilai yang diterima:
>
> - DefaultUid: 1000 dan defaultGid: 100
> - DefaultUid: 0 dan defaultGid: 0

```
{
    "AppImageConfigName": "custom-image-config",
    "KernelGatewayImageConfig": {
        "KernelSpecs": [
            {
                "Name": "python3",
                "DisplayName": "Python 3 (ipykernel)"
            }
        ],
        "FileSystemConfig": {
            "MountPath": "/home/sagemaker-user",
            "DefaultUid": 1000,
            "DefaultGid": 100
        }
    }
}
```

5. Buat AppImageConfig menggunakan file yang dibuat pada langkah sebelumnya.

```
aws sagemaker create-app-image-config \
    --cli-input-json file://app-image-config-input.json
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "AppImageConfigArn": "arn:aws:sagemaker:us-east-2:acct-id:app-image-config/
custom-image-config"
}
```

Lampirkan SageMaker gambar khusus

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Untuk menggunakan SageMaker gambar kustom, Anda harus melampirkan versi gambar ke domain atau ruang bersama Anda. Saat Anda melampirkan versi gambar, itu akan muncul di SageMaker Studio Classic Launcher dan tersedia di daftar tarik-turun Pilih gambar, yang digunakan pengguna untuk meluncurkan aktivitas atau mengubah gambar yang digunakan oleh buku catatan.

Untuk membuat SageMaker gambar kustom tersedia untuk semua pengguna dalam domain, Anda melampirkan gambar ke domain. Untuk membuat gambar tersedia bagi semua pengguna dalam ruang bersama, Anda dapat melampirkan gambar ke ruang bersama. Untuk membuat gambar tersedia untuk satu pengguna, Anda melampirkan gambar ke profil pengguna. Saat Anda melampirkan gambar, SageMaker gunakan versi gambar terbaru secara default. Anda juga dapat melampirkan versi gambar tertentu. Setelah Anda melampirkan versi, Anda dapat memilih versi dari SageMaker Peluncur atau pemilih gambar saat Anda meluncurkan buku catatan.

Ada batasan jumlah versi gambar yang dapat dilampirkan pada waktu tertentu. Setelah Anda mencapai batas, Anda harus melepaskan versi untuk melampirkan versi lain dari gambar.

Bagian berikut menunjukkan cara melampirkan SageMaker gambar kustom ke domain Anda menggunakan SageMaker konsol atauAWS CLI. Anda hanya dapat melampirkan gambar kustom ke ruang berbagi menggunakanAWS CLI.

Lampirkan SageMaker gambar ke Domain

Lampirkan SageMaker gambar menggunakan Konsol

Topik ini menjelaskan bagaimana Anda dapat melampirkan versi SageMaker gambar kustom yang ada ke domain Anda menggunakan panel SageMaker kontrol. Anda juga dapat membuat SageMaker gambar kustom dan versi gambar, dan kemudian melampirkan versi itu ke domain Anda. Untuk prosedur membuat versi gambar dan gambar, lihatBuat SageMaker gambar kustom.

Untuk melampirkan gambar yang ada

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari halaman Domain, pilih Domain untuk melampirkan gambar.
5. Dari halaman Detail domain, pilih tab Lingkungan.
6. Pada tab Lingkungan, di bawah gambar Custom SageMaker Studio Classic yang dilampirkan ke domain, pilih Lampirkan gambar.
7. Untuk Sumber gambar, pilih Gambar yang ada.
8. Pilih gambar yang ada dari daftar.
9. Pilih versi gambar dari daftar.
10. Pilih Berikutnya.
11. Verifikasi nilai untuk nama Gambar, nama tampilan Gambar, dan Deskripsi.
12. Pilih peran IAM. Untuk informasi selengkapnya, lihat Buat SageMaker gambar kustom.
13. (Opsional) Tambahkan tag untuk gambar.
14. Tentukan jalur pemasangan EFS. Ini adalah jalur dalam gambar untuk me-mount direktori home Amazon Elastic File System (EFS) pengguna.
15. Untuk jenis Gambar, pilih Gambar SageMaker studio
16. Untuk nama Kernel, masukkan nama kernel yang ada di gambar. Untuk informasi tentang cara mendapatkan informasi kernel dari gambar, lihat PENGEMBANGAN di repositori Sampel Gambar Kustom SageMaker Studio Classic. Untuk informasi selengkapnya, lihat bagian Penemuan Kernel dan data PenggunaSpesifikasi SageMaker gambar kustom.
17. (Opsional) Untuk nama tampilan Kernel, masukkan nama tampilan untuk kernel.
18. Pilih Tambahkan kernel.
19. Pilih Kirim.

    - Tunggu hingga versi gambar dilampirkan ke domain. Saat dilampirkan, versi ditampilkan dalam daftar Gambar khusus dan disorot secara singkat.

Lampirkan SageMaker gambar menggunakan AWS CLI

Bagian berikut menunjukkan cara melampirkan SageMaker gambar kustom saat membuat domain baru atau memperbarui domain Anda yang ada menggunakanAWS CLI.

Lampirkan SageMaker gambar ke domain baru

Bagian berikut menunjukkan cara membuat domain baru dengan versi terlampir. Langkah-langkah ini mengharuskan Anda menentukan informasi Amazon Virtual Private Cloud (VPC) dan peran eksekusi yang diperlukan untuk membuat domain. Anda melakukan langkah-langkah berikut untuk membuat domain dan melampirkan SageMaker gambar kustom:

- Dapatkan ID VPC default dan ID subnet Anda.
- Buat file konfigurasi untuk domain, yang menentukan gambar.
- Buat domain dengan file konfigurasi.

Untuk menambahkan SageMaker gambar kustom ke domain Anda

1. Dapatkan ID VPC default Anda.

   ```
   aws ec2 describe-vpcs \
       --filters Name=isDefault,Values=true \
       --query "Vpcs[0].VpcId" --output text
   ```

   Responsnya akan terlihat mirip dengan yang berikut ini.

   ```
   vpc-xxxxxxxx
   ```

2. Dapatkan ID subnet default Anda menggunakan ID VPC dari langkah sebelumnya.

   ```
   aws ec2 describe-subnets \
       --filters Name=vpc-id,Values=<vpc-id> \
       --query "Subnets[*].SubnetId" --output json
   ```

   Responsnya akan terlihat mirip dengan yang berikut ini.

   ```
   [
       "subnet-b55171dd",
       "subnet-8a5f99c6",
       "subnet-e88d1392"
   ]
   ```

3. Buat file konfigurasi bernama`create-domain-input.json`. Masukkan ID VPC, ID subnet`ImageName`, dan `AppImageConfigName` dari langkah sebelumnya. Karena

`ImageVersionNumber` tidak ditentukan, versi terbaru dari gambar digunakan, yang merupakan satu-satunya versi dalam kasus ini.

```
{
    "DomainName": "domain-with-custom-image",
    "VpcId": "<vpc-id>",
    "SubnetIds": [
        "<subnet-ids>"
    ],
    "DefaultUserSettings": {
        "ExecutionRole": "<execution-role>",
        "KernelGatewayAppSettings": {
            "CustomImages": [
                {
                    "ImageName": "custom-image",
                    "AppImageConfigName": "custom-image-config"
                }
            ]
        }
    },
    "AuthMode": "IAM"
}
```

4. Buat domain dengan SageMaker gambar kustom terlampir.

```
aws sagemaker create-domain \
    --cli-input-json file://create-domain-input.json
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "DomainArn": "arn:aws:sagemaker:us-east-2:acct-id:domain/d-xxxxxxxxxxxx",
    "Url": "https://d-xxxxxxxxxxxx.studio.us-east-2.sagemaker.aws/..."
}
```

Lampirkan SageMaker gambar ke domain Anda saat ini

Jika Anda telah onboard ke SageMaker domain, Anda dapat melampirkan gambar kustom ke domain Anda saat ini. Untuk informasi selengkapnya tentang orientasi ke SageMaker domain, lihat[Ikhtisar SageMaker Domain Amazon](). Anda tidak perlu menentukan informasi VPC dan peran eksekusi

saat melampirkan gambar khusus ke domain Anda saat ini. Setelah melampirkan versi, Anda harus menghapus semua aplikasi di domain Anda dan membuka kembali Studio Classic. Untuk informasi tentang menghapus aplikasi, lihatHapus SageMaker Domain Amazon.

Anda melakukan langkah-langkah berikut untuk menambahkan SageMaker gambar ke domain Anda saat ini.

- Dapatkan `DomainID` dari panel SageMaker kontrol.
- Gunakan `DomainID` untuk mendapatkan domain. `DefaultUserSettings`
- Tambahkan `ImageName` dan `AppImageConfig` sebagai a `CustomImage` ke`DefaultUserSettings`.
- Perbarui domain Anda untuk menyertakan gambar kustom.

Untuk menambahkan SageMaker gambar kustom ke domain Anda

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari halaman Domain, pilih Domain untuk melampirkan gambar.
5. Dari halaman Detail domain, pilih tab Pengaturan domain.
6. Dari tab Pengaturan domain, di bawah Pengaturan umum, temukan`DomainId`. ID dalam format berikut:`d-xxxxxxxxxxxx`.
7. Gunakan ID domain untuk mendapatkan deskripsi domain.

```
aws sagemaker describe-domain \
    --domain-id <d-xxxxxxxxxxxx>
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "DomainId": "d-xxxxxxxxxxxx",
    "DefaultUserSettings": {
      "KernelGatewayAppSettings": {
        "CustomImages": [
        ],
        ...
```

```
      }
    }
}
```

8. Simpan bagian pengaturan pengguna default dari respons ke file bernama`default-user-settings.json`.
9. Masukkan `ImageName` dan `AppImageConfigName` dari langkah sebelumnya sebagai gambar khusus. Karena `ImageVersionNumber` tidak ditentukan, versi terbaru dari gambar digunakan, yang merupakan satu-satunya versi dalam kasus ini.

```
{
    "DefaultUserSettings": {
        "KernelGatewayAppSettings": {
           "CustomImages": [
              {
                 "ImageName": "string",
                 "AppImageConfigName": "string"
              }
           ],
           ...
        }
    }
}
```

10. Gunakan ID domain dan file pengaturan pengguna default untuk memperbarui domain Anda.

```
aws sagemaker update-domain \
    --domain-id <d-xxxxxxxxxxxx> \
    --cli-input-json file://default-user-settings.json
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "DomainArn": "arn:aws:sagemaker:us-east-2:acct-id:domain/d-xxxxxxxxxxxx"
}
```

Lampirkan SageMaker gambar ke ruang bersama

Anda hanya dapat melampirkan SageMaker gambar ke ruang bersama menggunakan fileAWS CLI. Setelah Anda melampirkan versi, Anda harus menghapus semua aplikasi di ruang bersama Anda

dan membuka kembali Studio Classic. Untuk informasi tentang menghapus aplikasi, lihatHapus SageMaker Domain Amazon.

Anda melakukan langkah-langkah berikut untuk menambahkan SageMaker gambar ke ruang bersama.

- Dapatkan `DomainID` dari panel SageMaker kontrol.
- Gunakan `DomainID` untuk mendapatkan domain. `DefaultSpaceSettings`
- Tambahkan `ImageName` dan `AppImageConfig` sebagai a `CustomImage` ke`DefaultSpaceSettings`.
- Perbarui domain Anda untuk menyertakan gambar kustom untuk ruang bersama.

Untuk menambahkan SageMaker gambar kustom ke ruang bersama Anda

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari halaman Domain, pilih Domain untuk melampirkan gambar.
5. Dari halaman Detail domain, pilih tab Pengaturan domain.
6. Dari tab Pengaturan domain, di bawah Pengaturan umum, temukan`DomainId`. ID dalam format berikut:`d-xxxxxxxxxxxx`.
7. Gunakan ID domain untuk mendapatkan deskripsi domain.

```
aws sagemaker describe-domain \
    --domain-id <d-xxxxxxxxxxxx>
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "DomainId": "d-xxxxxxxxxxxx",
    ...
    "DefaultSpaceSettings": {
      "KernelGatewayAppSettings": {
        "CustomImages": [
        ],
        ...
      }
```

```
    }
}
```

8. Simpan bagian pengaturan ruang default dari respons ke file bernama`default-space-settings.json`.
9. Masukkan `ImageName` dan `AppImageConfigName` dari langkah sebelumnya sebagai gambar khusus. Karena `ImageVersionNumber` tidak ditentukan, versi terbaru dari gambar digunakan, yang merupakan satu-satunya versi dalam kasus ini.

```
{
    "DefaultSpaceSettings": {
        "KernelGatewayAppSettings": {
           "CustomImages": [
              {
                 "ImageName": "string",
                 "AppImageConfigName": "string"
              }
           ],
           ...
        }
    }
}
```

10. Gunakan ID domain dan file pengaturan ruang default untuk memperbarui domain Anda.

```
aws sagemaker update-domain \
    --domain-id <d-xxxxxxxxxxxx> \
    --cli-input-json file://default-space-settings.json
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "DomainArn": "arn:aws:sagemaker:us-east-2:acct-id:domain/d-xxxxxxxxxxxx"
}
```

Lihat gambar terlampir di SageMaker

Setelah Anda membuat SageMaker gambar kustom dan melampirkannya ke domain Anda, gambar akan muncul di tab Lingkungan Domain. Anda hanya dapat melihat gambar terlampir untuk ruang bersama menggunakan AWS CLI dengan menggunakan perintah berikut.

```
aws sagemaker describe-domain \
    --domain-id <d-xxxxxxxxxxxx>
```

Luncurkan SageMaker gambar khusus di Amazon SageMaker Studio Classic

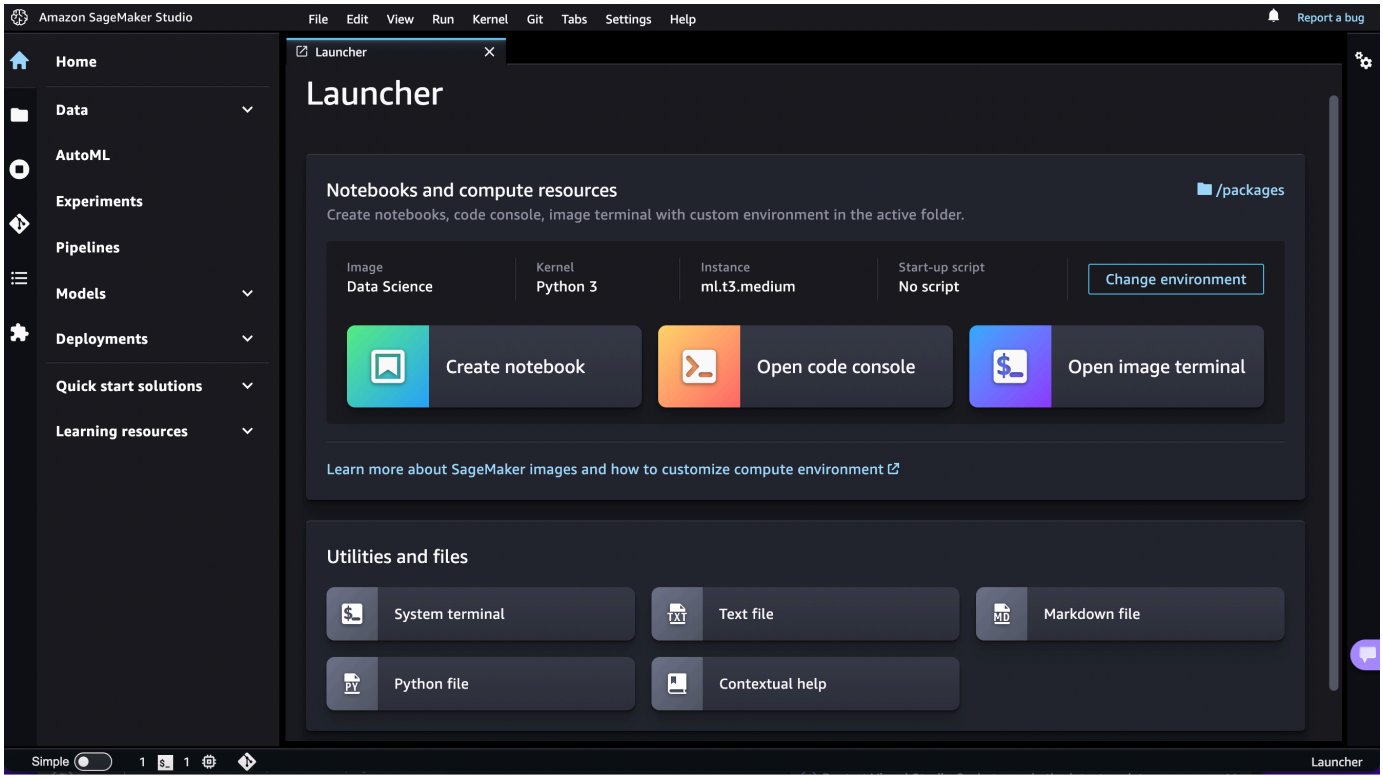
> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Setelah Anda membuat SageMaker gambar kustom dan melampirkannya ke domain atau ruang bersama, gambar kustom dan kernel muncul di penyeleksi di kotak dialog Ubah lingkungan dari Studio Classic Launcher.

Untuk meluncurkan dan memilih gambar dan kernel kustom Anda

1. Di Amazon SageMaker Studio Classic, buka Launcher. Untuk membuka Launcher, pilih Amazon SageMaker Studio Classic di kiri atas antarmuka Studio Classic atau gunakan pintasan `Ctrl + Shift + L` keyboard.

   Untuk mempelajari semua cara yang tersedia untuk membuka Peluncur, lihat Gunakan Amazon SageMaker Studio Classic Launcher



2. Di Peluncur, di bagian Notebook dan sumber daya komputasi, pilih Ubah lingkungan.
3. Dalam dialog Ubah lingkungan, gunakan menu tarik-turun untuk memilih Gambar Anda dari bagian Gambar Kustom, dan Kernel Anda, lalu pilih Pilih.
4. Di Launcher, pilih Buat notebook atau Buka terminal gambar. Notebook atau terminal Anda diluncurkan di gambar dan kernel kustom yang dipilih.

Untuk mengubah gambar atau kernel Anda di buku catatan terbuka, lihatMengubah Gambar atau Kernel.

> **Note**
>
> Jika Anda mengalami kesalahan saat meluncurkan gambar, periksa CloudWatch log Amazon Anda. Nama grup log adalah`/aws/sagemaker/studio`. Nama aliran log adalah`$domainID/$userProfileName/KernelGateway/$appName`.

Pembersihan sumber daya

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Bagian berikut menunjukkan cara membersihkan sumber daya yang Anda buat di bagian sebelumnya dari SageMaker konsol atauAWS CLI. Anda melakukan langkah-langkah berikut untuk membersihkan sumber daya:

- Lepaskan versi gambar dan gambar dari domain Anda.
- Hapus gambar, versi gambar, dan konfigurasi gambar aplikasi.
- Hapus gambar kontainer dan repositori dari Amazon ECR. Untuk informasi selengkapnya, lihat Menghapus repositori.

Bersihkan sumber daya dari SageMaker konsol

Bagian berikut menunjukkan cara membersihkan sumber daya dari SageMaker konsol.

Saat Anda melepaskan gambar dari domain, semua versi gambar akan terlepas. Ketika gambar terlepas, semua pengguna domain kehilangan akses ke versi gambar. Notebook yang berjalan yang memiliki sesi kernel pada versi gambar saat versi terlepas, terus berjalan. Ketika notebook dihentikan atau kernel dimatikan, versi gambar menjadi tidak tersedia.

Untuk melepaskan gambar

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Gambar.
4. Di bawah gambar Custom SageMaker Studio Classic yang dilampirkan ke domain, pilih gambar dan kemudian pilih Lepaskan.
5. (Opsional) Untuk menghapus gambar dan semua versi dari SageMaker, pilih Juga hapus gambar yang dipilih... . Ini tidak menghapus gambar kontainer terkait dari Amazon ECR.

6. Pilih Lepaskan.

Membersihkan sumber daya dari AWS CLI

Bagian berikut menunjukkan cara membersihkan sumber daya dariAWS CLI.

Untuk membersihkan sumber daya

1. Lepaskan versi gambar dan gambar dari domain Anda dengan meneruskan daftar gambar kustom kosong ke domain. Buka `default-user-settings.json` file yang Anda buatLampirkan SageMaker gambar ke domain Anda saat ini. Untuk melepaskan versi gambar dan gambar dari ruang bersama, buka `default-space-settings.json` file.
2. Hapus gambar kustom dan kemudian simpan file.

```
"DefaultUserSettings": {
  "KernelGatewayAppSettings": {
     "CustomImages": [
     ],
     ...
  },
  ...
}
```

3. Gunakan ID domain dan file pengaturan pengguna default untuk memperbarui domain Anda. Untuk memperbarui ruang bersama Anda, gunakan file pengaturan ruang default.

```
aws sagemaker update-domain \
    --domain-id <d-xxxxxxxxxxxx> \
    --cli-input-json file://default-user-settings.json
```

Responsnya akan terlihat mirip dengan yang berikut ini.

```
{
    "DomainArn": "arn:aws:sagemaker:us-east-2:acct-id:domain/d-xxxxxxxxxxxx"
}
```

4. Hapus konfigurasi gambar aplikasi.

```
aws sagemaker delete-app-image-config \
    --app-image-config-name custom-image-config
```

5. Hapus SageMaker gambar, yang juga menghapus semua versi gambar. Gambar kontainer di ECR yang diwakili oleh versi gambar tidak dihapus.

```
aws sagemaker delete-image \
    --image-name custom-image
```

Menggunakan konfigurasi siklus hidup dengan Amazon Studio Classic SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Konfigurasi siklus hidup adalah skrip shell yang dipicu oleh peristiwa siklus hidup Amazon SageMaker Studio Classic, seperti memulai notebook Studio Classic baru. Anda dapat menggunakan konfigurasi siklus hidup untuk mengotomatiskan penyesuaian lingkungan Studio Classic Anda. Kustomisasi ini termasuk menginstal paket kustom, mengkonfigurasi ekstensi notebook, preloading dataset, dan menyiapkan repositori kode sumber.

Menggunakan konfigurasi siklus hidup memberi Anda fleksibilitas dan kontrol untuk mengonfigurasi Studio Classic untuk memenuhi kebutuhan spesifik Anda. Misalnya, Anda dapat membuat kumpulan minimal gambar kontainer dasar dengan paket dan pustaka yang paling umum digunakan, lalu menggunakan konfigurasi siklus hidup untuk menginstal paket tambahan untuk kasus penggunaan tertentu di seluruh tim ilmu data dan pembelajaran mesin Anda.

Misalnya skrip konfigurasi siklus hidup, lihat repositori contoh Konfigurasi Siklus Hidup Studio Classic. GitHub Untuk blog tentang penerapan konfigurasi siklus hidup, lihat Menyesuaikan Amazon SageMaker Studio Classic menggunakan Konfigurasi Siklus Hidup.

> **Note**
>
> Setiap skrip memiliki batas 16384 karakter.

Topik

- Membuat dan mengaitkan konfigurasi siklus hidup
- Tetapkan konfigurasi siklus hidup default
- Debug konfigurasi siklus hidup
- Perbarui dan lepaskan konfigurasi siklus hidup

Membuat dan mengaitkan konfigurasi siklus hidup

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Amazon SageMaker menyediakan aplikasi interaktif yang memungkinkan antarmuka visual Studio Classic, pembuatan kode, dan pengalaman menjalankan. Seri ini menunjukkan cara membuat konfigurasi siklus hidup dan mengaitkannya dengan SageMaker domain.

Jenis aplikasi dapat berupa `JupyterServer` atau`KernelGateway`.

- **`JupyterServer`**aplikasi: Jenis aplikasi ini memungkinkan akses ke antarmuka visual untuk Studio Classic. Setiap pengguna dan ruang bersama di Studio Classic mendapatkan JupyterServer aplikasinya sendiri.
- **`KernelGateway`**aplikasi: Jenis aplikasi ini memungkinkan akses ke lingkungan menjalankan kode dan kernel untuk notebook dan terminal Studio Classic Anda. Untuk informasi lebih lanjut, lihat Jupyter Kernel Gateway.

Untuk informasi selengkapnya tentang arsitektur Studio Classic dan aplikasi Studio Classic, lihat Menggunakan Notebook Amazon SageMaker Studio Classic.

Topik

- Buat konfigurasi siklus hidup dari AWS CLI
- Buat konfigurasi siklus hidup dari konsol SageMaker

Buat konfigurasi siklus hidup dari AWS CLI

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara membuat konfigurasi siklus hidup menggunakan untuk mengotomatiskan penyesuaian AWS CLI untuk lingkungan Studio Classic Anda.

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensyal, lihat Memahami dan mendapatkan kredensyal Anda AWS.
- Onboard ke SageMaker domain dengan mengikuti langkah-langkah dilkhtisar SageMaker Domain Amazon.

Langkah 1: Buat konfigurasi siklus hidup

Prosedur berikut menunjukkan cara membuat skrip konfigurasi siklus hidup yang mencetak. `Hello World`

> **Note**
>
> Setiap skrip dapat memiliki hingga 16.384 karakter.

1. Dari mesin lokal Anda, buat file bernama `my-script.sh` dengan konten berikut.

```
#!/bin/bash
set -eux
echo 'Hello World!'
```

2. Konversikan `my-script.sh` file Anda menjadi format base64. Persyaratan ini mencegah kesalahan yang terjadi dari spasi dan pengkodean jeda baris.

```
LCC_CONTENT=`openssl base64 -A -in my-script.sh`
```

3. Buat konfigurasi siklus hidup untuk digunakan dengan Studio Classic. Perintah berikut membuat konfigurasi siklus hidup yang berjalan saat Anda meluncurkan aplikasi terkait`KernelGateway`.

```
aws sagemaker create-studio-lifecycle-config \
--region region \
--studio-lifecycle-config-name my-studio-lcc \
--studio-lifecycle-config-content $LCC_CONTENT \
--studio-lifecycle-config-app-type KernelGateway
```

   Perhatikan ARN dari konfigurasi siklus hidup yang baru dibuat yang dikembalikan. ARN ini diperlukan untuk melampirkan konfigurasi siklus hidup ke aplikasi Anda.

Langkah 2: Lampirkan konfigurasi siklus hidup ke domain, profil pengguna, atau ruang bersama

Untuk melampirkan konfigurasi siklus hidup, Anda harus memperbarui `UserSettings` untuk domain atau profil pengguna, atau `SpaceSettings` untuk ruang bersama. Skrip konfigurasi siklus hidup yang terkait pada tingkat domain diwarisi oleh semua pengguna. Namun, skrip yang terkait pada tingkat profil pengguna dicakup oleh pengguna tertentu, sementara skrip yang terkait pada tingkat ruang bersama dicakup ke ruang bersama.

Contoh berikut menunjukkan cara membuat profil pengguna baru dengan konfigurasi siklus hidup terlampir. Anda juga dapat membuat domain atau ruang baru dengan konfigurasi siklus hidup yang dilampirkan menggunakan perintah create-domain dan create-space, masing-masing.

Tambahkan ARN konfigurasi siklus hidup dari langkah sebelumnya ke pengaturan untuk jenis aplikasi yang sesuai. Misalnya, letakkan `JupyterServerAppSettings` di pengguna. Anda dapat menambahkan beberapa konfigurasi siklus hidup secara bersamaan dengan meneruskan daftar konfigurasi siklus hidup. Ketika pengguna meluncurkan JupyterServer aplikasi denganAWS CLI, mereka dapat meneruskan konfigurasi siklus hidup untuk digunakan alih-alih default. Konfigurasi siklus hidup yang dilewati pengguna harus termasuk dalam daftar konfigurasi siklus hidup. `JupyterServerAppSettings`

```
# Create a new UserProfile
aws sagemaker create-user-profile --domain-id domain-id \
```

```
--user-profile-name user-profile-name \
--region region \
--user-settings '{
"JupyterServerAppSettings": {
  "LifecycleConfigArns":
    [lifecycle-configuration-arn-list]
  }
}'
```

Contoh berikut menunjukkan cara memperbarui ruang bersama yang ada untuk melampirkan konfigurasi siklus hidup. Anda juga dapat memperbarui domain atau profil pengguna yang ada dengan konfigurasi siklus hidup yang dilampirkan menggunakan update-domain atau perintah. update-user-profile Saat memperbarui daftar konfigurasi siklus hidup yang dilampirkan, Anda harus meneruskan semua konfigurasi siklus hidup sebagai bagian dari daftar. Jika konfigurasi siklus hidup bukan bagian dari daftar ini, itu tidak akan dilampirkan ke aplikasi.

```
aws sagemaker update-space --domain-id domain-id \
--space-name space-name \
--region region \
--space-settings '{
"JupyterServerAppSettings": {
  "LifecycleConfigArns":
    [lifecycle-configuration-arn-list]
  }
}'
```

Untuk informasi tentang menyetel konfigurasi siklus hidup default untuk sumber daya, lihat. Tetapkan konfigurasi siklus hidup default

Langkah 3: Luncurkan aplikasi dengan konfigurasi siklus hidup

Setelah Anda melampirkan konfigurasi siklus hidup ke domain, profil pengguna, atau spasi, pengguna dapat memilihnya saat meluncurkan aplikasi dengan file. AWS CLI Bagian ini menjelaskan cara meluncurkan aplikasi dengan konfigurasi siklus hidup terlampir. Untuk informasi tentang mengubah konfigurasi siklus hidup default setelah meluncurkan JupyterServer aplikasi, lihat. Tetapkan konfigurasi siklus hidup default

Luncurkan jenis aplikasi yang diinginkan menggunakan `create-app` perintah dan tentukan konfigurasi siklus hidup ARN dalam argumen. `resource-spec`

- Contoh berikut menunjukkan cara membuat `JupyterServer` aplikasi dengan konfigurasi siklus hidup terkait. Saat membuat`JupyterServer`, `app-name` harus`default`. Konfigurasi siklus hidup ARN yang diteruskan sebagai bagian dari `resource-spec` parameter harus menjadi bagian dari daftar ARN konfigurasi siklus hidup yang ditentukan untuk domain atau profil pengguna Anda, atau `UserSettings` untuk ruang bersama. `SpaceSettings`

```
aws sagemaker create-app --domain-id domain-id \
--region region \
--user-profile-name user-profile-name \
--app-type JupyterServer \
--resource-spec LifecycleConfigArn=lifecycle-configuration-arn \
--app-name default
```

- Contoh berikut menunjukkan cara membuat `KernelGateway` aplikasi dengan konfigurasi siklus hidup terkait.

```
aws sagemaker create-app --domain-id domain-id \
--region region \
--user-profile-name user-profile-name \
--app-type KernelGateway \
--resource-spec LifecycleConfigArn=lifecycle-configuration-
arn,SageMakerImageArn=sagemaker-image-arn,InstanceType=instance-type \
--app-name app-name
```

Buat konfigurasi siklus hidup dari konsol SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara membuat konfigurasi siklus hidup dari SageMaker konsol Amazon untuk mengotomatiskan penyesuaian lingkungan Studio Classic Anda.

Prasyarat

Sebelum Anda dapat memulai tutorial ini, lengkapi prasyarat berikut:

- Onboard ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Onboard ke Amazon SageMaker Studio Classic.

Langkah 1: Buat konfigurasi siklus hidup baru

Anda dapat membuat konfigurasi siklus hidup dengan memasukkan skrip dari konsol Amazon SageMaker.

> **Note**
>
> Setiap skrip dapat memiliki hingga 16.384 karakter.

Prosedur berikut menunjukkan cara membuat skrip konfigurasi siklus hidup yang mencetak. `Hello World`

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Konfigurasi Siklus Hidup.
4. Pilih tab Studio.
5. Pilih Buat konfigurasi.
6. Di bawah Pilih jenis konfigurasi, pilih jenis aplikasi yang harus dilampirkan konfigurasi siklus hidup. Untuk informasi selengkapnya tentang memilih aplikasi mana yang akan dilampirkan konfigurasi siklus hidup, lihat. Tetapkan konfigurasi siklus hidup default
7. Pilih Berikutnya.
8. Di bagian yang disebut Pengaturan konfigurasi, masukkan nama untuk konfigurasi siklus hidup Anda.
9. Di bagian Skrip, masukkan konten berikut.

```
#!/bin/bash
set -eux
echo 'Hello World!'
```

10. (Opsional) Buat tag untuk konfigurasi siklus hidup Anda.
11. Pilih Kirim.

Langkah 2: Lampirkan konfigurasi siklus hidup ke domain atau profil pengguna

Skrip konfigurasi siklus hidup yang terkait pada tingkat domain diwarisi oleh semua pengguna. Namun, skrip yang terkait pada tingkat profil pengguna dicakup oleh pengguna tertentu.

Anda dapat melampirkan beberapa konfigurasi siklus hidup ke domain atau profil pengguna untuk keduanya JupyterServer dan aplikasi. KernelGateway

> **Note**
>
> Untuk melampirkan konfigurasi siklus hidup ke ruang bersama, Anda harus menggunakan. AWS CLI Untuk informasi selengkapnya, lihat Buat konfigurasi siklus hidup dari AWS CLI.

Bagian berikut menunjukkan cara melampirkan konfigurasi siklus hidup ke domain atau profil pengguna Anda.

Lampirkan ke domain

Berikut ini menunjukkan cara melampirkan konfigurasi siklus hidup ke domain yang ada dari konsol. SageMaker

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain untuk melampirkan konfigurasi siklus hidup.
5. Dari detail Domain, pilih tab Lingkungan.
6. Di bawah Konfigurasi Siklus Hidup untuk aplikasi Studio pribadi, pilih Lampirkan.
7. Di bawah Sumber, pilih Konfigurasi yang ada.
8. Di bawah Konfigurasi siklus hidup Studio, pilih konfigurasi siklus hidup yang Anda buat pada langkah sebelumnya.
9. Pilih Lampirkan ke domain.

Lampirkan ke profil pengguna Anda

Berikut ini menunjukkan cara melampirkan konfigurasi siklus hidup ke profil pengguna yang ada.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang berisi profil pengguna untuk melampirkan konfigurasi siklus hidup.
5. Di bawah Profil pengguna, pilih profil pengguna.
6. Dari halaman Detail Pengguna, pilih Edit.
7. Di navigasi kiri, pilih Pengaturan studio.
8. Di bawah Konfigurasi Siklus Hidup yang dilampirkan ke pengguna, pilih Lampirkan.
9. Di bawah Sumber, pilih Konfigurasi yang ada.
10. Di bawah Konfigurasi siklus hidup Studio, pilih konfigurasi siklus hidup yang Anda buat pada langkah sebelumnya.
11. Pilih Lampirkan ke profil pengguna.

Langkah 3: Luncurkan aplikasi dengan konfigurasi siklus hidup

Setelah melampirkan konfigurasi siklus hidup ke domain atau profil pengguna, Anda dapat meluncurkan aplikasi dengan konfigurasi siklus hidup terlampir tersebut. Memilih konfigurasi siklus hidup mana yang akan diluncurkan bergantung pada jenis aplikasi.

- JupyterServer: Saat meluncurkan JupyterServer aplikasi dari konsol, SageMaker selalu gunakan konfigurasi siklus hidup default. Anda tidak dapat menggunakan konfigurasi siklus hidup yang berbeda saat meluncurkan dari konsol. Untuk informasi tentang mengubah konfigurasi siklus hidup default setelah meluncurkan JupyterServer aplikasi, lihat. Tetapkan konfigurasi siklus hidup default

  Untuk memilih konfigurasi siklus hidup terlampir yang berbeda, Anda harus meluncurkan dengan file. AWS CLI Untuk informasi selengkapnya tentang meluncurkan JupyterServer aplikasi dengan konfigurasi siklus hidup terlampir dariAWS CLI, lihat. Buat konfigurasi siklus hidup dari AWS CLI
- KernelGateway: Anda dapat memilih salah satu konfigurasi siklus hidup terlampir saat meluncurkan KernelGateway aplikasi menggunakan Studio Classic Launcher.

Prosedur berikut menjelaskan cara meluncurkan KernelGateway aplikasi dengan konfigurasi siklus hidup terlampir dari konsol. SageMaker

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.

2. Luncurkan Studio Klasik. Untuk informasi selengkapnya, lihat Luncurkan Amazon SageMaker Studio Classic.
3. Di UI Studio Classic, buka Studio Classic Launcher. Untuk informasi selengkapnya, lihat Gunakan Amazon SageMaker Studio Classic Launcher.
4. Di Studio Classic Launcher, navigasikan ke bagian Notebook dan compute resources.
5. Klik tombol Ubah lingkungan.
6. Pada dialog Ubah lingkungan, gunakan menu tarik-turun untuk memilih Image, Kernel, Instance type, dan skrip Start-up. Jika tidak ada konfigurasi siklus hidup default, nilai skrip Start-up default ke. `No script` Jika tidak, nilai skrip Start-up adalah konfigurasi siklus hidup default Anda. Setelah Anda memilih konfigurasi siklus hidup, Anda dapat melihat seluruh skrip.
7. Klik Pilih.
8. Kembali ke Peluncur, klik Buat buku catatan untuk meluncurkan kernel notebook baru dengan gambar dan konfigurasi siklus hidup yang Anda pilih.

Langkah 4: Lihat log untuk konfigurasi siklus hidup

Anda dapat melihat log untuk konfigurasi siklus hidup Anda setelah dilampirkan ke domain atau profil pengguna.

1. Pertama, berikan akses CloudWatch untuk peran AWS Identity and Access Management (IAM) Anda. Tambahkan izin baca untuk grup log dan aliran log berikut.

   - Grup log: `/aws/sagemaker/studio`
   - Aliran log: *`domain`*`/`*`user-profile`*`/`*`app-type`*`/`*`app-name`*`/LifecycleConfigOnStart`

   Untuk informasi tentang menambahkan izin, lihat Mengaktifkan logging dari layanan tertentu AWS.
2. Dari dalam Studio Classic, arahkan ke

   ikon Running Terminals dan Kernels untuk memantau konfigurasi siklus hidup Anda.
3. Pilih aplikasi dari daftar aplikasi yang sedang berjalan. Aplikasi dengan konfigurasi siklus hidup terlampir memiliki ikon indikator terlampir.

4. Pilih ikon indikator untuk aplikasi Anda. Ini membuka panel baru yang mencantumkan konfigurasi siklus hidup.
5. Dari panel baru, pilih`View logs`. Ini membuka tab baru yang menampilkan log.

Tetapkan konfigurasi siklus hidup default

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Meskipun Anda dapat melampirkan beberapa skrip konfigurasi siklus hidup ke satu sumber daya, Anda hanya dapat mengatur satu konfigurasi siklus hidup default untuk masing-masing atau aplikasi. JupyterServer KernelGateway Perilaku konfigurasi siklus hidup default bergantung pada apakah konfigurasi tersebut disetel untuk JupyterServer atau KernelGateway aplikasi.

- JupyterServer apps: Saat ditetapkan sebagai skrip konfigurasi siklus hidup default untuk JupyterServer aplikasi, skrip konfigurasi siklus hidup berjalan secara otomatis saat pengguna masuk ke Studio Classic untuk pertama kalinya atau memulai ulang Studio Classic. Gunakan konfigurasi siklus hidup default ini untuk mengotomatiskan tindakan penyiapan satu kali untuk lingkungan developer Studio Classic, seperti menginstal ekstensi notebook atau menyiapkan repo. GitHub Untuk contoh ini, lihat Menyesuaikan Amazon SageMaker Studio menggunakan Konfigurasi Siklus Hidup.
- KernelGateway apps: Bila disetel sebagai skrip konfigurasi siklus hidup default untuk KernelGateway aplikasi, konfigurasi siklus hidup dipilih secara default di peluncur Studio Classic. Pengguna dapat meluncurkan notebook atau terminal dengan skrip default yang dipilih, atau mereka dapat memilih yang berbeda dari daftar konfigurasi siklus hidup.

SageMaker mendukung pengaturan konfigurasi siklus hidup default untuk sumber daya berikut:

- Domain
- Profil pengguna
- Ruang bersama

Meskipun domain dan profil pengguna mendukung pengaturan konfigurasi siklus hidup default dari SageMaker konsol Amazon danAWS Command Line Interface, spasi bersama hanya mendukung pengaturan konfigurasi siklus hidup default dari. AWS CLI

Anda dapat mengatur konfigurasi siklus hidup sebagai default saat membuat sumber daya baru atau memperbarui sumber daya yang ada. Topik berikut menunjukkan cara menyetel konfigurasi siklus hidup default menggunakan SageMaker konsol dan. AWS CLI

Warisan konfigurasi siklus hidup default

Konfigurasi siklus hidup default yang ditetapkan pada tingkat domain diwarisi oleh semua pengguna dan spasi bersama. Konfigurasi siklus hidup default yang ditetapkan pada pengguna dan tingkat ruang bersama hanya dicakup oleh pengguna atau ruang bersama tersebut. Default pengguna dan ruang mengganti default yang ditetapkan pada tingkat domain.

Konfigurasi KernelGateway siklus hidup default yang ditetapkan untuk domain berlaku untuk semua KernelGateway aplikasi yang diluncurkan di domain. Kecuali pengguna memilih konfigurasi siklus hidup yang berbeda dari daftar yang disajikan di peluncur Studio Classic, konfigurasi siklus hidup default akan digunakan. Skrip default juga berjalan `No Script` jika dipilih oleh pengguna. Untuk informasi selengkapnya tentang memilih skrip, lihatLangkah 3: Luncurkan aplikasi dengan konfigurasi siklus hidup.

Topik

Tetapkan default dari AWS CLI

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat mengatur skrip konfigurasi siklus hidup default dari sumber daya AWS CLI berikut:

- Domain

- Profil pengguna
- Ruang bersama

Bagian berikut menguraikan cara mengatur skrip konfigurasi siklus hidup default dari. AWS CLI

Topik

- Prasyarat
- Tetapkan konfigurasi siklus hidup default saat membuat sumber daya baru
- Menetapkan konfigurasi siklus hidup default untuk sumber daya yang ada

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensyal, lihat Memahami dan mendapatkan kredensyal Anda AWS.
- Onboard ke SageMaker domain dengan mengikuti langkah-langkah diIkhtisar SageMaker Domain Amazon.
- Buat konfigurasi siklus hidup mengikuti langkah-langkah di. Membuat dan mengaitkan konfigurasi siklus hidup

Tetapkan konfigurasi siklus hidup default saat membuat sumber daya baru

Untuk menetapkan konfigurasi siklus hidup default saat membuat domain, profil pengguna, atau spasi baru, teruskan ARN konfigurasi siklus hidup yang Anda buat sebelumnya sebagai bagian dari salah satu perintah berikut: AWS CLI

- create-user-profile
- buat-domain
- buat-ruang

Anda harus meneruskan ARN konfigurasi siklus hidup untuk nilai berikut dalam pengaturan atau default: KernelGateway JupyterServer

- `DefaultResourceSpec`: `LifecycleConfigArn` - Ini menentukan konfigurasi siklus hidup default untuk jenis aplikasi.

- `LifecycleConfigArns`- Ini adalah daftar semua konfigurasi siklus hidup yang dilampirkan ke jenis aplikasi. Konfigurasi siklus hidup default juga harus menjadi bagian dari daftar ini.

Misalnya, panggilan API berikut membuat profil pengguna baru dengan konfigurasi siklus hidup default.

```
aws sagemaker create-user-profile --domain-id domain-id \
--user-profile-name user-profile-name \
--region region \
--user-settings '{
"KernelGatewayAppSettings": {
    "DefaultResourceSpec": {
            "InstanceType": "ml.t3.medium",
            "LifecycleConfigArn": "lifecycle-configuration-arn"
        },
    "LifecycleConfigArns": [lifecycle-configuration-arn-list]
  }
}'
```

Menetapkan konfigurasi siklus hidup default untuk sumber daya yang ada

Untuk menyetel atau memperbarui konfigurasi siklus hidup default untuk sumber daya yang ada, teruskan ARN konfigurasi siklus hidup yang Anda buat sebelumnya sebagai bagian dari salah satu perintah berikut: AWS CLI

- update-user-profile
- pembaruan-domain
- ruang pembaruan-

Anda harus meneruskan ARN konfigurasi siklus hidup untuk nilai berikut dalam pengaturan atau default: KernelGateway JupyterServer

- `DefaultResourceSpec`: `LifecycleConfigArn` - Ini menentukan konfigurasi siklus hidup default untuk jenis aplikasi.

- `LifecycleConfigArns`- Ini adalah daftar semua konfigurasi siklus hidup yang dilampirkan ke jenis aplikasi. Konfigurasi siklus hidup default juga harus menjadi bagian dari daftar ini.

Misalnya, panggilan API berikut memperbarui profil pengguna dengan konfigurasi siklus hidup default.

```
aws sagemaker update-user-profile --domain-id domain-id \
--user-profile-name user-profile-name \
--region region \
--user-settings '{
"KernelGatewayAppSettings": {
    "DefaultResourceSpec": {
            "InstanceType": "ml.t3.medium",
            "LifecycleConfigArn": "lifecycle-configuration-arn"
         },
    "LifecycleConfigArns": [lifecycle-configuration-arn-list]
  }
}'
```

Panggilan API berikut memperbarui domain untuk menetapkan konfigurasi siklus hidup default yang baru.

```
aws sagemaker update-domain --domain-id domain-id \
--region region \
--default-user-settings '{
"JupyterServerAppSettings": {
    "DefaultResourceSpec": {
            "InstanceType": "ml.t3.medium",
            "LifecycleConfigArn": "lifecycle-configuration-arn"
         },
    "LifecycleConfigArns": [lifecycle-configuration-arn-list]
  }
}'
```

Setel default dari konsol SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan

aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat menyetel skrip konfigurasi siklus hidup default dari SageMaker konsol untuk sumber daya berikut.

- Domain
- Profil pengguna

Anda tidak dapat menyetel skrip konfigurasi siklus hidup default untuk spasi bersama dari konsol. SageMaker Untuk informasi tentang menyetel default untuk spasi bersama, lihat. Tetapkan default dari AWS CLI

Bagian berikut menguraikan cara mengatur skrip konfigurasi siklus hidup default dari konsol. SageMaker

Topik

- Prasyarat
- Menetapkan konfigurasi siklus hidup default untuk domain
- Menetapkan konfigurasi siklus hidup default untuk profil pengguna

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Onboard ke SageMaker domain dengan mengikuti langkah-langkah diIkhtisar SageMaker Domain Amazon.
- Buat konfigurasi siklus hidup mengikuti langkah-langkah di. Membuat dan mengaitkan konfigurasi siklus hidup

Menetapkan konfigurasi siklus hidup default untuk domain

Prosedur berikut menunjukkan cara menyetel konfigurasi siklus hidup default untuk domain dari konsol. SageMaker

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.

2. Dari daftar domain, pilih nama domain untuk mengatur konfigurasi siklus hidup default.
3. Dari halaman Detail domain, pilih tab Lingkungan.
4. Di bawah Konfigurasi Siklus Hidup untuk aplikasi Studio pribadi, pilih konfigurasi siklus hidup yang ingin Anda tetapkan sebagai default untuk domain. Anda dapat mengatur default yang berbeda untuk JupyterServer dan aplikasi. KernelGateway
5. Pilih Tetapkan sebagai default. Ini membuka jendela pop up yang mencantumkan default saat ini untuk JupyterServer dan aplikasi. KernelGateway
6. Pilih Set as default untuk mengatur konfigurasi siklus hidup sebagai default untuk jenis aplikasi masing-masing.

Menetapkan konfigurasi siklus hidup default untuk profil pengguna

Prosedur berikut menunjukkan cara menyetel konfigurasi siklus hidup default untuk profil pengguna dari konsol. SageMaker

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Dari daftar domain, pilih nama domain yang berisi profil pengguna yang ingin Anda atur konfigurasi siklus hidup default.
3. Dari halaman Detail domain, pilih tab Profil pengguna.
4. Pilih nama profil pengguna untuk mengatur konfigurasi siklus hidup default. Ini membuka halaman Detail Pengguna.
5. Dari halaman Detail Pengguna, pilih Edit. Ini membuka halaman Edit profil pengguna.
6. Dari halaman Edit profil pengguna, pilih Pengaturan Step 2 Studio.
7. Di bawah Konfigurasi Siklus Hidup yang dilampirkan ke pengguna, pilih konfigurasi siklus hidup yang ingin Anda tetapkan sebagai default untuk profil pengguna. Anda dapat mengatur default yang berbeda untuk JupyterServer dan aplikasi. KernelGateway
8. Pilih Tetapkan sebagai default. Ini membuka jendela pop up yang mencantumkan default saat ini untuk JupyterServer dan aplikasi. KernelGateway
9. Pilih Set as default untuk mengatur konfigurasi siklus hidup sebagai default untuk jenis aplikasi masing-masing.

Debug konfigurasi siklus hidup

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara mendapatkan informasi tentang dan men-debug konfigurasi siklus hidup Anda.

Topik

Verifikasi proses konfigurasi siklus hidup dari Log CloudWatch

Konfigurasi siklus hidup hanya log dan. `STDOUT` `STDERR`

`STDOUT`adalah output default untuk skrip bash. Anda dapat menulis ke `STDERR` dengan menambahkan `>&2` ke akhir perintah bash. Misalnya, `echo 'hello'>&2`.

Log untuk konfigurasi siklus hidup Anda dipublikasikan ke Anda menggunakan Akun AWS Amazon. CloudWatch Log ini dapat ditemukan di aliran `/aws/sagemaker/studio` log di CloudWatch konsol.

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pilih Log dari sisi kiri. Dari menu tarik-turun, pilih Grup log.
3. Pada halaman Grup log, cari`aws/sagemaker/studio`.
4. Pilih grup log .
5. Pada halaman Detail grup log, pilih tab Aliran log.
6. Untuk menemukan log untuk aplikasi tertentu, cari aliran log menggunakan format berikut:

```
domain-id/user-profile-name/app-type/app-name
```

Misalnya, untuk menemukan log konfigurasi siklus hidup untuk domaind-m85lcu8vbqmz, profil pengguna, jenis aplikasi`i-sonic-js`, `JupyterServer` dan nama aplikasi`test-lcc-echo`, gunakan string pencarian berikut:

```
d-m85lcu8vbqmz/i-sonic-js/JupyterServer/test-lcc-echo
```

7. Pilih aliran log yang ditambahkan `LifecycleConfigOnStart` untuk melihat log eksekusi skrip.

JupyterServer kegagalan aplikasi

Jika JupyterServer aplikasi Anda mogok karena masalah dengan konfigurasi siklus hidup terlampir, Studio Classic akan menampilkan pesan galat berikut di layar startup Studio Classic.

```
Failed to create SageMaker Studio due to start-up script failure
```

Pilih `View script logs` tautan untuk melihat CloudWatch log untuk JupyterServer aplikasi Anda.

Jika konfigurasi siklus hidup yang salah ditentukan dalam domain, profil pengguna, atau ruang bersama, Studio Classic terus menggunakan konfigurasi siklus hidup bahkan setelah memulai ulang Studio Classic. `DefaultResourceSpec`

Untuk mengatasi kesalahan ini, ikuti langkah-langkah Tetapkan konfigurasi siklus hidup default untuk menghapus skrip konfigurasi siklus hidup dari `DefaultResourceSpec` atau pilih skrip lain sebagai default. Kemudian luncurkan JupyterServer aplikasi baru.

KernelGateway kegagalan aplikasi

Jika KernelGateway aplikasi Anda mogok karena masalah dengan konfigurasi siklus hidup terlampir, Studio Classic akan menampilkan pesan galat di Notebook Studio Classic Anda.

Pilih `View script logs` untuk melihat CloudWatch log untuk KernelGateway aplikasi Anda.

Dalam hal ini, konfigurasi siklus hidup Anda ditentukan di Studio Classic Launcher saat meluncurkan Notebook Studio Classic baru.

Untuk mengatasi kesalahan ini, gunakan peluncur Studio Classic untuk memilih konfigurasi siklus hidup yang berbeda atau pilih. `No script`

> **Note**
>
> Konfigurasi KernelGateway siklus hidup default yang ditentukan dalam `DefaultResourceSpec` berlaku untuk semua KernelGateway gambar di domain, profil pengguna, atau ruang bersama kecuali pengguna memilih skrip yang berbeda dari daftar yang disajikan di peluncur Studio Classic. Skrip default juga berjalan `No Script` jika dipilih oleh pengguna. Untuk informasi selengkapnya tentang memilih skrip, lihatLangkah 3: Luncurkan aplikasi dengan konfigurasi siklus hidup.

Batas waktu konfigurasi siklus hidup

Ada batasan batas waktu konfigurasi siklus hidup 5 menit. Jika skrip konfigurasi siklus hidup membutuhkan waktu lebih dari 5 menit untuk dijalankan, Studio Classic akan memunculkan kesalahan.

Untuk mengatasi kesalahan ini, pastikan skrip konfigurasi siklus hidup Anda selesai dalam waktu kurang dari 5 menit.

Untuk membantu mengurangi waktu berjalan skrip, coba yang berikut ini:

- Kurangi langkah-langkah yang diperlukan. Misalnya, batasi lingkungan conda mana untuk menginstal paket besar.
- Jalankan tugas dalam proses paralel.
- Gunakan nohup perintah dalam skrip Anda untuk memastikan bahwa sinyal hangup diabaikan dan tidak menghentikan eksekusi skrip.

Perbarui dan lepaskan konfigurasi siklus hidup

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Skrip konfigurasi siklus hidup tidak dapat diubah setelah dibuat. Untuk memperbarui skrip Anda, Anda harus membuat skrip konfigurasi siklus hidup baru dan melampirkannya ke domain masing-masing, profil pengguna, atau ruang bersama. Untuk informasi selengkapnya tentang membuat dan melampirkan konfigurasi siklus hidup, lihat. Membuat dan mengaitkan konfigurasi siklus hidup

Topik berikut menunjukkan cara melepaskan konfigurasi siklus hidup menggunakan dan konsol. AWS CLI SageMaker

Topik

- Prasyarat
- Lepaskan menggunakan AWS CLI

Prasyarat

Sebelum melepaskan konfigurasi siklus hidup, Anda harus menyelesaikan prasyarat berikut.

- Agar berhasil melepaskan konfigurasi siklus hidup, tidak ada aplikasi yang berjalan yang dapat menggunakan konfigurasi siklus hidup. Anda harus terlebih dahulu mematikan aplikasi yang berjalan seperti yang ditunjukkan padaMatikan dan Perbarui Aplikasi SageMaker Studio Classic dan Studio Classic.

Lepaskan menggunakan AWS CLI

Untuk melepaskan konfigurasi siklus hidup menggunakanAWS CLI, hapus konfigurasi siklus hidup yang diinginkan dari daftar konfigurasi siklus hidup yang dilampirkan ke sumber daya dan teruskan daftar sebagai bagian dari perintah masing-masing:

- update-user-profile
- pembaruan-domain
- ruang pembaruan-

Misalnya, perintah berikut menghapus semua konfigurasi siklus hidup untuk KernelGateways dilampirkan ke domain.

```
aws sagemaker update-domain --domain-id domain-id \
--region region \
--default-user-settings '{
```

```
"KernelGatewayAppSettings": {
  "LifecycleConfigArns":
    []
  }
}'
```

Lampirkan Repos Git yang Disarankan ke Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Amazon SageMaker Studio Classic menawarkan ekstensi Git bagi Anda untuk memasukkan URL repositori Git (repo), mengkloningnya ke lingkungan Anda, mendorong perubahan, dan melihat riwayat komit. Selain ekstensi Git ini, Anda juga dapat melampirkan URL repositori Git yang disarankan di SageMaker Domain Amazon atau tingkat profil pengguna. Kemudian, Anda dapat memilih URL repo dari daftar saran dan mengkloningnya ke lingkungan Anda menggunakan ekstensi Git di Studio Classic.

Topik berikut menunjukkan cara melampirkan URL repo Git ke Domain atau profil pengguna dari AWS CLI dan SageMaker konsol. Anda juga akan belajar cara melepaskan URL repositori ini.

Topik

- Lampirkan Repositori Git dari AWS CLI
- Lampirkan Repositori Git dari Konsol SageMaker
- Lepaskan Repo Git

Lampirkan Repositori Git dari AWS CLI

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan

aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara melampirkan URL repositori Git menggunakanAWS CLI, sehingga Amazon SageMaker Studio Classic secara otomatis menyarankannya untuk kloning. Setelah Anda melampirkan URL repositori Git, Anda dapat mengkloningnya dengan mengikuti langkah-langkah di. Mengkloning Repositori Git di Studio Classic SageMaker

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal Versi AWS CLI saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensyal, lihat Memahami dan mendapatkan kredensyal Anda AWS.
- Onboard ke SageMaker Domain Amazon. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.

Lampirkan repo Git ke Domain atau profil pengguna

URL repo Git yang terkait di tingkat Domain diwarisi oleh semua pengguna. Namun, URL repo Git yang terkait pada tingkat profil pengguna dicakup oleh pengguna tertentu. Anda dapat melampirkan beberapa URL repo Git ke Domain atau profil pengguna dengan meneruskan daftar URL repositori.

Bagian berikut menunjukkan cara melampirkan URL repo Git ke Domain dan profil pengguna Anda.

Lampirkan ke Domain

Perintah berikut melampirkan URL repo Git ke Domain yang ada.

```
aws sagemaker update-domain --region region --domain-id domain-id \
    --default-user-settings
 JupyterServerAppSettings={CodeRepositories=[{RepositoryUrl="repository"}]}
```

Lampirkan ke profil pengguna

Berikut ini menunjukkan cara melampirkan URL repo Git ke profil pengguna yang ada.

```
aws sagemaker update-user-profile --domain-id domain-id --user-profile-name user-name\
```

```
    --user-settings
JupyterServerAppSettings={CodeRepositories=[{RepositoryUrl="repository"}]}
```

Lampirkan Repositori Git dari Konsol SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara mengaitkan URL repositori Git dari SageMaker konsol Amazon untuk mengkloningnya di lingkungan Studio Classic Anda. Setelah Anda mengaitkan URL repositori Git, Anda dapat mengkloningnya dengan mengikuti langkah-langkahnya. Mengkloning Repositori Git di Studio Classic SageMaker

Prasyarat

Sebelum Anda dapat memulai tutorial ini, Anda harus onboard ke Amazon SageMaker Domain. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.

Lampirkan repo Git ke Domain atau profil pengguna

URL repo Git yang terkait di tingkat Domain diwarisi oleh semua pengguna. Namun, URL repo Git yang terkait pada tingkat profil pengguna dicakup oleh pengguna tertentu.

Bagian berikut menunjukkan cara melampirkan URL repo Git ke Domain dan profil pengguna.

Lampirkan ke Domain

Untuk melampirkan URL repo Git ke Domain yang ada

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih Domain untuk melampirkan repo Git ke.
5. Pada halaman Detail domain, pilih tab Lingkungan.
6. Pada repositori kode yang disarankan untuk tab domain, pilih Lampirkan.

7. Di bawah Sumber, masukkan URL repositori Git.
8. Pilih Lampirkan ke domain.

Lampirkan ke profil pengguna

Berikut ini menunjukkan cara melampirkan URL repositori Git ke profil pengguna yang ada.

Untuk melampirkan URL repositori Git ke profil pengguna

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih Domain yang menyertakan profil pengguna untuk melampirkan repo Git.
5. Pada halaman Detail domain, pilih tab Profil pengguna.
6. Pilih profil pengguna untuk melampirkan URL repo Git.
7. Pada halaman Detail pengguna, pilih Edit.
8. Pada halaman Pengaturan Studio, pilih Lampirkan dari repositori kode yang disarankan untuk bagian pengguna.
9. Di bawah Sumber, masukkan URL repositori Git.
10. Pilih Lampirkan ke pengguna.

Lepaskan Repo Git

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Panduan ini menunjukkan cara melepaskan URL repositori Git dari SageMaker Domain Amazon atau profil pengguna menggunakan konsol atau AWS CLI Amazon. SageMaker

Topik

- Lepaskan repo Git menggunakan AWS CLI

- Lepaskan repo Git menggunakan konsol SageMaker

Lepaskan repo Git menggunakan AWS CLI

Untuk melepaskan semua URL repo Git dari Domain atau profil pengguna, Anda harus meneruskan daftar kosong repositori kode. Daftar ini dilewatkan sebagai bagian dari `JupyterServerAppSettings` parameter dalam `update-user-profile` perintah `update-domain` atau. Untuk melepaskan hanya satu URL repo Git, masukkan daftar repositori kode tanpa URL repo Git yang diinginkan. Bagian ini menunjukkan cara melepaskan semua URL repo Git dari Domain atau profil pengguna menggunakan (). AWS Command Line Interface AWS CLI

Lepaskan dari Domain

Perintah berikut melepaskan semua URL repo Git dari Domain.

```
aws sagemaker update-domain --region region --domain-name domain-name \
    --domain-settings JupyterServerAppSettings={CodeRepositories=[]}
```

Lepaskan dari profil pengguna

Perintah berikut melepaskan semua URL repo Git dari profil pengguna.

```
aws sagemaker update-user-profile --domain-name domain-name --user-profile-name user-
name\
    --user-settings JupyterServerAppSettings={CodeRepositories=[]}
```

Lepaskan repo Git menggunakan konsol SageMaker

Bagian berikut menunjukkan cara melepaskan URL repo Git dari Domain atau profil pengguna menggunakan konsol. SageMaker

Lepaskan dari Domain

Gunakan langkah-langkah berikut untuk melepaskan URL repo Git dari Domain yang ada.

Untuk melepaskan URL repo Git dari Domain yang ada

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.

4. Pilih Domain dengan URL repo Git yang ingin Anda lepaskan.
5. Pada halaman Detail domain, pilih tab Lingkungan.
6. Pada repositori kode yang disarankan untuk tab domain, pilih URL repositori Git yang akan dilepas.
7. Pilih Lepaskan.
8. Dari jendela baru, pilih Lepaskan.

Lepaskan dari profil pengguna

Gunakan langkah-langkah berikut untuk melepaskan URL repo Git dari profil pengguna.

Untuk melepaskan URL repo Git dari profil pengguna

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pilih Domain yang menyertakan profil pengguna dengan URL repo Git yang ingin Anda lepaskan.
5. Pada halaman Detail domain, pilih tab Profil pengguna.
6. Pilih profil pengguna dengan URL repo Git yang ingin Anda lepaskan.
7. Pada halaman Detail pengguna, pilih Edit.
8. Pada halaman pengaturan Studio, pilih URL repo Git untuk melepaskan dari repositori kode yang disarankan untuk tab pengguna.
9. Pilih Lepaskan.
10. Dari jendela baru, pilih Lepaskan.

Lakukan Tugas Umum di Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Bagian berikut menjelaskan cara melakukan tugas umum di Amazon SageMaker Studio Classic. Untuk ikhtisar antarmuka Studio Classic, lihatIkhtisar UI Amazon SageMaker Studio Classic.

Topik

Unggah File ke SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Saat Anda onboard ke Amazon SageMaker Studio Classic, direktori home dibuat untuk Anda dalam volume Amazon Elastic File System (Amazon EFS) yang dibuat untuk tim Anda. Studio Classic hanya dapat membuka file yang telah diunggah ke direktori Anda. Browser file Studio Classic memetakan ke direktori home Anda.

> **Note**
>
> Studio Classic tidak mendukung pengunggahan folder. Meskipun Anda hanya dapat mengunggah file individual, Anda dapat mengunggah beberapa file secara bersamaan.

Untuk meng-upload file ke direktori home

1. Di sidebar kiri, pilih ikon File Browser
   ().
2. Di browser file, pilih ikon Unggah File
   ().
3. Pilih file yang ingin Anda unggah lalu pilih Buka.
4. Klik dua kali file untuk membuka file di tab baru di Studio Classic.

Mengkloning Repositori Git di Studio Classic SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Amazon SageMaker Studio Classic hanya dapat terhubung ke repositori Git lokal (repo). Ini berarti Anda harus mengkloning repo Git dari dalam Studio Classic untuk mengakses file di repo. Studio Classic menawarkan ekstensi Git bagi Anda untuk memasukkan URL repo Git, mengkloningnya ke lingkungan Anda, mendorong perubahan, dan melihat riwayat komit. Jika repo bersifat pribadi dan memerlukan kredensyal untuk mengakses, maka Anda diminta untuk memasukkan kredensyal pengguna Anda. Ini termasuk nama pengguna dan token akses pribadi Anda. Untuk informasi selengkapnya tentang token akses pribadi, lihat Mengelola token akses pribadi Anda.

Admin juga dapat melampirkan URL repositori Git yang disarankan di tingkat domain atau profil pengguna SageMaker Amazon. Pengguna kemudian dapat memilih URL repo dari daftar saran dan mengkloningnya ke Studio Classic. Untuk informasi lebih lanjut tentang melampirkan repo yang disarankan, lihat. Lampirkan Repos Git yang Disarankan ke Studio Classic

Prosedur berikut menunjukkan cara mengkloning GitHub repo dari Studio Classic.

Untuk mengkloning repo

1. Di sidebar kiri, pilih ikon Git
   ().
2. Pilih Clone a Repository. Ini membuka jendela baru.
3. Di jendela Clone Git Repository, masukkan URL dalam format berikut untuk repo Git yang ingin Anda kloning atau pilih repositori dari daftar repositori yang Disarankan.

   ```
   https://github.com/path-to-git-repo/repo.git
   ```
4. Jika Anda memasukkan URL repo Git secara manual, pilih Clone "***git-url***" dari menu dropdown.
5. Di bawah direktori Project untuk dikloning, masukkan jalur ke direktori lokal tempat Anda ingin mengkloning repo Git. Jika nilai ini dibiarkan kosong, Studio Classic mengkloning repo ke direktori JupyterLab root.
6. Pilih Klon. Ini membuka jendela terminal baru.
7. Jika repo memerlukan kredensyal, Anda diminta untuk memasukkan nama pengguna dan token akses pribadi Anda. Prompt ini tidak menerima kata sandi, Anda harus menggunakan token akses pribadi. Untuk informasi selengkapnya tentang token akses pribadi, lihat Mengelola token akses pribadi Anda.
8. Tunggu hingga unduhan selesai. Setelah repo dikloning, File Browser terbuka untuk menampilkan repo kloning.
9. Klik dua kali repo untuk membukanya.
10. Pilih ikon Git untuk melihat antarmuka pengguna Git yang sekarang melacak repo.
11. Untuk melacak repo yang berbeda, buka repo di browser file dan kemudian pilih ikon Git.

Hentikan Training Job di SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Anda dapat menghentikan pekerjaan pelatihan dengan Amazon SageMaker Studio Classic UI. Ketika Anda menghentikan pekerjaan pelatihan, statusnya berubah menjadi `Stopping` saat penagihan berhenti. Algoritma dapat menunda penghentian untuk menyimpan artefak model setelah status pekerjaan berubah menjadi`Stopped`. Untuk informasi selengkapnya, lihat metode stop_training_job di. AWS SDK for Python (Boto3)

Untuk menghentikan pekerjaan pelatihan

1. Ikuti Lihat, cari, dan bandingkan percobaan yang dijalankan prosedur di halaman ini sampai Anda membuka tab Describe Trial Component.
2. Di sisi kanan atas tab, pilih Hentikan pekerjaan pelatihan. Status di kiri atas tab berubah menjadi Berhenti.
3. Untuk melihat waktu pelatihan dan waktu penagihan, pilih AWSPengaturan.

Gunakan TensorBoard di Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Dokumen berikut menguraikan cara menginstal dan menjalankan TensorBoard di Amazon SageMaker Studio Classic.

> **Note**
>
> Panduan ini menunjukkan cara membuka TensorBoard aplikasi melalui server notebook SageMaker Studio Classic dari profil pengguna SageMaker Domain individual. Untuk TensorBoard pengalaman yang lebih komprehensif yang terintegrasi dengan SageMaker Pelatihan dan fungsi kontrol akses SageMaker Domain, lihatGunakan TensorBoard untuk men-debug dan menganalisis pekerjaan pelatihan di Amazon SageMaker.

Prasyarat

Tutorial ini membutuhkan SageMaker domain. Lihat informasi yang lebih lengkap di Ikhtisar SageMaker Domain Amazon

Mengatur **`TensorBoardCallback`**

1. Luncurkan Studio Classic, dan buka Launcher. Lihat informasi yang lebih lengkap di Gunakan Amazon SageMaker Studio Classic Launcher
2. Di Amazon SageMaker Studio Classic Launcher, di bawah`Notebooks and compute resources`, pilih tombol Ubah lingkungan.
3. Pada dialog Ubah lingkungan, gunakan menu tarik-turun untuk memilih **`TensorFlow 2.6 Python 3.8 CPU Optimized`** Studio Classic Image.
4. Kembali ke Launcher, klik ubin Buat notebook. Notebook Anda diluncurkan dan terbuka di tab Studio Classic baru.
5. Jalankan kode ini dari dalam sel notebook Anda.
6. Impor paket yang diperlukan.

```
import os
import datetime
import tensorflow as tf
```

7. Buat model Keras.

```
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

def create_model():
  return tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
  ])
```

8. Buat direktori untuk TensorBoard log Anda

```
LOG_DIR = os.path.join(os.getcwd(), "logs/fit/" +
 datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
```

9. Jalankan pelatihan dengan TensorBoard.

```
model = create_model()
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])


tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=LOG_DIR,
 histogram_freq=1)

model.fit(x=x_train,
          y=y_train,
          epochs=5,
          validation_data=(x_test, y_test),
          callbacks=[tensorboard_callback])
```

10. Hasilkan jalur EFS untuk TensorBoard log. Anda menggunakan jalur ini untuk mengatur log Anda dari terminal.

```
EFS_PATH_LOG_DIR = "/".join(LOG_DIR.strip("/").split('/')[1:-1])
print (EFS_PATH_LOG_DIR)
```

Ambil `EFS_PATH_LOG_DIR`. Anda akan membutuhkannya di bagian TensorBoard instalasi.

Instal TensorBoard

1. Klik `Amazon SageMaker Studio Classic` tombol di sudut kiri atas Studio Classic untuk membuka Amazon SageMaker Studio Classic Launcher. Peluncur ini harus dibuka dari direktori root Anda. Lihat informasi yang lebih lengkap di Gunakan Amazon SageMaker Studio Classic Launcher
2. Di Peluncur, di bawah`Utilities and files`, klik`System terminal`.
3. Dari terminal, jalankan perintah berikut. Salin `EFS_PATH_LOG_DIR` dari notebook Jupyter. Anda harus menjalankan ini dari direktori `/home/sagemaker-user` root.

```
pip install tensorboard
```

```
tensorboard --logdir <EFS_PATH_LOG_DIR>
```

Peluncuran TensorBoard

1. Untuk memulai TensorBoard, salin URL Studio Classic Anda dan ganti `lab?` dengan `proxy/6006/` sebagai berikut. Anda harus menyertakan `/` karakter trailing.

   ```
   https://<YOUR_URL>.studio.region.sagemaker.aws/jupyter/default/proxy/6006/
   ```

2. Arahkan ke URL untuk memeriksa hasil Anda.

Menggunakan CodeWhisperer dan CodeGuru ekstensi dengan SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Amazon SageMaker Studio Classic adalah lingkungan pembelajaran mesin terintegrasi tempat Anda dapat membuat, melatih, menerapkan, dan menganalisis semua model Anda dalam aplikasi yang sama. Topik ini menunjukkan cara menghasilkan rekomendasi kode dan menyarankan peningkatan yang terkait dengan masalah kode dengan menggunakan Amazon CodeWhisperer dan Amazon CodeGuru dengan Amazon SageMaker.

Ekstensi berikut mendukung penulisan kode dengan menghasilkan rekomendasi kode dan menyarankan perbaikan yang terkait dengan masalah kode:

- Amazon CodeWhisperer
- Amazon CodeGuru

Apa itu Amazon CodeWhisperer?

Amazon CodeWhisperer adalah layanan yang didukung oleh pembelajaran mesin yang membantu meningkatkan produktivitas pengembang. CodeWhisperer mencapai ini dengan menghasilkan

rekomendasi kode berdasarkan komentar pengembang dalam bahasa alami dan kode mereka di IDE. Selama pratinjau, Amazon CodeWhisperer tersedia untuk Java, Python JavaScript, C # dan TypeScript bahasa pemrograman. Layanan ini terintegrasi dengan JupyterLab, Amazon SageMaker Studio Classic, instans SageMaker notebook Amazon, dan lingkungan pengembangan terintegrasi (IDE) lainnya.

Untuk informasi selengkapnya, lihat Menyiapkan CodeWhisperer dengan Amazon SageMaker Studio Classic.

Apa itu Amazon CodeGuru?

Amazon CodeGuru Security menggunakan penalaran otomatis dan pembelajaran mesin yang diinformasikan oleh praktik terbaik AWS keamanan. CodeGuru Keamanan secara otomatis membuat kebijakan keamanan yang komprehensif, mendeteksi kerentanan keamanan dalam kode Anda, dan menyarankan peningkatan kualitas. Bersama-sama, rekomendasi ini dapat membantu Anda membuat dan menerapkan aplikasi yang aman.

CodeGuru Keamanan meningkatkan keamanan kode Anda dengan cara berikut:

- Secara proaktif mendeteksi pelanggaran dan kerentanan kebijakan keamanan.
- Memberikan rekomendasi untuk mengatasi risiko keamanan.
- Menyarankan perbaikan pada metode yang tidak efisien.

Dari SageMaker, Anda dapat memanggil CodeGuru Keamanan dengan menggunakan plugin Jupyter open-source. Anda dapat menggunakan CodeGuru Security untuk memindai notebook untuk berbagai masalah yang dapat memengaruhi keamanan, kebenaran, reproduktifitas, pemeliharaan, dan kinerja kode Anda. Untuk informasi selengkapnya, lihat Tutorial: Menjalankan pemindaian dengan SageMaker Studio Classic dan JupyterLab.

Kelola Volume Penyimpanan Amazon EFS Anda di SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Pertama kali pengguna di tim Anda melakukan onboard ke Amazon SageMaker Studio Classic, Amazon SageMaker membuat volume Amazon Elastic File System (Amazon EFS) untuk tim. Direktori home dibuat dalam volume untuk setiap pengguna yang melakukan onboard ke Studio Classic sebagai bagian dari tim Anda. File notebook dan file data disimpan dalam direktori ini. Pengguna tidak memiliki akses ke direktori home anggota tim lainnya. SageMaker Domain Amazon tidak mendukung pemasangan volume Amazon EFS khusus atau tambahan.

> **Important**
>
> Jangan hapus volume Amazon EFS. Jika Anda menghapusnya, domain tidak akan berfungsi lagi dan semua pengguna Anda akan kehilangan pekerjaan mereka.

Untuk menemukan volume Amazon EFS Anda

1. Buka konsol SageMaker .
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari halaman Domain, pilih domain untuk menemukan ID.
5. Dari halaman Detail domain, pilih tab Pengaturan domain.
6. Di bawah Pengaturan umum, temukan ID Domain. ID akan dalam format berikut:`d-xxxxxxxxxxxx`.
7. Lewati`Domain ID`, as`DomainId`, ke metode describe_domain.
8. Dalam tanggapan dari`describe_domain`, perhatikan nilai untuk `HomeEfsFileSystemId` kunci. Ini adalah ID sistem file Amazon EFS.
9. Buka konsol Amazon EFS. Pastikan AWS Region adalah Region yang sama dengan yang digunakan oleh Studio Classic.
10. Di bawah Sistem file, pilih ID sistem file dari langkah sebelumnya.
11. Untuk memverifikasi bahwa Anda telah memilih sistem file yang benar, pilih judul Tag. Nilai yang sesuai dengan `ManagedByAmazonSageMakerResource` kunci harus cocok dengan`Studio Classic ID`.

Untuk informasi tentang cara mengakses volume Amazon EFS, lihat Menggunakan sistem file di Amazon EFS.

Untuk menghapus volume Amazon EFS, lihat Menghapus sistem file Amazon EFS.

Memberikan Umpan Balik tentang SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Amazon SageMaker menanggapi umpan balik Anda dengan serius. Kami mendorong Anda untuk memberikan umpan balik.

Untuk memberikan umpan balik

1. Di sebelah kanan SageMaker Studio Classic, temukan ikon Umpan Balik ().
2. Pilih emoji smiley untuk memberi tahu kami betapa puasnya Anda dengan SageMaker Studio Classic dan tambahkan umpan balik yang ingin Anda bagikan kepada kami.
3. Putuskan apakah akan membagikan identitas Anda dengan kami, lalu pilih Kirim.

Matikan dan Perbarui Aplikasi SageMaker Studio Classic dan Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik berikut menunjukkan cara mematikan dan memperbarui Aplikasi SageMaker Studio Classic dan Studio Classic.

Studio Classic menyediakan ikon notifikasi
()
di sudut kanan atas UI Studio Classic. Ikon notifikasi ini menampilkan jumlah pemberitahuan yang belum dibaca. Untuk membaca pemberitahuan, pilih ikon.

Studio Classic menyediakan dua jenis notifikasi:

- Upgrade — Ditampilkan saat Studio Classic atau salah satu aplikasi Studio Classic telah merilis versi baru. Untuk memperbarui Studio Classic, lihatMatikan dan Perbarui SageMaker Studio Classic. Untuk memperbarui aplikasi Studio Classic, lihatMatikan dan Perbarui Aplikasi Studio Classic.
- Informasi - Ditampilkan untuk fitur baru dan informasi lainnya.

Untuk mengatur ulang ikon notifikasi, Anda harus memilih tautan di setiap pemberitahuan. Pemberitahuan baca mungkin masih ditampilkan di ikon. Ini tidak menunjukkan bahwa pembaruan masih diperlukan setelah Anda memperbarui Aplikasi Studio Classic dan Studio Classic.

Untuk mempelajari cara memperbarui Amazon SageMaker Data Wrangler, lihat. Matikan dan Perbarui Aplikasi Studio Classic

Untuk memastikan bahwa Anda memiliki pembaruan perangkat lunak terbaru, perbarui Amazon SageMaker Studio Classic dan aplikasi Studio Classic Anda menggunakan metode yang diuraikan dalam topik berikut.

Topik

- Matikan dan Perbarui SageMaker Studio Classic
- Matikan dan Perbarui Aplikasi Studio Classic

Matikan dan Perbarui SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Untuk memperbarui Amazon SageMaker Studio Classic ke rilis terbaru, Anda harus mematikan JupyterServer aplikasi. Anda dapat mematikan JupyterServer aplikasi dari SageMaker konsol, dari Amazon SageMaker Studio atau dari dalam Studio Classic. Setelah JupyterServer aplikasi dimatikan, Anda harus membuka kembali Studio Classic melalui SageMaker konsol atau dari Studio yang membuat versi baru JupyterServer aplikasi.

Anda tidak dapat menghapus JupyterServer aplikasi saat UI Studio Classic masih terbuka di browser. Jika Anda menghapus JupyterServer aplikasi saat UI Studio Classic masih terbuka di browser, SageMaker secara otomatis membuat ulang JupyterServer aplikasi.

Setiap informasi buku catatan yang belum disimpan hilang dalam prosesnya. Data pengguna dalam volume Amazon EFS tidak terpengaruh.

Beberapa layanan dalam Studio Classic, seperti Data Wrangler, berjalan di aplikasi mereka sendiri. Untuk memperbarui layanan ini, Anda harus menghapus aplikasi untuk layanan tersebut. Untuk mempelajari selengkapnya, lihat Matikan dan Perbarui Aplikasi Studio Classic.

> **Note**
>
> Sebuah JupyterServer aplikasi dikaitkan dengan satu pengguna Studio Classic. Saat Anda memperbarui aplikasi untuk satu pengguna, itu tidak memengaruhi pengguna lain.

Halaman berikut menunjukkan cara memperbarui JupyterServer Aplikasi dari SageMaker konsol, dari Studio, atau dari dalam Studio Classic.

Matikan dan perbarui dari SageMaker konsol

1. Arahkan ke https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pilih Domain yang menyertakan aplikasi Studio Classic yang ingin Anda perbarui.
5. Di bawah Profil pengguna, pilih nama pengguna Anda.
6. Di bawah Aplikasi, di baris yang ditampilkan JupyterServer, pilih Tindakan, lalu pilih Hapus.
7. Pilih Ya, hapus aplikasi.
8. Ketik **`delete`** di kotak konfirmasi.
9. Pilih Hapus.

10. Setelah aplikasi dihapus, luncurkan aplikasi Studio Classic baru untuk mendapatkan versi terbaru.

Matikan dan perbarui dari Studio

1. Arahkan ke Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Dari UI Studio, cari panel aplikasi di sisi kiri.
3. Dari panel aplikasi, pilih Studio Classic.
4. Dari halaman landing Studio Classic, pilih instans Studio Classic untuk berhenti.
5. Pilih Berhenti.
6. Setelah aplikasi dihentikan, pilih Jalankan untuk menggunakan versi terbaru.

Matikan dan perbarui dari dalam Studio Classic

1. Luncurkan Studio Klasik.
2. Di menu atas, pilih File lalu Shut Down.
3. Pilih salah satu opsi berikut:
   - Shutdown Server — Mematikan aplikasi. JupyterServer Sesi terminal, sesi kernel, SageMaker gambar, dan instance tidak dimatikan. Sumber daya ini terus bertambah biaya.
   - Shutdown All — Menutup semua aplikasi, sesi terminal, sesi kernel, SageMaker gambar, dan instance. Sumber daya ini tidak lagi menimbulkan biaya.
4. Tutup jendelanya.
5. Setelah aplikasi dihapus, luncurkan aplikasi Studio Classic baru untuk menggunakan versi terbaru.

Matikan dan Perbarui Aplikasi Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Untuk memperbarui aplikasi Amazon SageMaker Studio Classic ke rilis terbaru, Anda harus terlebih dahulu mematikan KernelGateway aplikasi terkait dari SageMaker konsol. Setelah KernelGateway aplikasi dimatikan, Anda harus membukanya kembali melalui SageMaker Studio Classic dengan menjalankan kernel baru. Kernel secara otomatis diperbarui. Setiap informasi buku catatan yang belum disimpan hilang dalam prosesnya. Data pengguna dalam volume Amazon EFS tidak terpengaruh.

Setelah aplikasi dimatikan selama 24 jam, SageMaker hapus semua metadata untuk aplikasi tersebut. Untuk dianggap sebagai pembaruan dan mempertahankan metadata aplikasi, aplikasi harus dimulai ulang dalam waktu 24 jam setelah aplikasi sebelumnya dimatikan. Setelah jendela waktu ini, pembuatan aplikasi dianggap sebagai aplikasi baru daripada pembaruan aplikasi sebelumnya.

> **Note**
>
> Sebuah KernelGateway aplikasi dikaitkan dengan satu pengguna Studio Classic. Saat Anda memperbarui aplikasi untuk satu pengguna, itu tidak memengaruhi pengguna lain.

Untuk memperbarui KernelGateway aplikasi

1. Arahkan ke https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pilih Domain yang menyertakan aplikasi yang ingin Anda perbarui.
5. Di bawah Profil pengguna, pilih nama pengguna Anda.
6. Di bawah Aplikasi, di baris yang menampilkan nama Aplikasi, pilih Tindakan, lalu pilih Hapus

   Untuk memperbarui Data Wrangler, hapus aplikasi yang dimulai dengan. sagemaker-data-wrang
7. Pilih Ya, hapus aplikasi.
8. Ketik **`delete`** di kotak konfirmasi.
9. Pilih Hapus.
10. Setelah aplikasi dihapus, luncurkan kernel baru dari dalam Studio Classic untuk menggunakan versi terbaru.

Harga Amazon SageMaker Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Saat anggota pertama tim Anda bergabung ke Amazon SageMaker Studio Classic, Amazon SageMaker membuat volume Amazon Elastic File System (Amazon EFS) untuk tim. Ketika anggota ini, atau anggota tim mana pun, membuka Studio Classic, direktori home dibuat dalam volume untuk anggota tersebut. Biaya penyimpanan dikeluarkan untuk direktori ini. Selanjutnya, biaya penyimpanan tambahan dikeluarkan untuk notebook dan file data yang disimpan di direktori home anggota. Untuk informasi harga di Amazon EFS, lihat Harga Amazon EFS.

Biaya tambahan dikeluarkan ketika operasi lain dijalankan di dalam Studio Classic, misalnya, menjalankan notebook, menjalankan pekerjaan pelatihan, dan menghosting model.

Untuk informasi tentang biaya yang terkait dengan penggunaan notebook Studio Classic, lihatPengukuran Penggunaan.

Untuk informasi tentang penagihan beserta contoh harga, lihat SageMaker Harga Amazon.

Jika Amazon SageMaker Studio adalah pengalaman default Anda, lihat Harga Amazon SageMaker Studio untuk informasi harga selengkapnya.

Memecahkan Masalah Amazon Studio Classic SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut ini khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Topik ini menjelaskan cara memecahkan masalah umum Amazon SageMaker Studio Classic selama penyiapan dan penggunaan. Berikut ini adalah kesalahan umum yang mungkin terjadi saat menggunakan Amazon SageMaker Studio Classic. Setiap kesalahan diikuti oleh solusinya.

Masalah aplikasi Studio Classic

Masalah berikut terjadi saat meluncurkan dan menggunakan aplikasi Studio Classic.

- Layar tidak dimuat: Membersihkan ruang kerja dan menunggu tidak membantu

  Saat meluncurkan aplikasi Studio Classic, pop-up menampilkan pesan berikut. Tidak peduli opsi mana yang dipilih, Studio Classic tidak memuat.

  ```
  Loading...
  The loading screen is taking a long time. Would you like to clear the workspace or
   keep waiting?
  ```

  Aplikasi Studio Classic dapat mengalami penundaan peluncuran jika beberapa tab terbuka di ruang kerja Studio Classic atau beberapa file ada di Amazon EFS. Pop-up ini akan hilang dalam beberapa detik setelah ruang kerja Studio Classic siap.

  Jika Anda terus melihat layar pemuatan dengan pemintal setelah memilih salah satu opsi, mungkin ada masalah konektivitas dengan Amazon Virtual Private Cloud yang digunakan oleh Studio Classic.

  Untuk mengatasi masalah konektivitas dengan Amazon Virtual Private Cloud (Amazon VPC) yang digunakan oleh Studio Classic, verifikasi konfigurasi jaringan berikut:

  - Jika domain Anda diatur dalam `VpcOnly` mode: Verifikasi bahwa ada titik akhir VPC Amazon untukAWS STS, atau NAT Gateway untuk lalu lintas keluar, termasuk lalu lintas melalui internet. Untuk melakukannya, ikuti langkah-langkah di Connect SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal.
  - Jika VPC Amazon Anda diatur dengan DNS khusus, bukan DNS yang disediakan oleh Amazon: Verifikasi bahwa rute dikonfigurasi menggunakan Dynamic Host Configuration Protocol (DHCP) untuk setiap titik akhir VPC Amazon yang ditambahkan ke VPC Amazon yang digunakan oleh Studio Classic. Untuk informasi selengkapnya tentang menyetel set opsi DHCP default dan kustom, lihat set opsi DHCP di Amazon VPC.
- Kegagalan Internal saat meluncurkan Studio Classic

Saat meluncurkan Studio Classic, Anda tidak dapat melihat UI Studio Classic. Anda juga melihat kesalahan yang mirip dengan berikut ini, dengan Kegagalan Internal sebagai detail kesalahan.

```
Amazon SageMaker Studio
The JupyterServer app default encountered a problem and was stopped.
```

Kesalahan ini dapat disebabkan oleh banyak faktor. Jika penyelesaian langkah-langkah ini tidak menyelesaikan masalah Anda, buat masalah dengan https://aws.amazon.com/premiumsupport/.

- Target pemasangan Amazon EFS tidak ada: Studio Classic menggunakan Amazon EFS untuk penyimpanan. Volume Amazon EFS memerlukan target pemasangan untuk setiap subnet tempat SageMaker domain Amazon dibuat. Jika target pemasangan Amazon EFS ini dihapus secara tidak sengaja, aplikasi Studio Classic tidak dapat memuat karena tidak dapat memasang direktori file pengguna. Untuk mengatasi masalah ini, selesaikan langkah-langkah berikut.

  Untuk memverifikasi atau membuat target mount.

  1. Temukan volume Amazon EFS yang terkait dengan domain menggunakan panggilan DescribeDomainAPI.
  2. Masuk ke AWS Management Console dan buka konsol Amazon EFS di https://console.aws.amazon.com/efs/.
  3. Dari daftar volume Amazon EFS, pilih volume Amazon EFS yang terkait dengan domain.
  4. Di halaman detail Amazon EFS, pilih tab Jaringan. Verifikasi bahwa ada target mount untuk semua subnet tempat domain diatur.
  5. Jika target pemasangan tidak ada, tambahkan target pemasangan Amazon EFS yang hilang. Untuk petunjuknya, lihat Membuat dan mengelola target pemasangan dan grup keamanan.
  6. Setelah target mount yang hilang dibuat, luncurkan aplikasi Studio Classic.

- File yang bertentangan di **`.local`** folder pengguna: Jika Anda menggunakan JupyterLab versi 1 di Studio Classic, pustaka yang bertentangan di `.local` folder Anda dapat menyebabkan masalah saat meluncurkan aplikasi Studio Classic. Untuk mengatasi hal ini, perbarui JupyterLab versi default profil pengguna Anda ke JupyterLab 3.0. Untuk informasi selengkapnya tentang melihat dan memperbarui JupyterLab versi, lihatJupyterLab Pembuatan Versi.

- ConfigurationError: LifecycleConfig saat meluncurkan Studio Classic

Anda tidak dapat melihat UI Studio Classic saat meluncurkan Studio Classic. Hal ini disebabkan oleh masalah dengan skrip konfigurasi siklus hidup default yang dilampirkan ke domain.

Untuk mengatasi masalah konfigurasi siklus hidup

1. Lihat CloudWatch Log Amazon untuk konfigurasi siklus hidup untuk melacak perintah yang menyebabkan kegagalan. Untuk melihat log, ikuti langkah-langkahnyaVerifikasi proses konfigurasi siklus hidup dari Log CloudWatch .
2. Lepaskan skrip default dari profil pengguna atau domain. Untuk informasi selengkapnya, lihat Perbarui dan lepaskan konfigurasi siklus hidup.
3. Luncurkan aplikasi Studio Classic.
4. Debug skrip konfigurasi siklus hidup Anda. Anda dapat menjalankan skrip konfigurasi siklus hidup dari terminal sistem untuk memecahkan masalah. Ketika skrip berjalan dengan sukses dari terminal, Anda dapat melampirkan skrip ke profil pengguna atau domain.

- SageMaker Fungsi inti Studio Classic tidak tersedia.

Jika Anda mendapatkan pesan kesalahan ini saat membuka Studio Classic, itu mungkin karena konflik versi paket Python. Ini terjadi jika Anda menggunakan perintah berikut di notebook atau terminal untuk menginstal paket Python yang memiliki konflik versi dengan dependensi SageMaker paket.

```
!pip install
```

```
pip install --user
```

Untuk mengatasi masalah ini, selesaikan langkah-langkah berikut:

1. Copot pemasangan paket Python yang baru saja diinstal. Jika Anda tidak yakin paket mana yang akan dihapus, buat masalah dengan https://aws.amazon.com/premiumsupport/.
2. Mulai ulang Studio Classic:
   a. Matikan Studio Classic dari menu File.
   b. Tunggu satu menit.
   c. Buka kembali Studio Classic dengan menyegarkan halaman atau membukanya dari. AWS Management Console

Masalahnya harus diselesaikan jika Anda telah menghapus paket yang menyebabkan konflik. Untuk menginstal paket tanpa menyebabkan masalah ini lagi, gunakan `%pip install` tanpa `--user` tanda.

Jika masalah berlanjut, buat profil pengguna baru dan atur lingkungan Anda dengan profil pengguna tersebut.

Jika solusi ini tidak memperbaiki masalah, buat masalah dengan https://aws.amazon.com/premiumsupport/.

- Tidak dapat membuka Studio Classic dari fileAWS Management Console.

  Jika Anda tidak dapat membuka Studio Classic dan tidak dapat membuat instance baru yang berjalan dengan semua pengaturan default, buat masalah dengan https://aws.amazon.com/premiumsupport/.

KernelGateway masalah aplikasi

Masalah berikut khusus untuk KernelGateway aplikasi yang diluncurkan di Studio Classic.

- Tidak dapat mengakses sesi Kernel

  Ketika pengguna meluncurkan notebook baru, mereka tidak dapat terhubung ke sesi notebook. Jika status KernelGateway aplikasi`In Service`, Anda dapat memverifikasi hal berikut untuk menyelesaikan masalah.

  - Periksa konfigurasi Grup Keamanan

    Jika domain diatur dalam `VPCOnly` mode, grup keamanan yang terkait dengan domain harus mengizinkan lalu lintas antara port dalam rentang `8192-65535` untuk konektivitas antara KernelGateway aplikasi JupyterServer dan aplikasi.

    Untuk memverifikasi aturan grup keamanan

    1. Dapatkan grup keamanan yang terkait dengan domain menggunakan panggilan DescribeDomainAPI.
    2. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
    3. Dari navigasi kiri, di bawah Keamanan, pilih Grup Keamanan.

4. Filter berdasarkan ID grup keamanan yang terkait dengan domain.
5. Untuk setiap kelompok keamanan:

   a. Pilih grup keamanan.

   b. Dari halaman detail grup keamanan, lihat aturan Masuk. Verifikasi bahwa lalu lintas diizinkan antar port dalam jangkauan`8192-65535`.

Untuk informasi selengkapnya tentang aturan grup keamanan, lihat Mengontrol lalu lintas ke sumber daya menggunakan grup keamanan. Untuk informasi selengkapnya tentang persyaratan untuk menggunakan Studio Classic dalam `VPCOnly` mode, lihatConnect SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal.

- Verifikasi firewall dan WebSocket koneksi

  Jika KernelGateway aplikasi memiliki `InService` status dan pengguna tidak dapat terhubung ke sesi notebook Studio Classic, verifikasi firewall dan WebSocket pengaturan.

  1. Luncurkan aplikasi Studio Classic. Untuk informasi selengkapnya, lihat Luncurkan Amazon SageMaker Studio Classic.
  2. Buka alat pengembang browser web Anda.
  3. Pilih tab Jaringan.
  4. Cari entri yang cocok dengan format berikut.

     ```
     wss://<domain-id>.studio.<region>.sagemaker.aws/jupyter/default/api/kernels/
     <unique-code>/channels?session_id=<unique-code>
     ```

     Jika status atau kode respons untuk entri adalah apa pun selain`101`, maka pengaturan jaringan Anda mencegah koneksi antara aplikasi Studio Classic dan KernelGateway aplikasi.

     Untuk mengatasi masalah ini, hubungi tim yang mengelola pengaturan jaringan Anda untuk mengizinkan daftar URL Studio Classic dan mengaktifkan WebSocket koneksi.

- Tidak dapat meluncurkan aplikasi yang disebabkan oleh kuota sumber daya yang terlampaui

  Saat pengguna mencoba meluncurkan buku catatan baru, pembuatan notebook gagal dengan salah satu kesalahan berikut. Ini disebabkan oleh melebihi kuota sumber daya.

- ```
  Unable to start more Apps of AppType [KernelGateway] and
   ResourceSpec(instanceType=[]) for UserProfile []. Please delete an App with a
   matching AppType and ResourceSpec, then try again
  ```

  Studio Classic mendukung hingga empat KernelGateway aplikasi yang berjalan pada instance yang sama. Untuk mengatasi masalah ini, Anda dapat melakukan salah satu hal berikut:

  - Hapus KernelGateway aplikasi yang ada yang berjalan pada instance, lalu restart notebook baru.
  - Mulai buku catatan baru pada jenis instans yang berbeda

  Untuk informasi selengkapnya, lihat Mengubah Tipe Instance.

- ```
  An error occurred (ResourceLimitExceeded) when calling the CreateApp operation
  ```

  Dalam hal ini, akun tidak memiliki batasan yang cukup untuk membuat aplikasi Studio Classic pada jenis instance yang ditentukan. Untuk mengatasi ini, navigasikan ke Service Quotas konsol di https://console.aws.amazon.com/servicequotas/. Di konsol itu, minta untuk menambah `Studio KernelGateway Apps running on` *`instance-type`* `instance` batas. Untuk informasi selengkapnya, lihat AWS service quotas.

SageMaker JupyterLab

Buat JupyterLab ruang di Amazon SageMaker Studio untuk meluncurkan JupyterLab aplikasi. JupyterLab Ruang adalah ruang pribadi dalam Studio yang mengelola penyimpanan dan menghitung sumber daya yang diperlukan untuk menjalankan JupyterLab aplikasi. JupyterLabAplikasi ini adalah lingkungan pengembangan interaktif berbasis web (IDE) untuk notebook, kode, dan data. Gunakan antarmuka JupyterLab aplikasi yang fleksibel dan luas untuk mengonfigurasi dan mengatur alur kerja pembelajaran mesin (ML).

Secara default, JupyterLab aplikasi ini dilengkapi dengan gambar SageMaker Distribusi. Gambar distribusi memiliki paket populer, seperti berikut ini:

- PyTorch
- TensorFlow
- Keras
- NumPy

- Pandas
- Scikit-learn

Di dalam JupyterLab aplikasi, Anda dapat menggunakan Amazon CodeWhisperer, pendamping kode bertenaga AI generatif untuk menghasilkan, men-debug, dan menjelaskan kode Anda.

Buat analitik terpadu dan alur kerja ML di buku catatan Jupyter yang sama. Jalankan Spark pekerjaan interaktif di Amazon EMR dan infrastruktur AWS Glue tanpa server, langsung dari buku catatan Anda. Pantau dan debug pekerjaan lebih cepat menggunakan UI inlineSpark. Dalam beberapa langkah, Anda dapat mengotomatiskan persiapan data Anda dengan menjadwalkan notebook sebagai pekerjaan.

JupyterLab Aplikasi ini membantu Anda bekerja secara kolaboratif dengan rekan-rekan Anda. Gunakan integrasi Git bawaan dalam JupyterLab IDE untuk berbagi dan kode versi. Edit kode di notebook dengan rekan-rekan Anda secara real-time. Bawa sistem penyimpanan file Anda sendiri jika Anda memiliki volume Amazon EFS.

JupyterLab Aplikasi ini berjalan pada satu instans Amazon Elastic Compute Cloud (Amazon EC2) dan menggunakan satu volume Amazon Elastic Block Store (Amazon EBS) untuk penyimpanan. Anda dapat mengganti instans yang lebih cepat atau menambah ukuran volume Amazon EBS untuk kebutuhan Anda.

Aplikasi Juptyerlab 4 berjalan di ruang dalam Studio. JupyterLab Studio Classic menggunakan aplikasi JupyterLab 3. JupyterLab 4 memberikan manfaat sebagai berikut:

- IDE yang lebih cepat daripada Amazon SageMaker Studio Classic, terutama dengan notebook besar
- Pencarian dokumen yang ditingkatkan
- Editor teks yang lebih berkinerja dan dapat diakses

Untuk informasi selengkapnya JupyterLab, lihat JupyterLabDokumentasi.

Topik

- JupyterLab panduan pengguna
- JupyterLab panduan administrator
- Migrasi dari SageMaker Studio Classic ke SageMaker Studio

JupyterLab panduan pengguna

Panduan ini menunjukkan kepada JupyterLab pengguna cara menjalankan alur kerja analitik dan pembelajaran mesin dalam SageMaker Studio. Anda bisa mendapatkan penyimpanan cepat dan skala komputasi Anda naik atau turun, tergantung pada kebutuhan Anda.

JupyterLab hanya mendukung ruang bersama pribadi, di mana ruang pribadi dicakup untuk satu pengguna dalam Domain. Untuk informasi tentang ruang Studio, lihatRuang Amazon SageMaker Studio.

Untuk mulai menggunakan JupyterLab, buat ruang pribadi dan luncurkan JupyterLab aplikasi Anda. Ruang pribadi yang menjalankan JupyterLab aplikasi Anda adalah JupyterLab ruang. JupyterLab Ruang ini menggunakan satu instans Amazon EC2 untuk komputasi Anda dan satu volume Amazon EBS untuk penyimpanan Anda. Segala sesuatu di ruang Anda seperti kode, profil git, dan variabel lingkungan disimpan pada volume Amazon EBS yang sama. Volume ini memiliki 3000 IOPS dan throughput 125 megabyte per detik (MBps). Anda dapat menggunakan penyimpanan cepat untuk membuka dan menjalankan beberapa notebook Jupyter pada instance yang sama. Anda juga dapat mengganti kernel di notebook dengan sangat cepat.

Administrator Anda telah mengonfigurasi pengaturan penyimpanan Amazon EBS default untuk ruang Anda. Ukuran penyimpanan default adalah 5 GB, tetapi Anda dapat meningkatkan jumlah ruang yang Anda dapatkan. Anda dapat berbicara dengan administrator Anda untuk memberi Anda panduan.

Anda dapat mengganti jenis instans Amazon EC2 yang Anda gunakan untuk menjalankan JupyterLab, menskalakan komputasi Anda ke atas atau ke bawah tergantung pada kebutuhan Anda. Instans peluncuran cepat dimulai jauh lebih cepat daripada instance lainnya.

Administrator Anda mungkin memberi Anda konfigurasi siklus hidup yang menyesuaikan lingkungan Anda. Anda dapat menentukan konfigurasi siklus hidup saat Anda membuat ruang.

Jika administrator memberi Anda akses ke Amazon EFS, Anda dapat mengonfigurasi JupyterLab ruang untuk mengaksesnya.

Secara default, JupyterLab aplikasi menggunakan gambar SageMaker distribusi. Ini termasuk dukungan untuk banyak pembelajaran mesin, analitik, dan paket pembelajaran mendalam. Namun, jika Anda memerlukan gambar khusus, administrator Anda dapat membantu menyediakan akses ke gambar khusus.

Volume Amazon EBS bertahan secara independen dari kehidupan sebuah instans. Anda tidak akan kehilangan data Anda ketika Anda mengubah instance. Gunakan pustaka manajemen paket conda

dan pip untuk membuat lingkungan kustom yang dapat direproduksi yang tetap ada bahkan saat Anda mengganti jenis instance.

Untuk mulai menggunakan JupyterLab, buat spasi atau pilih ruang yang dibuat administrator untuk Anda dan buka JupyterLab.

Gunakan prosedur berikut untuk membuat ruang dan terbuka JupyterLab.

Untuk membuat ruang dan terbuka JupyterLab

1. Buka Studio. Untuk informasi tentang membuka Studio, lihatLuncurkan Amazon SageMaker Studio.
2. Pilih JupyterLab.
3. Pilih Buat JupyterLab ruang.
4. Untuk Nama, tentukan nama spasi.
5. Pilih Buat ruang.
6. (Opsional) Misalnya, tentukan instans Amazon EC2 yang menjalankan spasi.
7. (Opsional) Untuk Gambar, tentukan gambar yang disediakan administrator Anda untuk menyesuaikan lingkungan Anda.
8. (Opsional) Untuk Pengaturan Ruang, tentukan yang berikut ini:
   - Penyimpanan (GB) — Hingga 100 GB atau jumlah yang ditentukan administrator Anda.
   - Konfigurasi Siklus Hidup — Konfigurasi siklus hidup yang ditentukan administrator Anda.
   - Lampirkan sistem file EFS khusus — Amazon EFS tempat administrator Anda menyediakan akses.
9. Pilih Jalankan ruang.
10. Pilih Buka JupyterLab.

Konfigurasikan ruang

Setelah Anda membuat JupyterLab spasi, Anda dapat mengonfigurasinya untuk melakukan hal berikut:

- Ubah jenis instance.
- Ubah volume penyimpanan.
- (Admin mengatur diperlukan) Gunakan gambar kustom.

- (Diperlukan pengaturan admin) Gunakan konfigurasi siklus hidup.
- (Diperlukan pengaturan admin) Lampirkan Amazon EFS khusus.

> **Important**
>
> Anda harus menghentikan JupyterLab ruang setiap kali Anda mengkonfigurasinya. Gunakan prosedur berikut untuk mengkonfigurasi ruang.

Untuk mengkonfigurasi spasi

1. Di dalam Studio, navigasikan ke halaman JupyterLab aplikasi.
2. Pilih nama spasi.
3. (Opsional) Untuk Gambar, tentukan gambar yang disediakan administrator Anda untuk menyesuaikan lingkungan Anda.
4. (Opsional) Untuk Pengaturan Ruang, tentukan yang berikut ini:
   - Penyimpanan (GB) — Hingga 100 GB atau jumlah yang dikonfigurasi administrator Anda untuk ruang tersebut.
   - Konfigurasi Siklus Hidup — Konfigurasi siklus hidup yang disediakan administrator Anda.
   - Lampirkan sistem file EFS khusus — Amazon EFS tempat administrator Anda menyediakan akses.
5. Pilih Jalankan ruang.

Saat Anda membuka JupyterLab aplikasi, ruang Anda memiliki konfigurasi yang diperbarui.

Setelah Anda membuka JupyterLab, Anda dapat mengonfigurasi lingkungan Anda menggunakan terminal. Untuk membuka terminal, arahkan ke Launcher dan pilih Terminal.

Berikut ini adalah contoh berbagai cara Anda dapat mengonfigurasi lingkungan JupyterLab.

> **Note**
>
> Di dalam Studio, Anda dapat menggunakan konfigurasi siklus hidup untuk menyesuaikan lingkungan, tetapi sebaiknya gunakan pengelola paket. Menggunakan konfigurasi siklus hidup adalah metode yang lebih rawan kesalahan. Lebih mudah untuk menambah atau

menghapus dependensi daripada men-debug skrip konfigurasi siklus hidup. Hal ini juga dapat meningkatkan waktu JupyterLab startup.
Untuk informasi tentang konfigurasi siklus hidup, lihat. Menggunakan konfigurasi siklus hidup dengan JupyterLab

Sesuaikan lingkungan Anda menggunakan manajer paket

Gunakan pip atau conda untuk menyesuaikan lingkungan Anda. Sebaiknya gunakan pengelola paket alih-alih skrip konfigurasi siklus hidup.

Buat dan aktifkan lingkungan kustom Anda

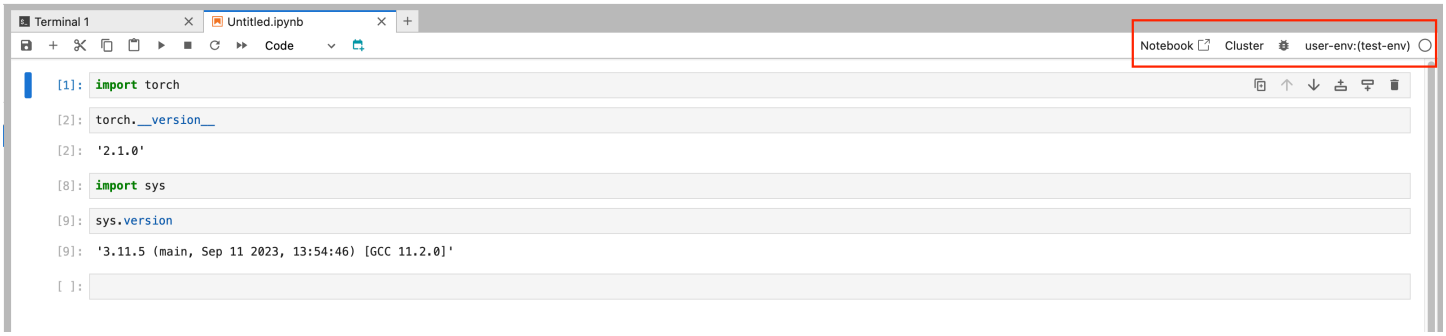
Bagian ini memberikan contoh berbagai cara untuk mengonfigurasi lingkungan JupyterLab.

Lingkungan conda dasar memiliki jumlah minimum paket yang diperlukan untuk alur kerja Anda. SageMaker Gunakan template berikut untuk membuat lingkungan conda dasar:

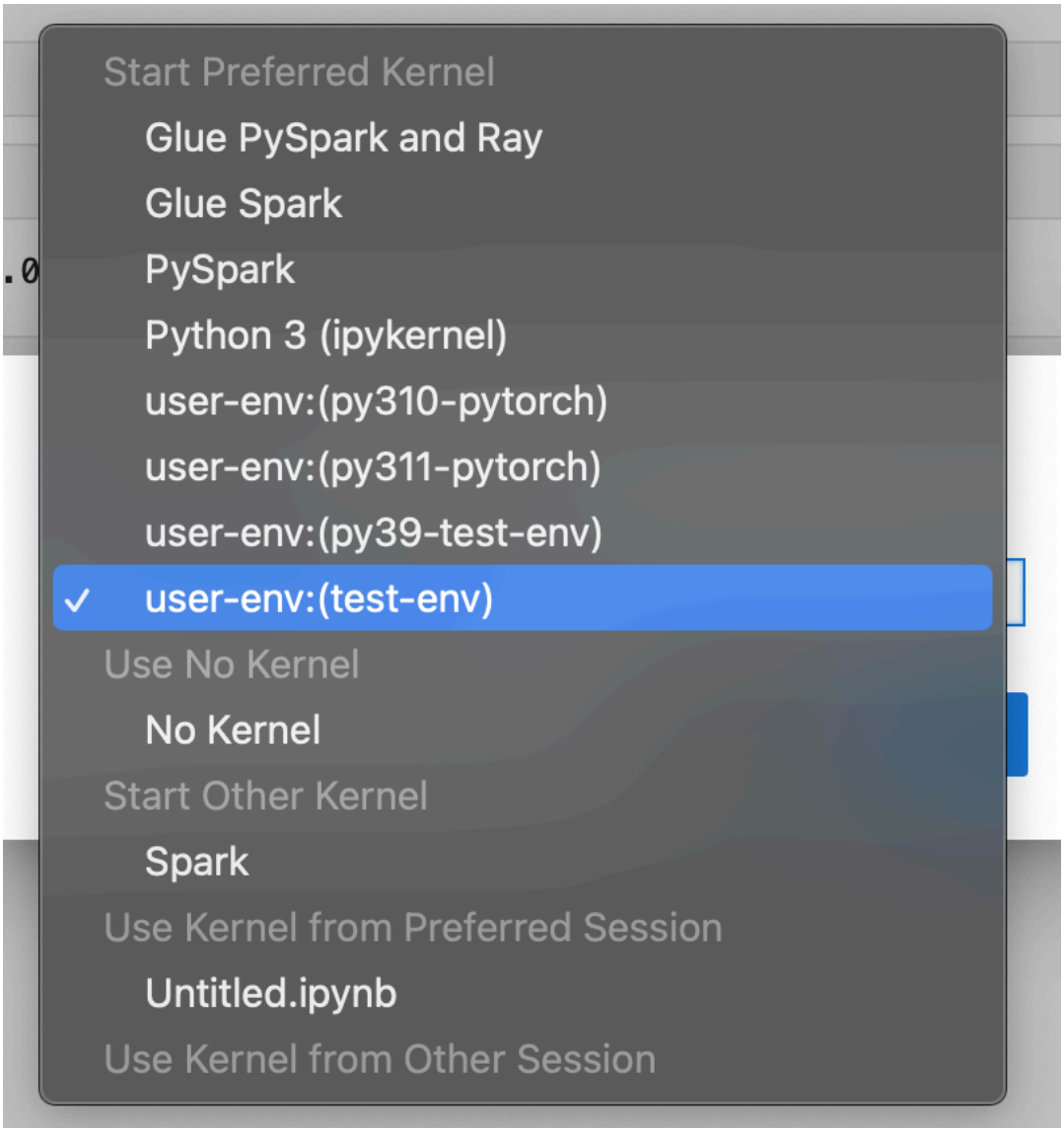
```
# initialize conda for shell interaction
conda init

# create a new fresh environment
conda create --name test-env

# check if your new environment is created successfully
conda info --envs

# activate the new environment
conda activate test-env

# install packages in your new conda environment
conda install pip boto3 pandas ipykernel

# list all packages install in your new environment
conda list

# parse env name information from your new environment
export CURRENT_ENV_NAME=$(conda info | grep "active environment" | cut -d : -f 2 | tr -
d ' ')

# register your new environment as Jupyter Kernel for execution
```

```
python3 -m ipykernel install --user --name $CURRENT_ENV_NAME --display-name "user-env:
($CURRENT_ENV_NAME)"

# to exit your new environment
conda deactivate
```

Gambar berikut menunjukkan lokasi lingkungan yang telah Anda buat.



Untuk mengubah lingkungan Anda, pilih dan pilih opsi dari menu tarik-turun.



Pilih Pilih untuk memilih kernel untuk lingkungan.

Bersihkan lingkungan conda

Membersihkan lingkungan conda yang tidak Anda gunakan dapat membantu membebaskan ruang disk dan meningkatkan kinerja. Gunakan template berikut untuk membersihkan lingkungan conda:

```
# list your environments to select an environment to clean
conda info --envs # or conda info -e

# once you've selected your environment to purge
conda remove --name test-env --all
```

```
# run conda environment list to ensure the target environment is purged
conda info --envs # or conda info -e
```

Buat lingkungan conda dengan versi Python tertentu

Membersihkan lingkungan conda yang tidak Anda gunakan dapat membantu membebaskan ruang disk dan meningkatkan kinerja. Gunakan template berikut untuk membersihkan lingkungan conda:

```
# create a conda environment with a specific python version
conda create --name py38-test-env python=3.8.10

# activate and test your new python version
conda activate py38-test-env & python3 --version

# Install ipykernel to facilicate env registration
conda install ipykernel

# parse env name information from your new environment
export CURRENT_ENV_NAME=$(conda info | grep "active environment" | cut -d : -f 2 | tr -
d ' ')

# register your new environment as Jupyter Kernel for execution
python3 -m ipykernel install --user --name $CURRENT_ENV_NAME --display-name "user-env:
($CURRENT_ENV_NAME)"

# deactivate your py38 test environment
conda deactivate
```

Buat lingkungan conda dengan satu set paket tertentu

Gunakan template berikut untuk membuat lingkungan conda dengan versi Python dan kumpulan paket tertentu:

```
# prefill your conda environment with a set of packages,
conda create --name py38-test-env python=3.8.10 pandas matplotlib=3.7 scipy ipykernel

# activate your conda environment and ensure these packages exist
conda activate py38-test-env
```

```
# check if these packages exist
conda list | grep -E 'pandas|matplotlib|scipy'

# parse env name information from your new environment
export CURRENT_ENV_NAME=$(conda info | grep "active environment" | cut -d : -f 2 | tr -
d ' ')

# register your new environment as Jupyter Kernel for execution
python3 -m ipykernel install --user --name $CURRENT_ENV_NAME --display-name "user-env:
($CURRENT_ENV_NAME)"

# deactivate your conda environment
conda deactivate
```

Kloning conda dari lingkungan yang ada

Kloning lingkungan conda Anda untuk mempertahankan status kerjanya. Anda bereksperimen di lingkungan kloning tanpa harus khawatir memperkenalkan perubahan yang melanggar di lingkungan pengujian Anda.

Gunakan perintah berikut untuk mengkloning lingkungan.

```
# create a fresh env from a base environment
conda create --name py310-base-ext --clone base # replace 'base' with another env

# activate your conda environment and ensure these packages exist
conda activate py310-base-ext

# install ipykernel to register your env
conda install ipykernel

# parse env name information from your new environment
export CURRENT_ENV_NAME=$(conda info | grep "active environment" | cut -d : -f 2 | tr -
d ' ')

# register your new environment as Jupyter Kernel for execution
python3 -m ipykernel install --user --name $CURRENT_ENV_NAME --display-name "user-env:
($CURRENT_ENV_NAME)"

# deactivate your conda environment
conda deactivate
```

Kloning conda dari file YAMAL referensi

Buat lingkungan conda dari file YAMAL referensi. Berikut ini adalah contoh file YAMB yang dapat Anda gunakan.

```
# anatomy of a reference environment.yml
name: py311-new-env
channels:
  - conda-forge
dependencies:
  - python=3.11
  - numpy
  - pandas
  - scipy
  - matplotlib
  - pip
  - ipykernel
  - pip:
      - git+https://github.com/huggingface/transformers
```

Di bawah`pip`, kami sarankan untuk menentukan hanya dependensi yang tidak tersedia dengan conda.

Gunakan perintah berikut untuk membuat lingkungan conda dari file YAMB.

```
# create your conda environment
conda create -f environment.yml

# activate your env
conda activate py311-new-env
```

Bagikan lingkungan antar tipe instance

Anda dapat berbagi lingkungan conda dengan menyimpannya ke direktori Amazon EFS di luar volume Amazon EBS Anda. Pengguna lain dapat mengakses lingkungan di direktori tempat Anda menyimpannya.

> **Important**
>
> Ada batasan dengan berbagi lingkungan Anda. Misalnya, kami tidak merekomendasikan lingkungan yang dimaksudkan untuk berjalan pada instans GPU Amazon EC2 melalui lingkungan yang berjalan pada instance CPU.

Gunakan perintah berikut sebagai templat untuk menentukan direktori target tempat Anda membuat lingkungan khusus. Anda membuat conda dalam jalur tertentu. Anda membuatnya di dalam direktori Amazon EFS. Anda dapat memutar instance baru dan melakukan conda activate path dan melakukannya di Amazon EFS.

```
# if you know your environment path for your conda environment
conda create --prefix /home/sagemaker-user/my-project/py39-test python=3.9

# activate the env with full path from prefix
conda activate home/sagemaker-user/my-project/py39-test

# parse env name information from your new environment
export CURRENT_ENV_NAME=$(conda info | grep "active environment" | awk -F' : ' '{print
 $2}' | awk -F'/' '{print $NF}')

# register your new environment as Jupyter Kernel for execution
python3 -m ipykernel install --user --name $CURRENT_ENV_NAME --display-name "user-env-
prefix:($CURRENT_ENV_NAME)"

# deactivate your conda environment
conda deactivate
```

JupyterLab panduan administrator

Panduan untuk administrator ini menjelaskan SageMaker JupyterLab sumber daya, seperti sumber daya dari Amazon Elastic Block Store (Amazon EBS) dan Amazon Elastic Compute Cloud (Amazon EC2). Topik juga menunjukkan cara menyediakan akses pengguna dan mengubah ukuran penyimpanan.

Sebuah SageMaker JupyterLab ruang terdiri dari sumber daya berikut:

- Volume Amazon EBS berbeda yang menyimpan semua data, seperti kode dan variabel lingkungan.
- Instans Amazon EC2 digunakan untuk menjalankan ruang.
- Gambar yang digunakan untuk menjalankan JupyterLab.

> **Note**
>
> Aplikasi tidak memiliki akses ke volume EBS aplikasi lain. Misalnya, Editor Kode, berdasarkan Code-OSS, Visual Studio Code - Open Source tidak memiliki akses ke volume EBS untuk. JupyterLab Untuk informasi selengkapnya tentang volume EBS, lihat Amazon Elastic Block Store (Amazon EBS).

Anda dapat menggunakan Amazon SageMaker API untuk melakukan hal berikut:

- Ubah ukuran penyimpanan default volume EBS untuk pengguna Anda.
- Ubah ukuran maksimum penyimpanan EBS
- Tentukan pengaturan pengguna untuk aplikasi. Misalnya, Anda dapat menentukan apakah pengguna menggunakan gambar khusus atau repositori kode.
- Tentukan jenis aplikasi dukungan.

Ukuran default volume Amazon EBS adalah 5 GB. Anda dapat meningkatkan ukuran volume hingga maksimum 16.384 GB. Jika Anda tidak melakukan apa pun, pengguna Anda dapat meningkatkan ukuran volumenya menjadi 100 GB.

Kernel yang terkait dengan JupyterLab aplikasi berjalan pada instans Amazon EC2 yang sama yang berjalan. JupyterLab Secara default, SageMaker Distribution Image terbaru digunakan untuk menjalankan JupyterLab. Untuk informasi selengkapnya tentang Gambar SageMaker Distribusi, lihatSageMaker Gambar Distribusi.

Bagian berikut memandu Anda melalui konfigurasi yang perlu Anda lakukan sebagai administrator.

Topik

- Berikan pengguna Anda akses ke ruang pribadi
- Ubah ukuran penyimpanan default untuk JupyterLab pengguna Anda
- Menggunakan konfigurasi siklus hidup dengan JupyterLab

- Lampirkan repo Git
- Sesuaikan lingkungan menggunakan gambar khusus
- Hapus sumber daya yang tidak digunakan
- Kuota

Berikan pengguna Anda akses ke ruang pribadi

Untuk memberi pengguna akses ke ruang pribadi, Anda harus melampirkan kebijakan izin ke peran IAM mereka. Anda juga dapat menggunakan kebijakan izin untuk membatasi ruang pribadi dan aplikasi terkait ke profil pengguna tertentu.

Kebijakan izin berikut memberikan akses ke ruang pribadi. Hal ini memungkinkan pengguna untuk membuat ruang pribadi mereka sendiri dan daftar ruang pribadi lainnya dalam Domain mereka. Pengguna dengan kebijakan ini tidak dapat mengakses ruang pribadi pengguna lain. Untuk informasi tentang ruang Studio, lihatRuang Amazon SageMaker Studio.

Kebijakan ini memberi pengguna izin untuk hal-hal berikut:

- Ruang pribadi.
- Profil pengguna untuk mengakses ruang pribadi.

Untuk memberikan izin, Anda dapat melampirkan kebijakan berikut ke peran IAM pengguna Anda. Anda juga dapat menggunakan kebijakan ini untuk membatasi ruang pribadi, dan aplikasi terkaitnya, ke profil pengguna tertentu.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {

      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateApp",
        "sagemaker:DeleteApp"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:app/*",
      "Condition": {
        "Null": {
```

```
          "sagemaker:OwnerUserProfileArn": "true"
        }
      }
    },
    {
      "Sid": "SMStudioCreatePresignedDomainUrlForUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreatePresignedDomainUrl"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:user-profile/
${sagemaker:DomainId}/${sagemaker:UserProfileName}"
    },
    {
      "Sid": "SMStudioAppPermissionsListAndDescribe",
      "Effect": "Allow",
      "Action": [
        "sagemaker:ListApps",
        "sagemaker:ListDomains",
        "sagemaker:ListUserProfiles",
        "sagemaker:ListSpaces",
        "sagemaker:DescribeApp",
        "sagemaker:DescribeDomain",
        "sagemaker:DescribeUserProfile",
        "sagemaker:DescribeSpace"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SMStudioAppPermissionsTagOnCreate",
      "Effect": "Allow",
      "Action": [
        "sagemaker:AddTags"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:*/*",
      "Condition": {
        "Null": {
          "sagemaker:TaggingAction": "false"
        }
      }
    },
    {
      "Sid": "SMStudioRestrictSharedSpacesWithoutOwners",
      "Effect": "Allow",
```

```
      "Action": [
        "sagemaker:CreateSpace",
        "sagemaker:UpdateSpace",
        "sagemaker:DeleteSpace"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:space/
${sagemaker:DomainId}/*",
      "Condition": {
        "Null": {
          "sagemaker:OwnerUserProfileArn": "true"
        }
      }
    },
    {
      "Sid": "SMStudioRestrictSpacesToOwnerUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateSpace",
        "sagemaker:UpdateSpace",
        "sagemaker:DeleteSpace"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:space/
${sagemaker:DomainId}/*",
      "Condition": {
        "ArnLike": {
          "sagemaker:OwnerUserProfileArn": "arn:aws:sagemaker:$Wilayah AWS:
$111122223333:user-profile/${sagemaker:DomainId}/${sagemaker:UserProfileName}"
        },
        "StringEquals": {
          "sagemaker:SpaceSharingType": [
            "Private",
            "Shared"
          ]
        }
      }
    },
    {
      "Sid": "SMStudioRestrictCreatePrivateSpaceAppsToOwnerUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateApp",
        "sagemaker:DeleteApp"
      ],
```

```
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:app/
${sagemaker:DomainId}/*",
      "Condition": {
        "ArnLike": {
          "sagemaker:OwnerUserProfileArn": "arn:aws:sagemaker:
${aws:Region}:${aws:PrincipalAccount}:user-profile/${sagemaker:DomainId}/
${sagemaker:UserProfileName}"
        },
        "StringEquals": {
          "sagemaker:SpaceSharingType": [
            "Private"
          ]
        }
      }
    },
  ]
}
```

Ubah ukuran penyimpanan default untuk JupyterLab pengguna Anda

Anda dapat mengubah pengaturan penyimpanan default untuk pengguna Anda. Anda juga dapat mengubah pengaturan penyimpanan default berdasarkan kebutuhan organisasi Anda dan kebutuhan pengguna Anda.

Untuk mengubah ukuran penyimpanan, bagian ini menyediakan perintah untuk melakukan hal berikut:

1. Perbarui pengaturan penyimpanan Amazon EBS di SageMaker Domain Amazon (Domain).
2. Buat profil pengguna dan tentukan pengaturan penyimpanan di dalamnya.

Gunakan perintah AWS Command Line Interface (AWS CLI) berikut untuk mengubah ukuran penyimpanan default.

Gunakan AWS CLI perintah berikut untuk memperbarui Domain:

```
aws --region $REGION sagemaker update-domain \
--domain-id $DOMAIN_ID \
--default-user-settings '{
    "SpaceStorageSettings": {
```

```
        "DefaultEbsStorageSettings":{
            "DefaultEbsVolumeSizeInGb":5,
            "MaximumEbsVolumeSizeInGb":100
        }
    }
}'
```

Gunakan AWS CLI perintah berikut untuk membuat profil pengguna dan menentukan pengaturan penyimpanan default:

```
aws --region $REGION sagemaker create-user-profile \
--domain-id $DOMAIN_ID \
--user-profile-name $USER_PROFILE_NAME \
--user-settings '{
    "SpaceStorageSettings": {
        "DefaultEbsStorageSettings":{
            "DefaultEbsVolumeSizeInGb":5,
            "MaximumEbsVolumeSizeInGb":100
        }
    }
}'
```

Gunakan AWS CLI perintah berikut untuk memperbarui pengaturan penyimpanan default di profil pengguna:

```
aws --region $REGION sagemaker update-user-profile \
--domain-id $DOMAIN_ID \
--user-profile-name $USER_PROFILE_NAME \
--user-settings '{
    "SpaceStorageSettings": {
        "DefaultEbsStorageSettings":{
            "DefaultEbsVolumeSizeInGb":25,
            "MaximumEbsVolumeSizeInGb":200
        }
    }
}'
```

Menggunakan konfigurasi siklus hidup dengan JupyterLab

Konfigurasi siklus hidup adalah skrip shell yang dipicu oleh peristiwa JupyterLab siklus hidup, seperti memulai buku catatan baru. JupyterLab Anda dapat menggunakan konfigurasi siklus hidup untuk mengotomatiskan penyesuaian untuk lingkungan Anda. JupyterLab Kustomisasi ini termasuk menginstal paket kustom, mengkonfigurasi ekstensi notebook, preloading dataset, dan menyiapkan repositori kode sumber.

Menggunakan konfigurasi siklus hidup memberi Anda fleksibilitas dan kontrol untuk mengonfigurasi JupyterLab untuk memenuhi kebutuhan spesifik Anda. Misalnya, Anda dapat membuat kumpulan minimal gambar wadah dasar dengan paket dan pustaka yang paling umum digunakan. Kemudian Anda dapat menggunakan konfigurasi siklus hidup untuk menginstal paket tambahan untuk kasus penggunaan tertentu di seluruh tim ilmu data dan pembelajaran mesin Anda.

> **Note**
>
> Setiap skrip memiliki batas 16.384 karakter.

Topik

- Membuat dan mengaitkan konfigurasi siklus hidup
- Debug konfigurasi siklus hidup
- Lepaskan konfigurasi siklus hidup

Membuat dan mengaitkan konfigurasi siklus hidup

Topik ini mencakup instruksi untuk membuat dan mengaitkan konfigurasi siklus hidup dengan. JupyterLab Anda menggunakan AWS Command Line Interface (AWS CLI) untuk mengotomatiskan kustomisasi untuk JupyterLab lingkungan Anda.

Konfigurasi siklus hidup adalah skrip shell yang dipicu oleh peristiwa JupyterLab siklus hidup, seperti memulai buku catatan baru. JupyterLab Untuk informasi selengkapnya tentang konfigurasi siklus hidup, lihat. Menggunakan konfigurasi siklus hidup dengan JupyterLab

Buat konfigurasi siklus hidup () AWS CLI

Pelajari cara membuat konfigurasi siklus hidup menggunakan AWS Command Line Interface (AWS CLI) untuk mengotomatiskan penyesuaian lingkungan Studio Anda.

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensil Anda. Untuk informasi tentang AWS kredensil, lihat Memahami dan mendapatkan kredensil Anda AWS.
- Onboard ke SageMaker Domain Amazon. Untuk informasi konseptual, lihat Ikhtisar SageMaker Domain Amazon. Untuk panduan memulai cepat, lihatCepat onboard ke Domain Amazon SageMaker .

Langkah 1: Buat konfigurasi siklus hidup

Prosedur berikut menunjukkan cara membuat skrip konfigurasi siklus hidup yang mencetak. `Hello World`

> **Note**
>
> Setiap skrip dapat memiliki hingga 16.384 karakter.

1. Dari mesin lokal Anda, buat file bernama `my-script.sh` dengan konten berikut:

   ```
   #!/bin/bash
   set -eux
   echo 'Hello World!'
   ```

2. Gunakan yang berikut ini untuk mengonversi `my-script.sh` file Anda menjadi format base64. Persyaratan ini mencegah kesalahan yang terjadi dari spasi dan pengkodean jeda baris.

   ```
   LCC_CONTENT=`openssl base64 -A -in my-script.sh`
   ```

3. Buat konfigurasi siklus hidup untuk digunakan dengan Studio. Perintah berikut membuat konfigurasi siklus hidup yang berjalan saat Anda meluncurkan aplikasi terkait`JupyterLab`:

   ```
   aws sagemaker create-studio-lifecycle-config \
   --region region \
   --studio-lifecycle-config-name my-jl-lcc \
   --studio-lifecycle-config-content $LCC_CONTENT \
   ```

```
--studio-lifecycle-config-app-type JupyterLab
```

Perhatikan ARN dari konfigurasi siklus hidup yang baru dibuat yang dikembalikan. ARN ini diperlukan untuk melampirkan konfigurasi siklus hidup ke aplikasi Anda.

Langkah 2: Lampirkan konfigurasi siklus hidup ke SageMaker Domain Amazon (Domain) dan profil pengguna

Untuk melampirkan konfigurasi siklus hidup, Anda harus memperbarui `UserSettings` untuk Domain atau profil pengguna Anda. Skrip konfigurasi siklus hidup yang terkait di tingkat Domain diwarisi oleh semua pengguna. Namun, skrip yang terkait pada tingkat profil pengguna dicakup oleh pengguna tertentu.

Anda dapat membuat profil pengguna, Domain, atau spasi baru dengan konfigurasi siklus hidup yang dilampirkan menggunakan perintah berikut:

- create-user-profile
- buat-domain
- menciptakan-ruang

Perintah berikut membuat profil pengguna dengan konfigurasi siklus hidup. Tambahkan ARN konfigurasi siklus hidup dari langkah sebelumnya ke pengguna. `JupyterLabAppSettings` Anda dapat menambahkan beberapa konfigurasi siklus hidup secara bersamaan dengan meneruskan daftarnya. Ketika pengguna meluncurkan JupyterLab aplikasi denganAWS CLI, mereka dapat menentukan konfigurasi siklus hidup alih-alih menggunakan konfigurasi default. Konfigurasi siklus hidup yang dilewati pengguna harus termasuk dalam daftar konfigurasi siklus hidup. `JupyterLabAppSettings`

```
# Create a new UserProfile
aws sagemaker create-user-profile --domain-id domain-id \
--user-profile-name user-profile-name \
--region region \
--user-settings '{
"JupyterLabAppSettings": {
  "LifecycleConfigArns":
    [lifecycle-configuration-arn-list]
  }
}'
```

Debug konfigurasi siklus hidup

Topik berikut menunjukkan cara mendapatkan informasi tentang dan men-debug konfigurasi siklus hidup Anda.

Topik

Verifikasi proses konfigurasi siklus hidup dari Log CloudWatch

Konfigurasi siklus hidup hanya log dan. `STDOUT` `STDERR`

`STDOUT`adalah output default untuk skrip bash. Anda dapat menulis ke `STDERR` dengan menambahkan `>&2` ke akhir perintah bash. Misalnya, `echo 'hello'>&2`.

Log untuk konfigurasi siklus hidup Anda dipublikasikan ke Anda menggunakan Akun AWS Amazon. CloudWatch Log ini dapat ditemukan di aliran `/aws/sagemaker/studio` log di CloudWatch konsol.

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pilih Log dari panel navigasi kiri. Dari menu tarik-turun, pilih Grup log.
3. Pada halaman Grup log, cari`aws/sagemaker/studio`.
4. Pilih grup log .
5. Pada halaman Detail grup log, pilih tab Aliran log.
6. Untuk menemukan log untuk aplikasi tertentu, cari aliran log menggunakan format berikut:

   ```
   domain-id/user-profile-name/app-type/app-name
   ```

   String pencarian berikut menemukan log konfigurasi siklus hidup untuk Domain`d-m85lcu8vbqmz`, profil pengguna, jenis aplikasi `i-sonic-jsJupyterLab`, dan nama aplikasi: `test-lcc-echo`

   ```
   d-m85lcu8vbqmz/i-sonic-js/JupyterLab/test-lcc-echo
   ```

7. Untuk melihat log eksekusi skrip, pilih aliran log yang ditambahkan. `LifecycleConfigOnStart`

Batas waktu konfigurasi siklus hidup

Ada batasan batas waktu konfigurasi siklus hidup 5 menit. Jika skrip konfigurasi siklus hidup membutuhkan waktu lebih dari 5 menit untuk dijalankan, Anda mendapatkan kesalahan.

Untuk mengatasi kesalahan ini, pastikan skrip konfigurasi siklus hidup Anda selesai dalam waktu kurang dari 5 menit.

Untuk membantu mengurangi runtime skrip, coba yang berikut ini:

- Kurangi langkah yang tidak perlu. Misalnya, batasi lingkungan conda mana untuk menginstal paket besar.
- Jalankan tugas dalam proses paralel.
- Gunakan perintah nohup dalam skrip Anda untuk memastikan bahwa sinyal hangup diabaikan sehingga skrip berjalan tanpa henti.

Lepaskan konfigurasi siklus hidup

Untuk memperbarui skrip Anda, Anda harus membuat skrip konfigurasi siklus hidup baru dan melampirkannya ke SageMaker Domain Amazon masing-masing (Domain), profil pengguna, atau ruang bersama. Skrip konfigurasi siklus hidup tidak dapat diubah setelah dibuat. Untuk informasi selengkapnya tentang membuat dan melampirkan konfigurasi siklus hidup, lihat. Membuat dan mengaitkan konfigurasi siklus hidup

Bagian berikut menunjukkan cara melepaskan konfigurasi siklus hidup menggunakan (). AWS Command Line Interface AWS CLI

Lepaskan menggunakan AWS CLI

Untuk melepaskan konfigurasi siklus hidup menggunakan (AWS CLI), hapus konfigurasi siklus hidup yang diinginkan dari daftar konfigurasi siklus hidup yang dilampirkan ke sumber daya. Anda kemudian meneruskan daftar sebagai bagian dari perintah masing-masing:

- update-user-profile
- pembaruan-domain
- ruang pembaruan-

Misalnya, perintah berikut menghapus semua konfigurasi siklus hidup untuk JupyterLab aplikasi yang dilampirkan ke Domain.

```
aws sagemaker update-domain --domain-id domain-id \
--region region \
--default-user-settings '{
"JupyterLabAppSettings": {
  "LifecycleConfigArns":
    []
  }
}'
```

Lampirkan repo Git

JupyterLab menawarkan ekstensi Git untuk memasukkan URL repositori Git (repo), mengkloning ke lingkungan, mendorong perubahan, dan melihat riwayat komit. Anda juga dapat melampirkan URL repo Git yang disarankan ke SageMaker Domain Amazon (Domain) atau profil pengguna.

Bagian berikut menunjukkan cara melampirkan URL repo Git ke Domain atau profil pengguna dari AWS Command Line Interface (AWS CLI) dan konsol. SageMaker Bagian juga menyediakan AWS CLI perintah untuk melepaskan URL repositori ini.

Lampirkan repositori Git () AWS CLI

Bagian ini menunjukkan cara melampirkan URL repositori Git (repo) menggunakan URL. AWS CLI Setelah Anda melampirkan URL repo Git, Anda dapat mengkloningnya dengan mengikuti langkah-langkah di. Mengkloning repo Git di Amazon Studio SageMaker

Prasyarat

Sebelum menggunakan fungsi , pastikan untuk melengkapi prasyarat berikut:

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS Command Line Interface Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensyal, lihat Memahami dan mendapatkan kredensyal Anda AWS.
- Onboard ke SageMaker Domain Amazon. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.

Lampirkan repo Git ke SageMaker Domain Amazon (Domain) atau profil pengguna

URL repo Git yang terkait pada tingkat Domain diwarisi oleh semua pengguna. Namun, URL repo Git yang terkait pada tingkat profil pengguna dicakup oleh pengguna tertentu. Anda dapat melampirkan

beberapa URL repo Git ke SageMaker Domain Amazon atau ke profil pengguna dengan meneruskan daftar URL repositori.

Bagian berikut menunjukkan cara melampirkan URL repo Git ke Domain dan profil pengguna Anda.

Lampirkan ke SageMaker Domain Amazon

Perintah berikut melampirkan URL repo Git ke Domain yang ada:

```
aws sagemaker update-domain --region region --domain-id domain-id \
    --default-user-settings
 JupyterLabAppSettings={CodeRepositories=[{RepositoryUrl="repository"}]}
```

Lampirkan ke profil pengguna

Perintah berikut melampirkan URL repo Git ke profil pengguna yang ada:

```
aws sagemaker update-user-profile --domain-id domain-id --user-profile-name user-name\
    --user-settings
 JupyterLabAppSettings={CodeRepositories=[{RepositoryUrl="repository"}]}
```

Mengkloning repo Git di Amazon Studio SageMaker

Amazon SageMaker Studio terhubung ke repo Git lokal saja. Untuk mengakses file di repo, kloning repo Git dari dalam Studio. Untuk melakukannya, Studio menawarkan ekstensi Git bagi Anda untuk memasukkan URL repo Git, mengkloningkannya ke lingkungan Anda, mendorong perubahan, dan melihat riwayat komit.

Jika repo bersifat pribadi dan memerlukan kredensyal untuk mengakses, Anda menerima prompt untuk memasukkan kredensyal pengguna Anda. Kredensyal Anda termasuk nama pengguna dan token akses pribadi Anda. Untuk informasi selengkapnya tentang token akses pribadi, lihat Mengelola token akses pribadi Anda.

Admin juga dapat melampirkan URL repositori Git yang disarankan di Domain SageMaker Amazon atau tingkat profil pengguna. Pengguna kemudian dapat memilih URL repo dari daftar saran dan mengkloningnya ke Studio. Untuk informasi lebih lanjut tentang melampirkan repo yang disarankan, lihat. Lampirkan Repos Git yang Disarankan ke Studio Classic

Lepaskan URL repo Git

Bagian ini menunjukkan cara melepaskan URL repositori Git dari SageMaker Domain Amazon (Domain) atau profil pengguna. Anda dapat melepaskan URL repo dengan menggunakan AWS Command Line Interface (AWS CLI) atau konsol Amazon. SageMaker

Lepaskan repo Git menggunakan AWS CLI

Untuk melepaskan semua URL repo Git dari Domain atau profil pengguna, Anda harus meneruskan daftar kosong repositori kode. Daftar ini dilewatkan sebagai bagian dari `JupyterLabAppSettings` parameter dalam `update-user-profile` perintah `update-domain` atau. Untuk melepaskan hanya satu URL repo Git, masukkan daftar repositori kode tanpa URL repo Git yang diinginkan.

Lepaskan dari Domain Amazon SageMaker

Perintah berikut melepaskan semua URL repo Git dari Domain:

```
aws sagemaker update-domain --region region --domain-name domain-name \
    --domain-settings JupyterLabAppSettings={CodeRepositories=[]}
```

Lepaskan dari profil pengguna

Perintah berikut melepaskan semua URL repo Git dari profil pengguna:

```
aws sagemaker update-user-profile --domain-name domain-name --user-profile-name user-
name\
    --user-settings JupyterLabAppSettings={CodeRepositories=[]}
```

Sesuaikan lingkungan menggunakan gambar khusus

Jika Anda memerlukan fungsionalitas yang berbeda dari yang disediakan oleh SageMaker distribusi, Anda dapat membawa gambar Anda sendiri dengan ekstensi dan paket khusus Anda. Anda juga dapat menggunakannya untuk mempersonalisasi JupyterLab UI untuk kebutuhan branding atau kepatuhan Anda sendiri.

Untuk tutorial yang membantu Anda membuat gambar yang dapat dijalankan pengguna Anda di JupyterLab lingkungan mereka, lihatBerikan pengguna akses ke gambar khusus.

Untuk persyaratan gambar Anda, lihatSpesifikasi Dockerfile.

Topik

Berikan pengguna akses ke gambar khusus

Dokumentasi ini memberikan step-by-step petunjuk untuk memberi pengguna Anda akses ke gambar khusus dalam JupyterLab lingkungan mereka. Anda dapat menggunakan informasi di halaman ini untuk membuat lingkungan kustom untuk alur kerja pengguna Anda. Prosesnya melibatkan pemanfaatan:

- Docker
- AWS Command Line Interface
- Amazon Elastic Container Registry
- Amazon SageMaker AWS Management Console

Setelah mengikuti panduan di halaman ini, JupyterLab pengguna di SageMaker Domain Amazon akan memiliki akses ke gambar dan lingkungan khusus dari ruang Jupyter mereka untuk memberdayakan alur kerja pembelajaran mesin mereka.

> **Important**
>
> Halaman ini mengasumsikan bahwa Anda memiliki AWS Command Line Interface dan Docker menginstal pada mesin lokal Anda.

Agar pengguna Anda berhasil menjalankan gambar mereka di dalamnya JupyterLab, Anda harus melakukan hal berikut:

Agar pengguna Anda berhasil menjalankan gambar

1. Buat Dockerfile
2. Bangun gambar dari Dockerfile
3. Unggah gambar ke Amazon Elastic Container Registry
4. Lampirkan gambar ke SageMaker Domain Amazon Anda
5. Minta pengguna Anda mengakses gambar dari JupyterLab ruang Anda

Langkah 1: Buat Dockerfile

Buat Dockerfile untuk menentukan langkah-langkah yang diperlukan untuk membuat lingkungan yang diperlukan untuk menjalankan aplikasi di wadah pengguna Anda.

> **Important**
>
> Dockerfile Anda harus memenuhi spesifikasi yang disediakan di. Spesifikasi Dockerfile

Gunakan template Dockerfile berikut untuk membuat gambar Amazon Linux 2:

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2

ARG NB_USER="sagemaker-user"
ARG NB_UID="1000"
ARG NB_GID="100"
RUN yum install --assumeyes python3 shadow-utils && \
    useradd --create-home --shell /bin/bash --gid "${NB_GID}" --uid ${NB_UID}
 ${NB_USER} && \
    yum clean all && \
    python3 -m pip install jupyterlab

RUN python3 -m pip install --upgrade pip

RUN python3 -m pip install --upgrade urllib3==1.26.6

USER ${NB_UID}
CMD jupyter lab --ip 0.0.0.0 --port 8888 \
  --ServerApp.base_url="/jupyterlab/default" \
  --ServerApp.token='' \
  --ServerApp.allow_origin='*'
```

Gunakan template Dockerfile berikut untuk membuat Gambar SageMaker Distribusi Amazon:

```
FROM public.ecr.aws/sagemaker/sagemaker-distribution:latest-cpu
ARG NB_USER="sagemaker-user"
```

```
ARG NB_UID=1000
ARG NB_GID=100

ENV MAMBA_USER=$NB_USER

USER root

RUN apt-get update
RUN micromamba install sagemaker-inference --freeze-installed --yes --channel conda-
forge --name base

USER $MAMBA_USER

ENTRYPOINT ["jupyter-lab"]
CMD ["--ServerApp.ip=0.0.0.0", "--ServerApp.port=8888", "--ServerApp.allow_origin=*",
 "--ServerApp.token=''", "--ServerApp.base_url=/jupyterlab/default"]
```

Langkah 2: Bangun Dockerfile

Di direktori yang sama dengan Dockerfile Anda, buat gambar Anda menggunakan perintah berikut:

```
docker build -t username/imagename:tag your-account-id.dkr.ecr.Wilayah
 AWS.amazonaws.com/your-repository-name:tag
```

> **Important**
>
> Gambar Anda harus ditandai dalam format berikut: *123456789012*.dkr.ecr.your-region.amazonaws.com/*your-repository-name*:*tag*
> Anda tidak akan dapat mendorongnya ke repositori Amazon Elastic Container Registry jika tidak.

Langkah 3: Dorong gambar ke repositori Amazon Elastic Container Registry

Setelah Anda membuat gambar Anda, masuk ke repositori Amazon ECR Anda menggunakan perintah berikut:

```
aws ecr get-login-password --region Wilayah AWS | docker login --username AWS --
password-stdin 123456789012.dkr.ecr.Wilayah AWS.amazonaws.com
```

Setelah Anda masuk, dorong Dockerfile Anda menggunakan perintah berikut:

```
docker push 123456789012.dkr.ecr.Wilayah AWS.amazonaws.com/your-repository-name:tag
```

Langkah 4: Lampirkan gambar ke SageMaker Domain Amazon pengguna Anda

Setelah Anda mendorong gambar, Anda harus mengaksesnya dari SageMaker Domain Amazon Anda. Gunakan prosedur berikut untuk melampirkan gambar ke SageMaker Domain:

1. Buka konsol SageMaker .
2. Di bawah Konfigurasi Admin, pilih Domain.
3. Dari daftar Domain, pilih Domain.
4. Buka tab Lingkungan.
5. Untuk gambar Kustom untuk aplikasi Studio pribadi, pilih Lampirkan gambar.
6. Tentukan sumber gambar.
7. Pilih Berikutnya.
8. Pilih Kirim.

Pengguna Anda sekarang dapat memilih gambar yang telah Anda lampirkan ke Domain mereka dari JupyterLab ruang mereka.

Spesifikasi Dockerfile

Gambar yang Anda tentukan di Dockerfile Anda harus sesuai dengan spesifikasi di bagian berikut untuk membuat gambar dengan sukses.

Menjalankan gambar

- `Entrypoint`— Kami merekomendasikan untuk menyematkan titik masuk ke dalam gambar menggunakan Docker `CMD` atau `Entrypoint` instruksi. Anda juga dapat mengonfigurasi `ContainerEntrypoint` dan `ContainerArguments` yang diteruskan ke wadah saat runtime.

- `EnvVariables`— Dengan Studio, Anda dapat mengonfigurasi `ContainerEnvironment` variabel yang tersedia untuk wadah. Variabel lingkungan ditimpa dengan variabel lingkungan dari SageMaker. Untuk memberi Anda pengalaman yang lebih baik, variabel lingkungan biasanya `AWS_` dan `SageMaker_namespaced` memprioritaskan lingkungan platform.

  Berikut ini adalah variabel lingkungan:

  - `AWS_REGION`
  - `AWS_DEFAULT_REGION`
  - `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`
  - `SageMaker_SPACE_NAME`

Spesifikasi untuk pengguna dan sistem file

- `WorkingDirectory`— Volume Amazon EBS digunakan sebagai penyimpanan. Ini dilampirkan ke`/home/sagemaker-user`, direktori kerja wadah. Gunakan `WORKDIR` perintah untuk menentukan direktori kerja.
- `UID`— ID pengguna Docker wadah. UID=1000 adalah nilai yang didukung. Anda dapat menambahkan akses sudo ke pengguna Anda. ID dipetakan ulang untuk mencegah proses yang berjalan di wadah memiliki lebih banyak hak istimewa daripada yang diperlukan.
- `GID`— ID grup Docker wadah. GID=100 adalah nilai yang didukung. Anda dapat menambahkan akses sudo ke pengguna Anda. ID dipetakan ulang untuk mencegah proses yang berjalan di wadah memiliki lebih banyak hak istimewa daripada yang diperlukan.
- Direktori metadata — Direktori `/opt/.sagemakerintenral` dan `/opt/ml` direktori yang digunakan oleh. AWS File metadata di `/opt/ml` berisi metadata tentang sumber daya seperti. `DomainId`

  Gunakan perintah berikut untuk menampilkan isi sistem file:

```
cat /opt/ml/metadata/resource-metadata.json
{"AppType":"JupyterLab","DomainId":"example-domain-id","UserProfileName":"example-
user-profile-name,"ResourceArn":"arn:aws:sagemaker:Wilayah
 AWS:111122223333;:app/Domain-ID/user-ID/Jupyte
rLab/default","ResourceName":"default","AppImageVersion":"current"}
```

- Direktori logging — `/var/logs/studio` dicadangkan untuk direktori logging JupyterLab dan ekstensi yang terkait dengannya. Kami menyarankan Anda untuk tidak menggunakan folder dalam membuat gambar Anda.

Pemeriksaan Kesehatan dan URL untuk aplikasi

- `Base URL`— URL dasar untuk aplikasi BYOI harus. `jupyterlab/default` Anda hanya dapat memiliki satu aplikasi dan harus selalu diberi nama`default`.
- `HealthCheck API`— `HostAgent` Menggunakan `HealthCheckAPI` at port 8888 untuk memeriksa kesehatan JupyterLab aplikasi. `jupyterlab/default/api/status`adalah titik akhir untuk pemeriksaan kesehatan.
- `Home/Default URL`— `/opt/ml` Direktori `/opt/.sagemakerinternal` dan direktori yang digunakan olehAWS. File metadata di `/opt/ml` berisi metadata tentang sumber daya seperti. `DomainId`
- Otentikasi — Untuk mengaktifkan otentikasi bagi pengguna Anda, matikan token notebook Jupyter atau otentikasi berbasis kata sandi dan izinkan semua asal.

Berikut ini adalah contoh Amazon Linux 2 Dockerfile yang memenuhi spesifikasi sebelumnya:

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2

ARG NB_USER="sagemaker-user"
ARG NB_UID="1000"
ARG NB_GID="100"
RUN yum install --assumeyes python3 shadow-utils && \
    useradd --create-home --shell /bin/bash --gid "${NB_GID}" --uid ${NB_UID}
 ${NB_USER} && \
    yum clean all && \
    python3 -m pip install jupyterlab

RUN python3 -m pip install --upgrade pip

RUN python3 -m pip install --upgrade urllib3==1.26.6

USER ${NB_UID}
CMD jupyter lab --ip 0.0.0.0 --port 8888 \
  --ServerApp.base_url="/jupyterlab/default" \
  --ServerApp.token='' \
```

```
  --ServerApp.allow_origin='*'
```

Berikut ini adalah contoh Amazon SageMaker Distribution Dockerfile yang memenuhi spesifikasi sebelumnya:

```
FROM public.ecr.aws/sagemaker/sagemaker-distribution:latest-cpu
ARG NB_USER="sagemaker-user"
ARG NB_UID=1000
ARG NB_GID=100

ENV MAMBA_USER=$NB_USER

USER root

RUN apt-get update
RUN micromamba install sagemaker-inference --freeze-installed --yes --channel conda-
forge --name base

USER $MAMBA_USER

ENTRYPOINT ["jupyter-lab"]
CMD ["--ServerApp.ip=0.0.0.0", "--ServerApp.port=8888", "--ServerApp.allow_origin=*",
 "--ServerApp.token=''", "--ServerApp.base_url=/jupyterlab/default"]
```

Hapus sumber daya yang tidak digunakan

Untuk menghindari biaya tambahan yang berjalan JupyterLab, sebaiknya hapus sumber daya yang tidak terpakai dengan urutan sebagai berikut:

1. JupyterLab aplikasi
2. Spasi
3. Profil pengguna
4. Domain

Gunakan perintah AWS Command Line Interface (AWS CLI) berikut untuk menghapus sumber daya dalam Domain:

Delete a JupyterLab application

```
aws --region Wilayah AWS sagemaker delete-app --domain-id example-Domain-id --app-
name default --app-type JupyterLab --space-name example-space-name
```

Delete a space

```
aws --region Wilayah AWS sagemaker delete-space --domain-id example-Domain-id  --
space-name example-space-name
```

Delete a user profile

```
aws --region Wilayah AWS sagemaker delete-user-profile --domain-id example-Domain-id
 --user-profile example-user-profile
```

Kuota

JupyterLab, memiliki kuota sebagai berikut:

- Jumlah semua volume Amazon EBS dalam fileAkun AWS.
- Jenis instans yang tersedia untuk pengguna Anda.
- Jumlah instance untuk tertentu yang dapat diluncurkan pengguna Anda.

Untuk mendapatkan lebih banyak penyimpanan dan komputasi bagi pengguna Anda, minta peningkatan AWS kuota Anda. Untuk informasi selengkapnya tentang meminta peningkatan kuota, lihat SageMaker titik akhir dan kuota Amazon.

Migrasi dari SageMaker Studio Classic ke SageMaker Studio

> **Important**
>
> Sebelum Anda dapat memigrasikan pengguna dari Amazon SageMaker Studio Classic ke Amazon SageMaker Studio, Anda harus memberi mereka izin di Studio. Untuk informasi selengkapnya, lihat Tetapkan Studio sebagai pengalaman default.

Memigrasi pengguna Anda dari Amazon SageMaker Studio Classic ke Amazon SageMaker Studio memungkinkan pengguna Anda memanfaatkan fitur dan peningkatan kinerja Studio. Panduan ini memberikan gambaran umum tentang proses migrasi. Ini memberikan panduan tentang mentransfer data pengguna, memigrasi konfigurasi siklus hidup, dan memastikan kompatibilitas ekstensi. JupyterLab

Migrasi pengguna Anda dari Studio Classic ke Studio melibatkan hal-hal berikut:

- Migrasi data pengguna
- Membawa konfigurasi siklus hidup pengguna
- Migrasi ekstensi pengguna JupyterLab

Migrasi data pengguna dari Amazon SageMaker Studio Classic ke Amazon SageMaker Studio

Studio Classic menyimpan file dalam volume Amazon Elastic File System (Amazon EFS). Studio menyimpan file di Amazon Elastic Block Store (Amazon EBS).

Kami menyarankan Anda mentransfer direktori home pengguna dari Amazon EFS ke Amazon EBS. Mentransfer direktori home membantu pengguna Anda mempertahankan lingkungan kerja mereka setelah migrasi.

Anda dapat memigrasi direktori home dengan berbagai cara. Berikut ini adalah contoh metode yang dapat Anda gunakan:

1. Identifikasi volume EFS yang Anda gunakan untuk penyimpanan pengguna Studio Classic.
2. Untuk setiap pengguna, salin direktori home mereka dari Amazon EFS ke volume EBS yang sesuai yang dialokasikan. JupyterLab

3. Perbarui izin volume EBS baru sehingga pengguna dapat mengakses direktori beranda yang dimigrasi. JupyterLab
4. Komunikasikan langkah-langkah kepada pengguna sehingga mereka tahu apa yang diharapkan saat meluncurkan JupyterLab untuk pertama kalinya setelah migrasi.

Anda dapat menggunakannya AWS DataSync untuk menyalin konten sistem file Studio Classic EFS ke bucket Amazon S3.

Setelah DataSync tugas selesai, konten direktori home pengguna Studio Classic disalin ke bucket Amazon S3 target. Tombol objek S3 memiliki awalan yang cocok dengan `HomeEfsFileSystemUid` nilai untuk setiap profil pengguna. Anda dapat menggunakan `DescribeUserProfile` operasi untuk mendapatkan nilai.

Anda dapat menerapkan skrip shell untuk menyalin direktori home dari Amazon S3 ke `/home/sagemaker-user` direktori pada volume Amazon EBS. Script harus melakukan semua hal berikut:

- Dapatkan profil dan `HomeEfsFileSystemUid` nilai pengguna saat ini.
- Gunakan `HomeEfsFileSystemUid` untuk mengidentifikasi awalan S3 untuk isi direktori home pengguna.
- Salin objek di bawah awalan S3 ke. `/home/sagemaker-user`

> **Important**
>
> Sebelum Anda menjalankan skrip, periksa hal-hal berikut:
>
> - Volume Amazon EBS cukup besar untuk menyimpan objek yang Anda ekspor.
> - Anda tidak memigrasi file dan folder tersembunyi, seperti `.bashrc` dan `.condarc` jika Anda tidak bermaksud melakukannya.
> - Peran eksekusi AWS Identity and Access Management (IAM) yang terkait dengan profil Studio Classic u ser memiliki kebijakan yang dikonfigurasi untuk mengakses hanya direktori home masing-masing di Amazon S3.

Anda dapat menjalankan skrip secara manual dalam JupyterLab aplikasi atau mengaturnya sebagai konfigurasi siklus hidup.

Saat Anda menerapkan skrip, lakukan hal berikut:

- Hanya salin direktori pengguna tertentu, tidak semua data Amazon S3.
- Selesaikan kesalahan atau transfer yang tidak lengkap.
- Pertahankan izin, kepemilikan, dan metadata file dan direktori.
- Untuk menghindari duplikat, hapus salinan file sebelumnya di`/home/sagemaker-user`.
- Verifikasi bahwa skrip dapat berjalan beberapa kali tanpa masalah.

Setelah Anda menyinkronkan direktori home untuk semua pengguna, hapus direktori home yang diekspor dari Amazon S3. Ini akan membantu mengurangi biaya penyimpanan dan melepaskan konfigurasi siklus hidup.

Bawa konfigurasi siklus hidup dari Studio ke JupyterLab

Notebook Amazon SageMaker Studio Classic didasarkan pada arsitektur gateway kernel. Jupyter ServerKernel dan kernel lainnya berjalan pada sumber daya komputasi terpisah. Dalam Studio Classic, Anda sering harus membuat konfigurasi siklus hidup terpisah untuk keduanya kernel gateway dan aplikasi. Jupyter Server Untuk informasi selengkapnya tentang konfigurasi siklus hidup Studio Classic, lihat. SageMaker JupyterLab

Di Amazon SageMaker Studio, keduanya Jupyter Server dan kernel berjalan pada instans Amazon EC2 yang sama. Lingkungan runtime JupyterLab aplikasi Studio didasarkan pada gambar dari SageMaker Distribusi. Runtime di Studio Classic berjalan pada gambar yang berbeda, kecuali gambar dari SageMaker Distribusi telah ditentukan.

Agar berhasil memigrasi lingkungan pengguna, kami sarankan untuk mengubah skrip LCC Anda dan mengujinya.

Migrasikan ekstensi JupyterLab

Studio Classic menggunakan JupyterLab 3. Studio menggunakan JupyterLab 4. JupyterLabEkstensi kustom di Studio Classic mungkin tidak berfungsi di Studio, jadi sebaiknya periksa kompatibilitas ekstensi. Anda mungkin harus meng-upgrade mereka.

Untuk informasi selengkapnya tentang ekstensi, lihat Kompatibilitas Ekstensi dengan JupyterLab 4.0.

Instans SageMaker Notebook Amazon

Instance SageMaker notebook Amazon adalah instance komputasi machine learning (ML) yang menjalankan Aplikasi Notebook Jupyter. SageMaker mengelola pembuatan instance dan sumber

daya terkait. Gunakan notebook Jupyter dalam instance notebook Anda untuk menyiapkan dan memproses data, menulis kode untuk melatih model, menyebarkan model ke SageMaker hosting, dan menguji atau memvalidasi model Anda.

SageMaker juga menyediakan contoh notebook yang berisi panduan kode lengkap. Panduan ini menunjukkan cara menggunakan SageMaker untuk melakukan tugas pembelajaran mesin yang umum. Untuk informasi selengkapnya, lihat Contoh Notebook.

Untuk informasi tentang penetapan harga dengan instans SageMaker notebook Amazon, lihat SageMaker Harga Amazon.

Maintenance

SageMaker memperbarui perangkat lunak yang mendasari Instans SageMaker Notebook Amazon setidaknya sekali setiap 90 hari. Beberapa pembaruan pemeliharaan, seperti peningkatan sistem operasi, mungkin mengharuskan aplikasi Anda diambil offline untuk waktu yang singkat. Tidak mungkin melakukan operasi apa pun selama periode ini saat perangkat lunak yang mendasarinya sedang diperbarui. Kami menyarankan Anda me-restart notebook Anda setidaknya sekali setiap 30 hari untuk secara otomatis mengkonsumsi patch.

Untuk informasi lebih lanjut, hubungi https://aws.amazon.com/premiumsupport/.

Topik

- Menggunakan Instans Notebook untuk membuat model
- Instans notebook Amazon Linux 2
- JupyterLab pembuatan versi
- Membuat Instance Notebook
- Akses Instans Notebook
- Memperbarui Instance Notebook
- Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup
- Contoh Notebook
- Mengatur Kernel Notebook
- Kaitkan Repositori Git dengan SageMaker Instans Notebook
- Metadata Instans Notebook
- Pantau Log Jupyter di Log Amazon CloudWatch

Menggunakan Instans Notebook untuk membuat model

Salah satu cara terbaik bagi praktisi machine learning (ML) untuk menggunakan Amazon SageMaker adalah dengan melatih dan menerapkan model ML menggunakan instance SageMaker notebook. Instans SageMaker notebook membantu menciptakan lingkungan dengan memulai server Jupyter di Amazon Elastic Compute Cloud (Amazon EC2) dan menyediakan kernel yang telah dikonfigurasi sebelumnya dengan paket-paket berikut: Amazon SageMaker Python SDK,, AWS CLI (), Conda, Pandas, pustaka kerangka pembelajaran mendalam AWS SDK for Python (Boto3)AWS Command Line Interface, dan pustaka lain untuk ilmu data dan pembelajaran mesin.

Machine Learning dengan SageMaker Python SDK

Untuk melatih, memvalidasi, menyebarkan, dan mengevaluasi model ML dalam instance SageMaker notebook, gunakan Python SageMaker SDK. SageMaker Python SDK abstrak AWS SDK for Python (Boto3) dan operasi API. SageMaker Ini memungkinkan Anda untuk berintegrasi dengan dan mengatur AWS layanan lain, seperti Amazon Simple Storage Service (Amazon S3) untuk menyimpan data dan artefak model, Amazon Elastic Container Registry (ECR) untuk mengimpor dan melayani model ML, Amazon Elastic Compute Cloud (Amazon EC2) untuk pelatihan dan inferensi.

Anda juga dapat memanfaatkan SageMaker fitur yang membantu Anda menangani setiap tahap siklus ML lengkap: pelabelan data, pra-pemrosesan data, pelatihan model, penerapan model, evaluasi kinerja prediksi, dan pemantauan kualitas model dalam produksi.

Jika Anda adalah SageMaker pengguna pertama kali, kami sarankan Anda untuk menggunakan SageMaker Python SDK, mengikuti end-to-end tutorial ML. Untuk menemukan dokumentasi open source, lihat Amazon SageMaker Python SDK.

Ikhtisar Tutorial

Tutorial Memulai ini memandu Anda melalui cara membuat instance SageMaker notebook, membuka notebook Jupyter dengan kernel yang telah dikonfigurasi sebelumnya dengan lingkungan Conda untuk pembelajaran mesin, dan memulai SageMaker sesi untuk menjalankan siklus ML. end-to-end Anda akan mempelajari cara menyimpan kumpulan data ke bucket Amazon S3 default yang dipasangkan secara otomatis dengan SageMaker sesi, mengirimkan tugas pelatihan model ML ke Amazon EC2, dan menerapkan model terlatih untuk prediksi dengan hosting atau inferensi batch melalui Amazon EC2.

Tutorial ini secara eksplisit menunjukkan alur lengkap dari pelatihan model XGBoost dari kumpulan model bawaan. SageMaker Anda menggunakan kumpulan data Sensus Dewasa AS, dan Anda mengevaluasi kinerja model SageMaker XGBoost terlatih dalam memprediksi pendapatan individu.

- SageMakerXGBoost - Model XGBoost disesuaikan dengan SageMaker lingkungan dan dikonfigurasikan sebelumnya sebagai wadah Docker. SageMakermenyediakan serangkaian algoritma bawaan yang disiapkan untuk menggunakan SageMaker fitur. Untuk mempelajari selengkapnya tentang algoritma ML yang disesuaikan SageMaker, lihat Memilih Algoritma dan Menggunakan Algoritma SageMaker Bawaan Amazon. Untuk operasi API algoritma SageMaker bawaan, lihat Algoritma Pihak Pertama di Amazon Python SageMaker SDK.
- Dataset Sensus Dewasa — Dataset dari database biro Sensus 1994 oleh Ronny Kohavi dan Barry Becker (Data Mining and Visualization, Silicon Graphics). Model SageMaker XGBoost dilatih menggunakan dataset ini untuk memprediksi apakah seseorang menghasilkan lebih dari $50.000 per tahun atau kurang.

Topik

Langkah 1: Buat Instans SageMaker Notebook Amazon

Instans SageMaker notebook Amazon adalah instans komputasi Amazon Elastic Compute Cloud (Amazon EC2) terkelola sepenuhnya yang menjalankan Aplikasi Notebook Jupyter. Anda menggunakan instance notebook untuk membuat dan mengelola notebook Jupyter untuk memproses data sebelumnya dan untuk melatih serta menerapkan model pembelajaran mesin.

Untuk membuat instance SageMaker notebook

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih instance Notebook, lalu pilih Buat instance notebook.
3. Pada halaman Create notebook instance, berikan informasi berikut (jika bidang tidak disebutkan, tinggalkan nilai default):

    a. Untuk nama instance Notebook, ketikkan nama untuk instance notebook Anda.

b. Untuk jenis Instance Notebook, pilih`ml.t2.medium`. Ini adalah jenis instance paling murah yang didukung oleh instance notebook, dan itu cukup untuk latihan ini. Jika jenis `ml.t2.medium` instans tidak tersedia di AWS Wilayah Anda saat ini, pilih`ml.t3.medium`.

c. Untuk Pengenal Platform, pilih jenis platform untuk membuat instance notebook. Jenis platform ini menentukan Sistem Operasi dan JupyterLab versi yang membuat instance notebook Anda. Untuk informasi tentang jenis pengenal platform, lihatInstans notebook Amazon Linux 2. Untuk informasi tentang versi JupyterLab, lihat JupyterLab pembuatan versi.

d. Untuk peran IAM, pilih Buat peran baru, lalu pilih Buat peran. Peran IAM ini secara otomatis mendapatkan izin untuk mengakses bucket S3 apa pun yang ada `sagemaker` di namanya. Ini mendapatkan izin ini melalui `AmazonSageMakerFullAccess` kebijakan, yang SageMaker melekat pada peran.

> **Note**
>
> Jika Anda ingin memberikan izin peran IAM untuk mengakses bucket S3 tanpa `sagemaker` nama, Anda harus melampirkan `S3FullAccess` kebijakan atau membatasi izin ke bucket S3 tertentu ke peran IAM. Untuk informasi selengkapnya dan contoh penambahan kebijakan bucket ke peran IAM, lihat Contoh Kebijakan Bucket.

e. Pilih Buat instans notebook.

Dalam beberapa menit, SageMaker meluncurkan instans komputasi ML—dalam hal ini, instance notebook—dan melampirkan volume penyimpanan Amazon EBS sebesar 5 GB ke dalamnya. Instance notebook memiliki server notebook Jupyter yang telah dikonfigurasi sebelumnya, dan pustaka AWS SDK, SageMaker dan satu set pustaka Anaconda.

Untuk informasi selengkapnya tentang membuat instance SageMaker notebook, lihat Membuat Instance Notebook.

(Opsional) Ubah Pengaturan Instans SageMaker Notebook

Jika Anda ingin mengubah jenis instans komputasi ML atau ukuran penyimpanan Amazon EBS dari instance SageMaker notebook yang sudah dibuat, Anda dapat mengedit setelan instans notebook.

Untuk mengubah dan memperbarui jenis instans SageMaker Notebook dan volume EBS

1. Pada halaman instance Notebook di SageMaker konsol, pilih instance notebook Anda.
2. Pilih Tindakan, pilih Berhenti, lalu tunggu hingga instance notebook berhenti sepenuhnya.
3. Setelah status instance buku catatan berubah menjadi Berhenti, pilih Tindakan, lalu pilih Perbarui pengaturan.

   a. Untuk jenis instance Notebook, pilih jenis instans ML yang berbeda.

   b. Untuk ukuran Volume dalam GB, ketik bilangan bulat yang berbeda untuk menentukan ukuran volume EBS baru.

   > **Note**
   >
   > Volume penyimpanan EBS dienkripsi, jadi tidak SageMaker dapat menentukan jumlah ruang kosong yang tersedia pada volume. Karena itu, Anda dapat meningkatkan ukuran volume saat memperbarui instance notebook, tetapi Anda tidak dapat mengurangi ukuran volume. Jika Anda ingin mengurangi ukuran volume penyimpanan ML yang digunakan, buat instance notebook baru dengan ukuran yang diinginkan.

4. Di bagian bawah halaman, pilih Perbarui instance notebook.
5. Saat pembaruan selesai, Mulai instance notebook dengan pengaturan baru.

Untuk informasi selengkapnya tentang memperbarui setelan instans SageMaker notebook, lihat Memperbarui Instance Notebook.

(Opsional) Pengaturan Lanjutan untuk Instans SageMaker Notebook

Video tutorial berikut menunjukkan cara mengatur dan menggunakan instance SageMaker notebook melalui SageMaker konsol dengan opsi lanjutan, seperti konfigurasi SageMaker siklus hidup dan mengimpor repositori. GitHub (Panjangnya: 26:04)

Untuk dokumentasi lengkap tentang instance SageMaker notebook, lihat Menggunakan Instans SageMaker notebook Amazon.

Langkah 2: Buat Notebook Jupyter

Untuk memulai pembuatan skrip untuk pelatihan dan penerapan model Anda, buat notebook Jupyter di instance notebook. SageMaker Dengan menggunakan notebook Jupyter, Anda dapat melakukan

eksperimen pembelajaran mesin (ML) untuk pelatihan dan inferensi sambil mengakses SageMaker fitur dan infrastruktur. AWS

Untuk membuat notebook Jupyter

1. Buka instance notebook sebagai berikut:

   a. Masuk ke SageMaker konsol di https://console.aws.amazon.com/sagemaker/.

   b. Pada halaman instance Notebook, buka instance notebook Anda dengan memilih Open JupyterLab for the JupyterLab interface atau Open Jupyter untuk tampilan Jupyter klasik.

   > **Note**
   >
   > Jika status instance buku catatan menunjukkan Pending di kolom Status, instance buku catatan Anda masih dibuat. Status akan berubah menjadi InServicesaat instance notebook siap digunakan.

2. Buat buku catatan sebagai berikut:

   - Jika Anda membuka buku catatan di JupyterLab tampilan, pada menu File, pilih Baru, lalu pilih Notebook. Untuk Select Kernel, pilih conda_python3. Lingkungan prainstal ini mencakup instalasi Anaconda default dan Python 3.
   - Jika Anda membuka buku catatan dalam tampilan Jupyter klasik, pada tab File, pilih Baru, lalu pilih conda_python3. Lingkungan prainstal ini mencakup instalasi Anaconda default dan Python 3.

3. Simpan notebook sebagai berikut:

   - Dalam JupyterLab tampilan, pilih File, pilih Save Notebook As... , dan kemudian ganti nama notebook.
   - Dalam tampilan klasik Jupyter, pilih File, pilih Save as... , dan kemudian ganti nama notebook.

Langkah 3: Unduh, Jelajahi, dan Ubah Dataset

Pada langkah ini, Anda memuat kumpulan data Sensus Dewasa ke instance notebook menggunakan Pustaka SHAP (Shapley Additive Explanations), meninjau kumpulan data, mengubahnya, dan mengunggahnya ke Amazon S3. SHAP adalah pendekatan teoritis permainan untuk menjelaskan output dari setiap model pembelajaran mesin. Untuk informasi selengkapnya tentang SHAP, lihat Selamat datang di dokumentasi SHAP.

Untuk menjalankan contoh berikut, tempelkan kode sampel ke dalam sel di instance notebook Anda.

Muat Dataset Sensus Dewasa Menggunakan SHAP

Menggunakan pustaka SHAP, impor dataset Sensus Dewasa seperti yang ditunjukkan berikut:

```
import shap
X, y = shap.datasets.adult()
X_display, y_display = shap.datasets.adult(display=True)
feature_names = list(X.columns)
feature_names
```

> **Note**
>
> Jika kernel Jupyter saat ini tidak memiliki pustaka SHAP, instal dengan menjalankan perintah berikut: `conda`
>
> ```
> %conda install -c conda-forge shap
> ```
>
> Jika Anda menggunakan JupyterLab, Anda harus menyegarkan kernel secara manual setelah instalasi dan pembaruan selesai. Jalankan skrip IPython berikut untuk mematikan kernel (kernel akan restart secara otomatis):
>
> ```
> import IPython
> IPython.Application.instance().kernel.do_shutdown(True)
> ```

Objek `feature_names` daftar harus mengembalikan daftar fitur berikut:

```
['Age',
 'Workclass',
 'Education-Num',
 'Marital Status',
 'Occupation',
 'Relationship',
 'Race',
 'Sex',
 'Capital Gain',
 'Capital Loss',
 'Hours per week',
```

```
 'Country']
```

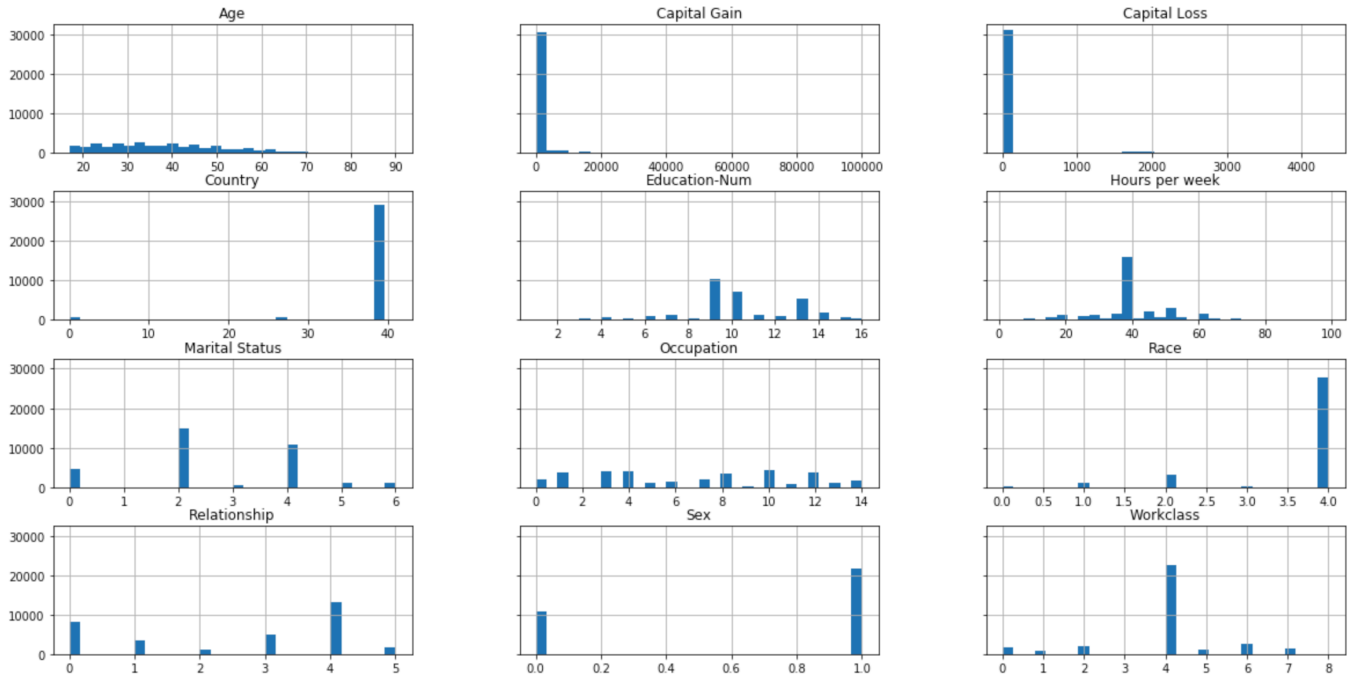
> **Tip**
>
> Jika memulai dengan data yang tidak berlabel, Anda dapat menggunakan Amazon SageMaker Ground Truth untuk membuat alur kerja pelabelan data dalam hitungan menit. Untuk mempelajari lebih lanjut, lihat Data Label.

Ikhtisar Dataset

Jalankan skrip berikut untuk menampilkan ikhtisar statistik dari kumpulan data dan histogram fitur numerik.

```
display(X.describe())
hist = X.hist(bins=30, sharey=True, figsize=(20, 10))
```

	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country
count	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581646	3.868892	10.080679	2.611836	6.572740	2.494518	3.665858	0.669205	1077.649170	87.303833	40.437454	36.718866
std	13.640442	1.455960	2.572562	1.506222	4.228857	1.758232	0.848806	0.470506	7385.911621	403.014771	12.347933	7.823782
min	17.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	28.000000	4.000000	9.000000	2.000000	3.000000	0.000000	4.000000	0.000000	0.000000	0.000000	40.000000	39.000000
50%	37.000000	4.000000	10.000000	2.000000	7.000000	3.000000	4.000000	1.000000	0.000000	0.000000	40.000000	39.000000
75%	48.000000	4.000000	12.000000	4.000000	10.000000	4.000000	4.000000	1.000000	0.000000	0.000000	45.000000	39.000000
max	90.000000	8.000000	16.000000	6.000000	14.000000	5.000000	4.000000	1.000000	99999.000000	4356.000000	99.000000	41.000000



> **Tip**
>
> Jika Anda ingin menggunakan kumpulan data yang perlu dibersihkan dan diubah, Anda dapat menyederhanakan dan merampingkan preprocessing data dan rekayasa fitur menggunakan Amazon Data Wrangler. SageMaker Untuk mempelajari lebih lanjut, lihat Mempersiapkan Data ML dengan Amazon SageMaker Data Wrangler.

Pisahkan Dataset menjadi Train, Validation, dan Test Datasets

Menggunakan Sklearn, bagi dataset menjadi satu set pelatihan dan set tes. Set pelatihan digunakan untuk melatih model, sedangkan set tes digunakan untuk mengevaluasi kinerja model terlatih akhir. Dataset diurutkan secara acak dengan benih acak tetap: 80 persen dari kumpulan data untuk set pelatihan dan 20 persennya untuk satu set tes.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
 random_state=1)
X_train_display = X_display.loc[X_train.index]
```

Pisahkan set pelatihan untuk memisahkan set validasi. Set validasi digunakan untuk mengevaluasi kinerja model yang dilatih sambil menyetel hiperparameter model. 75 persen dari set pelatihan menjadi set pelatihan akhir, dan sisanya adalah set validasi.

```
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25,
 random_state=1)
X_train_display = X_display.loc[X_train.index]
X_val_display = X_display.loc[X_val.index]
```

Menggunakan paket pandas, secara eksplisit menyelaraskan setiap dataset dengan menggabungkan fitur numerik dengan label sebenarnya.

```
import pandas as pd
train = pd.concat([pd.Series(y_train, index=X_train.index,
                             name='Income>50K', dtype=int), X_train], axis=1)
validation = pd.concat([pd.Series(y_val, index=X_val.index,
                            name='Income>50K', dtype=int), X_val], axis=1)
test = pd.concat([pd.Series(y_test, index=X_test.index,
                            name='Income>50K', dtype=int), X_test], axis=1)
```

Periksa apakah kumpulan data dibagi dan terstruktur seperti yang diharapkan:

```
train
```

	Income>50K	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country
10911	1	47.0	4	9.0	2	3	4	4	1	0.0	0.0	40.0	39
17852	0	31.0	4	13.0	2	7	4	3	1	0.0	0.0	36.0	26
29165	1	32.0	4	10.0	2	13	5	4	0	0.0	0.0	32.0	39
30287	0	58.0	4	9.0	2	3	4	2	1	0.0	0.0	40.0	39
24019	0	17.0	4	6.0	4	6	3	4	1	0.0	0.0	20.0	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...
21168	0	43.0	4	8.0	2	14	4	4	1	0.0	0.0	40.0	39
6452	0	26.0	4	9.0	4	7	0	4	1	0.0	0.0	52.0	39
31352	0	32.0	7	14.0	2	10	4	4	1	0.0	0.0	50.0	39
6575	0	45.0	4	9.0	4	6	0	4	1	0.0	0.0	40.0	39
23608	0	23.0	4	9.0	4	1	1	4	0	0.0	0.0	40.0	39

19536 rows × 13 columns

```
validation
```

	Income>50K	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country
16530	0	25.0	4	4.0	2	6	4	4	1	0.0	0.0	40.0	26
26723	0	41.0	6	9.0	2	5	5	4	0	0.0	0.0	40.0	39
3338	0	79.0	0	9.0	6	0	0	2	0	0.0	0.0	30.0	39
19367	1	43.0	2	15.0	2	10	4	4	1	15024.0	0.0	45.0	39
30274	0	51.0	5	9.0	4	12	2	4	1	0.0	0.0	40.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1604	0	46.0	7	9.0	2	13	4	4	1	0.0	0.0	40.0	39
5937	1	71.0	4	10.0	6	12	0	4	1	0.0	0.0	35.0	39
11034	0	36.0	4	9.0	5	14	2	4	1	0.0	0.0	60.0	26
2819	0	31.0	4	9.0	4	8	0	4	0	0.0	0.0	40.0	39
14152	1	37.0	4	10.0	2	12	4	4	1	0.0	0.0	50.0	11

6512 rows × 13 columns

```
test
```

	Income>50K	Age	Workclass	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country
9646	0	62.0	6	4.0	6	8	0	4	0	0.0	0.0	66.0	39
709	0	18.0	4	7.0	4	8	2	4	1	0.0	0.0	25.0	39
7385	1	25.0	4	13.0	4	5	3	4	1	27828.0	0.0	50.0	39
16671	0	33.0	4	9.0	2	10	4	4	1	0.0	0.0	40.0	39
21932	0	36.0	4	7.0	4	7	1	4	0	0.0	0.0	40.0	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...
5889	1	39.0	4	13.0	2	10	5	4	0	0.0	0.0	20.0	39
25723	0	17.0	4	6.0	4	12	3	4	0	0.0	0.0	20.0	39
29514	0	35.0	4	9.0	4	14	3	4	1	0.0	0.0	40.0	39
1600	0	30.0	4	7.0	2	3	4	4	1	0.0	0.0	45.0	39
639	1	52.0	6	16.0	2	10	4	4	1	0.0	0.0	60.0	39

6513 rows × 13 columns

Ubah Kumpulan Data Kereta dan Validasi ke File CSV

Konversi objek `train` dan kerangka `validation` data ke file CSV agar sesuai dengan format file input untuk algoritma XGBoost.

```
# Use 'csv' format to store the data
# The first column is expected to be the output column
train.to_csv('train.csv', index=False, header=False)
validation.to_csv('validation.csv', index=False, header=False)
```

Unggah Dataset ke Amazon S3

Menggunakan SageMaker dan Boto3, unggah kumpulan data pelatihan dan validasi ke bucket Amazon S3 default. Kumpulan data dalam bucket S3 akan digunakan oleh instans yang dioptimalkan komputasi SageMaker di Amazon EC2 untuk pelatihan.

Kode berikut menyiapkan URI bucket S3 default untuk SageMaker sesi Anda saat ini, membuat `demo-sagemaker-xgboost-adult-income-prediction` folder baru, dan mengunggah kumpulan data pelatihan dan validasi ke subfolder. `data`

```
import sagemaker, boto3, os
bucket = sagemaker.Session().default_bucket()
prefix = "demo-sagemaker-xgboost-adult-income-prediction"

boto3.Session().resource('s3').Bucket(bucket).Object(
    os.path.join(prefix, 'data/train.csv')).upload_file('train.csv')
boto3.Session().resource('s3').Bucket(bucket).Object(
```

```
    os.path.join(prefix, 'data/validation.csv')).upload_file('validation.csv')
```

Jalankan berikut ini AWS CLI untuk memeriksa apakah file CSV berhasil diunggah ke bucket S3.

```
! aws s3 ls {bucket}/{prefix}/data --recursive
```

Ini harus mengembalikan output berikut:

```
2021-01-14 17:52:09     786285 demo-sagemaker-xgboost-adult-income-prediction/data/train.csv
2021-01-14 17:52:10     262122 demo-sagemaker-xgboost-adult-income-prediction/data/validation.csv
```

Langkah 4: Latih Model

Amazon SageMaker Python SDK menyediakan estimator kerangka kerja dan estimator generik untuk melatih model Anda sambil mengatur siklus hidup machine learning (ML) yang mengakses fitur SageMaker untuk pelatihan dan infrastruktur, seperti Amazon Elastic Container Registry (Amazon ECR), Amazon Elastic Compute Cloud (Amazon EC2), Amazon AWS Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3). Untuk informasi selengkapnya tentang estimator kerangka kerja SageMaker bawaan, lihat Kerangka kerja dalam dokumentasi Amazon SageMaker Python SDK. Untuk informasi selengkapnya tentang algoritma bawaan, lihatGunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih.

Topik

- Pilih Algoritma Pelatihan
- Membuat dan Menjalankan Training Job

Pilih Algoritma Pelatihan

Untuk memilih algoritme yang tepat untuk kumpulan data Anda, Anda biasanya perlu mengevaluasi model yang berbeda untuk menemukan model yang paling sesuai dengan data Anda. Untuk kesederhanaan, algoritma SageMaker Algoritma XGBoost built-in digunakan di seluruh tutorial ini tanpa pra-evaluasi model.

> **Tip**
>
> Jika Anda SageMaker ingin menemukan model yang sesuai untuk kumpulan data tabular Anda, gunakan Amazon SageMaker Autopilot yang mengotomatiskan solusi pembelajaran mesin. Untuk informasi selengkapnya, lihat SageMaker Autopilot.

Membuat dan Menjalankan Training Job

Setelah Anda mengetahui model mana yang akan digunakan, mulailah membuat SageMaker estimator untuk pelatihan. Tutorial ini menggunakan algoritma built-in XGBoost untuk estimator SageMaker generik.

Untuk menjalankan pekerjaan pelatihan model

1. Impor Amazon SageMaker Python SDK dan mulai dengan mengambil informasi dasar dari sesi Anda saat ini. SageMaker

```
import sagemaker

region = sagemaker.Session().boto_region_name
print("AWS Region: {}".format(region))

role = sagemaker.get_execution_role()
print("RoleArn: {}".format(role))
```

   Ini mengembalikan informasi berikut:

   - `region`— AWS Wilayah saat ini tempat instance SageMaker notebook berjalan.
   - `role`— Peran IAM yang digunakan oleh instance notebook.

   > **Note**
   >
   > Periksa versi SageMaker Python SDK dengan menjalankan. `sagemaker.__version__` Tutorial ini didasarkan pada`sagemaker>=2.20`. Jika SDK sudah usang, instal versi terbaru dengan menjalankan perintah berikut:
   >
   > ```
   > ! pip install -qU sagemaker
   > ```
   >
   > Jika Anda menjalankan instalasi ini di instance SageMaker Studio atau notebook yang keluar, Anda perlu menyegarkan kernel secara manual untuk menyelesaikan penerapan pembaruan versi.

2. Buat estimator XGBoost menggunakan kelas. `sagemaker.estimator.Estimator` Dalam kode contoh berikut, estimator XGBoost diberi nama. `xgb_model`

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs
from sagemaker.session import TrainingInput

s3_output_location='s3://{}/{}/{}'.format(bucket, prefix, 'xgboost_model')

container=sagemaker.image_uris.retrieve("xgboost", region, "1.2-1")
print(container)

xgb_model=sagemaker.estimator.Estimator(
    image_uri=container,
    role=role,
    instance_count=1,
    instance_type='ml.m4.xlarge',
    volume_size=5,
    output_path=s3_output_location,
    sagemaker_session=sagemaker.Session(),
    rules=[
        Rule.sagemaker(rule_configs.create_xgboost_report()),
        ProfilerRule.sagemaker(rule_configs.ProfilerReport())
    ]
)
```

Untuk membangun SageMaker estimator, tentukan parameter berikut:

- `image_uri`— Tentukan URI gambar wadah pelatihan. Dalam contoh ini, URI wadah pelatihan SageMaker XGBoost ditentukan menggunakan. `sagemaker.image_uris.retrieve`
- `role`— Peran AWS Identity and Access Management (IAM) yang SageMaker digunakan untuk melakukan tugas atas nama Anda (misalnya, membaca hasil pelatihan, memanggil artefak model dari Amazon S3, dan menulis hasil pelatihan ke Amazon S3).
- `instance_count`dan `instance_type` — Jenis dan jumlah instans komputasi Amazon EC2 ML yang akan digunakan untuk pelatihan model. Untuk latihan ini, Anda menggunakan satu `ml.m4.xlarge` instans, yang memiliki 4 CPU, memori 16 GB, penyimpanan Amazon Elastic Block Store (Amazon EBS), dan kinerja jaringan yang tinggi. Untuk informasi selengkapnya tentang jenis instans komputasi EC2, lihat Jenis Instans Amazon EC2. Untuk informasi selengkapnya tentang penagihan, lihat SageMaker harga Amazon.
- `volume_size`— Ukuran, dalam GB, volume penyimpanan EBS untuk dilampirkan ke instance pelatihan. Ini harus cukup besar untuk menyimpan data pelatihan jika Anda menggunakan `File` mode (`File`mode aktif secara default). Jika Anda tidak menentukan parameter ini, nilainya default ke 30.

- `output_path`— Jalur ke ember S3 tempat SageMaker menyimpan artefak model dan hasil pelatihan.
- `sagemaker_session`Objek sesi yang mengelola interaksi dengan operasi SageMaker API dan AWS layanan lain yang digunakan oleh pekerjaan pelatihan.
- `rules`— Tentukan daftar aturan bawaan SageMaker Debugger. Dalam contoh ini, `create_xgboost_report()` aturan membuat laporan XGBoost yang memberikan wawasan tentang kemajuan dan hasil pelatihan, dan `ProfilerReport()` aturan tersebut membuat laporan mengenai pemanfaatan sumber daya komputasi EC2. Untuk informasi selengkapnya, lihat SageMaker Laporan Pelatihan Debugger XGBoost.

> **Tip**
>
> Jika Anda ingin menjalankan pelatihan terdistribusi model pembelajaran mendalam berukuran besar, seperti model jaringan saraf konvolusional (CNN) dan pemrosesan bahasa alami (NLP), gunakan SageMaker Distributed untuk paralelisme data atau paralelisme model. Untuk informasi selengkapnya, lihat Pelatihan terdistribusi di Amazon SageMaker.

3. Tetapkan hyperparameters untuk algoritma XGBoost dengan memanggil `set_hyperparameters` metode estimator. Untuk daftar lengkap hyperparameters XGBoost, lihat. XGBoost Hyperparameter

```
xgb_model.set_hyperparameters(
    max_depth = 5,
    eta = 0.2,
    gamma = 4,
    min_child_weight = 6,
    subsample = 0.7,
    objective = "binary:logistic",
    num_round = 1000
)
```

> **Tip**
>
> Anda juga dapat menyetel hyperparameters menggunakan fitur optimasi SageMaker hyperparameter. Untuk informasi selengkapnya, lihat Lakukan Penyetelan Model Otomatis dengan SageMaker.

4. Gunakan `TrainingInput` kelas untuk mengkonfigurasi aliran input data untuk pelatihan. Kode contoh berikut menunjukkan cara mengonfigurasi `TrainingInput` objek untuk menggunakan kumpulan data pelatihan dan validasi yang Anda unggah ke Amazon S3 di bagian tersebut. Pisahkan Dataset menjadi Train, Validation, dan Test Datasets

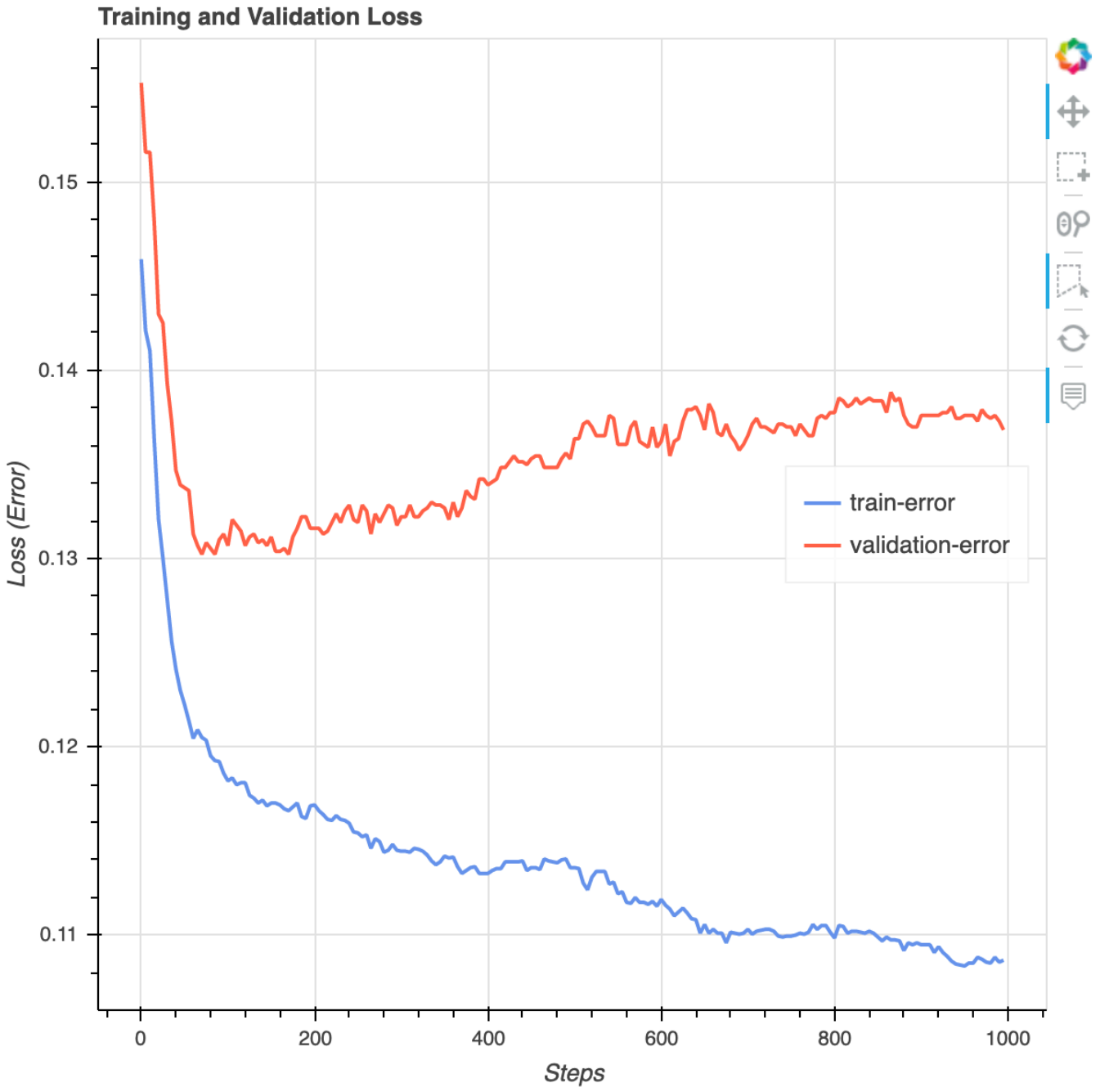
```
from sagemaker.session import TrainingInput

train_input = TrainingInput(
    "s3://{}/{}/{}".format(bucket, prefix, "data/train.csv"), content_type="csv"
)
validation_input = TrainingInput(
    "s3://{}/{}/{}".format(bucket, prefix, "data/validation.csv"),
 content_type="csv"
)
```

5. Untuk memulai pelatihan model, hubungi `fit` metode estimator dengan kumpulan data pelatihan dan validasi. Dengan pengaturan`wait=True`, `fit` metode ini menampilkan log kemajuan dan menunggu hingga pelatihan selesai.

```
xgb_model.fit({"train": train_input, "validation": validation_input}, wait=True)
```

Untuk informasi lebih lanjut tentang pelatihan model, lihatLatih Model dengan Amazon SageMaker. Pekerjaan pelatihan tutorial ini mungkin memakan waktu hingga 10 menit.

Setelah pekerjaan pelatihan selesai, Anda dapat mengunduh laporan pelatihan XGBoost dan laporan pembuatan profil yang dihasilkan oleh Debugger. SageMaker Laporan pelatihan XGBoost menawarkan Anda wawasan tentang kemajuan dan hasil pelatihan, seperti fungsi kerugian sehubungan dengan iterasi, kepentingan fitur, matriks kebingungan, kurva akurasi, dan hasil statistik pelatihan lainnya. Misalnya, Anda dapat menemukan kurva kerugian berikut dari laporan pelatihan XGBoost yang dengan jelas menunjukkan bahwa ada masalah overfitting.



Jalankan kode berikut untuk menentukan URI bucket S3 tempat laporan pelatihan Debugger dibuat dan periksa apakah laporan tersebut ada.

```
rule_output_path = xgb_model.output_path + "/" +
 xgb_model.latest_training_job.job_name + "/rule-output"
! aws s3 ls {rule_output_path} --recursive
```

Unduh laporan pelatihan dan pembuatan profil Debugger XGBoost ke ruang kerja saat ini:

```
! aws s3 cp {rule_output_path} ./ --recursive
```

Jalankan skrip IPython berikut untuk mendapatkan tautan file dari laporan pelatihan XGBoost:

```
from IPython.display import FileLink, FileLinks
display("Click link below to view the XGBoost Training report",
 FileLink("CreateXgboostReport/xgboost_report.html"))
```

Skrip IPython berikut mengembalikan tautan file dari laporan profil Debugger yang menunjukkan ringkasan dan detail pemanfaatan sumber daya instans EC2, hasil deteksi kemacetan sistem, dan hasil pembuatan profil operasi python:

```
profiler_report_name = [rule["RuleConfigurationName"]
                        for rule in
 xgb_model.latest_training_job.rule_job_summary()
                        if "Profiler" in rule["RuleConfigurationName"]][0]
profiler_report_name
display("Click link below to view the profiler report",
 FileLink(profiler_report_name+"/profiler-output/profiler-report.html"))
```

> **Tip**
>
> Jika laporan HTML tidak membuat plot dalam JupyterLab tampilan, Anda harus memilih Trust HTML di bagian atas laporan.
> Untuk mengidentifikasi masalah pelatihan, seperti overfitting, gradien menghilang, dan masalah lain yang mencegah model Anda dari konvergen, gunakan SageMaker Debugger dan lakukan tindakan otomatis saat membuat prototipe dan melatih model ML Anda. Untuk informasi selengkapnya, lihat Gunakan Amazon SageMaker Debugger untuk men-debug dan meningkatkan kinerja model. Untuk menemukan analisis lengkap parameter model, lihat buku catatan contoh Explainability with Amazon SageMaker Debugger.

Anda sekarang memiliki model XGBoost terlatih. SageMaker menyimpan artefak model di ember S3 Anda. Untuk menemukan lokasi artefak model, jalankan kode berikut untuk mencetak atribut model_data estimator: `xgb_model`

```
xgb_model.model_data
```

> **Tip**
>
> Untuk mengukur bias yang dapat terjadi selama setiap tahap siklus hidup ML (pengumpulan data, pelatihan dan penyetelan model, dan pemantauan model ML yang digunakan untuk prediksi), gunakan Clarify. SageMaker Untuk informasi selengkapnya, lihat Penjelasan Model. end-to-end Sebagai contoh, lihat contoh Keadilan dan Keterjelasan dengan SageMaker Clarify notebook.

Langkah 5: Menyebarkan Model ke Amazon EC2

Untuk mendapatkan prediksi, terapkan model Anda ke Amazon EC2 menggunakan Amazon. SageMaker

Topik

- Menyebarkan Model ke SageMaker Layanan Hosting
- (Opsional) Gunakan SageMaker Prediktor untuk Menggunakan Kembali Titik Akhir yang Dihosting
- (Opsional) Buat Prediksi dengan Batch Transform

Menyebarkan Model ke SageMaker Layanan Hosting

Untuk meng-host model melalui Amazon EC2 menggunakan Amazon SageMaker, terapkan model yang Anda latih Membuat dan Menjalankan Training Job dengan memanggil `deploy` metode estimator. `xgb_model` Ketika Anda memanggil `deploy` metode, Anda harus menentukan nomor dan jenis instance EC2 MLyang ingin Anda gunakan untuk hosting endpoint.

```
import sagemaker
from sagemaker.serializers import CSVSerializer
xgb_predictor=xgb_model.deploy(
    initial_instance_count=1,
    instance_type='ml.t2.medium',
    serializer=CSVSerializer()
```

```
)
```

- `initial_instance_count`(int) — Jumlah instance untuk menyebarkan model.
- `instance_type`(str) - Jenis instance yang ingin Anda operasikan model yang Anda gunakan.
- `serializer`(int) - Serialisasi data input dari berbagai format (NumPy array, daftar, file, atau buffer) ke string berformat CSV. Kami menggunakan ini karena algoritma XGBoost menerima file input dalam format CSV.

`deploy`Metode ini membuat model deployable, mengkonfigurasi endpoint layanan SageMaker hosting, dan meluncurkan endpoint untuk meng-host model. Untuk informasi selengkapnya, lihat metode class penerapan Estimator SageMaker generik di Amazon Python SageMaker SDK. Untuk mengambil nama endpoint yang dihasilkan oleh `deploy` metode, jalankan kode berikut:

```
xgb_predictor.endpoint_name
```

Ini harus mengembalikan nama titik akhir dari. `xgb_predictor` Format nama endpoint adalah`"sagemaker-xgboost-YYYY-MM-DD-HH-MM-SS-SSS"`. Endpoint ini tetap aktif dalam instance ML, dan Anda dapat membuat prediksi seketika kapan saja kecuali jika Anda mematikannya nanti. Salin nama titik akhir ini dan simpan untuk digunakan kembali dan buat prediksi real-time di tempat lain di instance SageMaker Studio atau SageMaker notebook.

> **Tip**
>
> Untuk mempelajari lebih lanjut tentang mengompilasi dan mengoptimalkan model Anda untuk penerapan ke instans Amazon EC2 atau perangkat edge, lihat Mengompilasi dan Menerapkan Model dengan Neo.

(Opsional) Gunakan SageMaker Prediktor untuk Menggunakan Kembali Titik Akhir yang Dihosting

Setelah menerapkan model ke titik akhir, Anda dapat menyiapkan SageMaker prediktor baru dengan memasangkan titik akhir dan terus membuat prediksi waktu nyata di buku catatan lain. Kode contoh berikut menunjukkan bagaimana menggunakan kelas SageMaker Predictor untuk mengatur objek prediktor baru menggunakan titik akhir yang sama. Gunakan kembali nama endpoint yang Anda gunakan untuk. `xgb_predictor`

```
import sagemaker
```

```
xgb_predictor_reuse=sagemaker.predictor.Predictor(
    endpoint_name="sagemaker-xgboost-YYYY-MM-DD-HH-MM-SS-SSS",
    sagemaker_session=sagemaker.Session(),
    serializer=sagemaker.serializers.CSVSerializer()
)
```

`xgb_predictor_reuse`Prediktor berperilaku persis sama dengan aslinya. `xgb_predictor` Untuk informasi selengkapnya, lihat kelas SageMaker Prediktor di Amazon SageMaker Python SDK.

(Opsional) Buat Prediksi dengan Batch Transform

Alih-alih menghosting titik akhir dalam produksi, Anda dapat menjalankan pekerjaan inferensi batch satu kali untuk membuat prediksi pada kumpulan data pengujian menggunakan transformasi batch. SageMaker Setelah pelatihan model Anda selesai, Anda dapat memperluas estimator ke `transformer` objek, yang didasarkan pada kelas SageMakerTransformer. Trafo batch membaca data input dari bucket S3 tertentu dan membuat prediksi.

Untuk menjalankan pekerjaan transformasi batch

1. Jalankan kode berikut untuk mengonversi kolom fitur kumpulan data pengujian menjadi file CSV dan unggah ke bucket S3:

   ```
   X_test.to_csv('test.csv', index=False, header=False)

   boto3.Session().resource('s3').Bucket(bucket).Object(
   os.path.join(prefix, 'test/test.csv')).upload_file('test.csv')
   ```

2. Tentukan URI bucket S3 dari input dan output untuk pekerjaan transformasi batch seperti yang ditunjukkan berikut:

   ```
   # The location of the test dataset
   batch_input = 's3://{}/{}/test'.format(bucket, prefix)

   # The location to store the results of the batch transform job
   batch_output = 's3://{}/{}/batch-prediction'.format(bucket, prefix)
   ```

3. Buat objek transformator yang menentukan jumlah minimal parameter: `instance_count` dan `instance_type` parameter untuk menjalankan pekerjaan transformasi batch, dan `output_path` untuk menyimpan data prediksi seperti yang ditunjukkan berikut:

   ```
   transformer = xgb_model.transformer(
   ```

```
    instance_count=1,
    instance_type='ml.m4.xlarge',
    output_path=batch_output
)
```

4. Memulai pekerjaan transformasi batch dengan mengeksekusi `transform()` metode `transformer` objek seperti yang ditunjukkan berikut:

```
transformer.transform(
    data=batch_input,
    data_type='S3Prefix',
    content_type='text/csv',
    split_type='Line'
)
transformer.wait()
```

5. Saat pekerjaan transformasi batch selesai, SageMaker buat data `test.csv.out` prediksi yang disimpan di `batch_output` jalur, yang seharusnya dalam format berikut:`s3://sagemaker-<region>-111122223333/demo-sagemaker-xgboost-adult-income-prediction/batch-prediction`. Jalankan yang berikut ini AWS CLI untuk mengunduh data keluaran dari pekerjaan transformasi batch:

```
! aws s3 cp {batch_output} ./ --recursive
```

   Ini harus membuat `test.csv.out` file di bawah direktori kerja saat ini. Anda akan dapat melihat nilai float yang diprediksi berdasarkan regresi logistik dari pekerjaan pelatihan XGBoost.

Langkah 6: Evaluasi Model

Sekarang setelah Anda melatih dan menerapkan model menggunakan Amazon SageMaker, evaluasi model tersebut untuk memastikan bahwa model tersebut menghasilkan prediksi yang akurat pada data baru. Untuk evaluasi model, gunakan kumpulan data pengujian yang Anda buat. Langkah 3: Unduh, Jelajahi, dan Ubah Dataset

Evaluasi Model yang Diterapkan ke SageMaker Layanan Hosting

Untuk mengevaluasi model dan menggunakannya dalam produksi, panggil titik akhir dengan kumpulan data pengujian dan periksa apakah kesimpulan yang Anda dapatkan mengembalikan akurasi target yang ingin Anda capai.

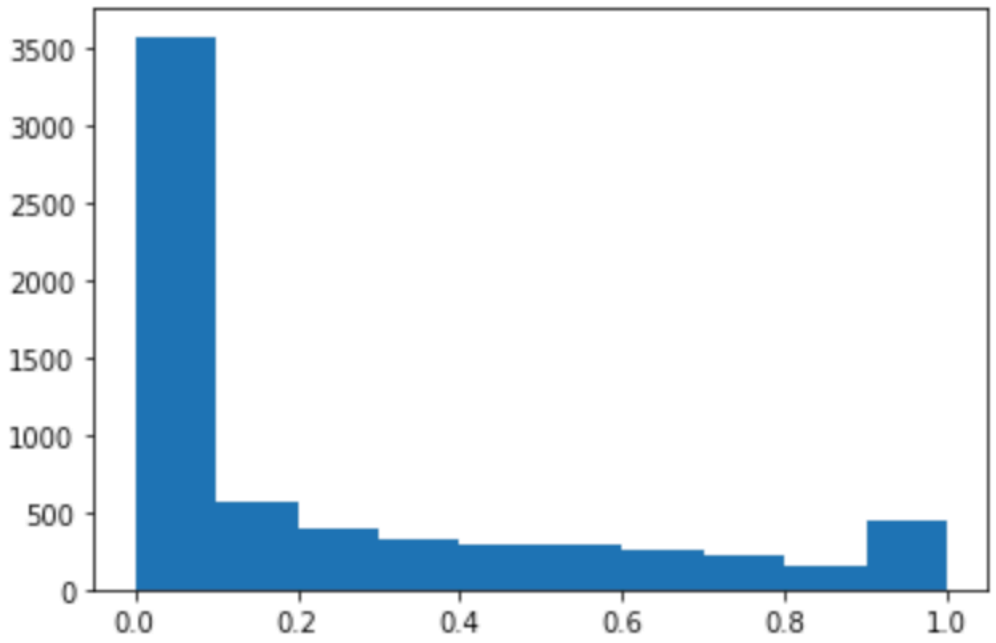
Untuk mengevaluasi model

1. Siapkan fungsi berikut untuk memprediksi setiap baris set tes. Dalam contoh kode berikut, `rows` argumennya adalah untuk menentukan jumlah baris yang akan diprediksi pada suatu waktu. Anda dapat mengubah nilainya untuk melakukan inferensi batch yang sepenuhnya memanfaatkan sumber daya perangkat keras instans.

```
import numpy as np
def predict(data, rows=1000):
    split_array = np.array_split(data, int(data.shape[0] / float(rows) + 1))
    predictions = ''
    for array in split_array:
        predictions = ','.join([predictions,
 xgb_predictor.predict(array).decode('utf-8')])
    return np.fromstring(predictions[1:], sep=',')
```

2. Jalankan kode berikut untuk membuat prediksi dataset pengujian dan plot histogram. Anda hanya perlu mengambil kolom fitur dari kumpulan data pengujian, tidak termasuk kolom ke-0 untuk nilai aktual.

```
import matplotlib.pyplot as plt

predictions=predict(test.to_numpy()[:,1:])
plt.hist(predictions)
plt.show()
```



3. Nilai yang diprediksi adalah tipe float. Untuk menentukan `True` atau `False` berdasarkan nilai float, Anda perlu menetapkan nilai cutoff. Seperti yang ditunjukkan pada contoh kode berikut, gunakan pustaka Scikit-learn untuk mengembalikan metrik kebingungan keluaran dan laporan klasifikasi dengan batas 0,5.

```
import sklearn

cutoff=0.5
print(sklearn.metrics.confusion_matrix(test.iloc[:, 0], np.where(predictions >
 cutoff, 1, 0)))
print(sklearn.metrics.classification_report(test.iloc[:, 0], np.where(predictions >
 cutoff, 1, 0)))
```

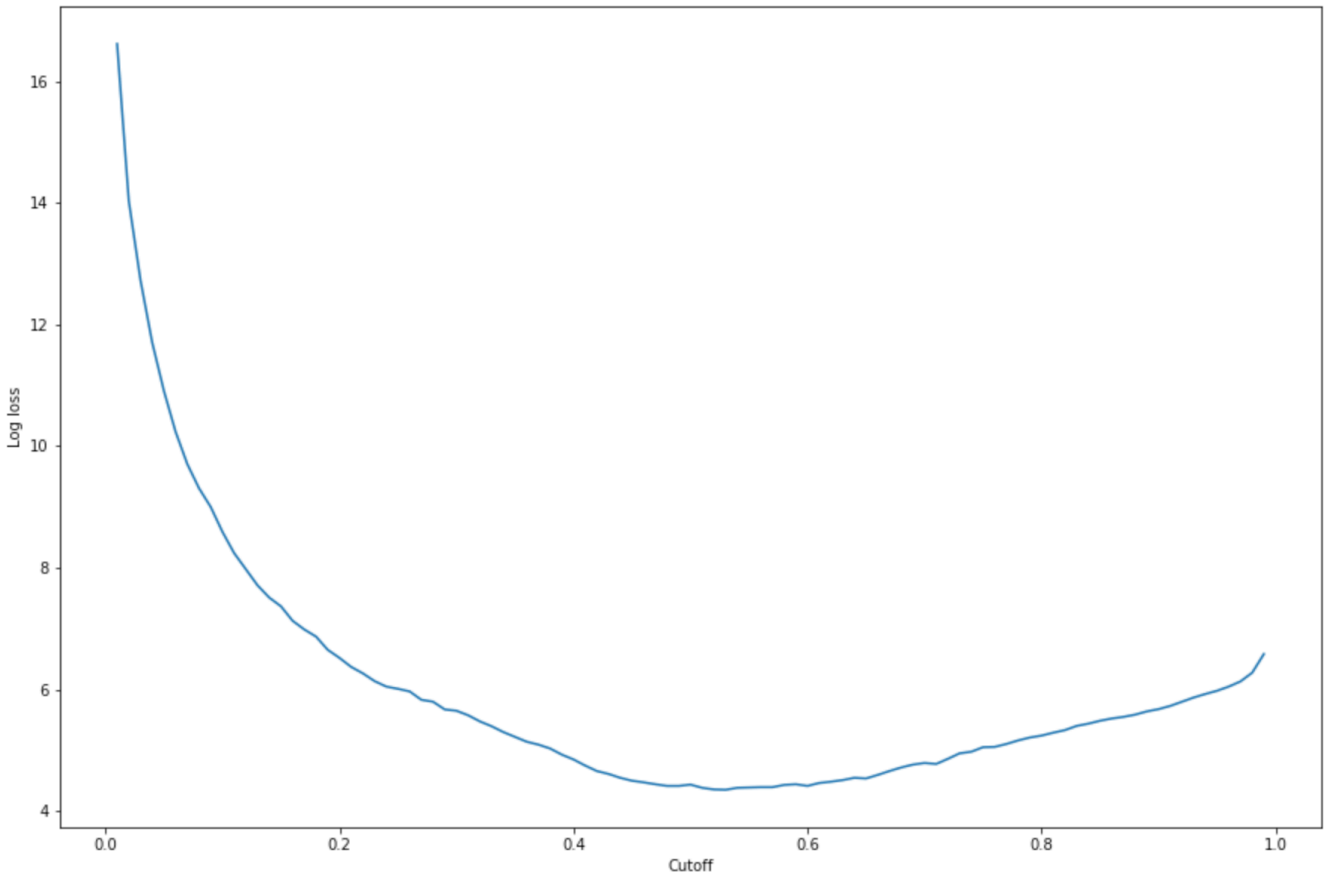
Ini akan mengembalikan matriks kebingungan berikut:

```
[[4670  356]
 [ 480 1007]]
              precision    recall  f1-score   support

           0       0.91      0.93      0.92      5026
           1       0.74      0.68      0.71      1487

    accuracy                           0.87      6513
   macro avg       0.82      0.80      0.81      6513
weighted avg       0.87      0.87      0.87      6513
```

4. Untuk menemukan cutoff terbaik dengan set pengujian yang diberikan, hitung fungsi kehilangan log dari regresi logistik. Fungsi kehilangan log didefinisikan sebagai kemungkinan log negatif dari model logistik yang mengembalikan probabilitas prediksi untuk label kebenaran dasarnya. Contoh kode berikut secara numerik dan iteratif menghitung nilai kehilangan log (`-(y*log(p)+(1-y)log(1-p)`), di mana label sebenarnya dan `y p` merupakan perkiraan probabilitas dari sampel uji yang sesuai. Ini mengembalikan log loss versus grafik cutoff.

```
import matplotlib.pyplot as plt

cutoffs = np.arange(0.01, 1, 0.01)
log_loss = []
for c in cutoffs:
    log_loss.append(
        sklearn.metrics.log_loss(test.iloc[:, 0], np.where(predictions > c, 1, 0))
    )

plt.figure(figsize=(15,10))
plt.plot(cutoffs, log_loss)
plt.xlabel("Cutoff")
plt.ylabel("Log loss")
plt.show()
```

Ini harus mengembalikan kurva kehilangan log berikut.



5. Temukan titik minimum kurva kesalahan menggunakan `min` fungsi NumPy `argmin` dan:

```
print(
    'Log loss is minimized at a cutoff of ', cutoffs[np.argmin(log_loss)],
    ', and the log loss value at the minimum is ', np.min(log_loss)
)
```

Ini harus kembali:`Log loss is minimized at a cutoff of 0.53, and the log loss value at the minimum is 4.348539186773897`.

Alih-alih menghitung dan meminimalkan fungsi kehilangan log, Anda dapat memperkirakan fungsi biaya sebagai alternatif. Misalnya, jika Anda ingin melatih model untuk melakukan klasifikasi biner untuk masalah bisnis seperti masalah prediksi churn pelanggan, Anda dapat mengatur bobot ke elemen matriks kebingungan dan menghitung fungsi biaya yang sesuai.

Anda sekarang telah melatih, menerapkan, dan mengevaluasi model pertama Anda. SageMaker

> **Tip**
>
> Untuk memantau kualitas model, kualitas data, dan penyimpangan bias, gunakan Amazon SageMaker Model Monitor dan SageMaker Clarify. Untuk mempelajari lebih lanjut, lihat Monitor SageMaker Model Amazon, Monitor Kualitas Data, Kualitas Model Monitor, Monitor Bias Drift, dan Monitor Fitur Atribusi Drift.

> **Tip**
>
> Untuk mendapatkan tinjauan manusia tentang prediksi ML kepercayaan rendah atau sampel prediksi acak, gunakan alur kerja tinjauan manusia Amazon Augmented AI. Untuk informasi selengkapnya, lihat Menggunakan Amazon Augmented AI for Human Review.

Langkah 7: Bersihkan

Untuk menghindari biaya yang tidak perlu, gunakan file AWS Management Console untuk menghapus titik akhir dan sumber daya yang Anda buat saat menjalankan latihan.

> **Note**
>
> Pekerjaan dan log pelatihan tidak dapat dihapus dan dipertahankan tanpa batas waktu.

> **Note**
>
> Jika Anda berencana untuk menjelajahi latihan lain dalam panduan ini, Anda mungkin ingin menyimpan beberapa sumber daya ini, seperti instance notebook, bucket S3, dan peran IAM.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/ dan hapus sumber daya berikut:
   - Titik akhir. Menghapus endpoint juga akan menghapus instance komputasi atau instance yang mendukungnya.
     1. Di bawah Inferensi, pilih Titik Akhir.

2. Pilih titik akhir yang Anda buat dalam contoh, pilih Tindakan, lalu pilih Hapus.

- Konfigurasi titik akhir.

  1. Di bawah Inferensi, pilih Konfigurasi titik akhir.
  2. Pilih konfigurasi titik akhir yang Anda buat dalam contoh, pilih Tindakan, lalu pilih Hapus.

- Model.

  1. Di bawah Inferensi, pilih Model.
  2. Pilih model yang Anda buat dalam contoh, pilih Tindakan, lalu pilih Hapus.

- Contoh notebook. Sebelum menghapus instance notebook, hentikan.

  1. Di bawah Notebook, pilih Instans Notebook.
  2. Pilih instance buku catatan yang Anda buat dalam contoh, pilih Tindakan, lalu pilih Berhenti. Contoh notebook membutuhkan waktu beberapa menit untuk berhenti. Ketika Status berubah menjadi Berhenti, lanjutkan ke langkah berikutnya.
  3. Pilih Tindakan, lalu pilih Hapus.

2. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/, lalu hapus bucket yang Anda buat untuk menyimpan artefak model dan kumpulan data pelatihan.
3. Buka CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/, lalu hapus semua grup log yang memiliki nama dimulai`/aws/sagemaker/`.

Instans notebook Amazon Linux 2

Instans SageMaker notebook Amazon saat ini mendukung sistem operasi Amazon Linux 2 (AL2). Anda dapat memilih sistem operasi yang menjadi dasar instans notebook Anda saat membuat instance notebook.

SageMaker mendukung instance notebook berdasarkan sistem operasi Amazon Linux 2 berikut.

- notebook-al2-v1: Instans notebook ini mendukung versi 1. JupyterLab Untuk informasi tentang JupyterLab versi, lihatJupyterLab pembuatan versi.
- notebook-al2-v2: Instans notebook ini mendukung versi 3. JupyterLab Untuk informasi tentang JupyterLab versi, lihatJupyterLab pembuatan versi.

Instans notebook yang dibuat sebelum 08/18/2021 secara otomatis berjalan di Amazon Linux (AL1). Instans notebook berdasarkan AL1 memasuki fase pemeliharaan pada 12/01/2022 dan tidak lagi

tersedia untuk pembuatan instans notebook baru mulai 02/01/2023. Untuk mengganti AL1, Anda sekarang memiliki opsi untuk membuat instance SageMaker notebook Amazon dengan AL2. Untuk informasi selengkapnya, lihat Rencana Fase Pemeliharaan AL1.

Topik

- Tipe instans yang didukung
- Kernel yang tersedia
- Rencana Fase Pemeliharaan AL1

Tipe instans yang didukung

Amazon Linux 2 mendukung jenis instans yang tercantum dalam Instans Notebook di Amazon SageMaker Pricing dengan pengecualian bahwa Amazon Linux 2 tidak mendukung `ml.p2` instans.

Kernel yang tersedia

Tabel berikut memberikan informasi tentang kernel yang tersedia untuk instance SageMaker notebook. Semua gambar ini didukung pada instance notebook berdasarkan sistem `notebook-al2-v2` operasi `notebook-al2-v1` dan sistem operasi.

SageMaker kernel contoh notebook

Nama kernel	Deskripsi
R	Kernel yang digunakan untuk melakukan analisis data dan visualisasi menggunakan kode R dari notebook Jupyter.
Sparkmagic () PySpark	Kernel digunakan untuk melakukan ilmu data dengan cluster Spark jarak jauh dari notebook Jupyter menggunakan bahasa pemrograman Python. Kernel ini dilengkapi dengan Python 3.10.
Sparkmagic (Percikan)	Kernel digunakan untuk melakukan ilmu data dengan cluster Spark jarak jauh dari notebook Jupyter menggunakan bahasa pemrograman

Nama kernel	Deskripsi
	Scala. Kernel ini dilengkapi dengan Python 3.10.
Sparkmagic (SparkR)	Kernel digunakan untuk melakukan ilmu data dengan cluster Spark jarak jauh dari notebook Jupyter menggunakan bahasa pemrograman R. Kernel ini dilengkapi dengan Python 3.10.
conda_python3	Lingkungan conda yang sudah diinstal sebelumnya dengan paket populer untuk ilmu data dan pembelajaran mesin. Kernel ini dilengkapi dengan Python 3.10.
conda_pytorch_p310	Lingkungan conda yang sudah diinstal sebelumnya dengan PyTorch versi 2.0.1, serta paket ilmu data dan pembelajaran mesin yang populer. Kernel ini dilengkapi dengan Python 3.10.
conda_tensorflow2_p310	Lingkungan conda yang sudah diinstal sebelumnya dengan TensorFlow versi 2.13, serta paket ilmu data dan pembelajaran mesin yang populer. Kernel ini dilengkapi dengan Python 3.10.

Rencana Fase Pemeliharaan AL1

Tabel berikut adalah garis waktu ketika AL1 memasuki fase pemeliharaan yang diperpanjang. Fase pemeliharaan AL1 juga bertepatan dengan penghentian Python 2 dan Chainer. Notebook berdasarkan AL2 tidak memiliki kernel Python 2 dan Chainer yang dikelola.

Tanggal	Deskripsi
08/18/2021	Instans notebook berdasarkan AL2 diluncurk an. Instans notebook yang baru diluncurkan masih default ke AL1. AL1 didukung dengan

Tanggal	Deskripsi
	patch dan pembaruan keamanan, tetapi tidak ada fitur baru. Anda dapat memilih antara dua sistem operasi saat meluncurkan instance notebook baru.
10/31/2022	Pengenal platform default untuk instance SageMaker notebook berubah dari Amazon Linux (al1-v1) ke Amazon Linux 2 (al2-v2). Anda dapat memilih antara dua sistem operasi saat meluncurkan instance notebook baru.
12/01/2022	AL1 tidak lagi didukung dengan patch dan pembaruan keamanan non-kritis. AL1 masih menerima perbaikan untuk masalah terkait keamanan kritis. Anda masih dapat meluncurkan instans di AL1, tetapi menanggung risiko yang terkait dengan penggunaan sistem operasi yang tidak didukung.
02/01/2023	AL1 tidak lagi menjadi opsi yang tersedia untuk pembuatan instance notebook baru. Setelah tanggal ini, pelanggan dapat membuat instance notebook dengan pengenal platform AL2. Instans notebook al1-v1 yang ada tidak terpengaruh.

Migrasi ke Amazon Linux 2

Instans notebook AL1 Anda yang ada tidak dimigrasikan secara otomatis ke Amazon Linux 2. Untuk memutakhirkan instans notebook AL1 Anda ke Amazon Linux 2, Anda harus membuat instance notebook baru, mereplikasi kode dan lingkungan, dan menghapus instance notebook lama Anda. Untuk informasi selengkapnya, lihat blog migrasi Amazon Linux 2.

JupyterLab pembuatan versi

Antarmuka instance SageMaker notebook Amazon didasarkan pada JupyterLab, yang merupakan lingkungan pengembangan interaktif berbasis web untuk notebook, kode, dan data. Notebook sekarang mendukung penggunaan JupyterLab 1 atau JupyterLab 3. Sebuah instance notebook tunggal dapat menjalankan satu instance JupyterLab (paling banyak). Anda dapat memiliki beberapa instance notebook dengan JupyterLab versi yang berbeda.

Anda dapat mengonfigurasi notebook Anda untuk menjalankan JupyterLab versi pilihan Anda dengan memilih pengenal platform yang sesuai. Gunakan salah satu AWS CLI atau SageMaker konsol saat membuat instance notebook Anda. Untuk informasi selengkapnya tentang pengenal platform, lihat instans notebook Amazon Linux 2 vs Amazon Linux. Jika Anda tidak mengonfigurasi pengenal platform secara eksplisit, instance notebook Anda secara default menjalankan 1. JupyterLab

Topik

- JupyterLab 3
- Membuat buku catatan dengan JupyterLab versi Anda
- Lihat JupyterLab versi notebook dari konsol

JupyterLab 3

JupyterLab Dukungan 3 hanya tersedia di platform sistem operasi Amazon Linux 2. JupyterLab 3 termasuk fitur berikut yang tidak tersedia di JupyterLab 1. Untuk informasi selengkapnya tentang fitur-fitur ini, lihat JupyterLab 3.0 dirilis! .

- Visual debugger saat menggunakan kernel berikut:
  - conda_pytorch_p38
  - conda_tensorflow2_p38
  - conda_amazonei_pytorch_latest_p37
- Filter peramban file
- Daftar Isi (TOC)
- Dukungan multi-bahasa
- Modus sederhana
- Mode antarmuka tunggal
- Mengedit file SVG langsung dengan rendering yang diperbarui

- Antarmuka pengguna untuk tag sel notebook

Perubahan penting ke JupyterLab 3

Untuk informasi tentang perubahan penting saat menggunakan JupyterLab 3, lihat log JupyterLab perubahan berikut:

- v2.0.0
- v3.0.0

Perubahan versi Package

JupyterLab 3 memiliki versi paket berikut berubah dari JupyterLab 1:

- JupyterLab telah ditingkatkan dari 1.x ke 3.x.
- Notebook Jupyter telah ditingkatkan dari 5.x ke 6.x.
- jupyterlab-git telah diperbarui ke versi 0.37.1.
- nbserverproxy 0.x (0.3.2) telah diganti dengan 3.x (3.2.1). jupyter-server-proxy

Membuat buku catatan dengan JupyterLab versi Anda

Anda dapat memilih JupyterLab versi saat membuat instance notebook dari konsol mengikuti langkah-langkahnyaMembuat Instance Notebook.

Anda juga dapat memilih JupyterLab versi dengan meneruskan `platform-identifier` parameter saat membuat instance notebook Anda menggunakan AWS CLI sebagai berikut:

```
create-notebook-instance --notebook-instance-name <NEW_NOTEBOOK_NAME> \
--instance-type <INSTANCE_TYPE> \
--role-arn <YOUR_ROLE_ARN> \
--platform-identifier <PLATFORM_TO_USE>
```

Lihat JupyterLab versi notebook dari konsol

Anda dapat melihat JupyterLab versi notebook menggunakan prosedur berikut:

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Dari navigasi kiri, pilih Notebook.

3. Dari menu tarik-turun, pilih instance Notebook untuk menavigasi ke halaman instance Notebook.
4. Dari daftar instance notebook, pilih nama instans notebook Anda.
5. Pada halaman pengaturan instans Notebook, lihat Pengenal Platform untuk melihat JupyterLab versi buku catatan.

Membuat Instance Notebook

Instance SageMaker notebook Amazon adalah instance komputasi ML yang menjalankan Aplikasi Notebook Jupyter. SageMaker mengelola pembuatan instance dan sumber daya terkait. Gunakan notebook Jupyter dalam instance notebook Anda untuk menyiapkan dan memproses data, menulis kode untuk melatih model, menyebarkan model ke SageMaker hosting, dan menguji atau memvalidasi model Anda.

Untuk membuat instance notebook, gunakan SageMaker konsol atau `CreateNotebookInstance`API.

Jenis instans notebook yang Anda pilih bergantung pada cara Anda menggunakan instance notebook Anda. Anda ingin memastikan bahwa instance notebook Anda tidak terikat oleh memori, CPU, atau IO. Jika Anda berencana untuk memuat kumpulan data ke dalam memori pada instance notebook untuk eksplorasi atau pra-pemrosesan, sebaiknya Anda memilih jenis instans dengan memori RAM yang cukup untuk kumpulan data Anda. Ini akan membutuhkan instance dengan setidaknya 16 GB memori (.xlarge atau lebih besar). Jika Anda berencana menggunakan notebook untuk pra-pemrosesan intensif komputasi, kami sarankan Anda memilih instance yang dioptimalkan komputasi seperti c4 atau c5.

Praktik terbaik saat menggunakan SageMaker notebook adalah dengan menggunakan instance notebook untuk mengatur layanan lain. AWS Misalnya, Anda dapat menggunakan instance notebook untuk mengelola pemrosesan kumpulan data besar dengan melakukan panggilan ke AWS Glue untuk layanan ETL (ekstrak, transformasi, dan muat) atau Amazon EMR untuk pemetaan dan pengurangan data menggunakan Hadoop. Anda dapat menggunakan AWS layanan sebagai bentuk perhitungan atau penyimpanan sementara untuk data Anda.

Anda dapat menyimpan dan mengambil data pelatihan dan pengujian menggunakan bucket Amazon S3. Anda kemudian dapat menggunakannya SageMaker untuk melatih dan membangun model Anda, sehingga jenis instance notebook Anda tidak akan berpengaruh pada kecepatan pelatihan dan pengujian model Anda.

Setelah menerima permintaan, SageMaker lakukan hal berikut:

- Membuat antarmuka jaringan —Jika Anda memilih konfigurasi VPC opsional SageMaker , buat antarmuka jaringan di VPC Anda. Ini menggunakan ID subnet yang Anda berikan dalam permintaan untuk menentukan Availability Zone mana untuk membuat subnet. SageMaker mengaitkan grup keamanan yang Anda berikan dalam permintaan dengan subnet. Untuk informasi selengkapnya, lihat Hubungkan Instance Notebook di VPC ke Sumber Daya Eksternal.
- Meluncurkan instance komputasi SageMaker ML-meluncurkan instance komputasi HTML di VPC. SageMaker SageMaker melakukan tugas konfigurasi yang memungkinkannya mengelola instance notebook Anda, dan jika Anda menentukan VPC Anda, ini memungkinkan lalu lintas antara VPC Anda dan instance notebook.
- Menginstal paket dan pustaka Anaconda untuk platform pembelajaran mendalam umum — SageMaker menginstal semua paket Anaconda yang disertakan dalam penginstal. Untuk informasi lebih lanjut, lihat daftar paket Anaconda. Selain itu, SageMaker instal perpustakaan pembelajaran mendalam TensorFlow dan Apache MXNet.
- Melampirkan volume penyimpanan SageMaker ML-melampirkan volume penyimpanan ML ke instans komputasi ML. Anda dapat menggunakan volume sebagai area kerja untuk membersihkan kumpulan data pelatihan atau untuk sementara menyimpan validasi, pengujian, atau data lainnya. Pilih ukuran apa pun antara 5 GB dan 16384 GB, dengan peningkatan 1 GB, untuk volume. Defaultnya adalah 5 GB. Volume penyimpanan ML dienkripsi, sehingga tidak SageMaker dapat menentukan jumlah ruang kosong yang tersedia pada volume. Karena itu, Anda dapat meningkatkan ukuran volume saat memperbarui instance notebook, tetapi Anda tidak dapat mengurangi ukuran volume. Jika Anda ingin mengurangi ukuran volume penyimpanan ML yang digunakan, buat instance notebook baru dengan ukuran yang diinginkan.

  Hanya file dan data yang disimpan dalam `/home/ec2-user/SageMaker` folder yang bertahan di antara sesi instance notebook. File dan data yang disimpan di luar direktori ini akan ditimpa saat instance notebook berhenti dan dimulai ulang. Setiap direktori /tmp instance notebook menyediakan minimal 10 GB penyimpanan di penyimpanan instance. Penyimpanan instance adalah penyimpanan sementara, tingkat blok yang tidak persisten. Ketika instance dihentikan atau dimulai ulang, SageMaker menghapus isi direktori. Penyimpanan sementara ini adalah bagian dari volume root dari instance notebook.
- Contoh salinan notebook Jupyter - Contoh kode Python ini menggambarkan pelatihan model dan latihan hosting menggunakan berbagai algoritme dan kumpulan data pelatihan.

Untuk membuat instance SageMaker notebook:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.

2. Pilih instance Notebook, lalu pilih Buat instance notebook.
3. Pada halaman Create notebook instance, berikan informasi berikut:

   a. Untuk nama instance Notebook, ketikkan nama untuk instance notebook Anda.

   b. Untuk jenis instans Notebook, pilih ukuran instans yang sesuai untuk kasus penggunaan Anda. Untuk daftar jenis dan kuota instans yang didukung, lihat Amazon SageMaker Service Quotas.

   c. Untuk Elastic Inference, pilih jenis akselerator inferensi untuk diasosiasikan dengan instance notebook jika Anda berencana untuk melakukan inferensi dari instance notebook, atau memilih none. Untuk informasi tentang inferensi elastis, lihatGunakan Amazon SageMaker Elastic Inference (EI) .

   d. Untuk Pengenal Platform, pilih jenis platform untuk membuat instance notebook. Jenis platform ini menentukan Sistem Operasi dan JupyterLab versi yang membuat instance notebook Anda. Untuk informasi tentang jenis pengenal platform, lihatInstans notebook Amazon Linux 2. Untuk informasi tentang versi JupyterLab, lihat JupyterLab pembuatan versi.

   e. (Opsional) Konfigurasi tambahan memungkinkan pengguna tingkat lanjut membuat skrip shell yang dapat dijalankan saat Anda membuat atau memulai instance. Skrip ini, yang disebut skrip konfigurasi siklus hidup, dapat digunakan untuk mengatur lingkungan untuk notebook atau untuk melakukan fungsi lainnya. Untuk informasi, lihat Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup.

   f. (Opsional) Konfigurasi tambahan juga memungkinkan Anda menentukan ukuran, dalam GB, volume penyimpanan ML yang dilampirkan ke instance notebook. Anda dapat memilih ukuran antara 5 GB dan 16.384 GB, dengan peningkatan 1 GB. Anda dapat menggunakan volume untuk membersihkan dataset pelatihan atau untuk sementara menyimpan validasi atau data lainnya.

   g. (Opsional) Untuk Versi IMDS Minimum, pilih versi dari daftar dropdown. Jika nilai ini diatur ke v1, kedua versi dapat digunakan dengan instance notebook. Jika v2 dipilih, maka hanya IMDSv2 yang dapat digunakan dengan instance notebook. Untuk informasi tentang IMDSv2, lihat Menggunakan IMDSv2.

      > **Note**
      >
      > Mulai 31 Oktober 2022, Versi IMDS minimum default untuk instance SageMaker notebook berubah dari IMDSv1 ke IMDSv2.

> Mulai 1 Februari 2023, IMDSv1 tidak lagi tersedia untuk pembuatan instance notebook baru. Setelah tanggal ini, Anda dapat membuat instance notebook dengan versi IMDS minimum 2.

h. Untuk peran IAM, pilih salah satu peran IAM yang ada di akun Anda yang memiliki izin yang diperlukan untuk mengakses SageMaker sumber daya atau pilih Buat peran baru. Jika Anda memilih Buat peran baru, SageMaker buat peran IAM bernama`AmazonSageMaker-ExecutionRole-`*`YYYYMMDD`*`T`*`HHmmSS`*. Kebijakan AWS `AmazonSageMakerFullAccess` terkelola melekat pada peran tersebut. Peran ini memberikan izin yang memungkinkan instance notebook menelepon SageMaker dan Amazon S3.

i. Untuk akses Root, untuk mengaktifkan akses root untuk semua pengguna instance notebook, pilih Aktifkan. Untuk menonaktifkan akses root bagi pengguna, pilih Nonaktifkan. Jika Anda mengaktifkan akses root, semua pengguna instance notebook memiliki hak administrator dan dapat mengakses dan mengedit semua file di dalamnya.

j. (Opsional) Kunci enkripsi memungkinkan Anda mengenkripsi data pada volume penyimpanan ML yang dilampirkan ke instance notebook menggunakan kunci AWS Key Management Service (AWS KMS). Jika Anda berencana untuk menyimpan informasi sensitif pada volume penyimpanan ML, pertimbangkan untuk mengenkripsi informasi.

k. (Opsional) Network memungkinkan Anda menempatkan instance notebook Anda di dalam Virtual Private Cloud (VPC). VPC memberikan keamanan tambahan dan membatasi akses ke sumber daya di VPC dari sumber di luar VPC. Untuk informasi selengkapnya tentang VPC, lihat Panduan Pengguna Amazon VPC.

   Untuk menambahkan instance notebook Anda ke VPC:

   i. Pilih VPC dan a. SubnetId

   ii. Untuk Grup Keamanan, pilih grup keamanan default VPC Anda.

   iii. Jika Anda memerlukan instance notebook Anda untuk memiliki akses internet, aktifkan akses internet langsung. Untuk akses internet langsung, pilih Aktifkan. Akses internet dapat membuat instance notebook Anda kurang aman. Untuk informasi selengkapnya, lihat Hubungkan Instance Notebook di VPC ke Sumber Daya Eksternal.

l. (Opsional) Untuk mengaitkan repositori Git dengan instance notebook, pilih repositori default dan hingga tiga repositori tambahan. Untuk informasi selengkapnya, lihat Kaitkan Repositori Git dengan SageMaker Instans Notebook.

m. Pilih Buat instans notebook.

Dalam beberapa menit, Amazon SageMaker meluncurkan instans komputasi ML—dalam hal ini, instance notebook—dan melampirkan volume penyimpanan ML ke dalamnya. Instance notebook memiliki server notebook Jupyter yang telah dikonfigurasi sebelumnya dan satu set pustaka Anaconda. Untuk informasi lebih lanjut, lihat `CreateNotebookInstance`API.

4. Ketika status instance notebook`InService`, di konsol, instance notebook siap digunakan. Pilih Buka Jupyter di sebelah nama notebook untuk membuka dasbor Jupyter klasik.

> **Note**
>
> Untuk meningkatkan keamanan instans SageMaker notebook Amazon Anda, semua *`notebook`*`.`*`region`*`.sagemaker.aws` domain regional terdaftar di Daftar Akhiran Publik (PSL) internet. Untuk keamanan lebih lanjut, kami menyarankan Anda menggunakan cookie dengan `__Host-` awalan jika Anda perlu mengatur cookie sensitif untuk domain instance SageMaker buku catatan Anda. Ini membantu mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat halaman Set-Cookie di situs web dokumentasi pengembang mozilla.org.

Anda dapat memilih Buka JupyterLab untuk membuka JupyterLab dasbor. Dasbor menyediakan akses ke instance notebook Anda dan contoh SageMaker buku catatan yang berisi panduan kode lengkap. Panduan ini menunjukkan cara menggunakan SageMaker untuk melakukan tugas pembelajaran mesin yang umum. Untuk informasi selengkapnya, lihat Contoh Notebook. Untuk informasi selengkapnya, lihat Kontrol akses root ke instance SageMaker notebook.

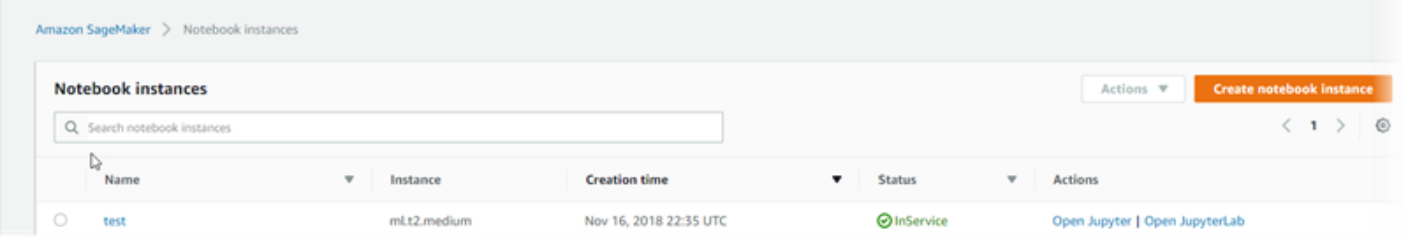
Untuk informasi lebih lanjut tentang notebook Jupyter, lihat Notebook Jupyter.

Akses Instans Notebook

Untuk mengakses instans SageMaker notebook Amazon Anda, pilih salah satu opsi berikut:

- Gunakan konsol .

  Pilih instans Notebook. Konsol menampilkan daftar instance notebook di akun Anda. Untuk membuka instance notebook dengan antarmuka Jupyter standar, pilih Open Jupyter untuk instance itu. Untuk membuka instance notebook dengan JupyterLab antarmuka, pilih Buka JupyterLab untuk instance itu.



Konsol menggunakan kredenal masuk Anda untuk mengirim `CreatePresignedNotebookInstanceUrl`Permintaan API ke SageMaker. SageMaker mengembalikan URL untuk instance notebook Anda, dan konsol membuka URL di tab browser lain dan menampilkan dasbor notebook Jupyter.

> **Note**
>
> URL yang Anda dapatkan dari panggilan ke `CreatePresignedNotebookInstanceUrl`hanya berlaku selama 5 menit. Jika Anda mencoba menggunakan URL setelah batas 5 menit berakhir, Anda akan diarahkan ke halaman AWS Management Console masuk.

- Gunakan API.

  Untuk mendapatkan URL untuk instance notebook, panggil `CreatePresignedNotebookInstanceUrl`API dan gunakan URL yang dikembalikan API untuk membuka instance notebook.

Gunakan dasbor notebook Jupyter untuk membuat dan mengelola buku catatan dan menulis kode. Untuk informasi lebih lanjut tentang notebook Jupyter, lihat http://jupyter.org/documentation.html.

Memperbarui Instance Notebook

Setelah membuat instance notebook, Anda dapat memperbaruinya menggunakan SageMaker konsol dan operasi `UpdateNotebookInstance`API.

Anda dapat memperbarui tag instance notebook yaitu`InService`. Untuk memperbarui atribut lain dari instance notebook, statusnya harus`Stopped`.

Untuk memperbarui instance notebook di SageMaker konsol:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.

2. Pilih instans Notebook.
3. Pilih instance notebook yang ingin Anda perbarui dengan memilih nama instance notebook dari daftar.
4. Jika Status buku catatan Anda tidak`Stopped`, pilih tombol Stop untuk menghentikan instance notebook.

   Ketika Anda melakukan ini, status instance notebook berubah menjadi`Stopping`. Tunggu hingga status berubah `Stopped` untuk menyelesaikan langkah-langkah berikut.
5. Pilih tombol Edit untuk membuka halaman contoh Edit notebook. Untuk informasi tentang properti notebook yang dapat Anda perbarui, lihatMembuat Instance Notebook.
6. Perbarui instance notebook Anda dan pilih tombol Perbarui instans notebook di bagian bawah halaman setelah Anda selesai untuk kembali ke halaman instance notebook. Status instans buku catatan Anda berubah menjadi Memperbarui.

   Ketika pembaruan instans notebook selesai, status berubah menjadi`Stopped`.

Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup

Untuk menginstal paket atau contoh notebook pada instance notebook Anda, mengkonfigurasi jaringan dan keamanan untuk itu, atau menggunakan skrip shell untuk menyesuaikannya, gunakan konfigurasi siklus hidup. Konfigurasi siklus hidup menyediakan skrip shell yang berjalan hanya saat Anda membuat instance notebook atau kapan pun Anda memulainya. Saat membuat instance notebook, Anda dapat membuat konfigurasi siklus hidup baru dan skrip yang digunakannya atau menerapkannya yang sudah Anda miliki.

Anda juga dapat menggunakan skrip konfigurasi siklus hidup untuk mengakses AWS layanan dari buku catatan Anda. Misalnya, Anda dapat membuat skrip yang memungkinkan Anda menggunakan buku catatan untuk mengontrol AWS sumber daya lain, seperti instans EMR Amazon.

Kami memelihara repositori publik skrip konfigurasi siklus hidup notebook yang menangani kasus penggunaan umum untuk menyesuaikan instance notebook di https://github.com/aws-samples/ -. amazon-sagemaker-notebook-instance lifecycle-configuration-samples

> **Note**
>
> Setiap skrip memiliki batas 16384 karakter.

Nilai variabel `$PATH` lingkungan yang tersedia untuk kedua skrip adalah`/usr/local/sbin:/usr/local/bin:/usr/bin:/usr/sbin:/sbin:/bin`. Direktori kerja, yang merupakan nilai variabel `$PWD` lingkungan, adalah`/`.
Lihat CloudWatch Log untuk konfigurasi siklus hidup instance notebook di grup `/aws/sagemaker/NotebookInstances` log dalam aliran log. `[notebook-instance-name]/[LifecycleConfigHook]`
Skrip tidak dapat berjalan lebih dari 5 menit. Jika skrip berjalan lebih dari 5 menit, skrip gagal dan instance notebook tidak dibuat atau dimulai. Untuk membantu mengurangi waktu berjalan skrip, coba yang berikut ini:

- Kurangi langkah-langkah yang diperlukan. Misalnya, batasi lingkungan conda mana untuk menginstal paket besar.
- Jalankan tugas dalam proses paralel.
- Gunakan `nohup` perintah dalam skrip Anda.

Anda dapat melihat daftar konfigurasi siklus hidup instance notebook yang sebelumnya dibuat dengan memilih Konfigurasi Siklus Hidup di konsol. SageMaker Anda dapat melampirkan konfigurasi siklus hidup instance notebook saat membuat instance notebook baru. Untuk informasi selengkapnya tentang membuat instance notebook, lihatMembuat Instance Notebook.

Untuk membuat konfigurasi siklus hidup

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Konfigurasi Siklus Hidup.
4. Dari halaman Konfigurasi Siklus Hidup, pilih tab Instance Notebook.
5. Pilih Buat konfigurasi.
6. Untuk Nama, ketikkan nama menggunakan karakter alfanumerik dan “-”, tetapi tidak ada spasi. Nama dapat memiliki maksimal 63 karakter.
7. (Opsional) Untuk membuat skrip yang berjalan saat Anda membuat buku catatan dan setiap kali Anda memulainya, pilih Mulai buku catatan.
8. Di editor Start notebook, ketik skrip.
9. (Opsional) Untuk membuat skrip yang berjalan hanya sekali, saat Anda membuat buku catatan, pilih Buat buku catatan.

10. Di editor Buat notebook, ketik skrip konfigurasi jaringan.
11. Pilih Buat konfigurasi.

Praktik Terbaik Konfigurasi Siklus Hidup

Berikut ini adalah praktik terbaik untuk menggunakan konfigurasi siklus hidup:

> **Important**
>
> Kami tidak menyarankan untuk menyimpan informasi sensitif dalam skrip konfigurasi siklus hidup Anda.

- Konfigurasi siklus hidup berjalan sebagai pengguna. `root` Jika skrip Anda membuat perubahan dalam `/home/ec2-user/SageMaker` direktori, (misalnya, menginstal paket dengan`pip`), gunakan perintah `sudo -u ec2-user` untuk menjalankan sebagai `ec2-user` pengguna. Ini adalah pengguna yang sama dengan yang SageMaker dijalankan Amazon.
- SageMaker instance notebook menggunakan `conda` lingkungan untuk mengimplementasikan kernel yang berbeda untuk notebook Jupyter. Jika Anda ingin menginstal paket yang tersedia untuk satu atau lebih kernel notebook, lampirkan perintah untuk menginstal paket dengan perintah `conda` lingkungan yang mengaktifkan lingkungan conda yang berisi kernel tempat Anda ingin menginstal paket.

  Misalnya, jika Anda ingin menginstal paket hanya untuk `python3` lingkungan, gunakan kode berikut:

```
#!/bin/bash
sudo -u ec2-user -i <<EOF

# This will affect only the Jupyter kernel called "conda_python3".
source activate python3

# Replace myPackage with the name of the package you want to install.
pip install myPackage
# You can also perform "conda install" here as well.

source deactivate

EOF
```

Jika Anda ingin menginstal paket di semua lingkungan conda dalam instance notebook, gunakan kode berikut:

```
#!/bin/bash
sudo -u ec2-user -i <<EOF

# Note that "base" is special environment name, include it there as well.
for env in base /home/ec2-user/anaconda3/envs/*; do
    source /home/ec2-user/anaconda3/bin/activate $(basename "$env")

    # Installing packages in the Jupyter system environment can affect stability of
 your SageMaker
    # Notebook Instance.  You can remove this check if you'd like to install Jupyter
 extensions, etc.
    if [ $env = 'JupyterSystemEnv' ]; then
      continue
    fi

    # Replace myPackage with the name of the package you want to install.
    pip install --upgrade --quiet myPackage
    # You can also perform "conda install" here as well.

    source /home/ec2-user/anaconda3/bin/deactivate
done

EOF
```

- Anda harus menyimpan semua lingkungan conda di folder lingkungan default (/home/user/anaconda3/envs).

> **Important**
>
> Saat Anda membuat atau mengubah skrip, sebaiknya gunakan editor teks yang menyediakan jeda baris bergaya Unix, seperti editor teks yang tersedia di konsol saat Anda membuat buku catatan. Menyalin teks dari sistem operasi non-Linux mungkin menyebabkan jeda baris yang tidak kompatibel dan mengakibatkan kesalahan yang tidak terduga.

Instal Pustaka dan Kernel Eksternal di Instans Notebook

Instans SageMaker notebook Amazon hadir dengan beberapa lingkungan yang sudah diinstal. Lingkungan ini berisi kernel Jupyter dan paket Python termasuk: scikit, Pandas,,, dan MxNet. NumPy TensorFlow Lingkungan ini, bersama dengan semua file di `sample-notebooks` folder, disegarkan saat Anda berhenti dan memulai instance notebook. Anda juga dapat menginstal lingkungan Anda sendiri yang berisi paket dan kernel pilihan Anda.

Kernel Jupyter yang berbeda dalam instance SageMaker notebook Amazon adalah lingkungan conda yang terpisah. Untuk informasi tentang lingkungan conda, lihat Mengelola lingkungan dalam dokumentasi Conda.

Instal lingkungan dan kernel khusus pada volume Amazon EBS instans notebook. Ini memastikan bahwa mereka bertahan ketika Anda menghentikan dan memulai ulang instance notebook, dan bahwa pustaka eksternal apa pun yang Anda instal tidak diperbarui oleh. SageMaker Untuk melakukannya, gunakan konfigurasi siklus hidup yang menyertakan skrip yang berjalan saat Anda membuat instance notebook (`on-create`)dan skrip yang berjalan setiap kali Anda memulai ulang instance notebook ()`on-start`. Untuk informasi selengkapnya tentang penggunaan konfigurasi siklus hidup instance notebook, lihat. Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup Ada GitHub repositori yang berisi contoh skrip konfigurasi siklus hidup di Sampel Konfigurasi Siklus Hidup SageMakerInstance Notebook.

Contoh di https://github.com/aws-samples/ amazon-sagemaker-notebook-instance - lifecycle-config-samples /blob/master/scripts/ persistent-conda-ebs /on-create.sh dan https://github.com/aws-samples/ amazon-sagemaker-notebook-instance - lifecycle-config-samples /blob/master/scripts/ persistent-conda-ebs /on-start.sh menunjukkan praktik terbaik untuk menginstal lingkungan dan kernel pada instance notebook. `on-create`Skrip menginstal `ipykernel` perpustakaan untuk membuat lingkungan kustom sebagai kernel Jupyter, kemudian menggunakan `pip install` dan `conda install` menginstal pustaka. Anda dapat menyesuaikan skrip untuk membuat lingkungan khusus dan menginstal pustaka yang Anda inginkan. SageMaker tidak memperbarui pustaka ini saat Anda menghentikan dan memulai ulang instance notebook, sehingga Anda dapat memastikan bahwa lingkungan kustom Anda memiliki versi pustaka tertentu yang Anda inginkan. `on-start`Skrip menginstal lingkungan kustom apa pun yang Anda buat sebagai kernel Jupyter, sehingga muncul di daftar dropdown di menu Jupyter New.

Alat instalasi Package

SageMaker notebook mendukung alat instalasi paket berikut:

- instal conda
- instal pip

Anda dapat menginstal paket menggunakan metode berikut:

- Skrip konfigurasi siklus hidup.

  Misalnya skrip, lihat Sampel Konfigurasi Siklus Hidup Instance SageMaker Notebook. Untuk informasi selengkapnya tentang konfigurasi siklus hidup, lihat Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup.
- Notebook — Perintah berikut didukung.
  - `%conda install`
  - `%pip install`
- Terminal Jupyter — Anda dapat menginstal paket menggunakan pip dan conda secara langsung.

Dari dalam notebook Anda dapat menggunakan sintaks perintah sistem (baris dimulai dengan!) untuk menginstal paket, misalnya, `!pip install` dan`!conda install`. Baru-baru ini, perintah baru telah ditambahkan ke IPython: `%pip` dan. `%conda` Perintah ini adalah cara yang disarankan untuk menginstal paket dari notebook karena mereka dengan benar memperhitungkan lingkungan aktif atau juru bahasa yang digunakan. Untuk informasi selengkapnya, lihat Menambahkan %pip dan %conda magic functions.

Conda

Conda adalah sistem manajemen paket open source dan sistem manajemen lingkungan, yang dapat menginstal paket dan dependensinya. SageMaker mendukung penggunaan Conda dengan salah satu dari dua saluran utama, saluran default, dan saluran conda-forge. Untuk informasi selengkapnya, lihat saluran Conda. Saluran conda-forge adalah saluran komunitas tempat kontributor dapat mengunggah paket.

> **Note**
>
> Karena bagaimana Conda menyelesaikan grafik ketergantungan, menginstal paket dari conda-forge dapat memakan waktu lebih lama secara signifikan (dalam kasus terburuk, lebih dari 10 menit).

Deep Learning AMI hadir dengan banyak lingkungan conda dan banyak paket yang sudah diinstal sebelumnya. Karena jumlah paket yang sudah diinstal sebelumnya, sulit menemukan satu set paket yang dijamin kompatibel. Anda mungkin melihat peringatan “Lingkungan tidak konsisten, silakan periksa paket paket dengan cermat”. Terlepas dari peringatan ini SageMaker , pastikan bahwa semua lingkungan SageMaker yang disediakan sudah benar. SageMaker tidak dapat menjamin bahwa setiap paket yang diinstal pengguna akan berfungsi dengan benar.

> **Note**
>
> Pengguna SageMaker, AWS Deep Learning AMI dan Amazon EMR dapat mengakses repositori Anaconda komersial tanpa mengambil lisensi komersial hingga 1 Februari 2024 saat menggunakan Anaconda dalam layanan tersebut. Untuk penggunaan apa pun di luar ketiga layanan ini, pelanggan bertanggung jawab untuk menentukan persyaratan lisensi Anaconda mereka sendiri.

Conda memiliki dua metode untuk mengaktifkan lingkungan: conda activate/deactivate, dan source activate/nonactivate. Untuk informasi lebih lanjut, lihat Haruskah saya menggunakan 'conda activate' atau 'source activate' di Linux.

SageMaker mendukung pemindahan lingkungan Conda ke volume Amazon EBS, yang dipertahankan saat instance dihentikan. Lingkungan tidak bertahan ketika lingkungan diinstal ke volume root, yang merupakan perilaku default. Untuk contoh skrip siklus hidup, lihat. persistent-conda-ebs

Operasi conda yang didukung (lihat catatan di bagian bawah topik ini)

- conda install paket dalam satu lingkungan
- conda menginstal paket di semua lingkungan
- conda install paket R di lingkungan R
- Menginstal paket dari repositori conda utama
- Menginstal paket dari conda-forge
- Mengubah lokasi pemasangan Conda untuk menggunakan EBS
- Mendukung conda activate dan source activate

Pip

Pip adalah alat de facto untuk menginstal dan mengelola paket Python. Pip mencari paket-paket pada Indeks Paket Python (PyPI) secara default. Tidak seperti Conda, pip tidak memiliki dukungan lingkungan bawaan, dan tidak selengkap Conda dalam hal paket dengan dependensi pustaka asli/sistem. Pip dapat digunakan untuk menginstal paket di lingkungan Conda.

Anda dapat menggunakan repositori paket alternatif dengan pip alih-alih PyPI. Untuk contoh skrip siklus hidup, lihat on-start.sh.

Operasi pip yang didukung (lihat catatan di bagian bawah topik ini)

- Menggunakan pip untuk menginstal paket tanpa lingkungan conda aktif (instal sistem paket di seluruh)
- Menggunakan pip untuk menginstal paket di lingkungan conda
- Menggunakan pip untuk menginstal paket di semua lingkungan conda
- Mengubah lokasi pemasangan pip untuk menggunakan EBS
- Menggunakan repositori alternatif untuk menginstal paket dengan pip

Tidak didukung

SageMaker bertujuan untuk mendukung sebanyak mungkin operasi instalasi paket. Namun, jika paket diinstal oleh SageMaker atau DLAMI, dan Anda menggunakan operasi berikut pada paket-paket ini, itu mungkin membuat instance notebook Anda tidak stabil:

- Menghapus instalasi
- Menurunkan
- Upgrading

Kami tidak memberikan dukungan untuk menginstal paket melalui yum install atau menginstal paket R dari CRAN.

Karena potensi masalah dengan kondisi jaringan atau konfigurasi, atau ketersediaan Conda atau PyPi, kami tidak dapat menjamin bahwa paket akan diinstal dalam jumlah waktu yang tetap atau deterministik.

> **Note**
>
> Kami tidak dapat menjamin bahwa instalasi paket akan berhasil. Mencoba menginstal paket di lingkungan dengan dependensi yang tidak kompatibel dapat mengakibatkan kegagalan. Dalam kasus seperti itu Anda harus menghubungi pengelola perpustakaan untuk melihat apakah mungkin untuk memperbarui dependensi paket. Atau Anda dapat mencoba untuk memodifikasi lingkungan sedemikian rupa untuk memungkinkan instalasi. Namun modifikasi ini kemungkinan berarti menghapus atau memperbarui paket yang ada, yang berarti kami tidak dapat lagi menjamin stabilitas lingkungan ini.

Pembaruan Perangkat Lunak Instans Notebook

Amazon SageMaker secara berkala menguji dan merilis perangkat lunak yang diinstal pada instance notebook. Hal ini mencakup:

- Pembaruan kernel
- Patch keamanan
- AWSPembaruan SDK
- Pembaruan Amazon SageMaker Python SDK
- Pembaruan perangkat lunak sumber terbuka

Untuk memastikan bahwa Anda memiliki pembaruan perangkat lunak terbaru, hentikan dan mulai ulang instance notebook Anda, baik di SageMaker konsol atau dengan menelepon `StopNotebookInstance`.

Anda juga dapat memperbarui perangkat lunak yang diinstal secara manual pada instance notebook Anda saat sedang berjalan dengan menggunakan perintah pembaruan di terminal atau di notebook.

> **Note**
>
> Memperbarui kernel dan beberapa paket mungkin bergantung pada apakah akses root diaktifkan untuk instance notebook. Untuk informasi selengkapnya, lihat Kontrol akses root ke instance SageMaker notebook.

Anda dapat memeriksa Personal Health Dashboard atau buletin keamanan di Buletin Keamanan untuk pembaruan.

Mengontrol Instans Spark EMR Amazon Menggunakan Notebook

Anda dapat menggunakan instance notebook yang dibuat dengan skrip konfigurasi siklus hidup kustom untuk mengakses AWS layanan dari buku catatan Anda. Misalnya, Anda dapat membuat skrip yang memungkinkan Anda menggunakan buku catatan dengan Sparkmagic untuk mengontrol AWS sumber daya lain, seperti instans EMR Amazon. Anda kemudian dapat menggunakan instans EMR Amazon untuk memproses data Anda alih-alih menjalankan analisis data pada notebook Anda. Ini memungkinkan Anda membuat instance notebook yang lebih kecil karena Anda tidak akan menggunakan instance untuk memproses data. Ini sangat membantu ketika Anda memiliki kumpulan data besar yang memerlukan instance notebook besar untuk memproses data.

Proses ini membutuhkan tiga prosedur menggunakan SageMaker konsol Amazon:

- Buat instans Amazon EMR Spark
- Buat Notebook Jupyter
- Uji koneksi EMR Notebook-to-Amazon

Untuk membuat instans Amazon EMR Spark yang dapat dikontrol dari notebook menggunakan Sparkmagic

1. Buka konsol Amazon EMR di https://console.aws.amazon.com/elasticmapreduce/.
2. Di panel navigasi, pilih Buat cluster.
3. Pada halaman Buat Cluster - Opsi Cepat, di bawah konfigurasi Perangkat Lunak, pilih Spark: Spark 2.4.4 di Hadoop 2.8.5 YARN dengan Ganglia 3.7.2 dan Zeppelin 0.8.2.
4. Tetapkan parameter tambahan pada halaman dan kemudian pilih Buat cluster.
5. Pada halaman Cluster, pilih nama cluster yang Anda buat. Perhatikan DNS Publik Master, grup keamanan master EMR, dan nama VPC serta subnet ID tempat cluster EMR dibuat. Anda akan menggunakan nilai-nilai ini saat membuat buku catatan.

Untuk membuat notebook yang menggunakan Sparkmagic untuk mengontrol instans Amazon EMR Spark

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, di bawah instance Notebook, pilih Buat buku catatan.

3. Masukkan nama instance notebook dan pilih jenis instans.
4. Pilih Konfigurasi tambahan, lalu, di bawah Konfigurasi Siklus Hidup, pilih Buat konfigurasi siklus hidup baru.
5. Tambahkan kode berikut ke skrip konfigurasi siklus hidup:

```
# OVERVIEW
# This script connects an Amazon EMR cluster to an Amazon SageMaker notebook
 instance that uses Sparkmagic.
#
# Note that this script will fail if the Amazon EMR cluster's master node IP
 address is not reachable.
#   1. Ensure that the EMR master node IP is resolvable from the notebook instance.
#      One way to accomplish this is to have the notebook instance and the Amazon
 EMR cluster in the same subnet.
#   2. Ensure the EMR master node security group provides inbound access from the
 notebook instance security group.
#       Type        - Protocol - Port - Source
#       Custom TCP  - TCP      - 8998 - $NOTEBOOK_SECURITY_GROUP
#   3. Ensure the notebook instance has internet connectivity to fetch the
 SparkMagic example config.
#
# https://aws.amazon.com/blogs/machine-learning/build-amazon-sagemaker-notebooks-
backed-by-spark-in-amazon-emr/

# PARAMETERS
EMR_MASTER_IP=your.emr.master.ip


cd /home/ec2-user/.sparkmagic

echo "Fetching Sparkmagic example config from GitHub..."
wget https://raw.githubusercontent.com/jupyter-incubator/sparkmagic/master/
sparkmagic/example_config.json

echo "Replacing EMR master node IP in Sparkmagic config..."
sed -i -- "s/localhost/$EMR_MASTER_IP/g" example_config.json
mv example_config.json config.json

echo "Sending a sample request to Livy.."
curl "$EMR_MASTER_IP:8998/sessions"
```

6. Di PARAMETERS bagian skrip, ganti `your.emr.master.ip` dengan nama Master Public DNS untuk instans EMR Amazon.
7. Pilih Buat konfigurasi.
8. Pada halaman Buat buku catatan, pilih Jaringan - opsional.
9. Pilih VPC dan subnet tempat instans EMR Amazon berada.
10. Pilih grup keamanan yang digunakan oleh simpul master EMR Amazon.
11. Pilih Buat instans notebook.

Saat instance notebook sedang dibuat, statusnya Tertunda. Setelah instance dibuat dan skrip konfigurasi siklus hidup berhasil dijalankan, statusnya adalah. InService

> **Note**
>
> Jika instance notebook tidak dapat terhubung ke instans EMR Amazon, tidak SageMaker dapat membuat instance notebook. Sambungan dapat gagal jika instans EMR Amazon dan notebook tidak berada dalam VPC dan subnet yang sama, jika grup keamanan master EMR Amazon tidak digunakan oleh notebook, atau jika nama Master Public DNS dalam skrip salah.

Untuk menguji koneksi antara instans EMR Amazon dan notebook

1. Saat status notebook InService, pilih Buka Jupyter untuk membuka buku catatan.
2. Pilih Baru, lalu pilih Sparkmagic () PySpark.
3. Di sel kode, masukkan **`%%info`** lalu jalankan sel.

   Outputnya harus mirip dengan yang berikut

```
Current session configs: {'driverMemory': '1000M', 'executorCores': 2, 'kind':
 'pyspark'}
                    No active sessions.
```

Contoh Notebook

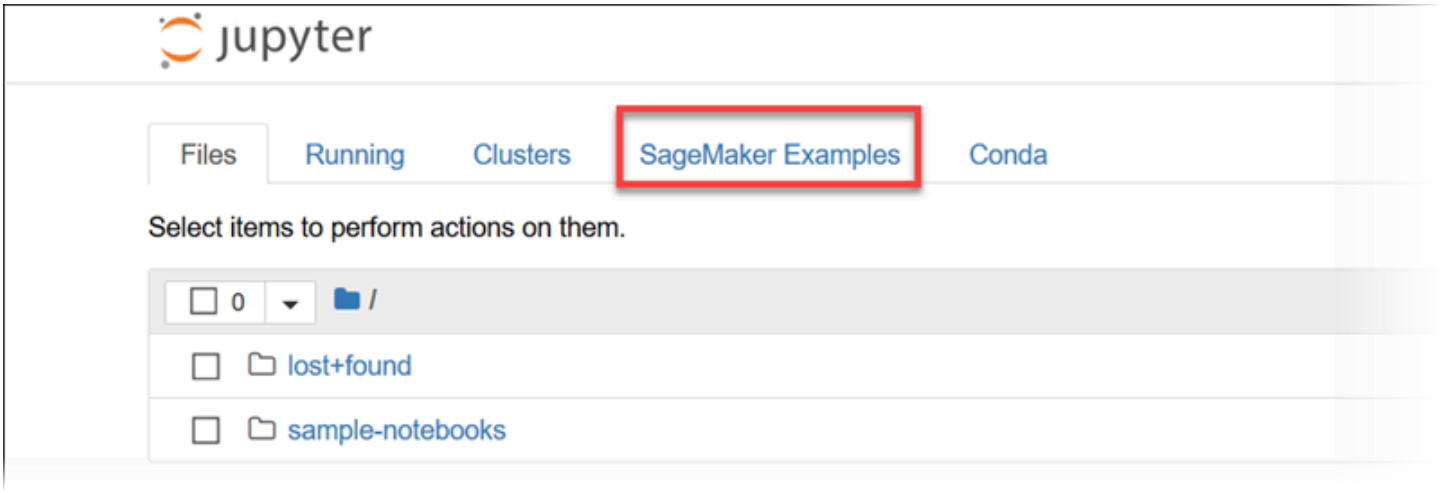
Instance notebook Anda berisi contoh notebook yang disediakan oleh Amazon SageMaker. Contoh notebook berisi kode yang menunjukkan cara menerapkan solusi pembelajaran mesin dengan

menggunakan SageMaker. Instans notebook menggunakan ekstensi `nbexamples` Jupyter, yang memungkinkan Anda untuk melihat versi read-only dari contoh notebook atau membuat salinannya yang dapat Anda ubah dan jalankan. Untuk informasi lebih lanjut tentang `nbexamples` ekstensi, lihat https://github.com/danielballan/nbexamples. Untuk informasi tentang contoh buku catatan untuk SageMaker Studio, lihatGunakan Notebook Amazon SageMaker Studio Classic.

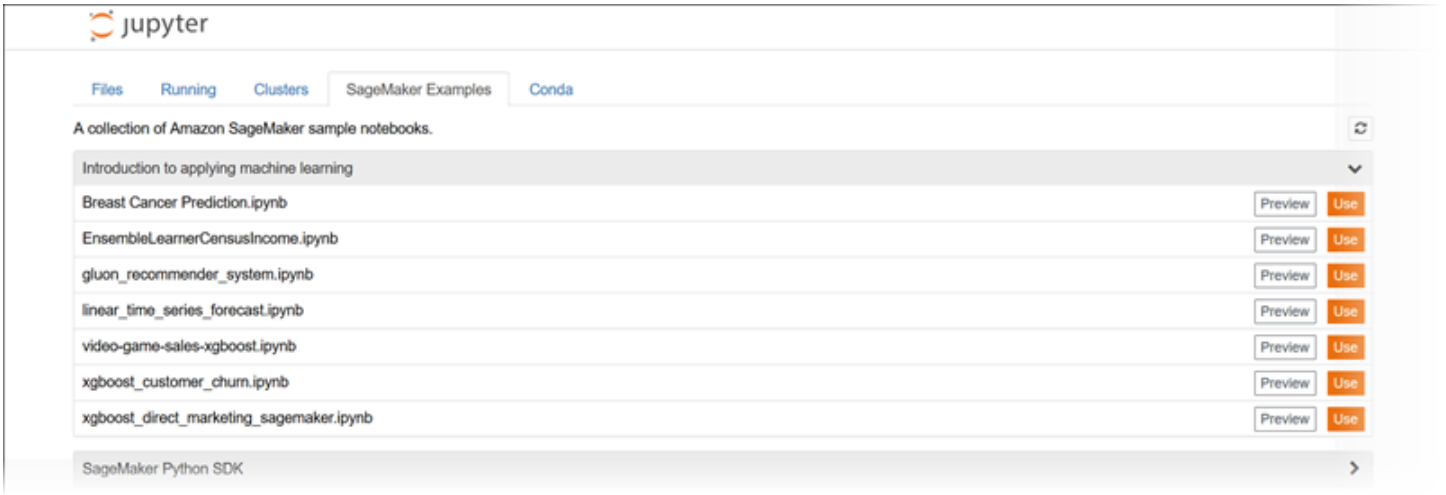
> **Note**
>
> Contoh notebook biasanya mengunduh dataset dari internet. Jika Anda menonaktifkan akses internet yang SageMaker disediakan saat membuat instance buku catatan, contoh buku catatan mungkin tidak berfungsi. Untuk informasi selengkapnya, lihat Hubungkan Instance Notebook di VPC ke Sumber Daya Eksternal.

Gunakan atau Lihat Contoh Notebook di Jupyter Classic

Untuk melihat atau menggunakan contoh buku catatan dalam tampilan Jupyter klasik, pilih tab Contoh. SageMaker

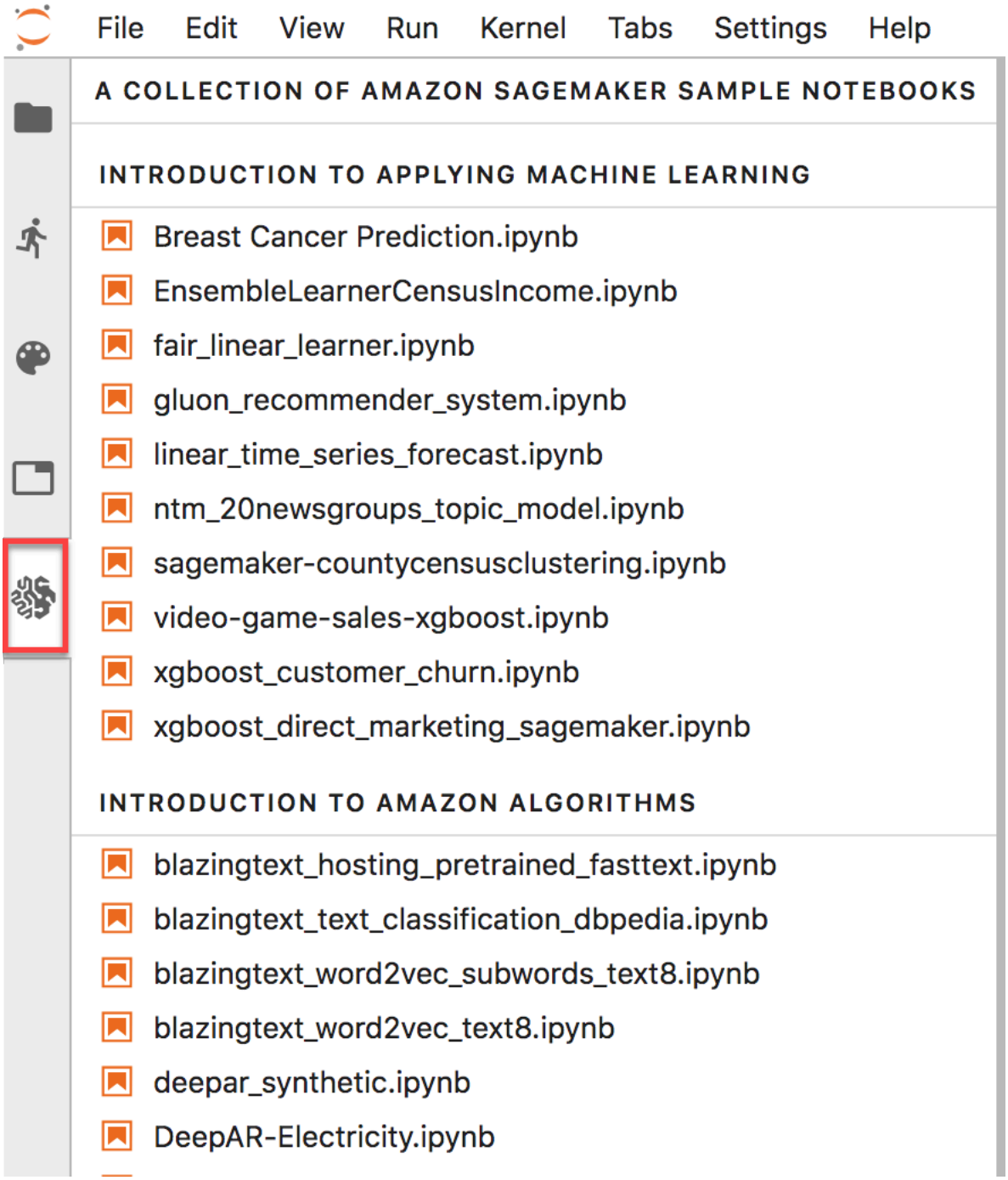


Untuk melihat versi read-only dari contoh notebook dalam tampilan klasik Jupyter, pada tab SageMaker Contoh, pilih Pratinjau untuk buku catatan itu. Untuk membuat salinan buku catatan contoh di direktori home instance notebook Anda, pilih Gunakan. Di kotak dialog, Anda dapat mengubah nama buku catatan sebelum menyimpannya.



Gunakan atau Lihat Contoh Notebook di Jupyterlab

Untuk melihat atau menggunakan contoh buku catatan dalam tampilan Jupyterlab, pilih ikon contoh di panel navigasi kiri.



Untuk melihat versi read-only dari contoh notebook, pilih nama notebook. Ini membuka notebook sebagai tab di area utama. Untuk membuat salinan buku catatan contoh di direktori home instance buku catatan Anda, pilih Buat Salinan di spanduk atas. Di kotak dialog, ketik nama untuk buku catatan dan kemudian pilih BUAT SALINAN.

Untuk informasi selengkapnya tentang contoh buku catatan, lihat SageMaker contoh GitHub repositori.

Mengatur Kernel Notebook

Amazon SageMaker menyediakan beberapa kernel untuk Jupyter yang menyediakan dukungan untuk Python 2 dan 3, Apache MXNet, dan. TensorFlow PySpark Untuk mengatur kernel untuk notebook baru di dasbor notebook Jupyter, pilih Baru, lalu pilih kernel dari daftar. Untuk informasi selengkapnya tentang kernel yang tersedia, lihatKernel yang tersedia.

Anda juga dapat membuat kernel kustom yang dapat Anda gunakan dalam instance notebook Anda. Untuk informasi, lihat Instal Pustaka dan Kernel Eksternal di Instans Notebook.

Kaitkan Repositori Git dengan SageMaker Instans Notebook

Kaitkan repositori Git dengan instance notebook Anda untuk menyimpan buku catatan Anda di lingkungan kontrol sumber yang tetap ada meskipun Anda menghentikan atau menghapus instance notebook Anda. Anda dapat mengaitkan satu repositori default dan hingga tiga repositori tambahan dengan instance notebook. Repositori dapat di-host diAWS CodeCommit, GitHub, atau di server Git lainnya. Mengaitkan repositori Git dengan instance notebook Anda dapat berguna untuk:

- Kegigihan - Notebook dalam instance notebook disimpan pada volume Amazon EBS yang tahan lama, tetapi tidak bertahan melampaui masa pakai instance notebook Anda. Menyimpan buku catatan di repositori Git memungkinkan Anda menyimpan dan menggunakan buku catatan meskipun Anda menghentikan atau menghapus instance buku catatan Anda.
- Kolaborasi - Rekan dalam tim sering mengerjakan proyek pembelajaran mesin bersama. Menyimpan notebook Anda di repositori Git memungkinkan peer yang bekerja di instans notebook yang berbeda untuk berbagi buku catatan dan berkolaborasi dengannya dalam lingkungan kontrol sumber.
- Pembelajaran - Banyak buku catatan Jupyter yang mendemonstrasikan teknik pembelajaran mesin tersedia di repositori Git yang dihosting publik, seperti on. GitHub Anda dapat mengaitkan instance notebook Anda dengan repositori untuk memuat notebook Jupyter dengan mudah yang terdapat dalam repositori tersebut.

Ada dua cara untuk mengaitkan repositori Git dengan instance notebook:

- Tambahkan repositori Git sebagai sumber daya di akun Amazon SageMaker Anda. Kemudian, untuk mengakses repositori, Anda dapat menentukan AWS rahasia Secrets Manager yang berisi kredensyal. Dengan begitu, Anda dapat mengakses repositori yang memerlukan otentikasi.
- Kaitkan repositori Git publik yang bukan sumber daya di akun Anda. Jika Anda melakukan ini, Anda tidak dapat menentukan kredensyal untuk mengakses repositori.

Topik

Tambahkan Repositori Git ke Akun Amazon Anda SageMaker

Untuk mengelola GitHub repositori Anda, kaitkan dengan mudah dengan instans notebook Anda, dan kaitkan kredensyal untuk repositori yang memerlukan otentikasi, tambahkan repositori sebagai sumber daya di akun Amazon Anda. SageMaker Anda dapat melihat daftar repositori yang disimpan di akun Anda dan detail tentang setiap repositori di SageMaker konsol dan dengan menggunakan API.

Anda dapat menambahkan repositori Git ke SageMaker akun Anda di SageMaker konsol atau dengan menggunakan. AWS CLI

> **Note**
>
> Anda dapat menggunakan SageMaker API `CreateCodeRepository`untuk menambahkan repositori Git ke SageMaker akun Anda, tetapi step-by-step instruksi tidak disediakan di sini.

Tambahkan Repositori Git ke SageMaker Akun Anda (Konsol)

Untuk menambahkan repositori Git sebagai sumber daya di akun Anda SageMaker

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.

2. Di bawah Notebook, pilih repositori Git, lalu pilih Tambah repositori.
3. Untuk menambahkan CodeCommit repositori, pilih. AWS CodeCommit Untuk menambahkan GitHub atau repositori berbasis Git lainnya, pilih GitHub /Other Git-based repo.

Untuk menambahkan CodeCommit repositori yang ada

1. Pilih Gunakan repositori yang ada.
2. Untuk Repositori, pilih repositori dari daftar.
3. Masukkan nama yang akan digunakan untuk repositori di. SageMaker Nama harus 1 hingga 63 karakter. Karakter yang valid adalah a-z, A-Z, 0-9, dan - (tanda hubung).
4. Pilih Menambahkan repositori.

Untuk membuat CodeCommit repositori baru

1. Pilih Buat repositori baru.
2. Masukkan nama untuk repositori yang dapat Anda gunakan di keduanya dan CodeCommit . SageMaker Nama harus 1 hingga 63 karakter. Karakter yang valid adalah a-z, A-Z, 0-9, dan - (tanda hubung).
3. Pilih Buat repositori.

Untuk menambahkan repositori Git yang dihosting di tempat lain selain CodeCommit

1. Pilih GitHub/Repo berbasis Git lainnya.
2. Masukkan nama hingga 63 karakter. Karakter yang valid termasuk karakter alfa-numerik, tanda hubung (-), dan 0-9.
3. Masukkan URL untuk repositori. Jangan berikan nama pengguna di URL. Tambahkan kredenal masuk AWS Secrets Manager seperti yang dijelaskan pada langkah berikutnya.
4. Untuk kredensyal Git, pilih kredensyal yang akan digunakan untuk mengautentikasi ke repositori. Ini diperlukan hanya jika repositori Git bersifat pribadi.

> **Note**
>
> Jika autentikasi dua faktor diaktifkan untuk repositori Git Anda, masukkan token akses pribadi yang dihasilkan oleh penyedia layanan Git Anda di bidang. `password`

a. Untuk menggunakan AWS rahasia Secrets Manager yang ada, pilih Gunakan rahasia yang ada, lalu pilih rahasia dari daftar. Untuk informasi tentang membuat dan menyimpan rahasia, lihat Membuat Rahasia Dasar di Panduan Pengguna AWS Secrets Manager. Nama rahasia yang Anda gunakan harus berisi string`sagemaker`.

> **Note**
>
> Rahasia harus memiliki label penahapan dari `AWSCURRENT` dan harus dalam format berikut:
> `{"username": UserName, "password": Password}`
> Untuk GitHub repositori, kami sarankan menggunakan token akses pribadi di lapangan. `password` Untuk informasi, lihat https://help.github.com/articles/ creating-a-personal-access - token-for-the-command -line/.

b. Untuk membuat AWS rahasia Secrets Manager baru, pilih Buat rahasia, masukkan nama untuk rahasia, lalu masukkan kredenal masuk yang akan digunakan untuk mengautentikasi ke repositori. Nama rahasia harus berisi string`sagemaker`.

> **Note**
>
> Peran IAM yang Anda gunakan untuk membuat rahasia harus memiliki `secretsmanager:GetSecretValue` izin dalam kebijakan IAM-nya.
> Rahasia harus memiliki label penahapan dari `AWSCURRENT` dan harus dalam format berikut:
> `{"username": UserName, "password": Password}`
> Untuk GitHub repositori, kami sarankan menggunakan token akses pribadi.

c. Untuk tidak menggunakan kredensi apa pun, pilih Tidak ada rahasia.

5. Pilih Buat rahasia.

Tambahkan Repositori Git ke SageMaker Akun Amazon Anda (CLI)

Gunakan perintah `create-code-repository` AWS CLI. Tentukan nama untuk repositori sebagai nilai argumen. `code-repository-name` Nama harus 1 hingga 63 karakter. Karakter yang valid adalah a-z, A-Z, 0-9, dan - (tanda hubung). Juga tentukan yang berikut ini:

- Cabang default
- URL dari repositori Git

> **Note**
>
> Jangan berikan nama pengguna di URL. Tambahkan kredenal masuk AWS Secrets Manager seperti yang dijelaskan pada langkah berikutnya.

- Nama Sumber Daya Amazon (ARN) dari AWS rahasia Secrets Manager yang berisi kredensyal yang akan digunakan untuk mengautentikasi repositori sebagai nilai argumen `git-config`

Untuk informasi tentang membuat dan menyimpan rahasia, lihat Membuat Rahasia Dasar di Panduan Pengguna AWS Secrets Manager. Perintah berikut membuat repositori baru bernama `MyRespository` di SageMaker akun Amazon Anda yang menunjuk ke repositori Git yang dihosting di. `https://github.com/myprofile/my-repo"`

Untuk Linux, OS X, atau Unix:

```
aws sagemaker create-code-repository \
                    --code-repository-name "MyRepository" \
                    --git-config Branch=branch,RepositoryUrl=https://github.com/
myprofile/my-repo,SecretArn=arn:aws:secretsmanager:us-east-2:012345678901:secret:my-
secret-ABc0DE
```

Untuk Windows:

```
aws sagemaker create-code-repository ^
                    --code-repository-name "MyRepository" ^
                    --git-config "{\"Branch\":\"master\", \"RepositoryUrl\" :
                    \"https://github.com/myprofile/my-repo\", \"SecretArn\" :
 \"arn:aws:secretsmanager:us-east-2:012345678901:secret:my-secret-ABc0DE\"}"
```

> **Note**
>
> Rahasia harus memiliki label penahapan dari `AWSCURRENT` dan harus dalam format berikut: `{"username": UserName, "password": Password}`
> Untuk GitHub repositori, kami sarankan menggunakan token akses pribadi.

Membuat Instance Notebook dengan Repositori Git Terkait

Anda dapat mengaitkan repositori Git dengan instance notebook saat Anda membuat instance notebook dengan menggunakanAWS Management Console, atau. AWS CLI Jika Anda ingin menggunakan CodeCommit repositori yang berada di AWS akun yang berbeda dari instance notebook, siapkan akses lintas akun untuk repositori. Untuk informasi, lihat Kaitkan CodeCommit Repositori di AWS Akun Berbeda dengan Instance Notebook.

Topik

- Membuat Instance Notebook dengan Repositori Git Terkait (Konsol)
- Membuat Instance Notebook dengan Repositori Git Terkait (CLI)

Membuat Instance Notebook dengan Repositori Git Terkait (Konsol)

Untuk membuat instance notebook dan mengaitkan repositori Git di konsol Amazon SageMaker

1. Ikuti petunjuk di Langkah 1: Buat Instans SageMaker Notebook Amazon.
2. Untuk repositori Git, pilih repositori Git untuk diasosiasikan dengan instance notebook.

    a. Untuk repositori Default, pilih repositori yang ingin Anda gunakan sebagai repositori default Anda. SageMakermengkloning repositori ini sebagai subdirektori di direktori startup Jupyter di. `/home/ec2-user/SageMaker` Ketika Anda membuka instance notebook Anda, itu terbuka di repositori ini. Untuk memilih repositori yang disimpan sebagai sumber daya di akun Anda, pilih namanya dari daftar. Untuk menambahkan repositori baru sebagai sumber daya di akun Anda, pilih Tambahkan repositori ke SageMaker (buka alur Tambah repositori di jendela baru) dan kemudian ikuti instruksi di. Membuat Instance Notebook dengan Repositori Git Terkait (Konsol) Untuk mengkloning repositori publik yang tidak disimpan di akun Anda, pilih Kloning repositori Git publik ke instance buku catatan ini saja, lalu tentukan URL untuk repositori tersebut.

    b. Untuk Repositori tambahan 1, pilih repositori yang ingin Anda tambahkan sebagai direktori tambahan. SageMakermengkloning repositori ini sebagai subdirektori di direktori startup Jupyter di. `/home/ec2-user/SageMaker` Untuk memilih repositori yang disimpan sebagai sumber daya di akun Anda, pilih namanya dari daftar. Untuk menambahkan repositori baru sebagai sumber daya di akun Anda, pilih Tambahkan repositori ke SageMaker (buka alur Tambah repositori di jendela baru) dan kemudian ikuti instruksi di. Membuat Instance Notebook dengan Repositori Git Terkait (Konsol) Untuk mengkloning repositori yang tidak

disimpan di akun Anda, pilih Kloning repositori Git publik ke instance notebook ini saja, lalu tentukan URL untuk repositori tersebut.

Ulangi langkah ini hingga tiga kali untuk menambahkan hingga tiga repositori tambahan ke instance notebook Anda.

Membuat Instance Notebook dengan Repositori Git Terkait (CLI)

Untuk membuat instance notebook dan mengaitkan repositori Git dengan menggunakanAWS CLI, gunakan `create-notebook-instance` perintah sebagai berikut:

- Tentukan repositori yang ingin Anda gunakan sebagai repositori default Anda sebagai nilai argumen. `default-code-repository` Amazon SageMaker mengkloning repositori ini sebagai subdirektori di direktori startup Jupyter di. `/home/ec2-user/SageMaker` Ketika Anda membuka instance notebook Anda, itu terbuka di repositori ini. Untuk menggunakan repositori yang disimpan sebagai sumber daya di SageMaker akun Anda, tentukan nama repositori sebagai nilai argumen. `default-code-repository` Untuk menggunakan repositori yang tidak disimpan di akun Anda, tentukan URL repositori sebagai nilai argumen. `default-code-repository`
- Tentukan hingga tiga repositori tambahan sebagai nilai argumen. `additional-code-repositories` SageMaker mengkloning repositori ini sebagai subdirektori di direktori startup Jupyter di`/home/ec2-user/SageMaker`, dan repositori dikecualikan dari repositori default dengan menambahkannya ke direktori repositori default. `.git/info/exclude` Untuk menggunakan repositori yang disimpan sebagai sumber daya di SageMaker akun Anda, tentukan nama repositori sebagai nilai argumen. `additional-code-repositories` Untuk menggunakan repositori yang tidak disimpan di akun Anda, tentukan URL repositori sebagai nilai argumen. `additional-code-repositories`

Misalnya, perintah berikut membuat instance notebook yang memiliki repositori bernama`MyGitRepo`, yang disimpan sebagai sumber daya di SageMaker akun Anda, sebagai repositori default, dan repositori tambahan yang di-host di: GitHub

```
aws sagemaker create-notebook-instance \
                    --notebook-instance-name "MyNotebookInstance" \
                    --instance-type "ml.t2.medium" \
                    --role-arn "arn:aws:iam::012345678901:role/service-role/
AmazonSageMaker-ExecutionRole-20181129T121390" \
                    --default-code-repository "MyGitRepo" \
```

```
                    --additional-code-repositories "https://github.com/myprofile/my-
other-repo"
```

> **Note**
>
> Jika Anda menggunakan AWS CodeCommit repositori yang tidak berisi "SageMaker" dalam namanya, tambahkan `codecommit:GitPull` dan `codecommit:GitPush` izin ke peran yang Anda berikan sebagai `role-arn` argumen ke perintah. `create-notebook-instance` Untuk informasi tentang cara menambahkan izin ke peran, lihat Menambahkan dan Menghapus Kebijakan IAM di AWS Identity and Access ManagementPanduan Pengguna.

Kaitkan CodeCommit Repositori di AWS Akun Berbeda dengan Instance Notebook

Untuk mengaitkan CodeCommit repositori di AWS akun yang berbeda dengan instance notebook Anda, siapkan akses lintas akun untuk repositori. CodeCommit

Untuk mengatur akses lintas akun untuk CodeCommit repositori dan mengaitkannya dengan instance notebook:

1. Di AWS akun yang berisi CodeCommit repositori, buat kebijakan IAM yang memungkinkan akses ke repositori dari pengguna di akun yang berisi instance notebook Anda. Untuk selengkapnya, lihat Langkah 1: Membuat Kebijakan untuk Akses Repositori di Accounta di Panduan Pengguna. CodeCommit
2. Di AWS akun yang berisi CodeCommit repositori, buat peran IAM, dan lampirkan kebijakan yang Anda buat di langkah sebelumnya ke peran tersebut. Untuk selengkapnya, lihat Langkah 2: Membuat Peran untuk Akses Repositori di Accounta di Panduan Pengguna. CodeCommit
3. Buat profil di instance buku catatan yang menggunakan peran yang Anda buat di langkah sebelumnya:

    a. Buka instance notebook.

    b. Buka terminal di instance notebook.

    c. Edit profil baru dengan mengetikkan yang berikut di terminal:

    ```
    vi /home/ec2-user/.aws/config
    ```

    d. Edit file dengan informasi profil berikut:

```
[profile CrossAccountAccessProfile]
region = us-west-2
role_arn =
 arn:aws:iam::CodeCommitAccount:role/CrossAccountRepositoryContributorRole
credential_source=Ec2InstanceMetadata
output = json
```

Di *CodeCommitAccount* mana akun yang berisi CodeCommit repositori, *CrossAccountAccessProfile* adalah nama profil baru, dan *CrossAccountRepositoryContributorRole* merupakan nama peran yang Anda buat pada langkah sebelumnya.

4. Pada instance notebook, konfigurasikan git untuk menggunakan profil yang Anda buat pada langkah sebelumnya:

   a. Buka instance notebook.

   b. Buka terminal di instance notebook.

   c. Edit file konfigurasi Git dengan mengetik berikut ini di terminal:

   ```
   vi /home/ec2-user/.gitconfig
   ```

   d. Edit file dengan informasi profil berikut:

   ```
   [credential]
           helper = !aws codecommit credential-helper --
   profile CrossAccountAccessProfile $@
           UseHttpPath = true
   ```

   Di *CrossAccountAccessProfile* mana nama profil yang Anda buat pada langkah sebelumnya.

Menggunakan Repositori Git dalam Instance Notebook

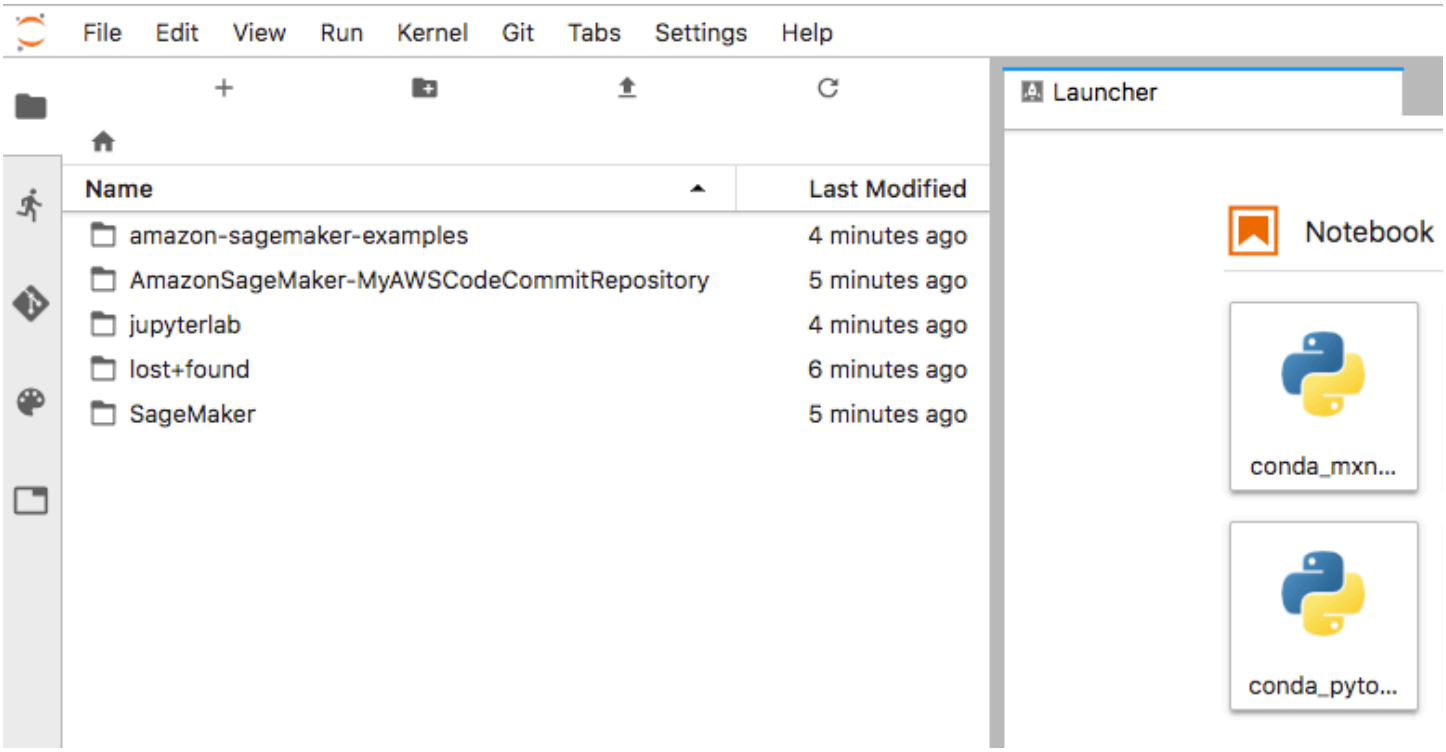
Saat Anda membuka instance notebook yang memiliki repositori Git yang terkait dengannya, instans tersebut akan terbuka di repositori default, yang dipasang di instance notebook Anda langsung di bawah. `/home/ec2-user/SageMaker` Anda dapat membuka dan membuat buku catatan, dan Anda dapat menjalankan perintah Git secara manual di sel notebook. Sebagai contoh:

```
!git pull origin master
```

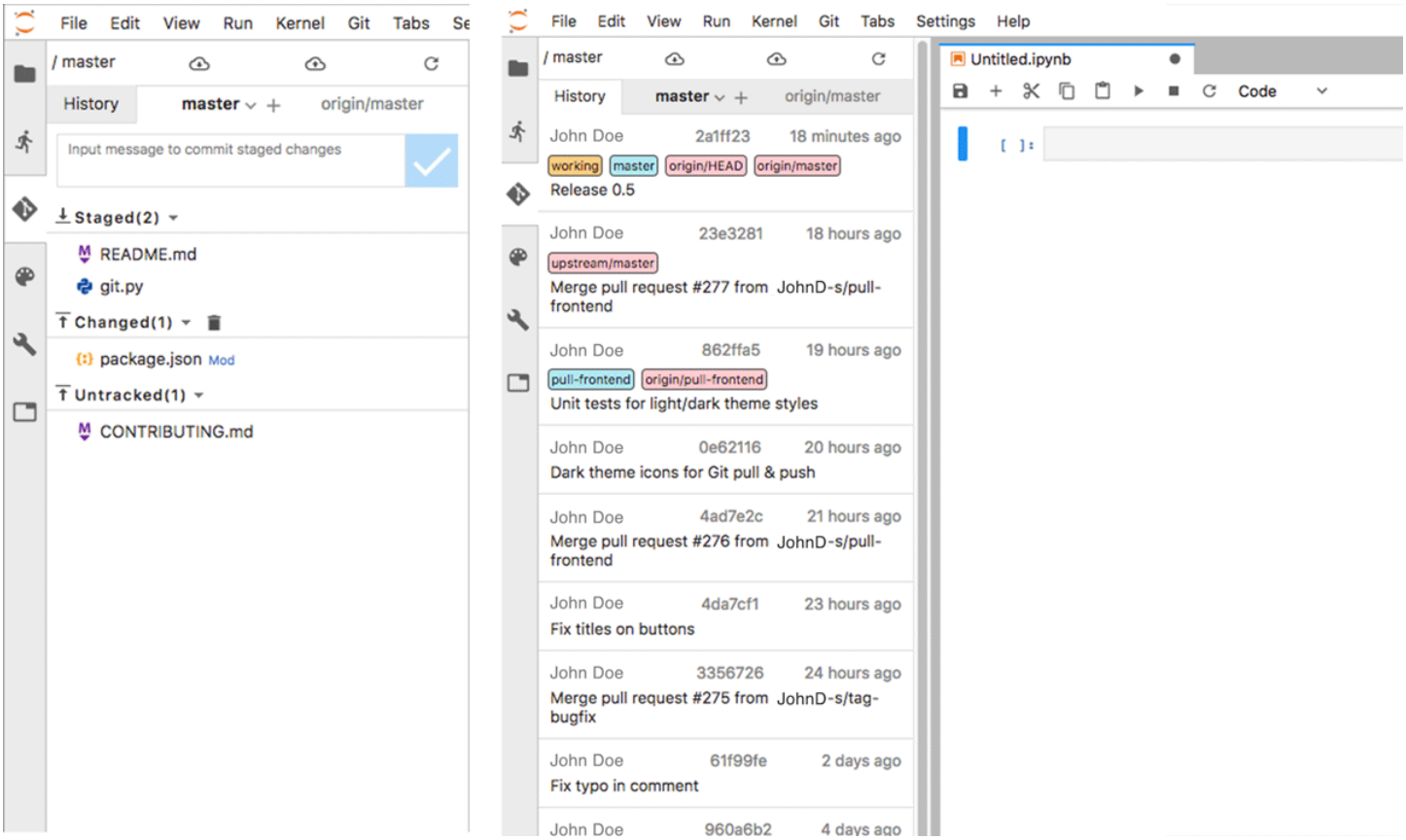
Untuk membuka salah satu repositori tambahan, navigasikan satu folder. Repositori tambahan juga diinstal sebagai direktori di bawah. `/home/ec2-user/SageMaker`

Jika Anda membuka instance notebook dengan JupyterLab antarmuka, ekstensi jupyter-git diinstal dan tersedia untuk digunakan. Untuk informasi tentang ekstensi jupyter-git untuk JupyterLab, lihat https://github.com/jupyterlab/jupyterlab-git.

Saat Anda membuka instance notebook di JupyterLab, Anda melihat repositori git yang terkait dengannya di menu sebelah kiri:



Anda dapat menggunakan ekstensi jupyter-git untuk mengelola git secara visual, alih-alih menggunakan baris perintah:



Metadata Instans Notebook

Saat Anda membuat instance notebook, Amazon SageMaker membuat file JSON pada instance di lokasi `/opt/ml/metadata/resource-metadata.json` yang berisi `ResourceName` dan `ResourceArn` instance notebook. Anda dapat mengakses metadata ini dari mana saja dalam instance notebook, termasuk dalam konfigurasi siklus hidup. Untuk informasi tentang konfigurasi siklus hidup instance notebook, lihat. [Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup]

> **Note**
>
> `resource-metadata.json`File dapat dimodifikasi dengan akses root.

`resource-metadata.json`File ini memiliki struktur sebagai berikut:

```
{
    "ResourceArn": "NotebookInstanceArn",
```

```
    "ResourceName": "NotebookInstanceName"
}
```

Anda dapat menggunakan metadata ini dari dalam instance notebook untuk mendapatkan informasi lain tentang instance notebook. Misalnya, perintah berikut mendapatkan tag yang terkait dengan instance notebook:

```
NOTEBOOK_ARN=$(jq '.ResourceArn'
            /opt/ml/metadata/resource-metadata.json --raw-output)
aws sagemaker list-tags --resource-arn $NOTEBOOK_ARN
```

Outputnya terlihat seperti berikut:

```
{
    "Tags": [
        {
            "Key": "test",
            "Value": "true"
        }
    ]
}
```

Pantau Log Jupyter di Log Amazon CloudWatch

Log Jupyter menyertakan informasi penting seperti peristiwa, metrik, dan informasi kesehatan yang memberikan wawasan yang dapat ditindaklanjuti saat menjalankan notebook Amazon. SageMaker Dengan mengimpor log Jupyter ke CloudWatch Log, pelanggan dapat menggunakan CloudWatch Log untuk mendeteksi perilaku anomali, mengatur alarm, dan menemukan wawasan agar notebook berjalan lebih lancar. SageMaker Anda dapat mengakses log meskipun instans Amazon EC2 yang meng-host notebook tidak responsif, dan menggunakan log untuk memecahkan masalah notebook yang tidak responsif. Informasi sensitif seperti ID AWS akun, kunci rahasia, dan token otentikasi di URL yang telah ditetapkan sebelumnya dihapus sehingga pelanggan dapat berbagi log tanpa membocorkan informasi pribadi.

Untuk melihat log Jupyter untuk instance notebook:

1. Masuk ke AWS Management Console dan buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Pilih instans Notebook.

3. Dalam daftar instance notebook, pilih instance notebook yang ingin Anda lihat log Jupyter dengan memilih Nama instance Notebook.

   Ini akan membawa Anda ke halaman detail untuk contoh notebook itu.
4. Di bawah Monitor pada halaman detail instance notebook, pilih Lihat log.
5. Di CloudWatch konsol, pilih aliran log untuk instance notebook Anda. Namanya dalam bentuk`NotebookInstanceName`/`jupyter.log`.

Untuk informasi selengkapnya tentang pemantauan CloudWatch log SageMaker, lihatLog SageMaker Acara Amazon dengan Amazon CloudWatch.

Lab SageMaker Studio Amazon

Amazon SageMaker Studio Lab adalah layanan gratis yang memberi pelanggan akses ke sumber daya AWS komputasi, dalam lingkungan berbasis sumber terbuka JupyterLab. Ini didasarkan pada arsitektur dan antarmuka pengguna yang sama dengan Amazon SageMaker Studio Classic, tetapi dengan subset kemampuan Studio Classic.

Dengan Studio Lab, Anda dapat menggunakan sumber daya AWS komputasi untuk membuat dan menjalankan buku catatan Jupyter tanpa mendaftar akun. AWS Karena Studio Lab didasarkan pada sumber terbuka JupyterLab, Anda dapat memanfaatkan ekstensi Jupyter sumber terbuka untuk menjalankan notebook Jupyter Anda.

Studio Lab dibandingkan dengan Amazon SageMaker Studio Classic

Meskipun Studio Lab menyediakan akses gratis ke sumber daya AWS komputasi, Amazon SageMaker Studio Classic menyediakan kemampuan pembelajaran mesin lanjutan berikut yang tidak didukung Studio Lab.

- Integrasi berkelanjutan dan pengiriman berkelanjutan (SageMaker Pipelines)
- Prediksi waktu nyata
- Pelatihan terdistribusi skala besar
- Persiapan data (Amazon SageMaker Data Wrangler)
- Pelabelan data (Amazon SageMaker Ground Truth)
- Toko Fitur
- Analisis bias (Klarifikasi)
- Penyebaran model

- Pemantauan model

Studio Classic juga mendukung kontrol akses dan keamanan berbutir halus dengan menggunakan AWS Identity and Access Management (IAM), Amazon Virtual Private Cloud (Amazon VPC), dan (). AWS Key Management Service AWS KMS Studio Lab tidak mendukung fitur Studio Classic ini, juga tidak mendukung penggunaan estimator dan SageMaker algoritme bawaan.

Untuk mengekspor proyek Studio Lab Anda untuk digunakan dengan Studio Classic, lihatEkspor lingkungan Amazon SageMaker Studio Lab ke Amazon SageMaker Studio Classic.

Topik berikut memberikan informasi tentang Studio Lab dan cara menggunakannya

Topik

Ikhtisar komponen Amazon SageMaker Studio Lab

Amazon SageMaker Studio Lab terdiri dari komponen-komponen berikut. Topik berikut memberikan rincian lebih lanjut tentang komponen-komponen ini.

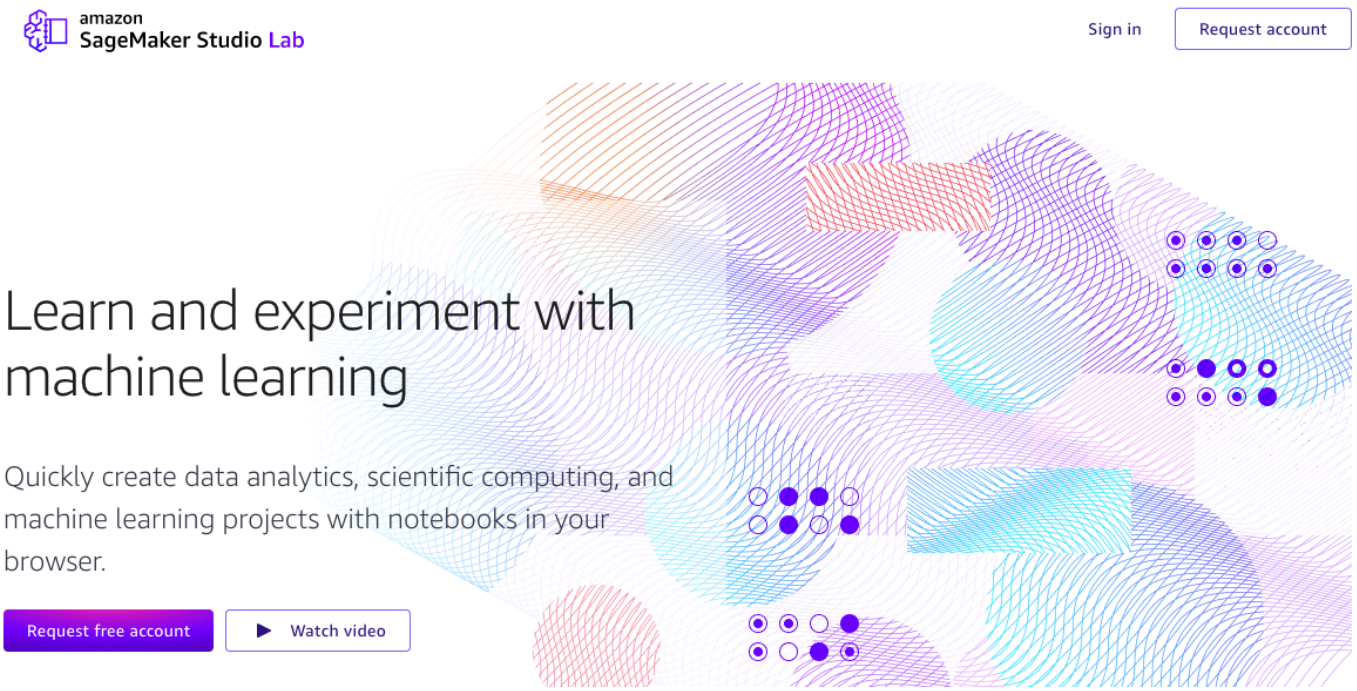
Topik

- Sesi

Halaman arahan

Anda dapat meminta akun dan masuk ke akun yang ada di halaman landing Anda. Untuk menavigasi ke halaman arahan, lihat situs web Amazon SageMaker Studio Lab. Untuk informasi selengkapnya tentang membuat akun Studio Lab, lihatNaik ke Amazon SageMaker Studio Lab.

Tangkapan layar berikut menunjukkan antarmuka halaman landing Studio Lab untuk meminta akun pengguna dan masuk.



Akun Studio Lab

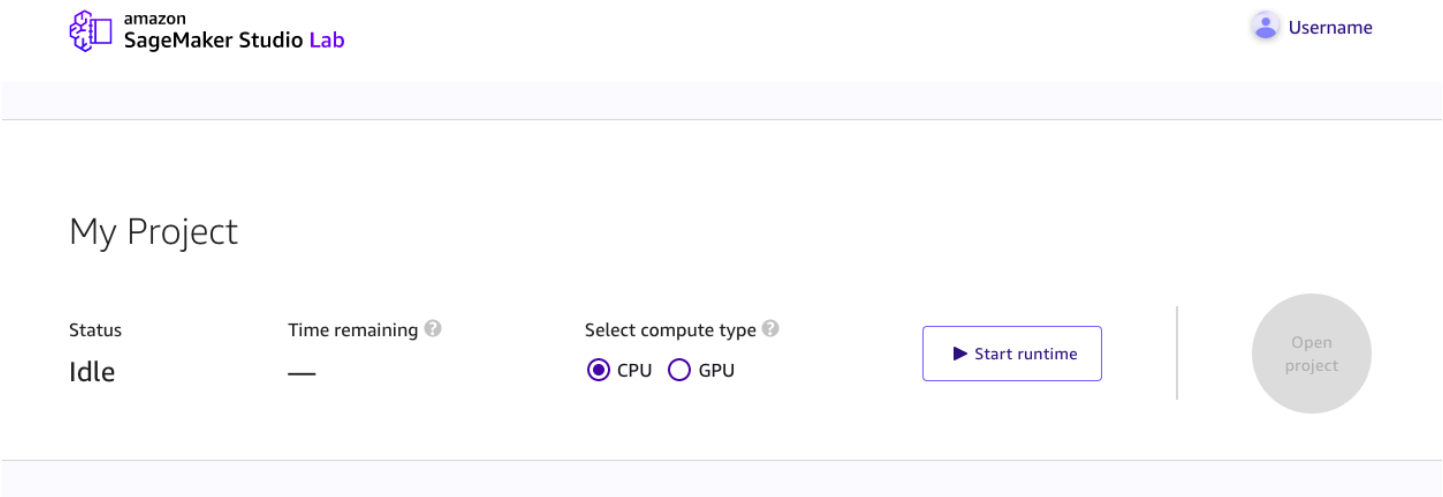
Akun Studio Lab Anda memberi Anda akses ke Studio Lab. Untuk informasi selengkapnya tentang membuat akun pengguna, lihatNaik ke Amazon SageMaker Studio Lab.

Halaman ikhtisar proyek

Anda dapat meluncurkan instance komputasi dan melihat informasi tentang proyek Anda di halaman ini. Untuk menavigasi ke halaman ini, Anda harus masuk dari situs web Amazon SageMaker Studio Lab. URL mengambil format berikut.

```
https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
```

Tangkapan layar berikut menunjukkan ikhtisar proyek di antarmuka pengguna Studio Lab.



Halaman pratinjau

Di halaman ini, Anda dapat mengakses pratinjau read-only notebook Jupyter. Anda tidak dapat menjalankan notebook dari pratinjau, tetapi Anda dapat menyalin buku catatan itu ke proyek Anda. Bagi banyak pelanggan, ini mungkin halaman Studio Lab pertama yang dilihat pelanggan, karena mereka mungkin membuka buku catatan dari GitHub notebook. Untuk informasi selengkapnya tentang cara menggunakan GitHub sumber daya, lihatGunakan GitHub sumber daya.

Untuk menyalin pratinjau buku catatan ke project Studio Lab Anda:

1. Masuk ke akun Studio Lab Anda. Untuk informasi selengkapnya tentang membuat akun Studio Lab, lihatNaik ke Amazon SageMaker Studio Lab.
2. Di bawah Instance komputasi Notebook, pilih jenis instans komputasi. Untuk informasi selengkapnya tentang jenis instans komputasi, lihatJenis instans komputasi.
3. Pilih Mulai runtime. Anda mungkin diminta untuk memecahkan teka-teki CAPTCHA. Untuk informasi lebih lanjut tentang CAPTCHA, lihat Apa itu teka-teki CAPTCHA?
4. Penyiapan satu kali, untuk pertama kalinya memulai runtime menggunakan akun Studio Lab Anda:

    a. Masukkan nomor ponsel untuk diasosiasikan dengan akun Amazon SageMaker Studio Lab Anda dan pilih Lanjutkan.

    Untuk informasi tentang negara dan wilayah yang didukung, lihat Negara dan wilayah yang didukung (saluran SMS).

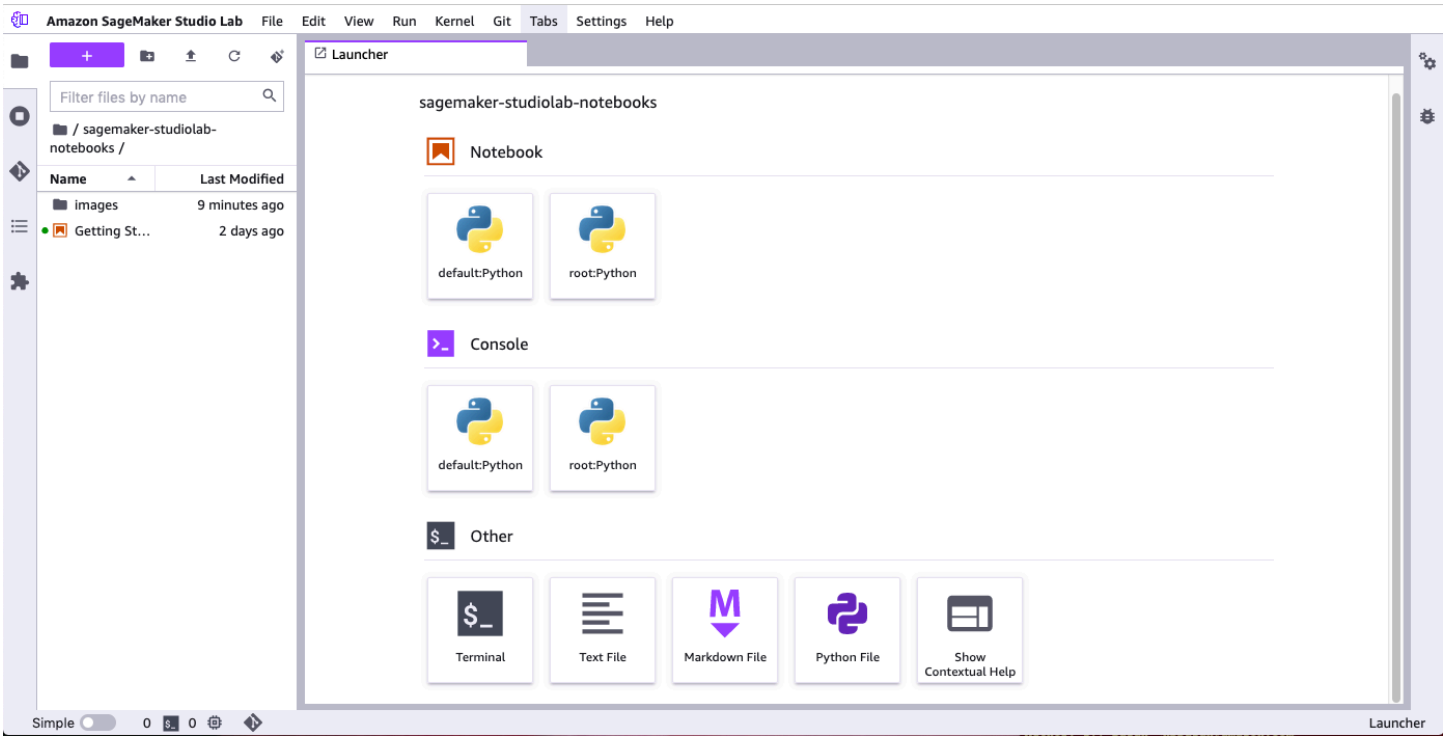
    b. Masukkan kode 6 digit yang dikirim ke nomor ponsel terkait dan pilih Verifikasi.

5. Pilih Salin ke proyek.

Proyek

Proyek Anda berisi semua file dan folder Anda, termasuk buku catatan Jupyter Anda. Anda memiliki kontrol penuh atas file dalam proyek Anda. Proyek Anda juga menyertakan antarmuka pengguna JupyterLab berbasis. Dari antarmuka ini, Anda dapat berinteraksi dengan notebook Jupyter Anda, mengedit file kode sumber Anda, mengintegrasikan dengan GitHub, dan terhubung ke Amazon S3. Untuk informasi selengkapnya, lihat Menggunakan runtime proyek Amazon SageMaker Studio Lab.

Tangkapan layar berikut menunjukkan proyek Studio Lab dengan browser file terbuka dan Studio Lab Launcher ditampilkan.



Jenis instans komputasi

Runtime project Amazon SageMaker Studio Lab Anda didasarkan pada instans EC2. Anda dialokasikan 15 GB penyimpanan dan 16 GB RAM. Ketersediaan instans komputasi tidak dijamin dan tunduk pada permintaan. Jika Anda memerlukan penyimpanan tambahan atau sumber daya komputasi, pertimbangkan untuk beralih ke Studio.

Amazon SageMaker Studio Lab menawarkan pilihan CPU (Central Processing Unit) dan GPU (Graphical Processing Unit). Bagian berikut memberikan informasi tentang dua opsi ini, termasuk panduan pemilihan.

CPU

Central Processing Unit (CPU) dirancang untuk menangani berbagai tugas secara efisien, tetapi terbatas pada berapa banyak tugas yang dapat dijalankan secara bersamaan. Untuk pembelajaran mesin, CPU direkomendasikan untuk menghitung algoritma intensif, seperti deret waktu, peramalan, dan data tabular.

Jenis komputasi CPU memiliki hingga 4 jam sekaligus dengan batas 8 jam dalam periode 24 jam.

GPU

Graphics Processing Unit (GPU) dirancang untuk membuat gambar dan video beresolusi tinggi secara bersamaan. GPU direkomendasikan untuk tugas pembelajaran mendalam, terutama untuk transformer dan visi komputer.

Jenis komputasi GPU memiliki hingga 4 jam sekaligus dengan batas 4 jam dalam periode 24 jam.

Menghitung waktu

Saat waktu komputasi untuk Studio Lab mencapai batas waktunya, instance menghentikan semua perhitungan yang sedang berjalan. Studio Lab tidak mendukung peningkatan batas waktu.

Studio Lab secara otomatis menyimpan lingkungan Anda ketika Anda memperbarui lingkungan Anda dan setiap kali Anda membuat file baru. Ekstensi dan paket yang diinstal khusus tetap ada bahkan setelah runtime Anda berakhir.

Pengeditan file disimpan secara berkala, tetapi tidak disimpan saat runtime Anda berakhir. Untuk memastikan bahwa Anda tidak kehilangan kemajuan Anda, simpan pekerjaan Anda secara manual. Jika Anda memiliki konten dalam proyek Studio Lab yang tidak ingin Anda hilangkan, sebaiknya Anda membuat cadangan konten di tempat lain. Untuk informasi selengkapnya tentang mengekspor lingkungan dan file Anda, lihatEkspor lingkungan Amazon SageMaker Studio Lab ke Amazon SageMaker Studio Classic.

Selama perhitungan panjang, Anda tidak perlu membiarkan proyek Anda tetap terbuka. Misalnya, Anda dapat mulai melatih model, lalu menutup browser Anda. Instance terus berjalan hingga batas jenis komputasi dalam periode 24 jam. Anda kemudian dapat masuk nanti untuk melanjutkan pekerjaan Anda.

Kami menyarankan Anda menggunakan checkpointing dalam pekerjaan pembelajaran mendalam Anda. Anda dapat menggunakan pos pemeriksaan yang disimpan untuk memulai kembali pekerjaan dari pos pemeriksaan yang disimpan sebelumnya. Untuk informasi selengkapnya, lihat File I/O.

Runtime proyek

Runtime proyek adalah periode waktu saat instance komputasi Anda berjalan.

Sesi

Sesi pengguna dimulai setiap kali Anda meluncurkan proyek Anda.

Naik ke Amazon SageMaker Studio Lab

Untuk bergabung ke Amazon SageMaker Studio Lab, ikuti langkah-langkah dalam panduan ini. Di bagian berikut, Anda mempelajari cara meminta akun Studio Lab, membuat akun, dan masuk.

Topik

Meminta akun Studio Lab

Untuk menggunakan Studio Lab, Anda harus terlebih dahulu meminta persetujuan untuk membuat akun Studio Lab. AWSAkun tidak dapat digunakan untuk orientasi ke Studio Lab.

Langkah-langkah berikut menunjukkan cara meminta Studio Lab.

1. Arahkan ke halaman arahan Studio Lab.
2. Pilih Minta akun.
3. Masukkan informasi yang diperlukan ke dalam formulir.
4. Pilih Kirim permintaan.
5. Jika menerima email untuk memverifikasi alamat email, ikuti petunjuk di email untuk menyelesaikan langkah ini.

Permintaan akun Anda harus disetujui sebelum Anda dapat mendaftar untuk akun Studio Lab. Permintaan Anda akan ditinjau dalam waktu lima kerja. Ketika permintaan akun disetujui, Anda

menerima email berisi tautan ke halaman pendaftaran Studio Lab. Tautan ini kedaluwarsa tujuh hari setelah permintaan Anda disetujui. Jika tautan kedaluwarsa, Anda harus mengirimkan permintaan akun baru.

Catatan: Permintaan akun Anda ditolak jika email Anda dikaitkan dengan aktivitas yang melanggar Ketentuan Layanan kami atau perjanjian lainnya.

Kode rujukan

Kode rujukan Studio Lab memungkinkan permintaan akun baru disetujui secara otomatis untuk mendukung acara pembelajaran mesin seperti lokakarya, hackathon, dan kelas. Dengan kode referral, tuan rumah tepercaya bisa mendapatkan peserta akses langsung ke Studio Lab. Setelah akun dibuat menggunakan kode referensi, akun terus ada setelah berakhirnya kode.

Untuk mendapatkan kode referensi, hubungi Support Penjualan. Untuk menggunakan kode referensi, masukkan kode sebagai bagian dari formulir permintaan akun.

Membuat Studio Lab

Setelah permintaan disetujui, selesaikan langkah-langkah berikut untuk membuat Studio Lab Anda.

1. Pilih Buat akun di email persetujuan permintaan akun untuk membuka halaman baru.
2. Dari halaman baru, masukkan Email Anda, Kata Sandi, dan Nama Pengguna.
3. Pilih Buat akun.

   Anda mungkin diminta untuk memecahkan teka-teki CAPTCHA. Untuk informasi lebih lanjut tentang CAPTCHA, lihat Apa itu teka-teki CAPTCHA?

Masuk ke Studio Lab

Setelah mendaftar akun, Anda dapat masuk ke Studio Lab.

1. Arahkan ke halaman arahan Studio Lab.
2. Pilih Masuk untuk membuka halaman baru.
3. Masukkan Email atau Nama Pengguna dan Kata Sandi Anda.
4. Pilih Masuk untuk membuka halaman baru untuk proyek Anda.

   Anda mungkin diminta untuk memecahkan teka-teki CAPTCHA. Untuk informasi lebih lanjut tentang CAPTCHA, lihat Apa itu teka-teki CAPTCHA?

Mengelola akun Anda

Topik berikut memberikan informasi tentang mengelola akun Anda, termasuk mengubah kata sandi, menghapus akun Anda, dan mendapatkan informasi yang telah kami kumpulkan. Topik ini mengharuskan Anda masuk ke akun Amazon SageMaker Studio Lab Anda. Untuk informasi selengkapnya, lihat Masuk ke Studio Lab.

Ubah kata sandi Anda

Ikuti langkah-langkah berikut untuk mengubah kata sandi Amazon SageMaker Studio Lab Anda.

1. Arahkan ke halaman ikhtisar proyek Studio Lab. URL mengambil format berikut.

   ```
   https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
   ```

2. Dari sudut kanan atas, pilih nama pengguna Anda untuk membuka menu tarik-turun.
3. Dari menu tarik-turun, pilih Ubah kata sandi untuk membuka halaman baru.
4. Masukkan kata sandi Anda saat ini ke kolom Masukkan kata sandi Anda saat ini.
5. Masukkan kata sandi baru Anda ke kolom Buat kata sandi baru dan Konfirmasi kata sandi baru Anda.
6. Pilih Kirim.

Menghapus akun Anda

Ikuti langkah-langkah ini untuk menghapus akun Studio Lab Anda.

1. Arahkan ke halaman ikhtisar proyek Studio Lab. URL mengambil format berikut.

   ```
   https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
   ```

2. Dari sudut kanan atas, pilih nama pengguna Anda untuk membuka menu tarik-turun.
3. Dari menu tarik-turun, pilih Hapus akun untuk membuka halaman baru.
4. Masukkan kata sandi Anda untuk mengonfirmasi penghapusan akun Studio Lab Anda.
5. Pilih Hapus.

informasi pelanggan

Studio Lab mengumpulkan alamat email, nama pengguna, sandi terenkripsi, file proyek, dan metadata Anda. Saat meminta akun, Anda dapat memilih untuk memberikan nama depan dan belakang, negara, nama organisasi, pekerjaan, dan alasan minat Anda pada produk ini. Kami melindungi semua data pribadi pelanggan dengan enkripsi. Untuk informasi selengkapnya tentang cara penanganan informasi pribadi Anda, lihat Pemberitahuan Privasi.

Saat Anda menghapus akun, semua informasi Anda segera dihapus. Jika Anda memiliki pertanyaan tentang hal ini, kirimkan Formulir Amazon SageMaker Studio Lab. Untuk informasi dan dukungan terkaitAWS kepatuhan, lihat Dukungan kepatuhan.

Luncurkan runtime proyek Amazon SageMaker Studio Lab

Runtime project Amazon SageMaker Studio Lab memungkinkan Anda menulis dan menjalankan kode langsung dari browser Anda. Ini didasarkan pada JupyterLab dan memiliki terminal dan konsol terintegrasi. Untuk informasi selengkapnyaJupyterLab, lihat JupyterLabDokumentasi.

Topik berikut memberikan informasi tentang cara mengelola runtime proyek Anda. Topik ini mengharuskan Anda masuk ke akun Amazon SageMaker Studio Lab Anda. Untuk informasi selengkapnya tentang masuk, lihatMasuk ke Studio Lab. Untuk informasi lebih lanjut tentang proyek Anda, lihatIkhtisar komponen Amazon SageMaker Studio Lab.

Topik

- Mulai Runtime Proyek Anda
- Hentikan runtime proyek Anda
- Lihat sisa waktu komputasi
- Ubah Tipe komputasi Anda

Mulai Runtime Proyek Anda

Untuk menggunakan Studio Lab, Anda harus memulai runtime proyek Anda. Runtime ini memberi Anda akses ke JupyterLab lingkungan.

1. Arahkan ke halaman ikhtisar proyek Studio Lab. URL mengambil format berikut.

```
https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
```

2. Di bawah Proyek Saya, pilih jenis komputasi. Untuk informasi selengkapnya tentang tipe komputasi, lihatJenis instans komputasi.
3. Pilih Mulai runtime.

   Anda mungkin diminta untuk memecahkan teka-teki CAPTCHA. Untuk informasi lebih lanjut tentang CAPTCHA, lihat Apa itu teka-teki CAPTCHA?
4. Penyiapan satu kali, untuk pertama kali memulai runtime menggunakan akun Studio Lab Anda:

   a. Masukkan nomor ponsel untuk dikaitkan dengan akun Amazon SageMaker Studio Lab Anda dan pilih Lanjutkan.

      Untuk informasi tentang negara dan wilayah yang didukung, lihat Negara dan wilayah yang didukung (saluran SMS).

   b. Masukkan kode 6 digit yang dikirim ke nomor ponsel terkait dan pilih Verifikasi.
5. Setelah runtime berjalan, pilih Open project untuk membuka lingkungan runtime proyek di tab browser baru.

Hentikan runtime proyek Anda

Ketika Anda menghentikan runtime proyek Anda, file Anda tidak disimpan secara otomatis. Untuk memastikan bahwa Anda tidak kehilangan pekerjaan Anda, simpan semua perubahan Anda sebelum menghentikan runtime proyek Anda.

- Di bawah Proyek Saya, pilih Hentikan runtime.

Lihat sisa waktu komputasi

Runtime proyek Anda memiliki waktu komputasi terbatas berdasarkan jenis komputasi yang Anda pilih. Untuk informasi selengkapnya tentang waktu komputasi di Studio Lab, lihatJenis instans komputasi.

- Di bawah Proyek Saya, lihat Sisa waktu.

Ubah Tipe komputasi Anda

Anda dapat mengganti jenis komputasi berdasarkan alur kerja Anda. Untuk informasi selengkapnya tentang tipe komputasi, lihatJenis instans komputasi.

1. Simpan file proyek apa pun sebelum mengubah jenis komputasi.
2. Arahkan ke halaman ikhtisar proyek Studio Lab. URL mengambil format berikut.

   ```
   https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
   ```

3. Di bawah Proyek Saya, pilih jenis komputasi yang diinginkan (CPU atau GPU).
4. Konfirmasikan pilihan Anda dengan memilih Restart di runtime proyek Restart? kotak dialog. Studio Lab menghentikan runtime proyek Anda saat ini, lalu memulai runtime proyek baru dengan jenis komputasi yang diperbarui.
5. Setelah runtime proyek Anda dimulai, pilih Open project. Ini terbuka lingkungan runtime proyek Anda di tab peramban baru. Untuk informasi tentang penggunaan lingkungan runtime proyek Anda, lihatMenggunakan runtime proyek Amazon SageMaker Studio Lab.

Menggunakan aset starter Amazon SageMaker Studio Lab

Amazon SageMaker Studio Lab mendukung aset berikut untuk membantu praktisi machine learning (ML) memulai. Panduan ini menunjukkan cara untuk mengkloning notebook untuk proyek Anda.

Memulai notebook

Studio Lab dilengkapi dengan notebook starter yang memberikan informasi umum dan memandu Anda melalui alur kerja utama. Ketika Anda meluncurkan runtime proyek Anda untuk pertama kalinya, notebook ini secara otomatis terbuka.

Menyelam ke Deep Learning

Dive into Deep Learning (D2L) adalah buku interaktif, open-source yang mengajarkan ide-ide, teori matematika, dan kode yang memperkuat pembelajaran mesin. Dengan lebih dari 150 notebook Jupyter, D2L memberikan gambaran menyeluruh tentang prinsip-prinsip pembelajaran mendalam. Untuk informasi lebih lanjut tentang D2L, lihat situs web D2L.

Prosedur berikut menunjukkan cara mengkloning notebook D2L Jupyter pada instans Anda.

1. Mulai dan buka lingkungan runtime proyek Studio Lab dengan mengikutiMulai Runtime Proyek Anda.
2. Setelah Studio Lab terbuka, pilih tab Git
   (
)
   di sidebar kiri.

3. Pilih Clone Repository. Di bawah Git repository URL (.git) paste MLU git repositori D2L dengan mengikuti langkah-langkah di bawah ini. Jika Anda tidak melihat opsi Clone a Repository karena Anda saat ini berada di repositori Git, kembali ke direktori pengguna untuk mengkloning repositori baru. Anda kembali ke direktori pengguna dengan memilih tab Folder () di sidebar kiri. Di tab Folder di bawah bilah pencarian file pilih ikon folder di sebelah kiri repositori yang sedang terbuka. Setelah Anda berada di direktori pengguna, pilih tab Git di sidebar kiri dan pilih Clone a Repository.
4. Arahkan ke halaman ikhtisar proyek Studio Lab. URL mengambil format berikut.

   ```
   https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
   ```
5. Di bawah Baru untuk pembelajaran mesin? , pilih Dive into Deep Learning.
6. Dari tab browser Dive into Deep Learning yang baru, pilih GitHubuntuk membuka halaman baru dengan contoh buku catatan.
7. Pilih Kode dan salin URL GitHub repositori di tab HTTPS.
8. Kembali ke tab browser proyek terbuka Studio Lab, tempel URL repositori D2L, dan kloning repositori.

AWSUniversitas Machine Learning

AWSMachine Learning University (MLU) menyediakan akses ke kursus pembelajaran mesin yang digunakan untuk melatih pengembang Amazon sendiri. DenganAWS MLU, pengembang mana pun dapat mempelajari cara menggunakan pembelajaran mesin dengan seri pembelajaran MLU Accelerator learn-at-your-own -pace. Seri MLU Accelerator dirancang untuk membantu pengembang memulai perjalanan MLnya. Ini menawarkan kursus dasar tiga hari pada tiga mata pelajaran ini: Pengolahan Bahasa Alami, Data Tabular, dan Visi Komputer. Untuk informasi selengkapnya, lihat Machine Learning University.

Prosedur berikut menunjukkan cara mengkloning notebookAWS MLU Jupyter pada instans Anda.

1. Mulai dan buka lingkungan runtime proyek Studio Lab dengan mengikutiMulai Runtime Proyek Anda.
2. Setelah Studio Lab terbuka, pilih tab Git () di sidebar kiri.

3. Pilih Clone Repository. Di bawah URL repositori Git (.git) tempelkan URL repositori git MLU dengan mengikuti langkah-langkah di bawah ini. Jika Anda tidak melihat opsi Clone a Repository karena Anda saat ini berada di repositori Git, kembali ke direktori pengguna untuk mengkloning repositori baru. Anda kembali ke direktori pengguna dengan memilih tab Folder ( ) di sidebar kiri. Di tab Folder di bawah bilah pencarian file pilih ikon folder di sebelah kiri repositori yang sedang terbuka. Setelah Anda berada di direktori pengguna, pilih tab Git di sidebar kiri dan pilih Clone a Repository.
4. Arahkan ke halaman ikhtisar proyek Studio Lab. URL mengambil format berikut.

   ```
   https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
   ```

5. Di bawah Baru untuk pembelajaran mesin? , pilih AWSMachine Learning University.
6. Dari tab browser AWSMachine Learning University yang baru, temukan kursus yang menarik minat Anda dengan membaca Ringkasan Kursus untuk setiap kursus.
7. Pilih GitHub repositori bunga yang sesuai di bawah Konten Kursus, untuk membuka halaman baru dengan contoh notebook.
8. Pilih Kode dan salin URL GitHub repositori di tab HTTPS.
9. Kembali ke tab browser proyek terbuka Studio Lab, tempel URL repositori D2L, dan pilih Clone untuk mengkloning repositori.

Roboflow

Roboflow memberi Anda alat untuk melatih, menyempurnakan, dan memberi label pada objek untuk aplikasi penglihatan komputer. Untuk informasi selengkapnya, lihat https://roboflow.com/.

Prosedur berikut menunjukkan cara mengkloning notebook Roboflow Jupyter pada instans Anda.

1. Arahkan ke halaman ikhtisar proyek Studio Lab. URL mengambil format berikut.

   ```
   https://studiolab.sagemaker.aws/users/<YOUR_USER_NAME>
   ```

2. Di bawah Sumber Daya dan komunitas, temukan Coba Computer Vision.
3. Di bawah Coba Computer Vision pilih model Roboflow. Untuk informasi selengkapnya, lihat https://roboflow.com/.
4. Ikuti tutorial di bawah pratinjau Notebook.

Lingkungan pra-instal Studio Lab

Amazon SageMaker Studio Lab menggunakan lingkungan conda untuk memuat paket (atau pustaka) Anda. Lingkungan adalah folder yang berisi paket yang telah Anda instal. Anda dapat berinteraksi dengan lingkungan dengan menggunakan terminal atau JupyterLab notebook Anda. Untuk menggunakan lingkungan dan paket yang diinstal di dalamnya, Anda harus memilih kernel yang sesuai yang berisi nama yang sama dengan lingkungan saat membuka JupyterLab notebook Anda. Untuk panduan tentang cara mengelola lingkungan Anda, lihat. Kelola lingkungan Anda Untuk informasi selengkapnya tentang menginstal paket di lingkungan Anda, lihatSesuaikan lingkungan Anda.

Studio Lab memiliki berbagai lingkungan yang sudah diinstal sebelumnya untuk Anda. Setiap perubahan yang dilakukan pada lingkungan memori persisten akan tetap ada untuk sesi Anda berikutnya. Setiap perubahan pada lingkungan memori non-persisten tidak akan tetap untuk sesi berikutnya, tetapi paket di dalamnya akan diperbarui dan diuji untuk kompatibilitasnya oleh Amazon. SageMaker Anda biasanya ingin menggunakan lingkungan memori `sagemaker-distribution` non-persisten jika Anda ingin menggunakan lingkungan yang dikelola sepenuhnya yang sudah berisi banyak paket populer yang digunakan oleh insinyur pembelajaran mesin (ML) dan ilmuwan data. Jika tidak, Anda dapat menggunakan `default` lingkungan jika Anda ingin menyesuaikan lingkungan Anda secara signifikan.

Berikut ini kami mencantumkan lingkungan pra-instal dan kasus penggunaannya. Untuk melihat paket yang diinstal di lingkungan, lihatSesuaikan lingkungan Anda.

- `sagemaker-distribution`: Lingkungan memori non-persisten yang diperbarui secara berkala dan diuji kompatibilitasnya, dikelola sepenuhnya oleh Amazon SageMaker. Lingkungan ini berisi paket populer yang digunakan dalam ML, ilmu data, dan visualisasi. `sagemaker-distribution`Lingkungan terkait erat dengan lingkungan yang digunakan di Amazon SageMaker Studio Classic, jadi setelah lulus dari Studio Lab ke Studio Classic notebook harus berjalan dengan cara yang sama. Untuk informasi tentang mengekspor lingkungan Anda dari Studio Lab ke Studio Classic, lihatEkspor lingkungan Amazon SageMaker Studio Lab ke Amazon SageMaker Studio Classic.
- `default`: Lingkungan memori persisten dengan sangat sedikit paket yang sudah diinstal sebelumnya. Setiap paket yang diinstal atau perubahan pada lingkungan ini akan berlanjut pada sesi Anda berikutnya.

- `studiolab`: Lingkungan memori persisten tempat JupyterLab dan paket terkait lainnya diinstal. Lingkungan ini hanya boleh digunakan untuk JupyterLab dan ekstensi server Jupyter, untuk mengkonfigurasi antarmuka pengguna. JupyterLab
- `studiolab-safemode`: Lingkungan memori non-persisten. Lingkungan ini diaktifkan secara otomatis ketika ada masalah saat memulai runtime proyek Anda. Digunakan untuk pemecahan masalah. Untuk informasi tentang pemecahan masalah, lihat. Memecahkan masalah
- `base`: Lingkungan memori non-persisten. Lingkungan ini hanya digunakan untuk perkakas sistem dan tidak boleh digunakan oleh pelanggan.

Untuk informasi tentang SageMaker gambar dan versinya, lihat SageMaker Gambar Amazon yang Tersedia.

Menggunakan runtime proyek Amazon SageMaker Studio Lab

Topik berikut memberikan informasi tentang penggunaan runtime proyek Amazon SageMaker Studio Lab. Sebelum Anda dapat menggunakan runtime proyek Studio Lab, Anda harus onboard ke Studio Lab dengan mengikuti langkah-langkah diNaik ke Amazon SageMaker Studio Lab.

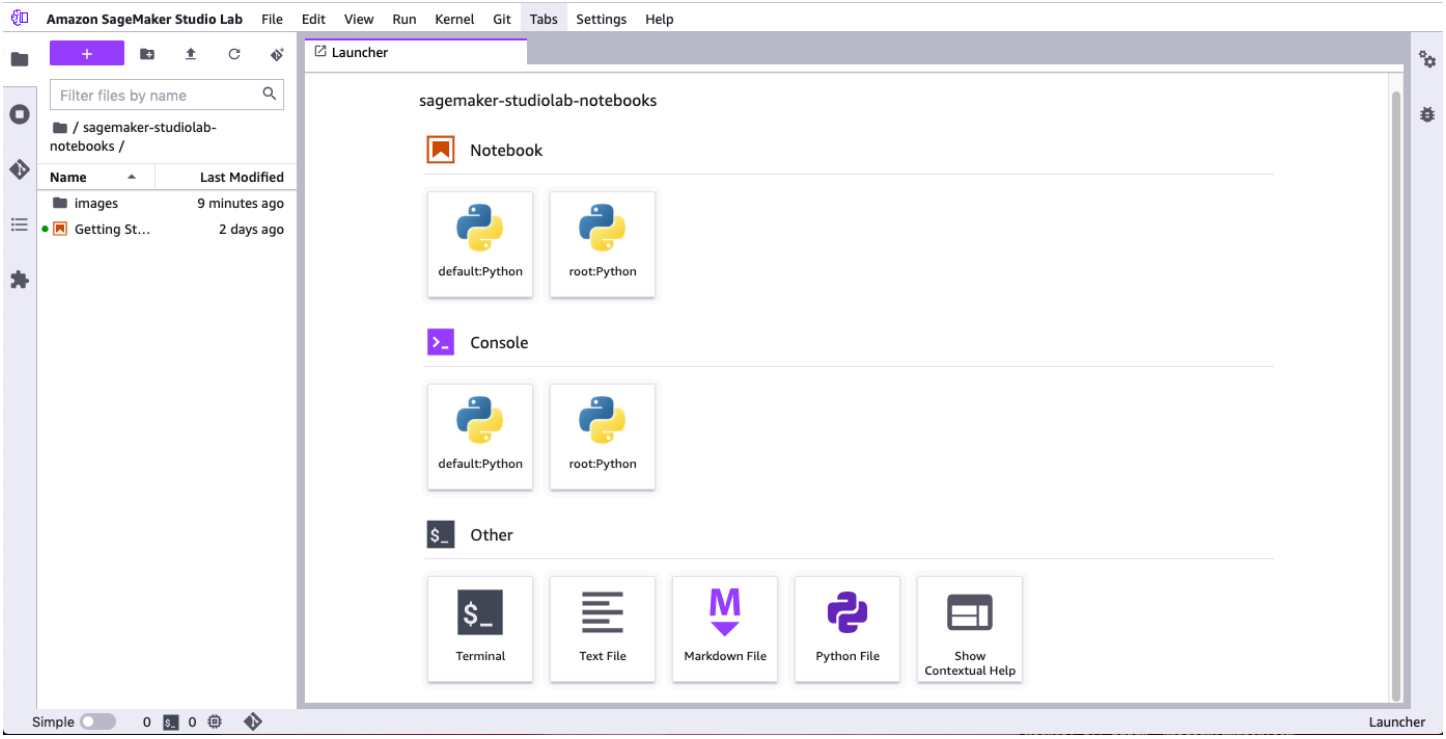
Topik

- Ikhtisar UI Amazon SageMaker Studio Lab
- Membuat atau membuka notebook Amazon SageMaker Studio Lab
- Menggunakan bilah alat notebook Amazon SageMaker Studio Lab
- Kelola lingkungan Anda
- Menggunakan sumber daya eksternal di Amazon SageMaker Studio Lab
- Dapatkan perbedaan notebook
- Ekspor lingkungan Amazon SageMaker Studio Lab ke Amazon SageMaker Studio Classic
- Matikan sumber daya

Ikhtisar UI Amazon SageMaker Studio Lab

Amazon SageMaker Studio Lab memperluas JupyterLab antarmuka. Pengguna sebelumnya JupyterLab akan melihat kesamaan antara UI Studio Lab JupyterLab dan, termasuk ruang kerja. Untuk ikhtisar JupyterLab antarmuka dasar, lihat JupyterLabAntarmuka.

Gambar berikut menunjukkan Studio Lab dengan browser file terbuka dan Studio Lab Launcher ditampilkan.



Anda akan menemukan bilah menu di bagian atas layar. Bilah sisi kiri berisi ikon untuk membuka browser file, browser sumber daya, dan alat. Bilah status terletak di sudut kiri bawah Studio Lab.

Area kerja utama dibagi secara horizontal menjadi dua panel. Panel kiri adalah file dan browser sumber daya. Panel kanan berisi satu atau beberapa tab untuk sumber daya, seperti notebook dan terminal.

Topik

- Sidebar kiri
- Peramban file dan sumber daya
- Area kerja utama

Sidebar kiri

Sidebar kiri mencakup ikon berikut. Saat Anda mengarahkan kursor ke ikon, tooltip akan menampilkan nama ikon. Saat Anda memilih ikon, file dan browser sumber daya menampilkan fungsionalitas yang dijelaskan. Untuk entri hirarkis, breadcrumb yang dapat dipilih di bagian atas browser menunjukkan lokasi Anda dalam hierarki.

Ikon	Deskripsi
	Peramban File Pilih ikon Upload Files ( ) untuk menambahkan file ke Studio Lab. Klik dua kali file untuk membuka file di tab baru. Untuk membuka file yang berdekatan, pilih tab yang berisi buku catatan, Python, atau file teks, lalu pilih Tampilan Baru untuk File. Pilih tanda plus (+) pada menu di bagian atas browser file untuk membuka Studio Lab Launcher.
	Menjalankan Terminal dan Kernel Anda dapat melihat daftar semua terminal dan kernel yang sedang berjalan di proyek Anda. Untuk informasi selengkapnya, lihat Matikan sumber daya.
	Git Anda dapat terhubung ke repositori Git dan kemudian mengakses berbagai alat dan operasi Git. Untuk informasi selengkapnya, lihat Menggunakan sumber daya eksternal di Amazon SageMaker Studio Lab.
	Daftar Isi Anda dapat mengakses Daftar Isi untuk notebook Jupyter Anda saat ini.
	Manajer Ekstensi Anda dapat mengaktifkan dan mengelola JupyterLab ekstensi pihak ketiga.

Peramban file dan sumber daya

File dan browser sumber daya menunjukkan daftar notebook dan file Anda. Pada menu di bagian atas browser file, pilih tanda plus (+) untuk membuka Studio Lab Launcher. Peluncur memungkinkan Anda membuat notebook atau membuka terminal.

Area kerja utama

Area kerja utama memiliki beberapa tab yang berisi notebook dan terminal terbuka Anda.

Membuat atau membuka notebook Amazon SageMaker Studio Lab

Saat Anda membuat notebook di Amazon SageMaker Studio Lab atau membuka notebook di Studio Lab, Anda harus memilih kernel untuk notebook. Topik berikut menjelaskan cara membuat dan membuka Notebook di Studio Notebook.

Untuk informasi tentang mematikan notebook, lihatMatikan sumber daya.

Topik

- Buka notebook Studio Lab
- Buat buku catatan dari menu file
- Buat buku catatan dari Peluncur

Buka notebook Studio Lab

Studio Lab hanya dapat membuka notebook yang tercantum di browser file Studio Lab. Untuk mengkloning notebook ke browser file Anda dari repositori eksternal, lihatMenggunakan sumber daya eksternal di Amazon SageMaker Studio Lab.

Untuk membuka buku catatan

1. Di bilah sisi kiri, pilih ikon File Browser
   ( )
   untuk menampilkan browser file.
2. Jelajahi file notebook dan klik dua kali untuk membuka buku catatan di tab baru.

Buat buku catatan dari menu file

Membuat buku catatan dari menu File

1. Dari menu Studio Lab, pilih File, pilih Baru, lalu pilih Notebook.
2. Untuk menggunakan kernel default, di kotak dialog Select Kernel, pilih Select. Jika tidak, untuk memilih kernel yang berbeda, gunakan menu dropdown.

Buat buku catatan dari Peluncur

Untuk membuat notebook dari Launcher

1. Buka Peluncur dengan menggunakan pintasan keyboard`Ctrl + Shift + L`.

   Atau, Anda dapat membuka Peluncur dari bilah sisi kiri: Pilih ikon File Browser, lalu pilih ikon plus (+).
2. Untuk menggunakan kernel default dari Launcher, di bawah Notebook, pilih Default:Python. Jika tidak, pilih kernel yang berbeda.

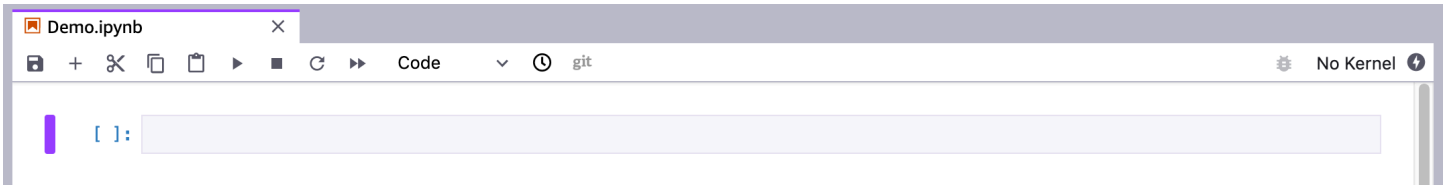
Setelah Anda memilih kernel, notebook Anda akan diluncurkan dan terbuka di tab Studio Lab baru.

Untuk melihat sesi kernel notebook, di bilah sisi kiri, pilih ikon Running Terminals and Kernels (). Anda dapat menghentikan sesi kernel notebook dari tampilan ini.

Menggunakan bilah alat notebook Amazon SageMaker Studio Lab

Notebook Amazon SageMaker Studio Lab memperluas JupyterLab antarmuka. Untuk ikhtisar JupyterLab antarmuka dasar, lihat JupyterLabAntarmuka.

Gambar berikut menunjukkan toolbar dan sel kosong dari notebook Studio Lab.



Saat Anda mengarahkan kursor ke ikon bilah alat, tooltip menampilkan fungsi ikon. Anda dapat menemukan perintah notebook tambahan di menu utama Studio Lab. Bilah alat mencakup ikon berikut:

Ikon	Deskripsi
	Simpan dan pos pemeriksaan Menyimpan notebook dan memperbarui file pos pemeriksaan.
	Masukkan sel Menyisipkan sel kode di bawah sel saat ini. Sel saat ini dicatat oleh penanda vertikal biru di margin kiri.
	Potong, salin, dan tempel sel Memotong, menyalin, dan menempelkan sel yang dipilih.
	Jalankan sel Menjalankan sel yang dipilih. Sel yang mengikuti sel yang dipilih terakhir menjadi sel yang baru dipilih.
	Kernel interupsi Menginterupsi kernel, yang membatalkan operasi yang sedang berjalan. Kernel tetap aktif.
	Mulai ulang kernel Memulai ulang kernel. Variabel diatur ulang. Informasi yang belum terpengaruh.
	Mulai ulang kernel dan jalankan ulang notebook Memulai ulang kernel. Variabel diatur ulang. Informasi yang belum terpengaruh. Kemudian jalankan kembali seluruh notebook.
	Jenis sel Menampilkan atau mengubah jenis sel saat ini. Jenis sel adalah: • Kode - Kode yang dijalankan kernel. • Penurunan harga - Teks diberikan sebagai penurunan harga.

Ikon	Deskripsi
	• Mentah - Konten, termasuk markup penurunan harga, yang ditampilkan sebagai teks.
	Diff pos pemeriksaan Membuka tab baru yang menampilkan perbedaan antara notebook dan file pos pemeriksaan. Untuk informasi selengkapnya, lihat Dapatkan perbedaan notebook.
	Git diff Hanya diaktifkan jika notebook dibuka dari repositori Git. Membuka tab baru yang menampilkan perbedaan antara notebook dan commit Git terakhir. Untuk informasi selengkapnya, lihat Dapatkan perbedaan notebook.
default	Kernel Menampilkan atau mengubah kernel yang memproses sel di notebook. No `Kernel`menunjukkan bahwa notebook dibuka tanpa menentukan kernel. Anda dapat mengedit notebook, tetapi Anda tidak dapat menjalankan sel apa pun.
	Status sibuk kernel Menampilkan status sibuk kernel dengan menunjukkan tepi lingkaran dan interiornya dengan warna yang sama. Kernel sibuk ketika mulai dan ketika sedang memproses sel. Status kernel tambahan ditampilkan di bilah status di sudut kiri bawah Studio Lab.

Kelola lingkungan Anda

Amazon SageMaker Studio Lab menyediakan lingkungan pra-instal untuk instans notebook Studio Lab Anda. Lingkungan memungkinkan Anda memulai instance notebook Studio Lab dengan paket yang ingin Anda gunakan. Ini dilakukan dengan menginstal paket di lingkungan dan kemudian memilih lingkungan sebagai Kernel.

Studio Lab memiliki berbagai lingkungan yang sudah diinstal sebelumnya untuk Anda. Anda biasanya ingin menggunakan`sagemaker-distribution`lingkungan jika Anda ingin menggunakan lingkungan yang dikelola sepenuhnya yang sudah berisi banyak paket populer yang digunakan untuk insinyur pembelajaran mesin (ML) dan ilmuwan data. Jika tidak, Anda dapat menggunakan`default`lingkungan jika Anda ingin kustomisasi persisten untuk lingkungan Anda. Untuk informasi selengkapnya tentang lingkungan Studio Lab yang sudah diinstal sebelumnya, lihatLingkungan pra-instal Studio Lab.

Anda dapat menyesuaikan lingkungan Anda dengan menambahkan paket baru (atau pustaka) ke dalamnya. Anda juga dapat membuat lingkungan baru dari Studio Lab, mengimpor lingkungan yang kompatibel, mengatur ulang lingkungan untuk menciptakan ruang, dan banyak lagi.

Perintah berikut adalah untuk berjalan di terminal Studio Lab. Namun, saat menginstal paket, sangat disarankan untuk menginstalnya di dalam notebook Studio Lab Jupyter Anda untuk memastikan bahwa paket diinstal di lingkungan yang dimaksud. Untuk menjalankan perintah di notebook Jupyter, awali perintah dengan`%`sebelum menjalankan sel. Misalnya, cuplikan kode`pip list`di terminal sama dengan`%pip list`di buku catatan Jupyter.

Bagian berikut memberikan informasi tentang`default`lingkungan conda, cara menyesuaikannya, dan cara menambah dan menghapus lingkungan conda. Untuk daftar contoh lingkungan yang dapat Anda instal ke Studio Lab, lihatMembuat Lingkungan Conda Kustom. Untuk menggunakan contoh file YAMAL lingkungan ini dengan Studio Lab, lihatLangkah 4: Instal lingkungan conda Studio Lab Anda di Studio Classic.

Topik

- Lingkungan default Anda
- Lihat lingkungan
- Buat, aktifkan, dan gunakan lingkungan conda baru
- Menggunakan lingkungan Studio Lab sampel
- Sesuaikan lingkungan Anda
- Segarkan Lab Studio

Lingkungan default Anda

Studio Lab menggunakan lingkungan conda untuk merangkum paket perangkat lunak yang diperlukan untuk menjalankan notebook. Proyek Anda berisi lingkungan conda default,

bernama`default`, dengan Kernel IPython. Lingkungan ini berfungsi sebagai kernel default untuk notebook Jupyter Anda.

Lihat lingkungan

Untuk melihat lingkungan di Studio Lab, Anda dapat menggunakan terminal atau notebook Jupyter. Perintah berikut akan untuk terminal Studio Lab. Jika Anda ingin menjalankan perintah yang sesuai di notebook Jupyter, lihat Kelola lingkungan Anda.

Buka terminal Studio Lab dengan membukaPeramban

Berkas(
panel, pilih plus (+) masuk di menu di bagian atas file untuk membukaPeluncur, lalu pilihTerminal. Dari terminal Studio Lab, daftarkan lingkungan conda dengan menjalankan yang berikut ini.

```
conda env list
```

Perintah ini menampilkan daftar lingkungan conda dan lokasinya di sistem file. Saat Anda melakukan onboard ke Studio Lab, Anda secara otomatis mengaktifkan`studiolab`lingkungan conda. Berikut ini adalah contoh lingkungan yang terdaftar setelah Anda onboard.

```
# conda environments:
#
default                  /home/studio-lab-user/.conda/envs/default
studiolab             *  /home/studio-lab-user/.conda/envs/studiolab
studiolab-safemode       /opt/amazon/sagemaker/safemode-home/.conda/envs/studiolab-
safemode
base                     /opt/conda
sagemaker-distribution     /opt/conda/envs/sagemaker-distribution
```

The*menandai lingkungan yang diaktifkan.

Buat, aktifkan, dan gunakan lingkungan conda baru

Jika Anda ingin mempertahankan beberapa lingkungan untuk kasus penggunaan yang berbeda, Anda dapat membuat lingkungan conda baru dalam proyek Anda. Bagian berikut menunjukkan cara membuat dan mengaktifkan lingkungan conda baru. Untuk notebook Jupyter yang menunjukkan cara membuat lingkungan kustom, lihat Menyiapkan Lingkungan Kustom di SageMaker Laboratorium Studio.

> **Note**
>
> Mempertahankan beberapa lingkungan akan diperhitungkan terhadap memori Studio Lab Anda yang tersedia.

Buat lingkungan conda

Untuk membuat lingkungan conda, jalankan perintah conda berikut dari terminal Anda. Contoh ini menciptakan lingkungan baru dengan Python 3.9.

```
conda create --name <ENVIRONMENT_NAME> python=3.9
```

Setelah lingkungan conda dibuat, Anda dapat melihat lingkungan di lingkungan Anda. Untuk informasi selengkapnya tentang cara melihat daftar lingkungan Anda, lihatLihat lingkungan.

Aktifkan lingkungan conda

Untuk mengaktifkan lingkungan conda apa pun, jalankan perintah berikut di terminal.

```
conda activate <ENVIRONMENT_NAME>
```

Saat Anda menjalankan perintah ini, paket apa pun yang diinstal menggunakan conda atau pip diinstal di lingkungan. Untuk informasi selengkapnya tentang menginstal paket, lihatSesuaikan lingkungan Anda.

Gunakan lingkungan conda

Untuk menggunakan lingkungan conda baru Anda dengan notebook, pastikan`ipykernel`paket diinstal di lingkungan.

```
conda install ipykernel
```

Setelah`ipykernel`paket diinstal di lingkungan, Anda dapat memilih lingkungan sebagai kernel untuk notebook Anda.

Anda mungkin perlu memulai ulang JupyterLab untuk melihat lingkungan yang tersedia sebagai kernel. Ini bisa dilakukan dengan memilihAmazon SageMaker Laboratorium Studiodi menu atas Studio Lab dan memilihMulai ulang JupyterLab....

Saat Anda membuat buku catatan baru dari Studio Lab Launcher, Anda akan memiliki opsi untuk memilih kernel di bawahBuku catatan. Untuk gambaran umum tentang UI Studio Lab, lihatIkhtisar UI Amazon SageMaker Studio Lab.

Saat notebook Jupyter terbuka, Anda dapat memilih kernel dengan memilihKerneldari menu atas dan pilihUbah Kernel....

Menggunakan lingkungan Studio Lab sampel

Studio Lab menyediakan contoh lingkungan kustom melaluiSageMaker Contoh Studio Labrepositori. Berikut ini menunjukkan cara mengkloning dan membangun lingkungan ini.

1. Kloning SageMaker Contoh Studio Lab GitHub repositori dengan mengikuti instruksi diGunakan GitHub sumber daya.
2. Di Studio Lab pilihPeramban Berkasikon
   ( )
   di menu sebelah kiri, sehinggaPeramban Berkaspanel ditampilkan di sebelah kiri.
3. Navigasikan ke`studio-lab-examples/custom-environments`direktori di File Browser.
4. Buka direktori lingkungan yang ingin Anda bangun.
5. Klik kanan`.yml`berkas di folder, lalu pilihBangun Lingkungan Conda.
6. Anda sekarang dapat menggunakan lingkungan sebagai kernel setelah lingkungan conda Anda selesai dibangun. Untuk petunjuk tentang cara menggunakan lingkungan yang ada sebagai kernel, lihatBuat, aktifkan, dan gunakan lingkungan conda baru

Sesuaikan lingkungan Anda

Anda dapat menyesuaikan lingkungan Anda dengan menginstal dan menghapus ekstensi dan paket sesuai kebutuhan. Studio Lab dilengkapi dengan lingkungan dengan paket yang sudah diinstal sebelumnya dan menggunakan lingkungan yang ada dapat menghemat waktu dan memori Anda, karena paket pra-instal tidak dihitung terhadap memori Studio Lab Anda yang tersedia. Untuk informasi selengkapnya tentang lingkungan Studio Lab yang sudah diinstal sebelumnya, lihatLingkungan pra-instal Studio Lab.

Setiap ekstensi dan paket yang diinstal pada`default`lingkungan akan bertahan di proyek Anda, jadi Anda tidak perlu menginstal paket Anda untuk setiap sesi runtime proyek. Namun, ekstensi dan paket diinstal pada`sagemaker-distribution`lingkungan tidak akan bertahan, jadi Anda perlu menginstal paket baru selama sesi berikutnya. Dengan demikian, sangat disarankan untuk

menginstal paket dalam notebook Anda untuk memastikan bahwa paket diinstal di lingkungan yang dimaksud.

Untuk melihat lingkungan Anda, jalankan perintah`conda env list`.

Untuk mengaktifkan lingkungan Anda, jalankan perintah`conda activate <ENVIRONMENT_NAME>`.

Untuk melihat paket di lingkungan, jalankan perintah`conda list`.

Instal paket

Sangat disarankan untuk menginstal paket Anda dalam notebook Jupyter Anda untuk memastikan bahwa paket Anda diinstal di lingkungan yang dimaksud. Untuk menginstal paket tambahan ke lingkungan Anda dari notebook Jupyter, jalankan salah satu perintah berikut di sel dalam notebook Jupyter Anda. Perintah ini menginstal paket di lingkungan yang saat ini diaktifkan.

- `%conda install <PACKAGE>`
- `%pip install <PACKAGE>`

Kami tidak merekomendasikan menggunakan`!pip`atau`!conda`perintah karena mereka dapat berperilaku dengan cara yang tidak terduga ketika Anda memiliki banyak lingkungan.

Setelah Anda menginstal paket baru ke lingkungan Anda, Anda mungkin perlu me-restart kernel untuk memastikan bahwa paket bekerja di notebook Anda. Ini bisa dilakukan dengan memilihAmazon SageMaker Laboratorium Studiodi menu atas Studio Lab dan memilihMulai ulang JupyterLab....

Hapus paket

Untuk menghapus paket, jalankan perintah

```
%conda remove <PACKAGE_NAME>
```

Perintah ini juga akan menghapus paket apa pun yang bergantung pada`<PACKAGE_NAME>`, kecuali pengganti dapat ditemukan tanpa ketergantungan itu.

Untuk menghapus semua paket di lingkungan, jalankan perintah

```
conda deactivate
&& conda env remove --name
<ENVIRONMENT_NAME>
```

Segarkan Lab Studio

Untuk me-refresh Studio Lab, hapus semua lingkungan dan file Anda.

1. Daftar semua lingkungan conda.

```
conda env list
```

2. Aktifkan lingkungan dasar.

```
conda activate base
```

3. Hapus setiap lingkungan dalam daftar lingkungan conda, selain basis.

```
conda remove --name <ENVIRONMENT_NAME> --all
```

4. Hapus semua file di Studio Lab Anda.

```
rm -rf *.*
```

Menggunakan sumber daya eksternal di Amazon SageMaker Studio Lab

Dengan Amazon SageMaker Studio Lab, Anda dapat mengintegrasikan sumber daya eksternal, seperti notebook dan data Jupyter, dari repositori Git dan Amazon S3. Anda juga dapat menambahkan tombol Open in Studio Lab ke GitHub repo dan notebook Anda. Tombol ini memungkinkan Anda mengkloning notebook Anda langsung dari Studio Lab.

Topik berikut menunjukkan cara mengintegrasikan sumber daya eksternal.

Topik

Gunakan GitHub sumber daya

Studio Lab menawarkan integrasi denganGitHub. Dengan integrasi ini, Anda dapat mengkloning notebook dan repositori langsung ke proyek Studio Lab Anda.

Topik berikut memberikan informasi tentang cara menggunakan GitHub sumber daya dengan Studio Lab.

Notebook sampel Studio Lab

Untuk memulai dengan repositori contoh notebook yang disesuaikan untuk Studio Lab, lihat Notebook Sampel Studio Lab.

Repositori ini menyediakan notebook untuk kasus penggunaan berikut dan lainnya.

- Visi komputer
- Menghubungkan ke AWS
- Membuat lingkungan khusus
- Analisis data geospasial
- Pemrosesan bahasa alami
- Menggunakan R

Kloning GitHub repo

Untuk mengkloning GitHub repo ke proyek Studio Lab Anda, ikuti langkah-langkah ini.

1. Mulai runtime proyek Studio Lab Anda. Untuk informasi selengkapnya tentang meluncurkan runtime proyek Studio Lab, lihatMulai Runtime Proyek Anda.
2. Di Studio Lab, pilih ikon File Browser
   ( )
   di menu sebelah kiri, sehingga panel File Browser ditampilkan di sebelah kiri.
3. Arahkan ke direktori pengguna Anda dengan memilih ikon file di bawah bilah pencarian file.
4. Pilih ikon Git
   ( )
   dari menu kiri untuk membuka menu dropdown baru.
5. Pilih Clone Repository.
6. Tempel URL repositori di bawah URL repositori Git (.git).
7. Pilih Clone.

Clone notebook individu dari GitHub

Untuk membuka notebook di Studio Lab, Anda harus memiliki akses ke repo tempat notebook berada. Contoh berikut menjelaskan perilaku terkait izin Studio Lab dalam berbagai situasi.

- Jika repo bersifat publik, Anda dapat secara otomatis mengkloning notebook ke proyek Anda dari halaman pratinjau Studio Lab.
- Jika repo bersifat pribadi, Anda akan diminta untuk masuk GitHub dari halaman pratinjau Studio Lab. Jika Anda memiliki akses ke repo pribadi, Anda dapat mengkloning notebook ke dalam proyek Anda.
- Jika Anda tidak memiliki akses ke repo pribadi, Anda tidak dapat mengkloning notebook dari halaman pratinjau Studio Lab.

Bagian berikut menunjukkan dua opsi bagi Anda untuk menyalin GitHub buku catatan di proyek Studio Lab Anda. Pilihan ini tergantung pada apakah notebook memiliki tombol Open in Studio Lab.

Opsi 1: Salin notebook dengan tombol Open in Studio Lab

Prosedur berikut menunjukkan cara menyalin notebook yang memiliki tombol Open in Studio Lab. Jika Anda ingin menambahkan tombol ini ke notebook Anda, lihatTambahkan tombol Open in Studio Lab ke notebook.

1. Masuk ke Studio Lab mengikuti langkah-langkah masukMasuk ke Studio Lab.
2. Di tab browser baru, navigasikan ke GitHub notebook yang ingin Anda kloning.
3. Di buku catatan, pilih tombol Buka di Studio Lab untuk membuka halaman baru di Studio Lab dengan pratinjau notebook.
4. Jika runtime proyek Anda belum berjalan, mulailah dengan memilih tombol Mulai runtime di bagian atas halaman pratinjau. Tunggu runtime dimulai sebelum melanjutkan ke langkah berikutnya.
5. Setelah runtime proyek Anda dimulai, pilih Salin ke proyek untuk membuka runtime proyek Anda di tab browser baru.
6. Di Copy dariGitHub? kotak dialog, pilih Salin notebook saja. Ini menyalin file notebook ke proyek Anda.

Opsi 2: Kloning GitHub notebook apa pun

Prosedur berikut menunjukkan cara menyalin notebook apa punGitHub.

1. Arahkan ke notebook diGitHub.
2. Di bilah alamat browser, ubah URL notebook, sebagai berikut.

```
# Original URL
https://github.com/<PATH_TO_NOTEBOOK>

# Modified URL
https://studiolab.sagemaker.aws/import/github/<PATH_TO_NOTEBOOK>
```

3. Arahkan ke URL yang dimodifikasi. Ini akan membuka pratinjau notebook di Studio Lab.
4. Jika runtime proyek Anda belum berjalan, mulailah dengan memilih tombol Mulai runtime di bagian atas halaman pratinjau. Tunggu runtime dimulai sebelum melanjutkan ke langkah berikutnya.
5. Setelah runtime proyek Anda dimulai, pilih Salin ke proyek untuk membuka runtime proyek Anda di tab browser baru.
6. Di Copy dariGitHub? kotak dialog, pilih Salin notebook hanya untuk menyalin file notebook ke proyek Anda.

Tambahkan tombol Open in Studio Lab ke notebook

Ketika Anda menambahkan tombol Open in Studio Lab ke notebook Anda, orang lain dapat mengkloning notebook atau repositori Anda langsung ke proyek Studio Lab mereka. Jika Anda membagikan buku catatan Anda dalam GitHub repositori publik, konten Anda akan dapat dibaca secara publik. Jangan berbagi konten pribadi, seperti kunci AWS akses atau AWS Identity and Access Management kredensi, di buku catatan Anda.

Untuk menambahkan tombol Open in Studio Lab fungsional ke notebook atau repositori Jupyter Anda, tambahkan penurunan harga berikut ke bagian atas notebook atau repositori Anda.

```
[![Open In SageMaker Studio Lab](https://studiolab.sagemaker.aws/studiolab.svg)]
(https://studiolab.sagemaker.aws/import/github/<PATH_TO_YOUR_NOTEBOOK_ON_GITHUB>)
```

Mengimpor file dari komputer Anda

Langkah-langkah berikut menunjukkan cara mengimpor file dari komputer Anda ke proyek Studio Lab Anda.

1. Buka runtime proyek Studio Lab.

2. Buka panel File Browser.
3. Di bilah tindakan panel File Browser, pilih tombol Unggah File.
4. Pilih file yang ingin Anda unggah dari komputer lokal Anda.
5. Pilih Buka.

Atau, Anda dapat menyeret dan menjatuhkan file dari komputer Anda ke panel File Browser.

Connect ke Amazon S3

AWS CLIMemungkinkan AWS integrasi dalam proyek Studio Lab Anda. Dengan integrasi ini, Anda dapat menarik sumber daya dari Amazon S3 untuk digunakan dengan notebook Jupyter Anda.

Untuk menggunakan AWS CLI Studio Lab, lakukan langkah-langkah berikut. Untuk buku catatan yang menguraikan integrasi ini, lihat Menggunakan Studio Lab dengan AWS Sumber Daya.

1. Instal langkah-langkah AWS CLI berikut dalam Menginstal atau memperbarui versi terbaru dari AWS CLI.
2. Konfigurasikan AWS kredensi Anda dengan mengikuti langkah-langkah dalam Pengaturan cepat. Peran untuk AWS akun Anda harus memiliki izin untuk mengakses bucket Amazon S3 tempat Anda menyalin data.
3. Dari notebook Jupyter Anda, kloning sumber daya dari bucket Amazon S3, sesuai kebutuhan. Perintah berikut menunjukkan cara mengkloning semua sumber daya dari jalur Amazon S3 ke proyek Anda. Untuk informasi selengkapnya, lihat AWS CLI Referensi Perintah.

```
!aws s3 cp s3://<BUCKET_NAME>/<PATH_TO_RESOURCES>/ <PROJECT_DESTINATION_PATH>/ --
recursive
```

Dapatkan perbedaan notebook

Anda dapat menampilkan perbedaan antara notebook saat ini dan pos pemeriksaan terakhir, atau commit Git terakhir, menggunakan UI proyek Amazon SageMaker Studio Lab.

Topik

- Dapatkan perbedaan antara pos pemeriksaan terakhir
- Dapatkan perbedaan antara komit terakhir

Dapatkan perbedaan antara pos pemeriksaan terakhir

Saat Anda membuat notebook, file pos pemeriksaan tersembunyi yang cocok dengan notebook akan dibuat. Anda dapat melihat perubahan antara notebook dan file pos pemeriksaan, atau mengembalikan notebook untuk mencocokkan file pos pemeriksaan.

Untuk menyimpan notebook Studio Lab dan memperbarui file pos pemeriksaan agar cocok: Pilih notebook Simpan dan buat ikon pos pemeriksaan (). Ini terletak di sisi kiri menu Studio Lab. Pintasan keyboard untuk Simpan notebook dan buat pos pemeriksaan adalah`Ctrl + s`.

Untuk melihat perubahan antara notebook Studio Lab dan file pos pemeriksaan: Pilih ikon diff Checkpoint (), yang terletak di tengah menu Studio Lab.

Untuk mengembalikan notebook Studio Lab ke file checkpoint: Pada menu Studio Lab utama, pilih File, dan kemudian Kembalikan Notebook ke Checkpoint.

Dapatkan perbedaan antara komit terakhir

Jika notebook dibuka dari repositori Git, Anda dapat melihat perbedaan antara notebook dan komit Git terakhir.

Untuk melihat perubahan pada notebook dari commit Git terakhir: Pilih ikon diff Git () di tengah menu notebook.

Ekspor lingkungan Amazon SageMaker Studio Lab ke Amazon SageMaker Studio Classic

Amazon SageMaker Studio Classic menawarkan banyak fitur untuk pembelajaran mesin dan alur kerja pembelajaran mendalam yang tidak tersedia di Amazon SageMaker Studio Lab. Halaman ini menunjukkan cara memigrasikan lingkungan Studio Lab ke Studio Classic untuk memanfaatkan lebih banyak kapasitas komputasi, penyimpanan, dan fitur. Namun, Anda mungkin ingin membiasakan diri dengan kontainer bawaan Studio Classic, yang dioptimalkan untuk pipeline MLOP penuh. Lihat informasi yang lebih lengkap di Lab SageMaker Studio Amazon

Untuk memigrasikan lingkungan Studio Lab ke Studio Classic, Anda harus terlebih dahulu melakukan onboard ke Studio Classic dengan mengikuti langkah-langkahnya. Ikhtisar SageMaker Domain Amazon

Topik

- Langkah 1: Ekspor lingkungan studio lab conda Anda
- Langkah 2: Simpan artefak Studio Lab Anda
- Langkah 3: Impor artefak Studio Lab Anda ke Studio Classic
- Langkah 4: Instal lingkungan conda Studio Lab Anda di Studio Classic

Langkah 1: Ekspor lingkungan studio lab conda Anda

Anda dapat mengekspor lingkungan conda dan menambahkan pustaka atau paket ke lingkungan dengan mengikuti langkah-langkah di. Kelola lingkungan Anda Contoh berikut menunjukkan penggunaan `default` lingkungan yang akan diekspor ke Studio Classic.

1. Buka terminal Studio Lab dengan membuka panel File Browser
   (),
   pilih tanda plus (+) pada menu di bagian atas browser file untuk membuka Launcher, lalu pilih Terminal. Dari terminal Studio Lab, daftarkan lingkungan conda dengan menjalankan yang berikut ini.

   ```
   conda env list
   ```

   Perintah ini menampilkan daftar lingkungan conda dan lokasinya di sistem file. Saat Anda onboard ke Studio Lab, Anda secara otomatis mengaktifkan lingkungan `studiolab` conda.

   ```
   # conda environments: #
              default                  /home/studio-lab-user/.conda/envs/default
              studiolab             *  /home/studio-lab-user/.conda/envs/studiolab
              studiolab-safemode       /opt/amazon/sagemaker/safemode-home/.conda/
   envs/studiolab-safemode
              base                     /opt/conda
   ```

   Kami menyarankan Anda untuk tidak mengeksporstudiolab,`studiolab-safemode`, dan `base` lingkungan. Lingkungan ini tidak dapat digunakan di Studio Classic karena alasan berikut:

- `studiolab`: Ini mengatur JupyterLab lingkungan untuk Studio Lab. Studio Lab menjalankan versi utama yang berbeda JupyterLab dari Studio Classic, sehingga tidak dapat digunakan di Studio Classic.
- `studiolab-safemode`: Ini juga mengatur JupyterLab lingkungan untuk Studio Lab. Studio Lab menjalankan versi utama yang berbeda JupyterLab dari Studio Classic, sehingga tidak dapat digunakan di Studio Classic.
- `base`: Lingkungan ini dilengkapi dengan conda secara default. `base`Lingkungan di Studio Lab dan `base` lingkungan di Studio Classic memiliki versi yang tidak kompatibel dari banyak paket.

2. Untuk lingkungan conda yang ingin Anda migrasikan ke Studio Classic, aktifkan dulu lingkungan conda. `default`Lingkungan kemudian diubah ketika pustaka baru diinstal atau dihapus darinya. Untuk mendapatkan keadaan lingkungan yang tepat, ekspor ke file YAMAL menggunakan baris perintah. Baris perintah berikut mengekspor lingkungan default ke dalam file YAMM, membuat file bernama`myenv.yml`.

```
conda activate default
conda env export > ~/myenv.yml
```

Langkah 2: Simpan artefak Studio Lab Anda

Sekarang setelah Anda menyimpan lingkungan Anda ke file YAMAL, Anda dapat memindahkan file lingkungan ke platform apa pun.

Save to a local machine using Studio Lab GUI

> **Note**
>
> Mengunduh direktori dari GUI Studio Lab dengan mengklik kanan pada direktori saat ini tidak tersedia. Jika Anda ingin mengekspor direktori, ikuti langkah-langkah menggunakan tab Save to Git repository.

Salah satu pilihan adalah untuk menyelamatkan lingkungan ke mesin lokal Anda. Untuk melakukannya, gunakan prosedur berikut.

1. Di Studio Lab, pilih ikon File Browser
   ()
   di menu sebelah kiri, sehingga panel File Browser ditampilkan di sebelah kiri.
2. Arahkan ke direktori pengguna Anda dengan memilih ikon file di bawah bilah pencarian file.
3. Pilih (klik kanan) `myenv.yml` file dan kemudian pilih Unduh. Anda dapat mengulangi proses ini untuk file lain yang ingin Anda impor ke Studio Classic.

Save to a Git repository

Pilihan lain adalah menyimpan lingkungan Anda ke repositori Git. Opsi ini digunakan GitHub sebagai contoh. Langkah-langkah ini memerlukan GitHub akun dan repositori. Untuk informasi selengkapnya, kunjungi GitHub. Prosedur berikut menunjukkan cara menyinkronkan konten Anda dengan GitHub menggunakan terminal Studio Lab.

1. Dari terminal Studio Lab, arahkan ke direktori pengguna Anda dan buat direktori baru untuk berisi file yang ingin Anda ekspor.

   ```
   cd ~
   mkdir <NEW_DIRECTORY_NAME>
   ```

2. Setelah Anda membuat direktori baru, salin file atau direktori apa pun yang ingin Anda ekspor*<NEW_DIRECTORY_NAME>*.

   Salin file menggunakan format kode berikut:

   ```
   cp <FILE_NAME> <NEW_DIRECTORY_NAME>
   ```

   Misalnya, ganti *<FILE_NAME>* dengan`myenv.yml`.

   Salin direktori apa pun menggunakan format kode berikut:

   ```
   cp -r <DIRECTORY_NAME> <NEW_DIRECTORY_NAME>
   ```

   Misalnya, ganti *<DIRECTORY_NAME>* dengan nama direktori apa pun di direktori pengguna Anda.

3. Arahkan ke direktori baru dan inisialisasi direktori sebagai repositori Git menggunakan perintah berikut. Untuk informasi selengkapnya, lihat dokumentasi git-init.

```
cd <NEW_DIRECTORY_NAME>
git init
```

4. Menggunakan Git, tambahkan semua file yang relevan dan kemudian komit perubahan Anda.

```
git add .
git commit -m "<COMMIT_MESSAGE>"
```

Misalnya, ganti *<COMMIT_MESSAGE>* dengan`Add Amazon SageMaker Studio Lab artifacts to GitHub repository to migrate to Amazon SageMaker Studio Classic .`

5. Dorong komit ke repositori jarak jauh Anda. Repositori ini memiliki format di `https://github.com/`*<GITHUB_USERNAME>*`/` *<REPOSITORY_NAME>*`.git` mana nama GitHub pengguna Anda dan *<GITHUB_USERNAME>* adalah nama repositori jarak jauh Anda. *<REPOSITORY_NAME>* Buat cabang *<BRANCH_NAME>* untuk mendorong konten ke GitHub repositori.

```
git branch -M <BRANCH_NAME>
git remote add origin https://github.com/<GITHUB_USERNAME>/<REPOSITORY_NAME>.git
git push -u origin <BRANCH_NAME>
```

Langkah 3: Impor artefak Studio Lab Anda ke Studio Classic

Prosedur berikut menunjukkan cara mengimpor artefak ke Studio Classic. Petunjuk tentang penggunaan Feature Store melalui konsol bergantung pada apakah Anda telah mengaktifkan Studio atau Studio Classic sebagai pengalaman default Anda. Untuk informasi tentang mengakses Studio Classic melalui konsol, lihatLuncurkan Studio Classic jika Studio adalah pengalaman default Anda.

Dari Studio Classic, Anda dapat mengimpor file dari mesin lokal Anda atau dari repositori Git. Anda dapat melakukan ini menggunakan GUI atau terminal Studio Classic. Prosedur berikut menggunakan contoh dariLangkah 2: Simpan artefak Studio Lab Anda.

Import using the Studio Classic GUI

Jika Anda menyimpan file ke mesin lokal Anda, Anda dapat mengimpor file ke Studio Classic menggunakan langkah-langkah berikut.

1. Buka panel File Browser
   ()
   di kiri atas Studio Classic.
2. Pilih ikon Upload Files
   ()
   pada menu di bagian atas panel File Browser.
3. Arahkan ke file yang ingin Anda impor, lalu pilih Buka.

> **Note**
>
> Untuk mengimpor direktori ke Studio Classic, pertama-tama kompres direktori di mesin lokal Anda ke file. Di Mac, klik kanan direktori dan pilih Kompres ""<DIRECTORY_NAME>. Di Windows, klik kanan direktori dan pilih Kirim ke, lalu pilih Folder terkompresi (zip). Setelah direktori dikompresi, impor file terkompresi menggunakan langkah-langkah sebelumnya. Buka zip file terkompresi dengan menavigasi ke terminal Studio Classic dan menjalankan perintah. `<DIRECTORY_NAME>.zip`

Import using a Git repository

Contoh ini menyediakan dua opsi untuk cara mengkloning GitHub repositori ke Studio Classic. Anda dapat menggunakan GUI Studio Classic dengan memilih tab Git
()
di sisi kiri Studio Classic. Pilih Clone a Repository, lalu tempel URL GitHub repositori Anda. Langkah 2: Simpan artefak Studio Lab Anda Pilihan lain adalah menggunakan terminal Studio Classic dengan menggunakan prosedur berikut.

1. Buka Studio Classic Launcher. Untuk informasi selengkapnya tentang membuka Peluncur, lihat Amazon SageMaker Studio Classic Launcher.
2. Di Peluncur, di bagian Notebook dan sumber daya komputasi, pilih Ubah lingkungan.
3. Di Studio Classic, buka Launcher. Untuk membuka Launcher, pilih Amazon SageMaker Studio Classic di pojok kiri atas Studio Classic.

   Untuk mempelajari semua cara yang tersedia untuk membuka Peluncur, lihatGunakan Amazon SageMaker Studio Classic Launcher.

4. Dalam dialog Ubah lingkungan, gunakan daftar dropdown Gambar untuk memilih gambar Ilmu Data dan pilih Pilih. Gambar ini dilengkapi dengan conda pra-instal.
5. Di Studio Classic Launcher, pilih Open image terminal.
6. Dari terminal gambar, jalankan perintah berikut untuk mengkloning repositori Anda. Perintah ini membuat direktori yang dinamai <REPOSITORY_NAME> dalam instance Studio Classic Anda dan mengkloning artefak Anda di repositori itu.

```
git clone https://github.com/<GITHUB_USERNAME>/<REPOSITORY_NAME>.git
```

Langkah 4: Instal lingkungan conda Studio Lab Anda di Studio Classic

Sekarang Anda dapat membuat ulang lingkungan conda Anda dengan menggunakan file YAMAL Anda di instans Studio Classic Anda. Buka Studio Classic Launcher. Untuk informasi selengkapnya tentang membuka Peluncur, lihat Amazon SageMaker Studio Classic Launcher. Dari Peluncur, pilih Buka terminal gambar. Di terminal navigasikan ke direktori yang berisi file YAMAL, lalu jalankan perintah berikut.

```
conda env create --file <ENVIRONMENT_NAME>.yml
conda activate <ENVIRONMENT_NAME>
```

Setelah perintah ini selesai, Anda dapat memilih lingkungan Anda sebagai kernel untuk instance notebook Studio Classic Anda. Untuk melihat lingkungan yang tersedia, jalankan`conda env list`. Untuk mengaktifkan lingkungan Anda, jalankan`conda activate <ENVIRONMENT_NAME>`.

Matikan sumber daya

Dalam panduan ini, Anda akan belajar cara mematikan sumber daya individu, termasuk notebook, terminal, dan kernel. Anda juga dapat mematikan semua sumber daya di salah satu kategori ini secara bersamaan.

Topik

- Matikan notebook terbuka
- Matikan sumber daya

Matikan notebook terbuka

Anda dapat mematikan notebook terbuka dari menu File Amazon SageMaker Studio Lab atau dari panel Running Terminals and Kernels.

> **Note**
>
> Saat Anda mematikan buku catatan, informasi apa pun yang belum disimpan di buku catatan akan hilang. Notebook tidak dihapus.

Untuk mematikan notebook yang terbuka dari menu File

1. Simpan isi notebook dengan memilih
   
   ikon, yang terletak di menu notebook.
2. Pilih File lalu Tutup dan Shutdown Notebook.
3. Pilih OKE.

Matikan sumber daya

Di sidebar kiri Studio Lab, Anda akan menemukan panel dan

ikon Running Terminals dan Kernels. Panel Running Terminal dan Kernels memiliki tiga bagian. Setiap bagian mencantumkan semua sumber daya dari jenis itu. Anda dapat mematikan setiap sumber daya secara individual, atau mematikan semua sumber daya di bagian secara bersamaan.

Saat Anda mematikan semua sumber daya di bagian, hal berikut terjadi:

- KERNELS - Semua kernel, notebook, dan konsol ditutup.
- TERMINAL - Semua terminal dimatikan.

Untuk mematikan sumber daya

1. Di bilah sisi kiri, pilih ikon Running Terminal dan Kernels
   ().

2. Lakukan salah satu dari langkah berikut:

   - Untuk mematikan sumber daya tertentu: Pilih ikon SHUT DOWN pada baris yang sama dengan sumber daya.
   - Untuk mematikan semua sumber daya di bagian: Pilih Shut Down All, yang terletak di sebelah kanan label bagian. Setelah kotak dialog konfirmasi muncul, pilih Matikan semua untuk melanjutkan.

Memecahkan masalah

Panduan ini menunjukkan kesalahan umum yang mungkin terjadi saat menggunakan Amazon SageMaker Studio Lab. Setiap kesalahan berisi deskripsi, serta solusi untuk kesalahan tersebut.

> **Note**
>
> Anda tidak dapat membagikan kata sandi Anda dengan beberapa pengguna atau menggunakan Studio Lab untuk menambang cryptocurrency. Kami tidak merekomendasikan penggunaan Studio Lab untuk tugas produksi karena batas waktu proses.

Tidak dapat mengakses akun

Jika Anda tidak dapat mengakses akun Anda, verifikasi bahwa Anda menggunakan email dan kata sandi yang benar. Jika Anda lupa kata sandi, gunakan langkah-langkah berikut untuk mengatur ulang kata sandi Anda. Jika Anda masih tidak dapat mengakses akun Anda, Anda harus meminta dan mendaftar untuk akun baru menggunakan instruksi diNaik ke Amazon SageMaker Studio Lab.

Lupa kata sandi

Jika Anda lupa kata sandi, Anda harus mengatur ulang menggunakan langkah-langkah berikut.

1. Arahkan ke halaman landing Studio Lab.
2. Pilih Masuk.
3. Pilih Lupa kata sandi? untuk membuka halaman baru.
4. Masukkan alamat email yang Anda gunakan untuk mendaftar akun.
5. Pilih Kirim tautan reset untuk mengirim email dengan tautan pengaturan ulang kata sandi.
6. Dari email pengaturan ulang kata sandi, pilih Setel ulang kata sandi Anda.

7. Masukkan kata sandi baru Anda.
8. Pilih Kirim.

Tidak dapat meluncurkan runtime proyek

Jika runtime proyek Studio Lab tidak diluncurkan, coba luncurkan lagi. Jika ini tidak berhasil, alihkan jenis instance dari CPU ke GPU (atau sebaliknya). Untuk informasi selengkapnya, lihat Ubah Tipe komputasi Anda.

Runtime berhenti berjalan secara tak terduga

Jika ada masalah dengan lingkungan yang digunakan untuk menjalankan JupyterLab, maka Studio Lab akan secara otomatis membuat ulang lingkungan. Studio Lab tidak mendukung aktivasi manual proses ini.

Versi yang bertentangan

Karena Anda dapat menambahkan paket dan memodifikasi lingkungan Anda sesuai kebutuhan, Anda mungkin mengalami konflik antar paket di lingkungan Anda. Jika ada konflik antar paket di lingkungan Anda, Anda harus menghapus paket yang bertentangan.

Membangun lingkungan gagal

Saat Anda membangun lingkungan dari file YAMAL, konflik versi paket atau masalah file dapat menyebabkan build gagal. Untuk mengatasi ini, hapus lingkungan dengan menjalankan perintah berikut. Lakukan ini sebelum mencoba membangunnya lagi.

```
conda remove --name <YOUR_ENVIRONMENT> --all
```

Pesan galat tentang mengizinkan untuk mengunduh skrip dari domain*.awswaf.com

Studio Classic menggunakan layanan firewall aplikasi web AWS WAF untuk melindungi sumber daya Anda, yang menggunakannya JavaScript. Jika Anda menggunakan plugin keamanan browser yang JavaScript mencegah pengunduhan, kesalahan ini mungkin muncul. Untuk menggunakan Studio Classic, izinkan JavaScript unduhan dari*.awswaf.com sebagai domain tepercaya. Untuk informasi lebih lanjut tentangAWS WAF, lihat AWS WAFdariAWS WAF,AWS Firewall Manager, danAWS Shield Advanced. Panduan Pengembang.

Ruang disk penuh

Jika Anda mengalami pemberitahuan yang menyebutkan bahwa ruang disk Anda penuh atau File Load Error <FILE_NAME>saat mencoba membuka file, Anda dapat menghapus file, direktori, pustaka, atau lingkungan untuk menambah ruang. Untuk informasi selengkapnya tentang mengelola pustaka dan lingkungan Anda, lihatKelola lingkungan Anda.

Runtime proyek dalam notifikasi mode aman

Jika Anda mengalami pemberitahuan bahwa runtime Project dalam mode aman, Anda harus mengosongkan beberapa ruang disk untuk melanjutkan menggunakan runtime proyek Studio Lab. Ikuti instruksi di item pemecahan masalah sebelumnya, Ruang disk penuh. Setelah setidaknya 500 MB ruang telah dihapus, Anda dapat memulai ulang runtime proyek untuk menggunakan Studio Lab. Ini dapat dilakukan dengan memilih Amazon SageMaker Studio Lab di menu atas Studio Lab dan memilih Restart JupyterLab... .

git Tidak dapat mengimpor **cv2**

Jika Anda mengalami kesalahan saat mengimpor `cv2` setelah menginstalo`pencv-python`, Anda harus menghapus `opencv-python` dan menginstal `opencv-python-headless` sebagai berikut.

```
%pip uninstall opencv-python --yes
%pip install opencv-python-headless
```

Anda kemudian dapat mengimpor `cv2` seperti yang diharapkan.

Studio Lab menjadi tidak responsif saat membuka file besar

Studio Lab IDE mungkin gagal dirender saat file besar dibuka, sehingga akses ke sumber daya Studio Lab diblokir. Untuk mengatasinya, setel ulang ruang kerja Studio Lab menggunakan prosedur berikut.

1. Setelah Anda membuka IDE, salin URL di bilah alamat browser Anda. URL ini harus dalam `https://xxxxxx.studio.us-east-2.sagemaker.aws/studiolab/default/jupyter/lab` format. Tutup tab.
2. Di tab baru, tempel URL dan hapus apa pun setelahnya`https://xxxxxx.studio.us-east-2.sagemaker.aws/studiolab/default/jupyter/lab`.
3. Tambahkan `?reset` ke akhir URL, jadi itu dalam `https://xxxxxx.studio.us-east-2.sagemaker.aws/studiolab/default/jupyter/lab?reset` format.
4. Arahkan ke URL yang diperbarui. Ini mengatur ulang status UI yang disimpan dan membuat Studio Lab IDE responsif.

SageMaker Kanvas Amazon

Amazon SageMaker Canvas memberi Anda kemampuan untuk menggunakan pembelajaran mesin untuk menghasilkan prediksi tanpa perlu menulis kode apa pun. Berikut ini adalah beberapa kasus penggunaan di mana Anda dapat menggunakan SageMaker Canvas:

- Memprediksi churn pelanggan
- Rencanakan inventaris secara efisien
- Optimalkan harga dan pendapatan
- Tingkatkan pengiriman tepat waktu
- Klasifikasi teks atau gambar berdasarkan kategori kustom
- Identifikasi objek dan teks dalam gambar
- Ekstrak informasi dari dokumen

Dengan Canvas, Anda dapat mengobrol dengan model bahasa besar (LLM) populer, mengakses eady-to-use model R, atau membuat model khusus yang dilatih pada data Anda.

Obrolan kanvas adalah fungsi yang memanfaatkan sumber terbuka dan Amazon LLM untuk membantu Anda meningkatkan produktivitas. Anda dapat meminta model untuk mendapatkan bantuan dengan tugas-tugas seperti membuat konten, meringkas atau mengkategorikan dokumen, dan menjawab pertanyaan. Untuk mempelajari selengkapnya, lihat Gunakan AI generatif dengan model foundation.

eady-to-use Model R di Canvas dapat mengekstrak wawasan dari data Anda untuk berbagai kasus penggunaan. Anda tidak perlu membuat model untuk menggunakan eady-to-use model R karena didukung oleh layanan Amazon AI, termasuk Amazon Rekognition, AmazonTextract, dan Amazon Comprehend. Anda hanya perlu mengimpor data Anda dan mulai menggunakan solusi untuk menghasilkan prediksi.

Jika Anda menginginkan model yang disesuaikan dengan kasus penggunaan Anda dan dilatih dengan data Anda, Anda dapat membuat model. Anda bisa mendapatkan prediksi yang disesuaikan dengan data Anda dengan melakukan hal berikut:

1. Impor data Anda dari satu atau beberapa sumber data.
2. Bangun model prediktif.
3. Mengevaluasi kinerja model.

4. Hasilkan prediksi dengan model.

Canvas mendukung jenis model kustom berikut:

- Prediksi numerik (juga dikenal sebagai regresi)
- Prediksi kategoris untuk 2 dan 3+ kategori (juga dikenal sebagai klasifikasi biner dan multi-kelas)
- Peramalan deret waktu
- Prediksi gambar label tunggal (juga dikenal sebagai klasifikasi gambar)
- Prediksi teks multi-kategori (juga dikenal sebagai klasifikasi teks multi-kelas)

Anda juga dapat membawa model Anda sendiri ke Canvas dari Amazon SageMaker Studio Classic.

Untuk mempelajari lebih lanjut tentang harga, lihat halaman harga SageMaker Canvas. Anda juga dapat melihat Kelola penagihan dan biaya di Canvas SageMaker untuk informasi lebih lanjut.

SageMaker Kanvas saat ini tersedia di Wilayah berikut:

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Seoul)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Eropa (Stockholm)
- Amerika Selatan (Sao Paulo)

Topik

Apakah Anda pengguna SageMaker Canvas pertama kali?

Jika Anda adalah pengguna SageMaker Canvas pertama kali, kami sarankan Anda mulai dengan membaca bagian berikut:

- Untuk administrator TI — Menyiapkan dan Mengelola Amazon SageMaker Canvas (untuk Administrator TI)
- Untuk analis dan pengguna individu — Memulai dengan menggunakan Amazon SageMaker Canvas

Memulai dengan menggunakan Amazon SageMaker Canvas

Panduan ini memberi tahu Anda cara memulai menggunakan SageMaker Canvas. Jika Anda seorang administrator TI dan ingin detail lebih mendalam, lihat Menyiapkan dan Mengelola Amazon SageMaker Canvas (untuk Administrator TI) untuk menyiapkan SageMaker Canvas untuk pengguna Anda.

Topik

Prasyarat untuk menyiapkan Amazon Canvas SageMaker

Untuk menyiapkan aplikasi SageMaker Canvas, Anda harus terlebih dahulu onboard ke Amazon SageMaker Domain, yang mendukung berbagai lingkungan machine learning (ML) seperti Canvas dan SageMaker Studio.

Bagian berikut menjelaskan cara menyiapkan SageMaker Domain Amazon dan memberi diri Anda izin Canvas.

> **Important**
>
> Agar Anda dapat mengatur Amazon SageMaker Canvas, versi Amazon SageMaker Studio Anda harus 3.19.0 atau yang lebih baru. Untuk informasi tentang memperbarui Amazon SageMaker Studio, lihatMatikan dan Perbarui SageMaker Studio Classic.

Onboard ke Domain

Untuk menyiapkan Domain Anda, pertama-tama lihat Ikhtisar SageMaker Domain Amazon untuk mempelajari lebih lanjut tentang Domain.

Kemudian, ketika Anda siap untuk mengatur Domain Anda, pilih salah satu metode penyiapan berikut:

1. (Cepat) Cepat onboard ke Domain Amazon SageMaker — Pilih opsi ini jika Anda ingin mengatur Domain dengan cepat. Ini memberi pengguna Anda semua izin Canvas default dan fungsionalitas dasar. Fitur tambahan apa pun seperti kueri dokumen dapat diaktifkan nanti oleh admin. Jika Anda ingin mengonfigurasi izin yang lebih terperinci, kami sarankan Anda memilih opsi 2 atau 3.
2. (Lanjutan) Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center — Pilih opsi ini jika Anda ingin menyelesaikan pengaturan Domain yang lebih maju. Untuk menggunakan metode IAM Identity Center, Anda harus menjadi bagian dari organisasi diAWS Organizations.
3. (Lanjutan) Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM — Pilih opsi ini jika Anda ingin menyelesaikan pengaturan Domain yang lebih maju dan tidak memerlukan pengaturan organisasi Pusat Identitas IAM.

Jika Anda melakukan pengaturan Cepat (opsi 1 di daftar sebelumnya), maka Anda dapat melewati sisa bagian ini dan melanjutkan ke. Langkah 1: Masuk ke SageMaker Canvas

Jika Anda melakukan pengaturan Lanjutan (opsi 2 atau 3), maka Anda dapat menentukan fitur Canvas yang ingin Anda berikan kepada pengguna Anda akses. Gunakan sisa bagian ini saat Anda menyelesaikan pengaturan Domain lanjutan untuk membantu Anda mengonfigurasi izin yang khusus untuk Canvas.

Baik dalam Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center atau instruksi Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM penyiapan, untuk Langkah 2: Pengguna dan Aktivitas ML, Anda harus memilih izin Canvas yang ingin Anda berikan. Di bagian aktivitas ML, Anda dapat memilih kebijakan izin berikut untuk memberikan akses ke fitur Canvas. Anda hanya dapat memilih hingga total 8 aktivitas ML saat menyiapkan Domain Anda. Dua izin pertama dalam daftar berikut diperlukan untuk menggunakan Canvas, sedangkan sisanya untuk fitur tambahan.

- Jalankan Aplikasi Studio — Izin ini diperlukan untuk memulai aplikasi Canvas.
- Akses Inti Canvas - Izin ini memberi Anda akses ke aplikasi Canvas dan fungsionalitas dasar Canvas, seperti membuat kumpulan data, menggunakan transformasi data dasar, dan membangun dan menganalisis model.
- (Opsional) Persiapan Data Kanvas (didukung oleh Data Wrangler) - Izin ini memberi Anda akses untuk membuat aliran data dan menggunakan transformasi lanjutan untuk menyiapkan data Anda di Canvas. Izin ini juga diperlukan untuk membuat pekerjaan pemrosesan data dan jadwal pekerjaan persiapan data.
- (Opsional) Layanan Canvas AI — Izin ini memberi Anda akses ke eady-to-use model R, model dasar, dan fitur Obrolan dengan Data di Canvas.
- (Opsional) Akses Kendra — Izin ini memberi Anda akses ke fitur kueri dokumen, tempat Anda dapat menanyakan dokumen yang disimpan dalam indeks Amazon Kendra menggunakan model dasar di Canvas.

  Jika Anda memilih opsi ini, maka di bagian Akses Kendra Kanvas, masukkan ID untuk indeks Amazon Kendra Anda yang ingin Anda berikan aksesnya.
- (Opsional) Canvas MLOPs — Izin ini memberi Anda akses ke fitur penerapan model di Canvas, tempat Anda dapat menerapkan model untuk digunakan dalam produksi.

Di bagian Langkah 3: Aplikasi pengaturan Domain, pilih Konfigurasi Kanvas dan kemudian lakukan hal berikut:

1. Untuk konfigurasi penyimpanan Canvas, tentukan di mana Anda ingin Canvas menyimpan data aplikasi, seperti artefak model, prediksi batch, kumpulan data, dan log. SageMaker membuat

`Canvas/` folder di dalam bucket ini untuk menyimpan data. Untuk informasi selengkapnya, lihat Konfigurasikan penyimpanan Amazon S3 Anda. Untuk bagian ini, lakukan hal berikut:

a. Pilih System managed jika Anda ingin menyetel lokasi ke bucket SageMaker -created default yang mengikuti pola`s3://sagemaker-{Region}-{your-account-id}`.

b. Pilih Custom S3 untuk menentukan bucket Amazon S3 Anda sendiri sebagai lokasi penyimpanan. Kemudian, masukkan URI Amazon S3.

c. (Opsional) Untuk kunci Enkripsi, tentukan kunci KMS untuk mengenkripsi artefak Canvas yang disimpan di lokasi yang ditentukan.

2. (Opsional) Untuk konfigurasi eady-to-use model Canvas R, lakukan hal berikut:

a. Biarkan opsi Aktifkan eady-to-use model R Canvas diaktifkan untuk memberikan izin kepada pengguna Anda untuk menghasilkan prediksi dengan eady-to-use model R di Canvas (diaktifkan secara default). Opsi ini juga memberi Anda izin untuk mengobrol dengan model yang didukung Generative-AI. Untuk informasi selengkapnya, lihat Gunakan AI generatif dengan model foundation.

b. Biarkan opsi Aktifkan kueri dokumen menggunakan Amazon Kendra diaktifkan untuk memberikan izin kepada pengguna Anda untuk menggunakan model dasar untuk menanyakan dokumen yang disimpan dalam indeks Amazon Kendra. Kemudian, dari menu tarik-turun, pilih indeks yang ada yang ingin Anda berikan akses. Untuk informasi selengkapnya, lihat Gunakan AI generatif dengan model foundation.

3. (Opsional) Untuk bagian konfigurasi izin Ops ML, lakukan hal berikut:

a. Biarkan opsi Aktifkan penerapan langsung model Canvas diaktifkan untuk memberikan izin kepada pengguna Anda untuk menerapkan model mereka dari Canvas ke titik akhir. SageMaker Untuk informasi selengkapnya tentang penerapan model di Canvas, lihatMenerapkan model Anda ke titik akhir.

b. Biarkan opsi Aktifkan izin pendaftaran Registri Model untuk semua pengguna diaktifkan untuk memberikan izin kepada pengguna Anda untuk mendaftarkan versi model mereka ke registri SageMaker model (diaktifkan secara default). Untuk informasi selengkapnya, lihat Daftarkan versi model di registri SageMaker model.

c. Jika Anda membiarkan opsi Aktifkan izin pendaftaran Registri Model untuk semua pengguna diaktifkan, lalu pilih Daftar ke Registri Model saja atau Daftar dan setujui model di Registri Model.

4. (Opsional) Untuk bagian konfigurasi unggahan file lokal, aktifkan opsi Aktifkan unggahan file lokal untuk memberikan izin kepada pengguna Anda untuk mengunggah file ke Canvas dari

mesin lokal mereka. Mengaktifkan opsi ini akan melampirkan kebijakan berbagi sumber daya lintas asal (CORS) ke bucket Amazon S3 yang ditentukan dalam konfigurasi penyimpanan Canvas (dan mengganti kebijakan CORS yang ada). Untuk mempelajari lebih lanjut tentang izin upload file lokal, lihatBerikan Izin Pengguna Anda untuk Mengunggah File Lokal.

5. (Opsional) Untuk bagian pengaturan OAuth, lakukan hal berikut:

   a. Pilih Tambahkan konfigurasi OAuth.

   b. Untuk Sumber data, pilih sumber data Anda.

   c. Untuk pengaturan Rahasia, pilih Buat rahasia baru dan masukkan informasi yang Anda miliki dari penyedia identitas Anda. Jika Anda belum melakukan pengaturan OAuth awal dengan sumber data Anda, lihat. Siapkan koneksi ke sumber data dengan OAuth

6. (Opsional) Untuk konfigurasi peramalan deret waktu, biarkan opsi Aktifkan peramalan deret waktu diaktifkan untuk memberikan izin kepada pengguna Anda untuk melakukan peramalan deret waktu di SageMaker Canvas (diaktifkan secara default).

   - Jika Anda membiarkan Aktifkan peramalan deret waktu diaktifkan, pilih Buat dan gunakan peran eksekusi baru, atau pilih Gunakan peran eksekusi yang ada jika Anda sudah memiliki peran IAM dengan izin Amazon Forecast yang diperlukan terlampir (untuk informasi selengkapnya, lihat metode penyiapan peran IAM.

7. Selesai mengonfigurasi sisa pengaturan Domain menggunakan Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM prosedur Orientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center atau.

> **Note**
>
> Jika Anda mengalami masalah dengan pemberian izin melalui konsol, seperti izin untuk eady-to-use model R, lihat topiknya. Memecahkan masalah dengan pemberian izin melalui konsol SageMaker

Anda sekarang harus memiliki pengaturan SageMaker Domain dan semua izin Canvas dikonfigurasi.

Anda dapat mengedit izin Canvas untuk Domain atau pengguna tertentu setelah penyiapan Domain awal. Pengaturan pengguna individu mengesampingkan pengaturan Domain. Untuk mempelajari cara melihat atau mengedit izin Canvas Anda di pengaturan Domain, lihatLihat dan Edit Domain.

Beri diri Anda izin untuk menggunakan fitur tertentu di Canvas

Informasi berikut menguraikan berbagai izin yang dapat Anda berikan kepada pengguna Canvas untuk memungkinkan penggunaan berbagai fitur dan fungsi dalam Canvas. Beberapa izin ini dapat diberikan selama pengaturan Domain, tetapi beberapa memerlukan izin atau konfigurasi tambahan. Lihat informasi izin khusus untuk setiap fitur yang ingin Anda aktifkan:

- Unggahan file lokal. Izin untuk upload file lokal diaktifkan secara default di izin dasar Canvas saat mengatur Domain Anda. Jika Anda tidak dapat mengunggah file lokal dari mesin ke SageMaker Canvas, Anda dapat melampirkan kebijakan CORS ke bucket Amazon S3 yang Anda tentukan dalam konfigurasi penyimpanan Canvas. Jika Anda SageMaker diizinkan menggunakan bucket default, bucket mengikuti pola penamaan`s3://sagemaker-{Region}-{your-account-id}`. Untuk informasi selengkapnya, lihat Memberi Izin Pengguna Anda untuk Mengunggah File Lokal.
- Model prediksi gambar dan teks khusus. Izin untuk membuat model prediksi gambar dan teks kustom diaktifkan secara default di izin dasar Canvas saat menyiapkan Domain Anda. Namun, jika Anda memiliki konfigurasi IAM khusus dan tidak ingin melampirkan AmazonSageMakerCanvasFullAccesskebijakan tersebut ke peran eksekusi IAM pengguna, Anda harus secara eksplisit memberikan izin yang diperlukan kepada pengguna Anda. Untuk informasi selengkapnya, lihat Berikan Izin Pengguna Anda untuk Membuat Model Prediksi Gambar dan Teks Kustom.
- eady-to-use Model R dan model pondasi. Anda mungkin ingin menggunakan eady-to-use model Canvas R untuk membuat prediksi untuk data Anda. Dengan izin eady-to-use model R, Anda juga dapat mengobrol dengan model bertenaga AI generatif. Izin diaktifkan secara default saat menyiapkan Domain Anda, atau Anda dapat mengedit izin untuk Domain yang telah Anda buat. Opsi izin eady-to-use model Canvas R menambahkan ServicesAccess kebijakan AmazonSageMakerCanvasAI ke peran eksekusi Anda. Untuk informasi selengkapnya, lihat Memulai bagian dokumentasi eady-to-use model R.

  Untuk informasi lebih lanjut tentang memulai dengan model foundation AI generatif, lihatGunakan AI generatif dengan model foundation.
- Model pondasi yang disempurnakan. Jika Anda ingin menyempurnakan model dasar di Canvas, Anda dapat menambahkan izin saat menyiapkan Domain Anda, atau Anda dapat mengedit izin untuk Domain atau profil pengguna setelah membuat Domain Anda. Anda harus menambahkan ServicesAccess kebijakan AmazonSageMakerCanvasAI ke peran AWS IAM yang Anda pilih saat menyiapkan profil pengguna, dan Anda juga harus menambahkan hubungan kepercayaan dengan Amazon Bedrock ke peran tersebut. Untuk petunjuk tentang cara menambahkan izin ini ke peran IAM Anda, lihat. Berikan Izin Pengguna untuk Menyempurnakan Model Foundation

- Peramalan deret waktu. Jika Anda ingin melakukan prakiraan pada data deret waktu, Anda dapat menambahkan izin peramalan deret waktu saat menyiapkan Domain Anda, atau Anda dapat mengedit izin untuk Domain atau profil pengguna setelah membuat Domain Anda. Izin yang diperlukan adalah kebijakan `AmazonSageMakerCanvasForecastAccess` terkelola dan hubungan kepercayaan dengan Amazon Forecast dengan peran AWS IAM yang Anda pilih saat menyiapkan profil pengguna. Untuk petunjuk tentang cara menambahkan izin ini ke peran IAM Anda, lihat Memberi Izin Pengguna Anda untuk Melakukan Peramalan Deret Waktu.
- Kirim prediksi batch ke Amazon QuickSight. Anda mungkin ingin mengirim prediksi batch, atau kumpulan data prediksi yang Anda hasilkan dari model khusus, ke Amazon untuk dianalisis. QuickSight Di QuickSight, Anda dapat membangun dan mempublikasikan dasbor prediktif dengan hasil prediksi Anda. Untuk petunjuk tentang cara menambahkan izin ini ke peran IAM pengguna Canvas Anda, lihat Memberi Izin Pengguna Anda untuk Mengirim Prediksi ke Amazon. QuickSight
- Terapkan model Canvas ke titik SageMaker akhir. SageMakerHosting menawarkan titik akhir yang dapat Anda gunakan untuk menerapkan model Anda untuk digunakan dalam produksi. Anda dapat menerapkan model bawaan Canvas ke SageMaker titik akhir dan kemudian membuat prediksi secara terprogram di lingkungan produksi. Untuk informasi selengkapnya, lihat Menerapkan model Anda ke titik akhir.
- Daftarkan versi model ke registri model. Anda mungkin ingin mendaftarkan versi model Anda ke registri SageMaker model, yang merupakan repositori untuk melacak status versi terbaru model Anda. Ilmuwan data atau tim MLOPs yang bekerja di registri SageMaker model dapat melihat versi model yang telah Anda buat dan menyetujui atau menolaknya. Kemudian, mereka dapat menerapkan versi model Anda ke produksi atau memulai alur kerja otomatis. Izin pendaftaran model diaktifkan secara default untuk Domain Anda. Anda dapat mengelola izin di tingkat profil pengguna dan memberikan atau menghapus izin kepada pengguna tertentu. Untuk informasi selengkapnya, lihat Daftarkan versi model di registri SageMaker model.
- Kolaborasi dengan ilmuwan data. Jika ingin berkolaborasi dengan pengguna Studio Classic dan berbagi model, Anda harus menambahkan izin tambahan ke peran AWS IAM yang Anda pilih saat menyiapkan profil pengguna. Untuk petunjuk tentang cara menambahkan kebijakan ke peran, lihat Memberi Izin Pengguna untuk Berkolaborasi dengan Studio Classic.
- Impor data dari Amazon Redshift. Jika Anda ingin mengimpor data dari Amazon Redshift, Anda harus memberi diri Anda izin tambahan. Anda harus menambahkan kebijakan `AmazonRedshiftFullAccess` terkelola ke peran AWS IAM yang Anda pilih saat menyiapkan profil pengguna. Untuk petunjuk tentang cara menambahkan kebijakan ke peran, lihat Memberi Izin Pengguna untuk Mengimpor Data Amazon Redshift.

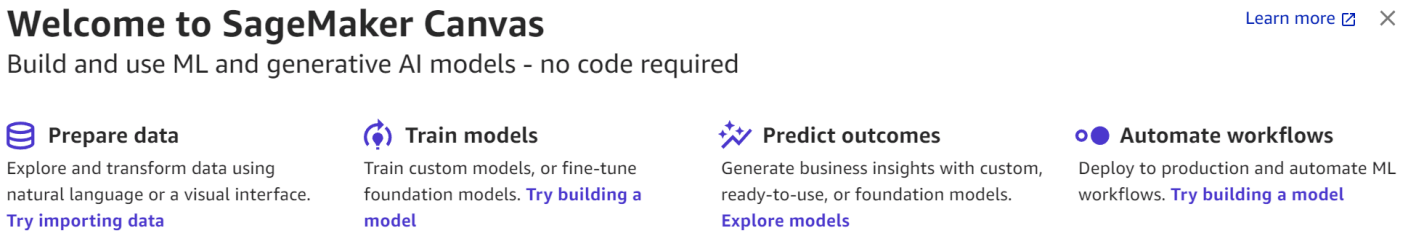
> **Note**
>
> Izin yang diperlukan untuk mengimpor melalui sumber data lain, seperti platform Amazon Athena dan SaaS, termasuk dalam kebijakan dan. AmazonSageMakerFullAccessAmazonSageMakerCanvasFullAccess Jika Anda mengikuti petunjuk penyiapan standar, kebijakan ini harus sudah dilampirkan ke peran eksekusi Anda. Untuk informasi selengkapnya tentang sumber data ini dan izinnya, lihatConnect ke sumber data.

Langkah 1: Masuk ke SageMaker Canvas

Ketika pengaturan awal selesai, Anda dapat mengakses SageMaker Canvas dengan salah satu metode berikut, tergantung pada kasus penggunaan Anda:

- Di SageMaker konsol, pilih Canvas di panel navigasi kiri. Kemudian, pada halaman Canvas, pilih pengguna Anda dari dropdown dan luncurkan aplikasi Canvas.
- Buka SageMaker Studio, dan di antarmuka Studio, buka halaman Canvas dan luncurkan aplikasi Canvas.
- Gunakan metode SSO berbasis SAMP 2.0 organisasi Anda, seperti Okta atau Pusat Identitas IAM.

Saat Anda masuk ke SageMaker Canvas untuk pertama kalinya, ada pesan selamat datang dengan tautan cepat untuk membantu Anda mulai menggunakan fitur utama Canvas.



Jika Anda menutup pesan ini, Anda dapat mengunjunginya kembali kapan saja dengan memilih tombol Bantuan di panel navigasi kiri aplikasi.

Langkah 2: Gunakan SageMaker Canvas untuk mendapatkan prediksi

Setelah masuk ke Canvas, Anda dapat mulai membuat model dan menghasilkan prediksi untuk data Anda.

Anda dapat menggunakan eady-to-use model Canvas R untuk membuat prediksi tanpa membuat model, atau Anda dapat membuat model khusus untuk masalah bisnis spesifik Anda. Tinjau informasi berikut untuk memutuskan apakah eady-to-use model R atau model khusus terbaik untuk kasus penggunaan Anda.

- eady-to-use Model R. Dengan eady-to-use model R, Anda dapat menggunakan model pra-bangun untuk mengekstrak wawasan dari data Anda. eady-to-use Model R mencakup berbagai kasus penggunaan, seperti deteksi bahasa dan analisis dokumen. Untuk mulai membuat prediksi dengan eady-to-use model R, lihatGunakan eady-to-use model R.
- Model kustom. Dengan model khusus, Anda dapat membuat berbagai jenis model yang disesuaikan untuk membuat prediksi data Anda. Gunakan model khusus jika Anda ingin membuat model yang dilatih pada data spesifik bisnis Anda dan jika Anda ingin menggunakan fitur seperti berkolaborasi dengan ilmuwan data dan mengevaluasi kinerja model Anda. Untuk memulai membangun model khusus, lihatGunakan model khusus.

Anda juga dapat membawa model Anda sendiri (BYOM) dari fitur lain. SageMaker Pengguna Amazon SageMaker Studio dapat berbagi model mereka dengan pengguna Canvas, dan pengguna Canvas dapat menghasilkan prediksi dengan model tersebut. Untuk mempelajari lebih lanjut, lihat Membawa model Anda sendiri ke SageMaker Canvas.

Menyiapkan dan Mengelola Amazon SageMaker Canvas (untuk Administrator TI)

Anda dapat menggunakan informasi di bagian ini untuk membantu pengguna Anda melakukan hal berikut:

- Opsional: Berikan izin kepada pengguna Anda untuk mengunggah file mereka secara lokal.
- Siapkan Okta SSO untuk pengguna Anda.
- Perbarui SageMaker Canvas.
- Bersihkan atau hapus instalasi SageMaker Canvas.
- Opsional: Siapkan Amazon Forecast sehingga pengguna dapat melakukan peramalan deret waktu.
- Opsional: Siapkan Amazon Virtual Private Cloud.
- Opsional: Enkripsi data menggunakanAWS Key Management Service.
- Opsional: Berikan izin kepada pengguna Anda untuk mengimpor data Amazon Redshift.

Anda juga dapat mengatur SageMaker Canvas untuk pengguna Anda denganAWS CloudFormation. Untuk informasi selengkapnya, lihat AWS:SageMaker: :App di Panduan AWS CloudFormation Pengguna.

Topik

Berikan Izin Pengguna Anda untuk Mengunggah File Lokal

Jika pengguna Anda mengunggah file dari mesin lokal mereka ke SageMaker Canvas, Anda harus melampirkan konfigurasi CORS (cross-origin resource sharing) ke bucket Amazon S3 yang mereka gunakan. Saat menyiapkan SageMaker Domain atau profil pengguna, Anda dapat menentukan lokasi Amazon S3 kustom atau lokasi default, yang merupakan bucket Amazon SageMaker S3 yang dibuat dengan nama yang menggunakan pola berikut:. `s3://sagemaker-{Region}-{your-account-id}` SageMaker Canvas menambahkan data pengguna Anda ke bucket setiap kali mereka mengunggah file.

Untuk memberikan izin kepada pengguna untuk mengunggah file lokal ke bucket, Anda dapat melampirkan konfigurasi CORS menggunakan salah satu prosedur berikut. Anda dapat

menggunakan metode pertama saat menyiapkan Domain atau mengedit pengaturan Domain yang ada, di mana Anda memilih untuk mengizinkan SageMaker untuk melampirkan konfigurasi CORS ke bucket untuk Anda. Metode kedua adalah metode manual, di mana Anda dapat melampirkan konfigurasi CORS ke ember sendiri.

Metode pengaturan domain

Untuk memberikan izin kepada pengguna untuk mengunggah file lokal, Anda dapat memilih Aktifkan izin Canvas saat menyiapkan Domain Anda. Ini melampirkan konfigurasi Cross-Origin Resource Sharing (CORS) ke bucket Amazon S3 konfigurasi penyimpanan Canvas dan memberikan izin kepada semua pengguna di Domain untuk mengunggah file lokal ke Canvas. SageMaker Secara default, opsi izin diaktifkan saat Anda menyiapkan Domain, tetapi Anda dapat menonaktifkan opsi ini jika Anda tidak ingin memberikan izin kepada pengguna untuk mengunggah file.

> **Note**
>
> Jika Anda memiliki konfigurasi CORS yang ada pada bucket Amazon S3 konfigurasi penyimpanan, mengaktifkan izin Aktifkan Kanvas akan menimpa konfigurasi yang ada dengan konfigurasi baru.

Prosedur berikut menunjukkan bagaimana Anda dapat mengaktifkan opsi ini saat melakukan Penyiapan cepat untuk Domain Anda di konsol.

1. Di bagian Profil pengguna, masukkan Nama untuk pengguna.
2. Pilih peran Eksekusi untuk pengguna.
3. Aktifkan Izin Aktifkan SageMaker Kanvas. (Secara default, opsi ini diaktifkan.)
4. Selesai menyiapkan Domain.

Jika Anda melakukan pengaturan Standar untuk Domain Anda, gunakan prosedur berikut untuk bagian pengaturan Canvas untuk mengaktifkan unggahan file lokal.

1. Untuk Aktifkan dan konfigurasikan izin Canvas, pilih Unggah file lokal. (Sudah diperiksa secara default.)
2. Pilih Berikutnya.
3. Selesai menyiapkan Domain.

Pengguna Anda sekarang dapat mengunggah file lokal ke dalam aplikasi SageMaker Canvas mereka.

Anda juga dapat mengaktifkan atau menonaktifkan izin upload lokal untuk Domain yang ada dengan menggunakan prosedur berikut.

1. Pergi ke SageMaker konsol Amazon.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain Anda.
5. Pada halaman Pengaturan domain, pilih tab Pengaturan domain.
6. Pilih Edit.
7. Di panel navigasi, pilih Pengaturan kanvas.
8. Pilih atau batal pilih Aktifkan unggahan file lokal.
9. Selesaikan modifikasi lain yang ingin Anda lakukan pada Domain, lalu pilih Kirim untuk mengirimkan perubahan Anda.

Metode bucket Amazon S3

Jika Anda ingin melampirkan konfigurasi CORS secara manual ke bucket SageMaker Amazon S3, gunakan prosedur berikut.

1. Masuk ke https://console.aws.amazon.com/s3/.
2. Pilih bucket Anda. Jika Domain Anda menggunakan bucket SageMaker -created default, nama bucket menggunakan pola berikut:`s3://sagemaker-{Region}-{your-account-id}`.
3. Pilih Izin.
4. Arahkan ke Cross-Origins Resource Sharing (CORS).
5. Pilih Edit.
6. Tambahkan kebijakan CORS berikut:

```
[
    {
        "AllowedHeaders": [
            "*"
```

```
        ],
        "AllowedMethods": [
            "POST"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": []
    }
]
```

7. Pilih Simpan perubahan.

Dalam prosedur sebelumnya, kebijakan CORS harus `"POST"` tercantum di bawah. `AllowedMethods`

Setelah Anda menjalani prosedur ini, Anda harus memiliki:

- Peran IAM yang ditetapkan untuk setiap pengguna Anda.
- Izin runtime Amazon SageMaker Studio Classic untuk setiap pengguna Anda. SageMaker Canvas menggunakan Studio Classic untuk menjalankan perintah dari pengguna Anda.
- Jika pengguna mengunggah file dari mesin lokal mereka, kebijakan CORS dilampirkan ke bucket Amazon S3 mereka.

Jika pengguna Anda masih tidak dapat mengunggah file lokal setelah Anda memperbarui kebijakan CORS, browser mungkin menyimpan setelan CORS dari upaya pengunggahan sebelumnya. Jika mereka mengalami masalah, instruksikan mereka untuk menghapus cache browser mereka dan coba lagi.

Mengatur SageMaker Kanvas untuk Pengguna Anda

Untuk menyiapkan Amazon SageMaker Canvas, lakukan hal berikut:

- Buat SageMaker Domain Amazon.
- Buat profil pengguna untuk Domain
- Siapkan Okta Single Sign On (Okta SSO) untuk pengguna Anda.
- Aktifkan berbagi tautan untuk model.

Gunakan Okta Single-Sign On (Okta SSO) untuk memberi pengguna Anda akses ke Amazon Canvas. SageMaker SageMaker Canvas mendukung metode SAFL 2.0 SSO. Bagian berikut memandu Anda melalui prosedur untuk mengatur Okta SSO.

Untuk menyiapkan Domain, lihat Onboard ke Amazon SageMaker Runtime Studio Menggunakan IAM. Anda dapat menggunakan informasi berikut untuk membantu Anda menyelesaikan prosedur di bagian ini:

- Anda dapat mengabaikan langkah tentang membuat proyek.
- Anda tidak perlu menyediakan akses ke bucket Amazon S3 tambahan. Pengguna Anda dapat menggunakan bucket default yang kami sediakan saat kami membuat peran.
- Untuk memberi pengguna Anda akses untuk berbagi buku catatan mereka dengan ilmuwan data, aktifkan Konfigurasi Berbagi Notebook.
- Gunakan Amazon SageMaker Studio Classic versi 3.19.0 atau yang lebih baru. Untuk informasi tentang memperbarui Amazon SageMaker Studio Classic, lihatMatikan dan Perbarui SageMaker Studio Classic.

Gunakan prosedur berikut untuk mengatur Okta. Untuk semua prosedur berikut, Anda menentukan peran IAM yang sama untuk*IAM-role*.

Tambahkan aplikasi SageMaker Canvas ke Okta

Siapkan metode sign-on untuk Okta.

1. Masuk ke dasbor Admin Okta.
2. Pilih Tambahkan aplikasi. Cari Federasi AWS Akun.
3. Pilih Tambahkan.
4. Opsional: Ubah nama menjadi Amazon SageMaker Canvas.
5. Pilih Berikutnya.
6. Pilih SAMP 2.0 sebagai metode Sign-On.
7. Pilih Metadata Penyedia Identitas untuk membuka file XMLmetadata. Simpan file secara lokal.
8. Pilih Selesai.

Mengatur federasi ID di IAM

AWS Identity and Access Management(IAM) adalah AWS layanan yang Anda gunakan untuk mendapatkan akses ke AWS akun Anda. Anda mendapatkan akses AWS melalui akun IAM.

1. Masuk ke AWS konsol.
2. Pilih AWS Identity and Access Management(IAM).
3. Pilih Penyedia Identitas.
4. Pilih Buat Penyedia.
5. Untuk Configure Provider, tentukan yang berikut ini:
   - Jenis Penyedia - Dari daftar dropdown, pilih SAMP.
   - Nama Penyedia — Tentukan Okta.
   - Dokumen Metadata — Unggah dokumen XHTML yang telah Anda simpan secara lokal dari langkah 7. Tambahkan aplikasi SageMaker Canvas ke Okta
6. Temukan penyedia identitas Anda di bawah Penyedia Identitas. Salin nilai ARN Penyedianya.
7. Untuk Peran, pilih peran IAM yang Anda gunakan untuk akses Okta SSO.
8. Di bawah Trust Relationship untuk peran IAM, pilih Edit Trust Relationship.
9. Ubah kebijakan hubungan kepercayaan IAM dengan menentukan nilai ARN Penyedia yang telah Anda salin dan tambahkan kebijakan berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::123456789012:saml-provider/Okta"
      },
      "Action": [
        "sts:AssumeRoleWithSAML",
        "sts:SetSourceIdentity",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "SAML:aud": "https://signin.aws.amazon.com/saml"
```

```
        }
      }
    }
  ]
}
```

10. Untuk Izin, tambahkan kebijakan berikut:

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Sid": "AmazonSageMakerPresignedUrlPolicy",
           "Effect": "Allow",
           "Action": [
                "sagemaker:CreatePresignedDomainUrl",
                "sagemaker:CreatePresignedDomainUrlWithPrincipalTag"
           ],
           "Resource": "*"
       }
  ]
}
```

Konfigurasikan SageMaker Canvas di Okta

Konfigurasikan Amazon SageMaker Canvas di Okta menggunakan prosedur berikut.

Untuk mengonfigurasi Amazon SageMaker Canvas untuk menggunakan Okta, ikuti langkah-langkah di bagian ini. Anda harus menentukan nama pengguna unik untuk setiap SageMakerStudioProfileNamebidang. Misalnya, Anda dapat menggunakan `user.login` sebagai nilai. Jika nama pengguna berbeda dari nama profil SageMaker Canvas, pilih atribut identifikasi unik yang berbeda. Misalnya, Anda dapat menggunakan nomor ID karyawan untuk nama profil.

Untuk contoh nilai yang dapat Anda atur untuk Atribut, lihat kode mengikuti prosedur.

1. Di bawah Direktori, pilih Grup.
2. Tambahkan grup dengan pola berikut:`sagemaker#canvas#`*`IAM-role`*`#`*`AWS-account-id`*.
3. Di Okta, buka konfigurasi integrasi aplikasi Federasi AWS Akun.

4. Pilih Masuk untuk aplikasi Federasi AWS Akun.
5. Pilih Edit dan tentukan yang berikut ini:
   - SAML 2.0
   - Status Relai Default - *https://Wilayah .console.aws.amazon.com/sagemaker/home region= Wilayah #/studio/canvas/open/. StudioId* Anda dapat menemukan ID Studio Classic di konsol: https://console.aws.amazon.com/sagemaker/
6. Pilih Atribut.
7. Di SageMakerStudioProfileNamebidang, tentukan nilai unik untuk setiap nama pengguna. Nama pengguna harus cocok dengan nama pengguna yang Anda buat di konsol. AWS

```
Attribute 1:
Name: https://aws.amazon.com/SAML/Attributes/
PrincipalTag:SageMakerStudioUserProfileName
Value: ${user.login}

Attribute 2:
Name: https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys
Value: {"SageMakerStudioUserProfileName"}
```

8. Pilih Jenis Lingkungan. Pilih Reguler AWS.
   - Jika jenis lingkungan Anda tidak terdaftar, Anda dapat mengatur URL ACS Anda di bidang URL ACS. Jika jenis lingkungan Anda terdaftar, Anda tidak perlu memasukkan URL ACS
9. Untuk ARN Penyedia Identitas, tentukan ARN yang Anda gunakan pada langkah 6 dari prosedur sebelumnya.
10. Tentukan Durasi Sesi.
11. Pilih Bergabung dengan semua peran.
12. Aktifkan Gunakan Pemetaan Grup dengan menentukan bidang berikut:
    - Filter Aplikasi - `okta`
    - Filter Grup - `^aws\#\S+\#(?`*`IAM-role`*`[\w\-]+)\#(?`*`accountid`*`\d+)$`
    - Pola Nilai Peran - `arn:aws:iam::$`*`accountid`*`:saml-provider/Okta,arn:aws:iam::$`*`accountid`*`:role/`*`IAM-role`*
13. Pilih Simpan/Berikutnya.

14. Di bawah Penugasan, tetapkan aplikasi ke grup yang telah Anda buat.

Tambahkan kebijakan opsional tentang kontrol akses di IAM

Di IAM, Anda dapat menerapkan kebijakan berikut ke pengguna administrator yang membuat profil pengguna.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateSageMakerStudioUserProfilePolicy",
            "Effect": "Allow",
            "Action": "sagemaker:CreateUserProfile",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": [
                        "studiouserid"
                    ]
                }
            }
        }
    ]
}
```

Jika Anda memilih untuk menambahkan kebijakan sebelumnya ke pengguna admin, Anda harus menggunakan izin berikut dari. Mengatur federasi ID di IAM

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Sid": "AmazonSageMakerPresignedUrlPolicy",
           "Effect": "Allow",
           "Action": [
                "sagemaker:CreatePresignedDomainUrl",
                "sagemaker:CreatePresignedDomainUrlWithPrincipalTag"
           ],
```

```
            "Resource": "*",
            "Condition": {
                   "StringEquals": {
                       "sagemaker:ResourceTag/studiouserid": "${aws:PrincipalTag/
SageMakerStudioUserProfileName}"
                  }
             }
      }
  ]
}
```

Konfigurasikan penyimpanan Amazon S3 Anda

Saat menyiapkan aplikasi SageMaker Canvas, lokasi penyimpanan default untuk artefak model, kumpulan data, dan data aplikasi lainnya adalah bucket Amazon S3 yang dibuat Canvas. Bucket Amazon S3 default ini mengikuti pola penamaan `s3://sagemaker-{Region}-{your-account-id}` dan ada di Wilayah yang sama dengan aplikasi Canvas Anda.

Namun, Anda dapat menyesuaikan lokasi penyimpanan dan menentukan bucket Amazon S3 Anda sendiri untuk menyimpan data aplikasi Canvas. Anda mungkin ingin menggunakan bucket Amazon S3 Anda sendiri untuk menyimpan data aplikasi karena salah satu alasan berikut:

- Organisasi Anda memiliki konvensi penamaan internal untuk bucket Amazon S3.
- Anda ingin mengaktifkan akses lintas akun ke artefak model atau data Canvas lainnya.
- Anda ingin mematuhi pedoman keamanan internal, seperti membatasi pengguna ke bucket Amazon S3 tertentu atau artefak model.
- Anda ingin meningkatkan visibilitas dan akses ke log yang diproduksi oleh Canvas, terlepas dari AWS konsol atau SageMaker Studio Classic.

Dengan menentukan bucket Amazon S3 Anda sendiri, Anda dapat meningkatkan kontrol atas penyimpanan Anda sendiri dan mematuhi organisasi Anda.

Untuk memulai, Anda dapat membuat SageMaker Domain baru atau profil pengguna, atau Anda dapat memperbarui Domain atau profil pengguna yang ada. Perhatikan bahwa pengaturan profil pengguna mengganti pengaturan tingkat Domain. Misalnya, Anda dapat menggunakan konfigurasi bucket default di tingkat Domain, tetapi Anda dapat menentukan bucket Amazon S3 khusus untuk pengguna individual. Setelah menentukan bucket Amazon S3 Anda sendiri untuk Domain atau profil pengguna, Canvas membuat subfolder yang `Canvas/<UserProfileName>` dipanggil di bawah

input Amazon S3 URI dan menyimpan semua artefak yang dihasilkan dalam aplikasi Canvas di bawah subfolder ini.

> **Important**
>
> Jika Anda memperbarui Domain atau profil pengguna yang ada, Anda tidak lagi memiliki akses ke artefak Canvas Anda dari lokasi sebelumnya. File Anda masih berada di lokasi Amazon S3 lama, tetapi Anda tidak dapat lagi melihatnya dari Canvas. Konfigurasi baru akan berlaku saat Anda masuk ke aplikasi berikutnya.

Untuk informasi selengkapnya tentang pemberian akses lintas akun ke bucket Amazon S3, lihat Memberikan izin objek lintas akun di Panduan Pengguna Amazon S3.

Bagian berikut menjelaskan cara menentukan bucket Amazon S3 khusus untuk konfigurasi penyimpanan Canvas Anda. Jika Anda menyiapkan SageMaker Domain baru (atau pengguna baru di Domain), gunakan domain Metode penyiapan Domain baru atau domainMetode pengaturan profil pengguna baru. Jika Anda memiliki profil pengguna Canvas yang ada dan ingin memperbarui konfigurasi penyimpanan profil, gunakan fileMetode pengguna yang ada.

Sebelum Anda memulai

Jika Anda menentukan URI Amazon S3 dari akun AWS lain, atau jika Anda menggunakan bucket yang dienkripsi, Anda harus mengonfigurasi izin sebelum AWS KMS melanjutkan. Anda harus memberikan izin AWS IAM untuk memastikan Canvas dapat mengunduh dan mengunggah objek ke dan dari bucket Anda. Untuk informasi terperinci tentang cara memberikan izin yang diperlukan, lihatBerikan izin untuk penyimpanan Amazon S3 lintas akun.

Selain itu, URI Amazon S3 terakhir untuk folder pelatihan di lokasi penyimpanan Canvas Anda harus 128 karakter atau kurang. URI Amazon S3 terakhir terdiri dari jalur bucket Anda `s3://<your-bucket-name>/<folder-name>/` ditambah jalur yang ditambahkan Canvas ke bucket Anda:. `Canvas/<user-profile-name>/Training` Misalnya, jalur yang dapat diterima yang kurang dari 128 karakter adalah`s3://<my-bucket>/<machine-learning>/Canvas/<user-1>/Training`.

Metode penyiapan Domain baru

Jika Anda menyiapkan aplikasi Domain dan Canvas baru, gunakan bagian ini untuk mengonfigurasi lokasi penyimpanan di tingkat Domain. Konfigurasi ini berlaku untuk semua pengguna baru yang

Anda buat di Domain, kecuali Anda menentukan lokasi penyimpanan yang berbeda untuk masing-masing profil pengguna.

Saat melakukan pengaturan Standar untuk Domain Anda, gunakan prosedur berikut untuk bagian pengaturan Canvas:

1. Untuk konfigurasi penyimpanan Canvas, lakukan hal berikut:
    a. Pilih System managed jika Anda ingin menyetel lokasi ke bucket SageMaker -created default yang mengikuti pola`s3://sagemaker-{Region}-{your-account-id}`.
    b. Pilih Custom S3 untuk menentukan bucket Amazon S3 Anda sendiri sebagai lokasi penyimpanan. Kemudian, masukkan URI Amazon S3.
    c. (Opsional) Untuk kunci Enkripsi, tentukan kunci KMS untuk mengenkripsi artefak Canvas yang disimpan di lokasi yang ditentukan.
2. Selesaikan pengaturan Domain dan pilih Kirim.

Domain Anda sekarang dikonfigurasi untuk menggunakan lokasi Amazon S3 yang Anda tentukan untuk penyimpanan aplikasi SageMaker Canvas.

Metode pengaturan profil pengguna baru

Jika Anda menyiapkan profil pengguna baru di Domain Anda, gunakan bagian ini untuk mengonfigurasi lokasi penyimpanan bagi pengguna. Konfigurasi ini mengesampingkan konfigurasi tingkat Domain.

Saat menambahkan profil pengguna ke Domain Anda, gunakan prosedur berikut untuk bagian Pengaturan Canvas:

1. Untuk konfigurasi penyimpanan Canvas, lakukan hal berikut:
    a. Pilih Sistem dikelola jika Anda ingin menyetel lokasi ke bucket yang SageMaker dibuat default yang mengikuti pola`s3://sagemaker-{Region}-{your-account-id}`.
    b. Pilih Custom S3 untuk menentukan bucket Amazon S3 Anda sendiri sebagai lokasi penyimpanan. Kemudian, masukkan URI Amazon S3.
    c. (Opsional) Untuk kunci Enkripsi, tentukan kunci KMS untuk mengenkripsi artefak Canvas yang disimpan di lokasi yang ditentukan.
2. Selesaikan pengaturan profil pengguna dan pilih Kirim.

Profil pengguna Anda sekarang dikonfigurasi untuk menggunakan lokasi Amazon S3 yang Anda tentukan untuk penyimpanan aplikasi SageMaker Canvas.

Metode pengguna yang ada

Jika Anda memiliki profil pengguna Canvas yang ada dan ingin memperbarui lokasi penyimpanan Amazon S3, Anda dapat mengedit pengaturan SageMaker Domain atau profil pengguna. Perubahan akan berlaku saat berikutnya Anda masuk ke aplikasi Canvas.

> **Note**
>
> Ketika Anda mengubah lokasi penyimpanan untuk aplikasi Canvas yang ada, Anda kehilangan akses ke artefak Canvas Anda dari lokasi penyimpanan sebelumnya. Artefak masih disimpan di lokasi Amazon S3 lama, tetapi Anda tidak dapat lagi melihatnya dari Canvas.

Ingat bahwa pengaturan profil pengguna mengganti pengaturan Domain umum, sehingga Anda dapat memperbarui lokasi penyimpanan Amazon S3 untuk profil pengguna tertentu tanpa mengubahnya untuk semua pengguna. Anda dapat memperbarui konfigurasi penyimpanan untuk Domain atau pengguna yang ada dengan menggunakan prosedur berikut.

Update an existing Domain

Gunakan prosedur berikut untuk memperbarui konfigurasi penyimpanan untuk Domain.

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain Anda.
5. Pada halaman Pengaturan domain, pilih tab Pengaturan domain.
6. Pilih Edit.
7. Di panel navigasi, pilih Pengaturan kanvas.
8. Untuk konfigurasi penyimpanan Canvas, lakukan hal berikut:

    a. Pilih Sistem dikelola jika Anda ingin menyetel lokasi ke bucket yang SageMaker dibuat default yang mengikuti pola `s3://sagemaker-{Region}-{your-account-id}`.

b. Pilih Custom S3 untuk menentukan bucket Amazon S3 Anda sendiri sebagai lokasi penyimpanan. Kemudian, masukkan URI Amazon S3.

c. (Opsional) Untuk kunci Enkripsi, tentukan kunci KMS untuk mengenkripsi artefak Canvas yang disimpan di lokasi yang ditentukan.

9. Selesaikan modifikasi lain yang ingin Anda lakukan pada Domain, lalu pilih Kirim untuk menyimpan perubahan Anda.

Update an existing user profile

Gunakan prosedur berikut untuk memperbarui konfigurasi penyimpanan untuk profil pengguna.

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain Anda.
5. Dari daftar pengguna di Domain, pilih pengguna yang konfigurasinya ingin Anda edit.
6. Pada halaman Detail Pengguna, pilih Edit.
7. Di panel navigasi, pilih Pengaturan kanvas.
8. Untuk konfigurasi penyimpanan Canvas, lakukan hal berikut:

   a. Pilih System managed jika Anda ingin menyetel lokasi ke bucket SageMaker -created default yang mengikuti pola`s3://sagemaker-{Region}-{your-account-id}`.

   b. Pilih Custom S3 untuk menentukan bucket Amazon S3 Anda sendiri sebagai lokasi penyimpanan. Kemudian, masukkan URI Amazon S3.

   c. (Opsional) Untuk kunci Enkripsi, tentukan kunci KMS untuk mengenkripsi artefak Canvas yang disimpan di lokasi yang ditentukan.

9. Selesaikan modifikasi lain yang ingin Anda lakukan pada profil pengguna, lalu pilih Kirim untuk menyimpan perubahan Anda.

Lokasi penyimpanan untuk profil pengguna Canvas Anda sekarang harus diperbarui. Lain kali Anda masuk ke aplikasi Canvas, Anda menerima pemberitahuan bahwa lokasi penyimpanan telah diperbarui. Anda kehilangan akses ke artefak sebelumnya yang Anda buat di Canvas. Anda masih dapat mengakses file di Amazon S3, tetapi Anda tidak dapat lagi melihatnya di Canvas.

Berikan izin untuk penyimpanan Amazon S3 lintas akun

Saat menyiapkan SageMaker Domain atau profil pengguna bagi pengguna untuk mengakses SageMaker Canvas, Anda menentukan lokasi penyimpanan Amazon S3 untuk artefak Canvas. Artefak ini termasuk salinan yang disimpan dari kumpulan data input Anda, artefak model, prediksi, dan data aplikasi lainnya. Anda dapat menggunakan bucket Amazon S3 default yang SageMaker dibuat, atau Anda dapat menyesuaikan lokasi penyimpanan dan menentukan bucket Anda sendiri untuk menyimpan data aplikasi Canvas.

Anda dapat menentukan bucket Amazon S3 di AWS akun lain untuk menyimpan data Canvas, tetapi pertama-tama Anda harus memberikan izin lintas akun agar Canvas dapat mengakses bucket.

Bagian berikut menjelaskan cara memberikan izin ke Canvas untuk mengunggah dan mengunduh objek ke dan dari bucket Amazon S3 di akun lain. Ada izin tambahan saat bucket Anda dienkripsi. AWS KMS

Persyaratan

Sebelum Anda mulai, tinjau persyaratan berikut:

- Bucket Amazon S3 lintas akun (dan kunci AWS KMS terkait) harus berada di Wilayah AWS yang sama dengan Domain pengguna Canvas atau profil pengguna.
- URI Amazon S3 terakhir untuk folder pelatihan di lokasi penyimpanan Canvas Anda harus 128 karakter atau kurang. URI S3 terakhir terdiri dari jalur bucket Anda `s3://<your-bucket-name>/<folder-name>/` ditambah jalur yang ditambahkan Canvas ke bucket Anda:`Canvas/<user-profile-name>/Training`. Misalnya, jalur yang dapat diterima yang kurang dari 128 karakter adalah`s3://<my-bucket>/<machine-learning>/Canvas/<user-1>/Training`.

Izin untuk bucket Amazon S3 lintas akun

Bagian berikut menguraikan langkah-langkah dasar untuk memberikan izin yang diperlukan sehingga Canvas dapat mengakses bucket Amazon S3 Anda di akun lain. Untuk petunjuk selengkapnya, lihat Contoh 2: Pemilik bucket yang memberikan izin bucket lintas akun di Panduan Pengguna Amazon S3.

1. Buat bucket Amazon S3,`bucketA`, di Akun A.
2. Pengguna Canvas ada di akun lain yang disebut Akun B. Pada langkah-langkah berikut, kita mengacu pada peran IAM pengguna Canvas seperti `roleB` pada Akun B.

Berikan peran IAM `roleB` di Akun B izin untuk mengunduh (`GetObject`) dan mengunggah (`PutObject`) objek ke dan dari `bucketA` Akun A dengan melampirkan kebijakan IAM.

Untuk membatasi akses ke folder bucket tertentu, tentukan nama folder di elemen sumber daya, seperti`arn:aws:s3:::<bucketA>/FolderName/*`. Untuk informasi selengkapnya, lihat Bagaimana cara menggunakan kebijakan IAM untuk memberikan akses khusus pengguna ke folder tertentu?

> **Note**
>
> Tindakan tingkat ember, seperti `GetBucketCors` dan, harus ditambahkan pada sumber daya tingkat ember`GetBucketLocation`, bukan folder.

Contoh berikut kebijakan IAM memberikan izin yang diperlukan `roleB` untuk mengakses objek di: `bucketA`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucketA/FolderName/*",
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketCors",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::bucketA",
```

```
            ]
        }
    ]
}
```

3. Konfigurasikan kebijakan bucket `bucketA` di Akun A untuk memberikan izin ke peran IAM `roleB` di Akun B.

> **Note**
>
> Admin juga harus menonaktifkan Blokir semua akses publik di bawah bagian Izin bucket.

Berikut ini adalah contoh kebijakan bucket `bucketA` untuk memberikan izin yang diperlukan untuk`roleB`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::accountB:role/roleB"
            },
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::bucketA/FolderName/*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::accountB:role/roleB"
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketCors",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucketA"
```

```
        }
    ]
}
```

Setelah mengonfigurasi izin sebelumnya, profil pengguna Canvas Anda di Akun B sekarang dapat menggunakan bucket Amazon S3 di Akun A sebagai lokasi penyimpanan artefak Canvas.

Izin untuk bucket Amazon S3 lintas akun yang dienkripsi AWS KMS

Prosedur berikut menunjukkan cara memberikan izin yang diperlukan agar Canvas dapat mengakses bucket Amazon S3 Anda di akun lain yang dienkripsi. AWS KMS Langkah-langkahnya mirip dengan prosedur di atas, tetapi dengan izin tambahan. Untuk informasi selengkapnya tentang pemberian akses kunci KMS lintas akun, lihat Mengizinkan pengguna di akun lain menggunakan kunci KMS di Panduan Pengembang. AWS KMS

1. Buat bucket Amazon S3 `bucketA` dan kunci Amazon S3 `s3KmsInAccountA` KMS di Akun A.
2. Pengguna Canvas ada di akun lain yang disebut Akun B. Pada langkah-langkah berikut, kita mengacu pada peran IAM pengguna Canvas seperti `roleB` pada Akun B.

   Berikan izin kepada peran IAM `roleB` di Akun B untuk melakukan hal berikut:

   - Unduh (`GetObject`) dan unggah (`PutObject`) objek ke dan dari `bucketA` di Akun A.
   - Akses AWS KMS kunci `s3KmsInAccountA` di Akun A.

   Contoh berikut kebijakan IAM memberikan izin yang diperlukan `roleB` untuk mengakses objek `bucketA` dan menggunakan kunci KMS: `s3KmsInAccountA`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucketA/FolderName/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketCors",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::bucketA"
            ]
        },
        {
            "Action": [
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:RetireGrant",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlainText",
                "kms:Decrypt"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:kms:{region}:accountA:key/s3KmsInAccountA"
        }
    ]
}
```

3. Konfigurasikan kebijakan bucket untuk `bucketA` dan kebijakan utama `s3KmsInAccountA` di Akun A untuk memberikan izin ke peran IAM `roleB` di Akun B.

   Berikut ini adalah contoh kebijakan bucket `bucketA` untuk memberikan izin yang diperlukan untuk`roleB`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::accountB:role/roleB"
            },
            "Action": [
                "s3:DeleteObject",
```

```
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::bucketA/FolderName/*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::accountB:role/roleB"
            },
            "Action": [
                "s3:GetBucketCors",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucketA"
        }
    ]
}
```

Contoh berikut adalah kebijakan kunci yang Anda lampirkan ke kunci KMS `s3KmsInAccountA` di Akun A untuk memberikan `roleB` akses. Untuk informasi selengkapnya tentang cara membuat dan melampirkan pernyataan kebijakan utama, lihat Membuat kebijakan kunci di Panduan AWS KMS Pengembang.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::accountB:role/roleB"
    ]
  },
  "Action": [
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:RetireGrant",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlainText",
        "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Setelah mengonfigurasi izin sebelumnya, profil pengguna Canvas Anda di Akun B sekarang dapat menggunakan bucket Amazon S3 terenkripsi di Akun A sebagai lokasi penyimpanan artefak Canvas.

Enkripsi Data SageMaker Kanvas Anda dengan AWS KMS

Anda mungkin memiliki data yang ingin Anda enkripsi saat menggunakan Amazon SageMaker Canvas, seperti informasi perusahaan pribadi atau data pelanggan Anda. SageMaker Canvas digunakan AWS Key Management Service untuk melindungi data Anda. AWS KMSadalah layanan yang dapat Anda gunakan untuk membuat dan mengelola kunci kriptografi untuk mengenkripsi data Anda. Untuk informasi selengkapnya tentang AWS KMS, lihat AWS Key Management Service di Panduan Developer AWS KMS.

Amazon SageMaker Canvas memberi Anda beberapa opsi untuk mengenkripsi data Anda. SageMaker Canvas menyediakan enkripsi default dalam aplikasi untuk tugas-tugas seperti membangun model Anda dan menghasilkan wawasan. Anda juga dapat memilih untuk mengenkripsi data yang disimpan di Amazon S3 untuk melindungi data Anda saat istirahat. SageMaker Canvas mendukung pengimporan kumpulan data terenkripsi, sehingga Anda dapat membuat alur kerja terenkripsi. Bagian berikut menjelaskan bagaimana Anda dapat menggunakan AWS KMS enkripsi untuk melindungi data Anda saat membuat model dengan SageMaker Canvas.

Enkripsi data Anda di Canvas SageMaker

Dengan SageMaker Canvas, Anda dapat menggunakan dua kunci AWS KMS enkripsi yang berbeda untuk mengenkripsi data Anda di SageMaker Canvas, yang dapat Anda tentukan saat mengatur Domain Anda. Kedua kunci ini bisa sama atau berbeda. SageMaker Canvas menggunakan satu kunci untuk penyimpanan aplikasi sementara, visualisasi, atau tujuan komputasi (seperti model bangunan). Anda dapat menggunakan kunci AWS terkelola default atau menentukan sendiri. Anda juga dapat menentukan kunci opsional yang digunakan SageMaker Canvas untuk penyimpanan jangka panjang objek model dan kumpulan data, yang disimpan di bucket SageMaker S3 default Wilayah untuk akun Anda.

Prasyarat

Untuk menggunakan kunci KMS Anda sendiri untuk salah satu tujuan yang dijelaskan sebelumnya, Anda harus terlebih dahulu memberikan izin peran IAM pengguna Anda untuk menggunakan kunci tersebut. Kemudian, Anda dapat menentukan kunci KMS saat mengatur Domain Anda.

Cara termudah untuk memberikan izin peran Anda untuk menggunakan kunci adalah dengan memodifikasi kebijakan kunci. Gunakan prosedur berikut untuk memberikan peran Anda izin yang diperlukan.

1. Buka konsol AWS KMS.
2. Di bagian Kebijakan Kunci, pilih Beralih ke tampilan kebijakan.
3. Ubah kebijakan kunci untuk memberikan izin `kms:GenerateDataKey` dan `kms:Decrypt` tindakan ke peran IAM. Anda dapat menambahkan pernyataan yang mirip dengan berikut ini:

```
{
  "Sid": "ExampleStmt",
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Effect": "Allow",
  "Principal": {
    "AWS": "<arn:aws:iam::111122223333:role/Jane>"
  },
  "Resource": "*"
}
```

4. Pilih Simpan perubahan.

Metode yang kurang disukai adalah memodifikasi peran IAM pengguna untuk memberikan izin pengguna untuk menggunakan atau mengelola kunci KMS. Jika Anda menggunakan metode ini, kebijakan kunci KMS juga harus mengizinkan manajemen akses melalui IAM. Untuk mempelajari cara memberikan izin ke kunci KMS melalui peran IAM pengguna, lihat Menentukan kunci KMS dalam pernyataan kebijakan IAM di Panduan Pengembang. AWS KMS

Prasyarat untuk peramalan deret waktu

Untuk menggunakan AWS KMS kunci Anda untuk mengenkripsi model peramalan deret waktu di SageMaker Canvas, Anda harus mengubah kebijakan kunci untuk kunci KMS yang digunakan untuk menyimpan objek ke Amazon S3. Kebijakan utama Anda harus memberikan izin untuk`AmazonSageMakerCanvasForecastRole`, yang SageMaker dibuat saat Anda memberikan izin peramalan deret waktu untuk pengguna Anda. Amazon Forecast menggunakan `AmazonSageMakerCanvasForecastRole` untuk melakukan operasi peramalan deret waktu di SageMaker Canvas. Kunci KMS Anda harus memberikan izin untuk peran ini untuk memastikan data dienkripsi untuk peramalan deret waktu.

Untuk mengubah izin kebijakan kunci KMS Anda agar memungkinkan peramalan deret waktu terenkripsi, lakukan hal berikut.

1. Buka konsol AWS KMS.
2. Di bagian Kebijakan Kunci, pilih Beralih ke tampilan kebijakan.
3. Ubah kebijakan kunci agar izin ditentukan dalam contoh berikut:

```
{
            "Sid": "Enable IAM Permissions for Amazon Forecast KMS access",
            "Effect": "Allow",
            "Principal": {
                "AWS": "<arn:aws:iam::111122223333:role/service-role/
AmazonSagemakerCanvasForecastRole-111122223333>"
            },
            "Action": [
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:RetireGrant",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlainText",
                "kms:Decrypt"
            ],
            "Resource": "*"
}
```

4. Pilih Simpan perubahan.

Anda sekarang dapat menggunakan kunci KMS Anda untuk mengenkripsi operasi peramalan deret waktu di Canvas. SageMaker

> **Note**
>
> Izin berikut hanya diperlukan jika Anda menggunakan metode pengaturan peran IAM untuk mengonfigurasi peramalan deret waktu. Tambahkan kebijakan izin berikut ke peran IAM pengguna Anda. Anda juga harus memperbarui kebijakan utama dengan kebijakan terbaru yang diperlukan untuk Amazon Forecast. Untuk informasi selengkapnya tentang izin yang diperlukan untuk peramalan deret waktu, lihat. Berikan Izin Pengguna Anda untuk Melakukan Peramalan Deret Waktu

```
{
            "Sid": "Enable IAM Permissions for Amazon Forecast KMS access",
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": "<arn:aws:iam::111122223333:role/AmazonSageMaker-111122223333>"
            },
            "Action": [
                    "kms:Decrypt",
                    "kms:DescribeKey",
                    "kms:CreateGrant",
                    "kms:RetireGrant",
                    "kms:GenerateDataKey"
                    "kms:GenerateDataKeyWithoutPlainText",
            ],
            "Resource": "*"
}
```

Enkripsi data Anda di aplikasi SageMaker Canvas

Kunci KMS pertama yang dapat Anda gunakan di SageMaker Canvas digunakan untuk mengenkripsi data aplikasi yang disimpan di volume Amazon Elastic Block Store (Amazon EBS) Block Store (Amazon EBS) dan di Amazon Elastic File System yang dibuat di Domain Anda. SageMaker SageMaker Canvas mengenkripsi data Anda dengan kunci ini di aplikasi dasar dan sistem penyimpanan sementara yang dibuat saat menggunakan instance komputasi untuk membuat model dan menghasilkan wawasan. SageMaker Canvas meneruskan kunci ke AWS layanan lain, seperti Autopilot, setiap kali SageMaker Canvas memulai pekerjaan dengan mereka untuk memproses data Anda.

Anda dapat menentukan kunci ini dengan menyetel panggilan `CreateDomain` API atau saat melakukan pengaturan Domain Standar di konsol. `KmsKeyID` Jika Anda tidak menentukan kunci KMS Anda sendiri, SageMaker gunakan kunci KMS AWS terkelola default untuk mengenkripsi data Anda dalam aplikasi Canvas. SageMaker

Untuk menentukan kunci KMS Anda sendiri untuk digunakan dalam aplikasi SageMaker Canvas melalui konsol, pertama-tama siapkan SageMaker Domain Amazon Anda menggunakan pengaturan Standar. Gunakan prosedur berikut untuk menyelesaikan Bagian Jaringan dan Penyimpanan untuk Domain.

1. Isi pengaturan VPC Amazon yang Anda inginkan.
2. Untuk kunci Enkripsi, pilih Masukkan ARN kunci KMS.
3. Untuk KMS ARN, masukkan ARN untuk kunci KMS Anda, yang seharusnya memiliki format yang mirip dengan berikut ini: `arn:aws:kms:example-region-1:123456789098:key/111aa2bb-333c-4d44-5555-a111bb2c33dd`

Enkripsi data SageMaker Canvas Anda yang disimpan di Amazon S3

Kunci KMS kedua yang dapat Anda tentukan digunakan untuk data yang disimpan SageMaker Canvas ke Amazon S3. SageMaker Canvas menyimpan duplikat kumpulan data input, data aplikasi dan model, serta data keluaran ke bucket SageMaker S3 default Region untuk akun Anda. Pola penamaan untuk bucket ini adalah`s3://sagemaker-{Region}-{your-account-id}`, dan SageMaker Canvas menyimpan data di `Canvas/` folder.

1. Aktifkan Aktifkan berbagi sumber daya notebook.
2. Untuk lokasi S3 untuk sumber daya notebook yang dapat dibagikan, tinggalkan jalur Amazon S3 default. Perhatikan bahwa SageMaker Canvas tidak menggunakan jalur Amazon S3 ini; jalur Amazon S3 ini digunakan untuk notebook Studio Classic.
3. Untuk kunci Enkripsi, pilih Masukkan ARN kunci KMS.
4. Untuk KMS ARN, masukkan ARN untuk kunci KMS Anda, yang seharusnya memiliki format yang mirip dengan berikut ini: `arn:aws:kms:us-east-1:111122223333:key/111aa2bb-333c-4d44-5555-a111bb2c33dd`

Impor kumpulan data terenkripsi dari Amazon S3

Pengguna Anda mungkin memiliki kumpulan data yang telah dienkripsi dengan kunci KMS. Sementara bagian sebelumnya menunjukkan cara mengenkripsi data di SageMaker Canvas dan data yang disimpan ke Amazon S3, Anda harus memberikan izin tambahan peran IAM pengguna Anda jika Anda ingin mengimpor data dari Amazon S3 yang sudah dienkripsi. AWS KMS

Untuk memberikan izin pengguna Anda untuk mengimpor kumpulan data terenkripsi dari Amazon S3 SageMaker ke Canvas, tambahkan izin berikut ke peran eksekusi IAM yang telah Anda gunakan untuk profil pengguna.

```
    "kms:Decrypt",
    "kms:GenerateDataKey"
```

Untuk mempelajari cara mengedit izin IAM untuk peran, lihat Menambahkan dan menghapus izin identitas IAM di Panduan Pengguna IAM. Untuk informasi selengkapnya tentang kunci KMS, lihat Kebijakan utama AWS Key Management Service di Panduan AWS KMS Pengembang.

FAQ

Lihat item FAQ berikut untuk jawaban atas pertanyaan umum tentang AWS KMS dukungan SageMaker Canvas.

T: Apakah SageMaker Canvas mempertahankan kunci KMS saya?

A: Tidak. SageMaker Canvas dapat menyimpan sementara kunci Anda atau meneruskannya ke AWS layanan lain (seperti Autopilot), tetapi SageMaker Canvas tidak menyimpan kunci KMS Anda.

T: Saya menentukan kunci KMS saat mengatur Domain saya. Mengapa kumpulan data saya gagal diimpor di SageMaker Canvas?

J: Peran IAM pengguna Anda mungkin tidak memiliki izin untuk menggunakan kunci KMS tersebut. Untuk memberikan izin pengguna Anda, lihat. Prasyarat Kesalahan lain yang mungkin terjadi adalah Anda memiliki kebijakan bucket di bucket Amazon S3 yang mengharuskan penggunaan kunci KMS tertentu yang tidak cocok dengan kunci KMS yang Anda tentukan di Domain. Pastikan Anda menentukan kunci KMS yang sama untuk bucket Amazon S3 dan Domain Anda.

T: Bagaimana cara menemukan bucket SageMaker Amazon S3 default Wilayah untuk akun saya?

J: Bucket Amazon S3 default mengikuti pola penamaan. `s3://sagemaker-{Region}-{your-account-id}` `Canvas/`Folder di bucket ini menyimpan data aplikasi SageMaker Canvas Anda.

T: Dapatkah saya mengubah bucket SageMaker Amazon S3 default yang digunakan untuk menyimpan data SageMaker Canvas?

A: Tidak, SageMaker buat ember ini untuk Anda.

T: Apa yang disimpan SageMaker Canvas di bucket SageMaker Amazon S3 default?

J: SageMaker Canvas menggunakan bucket SageMaker Amazon S3 default untuk menyimpan duplikat kumpulan data input, artefak model, dan keluaran model Anda.

T: Kasus penggunaan apa yang didukung untuk menggunakan kunci KMS dengan SageMaker Canvas?

J: Dengan SageMaker Canvas, Anda dapat menggunakan kunci enkripsi Anda sendiri AWS KMS untuk membangun regresi, klasifikasi biner dan multi-kelas, dan model peramalan deret waktu, serta untuk inferensi batch dengan model Anda.

T: Dapatkah saya mengenkripsi model peramalan deret waktu di Canvas? SageMaker

J: Ya. Anda harus memberikan izin tambahan kunci KMS Anda untuk melakukan peramalan deret waktu terenkripsi. Untuk informasi selengkapnya tentang cara mengubah kebijakan kunci Anda untuk memberikan izin peramalan deret waktu, lihat. Prasyarat untuk peramalan deret waktu

Berikan Izin Pengguna Anda untuk Membuat Model Prediksi Gambar dan Teks Kustom

Di Amazon SageMaker Canvas, Anda dapat membuat model khusus untuk memenuhi kebutuhan bisnis spesifik Anda. Dua dari jenis model khusus ini adalah predisi gambar label tunggal dan prediksi teks multi-kategori. Izin untuk membangun tipe model ini disertakan dalam kebijakan AWS Identity and Access Management (IAM) yang disebut AmazonSageMakerCanvasFullAccess, yang secara default SageMaker dilampirkan ke peran eksekusi IAM pengguna Anda jika Anda membiarkan izin dasar Canvas diaktifkan.

Namun, jika Anda menggunakan konfigurasi IAM kustom, maka Anda harus secara eksplisit menambahkan izin ke peran eksekusi IAM pengguna Anda sehingga mereka dapat membangun gambar kustom dan jenis model prediksi teks. Untuk memberikan izin yang diperlukan untuk membuat model prediksi gambar dan teks, baca bagian berikut untuk mempelajari cara melampirkan kebijakan izin terkecil ke peran Anda.

Untuk menambahkan izin ke peran IAM pengguna, lakukan hal berikut:

1. Buka Konsol IAM.
2. Pilih Peran.
3. Di kotak pencarian, cari peran IAM pengguna berdasarkan nama dan pilih.
4. Pada halaman untuk peran pengguna, di bawah Izin, pilih Tambahkan izin.
5. Pilih Buat kebijakan sebaris.
6. Pilih tab JSON, lalu tempelkan kebijakan izin terkecil berikut ke editor.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateAutoMLJobV2",
```

```
                "sagemaker:DescribeAutoMLJobV2"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Pilih Tinjau kebijakan.
8. Masukkan Nama untuk kebijakan tersebut.
9. Pilih Buat kebijakan.

Untuk informasi selengkapnya tentang kebijakan AWS terkelola, lihat Kebijakan terkelola dan kebijakan sebaris di Panduan Pengguna IAM.

Berikan Izin Pengguna Anda untuk Melakukan Peramalan Deret Waktu

Untuk melakukan prakiraan deret waktu di Amazon SageMaker Canvas, pengguna Anda harus memiliki izin yang diperlukan. Metode yang lebih disukai untuk memberikan izin ini kepada pengguna Anda adalah mengaktifkan opsi peramalan deret waktu saat menyiapkan SageMaker Domain Amazon, atau saat mengedit pengaturan untuk profil pengguna tertentu. Anda juga dapat menggunakan metode manual untuk melampirkan kebijakan dan hubungan kepercayaan untuk Amazon Forecast ke peran AWS Identity and Access Management (IAM).

Jika Anda ingin mengenkripsi perkiraan deret waktu dengan kunci Anda sendiri, Anda harus menggunakan kunci dan memodifikasi kebijakan AWS KMS kunci KMS Anda untuk memberikan izin ke peran yang digunakan oleh Amazon Forecast. Untuk informasi selengkapnya tentang menyiapkan kunci KMS Anda dan memodifikasi kebijakan untuk peramalan deret waktu, lihat. Prasyarat untuk peramalan deret waktu

Metode pengaturan domain

SageMaker memberi Anda opsi untuk memberikan izin peramalan deret waktu kepada pengguna melalui pengaturan Domain. Anda dapat mengaktifkan izin untuk semua pengguna di Domain Anda, dan SageMaker mengelola melampirkan kebijakan IAM yang diperlukan dan hubungan kepercayaan untuk Anda.

Jika Anda menyiapkan SageMaker Domain Amazon untuk pertama kalinya dan ingin mengaktifkan izin peramalan deret waktu untuk semua pengguna di Domain, gunakan prosedur berikut.

Quick setup

Gunakan prosedur berikut untuk mengaktifkan izin peramalan deret waktu SageMaker Canvas saat melakukan penyiapan Cepat untuk Domain Anda.

1. Di penyiapan Cepat SageMaker Domain Amazon, isi kolom peran eksekusi Nama dan Default di bagian Profil pengguna.
2. Biarkan opsi Aktifkan izin SageMaker Canvas diaktifkan. Ini dihidupkan secara default.
3. Pilih Kirim untuk menyelesaikan pengaturan Domain Anda.

Standard setup

Gunakan prosedur berikut untuk mengaktifkan izin peramalan deret waktu SageMaker Canvas saat melakukan pengaturan Standar untuk Domain Anda.

1. Di pengaturan Standar SageMaker Domain Amazon, isi halaman pengaturan Umum, pengaturan Studio, dan pengaturan RStudio.
2. Pilih halaman pengaturan Canvas.
3. Untuk konfigurasi izin dasar Canvas, biarkan opsi Aktifkan izin dasar Canvas diaktifkan. Ini dihidupkan secara default. Izin ini diperlukan untuk mengaktifkan izin peramalan deret waktu.
4. Untuk konfigurasi peramalan deret waktu, biarkan opsi Aktifkan peramalan deret waktu diaktifkan. Ini dihidupkan secara default.
5. Pilih Buat dan gunakan peran eksekusi baru, atau pilih Gunakan peran eksekusi yang ada jika Anda sudah memiliki peran IAM dengan izin Amazon Forecast yang diperlukan dilampirkan. Lihat informasi yang lebih lengkap di Metode pengaturan peran IAM.
6. Selesai membuat perubahan lain pada penyiapan Domain Anda, lalu pilih Kirim.

Pengguna Anda sekarang harus memiliki izin yang diperlukan untuk melakukan peramalan deret waktu di SageMaker Canvas.

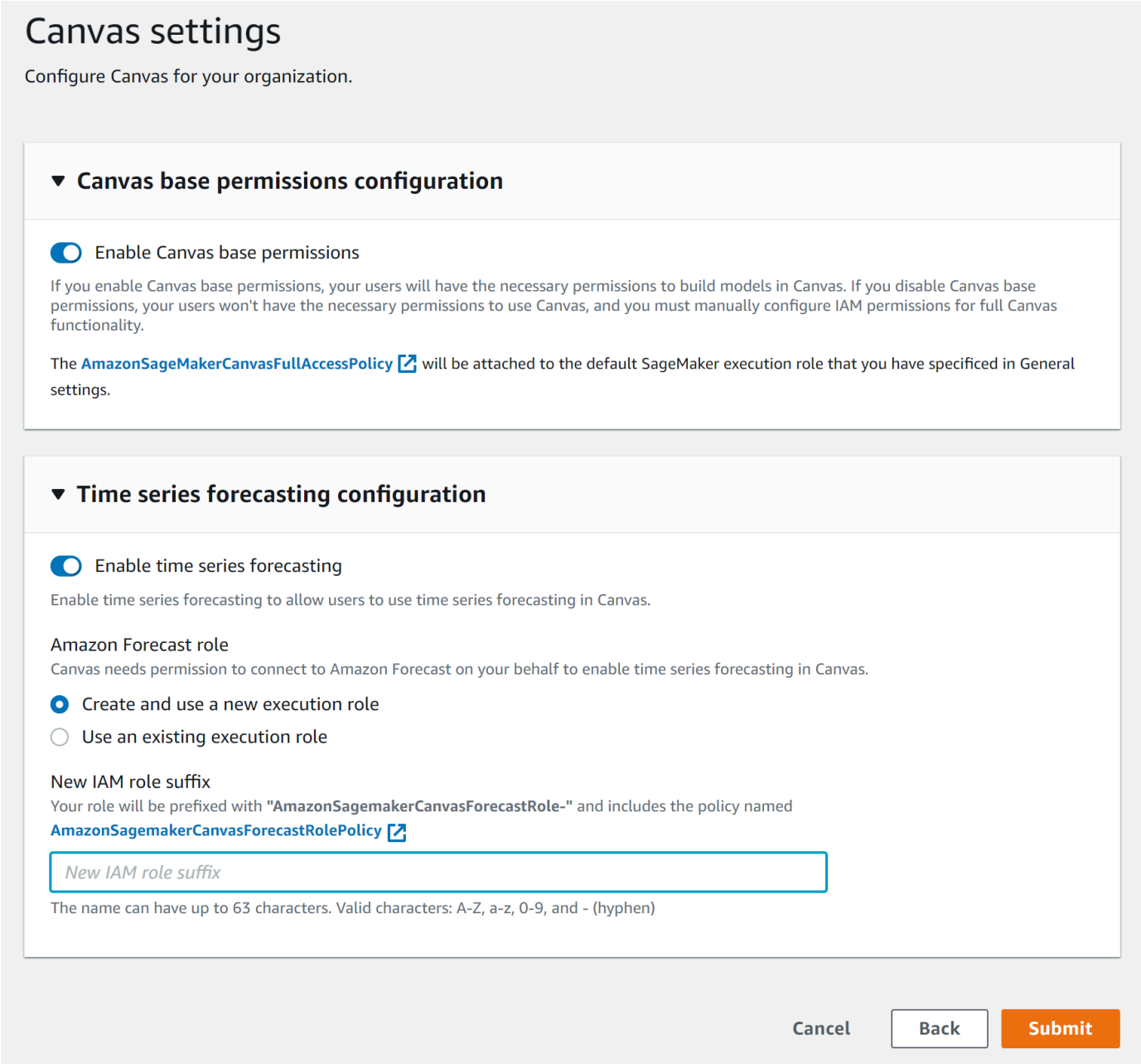
Metode penyiapan pengguna

Anda dapat mengonfigurasi izin peramalan deret waktu untuk pengguna individual di Domain yang ada. Pengaturan profil pengguna mengganti pengaturan Domain umum, sehingga Anda dapat memberikan izin kepada pengguna tertentu tanpa memberikan izin kepada semua pengguna Anda. Untuk memberikan izin peramalan deret waktu kepada pengguna tertentu yang belum memiliki izin, gunakan prosedur berikut.

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pada halaman Domain, pilih Domain Anda.
5. Di tab Profil pengguna, pilih nama pengguna yang izinnya ingin Anda edit.
6. Pada halaman Detail Pengguna, pilih Edit.
7. Pilih halaman pengaturan Canvas.
8. Aktifkan Aktifkan izin dasar Canvas. Izin ini diperlukan untuk mengaktifkan izin peramalan deret waktu.
9. Aktifkan opsi Aktifkan peramalan deret waktu.
10. Jika Anda ingin menggunakan peran eksekusi yang berbeda untuk pengguna daripada peran yang ditentukan dalam Domain, pilih Buat dan gunakan peran eksekusi baru, atau Gunakan peran eksekusi yang ada jika Anda sudah memiliki peran IAM yang siap digunakan.

> **Note**
>
> Jika Anda ingin menggunakan peran IAM yang ada, pastikan bahwa ia memiliki kebijakan IAM yang `AmazonSageMakerCanvasForecastAccess` dilampirkan dan memiliki hubungan kepercayaan yang menetapkan Amazon Forecast sebagai prinsipal layanan. Untuk informasi selengkapnya, lihat bagian Metode pengaturan peran IAM.

11. Halaman pengaturan Canvas akan terlihat seperti tangkapan layar berikut. Selesai membuat perubahan lain pada profil pengguna Anda, lalu pilih Kirim untuk menyimpan perubahan Anda.



Pengguna Anda sekarang harus memiliki izin untuk melakukan peramalan deret waktu di SageMaker Canvas.

Anda juga dapat menghapus izin pengguna Anda dengan menggunakan prosedur sebelumnya dan mematikan opsi Aktifkan peramalan deret waktu.

Metode pengaturan peran IAM

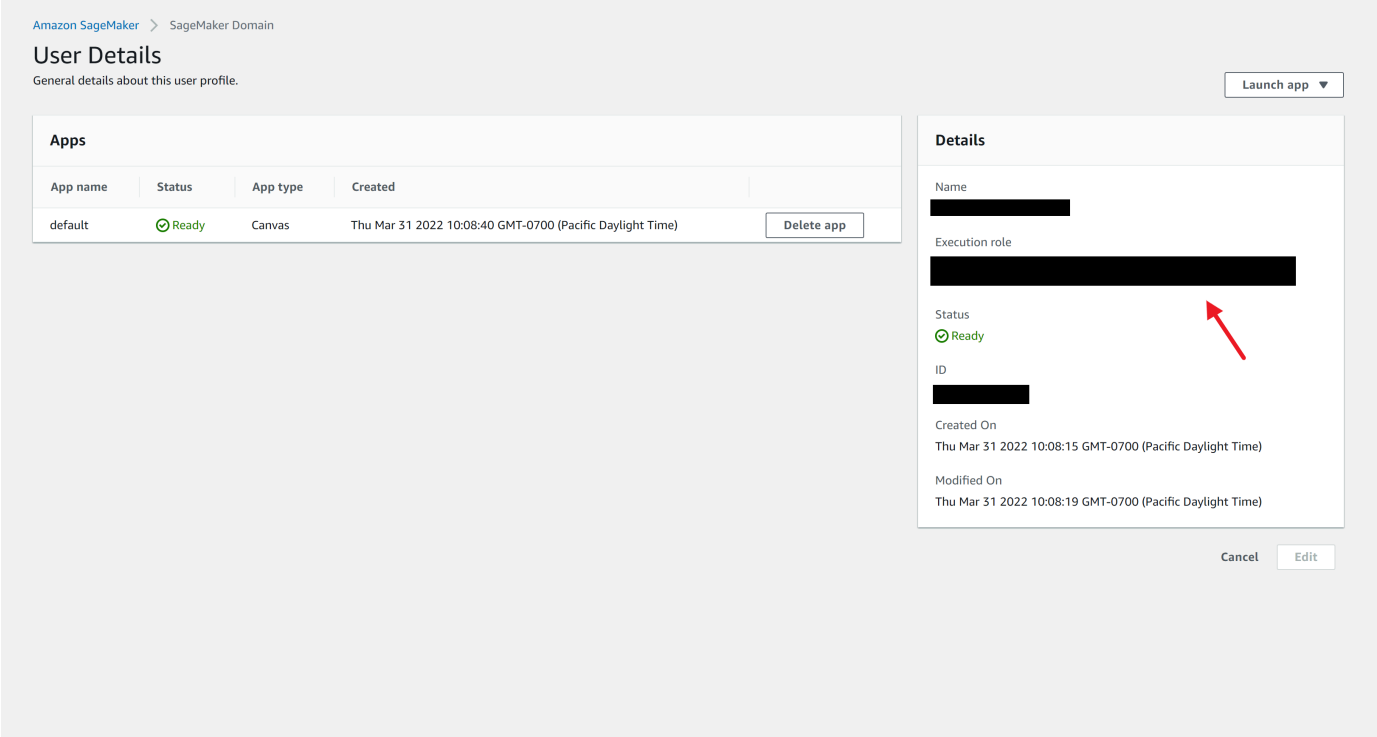
Anda dapat secara manual memberikan izin kepada pengguna untuk melakukan peramalan deret waktu di Amazon SageMaker Canvas dengan menambahkan izin tambahan ke peran AWS Identity and Access Management (IAM) yang ditentukan untuk profil pengguna. Peran IAM harus memiliki hubungan kepercayaan dengan Amazon Forecast dan kebijakan terlampir yang memberikan izin ke Forecast.

Bagian berikut menunjukkan cara membuat hubungan kepercayaan dan melampirkan kebijakan `AmazonSageMakerCanvasForecastAccess` terkelola ke peran IAM Anda, yang memberikan izin minimum yang diperlukan agar peramalan deret waktu berfungsi di Canvas. SageMaker

> **Note**
>
> `AmazonSageMakerCanvasForecastAccess`Kebijakan ini memberikan izin untuk mengakses bucket Amazon SageMaker S3 yang dibuat, yang merupakan lokasi penyimpanan default untuk data aplikasi Canvas. Jika Anda telah menentukan lokasi penyimpanan Amazon S3 khusus untuk data aplikasi Canvas, Anda harus memperbarui izin dalam kebijakan ke bucket Amazon S3 Anda sendiri. Untuk informasi selengkapnya tentang lokasi penyimpanan Amazon S3 khusus untuk Canvas, lihat. Konfigurasikan penyimpanan Amazon S3 Anda

Untuk mengkonfigurasi peran IAM dengan metode manual, gunakan prosedur berikut.

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pada halaman Domain, pilih Domain Anda.
5. Dari daftar profil Pengguna, pilih profil pengguna yang ingin Anda berikan izin peramalan deret waktu.
6. Di bawah Detail, salin atau catat nama peran Eksekusi pengguna. Nama peran IAM harus mirip dengan yang berikut:`111122223333`.



7. Setelah Anda memiliki nama peran IAM pengguna, buka konsol IAM.
8. Pilih Peran.
9. Cari peran IAM pengguna berdasarkan nama dari daftar peran dan pilih.
10. Di bawah Izin, pilih Tambahkan izin.
11. Pilih Lampirkan kebijakan.
12. Cari kebijakan `AmazonSageMakerCanvasForecastAccess` terkelola dan pilih. Pilih Lampirkan kebijakan untuk melampirkan kebijakan ke peran.

    Setelah melampirkan kebijakan, bagian Izin peran sekarang harus disertakan. `AmazonSageMakerCanvasForecastAccess`
13. Kembali ke halaman peran IAM, dan di bawah Hubungan kepercayaan, pilih Edit kebijakan kepercayaan.
14. Di editor kebijakan Edit trust, perbarui kebijakan kepercayaan untuk menambahkan Forecast sebagai prinsipal layanan. Kebijakan akan terlihat seperti contoh berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
        "Service": [
            "sagemaker.amazonaws.com",
            "forecast.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

15. Setelah mengedit kebijakan kepercayaan, pilih Perbarui kebijakan.

Anda sekarang harus memiliki peran IAM yang memiliki kebijakan yang `AmazonSageMakerCanvasForecastAccess` melekat padanya dan hubungan kepercayaan yang dibuat dengan Amazon Forecast, memberikan izin kepada pengguna untuk melakukan peramalan deret waktu di SageMaker Canvas. Untuk informasi tentang kebijakan AWS terkelola, lihat Kebijakan terkelola dan kebijakan sebaris.

> **Note**
>
> Jika Anda menggunakan metode ini untuk mengatur peramalan deret waktu dan ingin menggunakan AWS KMS enkripsi untuk perkiraan Anda, maka Anda harus mengonfigurasi kebijakan kunci KMS Anda untuk memberikan izin tambahan. Untuk informasi selengkapnya, lihat Prasyarat untuk peramalan deret waktu.

Berikan Izin Pengguna untuk Menyempurnakan Model Foundation

Untuk memberikan izin untuk model foundation fine-tuning di Amazon SageMaker Canvas, Anda harus menyelesaikan penyiapan izin yang dijelaskan di halaman ini. Anda harus memberikan izin pengguna untuk eady-to-use model R, yang melampirkan ServicesAccess kebijakan AmazonSageMakerCanvasAI ke peran eksekusi AWS IAM pengguna Anda. Anda juga harus membuat hubungan kepercayaan antara Amazon Bedrock dan peran eksekusi IAM.

Berikan izin eady-to-use model R

Izin yang Anda perlukan untuk model foundation fine-tuning disertakan dalam izin model Canvas R. eady-to-use Anda harus mengaktifkan izin konfigurasi eady-to-use model Canvas R saat menyiapkan SageMaker Domain Amazon Anda. Untuk informasi selengkapnya, lihat Prasyarat untuk menyiapkan Amazon Canvas SageMaker .

Anda dapat mengedit pengaturan Domain atau profil pengguna untuk mengaktifkan pengaturan konfigurasi eady-to-use model Canvas R. Untuk informasi selengkapnya tentang mengedit setelan Domain, lihat Melihat dan Mengedit Domain.

Anda juga dapat melampirkan ServicesAccess kebijakan AmazonSageMakerCanvasAI secara manual ke peran eksekusi IAM pengguna melalui konsol IAM. Untuk informasi selengkapnya tentang melampirkan kebijakan ke peran, lihat Menambahkan dan menghapus izin identitas IAM di Panduan Pengguna AWSIAM.

Buat hubungan kepercayaan dengan Amazon Bedrock

Anda harus membuat hubungan kepercayaan dengan Amazon Bedrock sehingga Amazon Bedrock dapat mengambil peran IAM pengguna Anda sambil menyempurnakan model foundation.

Untuk menambahkan hubungan kepercayaan pada Anda, perhatikan peran IAM Domain atau profil pengguna Anda. Kemudian, lakukan hal berikut:

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi sebelah kiri, pilih Peran.
3. Cari peran IAM pengguna berdasarkan nama dari daftar peran dan pilih.
4. Pilih tab Trust relationship.
5. Pilih Edit kebijakan kepercayaan.
6. Di editor kebijakan, temukan opsi Tambahkan prinsipal di panel kanan dan pilih Tambah.
7. Di kotak dialog, untuk tipe Principal, pilih AWSlayanan.
8. Untuk ARN, masukkan. `bedrock.amazonaws.com`
9. Pilih Tambahkan prinsipal.
10. Pilih Perbarui kebijakan.

Anda sekarang harus memiliki hubungan kepercayaan dengan Amazon Bedrock dan izin yang diperlukan untuk menyempurnakan model fondasi di Canvas. SageMaker

Perbarui SageMaker Canvas untuk Pengguna Anda

Anda dapat memperbarui ke Amazon SageMaker Canvas versi terbaru sebagai pengguna atau administrator TI. Anda dapat memperbarui Amazon SageMaker Canvas untuk satu pengguna sekaligus.

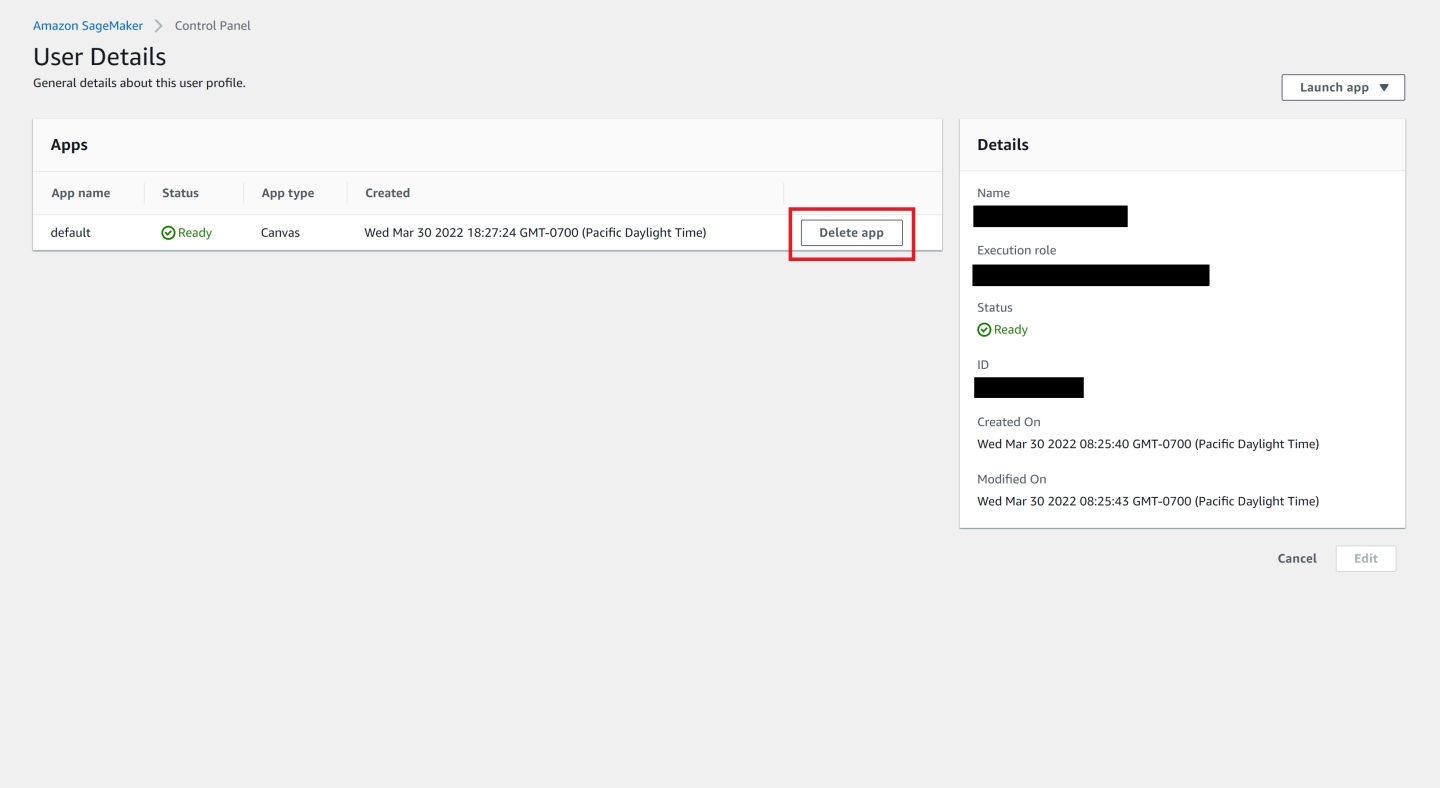
Untuk memperbarui aplikasi Amazon SageMaker Canvas, Anda harus menghapus versi sebelumnya.

> **Important**
>
> Menghapus Amazon SageMaker Canvas versi sebelumnya tidak menghapus data atau model yang telah dibuat pengguna.

Gunakan prosedur berikut untuk masukAWS, membuka SageMaker Domain Amazon, dan memperbarui Amazon SageMaker Canvas. Pengguna dapat mulai menggunakan aplikasi SageMaker Canvas ketika mereka masuk kembali.

1. Masuk ke SageMaker konsol Amazon di Amazon SageMaker Runtime.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pada halaman Domain, pilih Domain Anda.
5. Dari daftar profil Pengguna, pilih profil pengguna.
6. Untuk daftar Apps, cari aplikasi Canvas (tipe App mengatakan Canvas) dan pilih Delete app.
7. Lengkapi kotak dialog dan pilih Konfirmasi tindakan.

Gambar berikut menunjukkan halaman profil pengguna dan menyoroti tindakan Hapus aplikasi dari prosedur sebelumnya.



Anda dapat meminta untuk Memperbanyak Kuota

Pengguna Anda mungkin menggunakan AWS sumber daya dalam jumlah yang melebihi yang ditentukan oleh kuota mereka. Jika pengguna Anda dibatasi sumber daya dan mengalami kesalahan di SageMaker Canvas, Anda dapat meminta peningkatan kuota untuk mereka.

Untuk detail selengkapnya tentang SageMaker kuota dan cara meminta kenaikan kuota, lihat Kuota.

Amazon SageMaker Canvas menggunakan layanan berikut untuk memproses permintaan pengguna Anda:

- SageMaker Autopilot Amazon
- Domain Klasik Amazon SageMaker Studio
- Amazon Forecast

Untuk daftar kuota yang tersedia untuk operasi SageMaker Canvas yang tidak digunakan untuk memperkirakan data deret waktu, lihat SageMaker titik akhir dan kuota Amazon.

Untuk daftar kuota yang tersedia untuk operasi SageMaker Canvas yang digunakan untuk memperkirakan data deret waktu, lihat titik akhir dan kuota Amazon Forecast.

Minta peningkatan instance untuk membuat model kustom

Saat membuat model kustom, jika Anda mengalami kesalahan selama analisis pasca-pembangunan yang memberi tahu Anda untuk meningkatkan kuota untuk `ml.m5.2xlarge` instance, gunakan informasi berikut untuk menyelesaikan masalah.

Anda harus meningkatkan kuota titik akhir SageMaker Hosting untuk jenis `ml.m5.2xlarge` instans menjadi nilai bukan nol di akun Anda. AWS Setelah membangun model, SageMaker Canvas menghosting model pada titik akhir SageMaker Hosting dan menggunakan titik akhir untuk menghasilkan analisis pasca-bangunan. Jika Anda tidak meningkatkan kuota akun default 0 untuk `ml.m5.2xlarge` instance, SageMaker Canvas tidak dapat menyelesaikan langkah ini dan menghasilkan kesalahan selama analisis pasca-pembangunan.

Untuk prosedur peningkatan kuota, lihat Meminta kenaikan kuota pada Panduan Pengguna Service Quotas.

Berikan Izin Pengguna untuk Mengimpor Data Amazon Redshift

Pengguna Anda mungkin memiliki kumpulan data yang disimpan di Amazon Redshift. Sebelum pengguna dapat mengimpor data dari Amazon Redshift ke SageMaker Canvas, Anda harus menambahkan kebijakan `AmazonRedshiftFullAccess` terkelola ke peran eksekusi IAM yang telah Anda gunakan untuk profil pengguna dan menambahkan Amazon Redshift sebagai prinsipal layanan ke kebijakan kepercayaan peran. Anda juga harus mengaitkan peran eksekusi IAM dengan klaster Amazon Redshift Anda. Selesaikan prosedur di bagian berikut untuk memberi pengguna izin yang diperlukan untuk mengimpor data Amazon Redshift.

Tambahkan izin Amazon Redshift ke peran IAM Anda

Anda harus memberikan izin Amazon Redshift ke peran IAM yang ditentukan dalam profil pengguna Anda.

Untuk menambahkan `AmazonRedshiftFullAccess` kebijakan ke peran IAM pengguna, lakukan hal berikut.

1. Masuk ke konsol IAM dan di https://console.aws.amazon.com/iam/.
2. Pilih Peran.
3. Di kotak pencarian, cari peran IAM pengguna berdasarkan nama dan pilih.
4. Pada halaman untuk peran pengguna, di bawah Izin, pilih Tambahkan izin.
5. Pilih Lampirkan kebijakan.

6. Cari kebijakan `AmazonRedshiftFullAccess` terkelola dan pilih.
7. Pilih Lampirkan kebijakan untuk melampirkan kebijakan ke peran.

Setelah melampirkan kebijakan, bagian Izin peran sekarang harus disertakan. `AmazonRedshiftFullAccess`

Untuk menambahkan Amazon Redshift sebagai prinsipal layanan ke peran IAM, lakukan hal berikut.

1. Pada halaman yang sama untuk peran IAM, di bawah Hubungan kepercayaan, pilih Edit kebijakan kepercayaan.
2. Di editor kebijakan Edit trust, perbarui kebijakan kepercayaan untuk menambahkan Amazon Redshift sebagai prinsipal layanan. Peran IAM yang memungkinkan Amazon Redshift mengakses layanan AWS lain atas nama Anda memiliki hubungan kepercayaan sebagai berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

3. Setelah mengedit kebijakan kepercayaan, pilih Perbarui kebijakan.

Anda sekarang harus memiliki peran IAM yang memiliki kebijakan yang `AmazonRedshiftFullAccess` melekat padanya dan hubungan kepercayaan yang dibuat dengan Amazon Redshift, memberikan izin kepada pengguna untuk mengimpor data Amazon Redshift ke Canvas. SageMaker Untuk informasi selengkapnya tentang kebijakan AWS terkelola, lihat Kebijakan terkelola dan kebijakan sebaris di Panduan Pengguna IAM.

Kaitkan peran IAM dengan klaster Amazon Redshift Anda

Dalam pengaturan untuk klaster Amazon Redshift, Anda harus mengaitkan peran IAM yang Anda berikan izin di bagian sebelumnya.

Untuk mengaitkan peran IAM dengan cluster Anda, lakukan hal berikut.

1. Masuk ke konsol Amazon Redshift di https://console.aws.amazon.com/redshift/.
2. Pada menu navigasi, pilih Cluster, lalu pilih nama cluster yang ingin Anda perbarui.
3. Di menu tarik-turun Tindakan, pilih Kelola peran IAM. Halaman izin Cluster muncul.
4. Untuk peran IAM yang Tersedia, masukkan ARN atau nama peran IAM, atau pilih peran IAM dari daftar.
5. Pilih peran IAM Associate untuk menambahkannya ke daftar peran IAM Terkait.
6. Pilih Simpan perubahan untuk mengaitkan peran IAM dengan cluster.

Amazon Redshift memodifikasi cluster untuk menyelesaikan perubahan, dan peran IAM yang sebelumnya Anda berikan izin Amazon Redshift sekarang dikaitkan dengan cluster Amazon Redshift Anda. Pengguna Anda sekarang memiliki izin yang diperlukan untuk mengimpor data SageMaker Amazon Redshift ke Canvas.

Berikan Izin Pengguna untuk Berkolaborasi dengan Studio Classic

> **Note**
>
> Fungsionalitas yang dijelaskan di halaman ini hanya berlaku untuk Amazon SageMaker Studio Classic. Saat ini, Anda hanya dapat berbagi model ke Canvas (atau melihat model Canvas bersama) di Studio Classic. Jika saat ini Anda menggunakan Studio versi terbaru, Anda harus menjalankan Studio Classic dari dalam versi terbaru Studio untuk berbagi model ke Canvas atau melihat model yang dibagikan dari Canvas. Untuk informasi selengkapnya tentang mengakses Studio Classic, lihat dokumentasi Studio Classic.

Pengguna Amazon SageMaker Canvas Anda mungkin ingin berbagi model mereka dengan pengguna di Amazon SageMaker Studio Classic untuk menerima umpan balik dan pembaruan model, dan pengguna Studio Classic mungkin ingin berbagi model dengan pengguna Canvas sehingga mereka dapat menghasilkan prediksi di Canvas. Izin berikut memberi pengguna Canvas dan pengguna Studio Classic akses untuk berbagi model satu sama lain.

Untuk informasi selengkapnya tentang cara pengguna Canvas dapat berbagi model dengan pengguna Studio Classic, lihatBerkolaborasi dengan ilmuwan data. Untuk informasi selengkapnya tentang cara pengguna Canvas dapat membawa model yang dibagikan dari Studio Classic, lihatBawa model Anda sendiri ke SageMaker Canvas.

Sebelum pengguna Canvas dan Studio Classic dapat berkolaborasi, pengguna harus berada di SageMaker Domain Amazon yang sama. Tambahkan izin IAM berikut yang ditambahkan ke peran eksekusi IAM yang sama dengan yang Anda gunakan untuk profil mereka.

Untuk menambahkan izin ke peran IAM pengguna, lakukan hal berikut:

1. Buka Konsol IAM.
2. Pilih Peran.
3. Di kotak pencarian, cari peran IAM pengguna berdasarkan nama dan pilih.
4. Pada halaman untuk peran pengguna, di bawah Izin, pilih Tambahkan izin.
5. Pilih Buat kebijakan sebaris.
6. Di editor Kebijakan, pilih JSON dan masukkan kebijakan IAM berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateSharedModel",
                "sagemaker:DescribeSharedModel",
                "sagemaker:ListSharedModelEvents",
                "sagemaker:ListSharedModels",
                "sagemaker:ListSharedModelVersions",
                "sagemaker:SendSharedModelEvent",
                "sagemaker:UpdateSharedModel"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Pilih Berikutnya.
8. Masukkan nama untuk kebijakan di bidang Nama kebijakan.
9. Pilih Buat kebijakan untuk membuat kebijakan dan melampirkannya ke peran.

Untuk informasi selengkapnya tentang kebijakan AWS terkelola, lihat Kebijakan terkelola dan kebijakan sebaris di Panduan Pengguna IAM.

Berikan Izin Pengguna Anda untuk Mengirim Prediksi ke Amazon QuickSight

Anda harus memberikan izin kepada pengguna SageMaker Canvas untuk mengirim prediksi batch ke Amazon. QuickSight Di Amazon QuickSight, pengguna dapat membuat analisis dan laporan dengan kumpulan data dan menyiapkan dasbor untuk membagikan hasilnya. Untuk informasi selengkapnya tentang mengirim prediksi QuickSight untuk analisis, lihatKirim prediksi ke Amazon QuickSight.

Untuk memberikan izin yang diperlukan untuk membagikan prediksi batch dengan pengguna QuickSight, Anda harus menambahkan kebijakan izin ke peran eksekusi AWS Identity and Access Management (IAM) yang telah Anda gunakan untuk profil pengguna. Bagian berikut menunjukkan cara melampirkan kebijakan izin paling sedikit ke peran Anda.

Menambahkan kebijakan izin ke peran IAM Anda

Untuk menambahkan kebijakan izin, gunakan prosedur berikut:

1. Masuk ke konsol IAM dan di https://console.aws.amazon.com/iam/.
2. Pilih Peran.
3. Di kotak pencarian, cari peran IAM pengguna berdasarkan nama dan pilih.
4. Pada halaman untuk peran pengguna, di bawah Izin, pilih Tambahkan izin.
5. Pilih Buat kebijakan sebaris.
6. Pilih tab JSON, lalu tempelkan kebijakan izin terkecil berikut ke editor. Ganti placeholder *<your-account-number>* dengan nomor AWS akun Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "quicksight:CreateDataSet",
                "quicksight:ListUsers",
                "quicksight:ListNamespaces",
                "quicksight:CreateDataSource",
                "quicksight:PassDataSet",
                "quicksight:PassDataSource"
            ],
            "Resource":[
                "arn:aws:quicksight:*:<your-account-number>:datasource/*",
                "arn:aws:quicksight:*:<your-account-number>:user/*",
```

```
                "arn:aws:quicksight:*:<your-account-number>:namespace/*",
                "arn:aws:quicksight:*:<your-account-number>:dataset/*"
            ]
        }
    ]
}
```

7. Pilih Tinjau kebijakan.
8. Masukkan Nama untuk kebijakan tersebut.
9. Pilih Buat kebijakan.

Anda sekarang harus memiliki kebijakan IAM yang dikelola pelanggan yang dilampirkan ke peran eksekusi Anda yang memberi pengguna Canvas izin yang diperlukan untuk mengirim prediksi batch ke pengguna. QuickSight

Kelola aplikasi

Bagian berikut menjelaskan bagaimana Anda dapat mengelola aplikasi SageMaker Canvas Anda. Anda dapat melihat, menghapus, atau meluncurkan kembali aplikasi Anda dari bagian Domain konsol. SageMaker

Periksa aplikasi yang aktif

Untuk memeriksa apakah Anda memiliki aplikasi SageMaker Canvas yang aktif berjalan, gunakan prosedur berikut.

1. Buka konsol SageMaker .
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pada halaman Domain, pilih Domain Anda.
5. Pada halaman detail Domain, di bawah Profil pengguna, pilih nama profil pengguna untuk aplikasi Canvas yang ingin Anda lihat.
6. Di bawah Aplikasi, temukan aplikasi yang bertuliskan Canvas di kolom Jenis aplikasi.

Kolom Status menampilkan status aplikasi, seperti Siap, Tertunda, atau Dihapus. Jika aplikasi sudah Siap, maka instance ruang kerja SageMaker Canvas Anda aktif. Anda dapat menghapus aplikasi dari konsol atau keluar dari antarmuka SageMaker Canvas.

Hapus aplikasi

Jika Anda ingin mengakhiri instance ruang kerja SageMaker Canvas Anda, Anda dapat keluar dari aplikasi SageMaker Canvas atau menghapus aplikasi Anda dari SageMaker konsol. Instans ruang kerja didedikasikan untuk penggunaan Anda dari saat Anda mulai menggunakan SageMaker Canvas ke titik ketika Anda berhenti menggunakannya. Menghapus aplikasi hanya mengakhiri instance ruang kerja. Model dan kumpulan data tidak terpengaruh, tetapi tugas pembuatan cepat otomatis dimulai ulang saat Anda masuk lagi. Penagihan untuk instance ruang kerja juga berhenti.

Untuk menghapus aplikasi Canvas Anda melalui AWS konsol, pertama-tama tutup tab browser tempat aplikasi Canvas Anda terbuka. Kemudian, gunakan prosedur berikut untuk menghapus aplikasi SageMaker Canvas Anda.

1. Buka konsol SageMaker .
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pada halaman Domain, pilih Domain Anda.
5. Pada halaman Detail Domain, di bawah Profil pengguna, pilih nama profil pengguna untuk aplikasi Canvas yang ingin Anda lihat.
6. Di bawah Aplikasi, temukan aplikasi yang mengatakan Canvas di kolom Jenis aplikasi.
7. Di kolom Tindakan, pilih Hapus aplikasi.
8. Di kotak dialog Hapus aplikasi, pilih perintah Ya, hapus aplikasi, konfirmasikan penghapusan dengan mengetikkan **`delete`** bidang teks, lalu pilih Hapus.

Setelah Anda berhasil menghapus aplikasi, kolom Status mengatakan Dihapus. Jika tidak, aplikasi Anda masih aktif.

Anda juga dapat mengakhiri instance ruang kerja dengan keluar dari dalam aplikasi SageMaker Canvas.

Luncurkan kembali aplikasi

Jika Anda menghapus atau keluar dari aplikasi SageMaker Canvas Anda dan ingin meluncurkan kembali aplikasi, gunakan prosedur berikut.

1. Navigasikan ke konsol SageMaker tersebut.
2. Di panel navigasi, pilih Canvas.

3. Pada halaman landing SageMaker Canvas, di kotak Memulai, pilih profil pengguna Anda dari dropdown.
4. Pilih Open Canvas untuk membuka aplikasi.

SageMaker Canvas mulai meluncurkan aplikasi.

Anda juga dapat menggunakan prosedur sekunder berikut jika Anda mengalami masalah dengan prosedur sebelumnya.

1. Buka konsol SageMaker .
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pada halaman Domain, pilih Domain Anda.
5. Pada halaman Detail Domain, di bawah Profil pengguna, pilih nama profil pengguna untuk aplikasi SageMaker Canvas yang ingin Anda lihat.
6. Pilih Luncurkan dan pilih Canvas dari daftar dropdown.

SageMaker Canvas mulai meluncurkan aplikasi.

Konfigurasikan Amazon SageMaker Canvas di VPC tanpa akses internet

Aplikasi Amazon SageMaker Canvas berjalan dalam wadah di Amazon Virtual Private Cloud (VPC) yang AWS dikelola. Jika Anda ingin mengontrol akses lebih lanjut ke sumber daya Anda atau menjalankan SageMaker Canvas tanpa akses internet publik, Anda dapat mengonfigurasi pengaturan SageMaker Domain dan VPC Amazon Anda. Dalam VPC Anda sendiri, Anda dapat mengonfigurasi pengaturan seperti grup keamanan (firewall virtual yang mengontrol lalu lintas masuk dan keluar dari instans Amazon EC2) dan subnet (rentang alamat IP di VPC Anda). Untuk mempelajari lebih lanjut tentang VPC, lihat Cara kerja Amazon VPC.

Ketika aplikasi SageMaker Canvas berjalan di VPC AWS terkelola, ia dapat berinteraksi dengan AWS layanan lain menggunakan koneksi internet atau melalui titik akhir VPC yang dibuat dalam VPC yang dikelola pelanggan (tanpa akses internet publik). SageMaker Aplikasi Canvas dapat mengakses titik akhir VPC ini melalui antarmuka jaringan yang dibuat Studio Classic yang menyediakan konektivitas ke VPC yang dikelola pelanggan. Perilaku default aplikasi SageMaker Canvas adalah memiliki akses internet. Saat menggunakan koneksi internet, wadah untuk pekerjaan sebelumnya mengakses AWS sumber daya melalui internet, seperti bucket Amazon S3 tempat Anda menyimpan data pelatihan dan artefak model.

Namun, jika Anda memiliki persyaratan keamanan untuk mengontrol akses ke data dan wadah pekerjaan Anda, kami sarankan Anda mengonfigurasi SageMaker Canvas dan VPC Anda sehingga data dan container Anda tidak dapat diakses melalui internet. SageMaker menggunakan pengaturan konfigurasi VPC yang Anda tentukan saat mengatur Domain Anda untuk SageMaker Kanvas.

Jika Anda ingin mengonfigurasi aplikasi SageMaker Canvas Anda tanpa akses internet, Anda harus mengonfigurasi pengaturan VPC saat Anda onboard ke SageMaker Domain Amazon, mengatur titik akhir VPC, dan memberikan izin yang diperlukan. AWS Identity and Access Management Untuk informasi tentang mengonfigurasi VPC di SageMaker Amazon, lihat. Pilih VPC Amazon Bagian berikut menjelaskan cara menjalankan SageMaker Canvas di VPC tanpa akses internet publik.

Konfigurasikan Amazon SageMaker Canvas di VPC tanpa akses internet

Anda dapat mengirim lalu lintas dari SageMaker Canvas ke AWS layanan lain melalui VPC Anda sendiri. Jika VPC Anda sendiri tidak memiliki akses internet publik dan Anda telah mengatur Domain Anda dalam mode VPC saja, maka SageMaker Canvas tidak akan memiliki akses internet publik juga. Ini mencakup semua permintaan, seperti mengakses kumpulan data di Amazon S3 atau pekerjaan pelatihan untuk build standar, dan permintaan melalui titik akhir VPC di VPC Anda, bukan internet publik. Saat Anda onboard ke Domain danPilih VPC Amazon, Anda dapat menentukan VPC Anda sendiri sebagai VPC default untuk Domain, bersama dengan grup keamanan dan pengaturan subnet yang Anda inginkan. Kemudian, SageMaker buat antarmuka jaringan di VPC Anda yang digunakan SageMaker Canvas untuk mengakses titik akhir VPC di VPC Anda. Perhatikan bahwa grup keamanan dan pengaturan subnet disetel setelah Anda selesai melakukan onboarding ke Domain.

Saat melakukan onboarding ke Domain, jika Anda memilih Internet Publik hanya sebagai jenis akses jaringan, VPC SageMaker dikelola dan memungkinkan akses internet.

Anda dapat mengubah perilaku ini dengan memilih VPC saja sehingga SageMaker mengirimkan semua lalu lintas ke antarmuka jaringan yang SageMaker dibuat di VPC yang Anda tentukan. Ketika Anda memilih opsi ini, Anda harus menyediakan subnet, grup keamanan, dan titik akhir VPC yang diperlukan untuk berkomunikasi dengan SageMaker API SageMaker dan Runtime, dan AWS berbagai layanan, seperti Amazon S3 dan CloudWatch Amazon, yang digunakan oleh Canvas. SageMaker Perhatikan bahwa Anda hanya dapat mengimpor data dari bucket Amazon S3 yang terletak di Wilayah yang sama dengan VPC Anda.

Prosedur berikut menunjukkan bagaimana Anda dapat mengonfigurasi pengaturan ini untuk menggunakan SageMaker Canvas tanpa internet.

Langkah 1: Onboard ke Domain Amazon SageMaker

Untuk mengirim lalu lintas SageMaker Canvas ke antarmuka jaringan di VPC Anda sendiri, bukan melalui internet, tentukan VPC yang ingin Anda gunakan saat melakukan onboarding ke Domain Amazon. SageMaker Anda juga harus menentukan setidaknya dua subnet di VPC Anda SageMaker yang dapat digunakan. Pilih Pengaturan standar dan lakukan prosedur berikut saat mengonfigurasi Bagian Jaringan dan Penyimpanan untuk Domain.

1. Pilih VPC yang Anda inginkan.
2. Pilih dua atau lebih Subnet. Jika Anda tidak menentukan subnet, SageMaker gunakan semua subnet di VPC.
3. Pilih satu atau beberapa grup Keamanan.
4. Pilih VPC Only untuk mematikan akses internet langsung di AWS VPC terkelola tempat SageMaker Canvas di-host.

Setelah menonaktifkan akses internet, selesaikan proses orientasi untuk mengatur Domain Anda. Untuk informasi selengkapnya tentang setelan VPC untuk SageMaker Domain Amazon, lihat. Pilih VPC Amazon

Langkah 2: Konfigurasikan titik akhir dan akses VPC

> **Note**
>
> Untuk mengonfigurasi Canvas di VPC Anda sendiri, Anda harus mengaktifkan nama host DNS pribadi untuk titik akhir VPC Anda. Untuk informasi selengkapnya, lihat Connect to SageMaker Through a VPC Interface Endpoint.

SageMaker Canvas hanya mengakses AWS layanan lain untuk mengelola dan menyimpan data untuk fungsinya. Misalnya, terhubung ke Amazon Redshift jika pengguna Anda mengakses database Amazon Redshift. Itu dapat terhubung ke AWS layanan seperti Amazon Redshift menggunakan koneksi internet atau titik akhir VPC. Gunakan titik akhir VPC jika Anda ingin mengatur koneksi dari VPC ke AWS layanan yang tidak menggunakan internet publik.

Endpoint VPC menciptakan koneksi pribadi ke AWS layanan yang menggunakan jalur jaringan yang terisolasi dari internet publik. Misalnya, jika Anda mengatur akses ke Amazon S3 menggunakan titik akhir VPC dari VPC Anda sendiri, maka aplikasi Canvas SageMaker dapat mengakses Amazon

S3 dengan melalui antarmuka jaringan di VPC Anda dan kemudian melalui titik akhir VPC yang terhubung ke Amazon S3. Komunikasi antara SageMaker Canvas dan Amazon S3 bersifat pribadi.

Untuk informasi selengkapnya tentang mengonfigurasi titik akhir VPC untuk VPC Anda, lihat. AWS PrivateLink Jika Anda menggunakan model Amazon Bedrock di Canvas dengan VPC, untuk informasi selengkapnya tentang mengontrol akses ke data Anda, lihat Melindungi pekerjaan menggunakan VPC di Panduan Pengguna Amazon Bedrock.

Berikut ini adalah titik akhir VPC untuk setiap layanan yang dapat Anda gunakan dengan Canvas: SageMaker

Layanan	Titik Akhir	Jenis titik akhir
AWSApplication Auto Scaling	com.amazonaws. *Region .application-autoscaling*	Antarmuka
Amazon Athena	com.amazonaws. *Wilayah .athena*	Antarmuka
Amazon SageMaker	com.amazonaws. *Wilayah .sagemaker.api* com.amazonaws. *Wilayah .sagemaker.runtime* com.amazonaws. *Wilayah.notebook*	Antarmuka
AWS Security Token Service	com.amazonaws. *Wilayah* .sts	Antarmuka
Amazon Elastic Container Registry (Amazon ECR)	com.amazonaws. *Wilayah .ecr.api* com.amazonaws. *Daerah .ecr.dkr*	Antarmuka

Layanan	Titik Akhir	Jenis titik akhir
Amazon Elastic Compute Cloud (Amazon EC2)	com.amazonaws. *Wilayah .ec2*	Antarmuka
Amazon Simple Storage Service (Amazon S3)	com.amazonaws. *Wilayah* .s3	Gateway
Amazon Redshift	com.amazonaws. *Wilayah .redshift-data*	Antarmuka
AWS Secrets Manager	com.amazonaws. *Region .secretsmanager*	Antarmuka
AWS Systems Manager	com.amazonaws. *Wilayah .ssm*	Antarmuka
Amazon CloudWatch	com.amazonaws. *Wilayah.monitoring*	Antarmuka
CloudWatch Log Amazon	com.amazonaws. *Wilayah.logs*	Antarmuka
Amazon Forecast	com.amazonaws. *Wilayah* .forecast com.amazonaws. *Wilayah .forecastquery*	Antarmuka
Amazon Textract	com.amazonaws. *Wilayah .textract*	Antarmuka
Amazon Comprehend	com.amazonaws. *Wilayah .comprehend*	Antarmuka

Layanan	Titik Akhir	Jenis titik akhir
Amazon Rekognition	com.amazonaws. *Region .rekognition*	Antarmuka
AWS Glue	com.amazonaws. *Wilayah* .lem	Antarmuka
AWSApplication Auto Scaling	com.amazonaws. *Region .application-autoscaling*	Antarmuka
Amazon Relational Database Service (Amazon RDS)	com.amazonaws. *Wilayah* .rds	Antarmuka
Amazon Bedrock	com.amazonaws. *Wilayah .bedrock-runtime*	Antarmuka
Amazon Kendra	com.amazonaws. *Wilayah .kendra*	Antarmuka

> **Note**
>
> Untuk Amazon Bedrock, nama layanan titik akhir antarmuka tidak digunakan `com.amazonaws.Region.bedrock` lagi. Buat titik akhir VPC baru dengan nama layanan yang tercantum dalam tabel sebelumnya.
> Selain itu, Anda tidak dapat menyempurnakan model foundation dari Canvas VPC tanpa akses internet. Ini karena Amazon Bedrock tidak mendukung titik akhir VPC untuk API kustomisasi model. Untuk mempelajari lebih lanjut tentang fine-tuning model foundation di Canvas, lihat. Model pondasi yang menyempurnakan

Anda juga harus menambahkan kebijakan titik akhir untuk Amazon S3 untuk AWS mengontrol akses utama ke titik akhir VPC Anda. Untuk informasi tentang cara memperbarui kebijakan titik akhir VPC Anda, lihat Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir.

Berikut ini adalah dua kebijakan titik akhir VPC yang dapat Anda gunakan. Gunakan kebijakan pertama jika Anda hanya ingin memberikan akses ke fungsionalitas dasar Canvas, seperti mengimpor data dan membuat model. Gunakan kebijakan kedua jika Anda ingin memberikan akses ke fitur AI geneneratif tambahan di Canvas.

Basic VPC endpoint policy

Kebijakan berikut memberikan akses yang diperlukan ke titik akhir VPC Anda untuk operasi dasar di Canvas.

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:CreateBucket",
                "s3:GetBucketCors",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
```

Generative AI VPC endpoint policy

Kebijakan berikut memberikan akses yang diperlukan ke titik akhir VPC Anda untuk operasi dasar di Canvas, serta menggunakan model dasar AI generatif.

```
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:CreateBucket",
                "s3:GetBucketCors",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*",
                "arn:aws:s3:::*fmeval/datasets*",
                "arn:aws:s3:::*jumpstart-cache-prod*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
```

Langkah 3: Berikan izin IAM

Pengguna SageMaker Canvas harus memiliki AWS Identity and Access Management izin yang diperlukan untuk memungkinkan koneksi ke titik akhir VPC. Peran IAM yang Anda berikan izin harus sama dengan yang Anda gunakan saat orientasi ke Domain Amazon. SageMaker Anda dapat melampirkan `AmazonSageMakerFullAccess` kebijakan SageMaker terkelola ke peran IAM agar pengguna dapat memberikan izin yang diperlukan kepada pengguna. Jika Anda memerlukan izin IAM yang lebih ketat dan menggunakan kebijakan khusus sebagai gantinya, berikan izin kepada peran pengguna. `ec2:DescribeVpcEndpointServices` SageMaker Canvas memerlukan izin ini untuk memverifikasi keberadaan titik akhir VPC yang diperlukan untuk pekerjaan build standar. Jika mendeteksi titik akhir VPC ini, maka pekerjaan build standar dijalankan secara default di VPC Anda. Jika tidak, mereka akan berjalan di VPC AWS terkelola default.

Untuk petunjuk tentang cara melampirkan kebijakan `AmazonSageMakerFullAccess` IAM ke peran IAM pengguna Anda, lihat Menambahkan dan menghapus izin identitas IAM.

Untuk memberikan `ec2:DescribeVpcEndpointServices` izin terperinci kepada peran IAM pengguna Anda, gunakan prosedur berikut.

1. Masuk ke AWS Management Console dan buka konsol IAM.
2. Di panel navigasi, silakan pilih Peran.
3. Dalam daftar, pilih nama peran yang ingin Anda berikan izin.
4. Pilih tab Izin.
5. Pilih Tambahkan izin, lalu pilih Buat kebijakan sebaris.
6. Pilih tab JSON dan masukkan kebijakan berikut, yang memberikan izin: `ec2:DescribeVpcEndpointServices`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ec2:DescribeVpcEndpointServices",
            "Resource": "*"
        }
    ]
}
```

7. Pilih Kebijakan tinjauan, lalu masukkan Nama untuk kebijakan (misalnya,`VPCEndpointPermissions`).
8. Pilih Buat kebijakan.

Peran IAM pengguna sekarang harus memiliki izin untuk mengakses titik akhir VPC yang dikonfigurasi di VPC Anda.

(Opsional) Langkah 4: Ganti pengaturan grup keamanan untuk pengguna tertentu

Jika Anda seorang administrator, Anda mungkin ingin pengguna yang berbeda memiliki pengaturan VPC yang berbeda, atau pengaturan VPC khusus pengguna. Saat Anda mengganti pengaturan grup keamanan VPC default untuk pengguna tertentu, pengaturan ini diteruskan ke aplikasi SageMaker Canvas untuk pengguna tersebut.

Anda dapat mengganti grup keamanan yang dapat diakses pengguna tertentu di VPC saat menyiapkan profil pengguna baru di Studio Classic. Anda dapat menggunakan panggilan

CreateUserProfile SageMaker API (atau create_user_profile dengan AWS CLI), dan kemudian di`UserSettings`, Anda dapat menentukan untuk pengguna. `SecurityGroups`

Siapkan koneksi ke sumber data dengan OAuth

Bagian berikut menjelaskan langkah-langkah yang harus Anda ambil untuk mengatur koneksi OAuth ke sumber data dari SageMaker Canvas. OAuth adalah platform otentikasi umum untuk memberikan akses ke sumber daya tanpa berbagi kata sandi. Dengan OAuth, Anda dapat dengan cepat terhubung ke data Anda dari Canvas dan mengimpornya untuk membangun model. Canvas saat ini mendukung OAuth untuk Snowflake dan Salesforce Data Cloud.

> **Note**
>
> Anda hanya dapat membuat satu koneksi OAuth untuk setiap sumber data.

Siapkan OAuth untuk Salesforce Data Cloud

Untuk menyiapkan OAuth untuk Salesforce Data Cloud, ikuti langkah-langkah umum berikut:

1. Masuk ke Salesforce Data Cloud.
2. Di Salesforce Data Cloud, buat koneksi aplikasi baru dan lakukan hal berikut:
    a. Aktifkan pengaturan OAuth.
    b. Saat diminta untuk URL callback (atau URL sumber daya yang mengakses data Anda), tentukan URL untuk aplikasi Canvas Anda. URL aplikasi Canvas mengikuti format ini: `https://<domain-id>.studio.<region>.sagemaker.aws/canvas/default`
    c. Salin kunci dan rahasia konsumen.
    d. Salin URL otorisasi dan URL token Anda.

Untuk petunjuk selengkapnya tentang melakukan tugas sebelumnya di Salesforce Data Cloud, lihat Impor data dari Salesforce Data Cloud di dokumentasi Data Wrangler untuk mengimpor data dari Salesforce Data Cloud.

Setelah mengaktifkan akses dari Salesforce Data Cloud dan mendapatkan informasi koneksi Anda, Anda harus membuat AWS Secrets Managerrahasia untuk menyimpan informasi dan menambahkannya ke SageMaker Domain Amazon atau profil pengguna Anda. Perhatikan bahwa

Anda dapat menambahkan rahasia ke Domain dan profil pengguna, tetapi Canvas mencari rahasia di profil pengguna terlebih dahulu.

Untuk menambahkan rahasia ke Domain atau profil pengguna Anda, lakukan hal berikut:

1. Pergi ke SageMaker konsol Amazon.
2. Pilih Domain di panel navigasi.
3. Dari daftar Domain, pilih Domain Anda.
    a. Jika menambahkan rahasia Anda ke Domain Anda, lakukan hal berikut:
        i. Pilih Domain.
        ii. Pada halaman Pengaturan domain, pilih tab Pengaturan domain.
        iii. Pilih Edit.
    b. Jika menambahkan rahasia ke profil pengguna Anda, lakukan hal berikut:
        i. Pilih domain pengguna.
        ii. Pada halaman Pengaturan domain, pilih profil pengguna.
        iii. Pada halaman Detail Pengguna, pilih Edit.
4. Di panel navigasi, pilih Pengaturan kanvas.
5. Untuk pengaturan OAuth, pilih Tambahkan konfigurasi OAuth.
6. Untuk sumber data, pilih Salesforce Data Cloud.
7. Untuk Pengaturan Rahasia, pilih Buat rahasia baru. Atau, jika Anda sudah membuat AWS Secrets Manager rahasia dengan kredensi Anda, masukkan ARN untuk rahasianya. Jika membuat rahasia baru, lakukan hal berikut:
    a. Untuk Penyedia Identitas, pilih SALESFORCE.
    b. Untuk ID Klien, Rahasia Klien, URL Otorisasi, dan URL Token, masukkan semua informasi yang Anda kumpulkan dari Salesforce Data Cloud dalam prosedur sebelumnya.
8. Simpan pengaturan Domain atau profil pengguna Anda.

Anda sekarang harus dapat membuat koneksi ke data Anda di Salesforce Data Cloud dari Canvas.

Siapkan OAuth untuk Snowflake

Untuk mengatur otentikasi untuk Snowflake, Canvas mendukung penyedia identitas yang dapat Anda gunakan alih-alih meminta pengguna langsung memasukkan kredensialnya ke Canvas.

Berikut ini adalah tautan ke dokumentasi Snowflake untuk penyedia identitas yang didukung Canvas:

- Azure AD
- Okta
- Federasi Ping

Proses berikut menjelaskan langkah-langkah umum yang harus Anda ambil. Untuk petunjuk lebih rinci tentang melakukan langkah-langkah ini, Anda dapat merujuk ke Menyiapkan Akses OAuth Snowflake bagian dalam dokumentasi Data Wrangler untuk mengimpor data dari Snowflake.

Untuk mengatur OAuth untuk Snowflake, lakukan hal berikut:

1. Daftarkan Canvas sebagai aplikasi dengan penyedia identitas. Ini memerlukan penentuan URL pengalihan ke Canvas, yang harus mengikuti format ini: `https://<domain-id>.studio.<region>.sagemaker.aws/canvas/default`
2. Dalam penyedia identitas, buat server atau API yang mengirimkan token OAuth ke Canvas sehingga Canvas dapat mengakses Snowflake. Saat menyiapkan server, gunakan kode otorisasi dan segarkan jenis pemberian token, tentukan masa pakai token akses, dan setel kebijakan token penyegaran. Selain itu, dalam Integrasi Keamanan OAuth Eksternal untuk Snowflake, aktifkan. `external_oauth_any_role_mode`
3. Dapatkan informasi berikut dari penyedia identitas: URL token, URL otorisasi, ID klien, rahasia klien. Untuk Azure AD, ambil juga kredensyal cakupan OAuth.
4. Simpan informasi yang diambil pada langkah sebelumnya secara AWS Secrets Manager rahasia.

    a. Untuk Okta dan Ping Federate, rahasianya akan terlihat seperti format berikut:

    ```
    {"token_url":"https://identityprovider.com/oauth2/example-portion-of-URL-path/
    v2/token",
    "client_id":"example-client-id", "client_secret":"example-client-secret",
     "identity_provider":"OKTA"|"PING_FEDERATE",
    "authorization_url":"https://identityprovider.com/oauth2/example-portion-of-
    URL-path/v2/authorize"}
    ```

    b. Untuk Azure AD, rahasianya juga harus menyertakan kredensyal cakupan OAuth sebagai bidang. `datasource_oauth_scope`

Setelah mengonfigurasi penyedia identitas dan rahasianya, Anda harus membuat AWS Secrets Managerrahasia untuk menyimpan informasi dan menambahkannya ke SageMaker Domain Amazon atau profil pengguna Anda. Perhatikan bahwa Anda dapat menambahkan rahasia ke Domain dan profil pengguna, tetapi Canvas mencari rahasia di profil pengguna terlebih dahulu.

Untuk menambahkan rahasia ke Domain atau profil pengguna Anda, lakukan hal berikut:

1. Pergi ke SageMaker konsol Amazon.
2. Pilih Domain di panel navigasi.
3. Dari daftar Domain, pilih Domain Anda.
    a. Jika menambahkan rahasia Anda ke Domain Anda, lakukan hal berikut:
        i. Pilih Domain.
        ii. Pada halaman Pengaturan domain, pilih tab Pengaturan domain.
        iii. Pilih Edit.
    b. Jika menambahkan rahasia ke profil pengguna Anda, lakukan hal berikut:
        i. Pilih domain pengguna.
        ii. Pada halaman Pengaturan domain, pilih profil pengguna.
        iii. Pada halaman Detail Pengguna, pilih Edit.
4. Di panel navigasi, pilih Pengaturan kanvas.
5. Untuk pengaturan OAuth, pilih Tambahkan konfigurasi OAuth.
6. Untuk Sumber data, pilih Snowflake.
7. Untuk Pengaturan Rahasia, pilih Buat rahasia baru. Atau, jika Anda sudah membuat AWS Secrets Manager rahasia dengan kredensi Anda, masukkan ARN untuk rahasianya. Jika membuat rahasia baru, lakukan hal berikut:
    a. Untuk Penyedia Identitas, pilih SNOWFLAKE.
    b. Untuk ID Klien, Rahasia Klien, URL Otorisasi, dan URL Token, masukkan semua informasi yang Anda kumpulkan dari penyedia identitas dalam prosedur sebelumnya.
8. Simpan pengaturan Domain atau profil pengguna Anda.

Anda sekarang harus dapat membuat koneksi ke data Anda di Snowflake dari Canvas.

Impor data ke Canvas

Amazon SageMaker Canvas mendukung mengimpor data tabel, gambar, dan dokumen. Anda dapat mengimpor data dari sumber data lokal dan eksternal ke Canvas. Gunakan kumpulan data yang Anda impor untuk membuat model dan membuat prediksi untuk kumpulan data lainnya.

Setiap kasus penggunaan yang dapat Anda buat model kustom menerima berbagai jenis input. Misalnya, jika Anda ingin membangun model klasifikasi gambar label tunggal, maka Anda harus mengimpor data gambar. Untuk informasi selengkapnya tentang berbagai jenis model dan data yang mereka terima, lihatMembangun model kustom. Anda dapat mengimpor data dan membuat model kustom di SageMaker Canvas untuk tipe data berikut:

- Tabular (CSV, Parket, atau tabel)
  - Kategoris — Gunakan data kategoris untuk membuat model prediksi kategoris khusus untuk prediksi kategori 2 dan 3+.
  - Numerik — Gunakan data numerik untuk membuat model prediksi numerik kustom.
  - Teks — Gunakan data teks untuk membuat model prediksi teks multi-kategori kustom.
  - Timeseries — Gunakan data timeseries untuk membuat model peramalan deret waktu kustom.
- Gambar (JPG atau PNG) - Gunakan data gambar untuk membuat model prediksi gambar label tunggal khusus.
- Dokumen (PDF, JPG, PNG, TIFF) - Data dokumen hanya didukung untuk model Canvas SageMaker Ready-to-use . Untuk mempelajari lebih lanjut tentang eady-to-use model R yang dapat membuat prediksi untuk data dokumen, lihatGunakan eady-to-use model R.

Anda dapat mengimpor data ke Canvas dari sumber data berikut:

- File lokal di komputer Anda
- Bucket Amazon S3
- Amazon Redshift
- AWS Glue Data Catalogmelalui Amazon Athena
- Amazon Aurora
- Amazon Relational Database Service (Amazon RDS)
- Awan Data Salesforce
- Kepingan salju
- Databricks, SQLServer, MariaDB, dan database populer lainnya melalui konektor JDBC

- Lebih dari 40 platform SaaS eksternal, seperti SAP OData

Untuk daftar lengkap sumber data yang dapat Anda impor, lihat tabel berikut:

Sumber	Tipe	Jenis data yang didukung
Unggahan file lokal	Lokal:	Tabular, Gambar, Dokumen
Amazon Aurora	Amazon internal	Tabular
Bucket Amazon S3	Amazon internal	Tabular, Gambar, Dokumen
Amazon RDS	Amazon internal	Tabular
Amazon Redshift	Amazon internal	Tabular
AWS Glue Data Catalog(melalui Amazon Athena)	Amazon internal	Tabular
Databricks	Eksternal	Tabular
Kepingan salju	Eksternal	Tabular
Awan Data Salesforce	Eksternal	Tabular
SQLServer	Eksternal	Tabular
MySQL	Eksternal	Tabular
PostgreSQL	Eksternal	Tabular
MariaDB	Eksternal	Tabular
Amplitudo	Platform SaaS eksternal	Tabular
CircleCI	Platform SaaS eksternal	Tabular
DocuSign Monitor	Platform SaaS eksternal	Tabular
Domo	Platform SaaS eksternal	Tabular

Sumber	Tipe	Jenis data yang didukung
Datadog	Platform SaaS eksternal	Tabular
Dynatrace	Platform SaaS eksternal	Tabular
Iklan Facebook	Platform SaaS eksternal	Tabular
Wawasan Halaman Facebook	Platform SaaS eksternal	Tabular
Iklan Google	Platform SaaS eksternal	Tabular
Google Analytics 4	Platform SaaS eksternal	Tabular
Konsol Penelusuran Google	Platform SaaS eksternal	Tabular
GitHub	Platform SaaS eksternal	Tabular
GitLab	Platform SaaS eksternal	Tabular
Infor Nexus	Platform SaaS eksternal	Tabular
Iklan Instagram	Platform SaaS eksternal	Tabular
Awan Jira	Platform SaaS eksternal	Tabular
LinkedIn Iklan	Platform SaaS eksternal	Tabular
LinkedIn Iklan	Platform SaaS eksternal	Tabular
MailChimp	Platform SaaS eksternal	Tabular
Marketo	Platform SaaS eksternal	Tabular
Tim Microsoft	Platform SaaS eksternal	Tabular
Mixpanel	Platform SaaS eksternal	Tabular
Okta	Platform SaaS eksternal	Tabular
Salesforce	Platform SaaS eksternal	Tabular

Sumber	Tipe	Jenis data yang didukung
Cloud Pemasaran Salesforce	Platform SaaS eksternal	Tabular
Salesforce Pardot	Platform SaaS eksternal	Tabular
SAP OData	Platform SaaS eksternal	Tabular
SendGrid	Platform SaaS eksternal	Tabular
ServiceNow	Platform SaaS eksternal	Tabular
Tunggal	Platform SaaS eksternal	Tabular
Kendur	Platform SaaS eksternal	Tabular
Stripe	Platform SaaS eksternal	Tabular
Tren Mikro	Platform SaaS eksternal	Tabular
Jenis huruf	Platform SaaS eksternal	Tabular
Veeva	Platform SaaS eksternal	Tabular
Zendesk	Platform SaaS eksternal	Tabular
Obrolan Zendesk	Platform SaaS eksternal	Tabular
Jual Zendesk	Platform SaaS eksternal	Tabular
Sinar Matahari Zendesk	Platform SaaS eksternal	Tabular
Pertemuan Zoom	Platform SaaS eksternal	Tabular

Untuk petunjuk tentang cara mengimpor data dan informasi mengenai persyaratan data input, seperti ukuran file maksimum untuk gambar, lihatBuat kumpulan data.

Canvas juga menyediakan beberapa kumpulan data sampel dalam aplikasi Anda untuk membantu Anda memulai. Untuk mempelajari lebih lanjut tentang kumpulan data sampel SageMaker yang disediakan yang dapat Anda coba, lihat Menggunakan kumpulan data sampel.

Setelah Anda mengimpor dataset ke Canvas, Anda dapat memperbarui dataset kapan saja. Anda dapat melakukan pembaruan manual atau Anda dapat mengatur jadwal untuk pembaruan dataset otomatis. Untuk informasi selengkapnya, lihat Memperbarui kumpulan data.

Untuk informasi selengkapnya yang spesifik untuk setiap jenis kumpulan data, lihat bagian berikut:

Tabular

Untuk mengimpor data dari sumber data eksternal (seperti database Snowflake atau platform SaaS), Anda harus mengautentikasi dan terhubung ke sumber data dalam aplikasi Canvas. Untuk informasi selengkapnya, lihat Connect ke sumber data.

Setelah membuat kumpulan data di Canvas, Anda dapat menggabungkan beberapa kumpulan data ke dalam satu kumpulan data. Bergabung dengan dataset hanya didukung untuk dataset tabular. Selama data Anda disusun ke dalam tabel, Anda dapat menggabungkan kumpulan data dari berbagai sumber, seperti Amazon Redshift, Amazon Athena, atau Snowflake. Untuk informasi tentang menggabungkan kumpulan data, lihat. Bergabunglah dengan data yang telah Anda impor ke SageMaker Canvas

Gambar

Untuk informasi tentang cara mengedit kumpulan data gambar dan melakukan tugas seperti menetapkan atau menetapkan ulang label, menambahkan gambar, atau menghapus gambar, lihat. Mengedit kumpulan data gambar

Buat kumpulan data

Bagian berikut menjelaskan cara membuat kumpulan data di Amazon SageMaker Canvas. Untuk model khusus, Anda dapat membuat kumpulan data untuk data tabel dan gambar. Untuk eady-to-use model R, Anda dapat menggunakan kumpulan data tabel dan gambar serta kumpulan data dokumen. Pilih alur kerja Anda berdasarkan informasi berikut:

- Untuk data kategoris, numerik, teks, dan timeseries, lihat. Impor data tabular
- Untuk data gambar, lihatImpor data gambar.
- Untuk data dokumen, lihatImpor data dokumen.

> **Note**
>
> Untuk informasi tentang cara mengimpor kumpulan data dokumen untuk eady-to-use model R yang menerima data dokumen, lihat Impor data dokumen alur kerja dalam dokumentasi eady-to-use model R.

Dataset dapat terdiri dari beberapa file. Misalnya, Anda mungkin memiliki beberapa file data inventaris dalam format CSV. Anda dapat mengunggah file-file ini bersama-sama sebagai kumpulan data selama skema (atau nama kolom dan tipe data) dari file tersebut cocok.

Canvas juga mendukung pengelolaan beberapa versi dataset Anda. Saat Anda membuat kumpulan data, versi pertama diberi label sebagai. V1 Anda dapat membuat versi baru dari dataset Anda dengan memperbarui dataset Anda. Anda dapat melakukan pembaruan manual, atau Anda dapat mengatur jadwal otomatis untuk memperbarui kumpulan data Anda dengan data baru. Untuk informasi selengkapnya, lihat Memperbarui kumpulan data.

Saat Anda mengimpor data ke Canvas, pastikan data tersebut memenuhi persyaratan dalam tabel berikut. Keterbatasan khusus untuk jenis model yang Anda bangun.

Kuota	2 kategori, 3+ kategori, numerik, dan model deret waktu	Model prediksi teks	Model prediksi gambar	* Data dokumen untuk model R eady-to-use
Tipe file yang didukung	CSV dan Parket (unggahan lokal, Amazon S3, atau database) JSON (database)	CSV dan Parket (unggahan lokal, Amazon S3, atau database) JSON (database)	JPG, PNG	PDF, JPG, PNG, TIFF
Ukuran maksimum file	5 GB (untuk semua file dalam dataset)	5 MB (untuk semua file dalam dataset)	30 MB per gambar	5 MB per dokumen

Kuota	2 kategori, 3+ kategori, numerik, dan model deret waktu	Model prediksi teks	Model prediksi gambar	* Data dokumen untuk model R eady-to-use
Jumlah maksimum file dalam kumpulan data tabular	50	50	N/A	N/A
Jumlah maksimum file dalam kumpulan data tabular untuk satu unggahan manual	20	20	N/A	N/A
Jumlah kolom maksimum	1000	1000	N/A	N/A
Jumlah maksimum entri (baris, gambar, atau dokumen) untuk build Cepat	50.000 baris	7500 baris	5000 gambar	N/A
Jumlah maksimum entri (baris, gambar, atau dokumen) untuk build Standar	N/A	150.000 baris	180.000 gambar	N/A

Kuota	2 kategori, 3+ kategori, numerik, dan model deret waktu	Model prediksi teks	Model prediksi gambar	* Data dokumen untuk model R eady-to-use
Jumlah minimum entri (baris) untuk build Cepat	2 kategori: 500 baris 3+ kategori, numerik, deret waktu: N/A	N/A	N/A	N/A
Jumlah minimum entri (baris, gambar, atau dokumen) untuk build Standar	250 baris	50 baris	50 gambar	N/A
Jumlah minimum entri (baris atau gambar) per label	N/A	25 baris	25 baris	N/A
Jumlah minimum label	2 kategori: 2 3+ kategori: 3 Numerik, deret waktu: N/A	2	2	N/A
Ukuran sampel minimum untuk pengambilan sampel acak	500	N/A	N/A	N/A

Kuota	2 kategori, 3+ kategori, numerik, dan model deret waktu	Model prediksi teks	Model prediksi gambar	* Data dokumen untuk model R eady-to-use
Ukuran sampel maksimum untuk pengambilan sampel acak	40.000	N/A	N/A	N/A
Jumlah label maksimum	2 kategori: 2 3+ kategori, numerik, deret waktu: N/A	1000	1000	N/A

*Data dokumen saat ini hanya didukung untuk eady-to-use model R yang menerima data dokumen. Anda tidak dapat membuat model kustom dengan data dokumen.

Perhatikan juga batasan berikut:

- Untuk data tabular, Canvas melarang memilih file apa pun dengan ekstensi selain .csv, .parquet, .parq, dan.pqt untuk unggahan lokal dan impor Amazon S3. File CSV harus dibatasi koma dan tidak memiliki karakter baris baru kecuali saat menunjukkan baris baru.
- Untuk data tabular menggunakan file Parket, perhatikan hal berikut:
  - File parket tidak dapat menyertakan tipe kompleks seperti peta dan daftar.
  - Nama kolom file Parket tidak dapat berisi spasi.
  - Jika menggunakan kompresi, file Parket harus menggunakan jenis kompresi gzip atau snappy. Untuk informasi selengkapnya tentang jenis kompresi sebelumnya, lihat dokumentasi gzip dan dokumentasi tajam.
- Untuk data gambar, jika Anda memiliki gambar yang tidak berlabel, Anda harus memberi label sebelum membuat model Anda. Untuk informasi tentang cara menetapkan label ke gambar dalam aplikasi Canvas, lihatMengedit kumpulan data gambar.

- Jika Anda mengatur pembaruan kumpulan data otomatis atau konfigurasi prediksi batch otomatis, Anda hanya dapat membuat total 20 konfigurasi di aplikasi Canvas Anda. Untuk informasi selengkapnya, lihat Kelola otomatisasi.

Setelah mengimpor kumpulan data, Anda dapat melihat kumpulan data Anda di halaman Datasets kapan saja.

Impor data tabular

Dengan kumpulan data tabular, Anda dapat membuat model prediksi kategoris, numerik, deret waktu, dan prediksi teks. Tinjau tabel batasan di bagian Impor kumpulan data sebelumnya untuk memastikan bahwa data Anda memenuhi persyaratan untuk data tabular (perhatikan bahwa batas ukuran sampel hanya berlaku saat melihat pratinjau data Anda sebelum membuat model Anda).

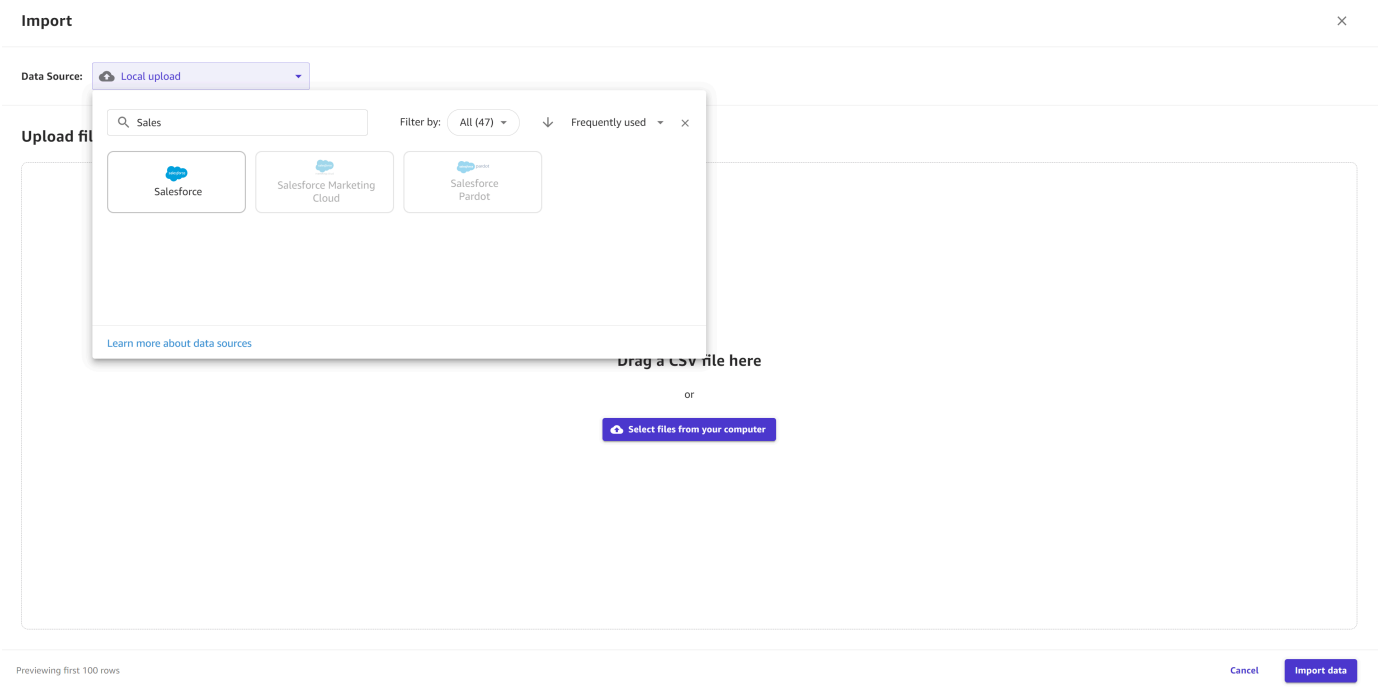
Gunakan prosedur berikut untuk mengimpor dataset tabular ke Canvas:

1. Buka aplikasi SageMaker Canvas Anda.
2. Di panel navigasi kiri, pilih Datasets.
3. Pilih Impor.
4. Di kotak dialog popup, di bidang Nama dataset, masukkan nama untuk kumpulan data dan pilih Buat.
5. Pada halaman Impor, buka menu tarik-turun Sumber Data.
6. Pilih sumber data Anda:

    - Untuk mengunggah file dari komputer Anda, pilih Unggah lokal.
    - Untuk mengimpor data dari sumber lain, seperti bucket Amazon S3 atau database Snowflake, cari sumber data Anda di bilah sumber data Penelusuran. Kemudian, pilih ubin untuk sumber data yang Anda inginkan.

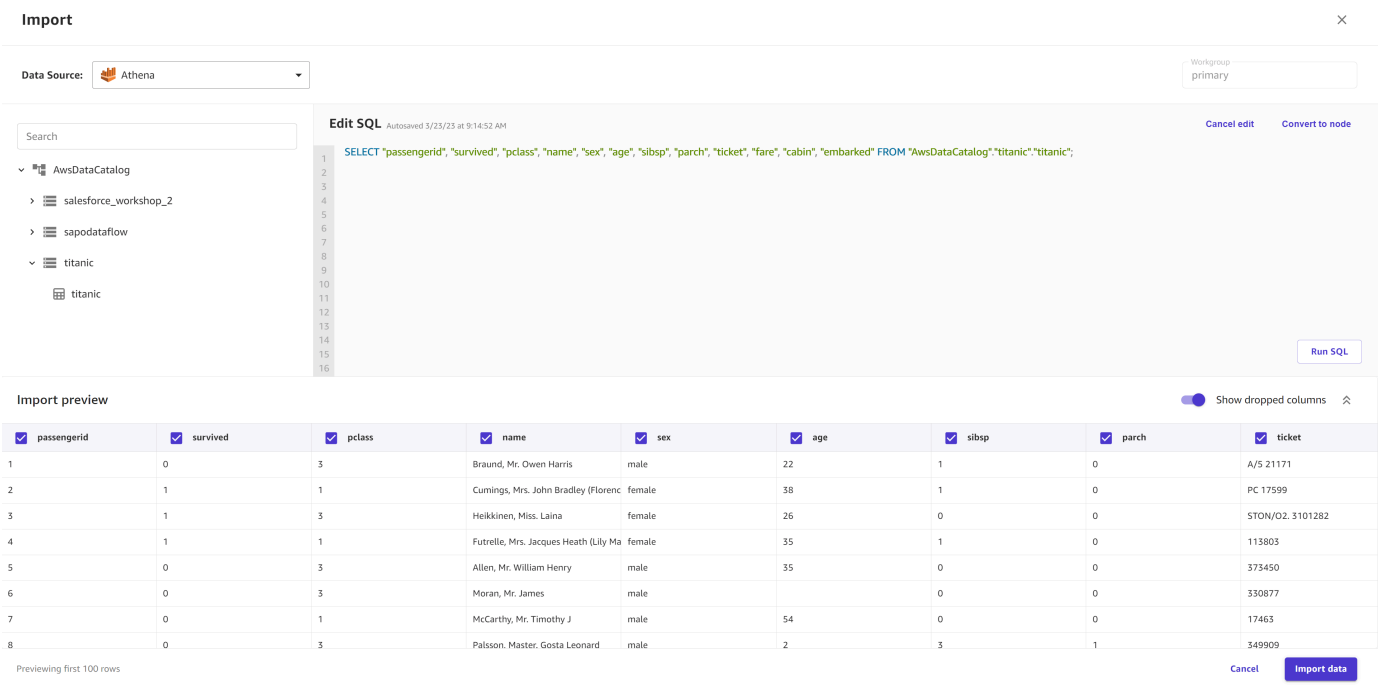
    > **Note**
    >
    > Anda hanya dapat mengimpor data dari ubin yang memiliki koneksi aktif. Jika Anda ingin terhubung ke sumber data yang tidak tersedia untuk Anda, hubungi administrator Anda. Jika Anda seorang administrator, lihatConnect ke sumber data.

    Tangkapan layar berikut menunjukkan Sumber Data menu tarik-turun.



7. (Opsional) Jika Anda menyambung ke database Amazon Redshift atau Snowflake untuk pertama kalinya, kotak dialog akan muncul untuk membuat sambungan. Isi kotak dialog dengan kredensialmu dan pilih Buat koneksi. Jika Anda sudah memiliki koneksi, pilih koneksi Anda.
8. Dari sumber data Anda, pilih file yang akan diimpor. Untuk mengunggah dan mengimpor lokal dari Amazon S3, Anda dapat memilih file. Hanya untuk Amazon S3, Anda juga memiliki opsi untuk langsung memasukkan URI S3 atau ARN bucket Anda di bidang titik akhir Input S3 dan kemudian memilih file yang akan diimpor. Untuk sumber database, Anda dapat tabel drag-and-drop data dari panel navigasi kiri.
9. (Opsional) Untuk sumber data tabular yang mendukung kueri SQL (seperti Amazon Redshift, Amazon Athena, atau Snowflake), Anda dapat memilih Edit di SQL untuk membuat kueri SQL dan menggabungkan tabel sebelum mengimpornya. Untuk informasi selengkapnya, lihat Bergabunglah dengan data yang telah Anda impor ke SageMaker Canvas.

   Tangkapan layar berikut menunjukkan tampilan Edit SQL untuk sumber data Amazon Athena.



10. (Opsional) Anda dapat memilih Pratinjau untuk melihat pratinjau kumpulan data Anda sebelum mengimpor. Untuk kumpulan data tabular, ini menunjukkan kepada Anda hingga 100 baris pertama dari kumpulan data Anda. Tangkapan layar berikut menunjukkan kepada Anda Impor pratinjau layar
11. Saat Anda siap mengimpor data, pilih Impor data.

Saat dataset Anda mengimpor ke Canvas, Anda dapat melihat kumpulan data Anda terdaftar di halaman Datasets. Dari halaman ini, Anda bisaLihat detail dataset Anda.

Ketika Status kumpulan data Anda ditampilkan sebagai`Ready`, Canvas berhasil mengimpor data Anda dan Anda dapat melanjutkan dengan membangun model.

Jika Anda memiliki koneksi ke sumber data, seperti database Amazon Redshift atau konektor SaaS, Anda dapat kembali ke koneksi itu. Untuk Amazon Redshift dan Snowflake, Anda dapat menambahkan koneksi lain dengan membuat kumpulan data lain, kembali ke halaman Impor data, dan memilih ubin Sumber Data untuk koneksi tersebut. Dari menu tarik-turun, Anda dapat membuka koneksi sebelumnya atau memilih Tambahkan koneksi.

> **Note**
>
> Untuk platform SaaS, Anda hanya dapat memiliki satu koneksi per sumber data.

Impor data gambar

Dengan kumpulan data gambar, Anda dapat membuat model kustom prediksi gambar label tunggal, yang memprediksi label untuk gambar. Tinjau batasan di bagian Impor kumpulan data sebelumnya untuk memastikan bahwa kumpulan data gambar Anda memenuhi persyaratan untuk data gambar.

> **Note**
>
> Anda hanya dapat mengimpor kumpulan data gambar dari unggahan file lokal atau bucket Amazon S3. Selain itu, untuk kumpulan data gambar, Anda harus memiliki setidaknya 25 gambar per label.

Gunakan prosedur berikut untuk mengimpor dataset gambar ke Canvas:

1. Buka aplikasi SageMaker Canvas Anda.
2. Di panel navigasi kiri, pilih Datasets.
3. Pilih Buat.
4. Dari menu dropdown, pilih Gambar.
5. Di kotak dialog popup, di bidang Nama dataset, masukkan nama untuk kumpulan data dan pilih Buat.
6. Pada halaman Impor, buka menu tarik-turun Sumber Data.
7. Pilih sumber data Anda. Untuk mengunggah file dari komputer Anda, pilih Unggah lokal. Untuk mengimpor file dari Amazon S3, pilih Amazon S3.
8. Dari komputer atau bucket Amazon S3 Anda, pilih gambar atau folder gambar yang ingin Anda unggah.
9. Saat Anda siap mengimpor data, pilih Impor data.

Saat dataset Anda mengimpor ke Canvas, Anda dapat melihat kumpulan data Anda terdaftar di halaman Datasets. Dari halaman ini, Anda bisaLihat detail dataset Anda.

Ketika Status kumpulan data Anda ditampilkan sebagai`Ready`, Canvas berhasil mengimpor data Anda dan Anda dapat melanjutkan dengan membangun model.

Saat membuat model, Anda dapat mengedit kumpulan data gambar, dan Anda dapat menetapkan atau menetapkan ulang label, menambahkan gambar, atau menghapus gambar dari kumpulan

data Anda. Untuk informasi selengkapnya tentang cara mengedit kumpulan data gambar Anda, lihatMengedit kumpulan data gambar.

Impor data dokumen

eady-to-use Model R untuk analisis pengeluaran, analisis dokumen identitas, analisis dokumen, dan kueri dokumen mendukung data dokumen. Anda tidak dapat membuat model kustom dengan data dokumen.

Dengan kumpulan data dokumen, Anda dapat menghasilkan prediksi untuk analisis pengeluaran, analisis dokumen identitas, analisis dokumen, dan model R kueri dokumen. eady-to-use Tinjau tabel batasan di Buat kumpulan data bagian untuk memastikan bahwa kumpulan data dokumen Anda memenuhi persyaratan untuk data dokumen.

> **Note**
>
> Anda hanya dapat mengimpor kumpulan data dokumen dari unggahan file lokal atau bucket Amazon S3.

Gunakan prosedur berikut untuk mengimpor dataset dokumen ke Canvas:

1. Buka aplikasi SageMaker Canvas Anda.
2. Di panel navigasi kiri, pilih Datasets.
3. Pilih Buat.
4. Dari menu dropdown, pilih Document.
5. Di kotak dialog popup, di bidang Nama dataset, masukkan nama untuk kumpulan data dan pilih Buat.
6. Pada halaman Impor, buka menu tarik-turun Sumber Data.
7. Pilih sumber data Anda. Untuk mengunggah file dari komputer Anda, pilih Unggah lokal. Untuk mengimpor file dari Amazon S3, pilih Amazon S3.
8. Dari komputer atau bucket Amazon S3 Anda, pilih file dokumen yang ingin Anda unggah.
9. Saat Anda siap mengimpor data, pilih Impor data.

Saat dataset Anda mengimpor ke Canvas, Anda dapat melihat kumpulan data Anda terdaftar di halaman Datasets. Dari halaman ini, Anda bisaLihat detail dataset Anda.

Ketika Status kumpulan data Anda ditampilkan sebagai`Ready`, Canvas telah berhasil mengimpor data Anda.

Pada halaman Datasets, Anda dapat memilih dataset Anda untuk melihat pratinjau, yang menunjukkan hingga 100 dokumen pertama dari dataset Anda.

Lihat detail dataset Anda

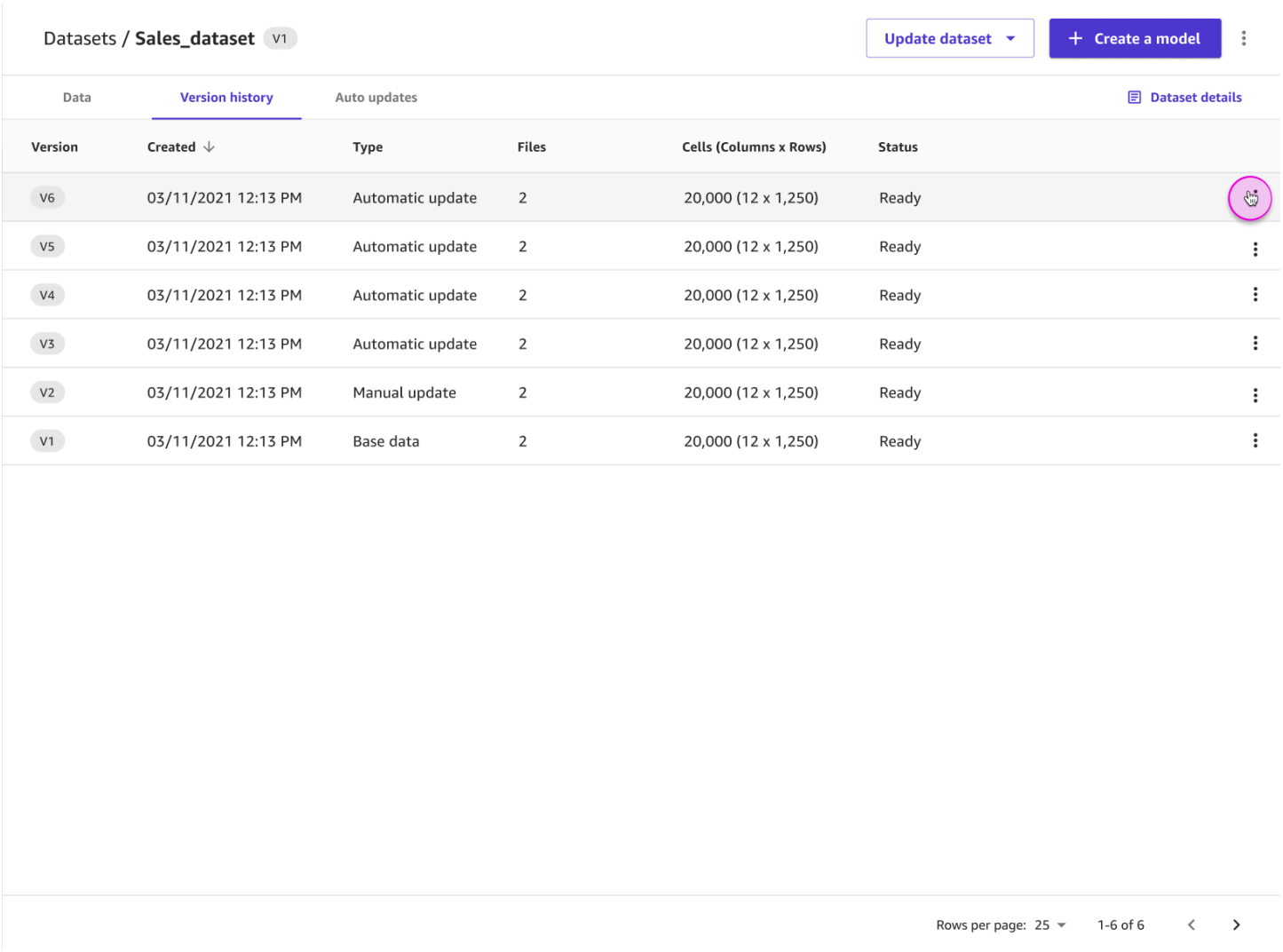
Untuk setiap kumpulan data, Anda dapat melihat semua file dalam kumpulan data, riwayat versi kumpulan data, dan konfigurasi pembaruan otomatis apa pun untuk kumpulan data. Dari halaman Datasets, Anda juga dapat memulai tindakan seperti atau. Memperbarui kumpulan data Membangun model kustom

Untuk melihat detail kumpulan data, lakukan hal berikut:

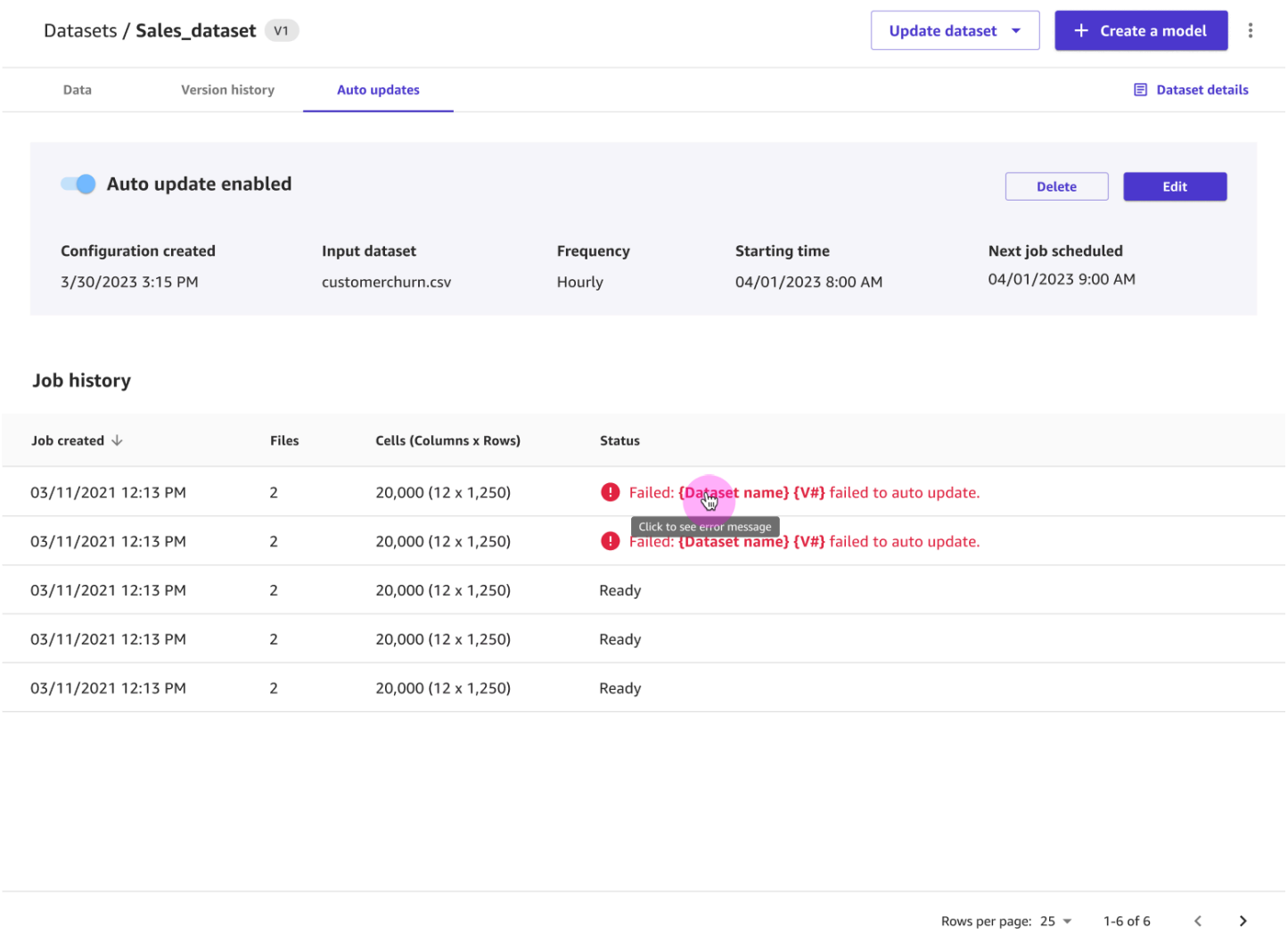
1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Datasets.
3. Dari daftar kumpulan data, pilih kumpulan data Anda.

Pada tab Data, Anda dapat melihat pratinjau data Anda. Jika Anda memilih detail Dataset, Anda dapat melihat semua file yang merupakan bagian dari kumpulan data Anda. Pilih file untuk melihat hanya data dari file itu di pratinjau. Untuk kumpulan data gambar, pratinjau hanya menampilkan 100 gambar pertama dari kumpulan data Anda.

Pada tab Riwayat versi, Anda dapat melihat daftar semua versi kumpulan data Anda. Versi baru dibuat setiap kali Anda memperbarui dataset. Untuk mempelajari lebih lanjut tentang memperbarui kumpulan data, lihatMemperbarui kumpulan data. Tangkapan layar berikut menunjukkan tab Riwayat versi di aplikasi Canvas.



Pada tab Pembaruan otomatis, Anda dapat mengaktifkan pembaruan otomatis untuk kumpulan data dan mengatur konfigurasi untuk memperbarui kumpulan data Anda pada jadwal reguler. Untuk mempelajari selengkapnya tentang menyiapkan pembaruan otomatis untuk kumpulan data, lihatKonfigurasikan pembaruan otomatis untuk kumpulan data. Tangkapan layar berikut menunjukkan tab Pembaruan otomatis dengan pembaruan otomatis diaktifkan dan daftar pekerjaan pembaruan otomatis yang telah dilakukan pada kumpulan data.



Memperbarui kumpulan data

Setelah mengimpor dataset awal ke Amazon SageMaker Canvas, Anda mungkin memiliki data tambahan yang ingin ditambahkan ke kumpulan data Anda. Misalnya, Anda mungkin mendapatkan data inventaris di akhir setiap minggu yang ingin Anda tambahkan ke kumpulan data Anda. Alih-alih mengimpor data Anda beberapa kali, Anda dapat memperbarui kumpulan data yang ada dan menambah atau menghapus file darinya.

> **Note**
>
> Anda hanya dapat memperbarui kumpulan data yang telah Anda impor melalui unggahan lokal atau Amazon S3.

Anda dapat memperbarui dataset Anda baik secara manual atau otomatis. Dengan pembaruan otomatis, Anda menentukan lokasi di mana Canvas memeriksa file pada frekuensi yang Anda tentukan. Jika Anda mengimpor file baru selama pembaruan, skema file harus sama persis dengan kumpulan data yang ada.

Setiap kali Anda memperbarui dataset Anda, Canvas membuat versi baru dari dataset Anda. Anda hanya dapat menggunakan versi terbaru dari dataset Anda untuk membuat model atau menghasilkan prediksi. Untuk informasi selengkapnya tentang melihat riwayat versi kumpulan data Anda, lihatLihat detail dataset Anda.

Anda juga dapat menggunakan pembaruan kumpulan data dengan prediksi batch otomatis, yang memulai pekerjaan prediksi batch setiap kali Anda memperbarui kumpulan data Anda. Untuk informasi selengkapnya, lihat Buat prediksi batch.

Bagian berikut menjelaskan cara melakukan pembaruan manual dan otomatis pada kumpulan data Anda.

Perbarui kumpulan data secara manual

Untuk melakukan pembaruan manual, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Datasets.
3. Dari daftar kumpulan data, pilih kumpulan data yang ingin Anda perbarui.
4. Pilih menu tarik-turun Perbarui kumpulan data dan pilih Pembaruan manual. Anda dibawa ke alur kerja data impor.
5. Dari menu tarik-turun sumber data, pilih Unggahan lokal atau Amazon S3.
6. Halaman ini menunjukkan pratinjau data Anda. Dari sini, Anda dapat menambah atau menghapus file dari kumpulan data. Jika Anda mengimpor data tabular, skema file baru (nama kolom dan tipe data) harus cocok dengan skema file yang ada. Selain itu, file baru Anda tidak boleh melebihi ukuran set data maksimum atau ukuran file. Untuk informasi selengkapnya tentang batasan ini, lihat Mengimpor kumpulan data.

   > **Note**
   >
   > Jika Anda menambahkan file dengan nama yang sama dengan file yang ada di kumpulan data Anda, file baru akan menimpa versi lama file tersebut.

7. Saat Anda siap menyimpan perubahan, pilih Perbarui kumpulan data.

Anda sekarang harus memiliki versi baru dari dataset Anda.

Pada halaman Datasets, Anda dapat memilih tab Riwayat versi untuk melihat semua versi kumpulan data Anda dan riwayat pembaruan manual dan otomatis yang telah Anda buat.

Konfigurasikan pembaruan otomatis untuk kumpulan data

Pembaruan otomatis adalah ketika Anda mengatur konfigurasi untuk Canvas untuk memperbarui dataset Anda pada frekuensi tertentu. Kami menyarankan Anda menggunakan opsi ini jika Anda secara teratur menerima file data baru yang ingin Anda tambahkan ke kumpulan data Anda.

Saat mengatur konfigurasi pembaruan otomatis, Anda menentukan lokasi Amazon S3 tempat Anda mengunggah file dan frekuensi di mana Canvas memeriksa lokasi dan mengimpor file. Setiap instance Canvas memperbarui dataset Anda disebut sebagai pekerjaan. Untuk setiap pekerjaan, Canvas mengimpor semua file di lokasi Amazon S3. Jika Anda memiliki file baru dengan nama yang sama dengan file yang ada di dataset Anda, Canvas menimpa file lama dengan file baru.

Untuk pembaruan dataset otomatis, Canvas tidak melakukan validasi skema. Jika skema file yang diimpor selama pembaruan otomatis tidak cocok dengan skema file yang ada atau melebihi batasan ukuran (lihat Mengimpor kumpulan data untuk tabel batasan ukuran file), maka Anda mendapatkan kesalahan saat pekerjaan Anda berjalan.

> **Note**
>
> Anda hanya dapat mengatur maksimum 20 konfigurasi otomatis di aplikasi Canvas Anda. Selain itu, Canvas hanya melakukan pembaruan otomatis saat Anda masuk ke aplikasi Canvas Anda. Jika Anda keluar dari aplikasi Canvas Anda, pembaruan otomatis berhenti sampai Anda masuk kembali.

Untuk mengonfigurasi pembaruan otomatis untuk kumpulan data Anda, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Datasets.
3. Dari daftar kumpulan data, pilih kumpulan data yang ingin Anda perbarui.
4. Pilih menu tarik-turun Perbarui kumpulan data dan pilih Pembaruan otomatis. Anda akan dibawa ke tab Pembaruan otomatis untuk kumpulan data.
5. Aktifkan sakelar Aktifkan pembaruan otomatis.

6. Untuk Tentukan sumber data, masukkan jalur Amazon S3 ke folder tempat Anda berencana untuk mengunggah file secara teratur.
7. Untuk Pilih frekuensi, pilih Per Jam, Mingguan, atau Harian.
8. Untuk Tentukan waktu mulai, gunakan kalender dan pemilih waktu untuk memilih kapan Anda ingin pekerjaan pembaruan otomatis pertama dimulai.
9. Saat Anda siap membuat konfigurasi pembaruan otomatis, pilih Simpan.

Canvas memulai pekerjaan pertama irama pembaruan otomatis Anda pada waktu mulai yang ditentukan.

Untuk informasi selengkapnya tentang melihat riwayat pekerjaan pembaruan otomatis atau membuat perubahan pada konfigurasi pembaruan otomatis melalui halaman Otomasi di aplikasi Canvas, lihatKelola otomatisasi.

Bagian berikut menjelaskan cara melihat, memperbarui, dan menghapus konfigurasi pembaruan otomatis Anda melalui halaman Datasets di aplikasi Canvas.

Melihat pekerjaan pembaruan kumpulan data otomatis Anda

Untuk melihat riwayat pekerjaan untuk pembaruan kumpulan data otomatis Anda, pada halaman detail kumpulan data Anda, pilih tab Pembaruan otomatis.

Setiap pembaruan otomatis ke kumpulan data ditampilkan sebagai pekerjaan di tab Pembaruan otomatis di bawah bagian Riwayat pekerjaan. Untuk setiap pekerjaan, Anda dapat melihat yang berikut:

- Job created — Stempel waktu ketika Canvas mulai memperbarui dataset.
- File — Jumlah file dalam dataset.
- Sel (Kolom x Baris) - Jumlah kolom dan baris dalam kumpulan data.
- Status - Status kumpulan data setelah pembaruan. Jika pekerjaan berhasil, statusnya Siap. Jika pekerjaan gagal karena alasan apa pun, statusnya Gagal, dan Anda dapat mengarahkan kursor ke status untuk detail selengkapnya.

Edit konfigurasi pembaruan kumpulan data otomatis Anda

Anda mungkin ingin membuat perubahan pada konfigurasi pembaruan otomatis untuk kumpulan data, seperti mengubah frekuensi pembaruan. Anda mungkin juga ingin menonaktifkan konfigurasi pembaruan otomatis untuk menjeda pembaruan pada kumpulan data Anda.

Untuk membuat perubahan pada konfigurasi pembaruan otomatis untuk kumpulan data, buka tab Pembaruan otomatis pada kumpulan data Anda dan pilih Edit untuk membuat perubahan pada konfigurasi.

Untuk menjeda pembaruan kumpulan data Anda, matikan konfigurasi otomatis Anda. Anda dapat menonaktifkan pembaruan otomatis dengan membuka tab Pembaruan otomatis pada kumpulan data Anda dan mematikan tombol Aktifkan pembaruan otomatis. Anda dapat mengaktifkan kembali sakelar ini kapan saja untuk melanjutkan jadwal pembaruan.

Hapus konfigurasi pembaruan kumpulan data otomatis Anda

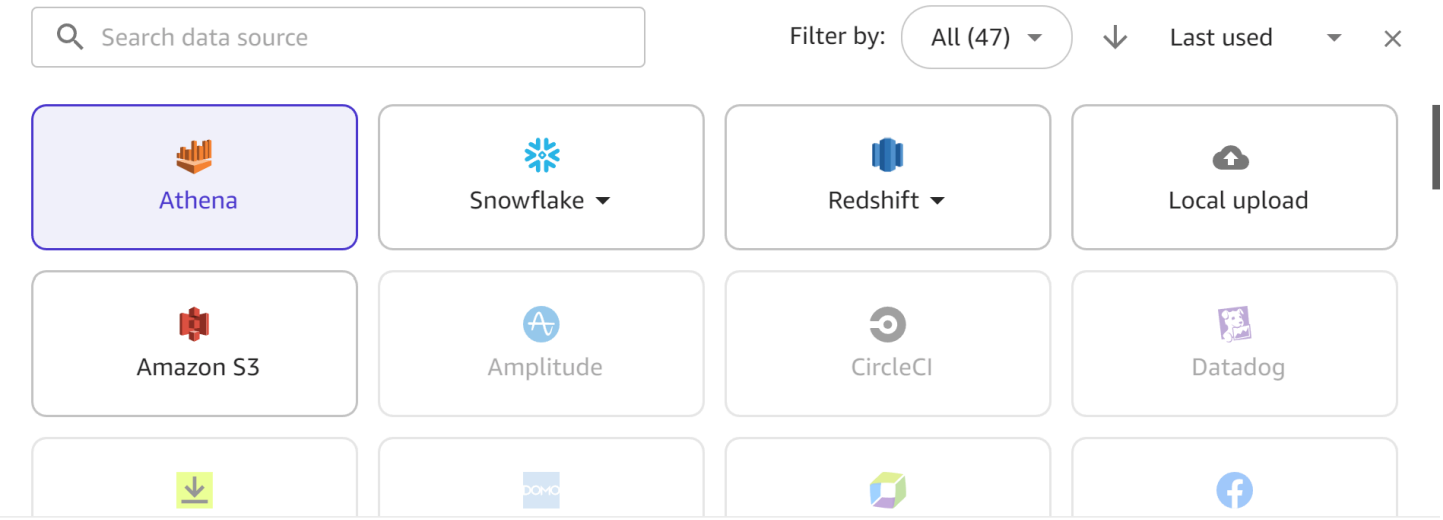
Untuk mempelajari cara menghapus konfigurasi Anda, lihatHapus konfigurasi otomatis.

Connect ke sumber data

Di Amazon SageMaker Canvas, Anda dapat mengimpor data dari lokasi di luar sistem file lokal Anda melalui AWS layanan, platform SaaS, atau database lain menggunakan konektor JDBC. Misalnya, Anda mungkin ingin mengimpor tabel dari gudang data di Amazon Redshift, atau Anda mungkin ingin mengimpor data Google Analytics.

Ketika Anda pergi melalui alur kerja Impor untuk mengimpor data dalam aplikasi Canvas, Anda dapat memilih sumber data Anda dan kemudian memilih data yang ingin Anda impor. Untuk sumber data tertentu, seperti Snowflake dan Amazon Redshift, Anda harus menentukan kredensyal Anda dan menambahkan koneksi ke sumber data.

Tangkapan layar berikut menunjukkan toolbar sumber data di alur kerja Impor, dengan semua sumber data yang tersedia disorot. Anda hanya dapat mengimpor data dari sumber data yang tersedia untuk Anda. Hubungi administrator Anda jika sumber data yang Anda inginkan tidak tersedia.



How to connect to data sources

Bagian berikut memberikan informasi tentang membangun koneksi ke sumber data eksternal dan dan mengimpor data dari mereka. Tinjau bagian berikut terlebih dahulu untuk menentukan izin apa yang Anda perlukan untuk mengimpor data dari sumber data Anda.

Izin

Tinjau informasi berikut untuk memastikan bahwa Anda memiliki izin yang diperlukan untuk mengimpor data dari sumber data Anda:

- Amazon S3: Anda dapat mengimpor data dari bucket Amazon S3 apa pun selama pengguna Anda memiliki izin untuk mengakses bucket. Untuk informasi selengkapnya tentang menggunakan AWS IAM untuk mengontrol akses ke bucket Amazon S3, lihat Manajemen identitas dan akses di Amazon S3 di Panduan Pengguna Amazon S3.
- Amazon Athena: Jika Anda memiliki AmazonSageMakerFullAccesskebijakan dan AmazonSageMakerCanvasFullAccesskebijakan yang dilampirkan pada peran eksekusi pengguna Anda, maka Anda akan dapat menanyakan AWS Glue Data Catalog dengan Amazon Athena. Jika Anda bagian dari workgroup Athena, pastikan bahwa pengguna Canvas memiliki izin untuk menjalankan kueri Athena pada data. Untuk informasi selengkapnya, lihat Menggunakan grup kerja untuk menjalankan kueri di Panduan Pengguna Amazon Athena.
- Amazon DocumentDB: Anda dapat mengimpor data dari database Amazon DocumentDB selama Anda memiliki kredensyal (nama pengguna dan kata sandi) untuk terhubung ke database dan memiliki izin Kanvas dasar minimum yang dilampirkan ke peran eksekusi pengguna Anda. Untuk

informasi selengkapnya tentang izin Canvas, lihat. Prasyarat untuk menyiapkan Amazon Canvas SageMaker

- Amazon Redshift: Untuk memberi Anda izin yang diperlukan untuk mengimpor data dari Amazon Redshift, lihat Memberi Izin Pengguna untuk Mengimpor Data Amazon Redshift.
- Amazon RDS: Jika Anda memiliki AmazonSageMakerCanvasFullAccesskebijakan yang dilampirkan ke peran eksekusi pengguna Anda, maka Anda akan dapat mengakses database Amazon RDS Anda dari Canvas.
- Platform SaaS: Jika Anda memiliki AmazonSageMakerFullAccesskebijakan dan kebijakan yang AmazonSageMakerCanvasFullAccessmelekat pada peran eksekusi pengguna Anda, maka Anda akan memiliki izin yang diperlukan untuk mengimpor data dari platform SaaS. Lihat Gunakan konektor SaaS dengan Canvas untuk informasi lebih lanjut tentang menghubungkan ke konektor SaaS tertentu.
- Konektor JDBC: Untuk sumber database seperti Databricks, MySQL atau MariaDB, Anda harus mengaktifkan otentikasi nama pengguna dan kata sandi pada database sumber sebelum mencoba terhubung dari Canvas. Jika Anda terhubung ke database Databricks, Anda harus memiliki URL JDBC yang berisi kredensi yang diperlukan.

Connect ke database yang disimpan di AWS

Anda mungkin ingin mengimpor data yang telah Anda simpanAWS. Anda dapat mengimpor data dari Amazon S3, menggunakan Amazon Athena untuk menanyakan database diAWS Glue Data Catalog, mengimpor data dari Amazon RDS, atau membuat sambungan ke database Amazon Redshift.

Anda dapat membuat beberapa koneksi ke Amazon Redshift. Untuk Amazon Athena, Anda dapat mengakses database apa pun yang Anda miliki di situs Anda. AWS Glue Data Catalog Untuk Amazon S3, Anda dapat mengimpor data dari bucket selama Anda memiliki izin yang diperlukan.

Tinjau bagian berikut untuk informasi lebih rinci.

Connect ke data di Amazon S3, Amazon Athena, atau Amazon RDS

Untuk Amazon S3, Anda dapat mengimpor data dari bucket Amazon S3 selama Anda memiliki izin untuk mengakses bucket.

Untuk Amazon Athena, Anda dapat mengakses database AWS Glue Data Catalog selama Anda memiliki izin melalui workgroup Amazon Athena Anda.

Untuk Amazon RDS, jika Anda memiliki AmazonSageMakerCanvasFullAccesskebijakan yang dilampirkan ke peran pengguna, maka Anda akan dapat mengimpor data dari database Amazon RDS Anda ke Canvas.

Untuk mengimpor data dari bucket Amazon S3, atau menjalankan kueri dan mengimpor tabel data dengan Amazon Athena, lihat. Buat kumpulan data Anda hanya dapat mengimpor data tabular dari Amazon Athena, dan Anda dapat mengimpor data tabel dan gambar dari Amazon S3.

Connect ke database Amazon DocumentDB

Amazon DocumentDB adalah layanan database dokumen yang dikelola sepenuhnya, tanpa server. Anda dapat mengimpor data dokumen tidak terstruktur yang disimpan dalam SageMaker database Amazon DocumentDB ke Canvas sebagai kumpulan data tabular, dan kemudian Anda dapat membuat model pembelajaran mesin dengan data tersebut.

> **Important**
>
> SageMaker Domain Anda harus dikonfigurasi dalam mode VPC saja untuk menambahkan koneksi ke Amazon DocumentDB. Anda hanya dapat mengakses cluster Amazon DocumentDB di Amazon VPC yang sama dengan aplikasi Canvas Anda. Selain itu, Canvas hanya dapat terhubung ke cluster Amazon DocumentDB yang mendukung TLS. Untuk informasi selengkapnya tentang cara mengatur Canvas dalam mode VPC saja, lihat. Konfigurasikan Amazon SageMaker Canvas di VPC tanpa akses internet

Untuk mengimpor data dari database Amazon DocumentDB, Anda harus memiliki kredensyal untuk mengakses database Amazon DocumentDB dan menentukan nama pengguna dan kata sandi saat membuat koneksi database. Anda dapat mengonfigurasi izin yang lebih terperinci dan membatasi akses dengan memodifikasi izin pengguna Amazon DocumentDB. Untuk mempelajari lebih lanjut tentang kontrol akses di Amazon DocumentDB, lihat Akses Database Menggunakan Kontrol Akses Berbasis Peran di Panduan Pengembang Amazon DocumentDB.

Saat Anda mengimpor dari Amazon DocumentDB, Canvas mengonversi data tidak terstruktur menjadi kumpulan data tabular dengan memetakan bidang ke kolom dalam tabel. Tabel tambahan dibuat untuk setiap bidang kompleks (atau struktur bersarang) dalam data, di mana kolom sesuai dengan sub-bidang bidang kompleks. Untuk informasi lebih rinci tentang proses ini dan contoh konversi skema, lihat halaman Amazon DocumentDB JDBC Driver Schema Discovery. GitHub

Canvas hanya dapat membuat koneksi ke satu database di Amazon DocumentDB. Untuk mengimpor data dari database yang berbeda, Anda harus membuat koneksi baru.

Anda dapat mengimpor data dari Amazon DocumentDB ke Canvas dengan menggunakan metode berikut:

- Buat kumpulan data. Anda dapat mengimpor data Amazon DocumentDB dan membuat kumpulan data tabular di Canvas. Jika Anda memilih metode ini, pastikan Anda mengikuti prosedur Impor data tabular.
- Buat Alur Data. Anda dapat membuat pipeline persiapan data di Canvas dan menambahkan database Amazon DocumentDB sebagai sumber data.

Untuk melanjutkan dengan mengimpor data Anda, ikuti prosedur untuk salah satu metode yang ditautkan dalam daftar sebelumnya.

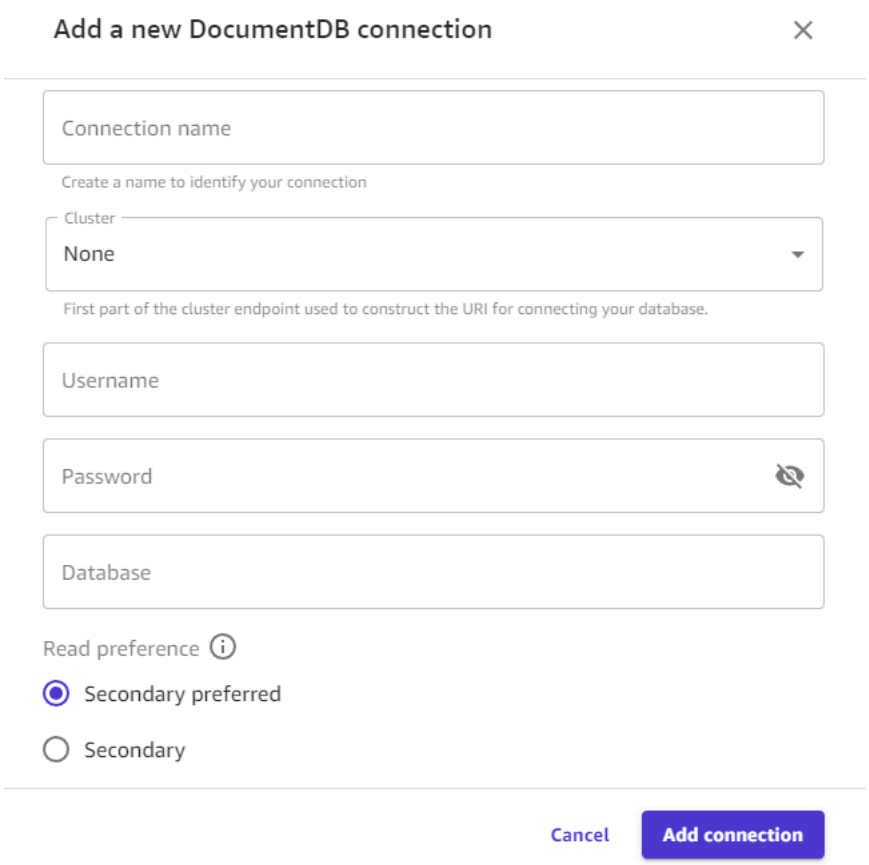
Saat Anda mencapai langkah dalam alur kerja untuk memilih sumber data (Langkah 5 untuk membuat kumpulan data, atau Langkah 6 untuk membuat aliran data), lakukan hal berikut:

1. Untuk Sumber Data, buka menu dropdown dan pilih DocumentDB.
2. Pilih Tambahkan koneksi.
3. Di kotak dialog, tentukan kredenal Amazon DocumentDB Anda:

    a. Masukkan nama Koneksi. Ini adalah nama yang digunakan oleh Canvas untuk mengidentifikasi koneksi ini.
    b. Untuk Cluster, pilih cluster di Amazon DocumentDB yang menyimpan data Anda. Canvas secara otomatis mengisi menu dropdown dengan cluster Amazon DocumentDB di VPC yang sama dengan aplikasi Canvas Anda.
    c. Masukkan Nama Pengguna untuk klaster Amazon DocumentDB Anda.
    d. Masukkan Kata Sandi untuk cluster Amazon DocumentDB Anda.
    e. Masukkan nama Database yang ingin Anda sambungkan.
    f. Opsi Preferensi Baca menentukan jenis instance di klaster Anda Canvas yang membaca datanya. Pilih salah satu dari berikut ini:

        - Pilihan sekunder — Canvas default membaca dari instance sekunder cluster, tetapi jika instance sekunder tidak tersedia, maka Canvas membaca dari instance utama.
        - Sekunder — Canvas hanya membaca dari instance sekunder cluster, yang mencegah operasi baca mengganggu operasi baca dan tulis reguler cluster.

g. Pilih Tambahkan koneksi. Gambar berikut menunjukkan kotak dialog dengan bidang sebelumnya untuk koneksi Amazon DocumentDB.



Anda sekarang harus memiliki koneksi Amazon DocumentDB, dan Anda dapat menggunakan data Amazon DocumentDB di Canvas untuk membuat kumpulan data atau aliran data.

Connect ke database Amazon Redshift

Anda dapat mengimpor data dari Amazon Redshift, gudang data tempat organisasi menyimpan datanya. Sebelum Anda dapat mengimpor data dari Amazon Redshift, peran AWS IAM yang Anda gunakan harus memiliki kebijakan `AmazonRedshiftFullAccess` terkelola yang dilampirkan. Untuk petunjuk tentang cara melampirkan kebijakan ini, lihatBerikan Izin Pengguna untuk Mengimpor Data Amazon Redshift.

Untuk mengimpor data dari Amazon Redshift, Anda melakukan hal berikut:

1. Buat koneksi ke database Amazon Redshift.
2. Pilih data yang Anda impor.

3. Impor data.

Anda dapat menggunakan editor Amazon Redshift untuk menyeret kumpulan data ke panel impor dan mengimpornya ke Canvas. SageMaker Untuk kontrol lebih lanjut atas nilai yang dikembalikan dalam kumpulan data, Anda dapat menggunakan yang berikut ini:

- Kueri SQL
- Gabungan

Kueri SQL memberi Anda kemampuan untuk menyesuaikan cara Anda mengimpor nilai dalam kumpulan data. Misalnya, Anda dapat menentukan kolom yang dikembalikan dalam kumpulan data atau rentang nilai untuk kolom.

Anda dapat menggunakan gabungan untuk menggabungkan beberapa kumpulan data dari Amazon Redshift menjadi satu kumpulan data. Anda dapat menyeret kumpulan data dari Amazon Redshift ke panel yang memberi Anda kemampuan untuk bergabung dengan kumpulan data.

Anda dapat menggunakan editor SQL untuk mengedit kumpulan data yang telah Anda gabungkan dan mengonversi kumpulan data yang digabungkan menjadi satu node. Anda dapat menggabungkan kumpulan data lain ke node. Anda dapat mengimpor data yang telah Anda pilih ke SageMaker Canvas.

Gunakan prosedur berikut untuk mengimpor data dari Amazon Redshift.

1. Dalam aplikasi SageMaker Canvas, buka halaman Datasets.
2. Pilih Buat, dan dari menu tarik-turun, pilih Tabular.
3. Masukkan nama untuk kumpulan data dan pilih Buat.
4. Untuk Sumber Data, buka menu tarik-turun dan pilih Redshift.
5. Pilih Tambahkan koneksi.
6. Di kotak dialog, tentukan kredenal Amazon Redshift Anda:
    a. Untuk metode Otentikasi, pilih IAM.
    b. Masukkan pengidentifikasi Cluster untuk menentukan cluster mana yang ingin Anda sambungkan. Masukkan hanya pengidentifikasi klaster dan bukan titik akhir penuh cluster Amazon Redshift.
    c. Masukkan nama Database database yang ingin Anda sambungkan.

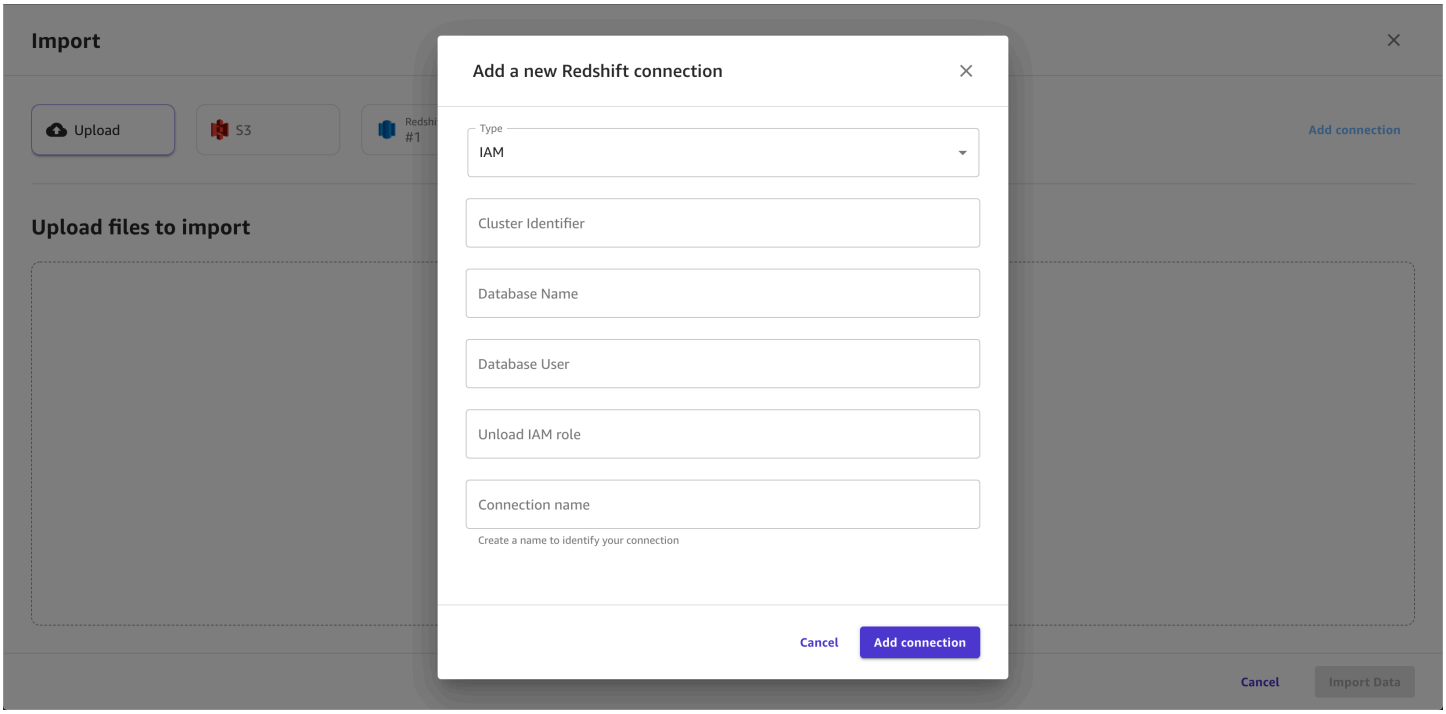
d. Masukkan pengguna Database untuk mengidentifikasi pengguna yang ingin Anda gunakan untuk terhubung ke database.

e. Untuk ARN, masukkan peran IAM ARN dari peran yang harus diasumsikan oleh cluster Amazon Redshift untuk memindahkan dan menulis data ke Amazon S3. Untuk informasi selengkapnya tentang peran ini, lihat Mengotorisasi Amazon Redshift untuk mengakses layanan AWS lain atas nama Anda di Panduan Manajemen Amazon Redshift.

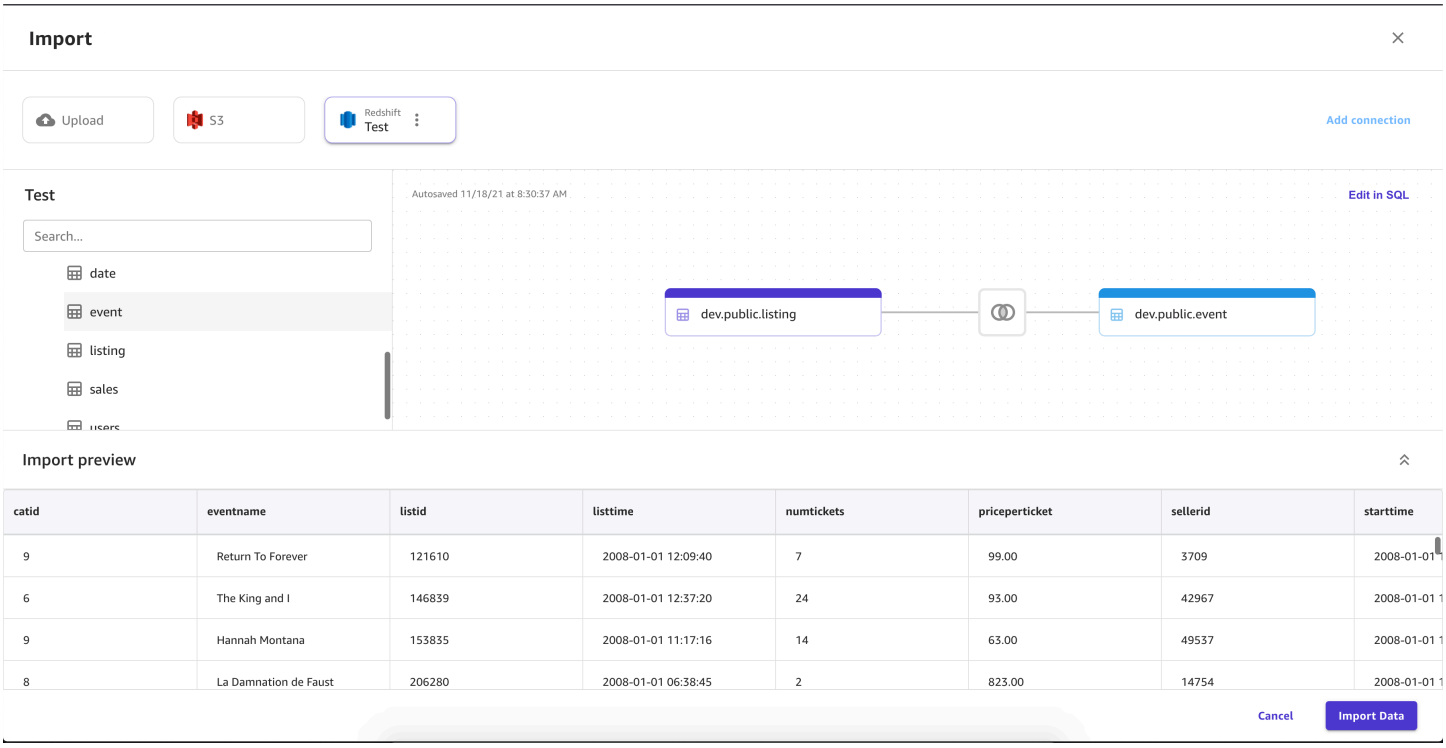
f. Masukkan nama Koneksi. Ini adalah nama yang digunakan oleh Canvas untuk mengidentifikasi koneksi ini.

7. Dari tab yang memiliki nama koneksi Anda, seret file.csv yang Anda impor ke panel Drag and drop to import table.
8. Opsional: Seret tabel tambahan ke panel impor. Anda dapat menggunakan GUI untuk bergabung dengan tabel. Untuk kekhususan lebih lanjut dalam bergabung Anda, pilih Edit di SQL.
9. Opsional: Jika Anda menggunakan SQL untuk menanyakan data, Anda dapat memilih Konteks untuk menambahkan konteks ke koneksi dengan menentukan nilai untuk hal berikut:
   - Gudang
   - Basis data
   - Skema
10. Pilih Impor data.

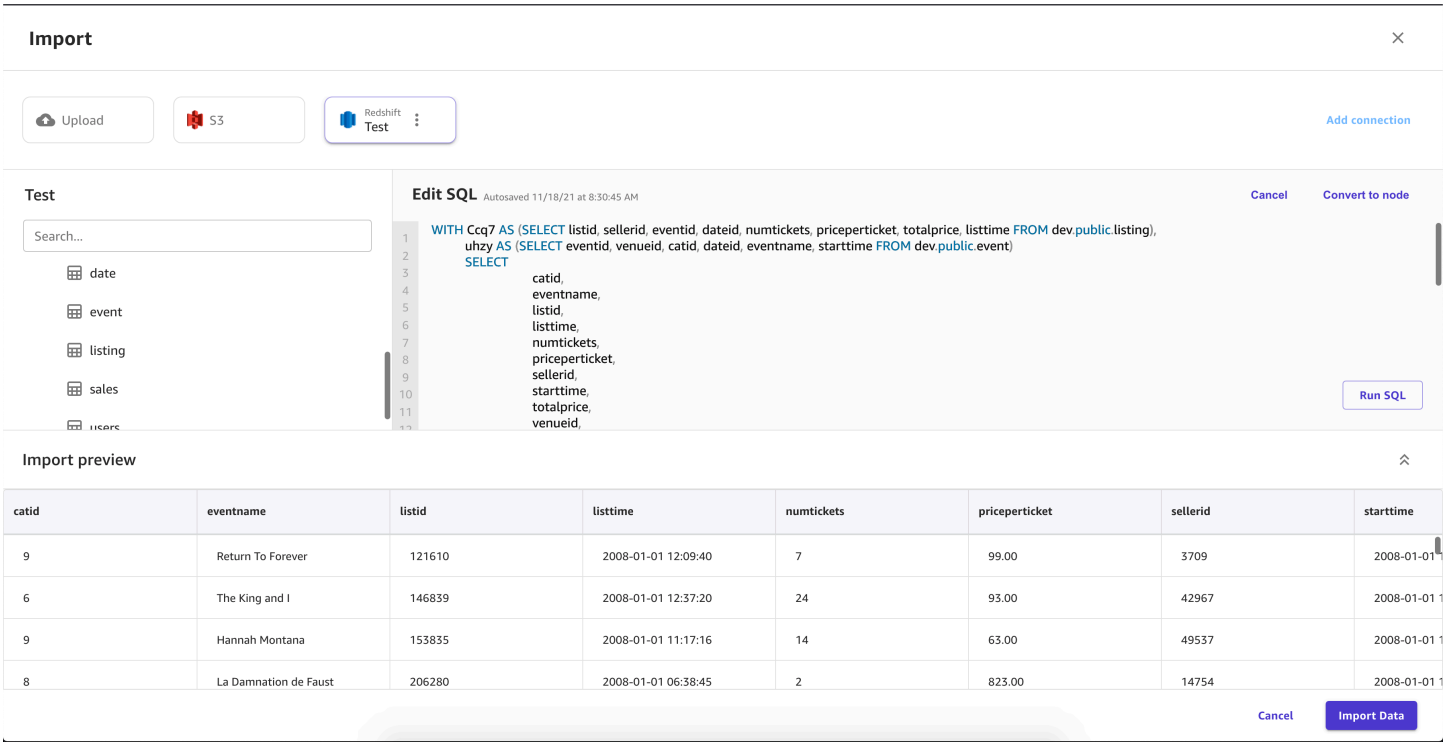
Gambar berikut menunjukkan contoh bidang yang ditentukan untuk koneksi Amazon Redshift.



Gambar berikut menunjukkan halaman yang digunakan untuk bergabung dengan kumpulan data di Amazon Redshift.



Gambar berikut menunjukkan kueri SQL yang digunakan untuk mengedit gabungan di Amazon Redshift.



Connect ke data Anda dengan konektor JDBC

Dengan JDBC, Anda dapat terhubung ke database Anda dari sumber seperti Databricks, SQLServer, MySQL, PostgreSQL, MariaDB, Amazon RDS, dan Amazon Aurora.

Anda harus memastikan bahwa Anda memiliki kredensi dan izin yang diperlukan untuk membuat koneksi dari Canvas.

- Untuk Databricks, Anda harus memberikan URL JDBC. Pemformatan URL dapat bervariasi antara instance Databricks. Untuk informasi tentang menemukan URL dan menentukan parameter di dalamnya, lihat konfigurasi JDBC dan parameter koneksi dalam dokumentasi Databricks. Berikut ini adalah contoh bagaimana URL dapat diformat: `jdbc:spark://aws-sagemaker-datawrangler.cloud.databricks.com:443/default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/3122619508517275/0909-200301-cut318;AuthMech=3;UID=token;PWD=personal-access-token`
- Untuk sumber database lainnya, Anda harus mengatur otentikasi nama pengguna dan kata sandi, lalu tentukan kredensyal tersebut saat menghubungkan ke database dari Canvas.

Selain itu, sumber data Anda harus dapat diakses melalui internet publik, atau jika aplikasi Canvas Anda berjalan dalam mode VPC saja, maka sumber data harus berjalan dalam VPC yang sama.

Untuk informasi selengkapnya tentang mengonfigurasi database Amazon RDS di VPC, lihat VPC VPC Amazon VPC dan Amazon RDS di Panduan Pengguna Amazon RDS.

Setelah mengonfigurasi kredensyal sumber data, Anda dapat masuk ke aplikasi Canvas dan membuat koneksi ke sumber data. Tentukan kredensil Anda (atau, untuk Databricks, URL) saat membuat koneksi.

Connect ke sumber data dengan OAuth

Canvas mendukung penggunaan OAuth sebagai metode otentikasi untuk menghubungkan ke data Anda di Snowflake dan Salesforce Data Cloud. OAuth adalah platform otentikasi umum untuk memberikan akses ke sumber daya tanpa berbagi kata sandi.

> **Note**
>
> Anda hanya dapat membuat satu koneksi OAuth untuk setiap sumber data.

Untuk mengotorisasi koneksi, Anda harus mengikuti pengaturan awal yang dijelaskan dalamSiapkan koneksi ke sumber data dengan OAuth.

Setelah menyiapkan kredensyal OAuth, Anda dapat melakukan hal berikut untuk menambahkan koneksi Snowflake atau Salesforce Data Cloud dengan OAuth:

1. Masuk ke aplikasi Canvas.
2. Buat dataset tabular. Saat diminta untuk mengunggah data, pilih Snowflake atau Salesforce Data Cloud sebagai sumber data Anda.
3. Buat koneksi baru ke sumber data Snowflake atau Salesforce Data Cloud Anda. Tentukan OAuth sebagai metode otentikasi dan masukkan detail koneksi Anda.

Anda sekarang harus dapat mengimpor data dari database Anda di Snowflake atau Salesforce Data Cloud.

Connect ke platform SaaS

Anda dapat mengimpor data dari Snowflake dan lebih dari 40 platform SaaS eksternal lainnya. Untuk daftar lengkap konektor, lihat tabel diImpor data ke Canvas.

> **Note**
>
> Anda hanya dapat mengimpor data tabular, seperti tabel data, dari platform SaaS.

Gunakan Snowflake dengan Canvas

Snowflake adalah layanan penyimpanan data dan analitik, dan Anda dapat mengimpor data Anda dari Snowflake ke Canvas. SageMaker Untuk informasi lebih lanjut tentang Snowflake, lihat dokumentasi Snowflake.

Anda dapat mengimpor data dari akun Snowflake Anda dengan melakukan hal berikut:

1. Buat koneksi ke database Snowflake.
2. Pilih data yang Anda impor dengan menyeret dan menjatuhkan tabel dari menu navigasi kiri ke editor.
3. Impor data.

Anda dapat menggunakan editor Snowflake untuk menyeret kumpulan data ke panel impor dan mengimpornya ke Canvas. SageMaker Untuk kontrol lebih lanjut atas nilai yang dikembalikan dalam kumpulan data, Anda dapat menggunakan yang berikut ini:

- Kueri SQL
- Gabungan

Kueri SQL memberi Anda kemampuan untuk menyesuaikan cara Anda mengimpor nilai dalam kumpulan data. Misalnya, Anda dapat menentukan kolom yang dikembalikan dalam kumpulan data atau rentang nilai untuk kolom.

Anda dapat menggabungkan beberapa kumpulan data Snowflake ke dalam satu kumpulan data sebelum Anda mengimpor ke Canvas menggunakan SQL atau antarmuka Canvas. Anda dapat menyeret kumpulan data Anda dari Snowflake ke panel yang memberi Anda kemampuan untuk bergabung dengan kumpulan data, atau Anda dapat mengedit gabungan di SQL dan mengonversi SQL menjadi satu node. Anda dapat menggabungkan node lain ke node yang telah Anda konversi. Anda kemudian dapat menggabungkan kumpulan data yang telah Anda gabungkan menjadi satu node dan menggabungkan node ke dataset Snowflake yang berbeda. Terakhir, Anda dapat mengimpor data yang telah Anda pilih ke Canvas.

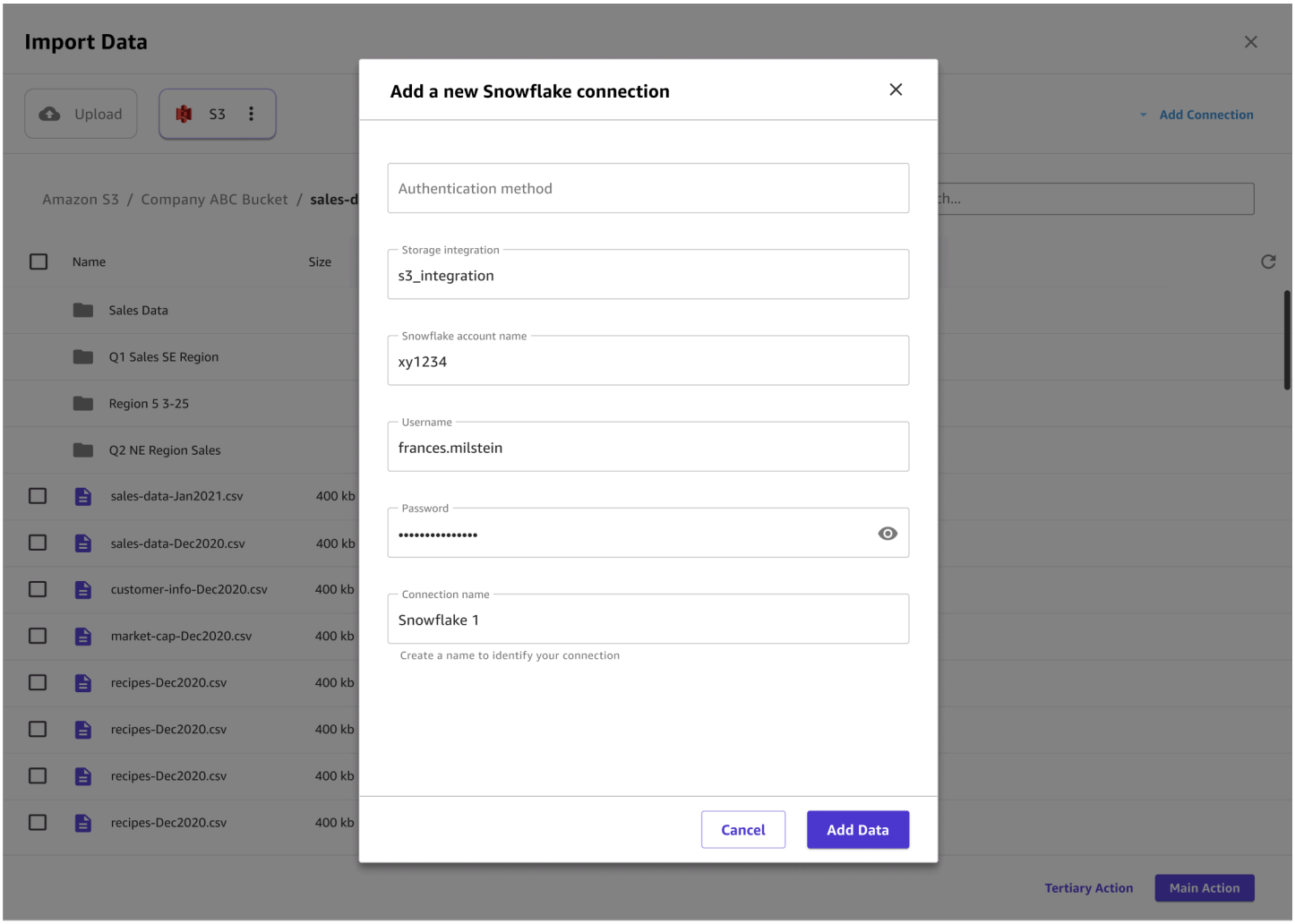
Gunakan prosedur berikut untuk mengimpor data dari Snowflake ke Amazon SageMaker Canvas.

1. Dalam aplikasi SageMaker Canvas, buka halaman Datasets.
2. Pilih Impor.
3. Untuk Sumber Data, buka menu tarik-turun dan pilih Snowflake.
4. Pilih Tambahkan koneksi.
5. Dalam kotak dialog Tambahkan koneksi Snowflake baru, tentukan kredensyal Snowflake Anda. Untuk metode Otentikasi, Anda dapat memilih Dasar - kata sandi nama pengguna, ARN atau OAuth. OAuth memungkinkan Anda mengautentikasi tanpa memberikan kata sandi tetapi memerlukan pengaturan tambahan. Untuk informasi selengkapnya tentang menyiapkan kredensyal OAuth untuk Snowflake, lihat. Siapkan koneksi ke sumber data dengan OAuth
6. Pilih Tambahkan koneksi.
7. Dari tab yang memiliki nama koneksi Anda, seret file.csv yang Anda impor ke panel Drag and drop to import table.
8. Opsional: Seret tabel tambahan ke panel impor. Anda dapat menggunakan antarmuka pengguna untuk bergabung dengan tabel. Untuk kekhususan lebih lanjut dalam bergabung Anda, pilih Edit di SQL.
9. Opsional: Jika Anda menggunakan SQL untuk menanyakan data, Anda dapat memilih Konteks untuk menambahkan konteks ke koneksi dengan menentukan nilai untuk hal berikut:

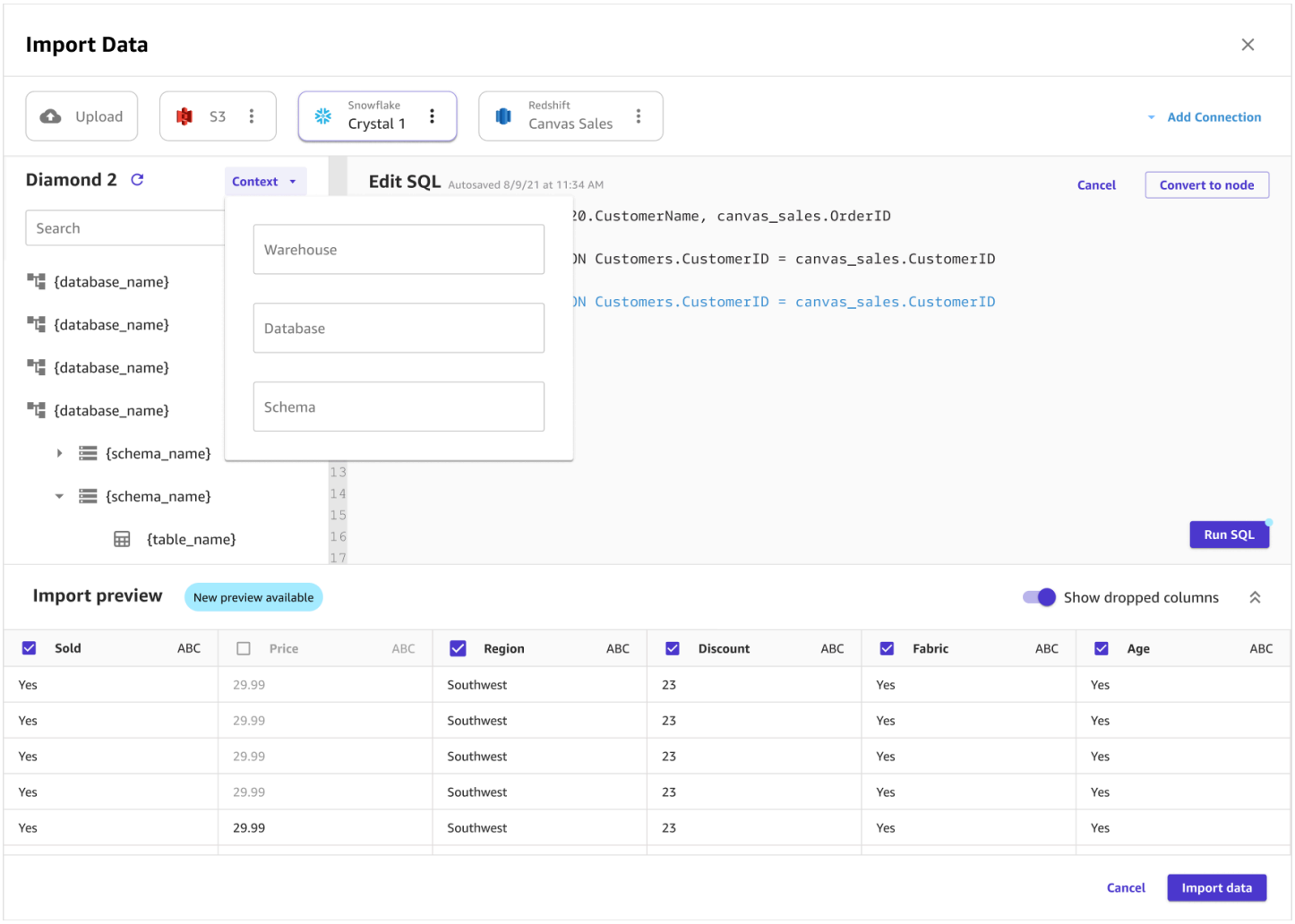
    - Gudang
    - Basis data
    - Skema

    Menambahkan konteks ke koneksi membuatnya lebih mudah untuk menentukan kueri future.
10. Pilih Impor data.

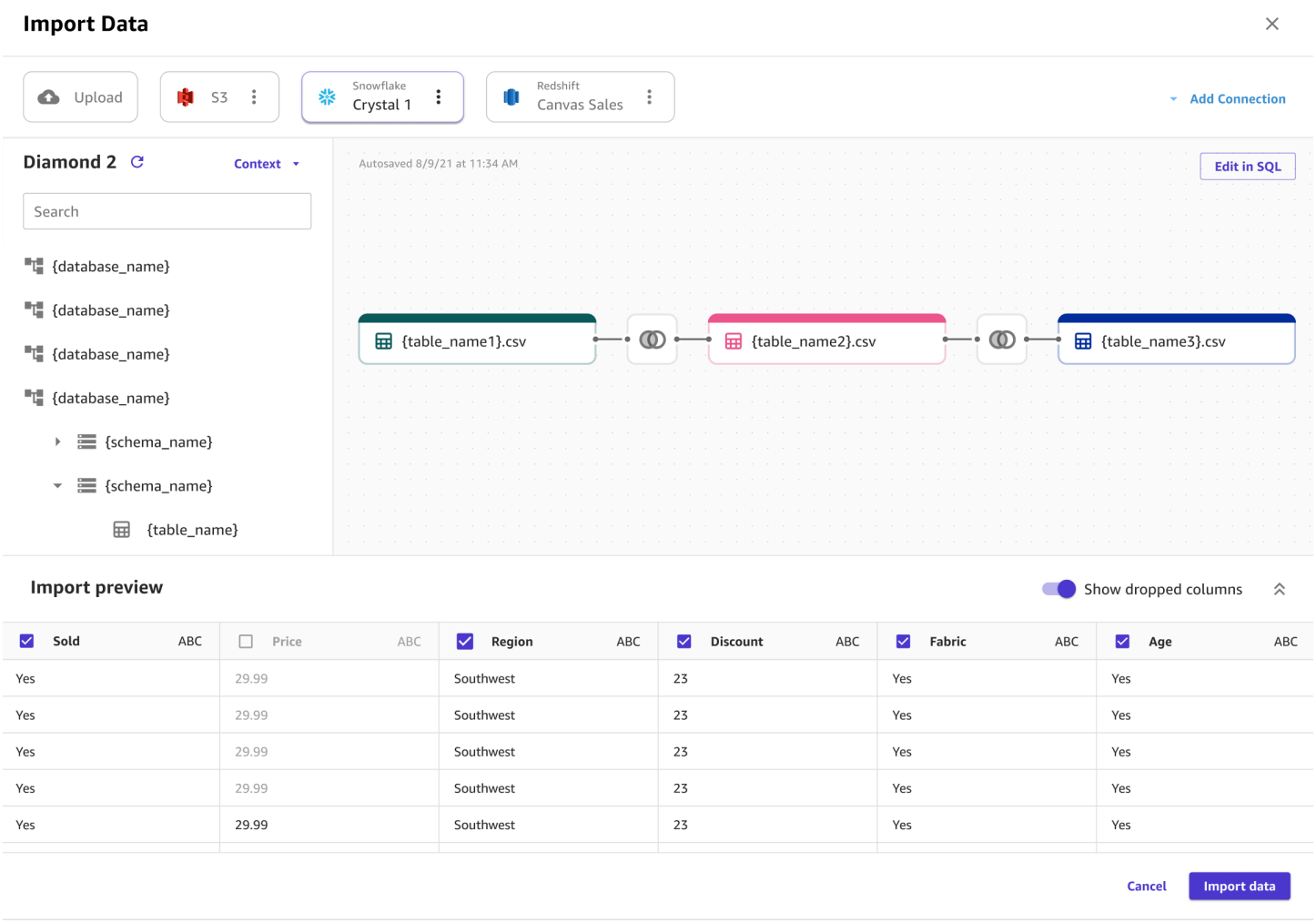
Gambar berikut menunjukkan contoh bidang yang ditentukan untuk koneksi Snowflake.



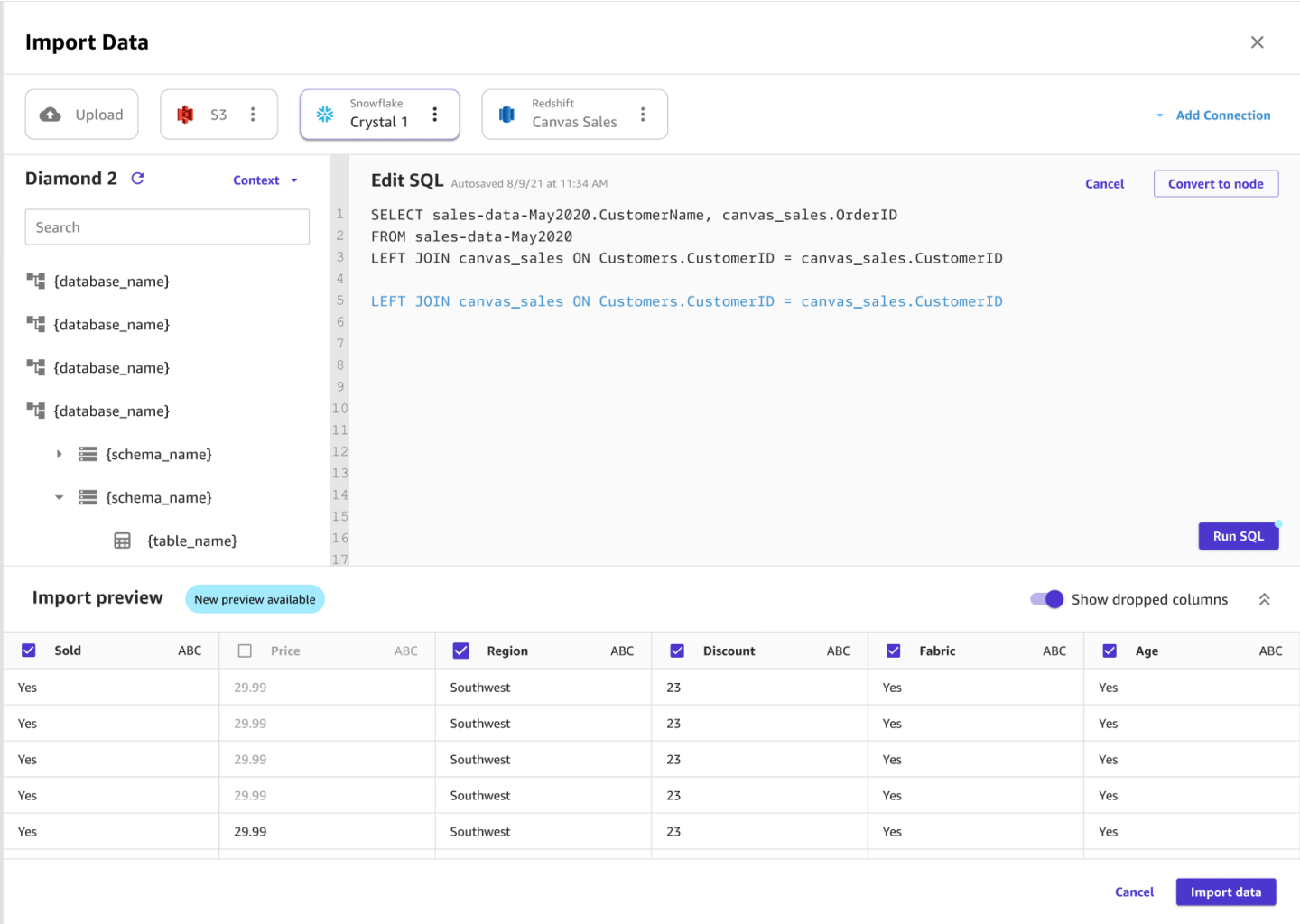
Gambar berikut menunjukkan halaman yang digunakan untuk menambahkan konteks ke koneksi.



Gambar berikut menunjukkan halaman yang digunakan untuk bergabung dengan dataset di Snowflake.



Gambar berikut menunjukkan query SQL yang digunakan untuk mengedit join di Snowflake.



Gunakan konektor SaaS dengan Canvas

> **Note**
>
> Untuk platform SaaS selain Snowflake, Anda hanya dapat memiliki satu koneksi per sumber data.

Sebelum Anda dapat mengimpor data dari platform SaaS, administrator Anda harus mengautentikasi dan membuat koneksi ke sumber data. Untuk informasi selengkapnya tentang cara administrator membuat koneksi dengan platform SaaS, lihat Mengelola koneksi AppFlow Amazon di Panduan Pengguna AppFlow Amazon.

Jika Anda seorang administrator yang memulai Amazon AppFlow untuk pertama kalinya, lihat Memulai di Panduan AppFlow Pengguna Amazon.

Untuk mengimpor data dari platform SaaS, Anda dapat mengikuti Impor data tabular prosedur standar, yang menunjukkan cara mengimpor kumpulan data tabular ke Canvas.

Bergabunglah dengan data yang telah Anda impor ke SageMaker Canvas

> **Note**
>
> Anda hanya dapat membuat gabungan untuk kumpulan data tabular di Canvas. SageMaker

Anda dapat menggunakan Amazon SageMaker Canvas untuk menggabungkan beberapa kumpulan data ke dalam satu kumpulan data. Gabungan menggabungkan dua kumpulan data. Secara default, SageMaker Canvas secara otomatis bergabung dengan kumpulan data pada nama kolom yang cocok. Opsi untuk menggabungkan beberapa kumpulan data mungkin memberi Anda kemampuan untuk mendapatkan lebih banyak wawasan dari model yang Anda buat.

Anda dapat membuat gabungan berikut untuk kumpulan data Anda:

- Inner - Mengembalikan dataset dengan nilai yang cocok di kedua dataset.
- Kiri - Mengembalikan dataset yang memiliki:
  - Semua baris dari kumpulan data di sebelah kiri gabungan.
  - Semua baris dari kumpulan data di sebelah kanan gabungan yang memiliki nilai yang cocok dengan kolom di sebelah kiri gabungan.
- Kanan - Mengembalikan dataset yang memiliki:
  - Semua baris dari kumpulan data di sebelah kanan gabungan.
  - Semua baris dari kumpulan data di sebelah kiri gabungan yang memiliki nilai yang cocok dengan kolom di sebelah kanan gabungan.
- Luar - Mengembalikan semua baris ketika ada kecocokan di dataset kiri atau kanan. Dataset dari gabungan luar mungkin memiliki nilai nol yang mungkin diperhitungkan SageMaker Canvas saat Anda membuat model.

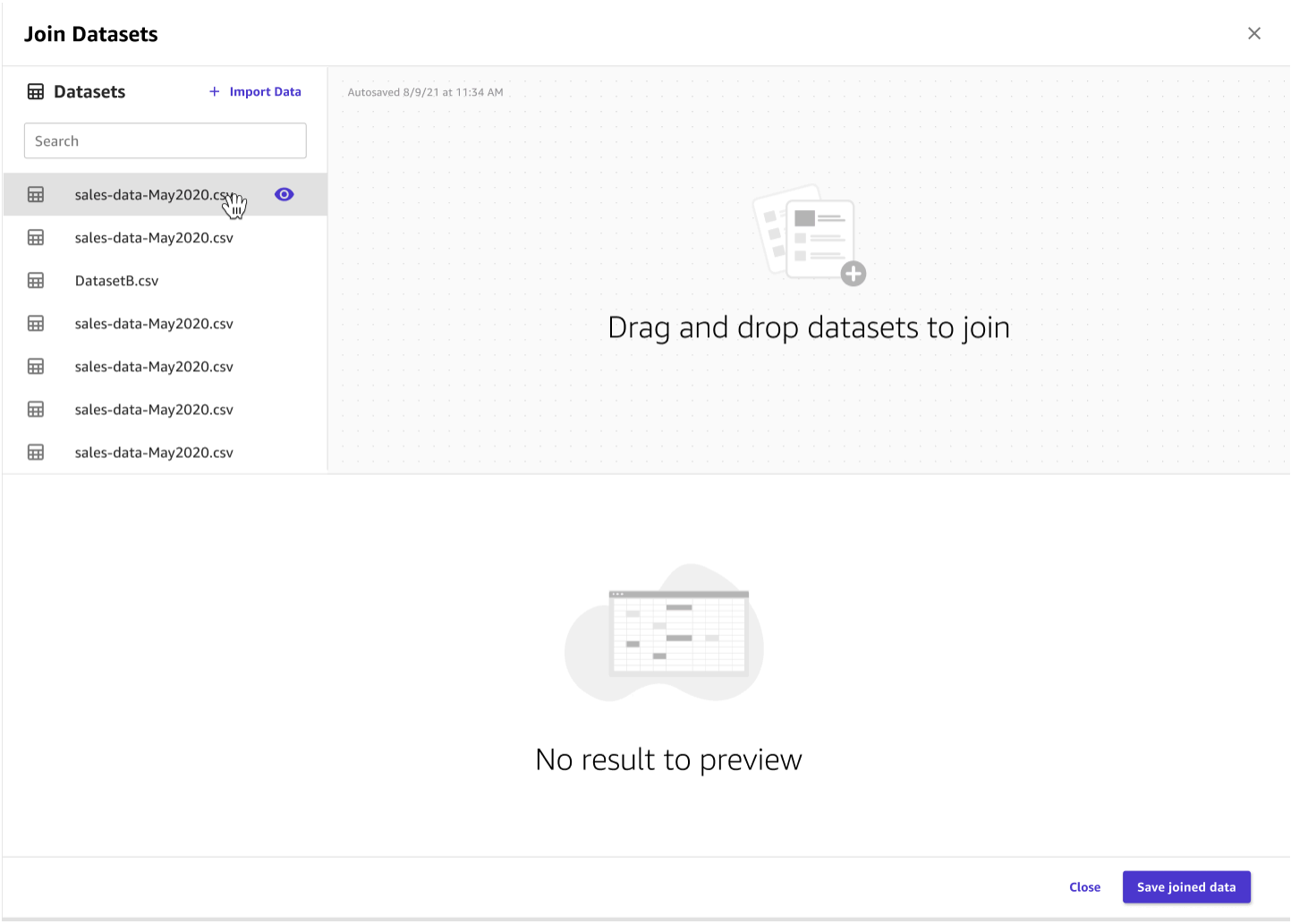
Gunakan prosedur berikut untuk bergabung dengan kumpulan data Anda.

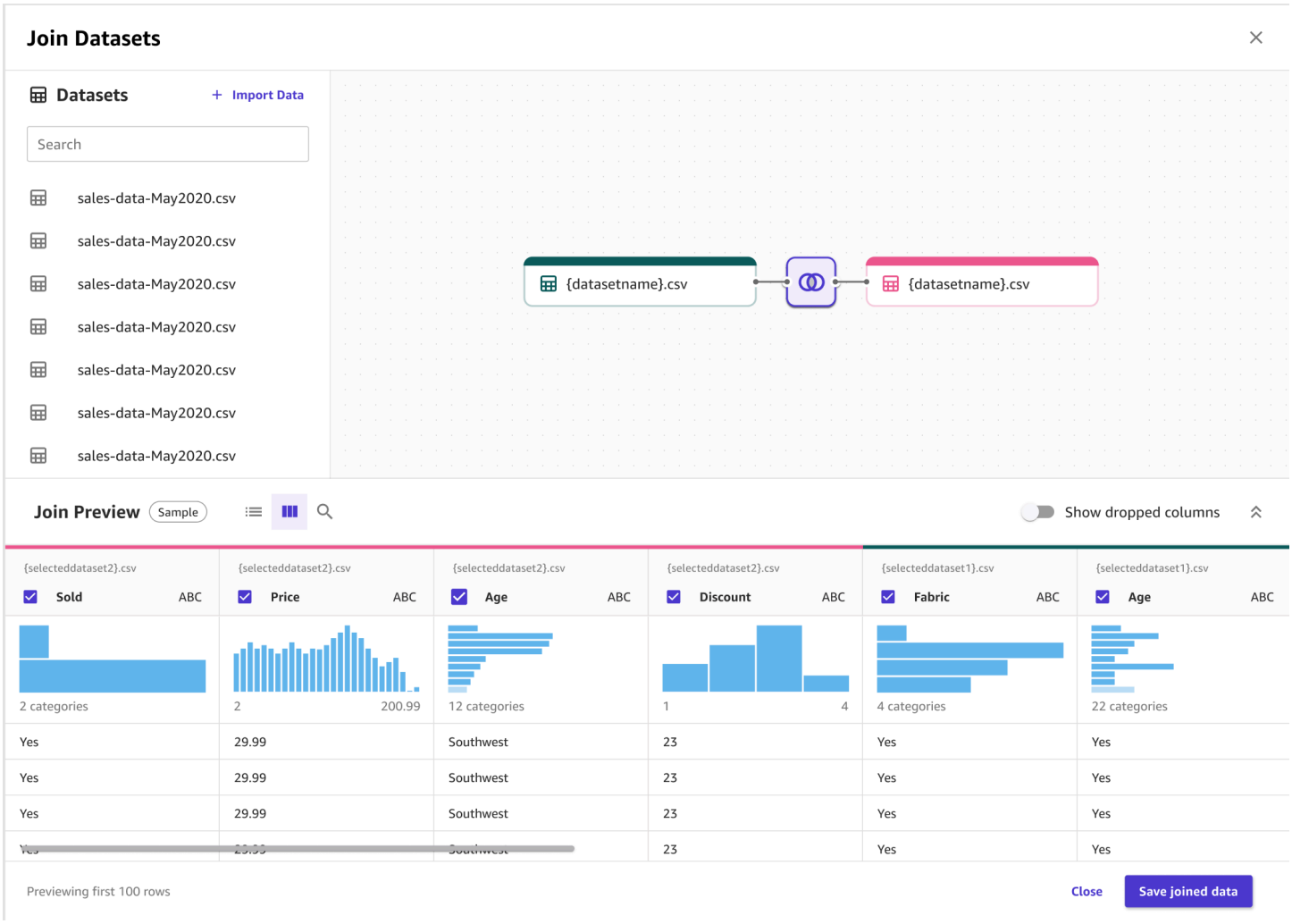
Untuk bergabung dengan kumpulan data, lakukan hal berikut.

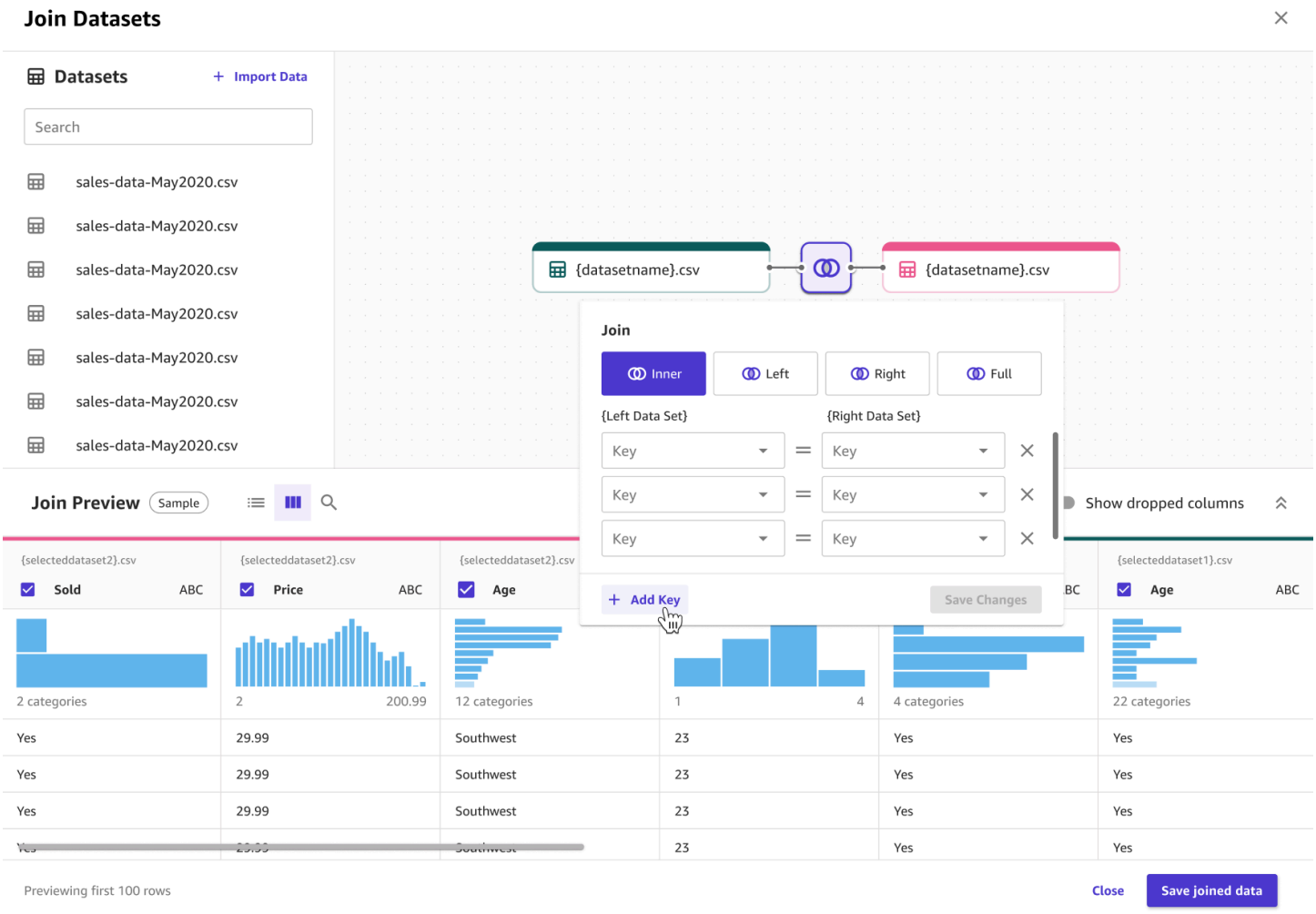
1. Arahkan ke halaman Datasets.
2. Pilih Gabung data.

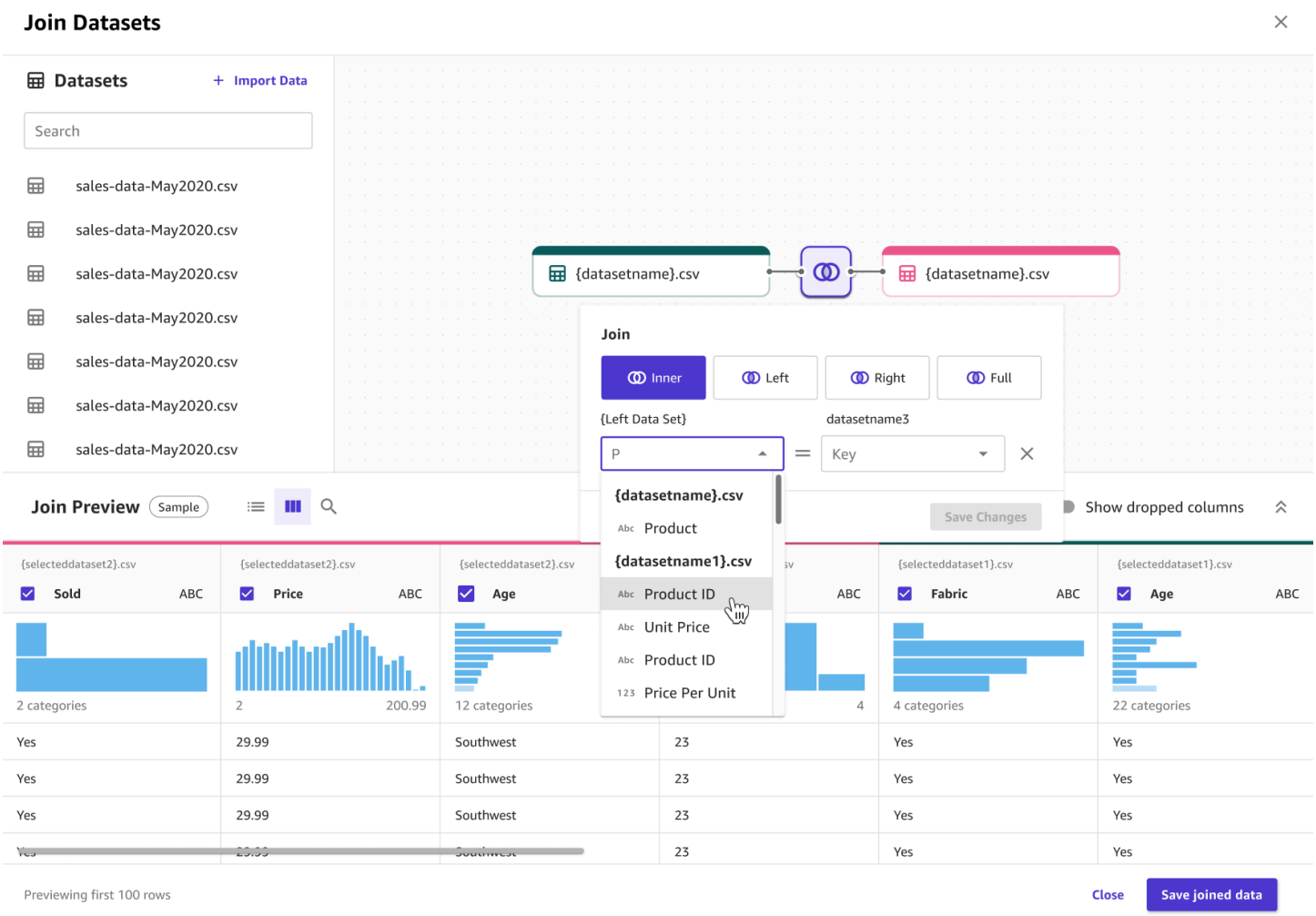
3. Seret dan lepas kumpulan data yang Anda gabungkan ke kotak Seret dan lepas kumpulan data untuk bergabung.
4. Konfigurasikan bergabung. Amazon SageMaker Canvas menunjukkan pratinjau data yang digabungkan setelah Anda mengonfigurasinya.
5. Pilih Simpan data yang digabungkan untuk menyimpan output dari gabungan.

Gambar berikut menunjukkan alur kerja dari prosedur sebelumnya.




Join Datasets
Datasets
+ Import Data
Search
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
{datasetname}.csv
{datasetname}.csv
Join Preview
Sample
Show dropped columns
{selecteddataset2}.csv
Sold ABC
2 categories
{selecteddataset2}.csv
Price ABC
2 200.99
{selecteddataset2}.csv
Age ABC
12 categories
{selecteddataset2}.csv
Discount ABC
1 4
{selecteddataset1}.csv
Fabric ABC
4 categories
{selecteddataset1}.csv
Age ABC
22 categories
Yes 29.99 Southwest 23 Yes Yes
Yes 29.99 Southwest 23 Yes Yes
Yes 29.99 Southwest 23 Yes Yes
23 Yes Yes
Previewing first 100 rows
Close
Save joined data


Join Datasets
Datasets
Import Data
Search
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
sales-data-May2020.csv
{datasetname}.csv
{datasetname}.csv
Join
Inner
Left
Right
Full
{Left Data Set}
{Right Data Set}
Key
Key
Key
Key
Key
Key
Add Key
Save Changes
Join Preview
Sample
Show dropped columns
{selecteddataset2}.csv
Sold
ABC
Price
Age
{selecteddataset1}.csv
2 categories
2
200.99
12 categories
1
4
4 categories
22 categories
Yes
29.99
Southwest
23
Previewing first 100 rows
Close
Save joined data



Gunakan kumpulan data sampel

SageMaker Canvas menyediakan kumpulan data sampel yang menangani kasus penggunaan unik sehingga Anda dapat mulai membangun, melatih, dan memvalidasi model dengan cepat tanpa menulis kode apa pun. Kasus penggunaan yang terkait dengan kumpulan data ini menyoroti kemampuan SageMaker Canvas, dan Anda dapat memanfaatkan kumpulan data ini untuk memulai pembuatan model. Anda dapat menemukan kumpulan data sampel di halaman Datasets aplikasi Canvas Anda. SageMaker

Contoh kumpulan data

Kumpulan data berikut adalah sampel yang disediakan SageMaker Canvas secara default. Kumpulan data ini mencakup kasus penggunaan seperti memprediksi harga rumah, default pinjaman, dan penerimaan kembali untuk pasien diabetes; memperkirakan penjualan; memprediksi kegagalan mesin untuk merampingkan pemeliharaan prediktif di unit manufaktur; dan menghasilkan prediksi

rantai pasokan untuk transportasi dan logistik. Kumpulan data disimpan di `sample_dataset` folder di bucket Amazon S3 default SageMaker yang dibuat untuk akun Anda di Wilayah.

- canvas-sample-diabetic-readmission.csv: Dataset ini berisi data historis termasuk lebih dari lima belas fitur dengan hasil pasien dan rumah sakit. Anda dapat menggunakan dataset ini untuk memprediksi apakah pasien diabetes berisiko tinggi kemungkinan akan diterima kembali ke rumah sakit dalam waktu 30 hari setelah keluar, setelah 30 hari, atau tidak sama sekali. Gunakan kolom yang diterima merah sebagai kolom target, dan gunakan tipe model prediksi kategori 3+ dengan kumpulan data ini. Untuk mempelajari lebih lanjut tentang cara membuat model dengan kumpulan data ini, lihat halaman lokakarya SageMaker Canvas. Dataset ini diperoleh dari UCI Machine Learning Repository.
- canvas-sample-housing.csv: Dataset ini berisi data tentang karakteristik yang terkait dengan harga perumahan tertentu. Anda dapat menggunakan dataset ini untuk memprediksi harga rumah. Gunakan kolom median_house_value sebagai kolom target, dan gunakan tipe model prediksi numerik dengan kumpulan data ini. Untuk mempelajari lebih lanjut tentang membuat model dengan kumpulan data ini, lihat halaman lokakarya SageMaker Canvas. Ini adalah dataset perumahan California yang diperoleh dari StatLib repositori.
- canvas-sample-loans.csv: Dataset ini berisi data pinjaman lengkap untuk semua pinjaman yang dikeluarkan dari 2007-2011, termasuk status pinjaman saat ini dan informasi pembayaran terbaru. Anda dapat menggunakan dataset ini untuk memprediksi apakah pelanggan akan membayar kembali pinjaman. Gunakan kolom loan_status sebagai kolom target, dan gunakan tipe model prediksi kategori 3+ dengan kumpulan data ini. Untuk mempelajari lebih lanjut tentang cara membuat model dengan kumpulan data ini, lihat halaman lokakarya SageMaker Canvas. Data ini menggunakan LendingClub data yang diperoleh dari Kaggle.
- canvas-sample-maintenance.csv: Dataset ini berisi data tentang karakteristik yang terkait dengan tipe kegagalan pemeliharaan tertentu. Anda dapat menggunakan kumpulan data ini untuk memprediksi kegagalan mana yang akan terjadi di masa depan. Gunakan kolom Jenis Kegagalan sebagai kolom target, dan gunakan tipe model prediksi kategori 3+ dengan kumpulan data ini. Untuk mempelajari lebih lanjut tentang cara membuat model dengan kumpulan data ini, lihat halaman lokakarya SageMaker Canvas. Dataset ini diperoleh dari UCI Machine Learning Repository.
- canvas-sample-shipping-logs.csv: Dataset ini berisi data pengiriman lengkap untuk semua produk yang dikirim, termasuk perkiraan waktu pengiriman prioritas, operator, dan asal. Anda dapat menggunakan kumpulan data ini untuk memprediksi perkiraan waktu kedatangan pengiriman dalam jumlah hari. Gunakan ActualShippingDayskolom sebagai kolom target, dan gunakan tipe model prediksi numerik dengan kumpulan data ini. Untuk mempelajari lebih lanjut tentang cara

membuat model dengan data ini, lihat halaman lokakarya SageMaker Canvas. Ini adalah kumpulan data sintetis yang dibuat oleh Amazon.

- canvas-sample-sales-forecasting.csv: Dataset ini berisi data penjualan deret waktu historis untuk toko ritel. Anda dapat menggunakan dataset ini untuk memperkirakan penjualan untuk toko ritel tertentu. Gunakan kolom penjualan sebagai kolom target, dan gunakan tipe model peramalan deret waktu dengan kumpulan data ini. Untuk mempelajari lebih lanjut tentang cara membuat model dengan kumpulan data ini, lihat halaman lokakarya SageMaker Canvas. Ini adalah kumpulan data sintetis yang dibuat oleh Amazon.

Impor ulang kumpulan data sampel yang dihapus

Jika Anda tidak lagi ingin menggunakan kumpulan data sampel, Anda dapat menghapusnya dari halaman Datasets aplikasi Canvas Anda. SageMaker Namun, kumpulan data ini masih disimpan di bucket Amazon S3 yang Anda tentukan sebagai lokasi penyimpanan Canvas, sehingga Anda selalu dapat mengaksesnya nanti.

Jika Anda menggunakan bucket Amazon S3 default, nama bucket mengikuti polanya. `sagemaker-{region}-{account ID}` Anda dapat menemukan kumpulan data sampel di jalur direktori. `Canvas/sample_dataset`

Jika Anda menghapus kumpulan data sampel dari aplikasi SageMaker Canvas Anda dan ingin mengakses kumpulan data sampel lagi, gunakan prosedur berikut.

1. Arahkan ke halaman Datasets di aplikasi SageMaker Canvas Anda.
2. Pilih Impor data.
3. Dari daftar bucket Amazon S3, pilih bucket yang merupakan lokasi penyimpanan Canvas Anda. Jika menggunakan bucket Amazon S3 default yang SageMaker dibuat, bucket tersebut mengikuti pola penamaan. `sagemaker-{region}-{account ID}`
4. Pilih folder Canvas.
5. Pilih folder sample_dataset, yang berisi semua kumpulan data sampel untuk Canvas. SageMaker
6. Pilih kumpulan data yang ingin Anda impor, lalu pilih Impor data.

Siapkan data

Gunakan Amazon SageMaker Data Wrangler di Amazon SageMaker Canvas untuk menyiapkan, menyesuaikan, dan menganalisis data Anda. Anda dapat mengintegrasikan alur persiapan

data Wrangler Data ke dalam alur kerja machine learning (ML) Anda untuk menyederhanakan dan merampingkan pra-pemrosesan data dan rekayasa fitur menggunakan sedikit atau tanpa pengkodean. Anda juga dapat menambahkan skrip dan transformasi Python Anda sendiri untuk menyesuaikan alur kerja.

- Aliran Data - Buat aliran data untuk menentukan serangkaian langkah persiapan data ML. Anda dapat menggunakan alur untuk menggabungkan kumpulan data dari sumber data yang berbeda, mengidentifikasi jumlah dan jenis transformasi yang ingin Anda terapkan ke kumpulan data, dan menentukan alur kerja persiapan data yang dapat diintegrasikan ke dalam pipeline ML.
- Transform - Bersihkan dan ubah dataset Anda menggunakan transformasi standar seperti string, vektor, dan alat pemformatan data numerik. Faturisasi data Anda menggunakan transformasi seperti penyematan teks dan tanggal/waktu serta pengkodean kategoris.
- Hasilkan Wawasan Data — Secara otomatis memverifikasi kualitas data dan mendeteksi kelainan pada data Anda dengan Laporan Kualitas dan Wawasan Data Wrangler Data.
- Analisis — Analisis fitur dalam kumpulan data Anda di setiap titik dalam alur Anda. Data Wrangler mencakup alat visualisasi data bawaan seperti plot pencar dan histogram, serta alat analisis data seperti analisis kebocoran target dan pemodelan cepat untuk memahami korelasi fitur.
- Ekspor - Ekspor alur kerja persiapan data Anda ke lokasi yang berbeda. Berikut ini adalah contoh lokasi:
  - bucket Amazon Simple Storage Service (Amazon S3)
  - Amazon SageMaker Feature Store — Simpan fitur dan datanya di toko terpusat.
- Mengotomatiskan persiapan data — Buat alur kerja pembelajaran mesin dari alur data Anda.
  - Amazon SageMaker Model Building Pipelines — Bangun alur kerja yang mengelola persiapan SageMaker data, pelatihan model, dan pekerjaan penerapan model Anda.
  - Pipa inferensi serial — Buat pipeline inferensi serial dari aliran data Anda. Gunakan untuk membuat prediksi pada data baru.
  - Skrip Python — Simpan data dan transformasinya dalam skrip Python untuk alur kerja kustom Anda.

Buat Alur Data

Gunakan aliran Data Wrangler di SageMaker Canvas, atau aliran data, untuk membuat dan memodifikasi pipeline persiapan data. Kumpulan data, transformasi, dan analisis yang Anda gunakan dalam aliran data direpresentasikan sebagai langkah.

Impor data ke dalam aliran data

Untuk memulai menggunakan aliran data, impor data Anda ke dalamnya. Untuk menggunakan kumpulan data yang lebih besar dari 5 GB, Anda harus mengimpor data langsung dari sumber data alih-alih menggunakan dataset SageMaker Canvas.

Gunakan prosedur berikut untuk mengimpor data Anda ke dalam aliran data.

Untuk mengimpor data Anda ke dalam aliran data

1. Buka SageMaker kanvas.
2. Di navigasi sebelah kiri, pilih.
   
3. Pilih Alur data.
4. Pilih Buat.
5. (Opsional) Untuk nama aliran data, tentukan nama untuk aliran data.
6. • Untuk menggunakan dataset SageMaker Canvas yang sudah diimpor ke SageMaker Canvas, pilih Pilih set data yang ada.
     a. Pilih jenis dataset.
     b. Pilih dataset SageMaker Canvas.
   • Untuk mengimpor data langsung dari sumber data, pilih Impor data.
     a. Untuk Sumber Data, pilih sumber data.
     b. Connect ke sumber data untuk menelusuri data dan mengimpor dataset. Untuk informasi tentang menghubungkan ke sumber data atau mengimpor data, lihat halaman berikut:
        • Impor data ke Canvas
        • Connect ke sumber data
     c. Pilih Impor data
     d. (Opsional) Jika baris pertama kumpulan data Anda adalah header, pilih Gunakan baris pertama sebagai header.
     e. Pilih Impor data.

UI Aliran Data

Saat Anda mengimpor dataset, dataset asli muncul di aliran data dan diberi nama Source. SageMaker Canvas secara otomatis menyimpulkan jenis setiap kolom dalam kumpulan data Anda dan membuat kerangka data baru bernama Tipe data. Anda dapat memilih bingkai ini untuk memperbarui tipe data yang disimpulkan.

Setiap kali Anda menambahkan langkah transformasi, Anda membuat kerangka data baru. Ketika beberapa langkah transformasi (selain Join atau Concatenate) ditambahkan ke kumpulan data yang sama, mereka ditumpuk.

Bergabunglah dan Gabungkan buat langkah mandiri yang berisi kumpulan data baru yang digabungkan atau digabungkan.

Tambahkan Langkah ke Alur Data Anda

Pilih + di sebelah kumpulan data apa pun atau langkah yang ditambahkan sebelumnya, lalu pilih salah satu opsi berikut:

- Mengedit tipe data (Hanya untuk langkah tipe data): Jika Anda belum menambahkan transformasi apa pun ke langkah Jenis data, Anda dapat memilih Edit tipe data untuk memperbarui tipe data yang disimpulkan oleh Wrangler Data saat mengimpor kumpulan data Anda.
- Tambahkan transformasi: Menambahkan langkah transformasi baru. Lihat Mengubah data untuk mempelajari lebih lanjut tentang transformasi data yang dapat Anda tambahkan.
- Tambahkan analisis: Menambahkan analisis. Anda dapat menggunakan opsi ini untuk menganalisis data Anda kapan saja dalam aliran data. Lihat Lakukan analisis data eksplorasi (EDA) untuk mempelajari lebih lanjut tentang analisis yang dapat Anda tambahkan.
- Gabung: Bergabung dengan dua kumpulan data dan menambahkan kumpulan data yang dihasilkan ke aliran data. Untuk mempelajari selengkapnya, lihat Bergabunglah dengan Datasets.
- Menggabungkan: Menggabungkan dua kumpulan data dan menambahkan kumpulan data yang dihasilkan ke aliran data. Untuk mempelajari selengkapnya, lihat Menggabungkan Dataset.

Hapus Langkah dari Alur Data Anda

Untuk menghapus langkah, pilih + di sebelah langkah dan pilih Hapus. Jika node adalah node yang memiliki input tunggal, Anda hanya menghapus langkah yang Anda pilih. Menghapus langkah yang memiliki satu input tidak menghapus langkah-langkah yang mengikutinya. Jika Anda menghapus

langkah untuk sumber, bergabung, atau menggabungkan node, semua langkah yang mengikutinya juga dihapus.

Untuk menghapus langkah dari tumpukan langkah, pilih tumpukan dan kemudian pilih langkah yang ingin Anda hapus.

Anda dapat menggunakan salah satu prosedur berikut untuk menghapus langkah tanpa menghapus langkah hilir.

Delete a step in the Data Wrangler flow

Anda dapat menghapus langkah individual untuk node dalam aliran data Anda yang memiliki satu input. Anda tidak dapat menghapus langkah individual untuk sumber, bergabung, dan menggabungkan node.

Gunakan prosedur berikut untuk menghapus langkah dalam aliran Data Wrangler.

1. Pilih kelompok langkah yang memiliki langkah yang Anda hapus.
2. Pilih ikon di sebelah langkah.
3. Pilih Hapus langkah.

Delete a step in the table view

Gunakan prosedur berikut untuk menghapus langkah dalam tampilan tabel.

Anda dapat menghapus langkah individual untuk node dalam aliran data Anda yang memiliki satu input. Anda tidak dapat menghapus langkah individual untuk sumber, bergabung, dan menggabungkan node.

1. Pilih langkah dan buka tampilan tabel untuk langkah tersebut.
2. Gerakkan kursor Anda ke atas langkah sehingga ikon elipsis muncul.
3. Pilih ikon di sebelah langkah.
4. Pilih Hapus.

Lakukan analisis data eksplorasi (EDA)

Data Wrangler mencakup analisis bawaan yang membantu Anda menghasilkan visualisasi dan analisis data dalam beberapa klik. Anda juga dapat membuat analisis kustom menggunakan kode Anda sendiri.

Anda menambahkan analisis ke kerangka data dengan memilih langkah dalam aliran data Anda, lalu memilih Tambahkan analisis. Untuk mengakses analisis yang telah Anda buat, pilih langkah yang berisi analisis, dan pilih analisis.

Semua analisis dihasilkan menggunakan 20.000 baris dataset Anda.

Anda dapat menambahkan analisis berikut ke kerangka data:

- Visualisasi data, termasuk histogram dan plot pencar.
- Ringkasan singkat kumpulan data Anda, termasuk jumlah entri, nilai minimum dan maksimum (untuk data numerik), dan kategori yang paling sering dan paling jarang (untuk data kategoris).
- Model cepat kumpulan data, yang dapat digunakan untuk menghasilkan skor penting untuk setiap fitur.
- Laporan kebocoran target, yang dapat Anda gunakan untuk menentukan apakah satu atau lebih fitur berkorelasi kuat dengan fitur target Anda.
- Visualisasi khusus menggunakan kode Anda sendiri.

Gunakan bagian berikut untuk mempelajari lebih lanjut tentang opsi ini.

Dapatkan wawasan tentang kualitas data dan data

Gunakan Laporan Kualitas Data dan Wawasan untuk melakukan analisis data yang telah Anda impor ke Data Wrangler. Kami menyarankan Anda membuat laporan setelah Anda mengimpor dataset Anda. Anda dapat menggunakan laporan untuk membantu Anda membersihkan dan memproses data Anda. Ini memberi Anda informasi seperti jumlah nilai yang hilang dan jumlah outlier. Jika Anda memiliki masalah dengan data Anda, seperti kebocoran target atau ketidakseimbangan, laporan wawasan dapat membawa masalah tersebut ke perhatian Anda.

Gunakan prosedur berikut untuk membuat laporan Kualitas Data dan Wawasan. Ini mengasumsikan bahwa Anda telah mengimpor dataset ke dalam aliran Data Wrangler Anda.

Untuk membuat laporan Kualitas Data dan Wawasan

1. Pilih + di sebelah node dalam alur Data Wrangler Anda.
2. Pilih Dapatkan wawasan data.
3. Untuk nama Analisis, tentukan nama untuk laporan wawasan.
4. (Opsional) Untuk kolom Target, tentukan kolom target.

5. Untuk jenis Masalah, tentukan Regresi atau Klasifikasi.
6. Untuk ukuran Data, tentukan salah satu dari berikut ini:

   - 20 K - Menggunakan 20000 baris pertama dari kumpulan data yang telah Anda impor untuk membuat laporan.
   - Seluruh kumpulan data — Menggunakan seluruh kumpulan data yang telah Anda impor untuk membuat laporan.

   > **Note**
   >
   > Membuat laporan Kualitas Data dan Wawasan di seluruh kumpulan data menggunakan pekerjaan SageMaker pemrosesan Amazon. Pekerjaan SageMaker pemrosesan menyediakan sumber daya komputasi tambahan yang diperlukan untuk mendapatkan wawasan untuk semua data Anda. Untuk informasi selengkapnya tentang SageMaker memproses pekerjaan, lihatMemproses data.

7. Pilih Buat.

Topik berikut menunjukkan bagian dari laporan:

Topik

- Ringkasan
- Kolom target
- Model cepat
- Ringkasan fitur
- Sampel
- Ketentuan

Anda dapat mengunduh laporan atau melihatnya secara online. Untuk mengunduh laporan, pilih tombol unduh di sudut kanan atas layar.

Ringkasan

Laporan wawasan memiliki ringkasan singkat dari data yang mencakup informasi umum seperti nilai yang hilang, nilai tidak valid, jenis fitur, jumlah outlier, dan banyak lagi. Ini juga dapat mencakup

peringatan tingkat keparahan tinggi yang menunjukkan kemungkinan masalah dengan data. Kami menyarankan Anda menyelidiki peringatan tersebut.

Kolom target

Saat Anda membuat Laporan Kualitas dan Wawasan Data, Data Wrangler memberi Anda opsi untuk memilih kolom target. Kolom target adalah kolom yang Anda coba prediksi. Saat Anda memilih kolom target, Data Wrangler secara otomatis membuat analisis kolom target. Ini juga memberi peringkat fitur dalam urutan kekuatan prediksi mereka. Saat memilih kolom target, Anda harus menentukan apakah Anda mencoba memecahkan masalah regresi atau klasifikasi.

Untuk klasifikasi, Data Wrangler menunjukkan tabel dan histogram dari kelas yang paling umum. Kelas adalah kategori. Ini juga menyajikan pengamatan, atau baris, dengan nilai target yang hilang atau tidak valid.

Untuk regresi, Data Wrangler menunjukkan histogram semua nilai di kolom target. Ini juga menyajikan pengamatan, atau baris, dengan nilai target yang hilang, tidak valid, atau outlier.

Model cepat

Model Cepat memberikan perkiraan kualitas prediksi yang diharapkan dari model yang Anda latih pada data Anda.

Data Wrangler membagi data Anda menjadi lipatan pelatihan dan validasi. Ini menggunakan 80% sampel untuk pelatihan dan 20% dari nilai untuk validasi. Untuk klasifikasi, sampel dibagi bertingkat. Untuk pemisahan bertingkat, setiap partisi data memiliki rasio label yang sama. Untuk masalah klasifikasi, penting untuk memiliki rasio label yang sama antara lipatan pelatihan dan klasifikasi. Data Wrangler melatih model XGBoost dengan hyperparameters default. Ini berlaku penghentian awal pada data validasi dan melakukan preprocessing fitur minimal.

Untuk model klasifikasi, Data Wrangler mengembalikan ringkasan model dan matriks kebingungan.

Untuk mempelajari lebih lanjut tentang informasi yang dikembalikan oleh ringkasan model klasifikasi, lihatKetentuan.

Matriks kebingungan memberi Anda informasi berikut:

- Berapa kali label yang diprediksi cocok dengan label sebenarnya.
- Berapa kali label yang diprediksi tidak cocok dengan label sebenarnya.

Label sebenarnya mewakili pengamatan aktual dalam data Anda. Misalnya, jika Anda menggunakan model untuk mendeteksi transaksi penipuan, label sebenarnya mewakili transaksi yang sebenarnya curang atau tidak curang. Label yang diprediksi mewakili label yang ditetapkan model Anda ke data.

Anda dapat menggunakan matriks kebingungan untuk melihat seberapa baik model memprediksi ada atau tidak adanya suatu kondisi. Jika Anda memprediksi transaksi penipuan, Anda dapat menggunakan matriks kebingungan untuk memahami sensitivitas dan kekhususan model. Sensitivitas mengacu pada kemampuan model untuk mendeteksi transaksi penipuan. Kekhususan mengacu pada kemampuan model untuk menghindari mendeteksi transaksi non-penipuan sebagai penipuan.

Ringkasan fitur

Saat Anda menentukan kolom target, Data Wrangler memesan fitur berdasarkan kekuatan prediksinya. Kekuatan prediksi diukur pada data setelah dibagi menjadi 80% pelatihan dan 20% lipatan validasi. Data Wrangler cocok dengan model untuk setiap fitur secara terpisah pada lipatan pelatihan. Ini menerapkan preprocessing fitur minimal dan mengukur kinerja prediksi pada data validasi.

Ini menormalkan skor ke kisaran [0,1]. Skor prediksi yang lebih tinggi menunjukkan kolom yang lebih berguna untuk memprediksi target sendiri. Skor yang lebih rendah menunjuk ke kolom yang tidak memprediksi kolom target.

Ini jarang untuk kolom yang tidak prediktif sendiri untuk menjadi prediktif ketika digunakan bersama-sama dengan kolom lain. Anda dapat dengan yakin menggunakan skor prediksi untuk menentukan apakah fitur dalam kumpulan data Anda bersifat prediktif.

Skor rendah biasanya menunjukkan fitur tersebut berlebihan. Skor 1 menyiratkan kemampuan prediksi sempurna, yang sering menunjukkan kebocoran target. Kebocoran target biasanya terjadi ketika kumpulan data berisi kolom yang tidak tersedia pada waktu prediksi. Misalnya, itu bisa menjadi duplikat dari kolom target.

Sampel

Data Wrangler memberikan informasi tentang apakah sampel Anda anomali atau jika ada duplikat dalam kumpulan data Anda.

Data Wrangler mendeteksi sampel anomali menggunakan algoritma hutan isolasi. Hutan isolasi mengaitkan skor anomali dengan setiap sampel (baris) dari kumpulan data. Skor anomali yang

rendah menunjukkan sampel anomali. Skor tinggi dikaitkan dengan sampel non-anomali. Sampel dengan skor anomali negatif biasanya dianggap anomali dan sampel dengan skor anomali positif dianggap non-anomali.

Ketika Anda melihat sampel yang mungkin anomali, kami sarankan Anda memperhatikan nilai-nilai yang tidak biasa. Misalnya, Anda mungkin memiliki nilai anomali yang dihasilkan dari kesalahan dalam mengumpulkan dan memproses data. Berikut ini adalah contoh sampel yang paling anomali menurut implementasi data Wrangler dari algoritma hutan isolasi. Sebaiknya gunakan pengetahuan domain dan logika bisnis saat Anda memeriksa sampel anomali.

Data Wrangler mendeteksi baris duplikat dan menghitung rasio baris duplikat dalam data Anda. Beberapa sumber data dapat menyertakan duplikat yang valid. Sumber data lain dapat memiliki duplikat yang menunjukkan masalah dalam pengumpulan data. Sampel duplikat yang dihasilkan dari pengumpulan data yang salah dapat mengganggu proses pembelajaran mesin yang mengandalkan pemisahan data menjadi pelatihan independen dan lipatan validasi.

Berikut ini adalah elemen laporan wawasan yang dapat dipengaruhi oleh sampel duplikat:

- Model cepat
- Estimasi daya prediksi
- Penyetelan hyperparameter otomatis

Anda dapat menghapus sampel duplikat dari kumpulan data menggunakan transformasi Drop duplikat di bawah Kelola baris. Data Wrangler menunjukkan baris yang paling sering diduplikasi.

Ketentuan

Berikut ini adalah definisi untuk istilah teknis yang digunakan dalam laporan wawasan data.

Feature types

Berikut ini adalah definisi untuk masing-masing jenis fitur:

- Numerik — Nilai numerik dapat berupa float atau bilangan bulat, seperti usia atau pendapatan. Model pembelajaran mesin mengasumsikan bahwa nilai numerik diurutkan dan jarak ditentukan di atasnya. Misalnya, 3 lebih dekat ke 4 daripada 10 dan 3 < 4 < 10.
- Categorical - Entri kolom milik satu set nilai unik, yang biasanya jauh lebih kecil dari jumlah entri di kolom. Misalnya, kolom dengan panjang 100 dapat berisi nilai unik`Dog`,`Cat`, dan`Mouse`. Nilai

bisa berupa numerik, teks, atau kombinasi keduanya. `Horse`,`House`,`8`,`Love`, dan semuanya `3.1` akan menjadi nilai yang valid dan dapat ditemukan di kolom kategoris yang sama. Model pembelajaran mesin tidak mengasumsikan urutan atau jarak pada nilai-nilai fitur kategoris, sebagai lawan dari fitur numerik, bahkan ketika semua nilai adalah angka.

- Biner — Fitur biner adalah jenis fitur kategoris khusus di mana kardinalitas himpunan nilai unik adalah 2.
- Teks - Kolom teks berisi banyak nilai unik non-numerik. Dalam kasus ekstrim, semua elemen kolom itu unik. Dalam kasus ekstrim, tidak ada dua entri yang sama.
- Datetime - Kolom datetime berisi informasi tentang tanggal atau waktu. Ini dapat memiliki informasi tentang tanggal dan waktu.

Feature statistics

Berikut ini adalah definisi untuk masing-masing statistik fitur:

- Kekuatan prediksi — Kekuatan prediksi mengukur seberapa berguna kolom dalam memprediksi target.
- Outlier (dalam kolom numerik) — Data Wrangler mendeteksi outlier menggunakan dua statistik yang kuat untuk outlier: median dan solid standard deviation (RSTD). RSTD diturunkan dengan memotong nilai fitur ke kisaran [5 persentil, 95 persentil] dan menghitung standar deviasi vektor yang terpotong. Semua nilai yang lebih besar dari median + 5* RSTD atau lebih kecil dari median - 5 * RSTD dianggap outlier.
- Skew (dalam kolom numerik) — Skew mengukur simetri distribusi dan didefinisikan sebagai momen ketiga distribusi dibagi dengan kekuatan ketiga dari standar deviasi. Kemiringan distribusi normal atau distribusi simetris lainnya adalah nol. Nilai positif menyiratkan bahwa ekor kanan distribusi lebih panjang dari ekor kiri. Nilai negatif menyiratkan bahwa ekor kiri distribusi lebih panjang dari ekor kanan. Sebagai aturan praktis, distribusi dianggap miring ketika nilai absolut kemiringan lebih besar dari 3.
- Kurtosis (dalam kolom numerik) — Kurtosis Pearson mengukur beratnya ekor distribusi. Ini didefinisikan sebagai momen keempat dari distribusi dibagi dengan kuadrat dari momen kedua. Kurtosis dari distribusi normal adalah 3. Nilai kurtosis yang lebih rendah dari 3 menyiratkan bahwa distribusi terkonsentrasi di sekitar rata-rata dan ekor lebih ringan dari ekor distribusi normal. Nilai kurtosis lebih tinggi dari 3 menyiratkan ekor atau outlier yang lebih berat.
- Nilai yang hilang - Objek seperti nol, string kosong, dan string yang hanya terdiri dari spasi putih dianggap hilang.

- Nilai yang valid untuk fitur numerik atau target regresi - Semua nilai yang dapat Anda lemparkan ke float terbatas valid. Nilai yang hilang tidak valid.
- Nilai yang valid untuk fitur kategoris, biner, atau teks, atau untuk target klasifikasi - Semua nilai yang tidak hilang valid.
- Fitur Datetime - Semua nilai yang dapat Anda transmisikan ke objek datetime valid. Nilai yang hilang tidak valid.
- Nilai tidak valid - Nilai yang hilang atau Anda tidak dapat mentransmisikan dengan benar. Misalnya, dalam kolom numerik, Anda tidak dapat mentransmisikan string `"six"` atau nilai null.

Quick model metrics for regression

Berikut ini adalah definisi untuk metrik model cepat:

- R2 atau koefisien determinasi) — R2 adalah proporsi variasi target yang diprediksi oleh model. R2 berada dalam kisaran [-infty, 1]. 1 adalah skor model yang memprediksi target dengan sempurna dan 0 adalah skor model sepele yang selalu memprediksi rata-rata target.
- MSE atau kesalahan kuadrat rata-rata — MSE berada dalam kisaran [0, infty]. 0 adalah skor model yang memprediksi target dengan sempurna.
- MAE atau kesalahan absolut rata-rata — MAE berada dalam kisaran [0, infty] di mana 0 adalah skor model yang memprediksi target dengan sempurna.
- RMSE atau kesalahan kuadrat rata-rata akar — RMSE berada dalam kisaran [0, infty] di mana 0 adalah skor model yang memprediksi target dengan sempurna.
- Kesalahan maks - Nilai absolut maksimum kesalahan atas kumpulan data. Kesalahan maks ada dalam kisaran [0, infty]. 0 adalah skor model yang memprediksi target dengan sempurna.
- Kesalahan absolut median — Kesalahan absolut median ada dalam kisaran [0, infty]. 0 adalah skor model yang memprediksi target dengan sempurna.

Quick model metrics for classification

Berikut ini adalah definisi untuk metrik model cepat:

- Akurasi — Akurasi adalah rasio sampel yang diprediksi secara akurat. Akurasi ada dalam kisaran [0, 1]. 0 adalah skor model yang memprediksi semua sampel secara tidak benar dan 1 adalah skor model sempurna.
- Akurasi seimbang — Akurasi seimbang adalah rasio sampel yang diprediksi secara akurat ketika bobot kelas disesuaikan untuk menyeimbangkan data. Semua kelas diberi kepentingan

yang sama, terlepas dari frekuensinya. Akurasi seimbang ada dalam kisaran [0, 1]. 0 adalah skor model yang memprediksi semua sampel salah. 1 adalah skor model yang sempurna.

- AUC (klasifikasi biner) — Ini adalah area di bawah kurva karakteristik operasi penerima. AUC berada dalam kisaran [0, 1] di mana model acak mengembalikan skor 0,5 dan model sempurna mengembalikan skor 1.
- AUC (OVR) — Untuk klasifikasi multiclass, ini adalah area di bawah kurva karakteristik operasi penerima yang dihitung secara terpisah untuk setiap label menggunakan satu versus istirahat. Data Wrangler melaporkan rata-rata area. AUC berada dalam kisaran [0, 1] di mana model acak mengembalikan skor 0,5 dan model sempurna mengembalikan skor 1.
- Presisi — Presisi didefinisikan untuk kelas tertentu. Presisi adalah fraksi positif sejati dari semua contoh yang diklasifikasikan model sebagai kelas itu. Presisi ada dalam kisaran [0, 1]. 1 adalah skor model yang tidak memiliki positif palsu untuk kelas. Untuk klasifikasi biner, Data Wrangler melaporkan ketepatan kelas positif.
- Ingat — Recall didefinisikan untuk kelas tertentu. Recall adalah fraksi dari instance kelas yang relevan yang berhasil diambil. Ingat ada dalam kisaran [0, 1]. 1 adalah skor model yang mengklasifikasikan semua contoh kelas dengan benar. Untuk klasifikasi biner, Data Wrangler melaporkan penarikan kembali kelas positif.
- F1 — F1 didefinisikan untuk kelas tertentu. Ini adalah rata-rata harmonik dari presisi dan ingatan. F1 berada dalam kisaran [0, 1]. 1 adalah skor model yang sempurna. Untuk klasifikasi biner, Data Wrangler melaporkan F1 untuk kelas dengan nilai positif.

Textual patterns

Pola menggambarkan format tekstual string menggunakan format yang mudah dibaca. Berikut ini adalah contoh pola tekstual:

- “{digits:4-7}” menggambarkan urutan digit yang memiliki panjang antara 4 dan 7.
- “{alnum:5}” menggambarkan string alfa-numerik dengan panjang tepat 5.

Data Wrangler menyimpulkan pola dengan melihat sampel string yang tidak kosong dari data Anda. Ini dapat menggambarkan banyak pola yang umum digunakan. Keyakinan yang dinyatakan sebagai persentase menunjukkan berapa banyak data yang diperkirakan cocok dengan pola. Dengan menggunakan pola tekstual, Anda dapat melihat baris mana dalam data Anda yang perlu Anda koreksi atau jatuhkan.

Berikut ini menjelaskan pola yang dapat dikenali oleh Data Wrangler:

Pola	Format Tekstual
{alnum}	String alfanumerik
{apapun}	Setiap string karakter kata
{digit}	Urutan digit
{lebih rendah}	Sebuah kata huruf kecil
{campuran}	Kata kasus campuran
{nama}	Sebuah kata yang dimulai dengan huruf kapital
{atas}	Sebuah kata huruf besar
{spasi}	Karakter spasi

Karakter kata adalah garis bawah atau karakter yang mungkin muncul dalam kata dalam bahasa apa pun. Misalnya, string `'Hello_word'` dan `'écoute'` keduanya terdiri dari karakter kata. 'H' dan 'é' keduanya merupakan contoh karakter kata.

Histogram

Gunakan histogram untuk melihat jumlah nilai fitur untuk fitur tertentu. Anda dapat memeriksa hubungan antar fitur menggunakan opsi Color by.

Anda dapat menggunakan fitur Facet by untuk membuat histogram dari satu kolom, untuk setiap nilai di kolom lain.

Plot pencar

Gunakan fitur Scatter Plot untuk memeriksa hubungan antar fitur. Untuk membuat plot pencar, pilih fitur untuk diplot pada sumbu X dan sumbu Y. Kedua kolom ini harus berupa kolom yang diketik numerik.

Anda dapat mewarnai plot pencar dengan kolom tambahan.

Selain itu, Anda dapat membagi plot pencar berdasarkan fitur.

Ringkasan tabel

Gunakan analisis Ringkasan Tabel untuk meringkas data Anda dengan cepat.

Untuk kolom dengan data numerik, termasuk data log dan float, ringkasan tabel melaporkan jumlah entri (hitungan), minimum (min), maksimum (maks), rata-rata, dan standar deviasi (stddev) untuk setiap kolom.

Untuk kolom dengan data non-numerik, termasuk kolom dengan string, Boolean, atau data tanggal/waktu, ringkasan tabel melaporkan jumlah entri (hitungan), nilai paling sering (min), dan nilai paling sering (maks).

Model cepat

Gunakan visualisasi Model Cepat untuk mengevaluasi data Anda dengan cepat dan menghasilkan skor penting untuk setiap fitur. Skor nilai kepentingan fitur menunjukkan seberapa berguna fitur dalam memprediksi label target. Skor kepentingan fitur adalah antara [0, 1] dan angka yang lebih tinggi menunjukkan bahwa fitur tersebut lebih penting untuk seluruh kumpulan data. Di bagian atas bagan model cepat, ada skor model. Masalah klasifikasi menunjukkan skor F1. Masalah regresi memiliki skor mean squared error (MSE).

Saat Anda membuat bagan model cepat, Anda memilih kumpulan data yang ingin dievaluasi, dan label target yang ingin Anda bandingkan dengan kepentingan fitur. Data Wrangler melakukan hal berikut:

- Menyimpulkan tipe data untuk label target dan setiap fitur dalam kumpulan data yang dipilih.
- Menentukan jenis masalah. Berdasarkan jumlah nilai yang berbeda di kolom label, Data Wrangler menentukan apakah ini adalah jenis masalah regresi atau klasifikasi. Data Wrangler menetapkan ambang kategoris ke 100. Jika ada lebih dari 100 nilai yang berbeda di kolom label, Data Wrangler mengklasifikasikannya sebagai masalah regresi; jika tidak, itu diklasifikasikan sebagai masalah klasifikasi.
- Fitur pra-proses dan data label untuk pelatihan. Algoritma yang digunakan membutuhkan fitur pengkodean untuk jenis vektor dan label pengkodean untuk tipe ganda.
- Melatih algoritma hutan acak dengan 70% data. Spark RandomForestRegressordigunakan untuk melatih model untuk masalah regresi. RandomForestClassifierIni digunakan untuk melatih model untuk masalah klasifikasi.
- Mengevaluasi model hutan acak dengan sisa 30% data. Data Wrangler mengevaluasi model klasifikasi menggunakan skor F1 dan mengevaluasi model regresi menggunakan skor MSE.

- Menghitung pentingnya fitur untuk setiap fitur menggunakan metode kepentingan Gini.

Kebocoran target

Kebocoran target terjadi ketika ada data dalam kumpulan data pelatihan pembelajaran mesin yang sangat berkorelasi dengan label target, tetapi tidak tersedia dalam data dunia nyata. Misalnya, Anda mungkin memiliki kolom dalam kumpulan data yang berfungsi sebagai proxy untuk kolom yang ingin Anda prediksi dengan model Anda.

Saat Anda menggunakan analisis Kebocoran Target, Anda menentukan yang berikut ini:

- Target: Ini adalah fitur yang Anda inginkan agar model ML Anda dapat membuat prediksi.
- Jenis masalah: Ini adalah jenis masalah ML tempat Anda bekerja. Jenis masalah dapat berupa klasifikasi atau regresi.
- (Opsional) Fitur maks: Ini adalah jumlah maksimum fitur untuk hadir dalam visualisasi, yang menunjukkan fitur yang diberi peringkat berdasarkan risiko kebocoran target.

Untuk klasifikasi, analisis kebocoran target menggunakan area di bawah karakteristik operasi penerima, atau kurva AUC - ROC untuk setiap kolom, hingga fitur Max. Untuk regresi, ia menggunakan koefisien determinasi, atau metrik R2.

Kurva AUC - ROC menyediakan metrik prediktif, dihitung secara individual untuk setiap kolom menggunakan validasi silang, pada sampel hingga sekitar 1000 baris. Skor 1 menunjukkan kemampuan prediksi sempurna, yang sering menunjukkan kebocoran target. Skor 0,5 atau lebih rendah menunjukkan bahwa informasi pada kolom tidak dapat memberikan, dengan sendirinya, informasi yang berguna untuk memprediksi target. Meskipun dapat terjadi bahwa kolom tidak informatif dengan sendirinya tetapi berguna dalam memprediksi target ketika digunakan bersama-sama dengan fitur lain, skor rendah dapat menunjukkan fitur tersebut berlebihan.

Multikolinieritas

Multikolinearitas adalah keadaan di mana dua atau lebih variabel prediktor terkait satu sama lain. Variabel prediktor adalah fitur dalam kumpulan data Anda yang Anda gunakan untuk memprediksi variabel target. Ketika Anda memiliki multikolinieritas, variabel prediktor tidak hanya memprediksi variabel target, tetapi juga prediktif satu sama lain.

Anda dapat menggunakan Variance Inflation Factor (VIF), Principal Component Analysis (PCA), atau pemilihan fitur Lasso sebagai ukuran multikolinearitas dalam data Anda. Untuk informasi selengkapnya, lihat hal berikut.

Variance Inflation Factor (VIF)

Faktor Inflasi Varians (VIF) adalah ukuran kolinearitas di antara pasangan variabel. Data Wrangler mengembalikan skor VIF sebagai ukuran seberapa dekat variabel terkait satu sama lain. Skor VIF adalah angka positif yang lebih besar dari atau sama dengan 1.

Skor 1 berarti bahwa variabel tidak berkorelasi dengan variabel lainnya. Skor lebih besar dari 1 menunjukkan korelasi yang lebih tinggi.

Secara teoritis, Anda dapat memiliki skor VIF dengan nilai tak terhingga. Data Wrangler klip skor tinggi menjadi 50. Jika Anda memiliki skor VIF lebih besar dari 50, Data Wrangler menetapkan skor menjadi 50.

Anda dapat menggunakan panduan berikut untuk menafsirkan skor VIF Anda:

- Skor VIF kurang dari atau sama dengan 5 menunjukkan bahwa variabel cukup berkorelasi dengan variabel lainnya.
- Skor VIF lebih besar dari atau sama dengan 5 menunjukkan bahwa variabel sangat berkorelasi dengan variabel lainnya.

Principle Component Analysis (PCA)

Principal Component Analysis (PCA) mengukur varians data di sepanjang arah yang berbeda di ruang fitur. Ruang fitur terdiri dari semua variabel prediktor yang Anda gunakan untuk memprediksi variabel target dalam kumpulan data Anda.

Misalnya, jika Anda mencoba memprediksi siapa yang selamat di RMS Titanic setelah menabrak gunung es, ruang fitur Anda dapat mencakup usia penumpang, jenis kelamin, dan tarif yang mereka bayar.

Dari ruang fitur, PCA menghasilkan daftar varians yang diurutkan. Varians ini juga dikenal sebagai nilai tunggal. Nilai dalam daftar varians lebih besar dari atau sama dengan 0. Kita dapat menggunakannya untuk menentukan berapa banyak multikolinearitas yang ada dalam data kita.

Ketika angka-angkanya kira-kira seragam, data memiliki sangat sedikit contoh multikolinieritas. Ketika ada banyak variabilitas di antara nilai-nilai, kami memiliki banyak contoh multikolinieritas. Sebelum melakukan PCA, Data Wrangler menormalkan setiap fitur untuk memiliki rata-rata 0 dan standar deviasi 1.

> **Note**
>
> PCA dalam keadaan ini juga dapat disebut sebagai Singular Value Decomposition (SVD).

Lasso feature selection

Pemilihan fitur laso menggunakan teknik regularisasi L1 untuk hanya menyertakan fitur yang paling prediktif dalam kumpulan data Anda.

Untuk klasifikasi dan regresi, teknik regularisasi menghasilkan koefisien untuk setiap fitur. Nilai absolut koefisien memberikan skor penting untuk fitur tersebut. Skor kepentingan yang lebih tinggi menunjukkan bahwa itu lebih prediktif dari variabel target. Metode pemilihan fitur yang umum adalah dengan menggunakan semua fitur yang memiliki koefisien laso bukan nol.

Mendeteksi anomali dalam data deret waktu

Anda dapat menggunakan visualisasi deteksi anomali untuk melihat outlier dalam data deret waktu Anda. Untuk memahami apa yang menentukan anomali, Anda perlu memahami bahwa kami menguraikan deret waktu menjadi istilah yang diprediksi dan istilah kesalahan. Kami memperlakukan musiman dan tren deret waktu sebagai istilah yang diprediksi. Kami memperlakukan residu sebagai istilah kesalahan.

Untuk istilah kesalahan, Anda menentukan ambang batas sebagai jumlah standar deviasi, residu dapat jauh dari rata-rata agar dianggap sebagai anomali. Misalnya, Anda dapat menentukan ambang batas sebagai 3 standar deviasi. Setiap residu yang lebih besar dari 3 standar deviasi dari mean adalah anomali.

Anda dapat menggunakan prosedur berikut untuk melakukan analisis deteksi anomali.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan analisis.
3. Untuk jenis Analisis, pilih Time Series.
4. Untuk Visualisasi, pilih Deteksi anomali.
5. Untuk ambang anomali, pilih ambang batas bahwa nilai dianggap anomali.
6. Pilih Pratinjau untuk menghasilkan pratinjau analisis.
7. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Dekomposisi tren musiman dalam data deret waktu

Anda dapat menentukan apakah ada musiman dalam data deret waktu Anda dengan menggunakan visualisasi Seasonal Trend Decomposition. Kami menggunakan metode STL (Seasonal Trend decomposition using LOESS) untuk melakukan dekomposisi. Kami menguraikan deret waktu menjadi komponen musiman, tren, dan sisa. Tren ini mencerminkan perkembangan jangka panjang dari seri ini. Komponen musiman adalah sinyal yang berulang dalam periode waktu tertentu. Setelah menghapus tren dan komponen musiman dari deret waktu, Anda memiliki residu.

Anda dapat menggunakan prosedur berikut untuk melakukan analisis dekomposisi Seasonal-Trend.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan analisis.
3. Untuk jenis Analisis, pilih Time Series.
4. Untuk Visualisasi, pilih dekomposisi Seasonal-Trend.
5. Untuk ambang anomali, pilih ambang batas bahwa nilai dianggap anomali.
6. Pilih Pratinjau untuk menghasilkan pratinjau analisis.
7. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Buat visualisasi khusus

Anda dapat menambahkan analisis ke alur Data Wrangler Anda untuk membuat visualisasi kustom. Dataset Anda, dengan semua transformasi yang Anda terapkan, tersedia sebagai Panda. DataFrame Data Wrangler menggunakan `df` variabel untuk menyimpan kerangka data. Anda mengakses kerangka data dengan memanggil variabel.

Anda harus memberikan variabel output,`chart`, untuk menyimpan bagan output Altair. Misalnya, Anda dapat menggunakan blok kode berikut untuk membuat histogram khusus menggunakan dataset Titanic.

```
import altair as alt
df = df.iloc[:30]
df = df.rename(columns={"Age": "value"})
df = df.assign(count=df.groupby('value').value.transform('count'))
df = df[["value", "count"]]
base = alt.Chart(df)
bar = base.mark_bar().encode(x=alt.X('value', bin=True, axis=None), y=alt.Y('count'))
```

```
rule = base.mark_rule(color='red').encode(
    x='mean(value):Q',
    size=alt.value(5))
chart = bar + rule
```

Untuk membuat visualisasi kustom:

1. Di samping node yang berisi transformasi yang ingin Anda visualisasikan, pilih +.
2. Pilih Tambahkan analisis.
3. Untuk jenis Analisis, pilih Visualisasi Kustom.
4. Untuk nama Analisis, tentukan nama.
5. Masukkan kode Anda di kotak kode.
6. Pilih Pratinjau untuk melihat visualisasi Anda.
7. Pilih Simpan untuk menambahkan visualisasi Anda.

Jika Anda tidak tahu cara menggunakan paket visualisasi Altair dengan Python, Anda dapat menggunakan cuplikan kode khusus untuk membantu Anda memulai.

Data Wrangler memiliki koleksi cuplikan visualisasi yang dapat dicari. Untuk menggunakan cuplikan visualisasi, pilih Cari contoh cuplikan dan tentukan kueri di bilah pencarian.

Contoh berikut menggunakan cuplikan kode scatterplot Binned. Ini memplot histogram untuk 2 dimensi.

Cuplikan memiliki komentar untuk membantu Anda memahami perubahan yang perlu Anda buat pada kode. Anda biasanya perlu menentukan nama kolom dataset Anda dalam kode.

```
import altair as alt

# Specify the number of top rows for plotting
rows_number = 1000
df = df.head(rows_number)
# You can also choose bottom rows or randomly sampled rows
# df = df.tail(rows_number)
# df = df.sample(rows_number)
```

```
chart = (
    alt.Chart(df)
    .mark_circle()
    .encode(
        # Specify the column names for binning and number of bins for X and Y axis
        x=alt.X("col1:Q", bin=alt.Bin(maxbins=20)),
        y=alt.Y("col2:Q", bin=alt.Bin(maxbins=20)),
        size="count()",
    )
)

# :Q specifies that label column has quantitative type.
# For more details on Altair typing refer to
# https://altair-viz.github.io/user_guide/encoding.html#encoding-data-types
```

Mengubah data

Amazon SageMaker Data Wrangler menyediakan banyak transformasi data ML untuk merampingkan pembersihan, transformasi, dan fitur data Anda. Ketika Anda menambahkan transformasi, itu menambahkan langkah ke aliran data. Setiap transformasi yang Anda tambahkan memodifikasi dataset Anda dan menghasilkan kerangka data baru. Semua transformasi selanjutnya berlaku untuk kerangka data yang dihasilkan.

Data Wrangler mencakup transformasi bawaan, yang dapat Anda gunakan untuk mengubah kolom tanpa kode apa pun. Anda juga dapat menambahkan transformasi kustom menggunakan PySpark, Python (User-Defined Function), panda, dan SQL. PySpark Beberapa transformasi beroperasi di tempat, sementara yang lain membuat kolom output baru di dataset Anda.

Anda dapat menerapkan transformasi ke beberapa kolom sekaligus. Misalnya, Anda dapat menghapus beberapa kolom dalam satu langkah.

Anda dapat menerapkan Process numeric dan Handle hilang transformasi hanya untuk satu kolom.

Gunakan halaman ini untuk mempelajari lebih lanjut tentang transformasi bawaan dan kustom ini.

Ubah UI

Sebagian besar transformasi bawaan terletak di tab Siapkan UI Data Wrangler. Anda dapat mengakses transformasi join dan concatenate melalui tampilan aliran data. Gunakan tabel berikut untuk melihat pratinjau dua tampilan ini.

Transform

Anda dapat menambahkan transformasi ke langkah apa pun dalam aliran data Anda. Gunakan prosedur berikut untuk menambahkan transformasi ke aliran data Anda.

Untuk menambahkan langkah ke aliran data Anda, lakukan hal berikut.

1. Pilih + di sebelah langkah dalam aliran data.
2. Pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih transformasi.
5. (Opsional) Anda dapat mencari transformasi yang ingin Anda gunakan. Data Wrangler menyoroti kueri dalam hasil.

Join View

Untuk menggabungkan dua kumpulan data, pilih kumpulan data pertama dalam aliran data Anda dan pilih Gabung. Saat Anda memilih Bergabung. Kumpulan data kiri dan kanan Anda ditampilkan di panel kiri. Panel utama menampilkan aliran data Anda, dengan dataset yang baru bergabung ditambahkan.

Ketika Anda memilih Konfigurasi untuk mengonfigurasi gabungan Anda, Anda melihat hasil yang mirip dengan yang ditunjukkan pada gambar berikut. Konfigurasi bergabung Anda ditampilkan di panel kiri. Anda dapat menggunakan panel ini untuk memilih nama kumpulan data yang bergabung, jenis gabungan, dan kolom untuk bergabung. Panel utama menampilkan tiga tabel. Dua tabel teratas menampilkan kumpulan data kiri dan kanan masing-masing di kiri dan kanan. Di bawah tabel ini, Anda dapat melihat pratinjau kumpulan data yang digabungkan.

Lihat Bergabunglah dengan Datasets untuk mempelajari selengkapnya.

Concatenate View

Untuk menggabungkan dua kumpulan data, Anda memilih kumpulan data pertama dalam aliran data Anda dan memilih Concatenate. Kumpulan data kiri dan kanan Anda ditampilkan di panel kiri. Panel utama menampilkan aliran data Anda, dengan dataset yang baru digabungkan ditambahkan.

Ketika Anda memilih Konfigurasi untuk mengonfigurasi rangkaian Anda, Anda melihat hasil yang mirip dengan yang ditunjukkan pada gambar berikut. Konfigurasi gabungan Anda ditampilkan di panel kiri. Anda dapat menggunakan panel ini untuk memilih nama kumpulan data gabungan,

dan memilih untuk menghapus duplikat setelah penggabungan dan menambahkan kolom untuk menunjukkan kerangka data sumber. Panel utama menampilkan tiga tabel. Dua tabel teratas menampilkan kumpulan data kiri dan kanan masing-masing di kiri dan kanan. Di bawah tabel ini, Anda dapat melihat pratinjau kumpulan data gabungan.

Lihat Menggabungkan Dataset untuk mempelajari selengkapnya.

Bergabunglah dengan Datasets

Anda bergabung dengan kerangka data secara langsung dalam aliran data Anda. Saat Anda menggabungkan dua kumpulan data, kumpulan data gabungan yang dihasilkan akan muncul di alur Anda. Jenis gabungan berikut didukung oleh Data Wrangler.

- Left Outer - Sertakan semua baris dari tabel kiri. Jika nilai untuk kolom bergabung pada baris tabel kiri tidak cocok dengan nilai baris tabel kanan, baris tersebut berisi nilai nol untuk semua kolom tabel kanan dalam tabel gabungan.
- Anti Kiri - Sertakan baris dari tabel kiri yang tidak mengandung nilai di tabel kanan untuk kolom yang digabungkan.
- Semi kiri - Sertakan satu baris dari tabel kiri untuk semua baris identik yang memenuhi kriteria dalam pernyataan gabungan. Ini tidak termasuk baris duplikat dari tabel kiri yang cocok dengan kriteria gabungan.
- Luar Kanan - Sertakan semua baris dari tabel kanan. Jika nilai untuk kolom bergabung di baris tabel kanan tidak cocok dengan nilai baris tabel kiri, baris tersebut berisi nilai nol untuk semua kolom tabel kiri dalam tabel gabungan.
- Inner - Sertakan baris dari tabel kiri dan kanan yang berisi nilai yang cocok di kolom yang digabungkan.
- Full Outer - Sertakan semua baris dari tabel kiri dan kanan. Jika nilai baris untuk kolom gabungan di salah satu tabel tidak cocok, baris terpisah dibuat dalam tabel gabungan. Jika baris tidak berisi nilai untuk kolom dalam tabel gabungan, null disisipkan untuk kolom itu.
- Cartesian Cross - Sertakan baris yang menggabungkan setiap baris dari tabel pertama dengan setiap baris dari tabel kedua. Ini adalah produk Cartesian dari baris dari tabel di join. Hasil dari produk ini adalah ukuran tabel kiri dikalikan ukuran meja kanan. Oleh karena itu, kami menyarankan agar berhati-hati dalam menggunakan gabungan ini antara kumpulan data yang sangat besar.

Gunakan prosedur berikut untuk menggabungkan dua kerangka data.

1. Pilih + di sebelah kerangka data kiri yang ingin Anda ikuti. Rangka data pertama yang Anda pilih selalu tabel kiri di gabungan Anda.
2. Pilih Bergabung.
3. Pilih kerangka data yang tepat. Rangka data kedua yang Anda pilih selalu merupakan tabel yang tepat dalam bergabung Anda.
4. Pilih Konfigurasi untuk mengonfigurasi gabungan Anda.
5. Beri nama kumpulan data gabungan Anda menggunakan bidang Nama.
6. Pilih jenis Gabung.
7. Pilih kolom dari tabel kiri dan kanan untuk bergabung.
8. Pilih Terapkan untuk melihat pratinjau kumpulan data yang bergabung di sebelah kanan.
9. Untuk menambahkan tabel gabungan ke alur data Anda, pilih Tambah.

Menggabungkan Dataset

Gabungkan dua kumpulan data:

1. Pilih + di sebelah kerangka data kiri yang ingin Anda gabungkan. Rangka data pertama yang Anda pilih selalu tabel kiri dalam rangkaian Anda.
2. Pilih Concatenate.
3. Pilih kerangka data yang tepat. Rangka data kedua yang Anda pilih selalu merupakan tabel yang tepat dalam rangkaian Anda.
4. Pilih Konfigurasi untuk mengonfigurasi rangkaian Anda.
5. Beri nama kumpulan data gabungan Anda menggunakan bidang Nama.
6. (Opsional) Pilih kotak centang di samping Hapus duplikat setelah penggabungan untuk menghapus kolom duplikat.
7. (Opsional) Pilih kotak centang di sebelah Tambahkan kolom untuk menunjukkan kerangka data sumber jika, untuk setiap kolom dalam kumpulan data baru, Anda ingin menambahkan indikator sumber kolom.
8. Pilih Terapkan untuk melihat pratinjau kumpulan data baru.
9. Pilih Tambah untuk menambahkan kumpulan data baru ke alur data Anda.

Data Saldo

Anda dapat menyeimbangkan data untuk kumpulan data dengan kategori yang kurang terwakili. Menyeimbangkan kumpulan data dapat membantu Anda membuat model yang lebih baik untuk klasifikasi biner.

> **Note**
>
> Anda tidak dapat menyeimbangkan kumpulan data yang berisi vektor kolom.

Anda dapat menggunakan operasi data Saldo untuk menyeimbangkan data Anda menggunakan salah satu operator berikut:

- Oversampling acak - Duplikat sampel secara acak dalam kategori minoritas. Misalnya, jika Anda mencoba mendeteksi penipuan, Anda mungkin hanya memiliki kasus penipuan di 10% data Anda. Untuk proporsi yang sama dari kasus penipuan dan non-penipuan, operator ini secara acak menduplikasi kasus penipuan dalam kumpulan data 8 kali.
- Undersampling acak — Kira-kira setara dengan oversampling acak. Secara acak menghapus sampel dari kategori yang terwakili secara berlebihan untuk mendapatkan proporsi sampel yang Anda inginkan.
- Synthetic Minority Oversampling Technique (SMOTE) — Menggunakan sampel dari kategori yang kurang terwakili untuk menginterpolasi sampel minoritas sintetis baru. Untuk informasi lebih lanjut tentang SMOTE, lihat deskripsi berikut.

Anda dapat menggunakan semua transformasi untuk kumpulan data yang berisi fitur numerik dan non-numerik. SMOTE menginterpolasi nilai dengan menggunakan sampel tetangga. Data Wrangler menggunakan jarak R-kuadrat untuk menentukan lingkungan untuk menginterpolasi sampel tambahan. Data Wrangler hanya menggunakan fitur numerik untuk menghitung jarak antara sampel dalam kelompok yang kurang terwakili.

Untuk dua sampel nyata dalam kelompok yang kurang terwakili, Data Wrangler menginterpolasi fitur numerik dengan menggunakan rata-rata tertimbang. Ini secara acak memberikan bobot untuk sampel tersebut dalam kisaran [0, 1]. Untuk fitur numerik, Data Wrangler menginterpolasi sampel menggunakan rata-rata tertimbang sampel. Untuk sampel A dan B, Data Wrangler dapat secara acak menetapkan berat 0,7 hingga A dan 0,3 hingga B. Sampel yang diinterpolasi memiliki nilai 0,7A +0,3B.

Data Wrangler menginterpolasi fitur non-numerik dengan menyalin dari salah satu sampel nyata yang diinterpolasi. Ini menyalin sampel dengan probabilitas bahwa itu secara acak menetapkan untuk setiap sampel. Untuk sampel A dan B, ia dapat menetapkan probabilitas 0,8 ke A dan 0,2 ke B. Untuk probabilitas yang ditetapkan, ia menyalin A 80% dari waktu.

Transformasi Kustom

Grup Custom Transforms memungkinkan Anda untuk menggunakan Python (User-Defined Function) PySpark,, pandas, PySpark atau (SQL) untuk menentukan transformasi kustom. Untuk ketiga opsi, Anda menggunakan variabel `df` untuk mengakses kerangka data yang ingin Anda terapkan transformasi. Untuk menerapkan kode kustom Anda ke kerangka data Anda, tetapkan kerangka data dengan transformasi yang telah Anda buat ke variabel. `df` Jika Anda tidak menggunakan Python (User-Defined Function), Anda tidak perlu menyertakan pernyataan pengembalian. Pilih Pratinjau untuk melihat hasil transformasi kustom. Pilih Tambah untuk menambahkan transformasi kustom ke daftar langkah Sebelumnya.

Anda dapat mengimpor pustaka populer dengan `import` pernyataan di blok kode transformasi kustom, seperti berikut ini:

- NumPy versi 1.19.0
- scikit-learn versi 0.23.2
- SciPy versi 1.5.4
- panda versi 1.0.3
- PySpark versi 3.0.0

> **Important**
>
> Transformasi kustom tidak mendukung kolom dengan spasi atau karakter khusus dalam nama. Kami menyarankan Anda menentukan nama kolom yang hanya memiliki karakter alfanumerik dan garis bawah. Anda dapat menggunakan Transformasi kolom Rename di grup Mengelola kolom transformasi untuk menghapus spasi dari nama kolom. Anda juga dapat menambahkan Python (Pandas) Custom transform mirip dengan berikut ini untuk menghapus spasi dari beberapa kolom dalam satu langkah. Contoh ini mengubah kolom bernama `A column` dan `B column` ke `A_column` dan `B_column` masing-masing.
>
> ```
> df.rename(columns={"A column": "A_column", "B column": "B_column"})
> ```

Jika Anda menyertakan pernyataan cetak di blok kode, hasilnya akan muncul saat Anda memilih Pratinjau. Anda dapat mengubah ukuran panel transformator kode khusus. Mengubah ukuran panel menyediakan lebih banyak ruang untuk menulis kode.

Bagian berikut memberikan konteks tambahan dan contoh untuk menulis kode transformasi kustom.

Python (Fungsi yang Ditentukan Pengguna)

Fungsi Python memberi Anda kemampuan untuk menulis transformasi khusus tanpa perlu mengetahui Apache Spark atau panda. Data Wrangler dioptimalkan untuk menjalankan kode kustom Anda dengan cepat. Anda mendapatkan kinerja serupa menggunakan kode Python khusus dan plugin Apache Spark.

Untuk menggunakan blok kode Python (User-Defined Function), Anda tentukan yang berikut ini:

- Kolom input - Kolom masukan tempat Anda menerapkan transformasi.
- Mode — Mode scripting, baik panda atau Python.
- Jenis pengembalian - Tipe data dari nilai yang Anda kembalikan.

Menggunakan mode panda memberikan kinerja yang lebih baik. Mode Python memudahkan Anda untuk menulis transformasi dengan menggunakan fungsi Python murni.

PySpark

Contoh berikut mengekstrak tanggal dan waktu dari stempel waktu.

```
from pyspark.sql.functions import from_unixtime, to_date, date_format
df = df.withColumn('DATE_TIME', from_unixtime('TIMESTAMP'))
df = df.withColumn( 'EVENT_DATE', to_date('DATE_TIME')).withColumn(
'EVENT_TIME', date_format('DATE_TIME', 'HH:mm:ss'))
```

panda

Contoh berikut memberikan ikhtisar kerangka data yang Anda tambahkan transformasi.

```
df.info()
```

PySpark (SQL)

Contoh berikut membuat kerangka data baru dengan empat kolom: name, fare, pclass, survived.

```
SELECT name, fare, pclass, survived FROM df
```

Jika Anda tidak tahu cara menggunakannya PySpark, Anda dapat menggunakan cuplikan kode khusus untuk membantu Anda memulai.

Data Wrangler memiliki kumpulan cuplikan kode yang dapat dicari. Anda dapat menggunakan potongan kode untuk melakukan tugas seperti menjatuhkan kolom, mengelompokkan berdasarkan kolom, atau pemodelan.

Untuk menggunakan cuplikan kode, pilih Cari contoh cuplikan dan tentukan kueri di bilah pencarian. Teks yang Anda tentukan dalam kueri tidak harus sama persis dengan nama cuplikan kode.

Contoh berikut menunjukkan cuplikan kode baris duplikat Jatuhkan yang dapat menghapus baris dengan data serupa di kumpulan data Anda. Anda dapat menemukan cuplikan kode dengan mencari salah satu dari berikut ini:

- Duplikat
- Identik
- Menghapus

Cuplikan berikut memiliki komentar untuk membantu Anda memahami perubahan yang perlu Anda buat. Untuk sebagian besar cuplikan, Anda harus menentukan nama kolom kumpulan data Anda dalam kode.

```
# Specify the subset of columns
# all rows having identical values in these columns will be dropped

subset = ["col1", "col2", "col3"]
df = df.dropDuplicates(subset)

# to drop the full-duplicate rows run
# df = df.dropDuplicates()
```

Untuk menggunakan cuplikan, salin dan tempel kontennya ke bidang Custom transform. Anda dapat menyalin dan menempelkan beberapa cuplikan kode ke bidang transformasi khusus.

Formula Kustom

Gunakan rumus Kustom untuk menentukan kolom baru menggunakan ekspresi Spark SQL untuk menanyakan data dalam kerangka data saat ini. Kueri harus menggunakan konvensi ekspresi Spark SQL.

> **Important**
>
> Rumus kustom tidak mendukung kolom dengan spasi atau karakter khusus dalam nama. Kami menyarankan Anda menentukan nama kolom yang hanya memiliki karakter alfanumerik dan garis bawah. Anda dapat menggunakan Transformasi kolom Rename di grup Mengelola kolom transformasi untuk menghapus spasi dari nama kolom. Anda juga dapat menambahkan Python (Pandas) Custom transform mirip dengan berikut ini untuk menghapus spasi dari beberapa kolom dalam satu langkah. Contoh ini mengubah kolom bernama `A column` dan `B column` ke `A_column` dan `B_column` masing-masing.
>
> ```
> df.rename(columns={"A column": "A_column", "B column": "B_column"})
> ```

Anda dapat menggunakan transformasi ini untuk melakukan operasi pada kolom, mereferensikan kolom dengan nama. Misalnya, dengan asumsi kerangka data saat ini berisi kolom bernama col_a dan col_b, Anda dapat menggunakan operasi berikut untuk menghasilkan kolom Output yang merupakan produk dari dua kolom ini dengan kode berikut:

```
col_a * col_b
```

Operasi umum lainnya termasuk yang berikut, dengan asumsi kerangka data berisi dan kolom: `col_a` `col_b`

- Gabungkan dua kolom: `concat(col_a, col_b)`
- Tambahkan dua kolom: `col_a + col_b`
- Kurangi dua kolom: `col_a - col_b`
- Bagilah dua kolom: `col_a / col_b`
- Ambil nilai absolut dari kolom: `abs(col_a)`

Untuk informasi selengkapnya, lihat dokumentasi Spark tentang memilih data.

Mengurangi Dimensionalitas dalam Dataset

Kurangi dimensi dalam data Anda dengan menggunakan Principal Component Analysis (PCA). Dimensi kumpulan data Anda sesuai dengan jumlah fitur. Saat Anda menggunakan pengurangan dimensi di Data Wrangler, Anda mendapatkan serangkaian fitur baru yang disebut komponen. Setiap komponen memperhitungkan beberapa variabilitas dalam data.

Komponen pertama menyumbang jumlah variasi terbesar dalam data. Komponen kedua menyumbang jumlah variasi terbesar kedua dalam data, dan seterusnya.

Anda dapat menggunakan pengurangan dimensi untuk mengurangi ukuran kumpulan data yang Anda gunakan untuk melatih model. Alih-alih menggunakan fitur dalam kumpulan data Anda, Anda dapat menggunakan komponen utama sebagai gantinya.

Untuk melakukan PCA, Data Wrangler membuat sumbu untuk data Anda. Sumbu adalah kombinasi affine kolom dalam kumpulan data Anda. Komponen utama pertama adalah nilai pada sumbu yang memiliki jumlah varians terbesar. Komponen utama kedua adalah nilai pada sumbu yang memiliki jumlah varians terbesar kedua. Komponen utama ke-n adalah nilai pada sumbu yang memiliki jumlah varians terbesar ke-n.

Anda dapat mengonfigurasi jumlah komponen utama yang dikembalikan Data Wrangler. Anda dapat menentukan jumlah komponen utama secara langsung atau Anda dapat menentukan persentase ambang varians. Setiap komponen utama menjelaskan sejumlah varians dalam data. Misalnya, Anda mungkin memiliki komponen utama dengan nilai 0,5. Komponen akan menjelaskan 50% variasi dalam data. Saat Anda menentukan persentase ambang varians, Data Wrangler mengembalikan jumlah komponen terkecil yang memenuhi persentase yang Anda tentukan.

Berikut ini adalah contoh komponen utama dengan jumlah varians yang mereka jelaskan dalam data.

- Komponen 1 — 0.5
- Komponen 2 - 0.45
- Komponen 3 - 0.05

Jika Anda menentukan persentase ambang batas varians dari 94 or95, Data Wrangler mengembalikan Komponen 1 dan Komponen 2. Jika Anda menentukan persentase ambang varians dari96, Data Wrangler mengembalikan ketiga komponen utama.

Anda dapat menggunakan prosedur berikut untuk menjalankan PCA pada dataset Anda.

Untuk menjalankan PCA pada dataset Anda, lakukan hal berikut.

1. Buka aliran data Wrangler Data Anda.
2. Pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Pengurangan Dimensi.
5. Untuk Kolom Input, pilih fitur yang Anda kurangi menjadi komponen utama.
6. (Opsional) Untuk Jumlah komponen utama, pilih jumlah komponen utama yang dikembalikan Data Wrangler dalam kumpulan data Anda. Jika menentukan nilai untuk bidang, Anda tidak dapat menentukan nilai untuk persentase ambang batas Varians.
7. (Opsional) Untuk persentase ambang batas Varians, tentukan persentase variasi dalam data yang ingin Anda jelaskan oleh komponen utama. Data Wrangler menggunakan nilai default 95 jika Anda tidak menentukan nilai untuk ambang varians. Anda tidak dapat menentukan persentase ambang varians jika Anda telah menentukan nilai untuk Jumlah komponen utama.
8. (Opsional) Batalkan pilihan Pusat untuk tidak menggunakan rata-rata kolom sebagai pusat data. Secara default, Data Wrangler memusatkan data dengan mean sebelum penskalaan.
9. (Opsional) Batalkan pilihan Skala untuk tidak menskalakan data dengan standar deviasi unit.
10. (Opsional) Pilih Kolom untuk menampilkan komponen ke kolom terpisah. Pilih Vector untuk menampilkan komponen sebagai vektor tunggal.
11. (Opsional) Untuk kolom Output, tentukan nama untuk kolom keluaran. Jika Anda mengeluarkan komponen ke kolom terpisah, nama yang Anda tentukan adalah awalan. Jika Anda mengeluarkan komponen ke vektor, nama yang Anda tentukan adalah nama kolom vektor.
12. (Opsional) Pilih Simpan kolom input. Kami tidak menyarankan memilih opsi ini jika Anda berencana hanya menggunakan komponen utama untuk melatih model Anda.
13. Pilih Pratinjau.
14. Pilih Tambahkan.

Encode Kategoris

Data kategoris biasanya terdiri dari sejumlah kategori yang terbatas, di mana setiap kategori diwakili dengan string. Misalnya, jika Anda memiliki tabel data pelanggan, kolom yang menunjukkan negara tempat seseorang tinggal adalah kategoris. Kategori-kategorinya adalah Afghanistan, Albania, Aljazair, dan sebagainya. Data kategoris dapat berupa nominal atau ordinal. Kategori ordinal memiliki urutan yang melekat, dan kategori nominal tidak. Gelar tertinggi yang diperoleh (SMA, Sarjana, Magister, dan sebagainya) adalah contoh kategori ordinal.

Pengkodean data kategoris adalah proses menciptakan representasi numerik untuk kategori. Misalnya, jika kategori Anda adalah Dog dan Cat, Anda dapat menyandikan informasi ini menjadi dua vektor, *[1,0]* untuk mewakili Dog, dan *[0,1]* untuk mewakili Cat.

Saat Anda menyandikan kategori ordinal, Anda mungkin perlu menerjemahkan urutan alami kategori ke dalam pengkodean Anda. Misalnya, Anda dapat mewakili derajat tertinggi yang diperoleh dengan peta berikut:`{"High school": 1, "Bachelors": 2, "Masters":3}`.

Gunakan pengkodean kategoris untuk menyandikan data kategoris yang dalam format string ke dalam array bilangan bulat.

Encoder kategoris Data Wrangler membuat pengkodean untuk semua kategori yang ada di kolom pada saat langkah ditentukan. Jika kategori baru telah ditambahkan ke kolom saat Anda memulai pekerjaan Data Wrangler untuk memproses kumpulan data Anda pada waktu t, dan kolom ini adalah masukan untuk transformasi pengkodean kategoris Data Wrangler pada waktu t -1, kategori baru ini dianggap hilang dalam pekerjaan Data Wrangler. Opsi yang Anda pilih untuk Strategi penanganan tidak valid diterapkan pada nilai yang hilang ini. Contoh kapan ini dapat terjadi adalah:

- Saat Anda menggunakan file.flow untuk membuat pekerjaan Data Wrangler untuk memproses kumpulan data yang diperbarui setelah pembuatan aliran data. Misalnya, Anda dapat menggunakan aliran data untuk memproses data penjualan secara teratur setiap bulan. Jika data penjualan diperbarui setiap minggu, kategori baru dapat dimasukkan ke dalam kolom yang menentukan langkah kategoris encode.
- Ketika Anda memilih Sampling ketika Anda mengimpor dataset Anda, beberapa kategori mungkin ditinggalkan dari sampel.

Dalam situasi ini, kategori baru ini dianggap nilai yang hilang dalam pekerjaan Data Wrangler.

Anda dapat memilih dari dan mengkonfigurasi ordinal dan encode satu-panas. Gunakan bagian berikut untuk mempelajari lebih lanjut tentang opsi ini.

Kedua transformasi membuat kolom baru bernama Output nama kolom. Anda menentukan format output kolom ini dengan gaya Output:

- Pilih Vektor untuk menghasilkan satu kolom dengan vektor jarang.
- Pilih Kolom untuk membuat kolom untuk setiap kategori dengan variabel indikator apakah teks di kolom asli berisi nilai yang sama dengan kategori tersebut.

Pengkodean Ordinal

Pilih Ordinal encode untuk menyandikan kategori menjadi bilangan bulat antara 0 dan jumlah total kategori di kolom Input yang Anda pilih.

Strategi penyerahan tidak valid: Pilih metode untuk menangani nilai yang tidak valid atau hilang.

- Pilih Lewati jika Anda ingin menghilangkan baris dengan nilai yang hilang.
- Pilih Simpan untuk mempertahankan nilai yang hilang sebagai kategori terakhir.
- Pilih Kesalahan jika Anda ingin Data Wrangler melempar kesalahan jika nilai yang hilang ditemukan di kolom Input.
- Pilih Ganti dengan NaN untuk mengganti yang hilang dengan NaN. Opsi ini direkomendasikan jika algoritme ML Anda dapat menangani nilai yang hilang. Jika tidak, tiga opsi pertama dalam daftar ini dapat menghasilkan hasil yang lebih baik.

One-Hot Encode

Pilih One-hot encode untuk Transform untuk menggunakan one-hot encoding. Konfigurasikan transformasi ini menggunakan yang berikut:

- Jatuhkan kategori terakhir: Jika`True`, kategori terakhir tidak memiliki indeks yang sesuai dalam pengkodean satu panas. Ketika nilai yang hilang dimungkinkan, kategori yang hilang selalu menjadi yang terakhir dan `True` menyetelnya berarti bahwa nilai yang hilang menghasilkan vektor nol.
- Strategi penyerahan tidak valid: Pilih metode untuk menangani nilai yang tidak valid atau hilang.
  - Pilih Lewati jika Anda ingin menghilangkan baris dengan nilai yang hilang.
  - Pilih Simpan untuk mempertahankan nilai yang hilang sebagai kategori terakhir.
  - Pilih Kesalahan jika Anda ingin Data Wrangler melempar kesalahan jika nilai yang hilang ditemukan di kolom Input.
- Apakah input ordinal dikodekan: Pilih opsi ini jika vektor input berisi data yang dikodekan ordinal. Opsi ini mengharuskan data input mengandung bilangan bulat non-negatif. Jika Benar, masukan i dikodekan sebagai vektor dengan bukan nol di lokasi ke-i.

Kesamaan menyandikan

Gunakan pengkodean kesamaan ketika Anda memiliki yang berikut:

- Sejumlah besar variabel kategoris
- Data berisik

Encoder kesamaan menciptakan embeddings untuk kolom dengan data kategoris. Embedding adalah pemetaan objek diskrit, seperti kata-kata, ke vektor bilangan real. Ini mengkodekan string yang mirip dengan vektor yang mengandung nilai serupa. Misalnya, ia menciptakan pengkodean yang sangat mirip untuk “California” dan “Calfornia”.

Data Wrangler mengonversi setiap kategori dalam kumpulan data Anda menjadi satu set token menggunakan tokenizer 3 gram. Ini mengubah token menjadi embedding menggunakan encoding min-hash.

Pengkodean kesamaan yang dibuat Data Wrangler:

- Memiliki dimensi rendah
- Dapat diskalakan untuk sejumlah besar kategori
- Kuat dan tahan terhadap kebisingan

Untuk alasan sebelumnya, pengkodean kesamaan lebih fleksibel daripada pengkodean satu panas.

Untuk menambahkan transformasi pengkodean kesamaan ke kumpulan data Anda, gunakan prosedur berikut.

Untuk menggunakan pengkodean kesamaan, lakukan hal berikut.

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Open Studio Classic.
3. Pilih Luncurkan aplikasi.
4. Pilih Studio.
5. Tentukan aliran data Anda.
6. Pilih langkah dengan transformasi.
7. Pilih Tambahkan langkah.
8. Pilih Encode kategoris.
9. Tentukan hal berikut:
    - Transform - Encode kesamaan

- Kolom input - Kolom yang berisi data kategoris yang Anda enkodekan.
- Dimensi target — (Opsional) Dimensi vektor embedding kategoris. Nilai default-nya adalah 30. Sebaiknya gunakan dimensi target yang lebih besar jika Anda memiliki kumpulan data besar dengan banyak kategori.
- Gaya keluaran — Pilih Vektor untuk vektor tunggal dengan semua nilai yang dikodekan. Pilih Kolom untuk memiliki nilai yang dikodekan di kolom terpisah.
- Kolom keluaran - (Opsional) Nama kolom keluaran untuk output yang dikodekan vektor. Untuk output yang dikodekan kolom, ini adalah awalan dari nama kolom diikuti dengan nomor yang terdaftar.

Featurize Teks

Gunakan grup transformasi Teks Featurize untuk memeriksa kolom yang diketik string dan gunakan penyematan teks untuk menyesuaikan kolom ini.

Grup fitur ini berisi dua fitur, statistik Karakter dan Vektor. Gunakan bagian berikut untuk mempelajari lebih lanjut tentang transformasi ini. Untuk kedua opsi, kolom Input harus berisi data teks (tipe string).

Statistik Karakter

Gunakan statistik Karakter untuk menghasilkan statistik untuk setiap baris dalam kolom yang berisi data teks.

Transformasi ini menghitung rasio dan hitungan berikut untuk setiap baris, dan membuat kolom baru untuk melaporkan hasilnya. Kolom baru diberi nama menggunakan nama kolom input sebagai awalan dan akhiran yang spesifik untuk rasio atau hitungan.

- Jumlah kata: Jumlah kata dalam baris itu. Sufiks untuk kolom keluaran ini adalah`-stats_word_count`.
- Jumlah karakter: Jumlah total karakter di baris itu. Sufiks untuk kolom keluaran ini adalah`-stats_char_count`.
- Rasio atas: Jumlah karakter huruf besar, dari A hingga Z, dibagi dengan semua karakter di kolom. Sufiks untuk kolom keluaran ini adalah`-stats_capital_ratio`.
- Rasio yang lebih rendah: Jumlah karakter huruf kecil, dari a hingga z, dibagi dengan semua karakter di kolom. Sufiks untuk kolom keluaran ini adalah`-stats_lower_ratio`.
- Rasio digit: Rasio digit dalam satu baris di atas jumlah digit di kolom input. Sufiks untuk kolom keluaran ini adalah`-stats_digit_ratio`.

- Rasio karakter khusus: Rasio karakter non-alfanumerik (seperti # $&%: @) terhadap jumlah semua karakter di kolom input. Sufiks untuk kolom keluaran ini adalah`-stats_special_ratio`.

Vektorisasi

Penyematan teks melibatkan pemetaan kata atau frasa dari kosakata ke vektor bilangan real. Gunakan transformasi penyematan teks Data Wrangler untuk memberi token dan memvektorisasi data teks menjadi vektor frekuensi terminal-inverse document frequency (TF-IDF).

Ketika TF-IDF dihitung untuk kolom data teks, setiap kata dalam setiap kalimat diubah menjadi bilangan real yang mewakili kepentingan semantiknya. Angka yang lebih tinggi dikaitkan dengan kata-kata yang lebih jarang, yang cenderung lebih bermakna.

Saat Anda menentukan langkah transformasi Vectorize, Data Wrangler menggunakan data dalam kumpulan data Anda untuk menentukan metode count vectorizer dan TF-IDF. Menjalankan pekerjaan Data Wrangler menggunakan metode yang sama.

Anda mengonfigurasi transformasi ini menggunakan yang berikut:

- Nama kolom keluaran: Transformasi ini membuat kolom baru dengan penyematan teks. Gunakan bidang ini untuk menentukan nama untuk kolom keluaran ini.
- Tokenizer: Tokenizer mengubah kalimat menjadi daftar kata, atau token.

  Pilih Standar untuk menggunakan tokenizer yang dibagi dengan spasi putih dan mengubah setiap kata menjadi huruf kecil. Misalnya, `"Good dog"` diberi token ke. `["good","dog"]`

  Pilih Custom untuk menggunakan tokenizer yang disesuaikan. Jika Anda memilih Custom, Anda dapat menggunakan bidang berikut untuk mengkonfigurasi tokenizer:

  - Panjang token minimum: Panjang minimum, dalam karakter, agar token valid. Default ke `1`. Misalnya, jika Anda menentukan `3` panjang token minimum, kata-kata seperti `a, at, in` dijatuhkan dari kalimat tokenized.
  - Haruskah regex terbelah pada celah: Jika dipilih, regex terbagi pada celah. Jika tidak, itu cocok dengan token. Default ke `True`.
  - Pola Regex: Pola Regex yang mendefinisikan proses tokenisasi. Default ke `' \\ s+'`.
  - Untuk huruf kecil: Jika dipilih, Data Wrangler mengonversi semua karakter menjadi huruf kecil sebelum tokenisasi. Default ke `True`.

  Untuk mempelajari lebih lanjut, lihat dokumentasi Spark di Tokenizer.

- Vectorizer: Vectorizer mengubah daftar token menjadi vektor numerik jarang. Setiap token sesuai dengan indeks dalam vektor dan bukan nol menunjukkan keberadaan token dalam kalimat input. Anda dapat memilih dari dua opsi vectorizer, Count dan Hashing.
  - Count vectorize memungkinkan penyesuaian yang memfilter token yang jarang atau terlalu umum. Parameter vektorisasi hitung meliputi yang berikut:
    - Frekuensi istilah minimum: Di setiap baris, istilah (token) dengan frekuensi yang lebih kecil disaring. Jika Anda menentukan bilangan bulat, ini adalah ambang absolut (inklusif). Jika Anda menentukan pecahan antara 0 (inklusif) dan 1, ambang batas relatif terhadap jumlah suku total. Default ke `1`.
    - Frekuensi dokumen minimum: Jumlah baris minimum di mana istilah (token) harus muncul untuk disertakan. Jika Anda menentukan bilangan bulat, ini adalah ambang absolut (inklusif). Jika Anda menentukan pecahan antara 0 (inklusif) dan 1, ambang batas relatif terhadap jumlah suku total. Default ke `1`.
    - Frekuensi dokumen maksimum: Jumlah maksimum dokumen (baris) di mana istilah (token) dapat muncul untuk dimasukkan. Jika Anda menentukan bilangan bulat, ini adalah ambang absolut (inklusif). Jika Anda menentukan pecahan antara 0 (inklusif) dan 1, ambang batas relatif terhadap jumlah suku total. Default ke `0.999`.
    - Ukuran kosakata maksimum: Ukuran maksimum kosakata. Kosakata terdiri dari semua istilah (token) di semua baris kolom. Default ke `262144`.
    - Output biner: Jika dipilih, output vektor tidak termasuk jumlah penampilan suatu istilah dalam dokumen, melainkan merupakan indikator biner dari penampilannya. Default ke `False`.

    Untuk mempelajari lebih lanjut tentang opsi ini, lihat dokumentasi Spark di CountVectorizer.
  - Hashing secara komputasi lebih cepat. Parameter vektor hash meliputi yang berikut:
    - Jumlah fitur selama hashing: Sebuah hash vectorizer memetakan token ke indeks vektor sesuai dengan nilai hash mereka. Fitur ini menentukan jumlah nilai hash yang mungkin. Nilai yang besar menghasilkan lebih sedikit tabrakan antara nilai hash tetapi vektor keluaran dimensi yang lebih tinggi.

    Untuk mempelajari lebih lanjut tentang opsi ini, lihat dokumentasi Spark di FeatureHasher
- Terapkan IDF menerapkan transformasi IDF, yang mengalikan frekuensi istilah dengan frekuensi dokumen terbalik standar yang digunakan untuk penyematan TF-IDF. Parameter IDF meliputi:
  - Frekuensi dokumen minimum: Jumlah minimum dokumen (baris) di mana istilah (token) harus muncul untuk disertakan. Jika count_vectorize adalah vectorizer yang dipilih, kami sarankan

Anda menyimpan nilai default dan hanya memodifikasi bidang min_doc_freq dalam parameter Count vectorize. Default ke 5.

- Format output: Format output dari setiap baris.
  - Pilih Vektor untuk menghasilkan satu kolom dengan vektor jarang.
  - Pilih Flattened untuk membuat kolom untuk setiap kategori dengan variabel indikator apakah teks di kolom asli berisi nilai yang sama dengan kategori tersebut. Anda hanya dapat memilih diratakan ketika Vectorizer ditetapkan sebagai Count vectorizer.

Mengubah Seri Waktu

Di Data Wrangler, Anda dapat mengubah data deret waktu. Nilai dalam kumpulan data deret waktu diindeks ke waktu tertentu. Misalnya, kumpulan data yang menunjukkan jumlah pelanggan di toko untuk setiap jam dalam sehari adalah kumpulan data deret waktu. Tabel berikut menunjukkan contoh dataset deret waktu.

Jumlah pelanggan per jam di toko

Jumlah pelanggan	Waktu (jam)
4	09:00
10	10:00
14	11:00
25	12:00
20	13:00
18	14:00

Untuk tabel sebelumnya, kolom Jumlah Pelanggan berisi data deret waktu. Data deret waktu diindeks pada data per jam di kolom Waktu (jam).

Anda mungkin perlu melakukan serangkaian transformasi pada data Anda untuk mendapatkannya dalam format yang dapat Anda gunakan untuk analisis Anda. Gunakan grup transformasi deret waktu untuk mengubah data deret waktu Anda. Untuk informasi selengkapnya tentang transformasi yang dapat Anda lakukan, lihat bagian berikut.

Topik

Kelompokkan berdasarkan Time Series

Anda dapat menggunakan grup berdasarkan operasi untuk mengelompokkan data deret waktu untuk nilai tertentu dalam kolom.

Misalnya, Anda memiliki tabel berikut yang melacak rata-rata penggunaan listrik harian dalam rumah tangga.

Rata-rata penggunaan listrik rumah tangga harian

ID Rumah Tangga	Stempel waktu harian	Penggunaan listrik (kWh)	Jumlah penghuni rumah tangga
rumah tangga_0	1/1/2020	30	2
rumah tangga_0	1/2/2020	40	2
rumah tangga_0	1/4/2020	35	3
rumah tangga_1	1/2/2020	45	3
rumah tangga_1	1/3/2020	55	4

Jika Anda memilih untuk mengelompokkan berdasarkan ID, Anda mendapatkan tabel berikut.

Penggunaan listrik dikelompokkan berdasarkan ID rumah tangga

ID Rumah Tangga	Seri penggunaan listrik (kWh)	Jumlah seri penghuni rumah tangga
rumah tangga_0	[30, 40, 35]	[2, 2, 3]
rumah tangga_1	[45, 55]	[3, 4]

Setiap entri dalam urutan deret waktu diurutkan oleh stempel waktu yang sesuai. Elemen pertama dari urutan sesuai dengan stempel waktu pertama dari seri. Untuk`household_0`, `30` adalah nilai pertama dari Seri Penggunaan Listrik. Nilai `30` sesuai dengan stempel waktu pertama. `1/1/2020`

Anda dapat menyertakan stempel waktu awal dan stempel waktu akhir. Tabel berikut menunjukkan bagaimana informasi itu muncul.

Penggunaan listrik dikelompokkan berdasarkan ID rumah tangga

ID Rumah Tangga	Seri penggunaan listrik (kWh)	Jumlah seri penghuni rumah tangga	Mulai_waktu	Waktu_akhir
rumah tangga_0	[30, 40, 35]	[2, 2, 3]	1/1/2020	1/4/2020
rumah tangga_1	[45, 55]	[3, 4]	1/2/2020	1/3/2020

Anda dapat menggunakan prosedur berikut untuk mengelompokkan berdasarkan kolom deret waktu.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Time Series.
5. Di bawah Transform, pilih Group by.
6. Tentukan kolom di Grup menurut kolom ini.
7. Untuk Terapkan ke kolom, tentukan nilai.
8. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
9. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Sampel Ulang Data Seri Waktu

Data deret waktu biasanya memiliki pengamatan yang tidak diambil secara berkala. Misalnya, kumpulan data dapat memiliki beberapa pengamatan yang direkam setiap jam dan pengamatan lain yang dicatat setiap dua jam.

Banyak analisis, seperti algoritma peramalan, memerlukan pengamatan yang harus dilakukan secara berkala. Resampling memberi Anda kemampuan untuk menetapkan interval reguler untuk pengamatan dalam kumpulan data Anda.

Anda dapat melakukan upsample atau downsample deret waktu. Downsampling meningkatkan interval antara pengamatan dalam dataset. Misalnya, jika Anda menurunkan sampel pengamatan yang diambil setiap jam atau setiap dua jam, setiap pengamatan dalam kumpulan data Anda dilakukan setiap dua jam. Pengamatan per jam dikumpulkan menjadi satu nilai menggunakan metode agregasi seperti mean atau median.

Upsampling mengurangi interval antara pengamatan dalam dataset. Misalnya, jika Anda mengambil sampel pengamatan yang dilakukan setiap dua jam ke dalam pengamatan per jam, Anda dapat menggunakan metode interpolasi untuk menyimpulkan pengamatan per jam dari pengamatan yang dilakukan setiap dua jam. Untuk informasi tentang metode interpolasi, lihat panda. DataFrame.interpolasi.

Anda dapat mengambil sampel ulang data numerik dan non-numerik.

Gunakan operasi Sampel Ulang untuk mengambil sampel ulang data deret waktu Anda. Jika Anda memiliki beberapa deret waktu dalam kumpulan data Anda, Data Wrangler menstandarisasi interval waktu untuk setiap deret waktu.

Tabel berikut menunjukkan contoh data deret waktu downsampling dengan menggunakan mean sebagai metode agregasi. Data di-downsample dari setiap dua jam menjadi setiap jam.

Pembacaan suhu per jam selama sehari sebelum downsampling

Stempel Waktu	Suhu (Celcius)
12:00	30
1:00	32
2:00	35

Stempel Waktu	Suhu (Celcius)
3:00	32
4:00	30

Pembacaan suhu diturunkan sampelnya menjadi setiap dua jam

Stempel Waktu	Suhu (Celcius)
12:00	30
2:00	33.5
4:00	35

Anda dapat menggunakan prosedur berikut untuk mengambil sampel ulang data deret waktu.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Sampel Ulang.
5. Untuk Timestamp, pilih kolom timestamp.
6. Untuk unit Frekuensi, tentukan frekuensi yang Anda resampling.
7. (Opsional) Tentukan nilai untuk kuantitas Frekuensi.
8. Konfigurasikan transformasi dengan menentukan bidang yang tersisa.
9. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
10. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Menangani Data Seri Waktu yang Hilang

Jika Anda memiliki nilai yang hilang dalam kumpulan data Anda, Anda dapat melakukan salah satu hal berikut:

- Untuk kumpulan data yang memiliki beberapa deret waktu, lepaskan deret waktu yang memiliki nilai hilang yang lebih besar dari ambang batas yang Anda tentukan.

- Imputasi nilai yang hilang dalam deret waktu dengan menggunakan nilai lain dalam deret waktu.

Mengimplikasikan nilai yang hilang melibatkan penggantian data dengan menentukan nilai atau dengan menggunakan metode inferensial. Berikut ini adalah metode yang dapat Anda gunakan untuk imputasi:

- Nilai konstan — Ganti semua data yang hilang dalam dataset Anda dengan nilai yang Anda tentukan.
- Nilai paling umum — Ganti semua data yang hilang dengan nilai yang memiliki frekuensi tertinggi dalam kumpulan data.
- Forward fill — Gunakan forward fill untuk mengganti nilai yang hilang dengan nilai yang tidak hilang yang mendahului nilai yang hilang. Untuk urutan: [2, 4, 7, NaN, NaN, NaN, 8], semua nilai yang hilang diganti dengan 7. Urutan yang dihasilkan dari penggunaan isian maju adalah [2, 4, 7, 7, 7, 7, 8].
- Isi mundur - Gunakan pengisian mundur untuk mengganti nilai yang hilang dengan nilai yang tidak hilang yang mengikuti nilai yang hilang. Untuk urutan: [2, 4, 7, NaN, NaN, NaN, 8], semua nilai yang hilang diganti dengan 8. Urutan yang dihasilkan dari penggunaan pengisian mundur adalah [2, 4, 7, 8, 8, 8, 8].
- Interpolasi — Menggunakan fungsi interpolasi untuk menghitung nilai yang hilang. Untuk informasi lebih lanjut tentang fungsi yang dapat Anda gunakan untuk interpolasi, lihat panda. DataFrame.interpolasi.

Beberapa metode imputasi mungkin tidak dapat memperhitungkan semua nilai yang hilang dalam kumpulan data Anda. Misalnya, Forward fill tidak dapat menyiratkan nilai yang hilang yang muncul di awal deret waktu. Anda dapat mengimputasi nilai dengan menggunakan isian maju atau pengisian mundur.

Anda dapat memasukkan nilai yang hilang di dalam sel atau di dalam kolom.

Contoh berikut menunjukkan bagaimana nilai-nilai yang diperhitungkan dalam sel.

Penggunaan listrik dengan nilai yang hilang

ID Rumah Tangga	Seri penggunaan listrik (kWh)
rumah tangga_0	[30, 40, 35, NaN, NaN]

ID Rumah Tangga	Seri penggunaan listrik (kWh)
rumah tangga_1	[45, NaN, 55]

Penggunaan listrik dengan nilai yang diperhitungkan menggunakan pengisian ke depan

ID Rumah Tangga	Seri penggunaan listrik (kWh)
rumah tangga_0	[30, 40, 35, 35, 35]
rumah tangga_1	[45, 45, 55]

Contoh berikut menunjukkan bagaimana nilai-nilai diperhitungkan dalam kolom.

Rata-rata penggunaan listrik rumah tangga harian dengan nilai yang hilang

ID Rumah Tangga	Penggunaan listrik (kWh)
rumah tangga_0	30
rumah tangga_0	40
rumah tangga_0	NaN
rumah tangga_1	NaN
rumah tangga_1	NaN

Rata-rata penggunaan listrik rumah tangga harian dengan nilai yang diperhitungkan menggunakan pengisian ke depan

ID Rumah Tangga	Penggunaan listrik (kWh)
rumah tangga_0	30
rumah tangga_0	40
rumah tangga_0	40

ID Rumah Tangga	Penggunaan listrik (kWh)
rumah tangga_1	40
rumah tangga_1	40

Anda dapat menggunakan prosedur berikut untuk menangani nilai yang hilang.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Handle hilang.
5. Untuk jenis input deret waktu, pilih apakah Anda ingin menangani nilai yang hilang di dalam sel atau di sepanjang kolom.
6. Untuk Impute nilai yang hilang untuk kolom ini, tentukan kolom yang memiliki nilai yang hilang.
7. Untuk Metode untuk menghitung nilai, pilih metode.
8. Konfigurasikan transformasi dengan menentukan bidang yang tersisa.
9. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
10. Jika Anda memiliki nilai yang hilang, Anda dapat menentukan metode untuk mengimplikasikan mereka di bawah Metode untuk memasukkan nilai.
11. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Validasi Stempel Waktu Data Deret Waktu Anda

Anda mungkin memiliki data stempel waktu yang tidak valid. Anda dapat menggunakan fungsi Validasi stempel waktu untuk menentukan apakah stempel waktu dalam kumpulan data Anda valid. Stempel waktu Anda dapat menjadi tidak valid karena satu atau beberapa alasan berikut:

- Kolom stempel waktu Anda memiliki nilai yang hilang.
- Nilai di kolom stempel waktu Anda tidak diformat dengan benar.

Jika Anda memiliki stempel waktu yang tidak valid dalam kumpulan data, Anda tidak dapat melakukan analisis dengan sukses. Anda dapat menggunakan Data Wrangler untuk mengidentifikasi stempel waktu yang tidak valid dan memahami di mana Anda perlu membersihkan data Anda.

Validasi deret waktu bekerja dalam salah satu dari dua cara:

Anda dapat mengonfigurasi Data Wrangler untuk melakukan salah satu hal berikut jika menemukan nilai yang hilang dalam kumpulan data Anda:

- Jatuhkan baris yang memiliki nilai hilang atau tidak valid.
- Identifikasi baris yang memiliki nilai hilang atau tidak valid.
- Lempar kesalahan jika menemukan nilai yang hilang atau tidak valid di kumpulan data Anda.

Anda dapat memvalidasi stempel waktu pada kolom yang memiliki `timestamp` tipe atau jenisnya. `string` Jika kolom memiliki `string` tipe, Data Wrangler mengubah jenis kolom ke `timestamp` dan melakukan validasi.

Anda dapat menggunakan prosedur berikut untuk memvalidasi stempel waktu dalam kumpulan data Anda.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Validasi stempel waktu.
5. Untuk Timestamp Column, pilih kolom timestamp.
6. Untuk Kebijakan, pilih apakah Anda ingin menangani stempel waktu yang hilang.
7. (Opsional) Untuk kolom Output, tentukan nama untuk kolom output.
8. Jika kolom waktu tanggal diformat untuk jenis string, pilih Cast to datetime.
9. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
10. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Standarisasi Panjang Deret Waktu

Jika Anda memiliki data deret waktu yang disimpan sebagai array, Anda dapat menstandarisasi setiap deret waktu dengan panjang yang sama. Standarisasi panjang array deret waktu mungkin memudahkan Anda untuk melakukan analisis pada data.

Anda dapat menstandarisasi deret waktu Anda untuk transformasi data yang memerlukan panjang data Anda untuk diperbaiki.

Banyak algoritma ML mengharuskan Anda untuk meratakan data deret waktu Anda sebelum Anda menggunakannya. Meratakan data deret waktu memisahkan setiap nilai deret waktu menjadi kolomnya sendiri dalam kumpulan data. Jumlah kolom dalam kumpulan data tidak dapat berubah, sehingga panjang deret waktu perlu distandarisasi antara Anda meratakan setiap array menjadi satu set fitur.

Setiap deret waktu diatur ke panjang yang Anda tentukan sebagai kuantil atau persentil dari rangkaian deret waktu. Misalnya, Anda dapat memiliki tiga urutan yang memiliki panjang sebagai berikut:

- 3
- 4
- 5

Anda dapat mengatur panjang semua urutan sebagai panjang urutan yang memiliki panjang persentil ke-50.

Array deret waktu yang lebih pendek dari panjang yang Anda tentukan memiliki nilai yang hilang ditambahkan. Berikut ini adalah contoh format standarisasi deret waktu ke panjang yang lebih panjang: [2, 4, 5, NaN, NaN, NaN].

Anda dapat menggunakan pendekatan yang berbeda untuk menangani nilai yang hilang. Untuk informasi tentang pendekatan tersebut, lihatMenangani Data Seri Waktu yang Hilang.

Array deret waktu yang lebih panjang dari panjang yang Anda tentukan terpotong.

Anda dapat menggunakan prosedur berikut untuk menstandarisasi panjang deret waktu.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Standarisasi panjang.
5. Untuk Standarisasi panjang deret waktu untuk kolom, pilih kolom.
6. (Opsional) Untuk kolom Output, tentukan nama untuk kolom output. Jika Anda tidak menentukan nama, transformasi dilakukan di tempat.
7. Jika kolom datetime diformat untuk jenis string, pilih Cast to datetime.

8. Pilih Cutoff quantile dan tentukan kuantil untuk mengatur panjang urutan.
9. Pilih Ratakan output untuk menampilkan nilai deret waktu ke dalam kolom terpisah.
10. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
11. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Ekstrak Fitur dari Data Seri Waktu Anda

Jika Anda menjalankan klasifikasi atau algoritma regresi pada data deret waktu Anda, sebaiknya ekstrak fitur dari deret waktu sebelum menjalankan algoritme. Mengekstrak fitur dapat meningkatkan kinerja algoritme Anda.

Gunakan opsi berikut untuk memilih bagaimana Anda ingin mengekstrak fitur dari data Anda:

- Gunakan subset Minimal untuk menentukan ekstraksi 8 fitur yang Anda tahu berguna dalam analisis hilir. Anda dapat menggunakan subset minimal saat Anda perlu melakukan perhitungan dengan cepat. Anda juga dapat menggunakannya ketika algoritme ML Anda memiliki risiko overfitting yang tinggi dan Anda ingin menyediakannya dengan lebih sedikit fitur.
- Gunakan subset Efisien untuk menentukan penggalian fitur sebanyak mungkin tanpa mengekstraksi fitur yang intensif secara komputasi dalam analisis Anda.
- Gunakan Semua fitur untuk menentukan ekstraksi semua fitur dari seri lagu.
- Gunakan subset Manual untuk memilih daftar fitur yang menurut Anda menjelaskan variasi data Anda dengan baik.

Gunakan prosedur berikut ini untuk mengekstrak fitur dari data deret waktu Anda.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih fitur Ekstrak.
5. Untuk fitur Ekstrak untuk kolom ini, pilih kolom.
6. (Opsional) Pilih Ratakan untuk menampilkan fitur ke kolom terpisah.
7. Untuk Strategi, pilih strategi untuk mengekstrak fitur.
8. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
9. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Gunakan Fitur Lagged dari Data Time Series Anda

Untuk banyak kasus penggunaan, cara terbaik untuk memprediksi perilaku future dari time series Anda adalah dengan menggunakan perilaku terbarunya.

Penggunaan paling umum dari fitur lagged adalah sebagai berikut:

- Mengumpulkan beberapa nilai masa lalu. Misalnya, untuk waktu, t + 1, Anda mengumpulkan t, t - 1, t - 2, dan t - 3.
- Mengumpulkan nilai-nilai yang sesuai dengan perilaku musiman dalam data. Misalnya, untuk memprediksi hunian di restoran pada pukul 13:00, Anda mungkin ingin menggunakan fitur mulai pukul 13.00 pada hari sebelumnya. Menggunakan fitur dari 12:00 PM atau 11:00 AM pada hari yang sama mungkin tidak prediktif seperti menggunakan fitur dari hari-hari sebelumnya.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih fitur Lag.
5. Untuk Menghasilkan fitur lag untuk kolom ini, pilih kolom.
6. Untuk Timestamp Column, pilih kolom yang berisi stempel waktu.
7. Untuk Lag, tentukan durasi lag.
8. (Opsional) Konfigurasikan output menggunakan salah satu opsi berikut:
    - Sertakan seluruh jendela lag
    - Ratakan output
    - Jatuhkan baris tanpa riwayat
9. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
10. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Buat Rentang Datetime Dalam Seri Waktu Anda

Anda mungkin memiliki data deret waktu yang tidak memiliki stempel waktu. Jika Anda tahu bahwa pengamatan dilakukan secara berkala, Anda dapat menghasilkan stempel waktu untuk deret waktu di kolom terpisah. Untuk menghasilkan stempel waktu, Anda menentukan nilai untuk stempel waktu awal dan frekuensi stempel waktu.

Misalnya, Anda mungkin memiliki data deret waktu berikut untuk jumlah pelanggan di restoran.

Data deret waktu tentang jumlah pelanggan di restoran

Jumlah pelanggan
10
14
24
40
30
20

Jika Anda tahu bahwa restoran dibuka pada pukul 17:00 dan pengamatan dilakukan setiap jam, Anda dapat menambahkan kolom stempel waktu yang sesuai dengan data deret waktu. Anda dapat melihat kolom timestamp pada tabel berikut.

Data deret waktu tentang jumlah pelanggan di restoran

Jumlah pelanggan	Stempel Waktu
10	1:00PM
14	14:00
24	15:00 SORE
40	16:00 SORE
30	17:00
20	6:00 SORE

Gunakan prosedur berikut untuk menambahkan rentang datetime ke data Anda.

1. Buka aliran data Wrangler Data Anda.

2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih rentang Datetime.
5. Untuk tipe Frekuensi, pilih unit yang digunakan untuk mengukur frekuensi stempel waktu.
6. Untuk Memulai stempel waktu, tentukan stempel waktu mulai.
7. Untuk kolom Output, tentukan nama untuk kolom output.
8. (Opsional) Konfigurasikan output menggunakan bidang yang tersisa.
9. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
10. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Gunakan Jendela Bergulir Dalam Seri Waktu Anda

Anda dapat mengekstrak fitur selama periode waktu tertentu. Misalnya, untuk waktu, t, dan panjang jendela waktu 3, dan untuk baris yang menunjukkan stempel waktu t th, kami menambahkan fitur yang diekstraksi dari deret waktu pada waktu t - 3, t -2, dan t - 1. Untuk informasi tentang mengekstraksi fitur, lihatEkstrak Fitur dari Data Seri Waktu Anda.

Anda dapat menggunakan prosedur berikut untuk mengekstrak fitur selama periode waktu tertentu.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih fitur jendela bergulir.
5. Untuk Menghasilkan fitur jendela bergulir untuk kolom ini, pilih kolom.
6. Untuk Timestamp Column, pilih kolom yang berisi stempel waktu.
7. (Opsional) Untuk Kolom Keluaran, tentukan nama kolom output.
8. Untuk ukuran jendela, tentukan ukuran jendela.
9. Untuk Strategi, pilih strategi ekstraksi.
10. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
11. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Featurize Datetime

Gunakan Featurize tanggal/waktu untuk membuat embedding vektor yang mewakili bidang datetime. Untuk menggunakan transformasi ini, data datetime Anda harus dalam salah satu format berikut:

- String yang menjelaskan datetime: Misalnya,. `"January 1st, 2020, 12:44pm"`
- Stempel waktu Unix: Stempel waktu Unix menggambarkan jumlah detik, milidetik, mikrodetik, atau nanodetik dari 1/1/1970.

Anda dapat memilih untuk Menyimpulkan format datetime dan menyediakan format Datetime. Jika Anda menyediakan format datetime, Anda harus menggunakan kode yang dijelaskan dalam dokumentasi Python. Opsi yang Anda pilih untuk dua konfigurasi ini memiliki implikasi untuk kecepatan operasi dan hasil akhir.

- Opsi paling manual dan tercepat secara komputasi adalah menentukan format Datetime dan pilih No for Infer datetime format.
- Untuk mengurangi tenaga kerja manual, Anda dapat memilih format Inf datetime dan tidak menentukan format datetime. Ini juga merupakan operasi komputasi cepat; Namun, format datetime pertama yang ditemui di kolom input diasumsikan sebagai format untuk seluruh kolom. Jika ada format lain di kolom, nilai-nilai ini adalah NaN di output akhir. Menyimpulkan format datetime dapat memberi Anda string yang tidak diurai.
- Jika Anda tidak menentukan format dan memilih No for Infer datetime format, Anda mendapatkan hasil yang paling kuat. Semua string datetime yang valid diuraikan. Namun, operasi ini bisa menjadi urutan besarnya lebih lambat dari dua opsi pertama dalam daftar ini.

Bila Anda menggunakan transformasi ini, Anda menentukan kolom Input yang berisi data datetime dalam salah satu format yang tercantum di atas. Transformasi menciptakan kolom output bernama Output nama kolom. Format kolom output tergantung pada konfigurasi Anda menggunakan yang berikut ini:

- Vektor: Mengeluarkan satu kolom sebagai vektor.
- Kolom: Membuat kolom baru untuk setiap fitur. Misalnya, jika output berisi tahun, bulan, dan hari, tiga kolom terpisah dibuat untuk tahun, bulan, dan hari.

Selain itu, Anda harus memilih mode Embedding. Untuk model linier dan jaringan dalam, kami sarankan memilih siklik. Untuk algoritma berbasis pohon, kami sarankan memilih ordinal.

Format String

Transformasi string Format berisi operasi pemformatan string standar. Misalnya, Anda dapat menggunakan operasi ini untuk menghapus karakter khusus, menormalkan panjang string, dan memperbarui casing string.

Grup fitur ini berisi transformasi berikut. Semua transformasi mengembalikan salinan string di kolom Input dan menambahkan hasilnya ke kolom keluaran baru.

Nama	Fungsi
Pad kiri	Left-pad string dengan karakter Fill yang diberikan ke lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.
Pad kanan	Right-pad string dengan karakter Fill yang diberikan ke lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.
Tengah (pad di kedua sisi)	Tengah-pad string (tambahkan padding di kedua sisi string) dengan karakter Fill yang diberikan ke lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.
Tambahkan nol	Isi kiri string numerik dengan nol, hingga lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.
Strip kiri dan kanan	Mengembalikan salinan string dengan karakter utama dan trailing dihapus.
Strip karakter dari kiri	Mengembalikan salinan string dengan karakter utama dihapus.

Nama	Fungsi
Strip karakter dari kanan	Mengembalikan salinan string dengan karakter trailing dihapus.
Huruf kecil	Ubah semua huruf dalam teks menjadi huruf kecil.
Kasus besar	Ubah semua huruf dalam teks menjadi huruf besar.
Kapitalisasi	Kapitalisasi huruf pertama di setiap kalimat.
Swap kasus	Mengkonversi semua karakter huruf besar ke huruf kecil dan semua karakter huruf kecil untuk karakter huruf besar dari string yang diberikan, dan mengembalikannya.
Tambahkan awalan atau akhiran	Menambahkan awalan dan akhiran kolom string. Anda harus menentukan setidaknya satu dari Awalan dan Akhiran.
Hapus simbol	Menghapus simbol yang diberikan dari string. Semua karakter yang terdaftar dihapus. Default ke ruang putih.

Tangani Outlier

Model pembelajaran mesin sensitif terhadap distribusi dan jangkauan nilai fitur Anda. Pencilan, atau nilai langka, dapat berdampak negatif pada akurasi model dan menyebabkan waktu pelatihan yang lebih lama. Gunakan grup fitur ini untuk mendeteksi dan memperbarui outlier dalam kumpulan data Anda.

Saat Anda menentukan langkah transformasi Handle outlier, statistik yang digunakan untuk mendeteksi outlier dihasilkan pada data yang tersedia di Data Wrangler saat mendefinisikan langkah ini. Statistik yang sama ini digunakan saat menjalankan pekerjaan Data Wrangler.

Gunakan bagian berikut untuk mempelajari lebih lanjut tentang transformasi yang terkandung dalam grup ini. Anda menentukan nama Output dan masing-masing transformasi ini menghasilkan kolom output dengan data yang dihasilkan.

Outlier numerik deviasi standar yang kuat

Transformasi ini mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan statistik yang kuat untuk outlier.

Anda harus menentukan kuantil Atas dan kuantil Bawah untuk statistik yang digunakan untuk menghitung outlier. Anda juga harus menentukan jumlah Standar deviasi dari mana nilai harus bervariasi dari rata-rata untuk dianggap sebagai outlier. Misalnya, jika Anda menentukan 3 untuk Standar deviasi, nilai harus jatuh lebih dari 3 standar deviasi dari rata-rata untuk dianggap sebagai outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.
- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Pencilan Numerik Deviasi Standar

Transformasi ini mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan mean dan standar deviasi.

Anda menentukan jumlah Standar deviasi suatu nilai harus bervariasi dari rata-rata untuk dianggap sebagai outlier. Misalnya, jika Anda menentukan 3 untuk Standar deviasi, nilai harus jatuh lebih dari 3 standar deviasi dari rata-rata untuk dianggap sebagai outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.
- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Pencilan Numerik Kuantil

Gunakan transformasi ini untuk mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan kuantil. Anda dapat menentukan kuantil Atas dan kuantil Bawah. Semua nilai yang berada di atas kuantil atas atau di bawah kuantil bawah dianggap outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.
- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Pencilan Numerik Min-Max

Transformasi ini mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan ambang batas atas dan bawah. Gunakan metode ini jika Anda mengetahui nilai ambang batas yang mendemark outlier.

Anda menentukan ambang atas dan ambang bawah, dan jika nilai jatuh di atas atau di bawah ambang tersebut masing-masing, mereka dianggap outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.
- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Ganti Rare

Saat Anda menggunakan Ganti transformasi langka, Anda menentukan ambang batas dan Data Wrangler menemukan semua nilai yang memenuhi ambang batas tersebut dan menggantinya dengan string yang Anda tentukan. Misalnya, Anda mungkin ingin menggunakan transformasi ini untuk mengkategorikan semua outlier dalam kolom ke dalam kategori “Lainnya”.

- String pengganti: String yang digunakan untuk mengganti outlier.
- Ambang batas absolut: Kategori jarang terjadi jika jumlah instance kurang dari atau sama dengan ambang absolut ini.

- Ambang pecahan: Kategori jarang terjadi jika jumlah instance kurang dari atau sama dengan ambang fraksi ini dikalikan dengan jumlah baris.
- Kategori umum maksimum: Kategori maksimum yang tidak langka yang tersisa setelah operasi. Jika ambang batas tidak menyaring kategori yang cukup, mereka yang memiliki jumlah penampilan teratas diklasifikasikan sebagai tidak jarang. Jika disetel ke 0 (default), tidak ada batasan keras untuk jumlah kategori.

Tangani Nilai yang Hilang

Nilai yang hilang adalah kejadian umum dalam kumpulan data pembelajaran mesin. Dalam beberapa situasi, adalah tepat untuk menghitung data yang hilang dengan nilai yang dihitung, seperti nilai rata-rata atau kategoris umum. Anda dapat memproses nilai yang hilang menggunakan grup transformasi nilai Handle yang hilang. Grup ini berisi transformasi berikut.

Isi Hilang

Gunakan Fill missing transform untuk mengganti nilai yang hilang dengan nilai Fill yang Anda tentukan.

Impute Hilang

Gunakan transformasi yang hilang Impute untuk membuat kolom baru yang berisi nilai yang diperhitungkan di mana nilai yang hilang ditemukan dalam data kategoris dan numerik input. Konfigurasi tergantung pada tipe data Anda.

Untuk data numerik, pilih strategi imputing, strategi yang digunakan untuk menentukan nilai baru yang akan diperhitungkan. Anda dapat memilih untuk menghitung mean atau median atas nilai yang ada dalam kumpulan data Anda. Data Wrangler menggunakan nilai yang dihitung untuk menghitung nilai yang hilang.

Untuk data kategoris, Data Wrangler menyiratkan nilai yang hilang menggunakan nilai yang paling sering di kolom. Untuk memasukkan string kustom, gunakan Fill missing transform sebagai gantinya.

Tambahkan Indikator untuk Hilang

Gunakan indikator Tambah untuk transformasi yang hilang untuk membuat kolom indikator baru, yang berisi Boolean `"false"` jika baris berisi nilai, dan `"true"` jika baris berisi nilai yang hilang.

Jatuhkan Hilang

Gunakan opsi Drop missing untuk menjatuhkan baris yang berisi nilai yang hilang dari kolom Input.

Kelola Kolom

Anda dapat menggunakan transformasi berikut untuk memperbarui dan mengelola kolom dengan cepat di kumpulan data Anda:

Nama	Fungsi
Jatuhkan Kolom	Hapus kolom.
Kolom Duplikat	Duplikat kolom.
Ganti Nama Kolom	Ganti nama kolom.
Pindahkan Kolom	Pindahkan lokasi kolom dalam kumpulan data. Pilih untuk memindahkan kolom Anda ke awal atau akhir kumpulan data, sebelum atau sesudah kolom referensi, atau ke indeks tertentu.

Kelola Baris

Gunakan grup transformasi ini untuk dengan cepat melakukan operasi pengurutan dan pengocokan pada baris. Grup ini berisi yang berikut:

- Urutkan: Urutkan seluruh kerangka data dengan kolom tertentu. Pilih kotak centang di sebelah Urutan naik untuk opsi ini; jika tidak, batalkan centang kotak dan urutan menurun digunakan untuk pengurutan.
- Shuffle: Aduk semua baris dalam kumpulan data secara acak.

Kelola Vektor

Gunakan grup transformasi ini untuk menggabungkan atau meratakan kolom vektor. Grup ini berisi transformasi berikut.

- Merakit: Gunakan transformasi ini untuk menggabungkan vektor Spark dan data numerik menjadi satu kolom. Misalnya, Anda dapat menggabungkan tiga kolom: dua berisi data numerik dan satu berisi vektor. Tambahkan semua kolom yang ingin Anda gabungkan Kolom input dan tentukan nama kolom Output untuk data gabungan.

- Flatten: Gunakan transformasi ini untuk meratakan satu kolom yang berisi data vektor. Kolom input harus berisi PySpark vektor atau objek seperti array. Anda dapat mengontrol jumlah kolom yang dibuat dengan menentukan Metode untuk mendeteksi jumlah output. Misalnya, jika Anda memilih Panjang vektor pertama, jumlah elemen dalam vektor atau larik valid pertama yang ditemukan di kolom menentukan jumlah kolom keluaran yang dibuat. Semua vektor input lainnya dengan terlalu banyak item terpotong. Masukan dengan terlalu sedikit item diisi dengan NaNs.

  Anda juga menentukan awalan Output, yang digunakan sebagai awalan untuk setiap kolom output.

Proses Numerik

Gunakan grup fitur Process Numeric untuk memproses data numerik. Setiap skalar dalam grup ini didefinisikan menggunakan perpustakaan Spark. Skalar berikut didukung:

- Standard Scaler: Standarisasi kolom input dengan mengurangi rata-rata dari setiap nilai dan penskalaan ke varians unit. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk StandardScaler.
- Robust Scaler: Skala kolom input menggunakan statistik yang kuat untuk outlier. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk RobustScaler.
- Min Max Scaler: Ubah kolom input dengan menskalakan setiap fitur ke rentang tertentu. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk MinMaxScaler.
- Max Absolute Scaler: Skala kolom input dengan membagi setiap nilai dengan nilai absolut maksimum. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk MaxAbsScaler.

Pengambilan sampel

Setelah mengimpor data, Anda dapat menggunakan transformator Sampling untuk mengambil satu atau lebih sampelnya. Saat Anda menggunakan transformator sampling, Data Wrangler mengambil sampel kumpulan data asli Anda.

Anda dapat memilih salah satu metode sampel berikut:

- Batas: Sampel kumpulan data mulai dari baris pertama hingga batas yang Anda tentukan.
- Acak: Mengambil sampel acak dari ukuran yang Anda tentukan.
- Bertingkat: Mengambil sampel acak bertingkat.

Anda dapat membuat stratifikasi sampel acak untuk memastikan bahwa sampel tersebut mewakili distribusi asli kumpulan data.

Anda mungkin melakukan persiapan data untuk beberapa kasus penggunaan. Untuk setiap kasus penggunaan, Anda dapat mengambil sampel yang berbeda dan menerapkan serangkaian transformasi yang berbeda.

Prosedur berikut menjelaskan proses pembuatan sampel acak.

Untuk mengambil sampel acak dari data Anda.

1. Pilih + di sebelah kanan kumpulan data yang telah Anda impor. Nama dataset Anda terletak di bawah +.
2. Pilih Tambahkan transformasi.
3. Pilih Pengambilan sampel.
4. Untuk metode Sampling, pilih metode sampling.
5. Untuk Perkiraan ukuran sampel, pilih perkiraan jumlah pengamatan yang Anda inginkan dalam sampel Anda.
6. (Opsional) Tentukan bilangan bulat untuk benih Acak untuk membuat sampel yang dapat direproduksi.

Prosedur berikut menjelaskan proses pembuatan sampel bertingkat.

Untuk mengambil sampel bertingkat dari data Anda.

1. Pilih + di sebelah kanan kumpulan data yang telah Anda impor. Nama dataset Anda terletak di bawah +.
2. Pilih Tambahkan transformasi.
3. Pilih Pengambilan sampel.
4. Untuk metode Sampling, pilih metode sampling.
5. Untuk Perkiraan ukuran sampel, pilih perkiraan jumlah pengamatan yang Anda inginkan dalam sampel Anda.
6. Untuk kolom Stratify, tentukan nama kolom yang ingin Anda stratifikasi.
7. (Opsional) Tentukan bilangan bulat untuk benih Acak untuk membuat sampel yang dapat direproduksi.

Cari dan Edit

Gunakan bagian ini untuk mencari dan mengedit pola tertentu dalam string. Misalnya, Anda dapat menemukan dan memperbarui string dalam kalimat atau dokumen, membagi string dengan pembatas, dan menemukan kemunculan string tertentu.

Transformasi berikut didukung di bawah Cari dan edit. Semua transformasi mengembalikan salinan string di kolom Input dan menambahkan hasilnya ke kolom output baru.

Nama	Fungsi
Temukan substring	Mengembalikan indeks kejadian pertama dari Substring yang Anda cari, Anda dapat memulai dan mengakhiri pencarian di Mulai dan Akhir masing-masing.
Temukan substring (dari kanan)	Mengembalikan indeks kejadian terakhir dari Substring yang Anda cari. Anda dapat memulai dan mengakhiri pencarian di Start dan End masing-masing.
Awalan kecocokan	Mengembalikan nilai Boolean jika string berisi Pola yang diberikan. Sebuah pola dapat berupa urutan karakter atau ekspresi reguler. Secara opsional, Anda dapat membuat pola peka huruf besar/huruf besar.
Temukan semua kejadian	Mengembalikan array dengan semua kejadian dari pola yang diberikan. Sebuah pola dapat berupa urutan karakter atau ekspresi reguler.
Ekstrak menggunakan regex	Mengembalikan string yang cocok dengan pola Regex tertentu.
Ekstrak antara pembatas	Mengembalikan string dengan semua karakter yang ditemukan antara pembatas Kiri dan pembatas Kanan.

Nama	Fungsi
Ekstrak dari posisi	Mengembalikan string, mulai dari posisi Mulai dalam string input, yang berisi semua karakter hingga posisi awal ditambah Panjang.
Temukan dan ganti substring	Mengembalikan string dengan semua kecocokan dari Pola tertentu (ekspresi reguler) digantikan oleh string Penggantian.
Ganti antara pembatas	Mengembalikan string dengan substring ditemukan antara penampilan pertama pembatas Kiri dan penampilan terakhir dari pembatas Kanan digantikan oleh string Penggantian. Jika tidak ada kecocokan yang ditemukan, tidak ada yang diganti.
Ganti dari posisi	Mengembalikan string dengan substring antara posisi Mulai dan posisi Mulai ditambah Panjang diganti dengan string Penggantian. Jika posisi Mulai ditambah Panjang lebih besar dari panjang string pengganti, output berisi....
Konversi regex menjadi hilang	Mengkonversi string ke None jika tidak valid dan mengembalikan hasilnya. Validitas didefinisikan dengan ekspresi reguler dalam Pola.
Pisahkan string dengan pembatas	Mengembalikan array string dari string input, dibagi dengan Delimiter, dengan sampai jumlah Max split (opsional). Delimiter default ke spasi putih.

Membagi data

Gunakan transformasi data Split untuk membagi kumpulan data Anda menjadi dua atau tiga kumpulan data. Misalnya, Anda dapat membagi kumpulan data menjadi kumpulan data yang digunakan untuk melatih model dan kumpulan data yang digunakan untuk mengujinya. Anda

dapat menentukan proporsi dataset yang masuk ke setiap split. Misalnya, jika Anda membagi satu kumpulan data menjadi dua kumpulan data, kumpulan data pelatihan dapat memiliki 80% data sementara kumpulan data pengujian memiliki 20%.

Memisahkan data Anda menjadi tiga kumpulan data memberi Anda kemampuan untuk membuat kumpulan data pelatihan, validasi, dan pengujian. Anda dapat melihat seberapa baik kinerja model pada kumpulan data pengujian dengan menjatuhkan kolom target.

Kasus penggunaan Anda menentukan berapa banyak kumpulan data asli yang didapat masing-masing kumpulan data Anda dan metode yang Anda gunakan untuk membagi data. Misalnya, Anda mungkin ingin menggunakan pemisahan bertingkat untuk memastikan bahwa distribusi pengamatan di kolom target sama di seluruh kumpulan data. Anda dapat menggunakan transformasi split berikut:

- Pemisahan acak - Setiap pemisahan adalah sampel acak dan tidak tumpang tindih dari kumpulan data asli. Untuk kumpulan data yang lebih besar, menggunakan pemisahan acak mungkin mahal secara komputasi dan membutuhkan waktu lebih lama daripada pemisahan yang dipesan.
- Pemisahan berurutan — Membagi kumpulan data berdasarkan urutan pengamatan yang berurutan. Misalnya, untuk pemisahan uji kereta 80/20, pengamatan pertama yang membentuk 80% dari kumpulan data masuk ke kumpulan data pelatihan. 20% terakhir dari pengamatan pergi ke dataset pengujian. Pemisahan yang dipesan efektif dalam menjaga urutan data yang ada di antara pemisahan.
- Pemisahan bertingkat — Membagi kumpulan data untuk memastikan bahwa jumlah pengamatan di kolom input memiliki representasi proporsional. Untuk kolom input yang memiliki pengamatan 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, pemisahan 80/20 pada kolom berarti bahwa sekitar 80% dari 1s, 80% dari 2s, dan 80% dari 3s pergi ke set pelatihan. Sekitar 20% dari setiap jenis pengamatan pergi ke set pengujian.
- Split by key — Menghindari data dengan kunci yang sama terjadi di lebih dari satu split. Misalnya, jika Anda memiliki kumpulan data dengan kolom 'customer_id' dan Anda menggunakannya sebagai kunci, tidak ada id pelanggan di lebih dari satu split.

Setelah Anda membagi data, Anda dapat menerapkan transformasi tambahan ke setiap kumpulan data. Untuk sebagian besar kasus penggunaan, mereka tidak diperlukan.

Data Wrangler menghitung proporsi perpecahan untuk kinerja. Anda dapat memilih ambang kesalahan untuk mengatur keakuratan pemisahan. Ambang kesalahan yang lebih rendah lebih akurat mencerminkan proporsi yang Anda tentukan untuk pemisahan. Jika Anda menetapkan ambang

kesalahan yang lebih tinggi, Anda mendapatkan kinerja yang lebih baik, tetapi akurasi yang lebih rendah.

Untuk membagi data dengan sempurna, atur ambang kesalahan ke 0. Anda dapat menentukan ambang batas antara 0 dan 1 untuk kinerja yang lebih baik. Jika Anda menentukan nilai yang lebih besar dari 1, Data Wrangler menafsirkan nilai itu sebagai 1.

Jika Anda memiliki 10000 baris dalam kumpulan data Anda dan Anda menentukan pemisahan 80/20 dengan kesalahan 0,001, Anda akan mendapatkan pengamatan yang mendekati salah satu hasil berikut:

- 8010 pengamatan di set pelatihan dan 1990 di set pengujian
- 7990 pengamatan di set pelatihan dan 2010 di set pengujian

Jumlah pengamatan untuk pengujian yang ditetapkan dalam contoh sebelumnya adalah dalam interval antara 8010 dan 7990.

Secara default, Data Wrangler menggunakan seed acak untuk membuat split dapat direproduksi. Anda dapat menentukan nilai yang berbeda untuk benih untuk membuat pemisahan yang dapat direproduksi yang berbeda.

Randomized split

Gunakan prosedur berikut untuk melakukan pemisahan acak pada kumpulan data Anda.

Untuk membagi kumpulan data Anda secara acak, lakukan hal berikut

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.
3. Pilih Pisahkan data.
4. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.
5. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua split. Proporsi harus berjumlah 1.
6. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
7. (Opsional) Tentukan nilai untuk benih Acak.
8. Pilih Pratinjau.
9. Pilih Tambahkan.

Ordered split

Gunakan prosedur berikut untuk melakukan pemisahan berurutan pada kumpulan data Anda.

Untuk membuat pemisahan berurutan dalam kumpulan data Anda, lakukan hal berikut.

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.
3. Untuk Transform, pilih Pemisahan yang dipesan.
4. Pilih Pisahkan data.
5. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.
6. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua split. Proporsi harus berjumlah 1.
7. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
8. (Opsional) Untuk kolom Input, tentukan kolom dengan nilai numerik. Menggunakan nilai kolom untuk menyimpulkan catatan mana yang ada di setiap split. Nilai yang lebih kecil berada dalam satu split dengan nilai yang lebih besar di split lainnya.
9. (Opsional) Pilih Tangani duplikat untuk menambahkan noise ke nilai duplikat dan buat kumpulan data dengan nilai yang sepenuhnya unik.
10. (Opsional) Tentukan nilai untuk benih Acak.
11. Pilih Pratinjau.
12. Pilih Tambahkan.

Stratified split

Gunakan prosedur berikut untuk melakukan pemisahan bertingkat pada kumpulan data Anda.

Untuk membuat pemisahan bertingkat dalam kumpulan data Anda, lakukan hal berikut.

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.
3. Pilih Pisahkan data.
4. Untuk Transform, pilih Stratified split.
5. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.

6. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua split. Proporsi harus berjumlah 1.
7. Untuk kolom Input, tentukan kolom dengan hingga 100 nilai unik. Data Wrangler tidak dapat membuat stratifikasi kolom dengan lebih dari 100 nilai unik.
8. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
9. (Opsional) Tentukan nilai untuk benih acak untuk menentukan benih yang berbeda.
10. Pilih Pratinjau.
11. Pilih Tambahkan.

Split by column keys

Gunakan prosedur berikut untuk membagi dengan kunci kolom dalam dataset Anda.

Untuk membagi dengan kunci kolom dalam kumpulan data Anda, lakukan hal berikut.

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.
3. Pilih Pisahkan data.
4. Untuk Transform, pilih Split by key.
5. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.
6. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua split. Proporsi harus berjumlah 1.
7. Untuk kolom Kunci, tentukan kolom dengan nilai yang tidak ingin Anda tampilkan di kedua kumpulan data.
8. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
9. Pilih Pratinjau.
10. Pilih Tambahkan.

Parse Nilai sebagai Tipe

Gunakan transformasi ini untuk mentransmisikan kolom ke tipe baru. Tipe data Data Wrangler yang didukung adalah:

- Panjang

- Desimal
- Boolean
- Tanggal, dalam format DD-MM-YYYY, masing-masing mewakili hari, bulan, dan tahun.
- String

Validasi String

Gunakan transformasi string Validasi untuk membuat kolom baru yang menunjukkan bahwa baris data teks memenuhi kondisi tertentu. Misalnya, Anda dapat menggunakan transformasi string Validasi untuk memverifikasi bahwa string hanya berisi karakter huruf kecil. Transformasi berikut didukung di bawah Validasi string.

Transformasi berikut termasuk dalam grup transformasi ini. Jika transformasi menghasilkan nilai Boolean, `True` diwakili dengan a `1` dan `False` diwakili dengan a. `0`

Nama	Fungsi
Panjang tali	Mengembalikan `True` jika panjang string sama dengan panjang tertentu. Jika tidak, mengembalikan `False`.
Starts with	Mengembalikan `True` jika string dimulai akan awalan tertentu. Jika tidak, mengembalikan `False`.
Ends with	Mengembalikan `True` jika panjang string sama dengan panjang tertentu. Jika tidak, mengembalikan `False`.
Apakah alfanumerik	Mengembalikan `True` jika string hanya berisi angka dan huruf. Jika tidak, mengembalikan `False`.
Apakah alpha (huruf)	Mengembalikan `True` jika string hanya berisi huruf. Jika tidak, mengembalikan `False`.
Adalah digit	Mengembalikan `True` jika string hanya berisi digit. Jika tidak, mengembalikan `False`.

Nama	Fungsi
Adalah ruang	Mengembalikan `True` jika string hanya berisi angka dan huruf. Jika tidak, mengembalikan `False`.
Adalah judul	Mengembalikan `True` jika string berisi spasi putih. Jika tidak, mengembalikan `False`.
Adalah huruf kecil	Mengembalikan `True` jika string hanya berisi huruf kecil. Jika tidak, mengembalikan `False`.
Adalah huruf besar	Mengembalikan `True` jika string hanya berisi huruf besar. Jika tidak, mengembalikan `False`.
Adalah numerik	Mengembalikan `True` jika string hanya berisi angka. Jika tidak, mengembalikan `False`.
Adalah desimal	Mengembalikan `True` jika string hanya berisi angka desimal. Jika tidak, mengembalikan `False`.

Data JSON Unnest

Jika Anda memiliki file.csv, Anda mungkin memiliki nilai dalam kumpulan data Anda yang merupakan string JSON. Demikian pula, Anda mungkin memiliki data bersarang di kolom file Parket atau dokumen JSON.

Gunakan operator terstruktur Flatten untuk memisahkan kunci tingkat pertama menjadi kolom terpisah. Kunci tingkat pertama adalah kunci yang tidak bersarang dalam nilai.

Misalnya, Anda mungkin memiliki kumpulan data yang memiliki kolom orang dengan informasi demografis pada setiap orang yang disimpan sebagai string JSON. String JSON mungkin terlihat seperti berikut ini.

```
"{"seq": 1,"name": {"first": "Nathaniel","last": "Ferguson"},"age": 59,"city":
"Posbotno","state": "WV"}"
```

Operator terstruktur Flatten mengonversi kunci tingkat pertama berikut menjadi kolom tambahan dalam kumpulan data Anda:

- seq
- name
- usia
- kota
- status

Data Wrangler menempatkan nilai-nilai kunci sebagai nilai di bawah kolom. Berikut ini menunjukkan nama kolom dan nilai-nilai JSON.

```
seq, name,                                    age, city, state
1, {"first": "Nathaniel","last": "Ferguson"}, 59, Posbotno, WV
```

Untuk setiap nilai dalam kumpulan data Anda yang berisi JSON, operator terstruktur Flatten membuat kolom untuk kunci tingkat pertama. Untuk membuat kolom untuk kunci bersarang, panggil operator lagi. Untuk contoh sebelumnya, memanggil operator membuat kolom:

- name_first
- name_last

Contoh berikut menunjukkan kumpulan data yang dihasilkan dari pemanggilan operasi lagi.

```
seq, name,                                    age, city, state, name_first, name_last
1, {"first": "Nathaniel","last": "Ferguson"}, 59, Posbotno, WV, Nathaniel, Ferguson
```

Pilih Kunci untuk diratakan untuk menentukan kunci tingkat pertama yang ingin diekstrak sebagai kolom terpisah. Jika Anda tidak menentukan kunci apa pun, Data Wrangler mengekstrak semua kunci secara default.

Meledak Array

Gunakan Explode array untuk memperluas nilai array menjadi baris output terpisah. Misalnya, operasi dapat mengambil setiap nilai dalam array, [[1, 2, 3,], [4, 5, 6], [7, 8, 9]] dan membuat kolom baru dengan baris berikut:

```
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
```

Data Wrangler menamai kolom baru, input_column_name_flatten.

Anda dapat memanggil operasi array Explode beberapa kali untuk mendapatkan nilai bersarang dari array ke dalam kolom output terpisah. Contoh berikut menunjukkan hasil pemanggilan operasi beberapa kali pada dataset dengan array bersarang.

Menempatkan nilai-nilai array bersarang ke dalam kolom terpisah

id	array	id	array_item	id	array_items_items
1	[[kucing, anjingnya], [kelelawar, katak]]	1	[kucing, anjingnya]	1	kucing
2	[[mawar, petunia], [lily, daisy]]	1	[kelelawar, katak]	1	anjingnya
		2	[mawar, petunia]	1	kelelawar
		2	[bunga bakung, daisy]	1	katak
			2	2	mawar

id	array	id	array_item	id	array_items_items
			2	2	petunia
			2	2	bunga bakung
			2	2	bunga aster

Mengubah Data Gambar

Gunakan Data Wrangler untuk mengimpor dan mengubah gambar yang Anda gunakan untuk pipeline machine learning (ML) Anda. Setelah menyiapkan data gambar, Anda dapat mengekspornya dari aliran Data Wrangler ke pipeline MLmu.

Anda dapat menggunakan informasi yang disediakan di sini untuk membiasakan diri dengan mengimpor dan mengubah data gambar di Data Wrangler. Data Wrangler menggunakan OpenCV untuk mengimpor gambar. Untuk informasi selengkapnya tentang format gambar yang didukung, lihat Membaca dan menulis file gambar.

Setelah Anda membiasakan diri dengan konsep mengubah data gambar Anda, ikuti tutorial berikut, Siapkan data gambar dengan Amazon SageMaker Data Wrangler.

Industri dan kasus penggunaan berikut adalah contoh di mana menerapkan pembelajaran mesin ke data gambar yang diubah dapat berguna:

- Manufaktur - Mengidentifikasi cacat pada item dari jalur perakitan
- Makanan — Mengidentifikasi makanan busuk atau busuk
- Kedokteran — Mengidentifikasi lesi pada jaringan

Saat Anda bekerja dengan data gambar di Data Wrangler, Anda melalui proses berikut:

1. Impor - Pilih gambar dengan memilih direktori yang berisi mereka di bucket Amazon S3 Anda.
2. Transform - Gunakan transformasi bawaan untuk menyiapkan gambar untuk pipeline pembelajaran mesin Anda.
3. Ekspor - Ekspor gambar yang telah Anda ubah ke lokasi yang dapat diakses dari pipeline.

Gunakan prosedur berikut untuk mengimpor data gambar Anda.

Untuk mengimpor data gambar Anda

1. Arahkan ke halaman Buat koneksi.
2. Pilih Amazon S3.
3. Tentukan jalur file Amazon S3 yang berisi data gambar.
4. Untuk jenis File, pilih Gambar.
5. (Opsional) Pilih Impor direktori bersarang untuk mengimpor gambar dari beberapa jalur Amazon S3.
6. Pilih Impor.

Data Wrangler menggunakan pustaka imgaug sumber terbuka untuk transformasi gambar bawaannya. Anda dapat menggunakan transformasi bawaan berikut:

- ResizeImage
- EnhanceImage
- CorruptImage
- SplitImage
- DropCorruptedImages
- DropImageDuplicates
- Kecerahan
- ColorChannels
- Skala Abu-abu
- Putar

Gunakan prosedur berikut untuk mengubah gambar Anda tanpa menulis kode.

Untuk mengubah data gambar tanpa menulis kode

1. Dari alur Data Wrangler Anda, pilih + di sebelah node yang mewakili gambar yang telah Anda impor.
2. Pilih Tambahkan transformasi.

3. Pilih Tambahkan langkah.
4. Pilih transformasi dan konfigurasikan.
5. Pilih Pratinjau.
6. Pilih Tambahkan.

Selain menggunakan transformasi yang disediakan Data Wrangler, Anda juga dapat menggunakan cuplikan kode kustom Anda sendiri. Untuk informasi selengkapnya tentang menggunakan cuplikan kode kustom, lihat. Transformasi Kustom Anda dapat mengimpor pustaka OpenCV dan imgaug dalam cuplikan kode Anda dan menggunakan transformasi yang terkait dengannya. Berikut ini adalah contoh cuplikan kode yang mendeteksi tepi dalam gambar.

```
# A table with your image data is stored in the `df` variable
import cv2
import numpy as np
from pyspark.sql.functions import column

from sagemaker_dataprep.compute.operators.transforms.image.constants import
 DEFAULT_IMAGE_COLUMN, IMAGE_COLUMN_TYPE
from sagemaker_dataprep.compute.operators.transforms.image.decorators import
 BasicImageOperationDecorator, PandasUDFOperationDecorator


@BasicImageOperationDecorator
def my_transform(image: np.ndarray) -> np.ndarray:
  # To use the code snippet on your image data, modify the following lines within the
 function
    HYST_THRLD_1, HYST_THRLD_2 = 100, 200
    edges = cv2.Canny(image,HYST_THRLD_1,HYST_THRLD_2)
    return edges


@PandasUDFOperationDecorator(IMAGE_COLUMN_TYPE)
def custom_image_udf(image_row):
    return my_transform(image_row)


df = df.withColumn(DEFAULT_IMAGE_COLUMN,
 custom_image_udf(column(DEFAULT_IMAGE_COLUMN)))
```

Saat menerapkan transformasi dalam alur Data Wrangler Anda, Data Wrangler hanya menerapkannya pada sampel gambar dalam kumpulan data Anda. Untuk mengoptimalkan pengalaman Anda dengan aplikasi, Data Wrangler tidak menerapkan transformasi ke semua gambar Anda.

Filter data

Gunakan Data Wrangler untuk memfilter data di kolom Anda. Saat Anda memfilter data dalam kolom, Anda menentukan bidang berikut:

- Nama kolom — Nama kolom yang Anda gunakan untuk memfilter data.
- Kondisi - Jenis filter yang Anda terapkan pada nilai di kolom.
- Nilai - Nilai atau kategori di kolom tempat Anda menerapkan filter.

Anda dapat memfilter pada kondisi berikut:

- = — Mengembalikan nilai yang cocok dengan nilai atau kategori yang Anda tentukan.
- ! = — Mengembalikan nilai yang tidak cocok dengan nilai atau kategori yang Anda tentukan.
- >= — Untuk data Long atau Float, filter untuk nilai yang lebih besar dari atau sama dengan nilai yang Anda tentukan.
- <= — Untuk data Long atau Float, filter untuk nilai yang kurang dari atau sama dengan nilai yang Anda tentukan.
- > — Untuk data Long atau Float, filter untuk nilai yang lebih besar dari nilai yang Anda tentukan.
- < — Untuk data Long atau Float, filter untuk nilai yang kurang dari nilai yang Anda tentukan.

Untuk kolom yang memiliki kategori, `male` dan`female`, Anda dapat memfilter semua `male` nilai. Anda juga dapat memfilter untuk semua `female` nilai. Karena hanya ada `male` dan `female` nilai di kolom, filter mengembalikan kolom yang hanya memiliki `female` nilai.

Anda juga dapat menambahkan beberapa filter. Filter dapat diterapkan di beberapa kolom atau kolom yang sama. Misalnya, jika Anda membuat kolom yang hanya memiliki nilai dalam rentang tertentu, Anda menambahkan dua filter berbeda. Satu filter menentukan bahwa kolom harus memiliki nilai yang lebih besar dari nilai yang Anda berikan. Filter lain menentukan bahwa kolom harus memiliki nilai kurang dari nilai yang Anda berikan.

Gunakan prosedur berikut untuk menambahkan transformasi filter ke data Anda.

Untuk memfilter data Anda

1. Dari alur Data Wrangler Anda, pilih + di sebelah node dengan data yang Anda filter.
2. Pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Filter data.
5. Tentukan bidang berikut:
   - Nama kolom - Kolom yang Anda filter.
   - Kondisi — Kondisi filter.
   - Nilai - Nilai atau kategori di kolom tempat Anda menerapkan filter.
6. (Opsional) Pilih + mengikuti filter yang telah Anda buat.
7. Konfigurasikan filter.
8. Pilih Pratinjau.
9. Pilih Tambahkan.

Obrolan untuk persiapan data

> **Important**
>
> Untuk administrator:
>
> - Obrolan untuk persiapan data memerlukan `AmazonSageMakerCanvasAIServicesAccess` kebijakan. Lihat informasi yang lebih lengkap di AWSkebijakan terkelola: AmazonSageMakerCanvas AI ServicesAccess
> - Obrolan untuk persiapan data memerlukan akses ke Amazon Bedrock dan model Anthropic Claude di dalamnya. Untuk informasi selengkapnya, lihat Menambahkan akses model.
> - Anda harus menjalankan persiapan data SageMaker Canvas Wilayah AWS sama dengan Region tempat Anda menjalankan model Anda. Obrolan untuk persiapan data tersedia di AS Timur (Virginia N.), AS Barat (Oregon), dan Eropa (Frankfurt). Wilayah AWS

Selain menggunakan transformasi dan analisis bawaan, Anda dapat menggunakan bahasa alami untuk mengeksplorasi, memvisualisasikan, dan mengubah data Anda dalam antarmuka percakapan.

Dalam antarmuka percakapan, Anda dapat menggunakan kueri bahasa alami untuk memahami dan menyiapkan data Anda untuk membangun model ML.

Berikut ini adalah contoh dari beberapa petunjuk yang dapat Anda gunakan:

- Ringkas data saya
- Jatuhkan kolom *example-column-name*
- Ganti nilai yang hilang dengan median
- Plot histogram harga
- Apa barang paling mahal yang dijual?
- Berapa banyak barang berbeda yang dijual?
- Urutkan data berdasarkan wilayah

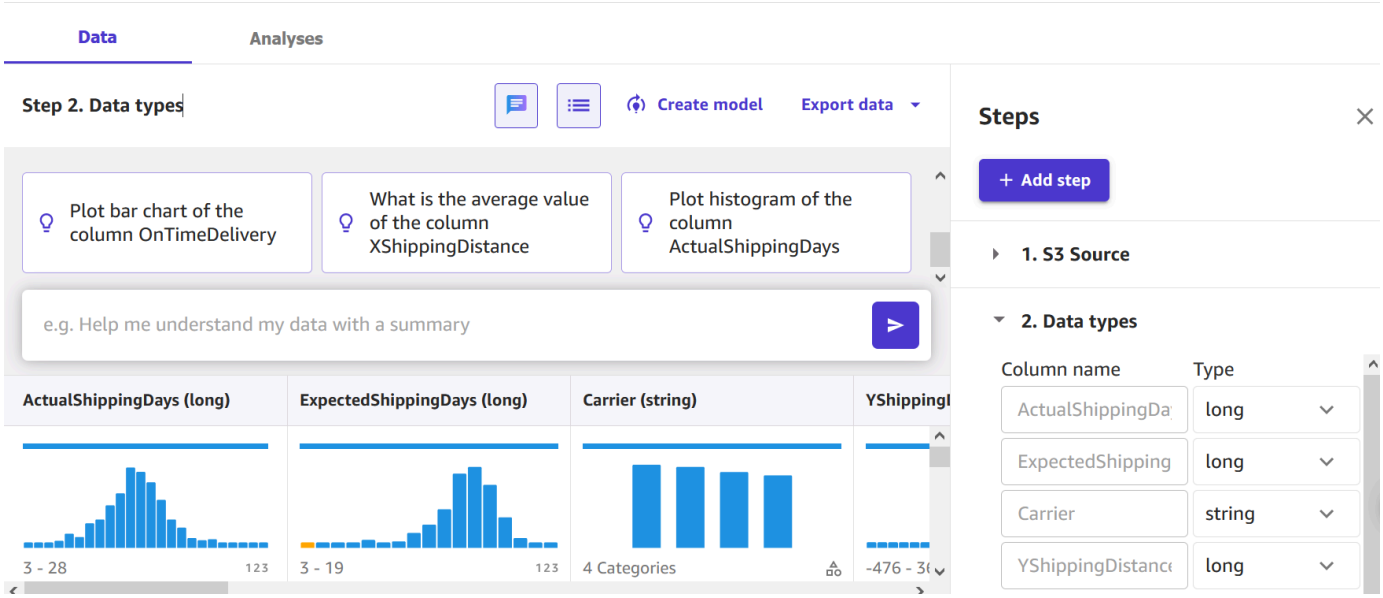
Saat mengubah data menggunakan prompt, Anda dapat melihat pratinjau yang menunjukkan bagaimana data diubah. Anda dapat memilih untuk menambahkannya sebagai langkah dalam aliran Data Wrangler Anda berdasarkan apa yang Anda lihat di pratinjau.

Tanggapan terhadap permintaan Anda menghasilkan kode untuk transformasi dan analisis Anda. Anda dapat memodifikasi kode untuk memperbarui output dari prompt. Misalnya, Anda dapat memodifikasi kode untuk analisis untuk mengubah nilai sumbu grafik.

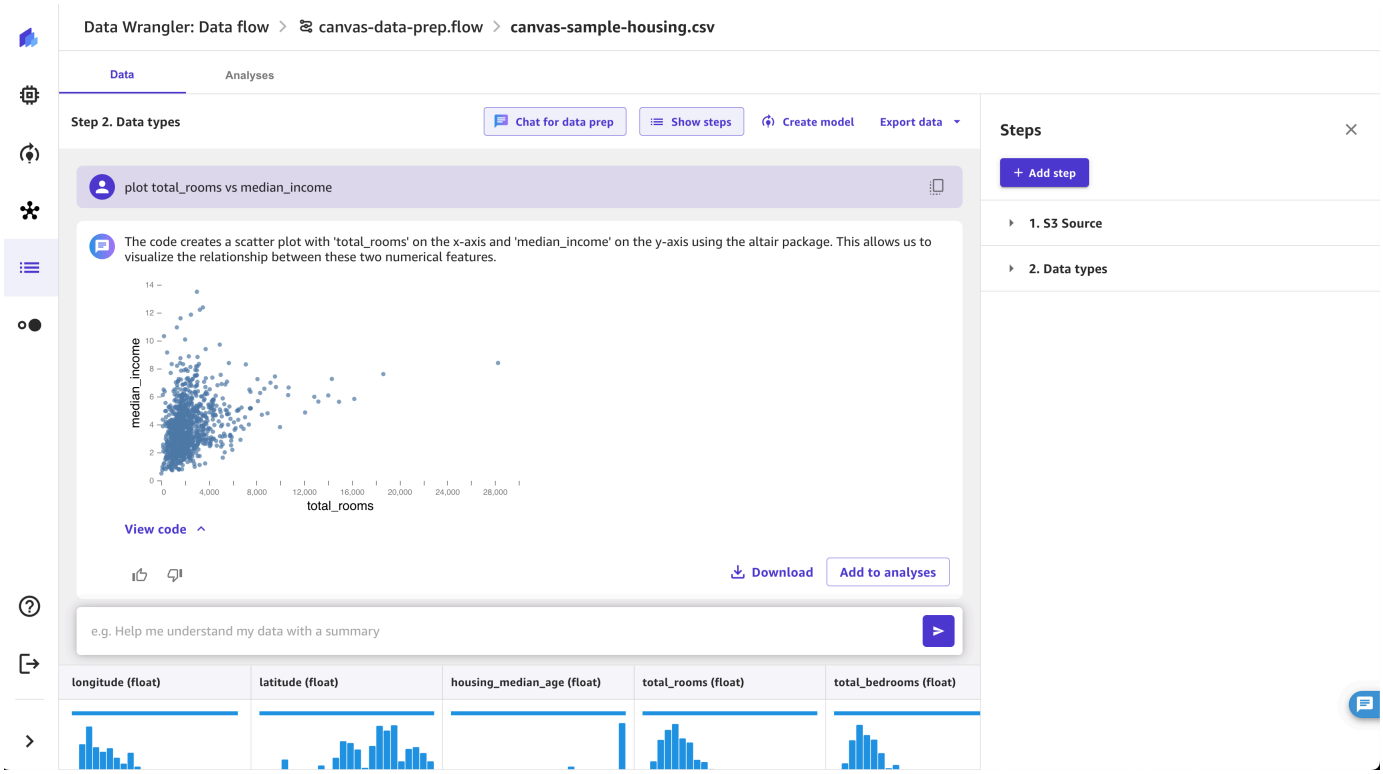
Gunakan prosedur berikut untuk mulai mengobrol dengan data Anda:

Untuk mengobrol dengan data Anda

1. Buka aliran data SageMaker Canvas.
2. Pilih gelembung pidato.

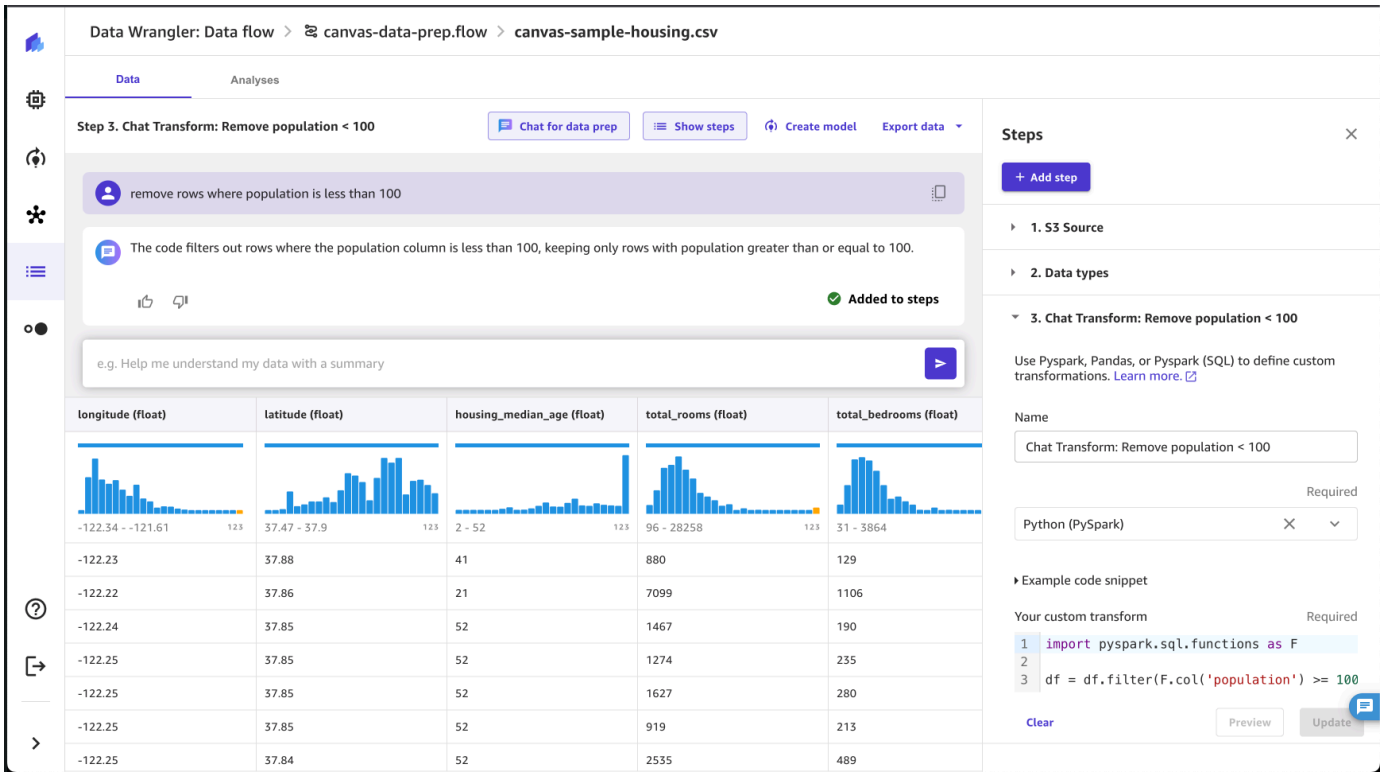


3. Tentukan prompt.
4. (Opsional) Jika analisis telah dihasilkan oleh kueri Anda, pilih Tambahkan ke analisis untuk mereferensikannya nanti.



5. (Opsional) Jika Anda telah mengubah data menggunakan prompt, lakukan hal berikut.
   a. Pilih Pratinjau untuk melihat hasilnya.
   b. (Opsional) Ubah kode dalam transformasi dan pilih Perbarui.

c. (Opsional) Jika Anda puas dengan hasil transformasi, pilih Tambahkan ke langkah untuk menambahkannya ke panel langkah di navigasi sebelah kanan.



Setelah menyiapkan data menggunakan bahasa alami, Anda dapat membuat model menggunakan data yang diubah. Untuk informasi selengkapnya tentang membuat model, lihatMembangun model kustom.

Data proses

Anda dapat memproses atau mengekspor data Anda ke lokasi yang sesuai dengan alur kerja pembelajaran mesin Anda. Misalnya, Anda dapat mengekspor data yang diubah sebagai dataset SageMaker Canvas dan membuat model pembelajaran mesin darinya.

Setelah mengekspor data, Anda dapat memilih Buat model untuk membuat model pembelajaran mesin dari data Anda. Untuk informasi selengkapnya tentang membuat model, lihatMembangun model kustom.

> **Important**
>
> Perhatikan bahwa kumpulan data SageMaker Canvas memiliki batas 5 GB. Jika Anda memproses lebih dari 5 GB data, Anda dapat menggunakan transformasi sampling untuk mengurangi ukuran kumpulan data sebelum Anda mengekspornya. Atau, Anda dapat mengekspor data ke Amazon S3. Untuk informasi selengkapnya tentang mengimpor kumpulan data, lihat. Buat kumpulan data

Dalam aliran Data Wrangler (aliran data), di SageMaker Canvas hanya menerapkan transformasi ke dataset Canvas. SageMaker Untuk memproses semua data yang telah diimpor, lihat Ekspor data menggunakan pekerjaan pemrosesan Anda dapat menggunakan salah satu metode berikut untuk memproses data Anda:

- Mengekspor data — Mengekspor dataset dengan transformasi yang telah Anda buat.
- Mengekspor aliran data Anda — Mengekspor aliran data Anda menggunakan notebook Jupyter. Anda dapat memodifikasi kode dan menggunakannya dalam alur kerja pembelajaran mesin Anda.
- Menjalankan pekerjaan pemrosesan dari aliran data Anda — Mengekspor kumpulan data dengan transformasi yang telah Anda buat dengan cara yang sangat berkinerja tinggi. Kami merekomendasikan menjalankan pekerjaan pemrosesan pada kumpulan data yang lebih besar dari 5 GB.

Ekspor data

Ekspor data Anda untuk membuat kumpulan data dengan transformasi dari alur yang diterapkan padanya. Anda dapat mengekspor node apa pun dalam aliran data Anda ke lokasi berikut:

- SageMaker Dataset kanvas
- Amazon S3

Untuk Amazon S3, Anda dapat mengekspor data Anda sebagai salah satu jenis file berikut:

- CSV
- Parquet

Anda dapat mengekspor dataset Anda sebagai dataset SageMaker Canvas untuk membuat model menggunakan data yang diubah. Gunakan prosedur berikut untuk mengekspor dataset SageMaker Canvas dari node dalam aliran data Anda.

Untuk mengekspor node dalam alur Anda sebagai dataset SageMaker Canvas

1. Arahkan ke aliran data Anda.
2. Pilih + di sebelah node yang Anda ekspor.
3. Pilih Ekspor data.
4. Pilih Dataset Kanvas.
5. Pilih Ekspor.

Ekspor kumpulan data Anda ke Amazon S3 untuk menggunakan data yang diubah dalam alur kerja pembelajaran mesin di luar Canvas. SageMaker

Untuk mengekspor dataset SageMaker Canvas ke Amazon S3

1. Arahkan ke aliran data Anda.
2. Pilih + di sebelah node yang Anda ekspor.
3. Pilih Ekspor data.
4. Pilih Amazon S3.
5. Tentukan nilai untuk bidang berikut:

    - Lokasi Amazon S3 — lokasi S3 tempat Anda mengekspor file.
    - Jenis file — Format file yang Anda ekspor.
    - Delimiter — nilai yang digunakan untuk memisahkan nilai dalam file.
    - Kompresi (Opsional) — Metode kompresi yang digunakan untuk mengurangi ukuran file. Anda dapat menggunakan metode kompresi berikut:
        - Tidak ada
        - bzip2
        - mengempiskan
        - gzip
    - KMS Key ID atau ARN (Opsional) - ARN atau ID kunci. AWS KMS Kunci KMS adalah kunci kriptografi. Anda dapat menggunakan kunci untuk mengenkripsi data output dari pekerjaan. Untuk informasi selengkapnya tentang kunci KMS, lihat AWS Key Management Service.

6. Pilih Ekspor.

Ekspor data menggunakan pekerjaan pemrosesan

Dataset SageMaker Canvas adalah contoh data Anda. Gunakan pekerjaan SageMaker pemrosesan Amazon untuk menerapkan transformasi dalam aliran data Anda ke semua data Anda.

Untuk membuat pekerjaan SageMaker pemrosesan, lakukan hal berikut:

1. Buat simpul tujuan
2. Buat pekerjaan

Sebuah node tujuan memberitahu SageMaker Canvas di mana untuk menyimpan data yang diproses. Anda membuat pekerjaan pemrosesan untuk menampilkan data yang diubah ke lokasi yang ditentukan oleh node tujuan. Membuat pekerjaan pemrosesan menjalankan sumber daya komputasi yang diperlukan untuk menampilkan data yang diubah ke Amazon S3.

Anda dapat menggunakan node tujuan untuk mengekspor beberapa transformasi atau semua transformasi yang telah Anda buat dalam alur Data Wrangler Anda.

Anda dapat menggunakan beberapa node tujuan untuk mengekspor transformasi atau set transformasi yang berbeda. Contoh berikut menunjukkan dua node tujuan dalam aliran Data Wrangler tunggal.

Gunakan prosedur berikut untuk membuat node tujuan.

Untuk membuat node tujuan

1. Pilih + di sebelah node yang mewakili transformasi yang ingin Anda ekspor.
2. Pilih Tambahkan tujuan.
3. Pilih Amazon S3.
4. Tentukan bidang berikut.
    - Nama Dataset — Nama yang Anda tentukan untuk dataset yang Anda ekspor.
    - Jenis file — Format file yang Anda ekspor.
    - Delimiter — Nilai yang digunakan untuk memisahkan nilai-nilai lain.
    - Kompresi (file CSV dan Parket saja) — Metode kompresi yang digunakan untuk mengurangi ukuran file. Anda dapat menggunakan metode kompresi berikut:

    - bzip2
    - mengempiskan
    - gzip
  - Lokasi Amazon S3 — Lokasi S3 yang Anda gunakan untuk menampilkan file.
  - (Opsional) Jumlah partisi — Jumlah kumpulan data yang Anda tulis sebagai output dari pekerjaan pemrosesan.
  - (Opsional) Partisi demi kolom - Menulis semua data dengan nilai unik yang sama dari kolom.
5. Pilih Tambahkan tujuan.

Anda dapat membuat beberapa node tujuan dalam aliran data yang sama. Saat Anda membuat pekerjaan pemrosesan, secara bersamaan dapat melakukan serangkaian transformasi yang berbeda pada data Anda dan menyimpannya ke lokasi Amazon S3 yang berbeda.

Aliran data ekspor

Mengekspor aliran data Anda menerjemahkan operasi yang telah Anda buat di Data Wrangler dan mengekspornya ke notebook Jupyter yang dapat Anda modifikasi dan jalankan.

Aliran data adalah serangkaian langkah persiapan data yang telah Anda lakukan pada data Anda. Dalam persiapan data Anda, Anda melakukan satu atau lebih transformasi ke data Anda. Setiap transformasi dilakukan dengan menggunakan langkah transformasi. Aliran memiliki serangkaian node yang mewakili impor data Anda dan transformasi yang telah Anda lakukan. Untuk contoh node, lihat gambar berikut.

Sebagai alternatif untuk menggunakan node tujuan, Anda dapat menggunakan opsi Ekspor aliran data untuk mengekspor aliran Data Wrangler Anda ke Amazon S3 menggunakan notebook Jupyter. Anda dapat mengintegrasikan kode output ke dalam pipeline pembelajaran mesin Anda. Anda dapat memilih node data apa pun dalam aliran data Anda dan mengekspornya. Mengekspor node data mengekspor transformasi yang diwakili oleh node dan transformasi yang mendahuluinya.

Gunakan prosedur berikut untuk membuat notebook Jupyter dan menjalankannya untuk mengekspor aliran data Anda ke Amazon S3.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor aliran data.
3. Pilih salah satu cara berikut:

- Simpan ke Amazon S3 (melalui notebook Jupyter).
- Amazon Personalisasi.

4. Pilih salah satu dari berikut ini:
   - Unduh salinan lokal
   - Ekspor ke lokasi S3
5. Jika Anda mengekspor ke Amazon S3, tentukan lokasi S3 tempat Anda mengekspor notebook.
6. Pilih Ekspor.

Buat jadwal untuk memproses data baru secara otomatis

Jika Anda memproses data secara berkala, Anda dapat membuat jadwal untuk menjalankan pekerjaan pemrosesan secara otomatis. Misalnya, Anda dapat membuat jadwal yang menjalankan pekerjaan pemrosesan secara otomatis saat Anda mendapatkan data baru. Untuk informasi selengkapnya tentang memproses pekerjaan, lihatEkspor data menggunakan pekerjaan pemrosesan.

Saat Anda membuat pekerjaan, Anda harus menentukan peran IAM yang memiliki izin untuk membuat pekerjaan. Anda dapat menggunakan kebijakan AmazonSageMakerCanvasDataPrepFullAccess untuk menambahkan izin.

Tambahkan kebijakan kepercayaan berikut ke peran untuk memungkinkan untuk EventBridge mengasumsikannya.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
```

> **Important**
>
> Saat Anda membuat jadwal, Data Wrangler membuat `eventRule` in. EventBridge Anda dikenakan biaya untuk aturan acara yang Anda buat dan instance yang digunakan untuk menjalankan pekerjaan pemrosesan.
> Untuk informasi tentang EventBridge harga, lihat EventBridge harga Amazon. Untuk informasi tentang memproses harga lowongan kerja, lihat SageMaker Harga Amazon.

Anda dapat mengatur jadwal menggunakan salah satu metode berikut:

- Ekspresi CRON

  > **Note**
  >
  > Data Wrangler tidak mendukung ekspresi berikut:
  >
  > - LW#
  > - Singkatan untuk hari
  > - Singkatan untuk bulan

- Ekspresi RATE
- Berulang — Tetapkan interval per jam atau harian untuk menjalankan pekerjaan.
- Waktu spesifik - Tetapkan hari dan waktu tertentu untuk menjalankan pekerjaan.

Bagian berikut menyediakan prosedur untuk menciptakan lapangan kerja.

CRON

Gunakan prosedur berikut untuk membuat jadwal dengan ekspresi CRON.

Untuk menentukan jadwal dengan ekspresi CRON, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya.

5. Pilih Jadwal Rekanan.
6. Pilih Buat jadwal baru.
7. Untuk Nama Jadwal, tentukan nama jadwal.
8. Untuk Run Frequency, pilih CRON.
9. Tentukan ekspresi CRON yang valid.
10. Pilih Buat.
11. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

> **Note**
>
> Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

12. Pilih salah satu cara berikut:
    - Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
    - Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.
13. Pilih Jalankan

RATE

Gunakan prosedur berikut untuk membuat jadwal dengan ekspresi RATE.

Untuk menentukan jadwal dengan ekspresi RATE, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya, 2. Konfigurasikan pekerjaan.
5. Pilih Jadwal Rekanan.
6. Pilih Buat jadwal baru.
7. Untuk Nama Jadwal, tentukan nama jadwal.

8. Untuk Run Frequency, pilih Rate.
9. Untuk Nilai, tentukan bilangan bulat.
10. Untuk Unit, pilih salah satu dari berikut ini:
    - Menit
    - Jam
    - Hari
11. Pilih Buat.
12. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

    > **Note**
    >
    > Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

13. Pilih salah satu cara berikut:
    - Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
    - Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.
14. Pilih Jalankan

Recurring

Gunakan prosedur berikut untuk membuat jadwal yang menjalankan pekerjaan secara berulang.

Untuk menentukan jadwal dengan ekspresi CRON, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya, 2. Konfigurasikan pekerjaan.
5. Pilih Jadwal Rekanan.
6. Pilih Buat jadwal baru.

7. Untuk Nama Jadwal, tentukan nama jadwal.
8. Untuk Run Frequency, pastikan Recurring dipilih secara default.
9. Untuk Setiap x jam, tentukan frekuensi per jam yang dijalankan pekerjaan pada siang hari. Nilai yang valid adalah bilangan bulat dalam rentang inklusif dan**1**. **23**
10. Untuk Pada hari, pilih salah satu opsi berikut:

    - Setiap hari
    - Akhir pekan
    - Hari kerja
    - Pilih Hari

    - (Opsional) Jika Anda telah memilih Pilih Hari, pilih hari dalam seminggu untuk menjalankan pekerjaan.

    > **Note**
    >
    > Jadwal diatur ulang setiap hari. Jika Anda menjadwalkan pekerjaan untuk dijalankan setiap lima jam, itu berjalan pada waktu-waktu berikut di siang hari:
    >
    > - 00:00
    > - 05:00
    > - 10:00
    > - 15:00
    > - 20:00

11. Pilih Buat.
12. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

    > **Note**
    >
    > Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

13. Pilih salah satu cara berikut:

- Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
- Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.

14. Pilih Jalankan

Specific time

Gunakan prosedur berikut untuk membuat jadwal yang menjalankan pekerjaan pada waktu tertentu.

Untuk menentukan jadwal dengan ekspresi CRON, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya, 2. Konfigurasikan pekerjaan.
5. Pilih Jadwal Rekanan.
6. Pilih Buat jadwal baru.
7. Untuk Nama Jadwal, tentukan nama jadwal.
8. Pilih Buat.
9. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

> **Note**
>
> Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

10. Pilih salah satu cara berikut:

- Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
- Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.

11. Pilih Jalankan

Anda dapat menggunakan SageMaker AWS Management Console untuk melihat pekerjaan yang dijadwalkan untuk dijalankan. Pekerjaan pemrosesan Anda berjalan di dalam SageMaker Pipelines. Setiap pekerjaan pemrosesan memiliki pipa sendiri. Ini berjalan sebagai langkah pemrosesan di dalam pipa. Anda dapat melihat jadwal yang telah Anda buat dalam pipeline. Untuk informasi tentang melihat pipeline, lihatMembuat Alur.

Gunakan prosedur berikut untuk melihat pekerjaan yang telah Anda jadwalkan.

Untuk melihat pekerjaan yang telah Anda jadwalkan, lakukan hal berikut.

1. Buka Amazon SageMaker Studio Classic.
2. Buka SageMaker Pipa
3. Lihat saluran pipa untuk pekerjaan yang telah Anda buat.

   Pipeline yang menjalankan pekerjaan menggunakan nama pekerjaan sebagai awalan. Misalnya, jika Anda telah membuat pekerjaan bernama`housing-data-feature-enginnering`, nama pipeline adalah`canvas-data-prep-housing-data-feature-engineering`.
4. Pilih pipeline yang berisi pekerjaan Anda.
5. Lihat status jaringan pipa. Pipelines dengan Status Sukses telah menjalankan pekerjaan pemrosesan dengan sukses.

Untuk menghentikan pekerjaan pemrosesan berjalan, lakukan hal berikut:

Untuk menghentikan pekerjaan pemrosesan agar tidak berjalan, hapus aturan acara yang menentukan jadwal. Menghapus aturan acara menghentikan semua pekerjaan yang terkait dengan jadwal berjalan. Untuk informasi tentang menghapus aturan, lihat Menonaktifkan atau menghapus aturan Amazon. EventBridge

Anda dapat menghentikan dan menghapus saluran pipa yang terkait dengan jadwal juga. Untuk informasi tentang menghentikan pipa, lihat StopPipelineExecution. Untuk informasi tentang menghapus pipeline, lihat DeletePipeline.

Reparasi berubah ke seluruh kumpulan data dan mengekspornya

Saat Anda mengimpor data, Data Wrangler menggunakan sampel data untuk menerapkan pengkodean. Data Wrangler menggunakan 20.000 baris pertama sebagai sampel.

Transformasi berikut dapat menggunakan data Anda untuk membuat kolom dalam kumpulan data:

- Encode Kategoris
- Featurize Teks
- Tangani Outlier
- Tangani Nilai yang Hilang

Jika Anda menggunakan sampling untuk mengimpor data Anda, transformasi sebelumnya hanya menggunakan data dari sampel untuk membuat kolom. Transformasi mungkin tidak menggunakan semua data yang relevan. Misalnya, jika Anda menggunakan transformasi Encode Categorical, mungkin ada kategori di seluruh kumpulan data yang tidak ada dalam sampel.

Anda dapat menggunakan node tujuan atau notebook Jupyter untuk mereparasi transformasi ke seluruh kumpulan data. Ketika Data Wrangler mengekspor transformasi dalam aliran, itu menciptakan pekerjaan pemrosesan. SageMaker Saat pekerjaan pemrosesan selesai, Data Wrangler menyimpan file berikut di lokasi Amazon S3 default atau lokasi S3 yang Anda tentukan:

- File aliran Data Wrangler yang menentukan transformasi yang direparasi ke kumpulan data
- Dataset dengan transformasi reparasi diterapkan padanya

Anda dapat membuka file aliran Data Wrangler dalam SageMaker Canvas dan menerapkan transformasi ke kumpulan data yang berbeda. Misalnya, jika Anda telah menerapkan transformasi ke kumpulan data pelatihan, Anda dapat membuka dan menggunakan file aliran Data Wrangler untuk menerapkan transformasi ke kumpulan data yang digunakan untuk inferensi.

Gunakan prosedur berikut untuk menjalankan notebook Jupyter untuk mereparasi transformasi dan mengekspor data.

Untuk menjalankan notebook Jupyter dan untuk mereparasi transformasi dan mengekspor alur Data Wrangler Anda, lakukan hal berikut.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor ke.
3. Pilih lokasi tempat Anda mengekspor data.
4. Untuk `refit_trained_params` objek, atur `refit` ke`True`.
5. Untuk `output_flow` bidang, tentukan nama file aliran output dengan transformasi reparasi.
6. Jalankan notebook Jupyter.

Otomatiskan persiapan data di Canvas SageMaker

Setelah mengubah data dalam aliran data, Anda dapat mengekspor transformasi ke alur kerja pembelajaran mesin. Saat Anda mengekspor transformasi Anda, SageMaker Canvas membuat notebook Jupyter. Anda harus menjalankan notebook dalam Amazon SageMaker Studio Classic. Untuk informasi tentang memulai Studio Classic, hubungi administrator Anda.

Otomatiskan persiapan data menggunakan Pipelines SageMaker

Saat ingin membangun dan menerapkan alur kerja machine learning (ML) skala besar, Anda dapat menggunakan SageMaker Pipelines untuk membuat alur kerja yang mengelola dan menerapkan pekerjaan. SageMaker Dengan SageMaker Pipelines, Anda dapat membangun alur kerja yang mengelola persiapan SageMaker data, pelatihan model, dan memodelkan pekerjaan penerapan. Anda dapat menggunakan algoritma pihak pertama yang SageMaker menawarkan dengan menggunakan SageMaker Pipelines. Untuk informasi lebih lanjut tentang SageMaker Pipelines, lihat SageMaker Pipelines.

Saat Anda mengekspor satu atau beberapa langkah dari aliran data ke SageMaker Pipelines, Data Wrangler akan membuat buku catatan Jupyter yang dapat Anda gunakan untuk menentukan, membuat instance, menjalankan, dan mengelola pipeline.

Menggunakan Notebook Jupyter untuk Membuat Pipeline

Gunakan prosedur berikut untuk membuat notebook Jupyter untuk mengekspor aliran Data Wrangler Anda ke Pipelines. SageMaker

Gunakan prosedur berikut untuk membuat notebook Jupyter dan menjalankannya untuk mengekspor aliran Data Wrangler Anda ke Pipelines. SageMaker

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor aliran data.
3. Pilih SageMaker Pipelines (melalui Jupyter Notebook).
4. Unduh notebook Jupyter atau salin ke lokasi Amazon S3. Kami merekomendasikan untuk menyalinnya ke lokasi Amazon S3 yang dapat Anda akses dalam Studio Classic. Hubungi administrator Anda jika Anda memerlukan panduan tentang lokasi yang sesuai.
5. Jalankan notebook Jupyter.

Anda dapat menggunakan notebook Jupyter yang dihasilkan Data Wrangler untuk menentukan pipeline. Pipeline mencakup langkah-langkah pemrosesan data yang ditentukan oleh alur Data Wrangler Anda.

Anda dapat menambahkan langkah tambahan ke pipeline dengan menambahkan langkah-langkah ke `steps` daftar dalam kode berikut di buku catatan:

```
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[instance_type, instance_count],
    steps=[step_process], #Add more steps to this list to run in your Pipeline
)
```

Untuk informasi selengkapnya tentang mendefinisikan pipeline, lihat Mendefinisikan SageMaker Pipeline.

Mengotomatiskan persiapan data menggunakan titik akhir inferensi

Gunakan alur Data Wrangler Anda untuk memproses data pada saat inferensi dengan membuat pipeline inferensi SageMaker serial dari alur Data Wrangler Anda. Pipa inferensi adalah serangkaian langkah yang menghasilkan model terlatih yang membuat prediksi pada data baru. Pipa inferensi serial dalam Data Wrangler mengubah data mentah dan menyediakannya ke model pembelajaran mesin untuk prediksi. Anda membuat, menjalankan, dan mengelola pipeline inferensi dari notebook Jupyter dalam Studio Classic. Untuk informasi selengkapnya tentang mengakses buku catatan, lihatGunakan buku catatan Jupyter untuk membuat titik akhir inferensi.

Di dalam buku catatan, Anda dapat melatih model pembelajaran mesin atau menentukan model yang sudah Anda latih. Anda dapat menggunakan Amazon SageMaker Autopilot atau XGBoost untuk melatih model menggunakan data yang telah Anda ubah dalam alur Data Wrangler Anda.

Pipeline menyediakan kemampuan untuk melakukan inferensi batch atau real-time. Anda juga dapat menambahkan aliran Data Wrangler ke SageMaker Model Registry. Untuk informasi selengkapnya tentang model hosting, lihatHost beberapa model dalam satu wadah di belakang satu titik akhir.

> **Important**
>
> Anda tidak dapat mengekspor aliran Data Wrangler ke titik akhir inferensi jika memiliki transformasi berikut:
>
> - Join

- Gabungan
- Grup oleh

Jika Anda harus menggunakan transformasi sebelumnya untuk menyiapkan data Anda, gunakan prosedur berikut.

Untuk mempersiapkan data Anda untuk inferensi dengan transformasi yang tidak didukung

1. Buat alur Data Wrangler.
2. Terapkan transformasi sebelumnya yang tidak didukung.
3. Ekspor data ke bucket Amazon S3.
4. Buat alur Data Wrangler terpisah.
5. Impor data yang telah Anda ekspor dari alur sebelumnya.
6. Terapkan transformasi yang tersisa.
7. Buat pipeline inferensi serial menggunakan notebook Jupyter yang kami sediakan.

Untuk informasi tentang mengekspor data ke bucket Amazon S3, lihat. Ekspor data Untuk informasi tentang membuka notebook Jupyter yang digunakan untuk membuat pipeline inferensi serial, lihat. Gunakan buku catatan Jupyter untuk membuat titik akhir inferensi

Data Wrangler mengabaikan transformasi yang menghapus data pada saat inferensi. Misalnya, Data Wrangler mengabaikan Tangani Nilai yang Hilang transformasi jika Anda menggunakan konfigurasi Drop missing.

Jika Anda telah mereparasi transformasi ke seluruh kumpulan data Anda, transformasi terbawa ke saluran inferensi Anda. Misalnya, jika Anda menggunakan nilai median untuk mengimputasi nilai yang hilang, nilai median dari refitting transformasi diterapkan ke permintaan inferensi Anda. Anda dapat mereparasi transformasi dari alur Data Wrangler saat menggunakan notebook Jupyter atau saat mengekspor data ke pipeline inferensi. Untuk informasi tentang memperbaiki transformasi, lihat. Reparasi berubah ke seluruh kumpulan data dan mengekspornya

Pipa inferensi serial mendukung tipe data berikut untuk string input dan output. Setiap tipe data memiliki seperangkat persyaratan.

Tipe data yang didukung

- `text/csv`— tipe data untuk string CSV
  - String tidak dapat memiliki header.
  - Fitur yang digunakan untuk pipa inferensi harus dalam urutan yang sama dengan fitur dalam kumpulan data pelatihan.
  - Harus ada pembatas koma di antara fitur.
  - Catatan harus dibatasi oleh karakter baris baru.

  Berikut ini adalah contoh string CSV yang diformat secara valid yang dapat Anda berikan dalam permintaan inferensi.

```
abc,0.0,"Doe, John",12345\ndef,1.1,"Doe, Jane",67890
```

- `application/json`— tipe data untuk string JSON
  - Fitur yang digunakan dalam kumpulan data untuk pipa inferensi harus dalam urutan yang sama dengan fitur dalam kumpulan data pelatihan.
  - Data harus memiliki skema tertentu. Anda mendefinisikan skema sebagai `instances` objek tunggal yang memiliki satu set. `features` Setiap `features` objek mewakili pengamatan.

  Berikut ini adalah contoh string JSON yang diformat secara valid yang dapat Anda berikan dalam permintaan inferensi.

```
{
    "instances": [
        {
            "features": ["abc", 0.0, "Doe, John", 12345]
        },
        {
            "features": ["def", 1.1, "Doe, Jane", 67890]
        }
    ]
}
```

Gunakan buku catatan Jupyter untuk membuat titik akhir inferensi

Gunakan prosedur berikut untuk mengekspor alur Data Wrangler Anda untuk membuat pipeline inferensi.

Untuk membuat pipeline inferensi menggunakan notebook Jupyter, lakukan hal berikut.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor aliran data.
3. Pilih SageMaker Inference Pipeline (melalui Jupyter Notebook).
4. Unduh notebook Jupyter atau salin ke lokasi Amazon S3. Kami merekomendasikan untuk menyalinnya ke lokasi Amazon S3 yang dapat Anda akses dalam Studio Classic. Hubungi administrator Anda jika Anda memerlukan panduan tentang lokasi yang sesuai.
5. Jalankan notebook Jupyter.

Saat Anda menjalankan notebook Jupyter, itu menciptakan artefak aliran inferensi. Artefak aliran inferensi adalah file aliran Data Wrangler dengan metadata tambahan yang digunakan untuk membuat pipeline inferensi serial. Node yang Anda ekspor mencakup semua transformasi dari node sebelumnya.

> **Important**
>
> Data Wrangler membutuhkan artefak aliran inferensi untuk menjalankan pipa inferensi. Anda tidak dapat menggunakan file aliran Anda sendiri sebagai artefak. Anda harus membuatnya dengan menggunakan prosedur sebelumnya.

Mengotomatiskan persiapan data menggunakan Kode Python

Untuk mengekspor semua langkah dalam aliran data Anda ke file Python yang dapat Anda integrasikan secara manual ke dalam alur kerja pemrosesan data apa pun, gunakan prosedur berikut.

Gunakan prosedur berikut untuk menghasilkan notebook Jupyter dan menjalankannya untuk mengekspor aliran Data Wrangler Anda ke kode Python.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor aliran data.

3. Pilih Kode Python.
4. Unduh notebook Jupyter atau salin ke lokasi Amazon S3. Kami merekomendasikan untuk menyalinnya ke lokasi Amazon S3 yang dapat Anda akses dalam Studio Classic. Hubungi administrator Anda jika Anda memerlukan panduan tentang lokasi yang sesuai.
5. Jalankan notebook Jupyter.

Anda mungkin perlu mengonfigurasi skrip Python untuk membuatnya berjalan di pipeline Anda. Misalnya, jika Anda menjalankan lingkungan Spark, pastikan Anda menjalankan skrip dari lingkungan yang memiliki izin untuk mengakses AWS sumber daya.

Gunakan AI generatif dengan model foundation

Amazon SageMaker Canvas menyediakan model dasar AI generatif yang dapat Anda gunakan untuk memulai obrolan percakapan. Model pembuatan konten ini dilatih pada sejumlah besar data teks untuk mempelajari pola statistik dan hubungan antar kata, dan mereka dapat menghasilkan teks yang koheren yang secara statistik mirip dengan teks tempat mereka dilatih. Anda dapat menggunakan kemampuan ini untuk meningkatkan produktivitas Anda dengan melakukan hal berikut:

- Menghasilkan konten, seperti garis besar dokumen, laporan, dan blog
- Meringkas teks dari kumpulan teks besar, seperti transkrip panggilan pendapatan, laporan tahunan, atau bab-babnya manual pengguna
- Ekstrak wawasan dan takeaways kunci dari bagian teks yang besar, seperti catatan rapat atau narasi
- Memperbaiki teks dan menangkap kesalahan tata bahasa atau kesalahan ketik

Model dasar adalah kombinasi dari Amazon SageMaker JumpStart dan Amazon Bedrock Large Language Models (LLM). Canvas menawarkan model-model berikut:

Model	Tipe	Deskripsi
Instruksi Falcon-7B	SageMaker JumpStart model	Falcon-7B-Instruct memiliki 7 miliar parameter dan disetel dengan baik pada campuran kumpulan data obrolan dan instruksi. Ini cocok sebagai asisten virtual dan berkinerj

Model	Tipe	Deskripsi
		a terbaik saat mengikuti instruksi atau terlibat dalam percakapan. Karena model ini dilatih pada sejumlah besar data web berbahasa Inggris, ia membawa stereotip dan bias yang biasa ditemukan online dan tidak cocok untuk bahasa selain bahasa Inggris. Dibandingkan dengan Falcon-40B-Instruc t, Falcon-7B-Instruct adalah model yang sedikit lebih kecil dan lebih kompak.

Model	Tipe	Deskripsi
Instruksi Falcon-40B	SageMaker JumpStart model	Falcon-40B-Instruct memiliki 40 miliar parameter dan disetel dengan baik pada campuran kumpulan data obrolan dan instruksi. Ini cocok sebagai asisten virtual dan berkinerja terbaik saat mengikuti instruksi atau terlibat dalam percakapan. Karena model ini dilatih pada sejumlah besar data web berbahasa Inggris, ia membawa stereotip dan bias yang biasa ditemukan online dan tidak cocok untuk bahasa selain bahasa Inggris. Dibandingkan dengan Falcon-7B-Instruct, Falcon-40B-Instruct adalah model yang sedikit lebih besar dan lebih kuat.

Model	Tipe	Deskripsi
Instruksi MPT-7B	SageMaker JumpStart model	MPT-7B-Instruct adalah model untuk tugas-tugas berikut instruksi bentuk panjang dan dapat membantu Anda dengan tugas menulis termasuk ringkasan teks dan menjawab pertanyaan untuk menghemat waktu dan tenaga Anda. Model ini dilatih pada sejumlah besar data yang disetel dengan baik dan dapat menangani input yang lebih besar, seperti dokumen yang kompleks. Gunakan model ini saat Anda ingin memproses teks dalam jumlah besar atau ingin model menghasilkan respons yang panjang.

Model	Tipe	Deskripsi
Amazon Titan	Model Amazon Bedrock	Amazon Titan adalah model bahasa tujuan umum yang kuat yang dapat Anda gunakan untuk tugas-tugas seperti meringkas, pembuatan teks (seperti membuat posting blog), klasifikasi, tanya jawab terbuka, dan ekstraksi informasi. Ini dilatih sebelumnya pada kumpulan data besar, sehingga cocok untuk tugas dan penalaran yang kompleks. Untuk terus mendukung praktik terbaik dalam penggunaan AI yang bertanggung jawab, model yayasan Amazon Titan dibuat untuk mendeteks i dan menghapus konten berbahaya dalam data, menolak konten yang tidak pantas dalam input pengguna, dan memfilter output model yang berisi konten yang tidak pantas (seperti ujaran kebencian, kata-kata kotor, dan kekerasan).

Model	Tipe	Deskripsi
Jurassic-2 Pertengahan	Model Amazon Bedrock	Jurassic-2 Mid adalah model pembuatan teks berkinerja tinggi yang dilatih pada kumpulan teks yang sangat besar (saat ini hingga pertengahan 2022). Ini sangat serbaguna, tujuan umum, dan mampu menyusun teks seperti manusia dan menyelesaikan tugas-tugas kompleks seperti menjawab pertanyaan, klasifikasi teks, dan banyak lainnya. Model ini menawarkan kemampuan instruksi zero-shot, memungkinkannya diarahkan hanya dengan bahasa alami dan tanpa menggunakan contoh. Ini bekerja hingga 30% lebih cepat dari pendahulunya, model Jurassic-1. Jurassic-2 Mid adalah model berukuran sedang AI21, dirancang dengan cermat untuk mencapai keseimbangan yang tepat antara kualitas luar biasa dan keterjangkauan.

Model	Tipe	Deskripsi
Jurassic-2 Ultra	Model Amazon Bedrock	Jurassic-2 Ultra adalah model pembuatan teks berkinerja tinggi yang dilatih pada kumpulan teks yang sangat besar (saat ini hingga pertengahan 2022). Ini sangat serbaguna, tujuan umum, dan mampu menyusun teks seperti manusia dan menyelesa ikan tugas-tugas kompleks seperti menjawab pertanyaa n, klasifikasi teks, dan banyak lainnya. Model ini menawarka n kemampuan instruksi zero-shot, memungkinkannya diarahkan hanya dengan bahasa alami dan tanpa menggunakan contoh. Ini bekerja hingga 30% lebih cepat dari pendahulunya, model Jurassic-1. Dibandingkan dengan Jurassic-2 Mid, Jurassic-2 Ultra adalah model yang sedikit lebih besar dan lebih kuat.

Model	Tipe	Deskripsi
Anthropic Claude Instant	Model Amazon Bedrock	Claude Instant Anthropic adalah model yang lebih cepat dan lebih hemat biaya namun masih sangat mumpuni. Model ini dapat menangani berbagai tugas termasuk dialog biasa, analisis teks, ringkasan, dan menjawab pertanyaan dokumen. Sama seperti Claude-2, Claude Instant dapat mendukung hingga 100.000 token di setiap prompt, setara dengan sekitar 200 halaman informasi.

Model	Tipe	Deskripsi
Anthropic Claude-2	Model Amazon Bedrock	Claude-2 adalah model Anthropic yang paling kuat, yang unggul dalam berbagai tugas mulai dari dialog canggih dan pembuatan konten kreatif hingga instruksi terperinci berikut. Claude-2 dapat mengambil hingga 100.000 token di setiap prompt, setara dengan sekitar 200 halaman informasi. Ini dapat menghasilkan respons yang lebih lama dibandingkan dengan versi sebelumnya. Ini mendukung kasus penggunaan seperti menjawab pertanyaan, ekstraksi informasi, menghapus PII, pembuatan konten, klasifikasi pilihan ganda, permainan peran, membandingkan teks, ringkasan, dan Tanya Jawab dokumen dengan kutipan.

Model dasar dari Amazon Bedrock saat ini hanya tersedia di Wilayah AS Timur (Virginia N.) dan AS Barat (Oregon). Selain itu, saat menggunakan model foundation dari Amazon Bedrock, Anda dikenakan biaya berdasarkan volume token input dan token output, seperti yang ditentukan oleh masing-masing penyedia model. Untuk informasi selengkapnya, lihat halaman harga Amazon Bedrock. Model SageMaker JumpStart dasar diterapkan pada instans SageMaker Hosting, dan Anda dikenakan biaya selama durasi penggunaan berdasarkan jenis instans yang digunakan. Untuk informasi selengkapnya tentang biaya berbagai jenis instans, lihat bagian Amazon SageMaker Hosting: Inferensi Waktu Nyata di halaman SageMaker harga.

Kueri dokumen adalah fitur tambahan yang dapat Anda gunakan untuk kueri dan mendapatkan wawasan dari dokumen yang disimpan dalam indeks menggunakan Amazon Kendra. Dengan fungsi ini, Anda dapat menghasilkan konten dari konteks dokumen-dokumen tersebut dan menerima tanggapan yang spesifik untuk kasus penggunaan bisnis Anda, sebagai lawan dari tanggapan yang umum terhadap sejumlah besar data di mana model yayasan dilatih. Untuk informasi selengkapnya tentang indeks di Amazon Kendra, lihat Panduan Pengembang Amazon Kendra.

Jika Anda ingin mendapatkan tanggapan dari salah satu model foundation yang disesuaikan dengan data dan kasus penggunaan Anda, Anda dapat menyempurnakan model foundation. Untuk mempelajari selengkapnya, lihat Model pondasi yang menyempurnakan.

Untuk memulai, lihat bagian berikut.

Prasyarat

Bagian berikut menguraikan prasyarat untuk berinteraksi dengan model dasar dan menggunakan fitur kueri dokumen di Canvas. Sisa konten di halaman ini mengasumsikan bahwa Anda telah memenuhi prasyarat untuk model pondasi. Fitur kueri dokumen memerlukan izin tambahan.

Prasyarat untuk model pondasi

Izin yang Anda perlukan untuk berinteraksi dengan model disertakan dalam izin eady-to-use model Canvas R. Untuk menggunakan model generatif yang didukung AI di Canvas, Anda harus mengaktifkan izin konfigurasi eady-to-use model Canvas R saat menyiapkan Domain Amazon Anda. SageMaker Untuk informasi selengkapnya, lihat Prasyarat untuk menyiapkan Amazon Canvas SageMaker . Konfigurasi eady-to-use model Canvas R melampirkan ServicesAccess kebijakan AmazonSageMakerCanvasAI ke peran eksekusi pengguna Canvas AWS Identity and Access Management (IAM) Anda. Jika Anda mengalami masalah dengan pemberian izin, lihat topiknya. Memecahkan masalah dengan pemberian izin melalui konsol SageMaker

Jika sudah menyiapkan Domain, Anda dapat mengedit pengaturan Domain dan mengaktifkan izin. Untuk petunjuk tentang cara mengedit setelan Domain Anda, lihatLihat dan Edit Domain. Saat mengedit pengaturan untuk Domain Anda, buka pengaturan Canvas dan nyalakan opsi Aktifkan eady-to-use model R Canvas.

Model SageMaker JumpStart dasar tertentu juga mengharuskan Anda meminta peningkatan kuota SageMaker instans. Canvas menghosting model yang saat ini berinteraksi dengan Anda pada instans ini, tetapi kuota default untuk akun Anda mungkin tidak mencukupi. Jika Anda mengalami kesalahan saat menjalankan salah satu model berikut, minta peningkatan kuota untuk jenis instance terkait:

- Falcon-40B —, `ml.g5.12xlarge` `ml.g5.24xlarge`
- Falcon-13B —,, `ml.g5.2xlarge` `ml.g5.4xlarge` `ml.g5.8xlarge`
- MPT-7B-Instruksikan -,, `ml.g5.2xlarge` `ml.g5.4xlarge` `ml.g5.8xlarge`

Untuk jenis instans sebelumnya, minta peningkatan dari 0 menjadi 1 untuk kuota penggunaan titik akhir. Untuk informasi selengkapnya tentang cara meningkatkan kuota instans untuk akun Anda, lihat Meminta peningkatan kuota di Panduan Pengguna Service Quotas.

Prasyarat untuk kueri dokumen

> **Note**
>
> Kueri dokumen didukung sebagai berikutWilayah AWS: US East (Virginia N.), US East (Ohio), US West (Oregon), Eropa (Irlandia), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), dan Asia Pasifik (Mumbai).

Fitur kueri dokumen mengharuskan Anda sudah memiliki indeks Amazon Kendra yang menyimpan dokumen dan metadata dokumen Anda. Untuk informasi selengkapnya tentang Amazon Kendra, lihat Panduan Pengembang Amazon Kendra. Untuk mempelajari lebih lanjut tentang kuota untuk mengkueri indeks, lihat Kuota di Panduan Pengembang Amazon Kendra.

Anda juga harus memastikan bahwa profil pengguna Canvas Anda memiliki izin yang diperlukan untuk kueri dokumen. AmazonSageMakerCanvasFullAccessKebijakan harus dilampirkan ke peran eksekusi AWS IAM untuk SageMaker Domain yang menghosting aplikasi Canvas Anda (kebijakan ini dilampirkan secara default ke semua profil pengguna Canvas baru dan yang sudah ada). Anda juga harus secara khusus memberikan izin kueri dokumen dan menentukan akses ke satu atau beberapa indeks Amazon Kendra.

Jika administrator Canvas Anda menyiapkan Domain atau profil pengguna baru, minta mereka mengatur Domain dengan mengikuti petunjuk diPrasyarat untuk menyiapkan Amazon Canvas SageMaker . Saat menyiapkan Domain, mereka dapat mengaktifkan izin kueri dokumen melalui konfigurasi eady-to-use model Canvas R.

Administrator Canvas dapat mengelola izin kueri dokumen di tingkat profil pengguna juga. Misalnya, jika administrator ingin memberikan izin kueri dokumen ke beberapa profil pengguna tetapi menghapus izin untuk orang lain, mereka dapat mengedit izin untuk pengguna tertentu.

Prosedur berikut menunjukkan cara mengaktifkan izin kueri dokumen untuk profil pengguna tertentu:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain profil pengguna.
5. Pada halaman Detail Domain, pilih profil Pengguna yang izinnya ingin Anda edit.
6. Pada halaman Detail Pengguna, pilih Edit.
7. Di panel navigasi kiri, pilih Pengaturan kanvas.
8. Di bagian konfigurasi eady-to-use model Canvas R, aktifkan tombol Aktifkan kueri dokumen menggunakan Amazon Kendra.
9. Di dropdown, pilih satu atau beberapa indeks Amazon Kendra yang ingin Anda berikan aksesnya.
10. Pilih Kirim untuk menyimpan perubahan pada pengaturan Domain Anda.

Anda sekarang harus dapat menggunakan model dasar Canvas untuk menanyakan dokumen dalam indeks Amazon Kendra yang ditentukan.

Mulai percakapan baru untuk menghasilkan, mengekstrak, atau meringkas konten

Untuk memulai dengan model foundation AI generatif di Canvas, Anda dapat memulai sesi obrolan baru dengan salah satu model. Untuk SageMaker JumpStart model, Anda dikenakan biaya saat model aktif, jadi Anda harus memulai model saat Anda ingin menggunakannya dan mematikannya ketika Anda selesai berinteraksi. Jika Anda tidak mematikan SageMaker JumpStart model, Canvas mematikannya setelah 2 jam tidak aktif. Untuk model Amazon Bedrock (seperti Amazon Titan), Anda dikenakan biaya dengan prompt; model sudah aktif dan tidak perlu dimulai atau dimatikan. Anda dikenakan biaya langsung untuk penggunaan model ini oleh Amazon Bedrock.

Untuk membuka obrolan dengan model, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih eady-to-usemodel R.
3. Pilih Menghasilkan, mengekstrak, dan meringkas konten.
4. Pada halaman selamat datang, Anda akan menerima rekomendasi untuk memulai model default. Anda dapat memulai model yang direkomendasikan, atau Anda dapat memilih Pilih model lain dari dropdown untuk memilih yang berbeda.

5. Jika Anda memilih model SageMaker JumpStart pondasi, Anda harus memulainya sebelum tersedia untuk digunakan. Pilih Mulai model, dan kemudian model diterapkan ke sebuah SageMaker instance. Mungkin perlu beberapa menit untuk menyelesaikan ini. Ketika model siap, Anda dapat memasukkan petunjuk dan mengajukan pertanyaan model.

   Jika Anda memilih model pondasi dari Amazon Bedrock, Anda dapat mulai menggunakannya secara instan dengan memasukkan prompt dan mengajukan pertanyaan.

Tergantung pada modelnya, Anda dapat melakukan berbagai tugas. Misalnya, Anda dapat memasukkan bagian teks dan meminta model untuk meringkasnya. Atau, Anda dapat meminta model untuk membuat ringkasan singkat tentang tren pasar di domain Anda.

Respons model dalam obrolan didasarkan pada konteks permintaan Anda sebelumnya. Jika Anda ingin mengajukan pertanyaan baru di obrolan yang tidak terkait dengan topik percakapan sebelumnya, kami sarankan Anda memulai obrolan baru dengan model tersebut.

Ekstrak informasi dari dokumen dengan kueri dokumen

> **Note**
>
> Bagian ini mengasumsikan bahwa Anda telah menyelesaikan bagian di atasPrasyarat untuk kueri dokumen.

Kueri dokumen adalah fitur yang dapat Anda gunakan saat berinteraksi dengan model dasar di Canvas. Dengan kueri dokumen, Anda dapat mengakses kumpulan dokumen yang disimpan dalam indeks Amazon Kendra, yang menyimpan konten dokumen Anda dan disusun sedemikian rupa untuk membuat dokumen dapat dicari. Anda dapat mengajukan pertanyaan spesifik yang ditargetkan ke data dalam indeks Amazon Kendra Anda, dan model yayasan mengembalikan jawaban atas pertanyaan Anda. Misalnya, Anda dapat menanyakan basis pengetahuan internal informasi TI dan mengajukan pertanyaan seperti “Bagaimana cara saya terhubung ke jaringan perusahaan saya?” Untuk informasi selengkapnya tentang menyiapkan indeks, lihat Panduan Pengembang Amazon Kendra.

Saat menggunakan fitur kueri dokumen, model dasar membatasi tanggapan mereka terhadap konten dokumen dalam indeks Anda dengan teknik yang disebut Retrieval Augmented Generation (RAG). Teknik ini menggabungkan informasi yang paling relevan dari indeks bersama dengan prompt pengguna dan mengirimkannya ke model foundation untuk mendapatkan respons. Respons terbatas

pada apa yang dapat ditemukan di indeks Anda, mencegah model memberi Anda respons yang salah berdasarkan data eksternal. Untuk informasi lebih lanjut tentang proses ini, lihat posting blog Membangun aplikasi AI Generatif dengan akurasi tinggi dengan cepat pada data perusahaan.

Untuk memulai, dalam obrolan dengan model dasar di Canvas, aktifkan sakelar Kueri dokumen di bagian atas halaman. Dari dropdown, pilih indeks Amazon Kendra yang ingin Anda kueri. Kemudian, Anda dapat mulai mengajukan pertanyaan terkait dengan dokumen dalam indeks Anda.

> **Important**
>
> Kueri dokumen mendukung Bandingkan output model fitur ini. Setiap riwayat obrolan yang ada akan ditimpa saat Anda memulai obrolan baru untuk membandingkan output model.

Manajemen model

> **Note**
>
> Bagian berikut menjelaskan memulai dan mematikan model, yang hanya berlaku untuk model SageMaker JumpStart pondasi, seperti Falcon-40B-Instruct. Anda dapat mengakses model Amazon Bedrock, seperti Amazon Titan, secara instan kapan saja.

Anda dapat memulai SageMaker JumpStart model sebanyak yang Anda suka. Setiap SageMaker JumpStart model aktif dikenakan biaya pada akun Anda, jadi kami menyarankan Anda untuk tidak memulai lebih banyak model daripada yang Anda gunakan saat ini.

Untuk memulai model lain, Anda dapat melakukan hal berikut:

1. Pada halaman Menghasilkan, mengekstrak, dan meringkas konten, pilih Obrolan baru.
2. Pilih model dari menu dropdown. Jika Anda ingin memilih model yang tidak ditampilkan di dropdown, pilih Mulai model lain, lalu pilih model yang ingin Anda mulai.
3. Pilih model Start up.

Model akan mulai memulai, dan dalam beberapa menit Anda dapat mengobrol dengan model.

Kami sangat menyarankan Anda mematikan model yang tidak Anda gunakan. Model secara otomatis mati setelah 2 jam tidak aktif. Namun, untuk mematikan model secara manual, Anda dapat melakukan hal berikut:

1. Pada halaman Menghasilkan, mengekstrak, dan meringkas konten, buka obrolan untuk model yang ingin Anda matikan.
2. Pada halaman obrolan, pilih ikon Opsi lainnya (⁝).
3. Pilih Shut down model.
4. Di kotak Matikan konfirmasi model, pilih Matikan.

Model mulai dimatikan. Jika obrolan Anda membandingkan dua model atau lebih, Anda dapat mematikan model individual dari halaman obrolan dengan memilih ikon Opsi Lainnya (⁝) model dan kemudian memilih Matikan model.

Bandingkan output model

Anda mungkin ingin membandingkan output dari model yang berbeda secara berdampingan untuk melihat output model mana yang Anda sukai. Ini dapat membantu Anda memutuskan model mana yang paling cocok untuk kasus penggunaan Anda. Anda dapat membandingkan hingga tiga model dalam obrolan.

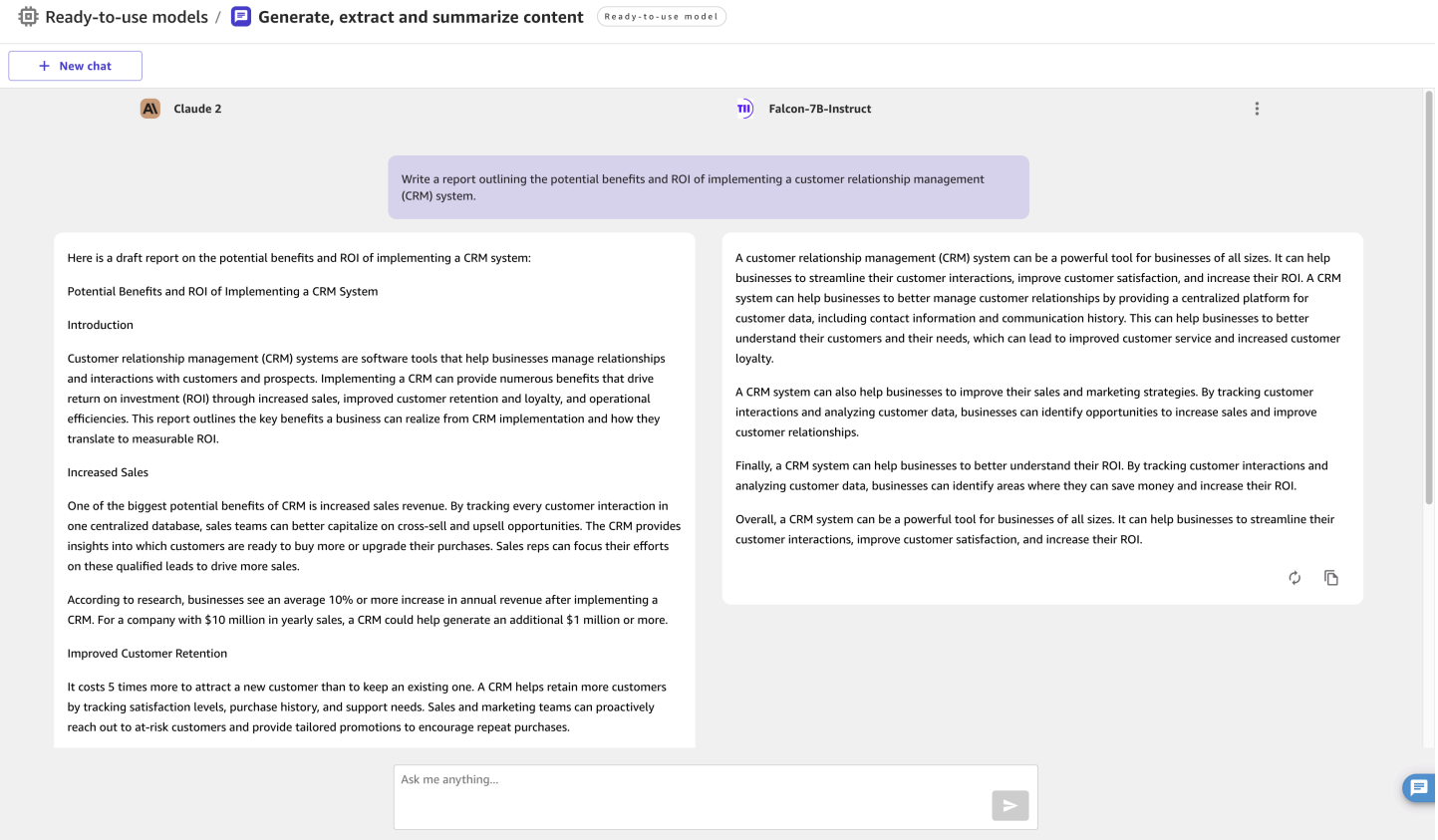
> **Note**
>
> Setiap model individu dikenakan biaya pada akun Anda.

Anda harus memulai obrolan baru untuk menambahkan model untuk perbandingan. Untuk membandingkan output model secara berdampingan dalam obrolan, lakukan hal berikut:

1. Dalam obrolan, pilih Obrolan baru.
2. Pilih Bandingkan, dan gunakan menu tarik-turun untuk memilih model yang ingin Anda tambahkan. Untuk menambahkan model ketiga, pilih Bandingkan lagi untuk menambahkan model lain.

   > **Note**
   >
   > Jika Anda ingin menggunakan SageMaker JumpStart model yang saat ini tidak aktif, Anda diminta untuk memulai model.

Saat model aktif, Anda melihat kedua model berdampingan dalam obrolan. Anda dapat mengirimkan prompt Anda, dan setiap model merespons dalam obrolan yang sama, seperti yang ditunjukkan pada gambar berikut.



Setelah selesai berinteraksi, pastikan untuk mematikan SageMaker JumpStart model apa pun secara individual untuk menghindari biaya lebih lanjut.

Model pondasi yang menyempurnakan

Model dasar yang dapat Anda akses melalui Amazon SageMaker Canvas dapat membantu Anda dengan berbagai tugas tujuan umum. Namun, jika Anda memiliki kasus penggunaan tertentu dan ingin respons yang disesuaikan berdasarkan data Anda sendiri, Anda dapat menyempurnakan model fondasi.

Untuk menyempurnakan model pondasi, Anda menyediakan kumpulan data yang terdiri dari petunjuk sampel dan respons model. Kemudian, Anda melatih model pondasi pada data. Akhirnya, model pondasi yang disetel dengan baik dapat memberi Anda respons yang lebih spesifik.

Topik ini menjelaskan cara menyempurnakan model foundation di Canvas.

Sebelum Anda memulai

Sebelum menyempurnakan model foundation, pastikan Anda memiliki izin untuk model eady-to-use R di Canvas dan hubungan kepercayaan dengan Amazon Bedrock. Lihat informasi yang lebih lengkap di Berikan Izin Pengguna untuk Menyempurnakan Model Foundation.

Anda juga harus memiliki kumpulan data yang diformat untuk menyempurnakan model bahasa besar (LLM). Berikut ini adalah daftar persyaratan untuk dataset Anda:

- Dataset harus berbentuk tabel dan berisi setidaknya dua kolom data teks — satu kolom input (yang berisi contoh petunjuk ke model) dan satu kolom keluaran (yang berisi contoh respons dari model).

  Contohnya adalah sebagai berikut:

Input	Output
Apa persyaratan pengiriman Anda?	Kami menawarkan pengiriman gratis untuk semua pesanan di atas $50. Pesanan di bawah $50 memiliki biaya pengiriman $5.99.
Bagaimana saya bisa mengembalikan barang?	Untuk mengembalikan barang, silakan kunjungi pusat pengembalian kami dan ikuti instruksinya. Anda harus memberikan nomor pesanan Anda dan alasan pengembalian.
Saya mengalami masalah dengan produk saya. Apa yang bisa saya lakukan?	Silakan hubungi tim dukungan pelanggan kami dan kami akan dengan senang hati membantu Anda memecahkan masalah ini.

- Kami merekomendasikan bahwa kumpulan data memiliki setidaknya 100 pasangan teks (baris item input dan output yang sesuai). Ini memastikan bahwa model pondasi memiliki data yang cukup untuk fine-tuning dan meningkatkan akurasi responsnya.
- Setiap item input dan output harus berisi maksimal 512 karakter. Apa pun yang lebih lama dikurangi menjadi 512 karakter saat menyempurnakan model pondasi.

Saat menyempurnakan model Amazon Bedrock, Anda harus mematuhi kuota Amazon Bedrock. Untuk informasi selengkapnya, lihat Kuota penyesuaian model di Panduan Pengguna Amazon Bedrock.

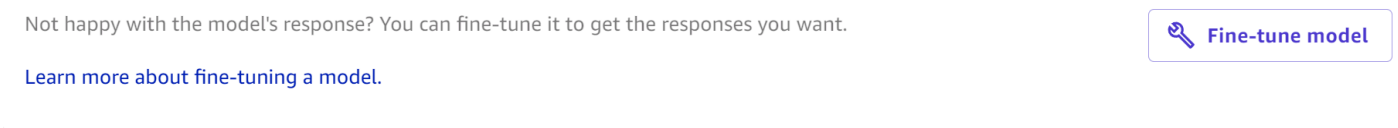
Untuk informasi selengkapnya tentang persyaratan dan batasan kumpulan data umum di Canvas, lihatBuat kumpulan data.

Sempurnakan model pondasi

Anda dapat menyempurnakan model foundation dengan menggunakan salah satu metode berikut dalam aplikasi Canvas:

- Saat dalam Menghasilkan, mengekstrak, dan meringkas obrolan konten dengan model foundation, pilih ikon model Fine-tune ().

- Saat dalam obrolan dengan model foundation, jika Anda telah membuat ulang respons dua kali atau lebih, Canvas menawarkan opsi untuk menyempurnakan model. Tangkapan layar berikut menunjukkan kepada Anda seperti apa tampilannya.



- Pada halaman Model saya, Anda dapat membuat model baru dengan memilih Model baru, lalu pilih Fine-tune foundation model.
- Pada halaman rumah eady-to-use model R, Anda dapat memilih Buat model Anda sendiri, dan kemudian di kotak dialog Buat model baru, pilih Fine-tune foundation model.
- Saat menjelajahi kumpulan data Anda di tab Data Wrangler, Anda dapat memilih kumpulan data dan memilih Buat model. Kemudian, pilih model foundation Fine-tune.

Setelah Anda mulai menyempurnakan model, lakukan hal berikut:

Pilih kumpulan data

Pada tab Select untuk menyempurnakan model, Anda memilih data yang ingin Anda latih model foundation.

Pilih kumpulan data yang ada atau buat kumpulan data baru yang memenuhi persyaratan yang tercantum di bagianSebelum Anda memulai. Untuk informasi selengkapnya tentang cara membuat kumpulan data, lihatBuat kumpulan data.

Ketika Anda telah memilih atau membuat kumpulan data dan Anda siap untuk melanjutkan, pilih Pilih kumpulan data.

Sempurnakan modelnya

Setelah memilih data Anda, Anda sekarang siap untuk memulai pelatihan dan menyempurnakan model.

Pada tab Fine-tune, lakukan hal berikut:

1. Untuk Pilih hingga 3 model dasar, buka menu tarik-turun dan periksa hingga 3 model dasar (hingga 2 SageMaker JumpStart model dan 1 model Amazon Bedrock) yang ingin Anda sesuaikan selama pekerjaan pelatihan. Dengan menyempurnakan beberapa model foundation, Anda dapat membandingkan kinerjanya dan akhirnya memilih yang paling cocok untuk kasus penggunaan Anda sebagai model default. Untuk informasi selengkapnya tentang model default, lihat???.
2. Untuk kolom Pilih Input, pilih kolom data teks dalam kumpulan data Anda yang berisi contoh petunjuk model.
3. Untuk kolom Select Output, pilih kolom data teks dalam kumpulan data Anda yang berisi contoh respons model.
4. (Opsional) Untuk mengonfigurasi pengaturan lanjutan untuk pekerjaan pelatihan, pilih Konfigurasi model. Untuk informasi selengkapnya tentang pengaturan pembuatan model lanjutan, lihatKonfigurasi bangunan model lanjutan.

   Di jendela pop-up Configure model, lakukan hal berikut:

   a. Untuk Hyperparameters, Anda dapat menyesuaikan hitungan Epoch, Ukuran Batch, Learning rate, dan Learning rate warmup untuk setiap model yang Anda pilih. Untuk informasi selengkapnya tentang parameter ini, lihat bagian Hyperparameters dalam SageMaker JumpStart dokumentasi.

   b. Untuk Pemisahan data, Anda dapat menentukan persentase cara membagi data antara set Pelatihan dan set Validasi.

   c. Untuk runtime pekerjaan Max, Anda dapat mengatur jumlah waktu maksimum Canvas menjalankan pekerjaan pembuatan. Fitur ini hanya tersedia untuk model SageMaker JumpStart foundation.

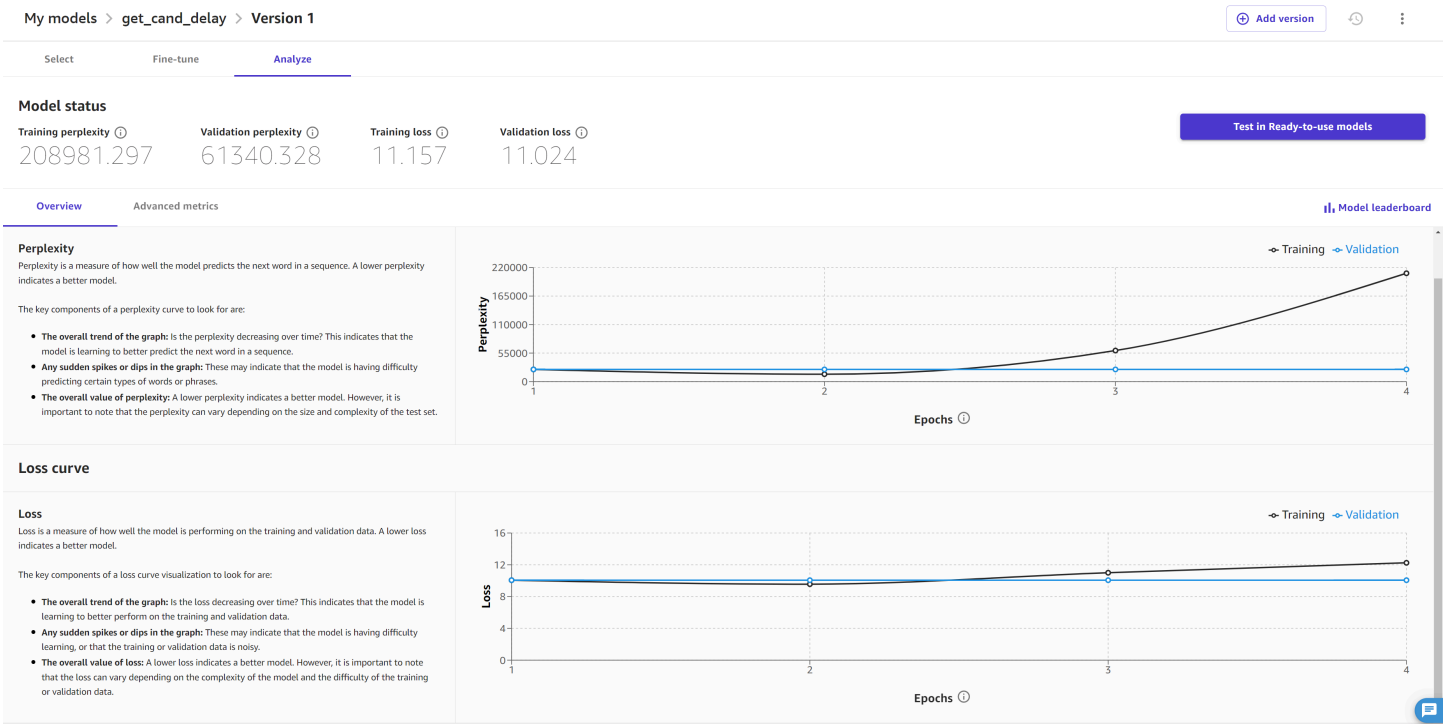
   d. Setelah mengonfigurasi pengaturan, pilih Simpan.
5. Pilih Fine-tune untuk mulai melatih model fondasi yang Anda pilih.

Setelah pekerjaan fine-tuning dimulai, Anda dapat meninggalkan halaman. Saat model ditampilkan sebagai Siap di halaman Model Saya, model siap digunakan, dan sekarang Anda dapat menganalisis kinerja model yang disetel dengan baik.

Analisis model yang disetel dengan baik

Pada tab Analisis model yang disetel dengan baik, Anda dapat melihat kinerja model.

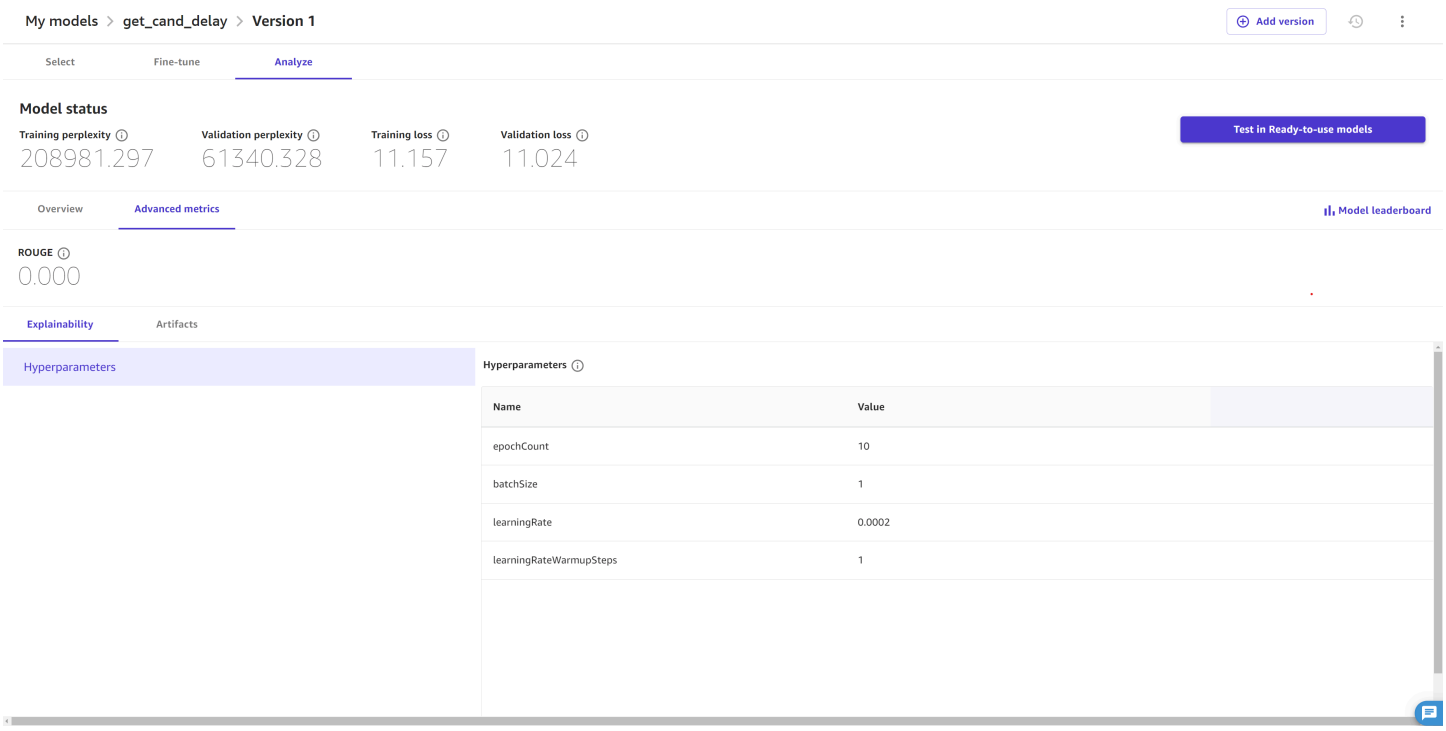
Tab Ikhtisar di halaman ini menunjukkan kepada Anda skor kebingungan dan kerugian, bersama dengan analisis yang memvisualisasikan peningkatan model dari waktu ke waktu selama pelatihan. Tangkapan layar berikut menunjukkan tab Ikhtisar.



Di halaman ini, Anda dapat melihat visualisasi berikut:

- Kurva kebingungan mengukur seberapa baik model memprediksi kata berikutnya secara berurutan, atau seberapa tata bahasa keluaran model tersebut. Idealnya, ketika model meningkat selama pelatihan, skor menurun dan menghasilkan kurva yang turun dan rata seiring waktu.
- Kurva Kerugian mengukur perbedaan antara output yang benar dan output yang diprediksi model. Kurva kerugian yang menurun dan rata dari waktu ke waktu menunjukkan bahwa model meningkatkan kemampuannya untuk membuat prediksi yang akurat.

Tab Metrik lanjutan menunjukkan hiperparameter dan metrik tambahan untuk model Anda. Sepertinya tangkapan layar berikut:



Tab Metrik lanjutan berisi informasi berikut:

- Bagian Explainability berisi Hyperparameters, yang merupakan nilai yang ditetapkan sebelum pekerjaan untuk memandu fine-tuning model. Jika Anda tidak menentukan hyperparameters kustom dalam pengaturan lanjutan model di Sempurnakan modelnya bagian, maka Canvas memilih hyperparameters default untuk Anda.

  Untuk SageMaker JumpStart model, Anda juga dapat melihat metrik lanjutan ROUGE (Recall-Oriented Understudy for Gisting Evaluation), yang mengevaluasi kualitas ringkasan yang dihasilkan oleh model. Ini mengukur seberapa baik model dapat merangkum poin-poin utama dari suatu bagian.
- Bagian Artefak memberi Anda tautan ke artefak yang dihasilkan selama pekerjaan fine-tuning. Anda dapat mengakses data pelatihan dan validasi yang disimpan di Amazon S3, serta tautan ke laporan evaluasi model (untuk mempelajari lebih lanjut, lihat paragraf berikut).

Untuk mendapatkan lebih banyak wawasan evaluasi model, Anda dapat mengunduh laporan yang dihasilkan menggunakan SageMaker Clarify, yang merupakan fitur yang dapat membantu Anda mendeteksi bias dalam model dan data Anda. Pertama, buat laporan dengan memilih Hasilkan laporan evaluasi di bagian bawah halaman. Setelah laporan dibuat, Anda dapat mengunduh laporan lengkap dengan memilih Unduh laporan atau dengan kembali ke bagian Artefak.

Anda juga dapat mengakses notebook Jupyter yang menunjukkan cara mereplikasi pekerjaan fine-tuning Anda dalam kode Python. Anda dapat menggunakan ini untuk mereplikasi atau membuat perubahan terprogram pada pekerjaan fine-tuning Anda atau mendapatkan pemahaman yang lebih dalam tentang bagaimana Canvas menyempurnakan model Anda. Untuk mempelajari lebih lanjut tentang notebook model dan cara mengaksesnya, lihatUnduh notebook model.

Untuk informasi selengkapnya tentang cara menafsirkan informasi di tab Analisis model yang disetel dengan baik, lihat topiknya. Evaluasi Kinerja Model Anda di Amazon SageMaker Canvas

Setelah menganalisis tab Gambaran Umum dan metrik lanjutan, Anda juga dapat memilih untuk membuka papan peringkat Model, yang menampilkan daftar model dasar yang dilatih selama pembuatan. Model dengan skor kerugian terendah dianggap sebagai model berkinerja terbaik dan dipilih sebagai model Default, yang merupakan model yang analisisnya Anda lihat di tab Analisis. Anda hanya dapat menguji dan menerapkan model default. Untuk informasi selengkapnya tentang papan peringkat model dan cara mengubah model default, lihat. ???

Uji model yang disetel dengan baik dalam obrolan

Setelah menganalisis kinerja model yang disetel dengan baik, Anda mungkin ingin mengujinya atau membandingkan responsnya dengan model dasar. Anda dapat menguji model yang disetel dengan baik dalam obrolan di fitur Menghasilkan, mengekstrak, dan meringkas konten.

Mulai obrolan dengan model yang disetel dengan memilih salah satu metode berikut:

- Pada tab Analisis model yang disetel dengan baik, pilih Test in R eady-to-use foundation models.
- Pada halaman eady-to-use model Canvas R, pilih Menghasilkan, mengekstrak, dan meringkas konten. Kemudian, pilih Obrolan baru dan pilih versi model yang disetel dengan baik yang ingin Anda uji.

Model dimulai dalam obrolan, dan Anda dapat berinteraksi dengannya seperti model dasar lainnya. Anda dapat menambahkan lebih banyak model ke obrolan dan membandingkan outputnya. Untuk informasi selengkapnya tentang fungsionalitas obrolan, lihatGunakan AI generatif dengan model foundation.

Gunakan eady-to-use model R

Dengan eady-to-use model Amazon SageMaker Canvas R, Anda dapat membuat prediksi pada data Anda tanpa menulis satu baris kode pun atau harus membuat model—yang harus Anda bawa

hanyalah data Anda. eady-to-use Model R menggunakan model pra-bangun untuk menghasilkan prediksi tanpa mengharuskan Anda menghabiskan waktu, keahlian, atau biaya yang diperlukan untuk membangun model, dan Anda dapat memilih dari berbagai kasus penggunaan mulai dari deteksi bahasa hingga analisis biaya.

Canvas terintegrasi dengan AWS layanan yang ada, seperti Amazon Ttract, Amazon Rekognition, dan Amazon Comprehend, untuk menganalisis data Anda dan membuat prediksi atau mengekstrak wawasan. Anda dapat menggunakan kekuatan prediksi layanan ini dari dalam aplikasi Canvas untuk mendapatkan prediksi berkualitas tinggi untuk data Anda.

Canvas mendukung jenis eady-to-use model R berikut:

eady-to-use Model R	Deskripsi	Tipe data yang didukung
Analisis sentimen	Mendeteksi sentimen dalam baris teks, yang bisa positif, negatif, netral, atau campuran. Saat ini, Anda hanya dapat melakukan analisis sentimen untuk teks bahasa Inggris.	Teks biasa atau tabel (CSV, Parket)
Entitas ekstraksi	Ekstrak entitas, yang merupakan objek dunia nyata seperti orang, tempat, dan barang komersial, atau unit seperti tanggal dan jumlah, dari teks.	Teks biasa atau tabel (CSV, Parket)
Deteksi bahasa	Tentukan bahasa dominan dalam teks seperti Inggris, Prancis, atau Jerman.	Teks biasa atau tabel (CSV, Parket)
Deteksi informasi pribadi	Mendeteksi informasi pribadi yang dapat digunakan untuk mengidentifikasi seseorang , seperti alamat, nomor rekening bank, dan nomor telepon, dari teks.	Teks biasa atau tabel (CSV, Parket)

eady-to-use Model R	Deskripsi	Tipe data yang didukung
Deteksi objek dalam gambar	Mendeteksi objek, konsep, adegan, dan tindakan dalam gambar Anda.	Gambar (JPG, PNG)
Deteksi teks dalam gambar	Deteksi teks dalam gambar Anda.	Gambar (JPG, PNG)
Analisis biaya	Ekstrak informasi dari faktur dan tanda terima, seperti tanggal, nomor, harga barang, jumlah total, dan ketentuan pembayaran.	Dokumen (PDF, JPG, PNG, TIFF)
Analisis dokumen identitas	Ekstrak informasi dari paspor, SIM, dan dokumenta si identitas lainnya yang dikeluarkan oleh Pemerintah AS.	Dokumen (PDF, JPG, PNG, TIFF)
Analisis dokumen	Menganalisis dokumen dan formulir untuk hubungan antara teks yang terdeteksi.	Dokumen (PDF, JPG, PNG, TIFF)
Kueri dokumen	Ekstrak informasi dari dokumen terstruktur seperti paystub, laporan bank, W-2, dan formulir aplikasi hipotek dengan mengajuka n pertanyaan menggunakan bahasa alami.	Dokumen (PDF)

Memulai

Untuk memulai dengan eady-to-use model R, tinjau informasi berikut.

Prasyarat

Untuk menggunakan eady-to-use model R di Canvas, Anda harus mengaktifkan izin konfigurasi eady-to-use model R Canvas saat menyiapkan SageMaker Domain Amazon Anda. Konfigurasi eady-to-use model Canvas R melampirkan ServicesAccess kebijakan AmazonSageMakerCanvasAI ke peran eksekusi pengguna Canvas AWS Identity and Access Management (IAM) Anda. Jika Anda mengalami masalah dengan pemberian izin, lihat topiknya. Memecahkan masalah dengan pemberian izin melalui konsol SageMaker

Jika sudah menyiapkan Domain, Anda dapat mengedit pengaturan Domain dan mengaktifkan izin. Untuk petunjuk tentang cara mengedit setelan Domain Anda, lihat Melihat dan Mengedit Domain. Saat mengedit pengaturan untuk Domain Anda, buka pengaturan Canvas dan nyalakan opsi Aktifkan eady-to-use model R Canvas.

(Opsional) Menyisih dari penyimpanan data layanan AI

Layanan AWS AI tertentu menyimpan dan menggunakan data Anda untuk melakukan perbaikan pada layanan. Anda dapat memilih untuk tidak menyimpan data Anda atau digunakan untuk peningkatan layanan. Untuk mempelajari lebih lanjut tentang cara memilih keluar, lihat kebijakan opt-out layanan AI di AWS OrganizationsPanduan Pengguna.

Cara menggunakan eady-to-use model R

Untuk memulai dengan eady-to-use model R, lakukan hal berikut:

1. (Opsional) Impor data Anda. Anda dapat mengimpor kumpulan data tabel, gambar, atau dokumen untuk menghasilkan prediksi batch, atau kumpulan data prediksi, dengan model R. eady-to-use Untuk memulai mengimpor dataset, lihat. Impor data ke dalam aliran data
2. Hasilkan prediksi. Anda dapat menghasilkan prediksi tunggal atau batch dengan eady-to-use model R pilihan Anda. Untuk memulai dengan membuat prediksi, lihatBuat prediksi dengan model R eady-to-use .

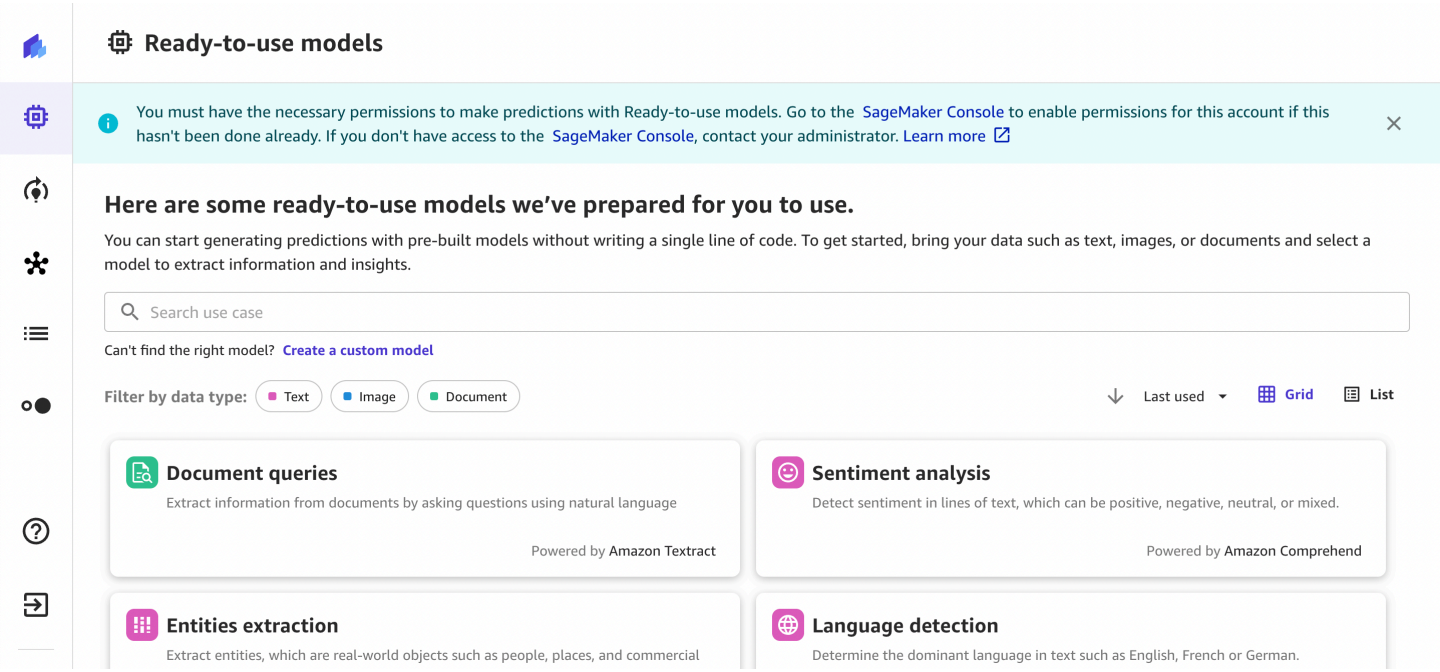
Buat prediksi dengan model R eady-to-use

eady-to-use Model R tersedia untuk data teks, gambar, dan dokumen. Setiap tipe data memiliki eady-to-use model R yang dirancang untuk bekerja paling baik untuk setiap kasus penggunaan. Gunakan panduan berikut untuk menentukan eady-to-use model R mana yang dapat Anda gunakan dengan data input Anda:

- Data teks: Analisis sentimen, ekstraksi entitas, deteksi bahasa, deteksi informasi pribadi

- Data gambar: Deteksi objek dalam gambar, deteksi teks dalam gambar
- Data dokumen: Analisis biaya, analisis dokumen identitas, analisis dokumen, kueri dokumen

Tangkapan layar berikut menunjukkan halaman arahan untuk eady-to-use model R, yang menampilkan semua solusi yang berbeda.



Setiap eady-to-use model R mendukung prediksi Tunggal dan prediksi Batch untuk kumpulan data Anda. Prediksi tunggal adalah ketika Anda hanya perlu membuat satu prediksi. Misalnya, Anda memiliki satu gambar dari mana Anda ingin mengekstrak teks, atau satu paragraf teks yang ingin Anda deteksi bahasa dominannya. Prediksi Batch adalah saat Anda ingin membuat prediksi untuk seluruh kumpulan data. Misalnya, Anda mungkin memiliki file CSV ulasan pelanggan yang ingin Anda analisis sentimen pelanggan, atau Anda mungkin memiliki file gambar di mana Anda ingin mendeteksi objek.

Bila Anda memiliki data dan telah mengidentifikasi kasus penggunaan Anda, pilih salah satu alur kerja berikut untuk membuat prediksi data Anda.

Buat prediksi untuk data teks

Prosedur berikut menjelaskan cara membuat prediksi tunggal dan batch untuk kumpulan data teks. Anda dapat menggunakan prosedur untuk jenis eady-to-use model R berikut: analisis sentimen, ekstraksi entitas, deteksi bahasa, dan deteksi informasi pribadi.

> **Note**
>
> Untuk analisis sentimen, Anda hanya dapat menggunakan teks bahasa Inggris.

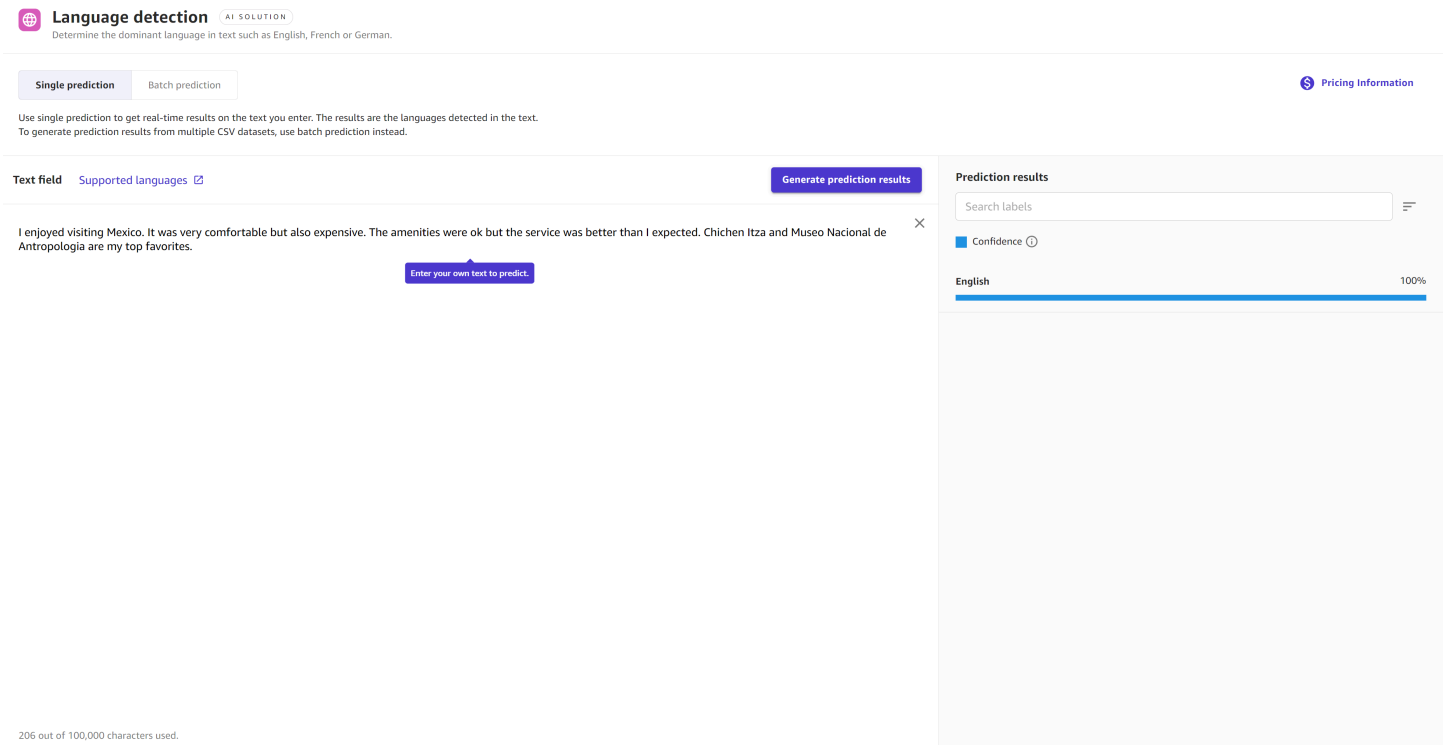
Prediksi tunggal

Untuk membuat prediksi tunggal untuk eady-to-use model R yang menerima data teks, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih eady-to-usemodel R.
2. Pada halaman eady-to-use model R, pilih eady-to-use model R untuk kasus penggunaan Anda. Untuk data teks, harus salah satu dari yang berikut: Analisis sentimen, ekstraksi Entitas, Deteksi bahasa, atau Deteksi informasi pribadi.
3. Pada halaman prediksi Jalankan untuk eady-to-use model R yang Anda pilih, pilih Prediksi tunggal.
4. Untuk bidang Teks, masukkan teks yang ingin Anda prediksi.
5. Pilih Hasilkan hasil prediksi untuk mendapatkan prediksi Anda.

Di panel kanan Hasil prediksi, Anda menerima analisis teks Anda selain skor Keyakinan untuk setiap hasil atau label. Misalnya, jika Anda memilih deteksi bahasa dan memasukkan bagian teks dalam bahasa Prancis, Anda mungkin mendapatkan bahasa Prancis dengan skor kepercayaan 95% dan jejak bahasa lain, seperti bahasa Inggris, dengan skor kepercayaan 5%.

Tangkapan layar berikut menunjukkan hasil untuk prediksi tunggal menggunakan deteksi bahasa di mana modelnya 100% yakin bahwa bagian tersebut adalah bahasa Inggris.



Prediksi Batch

Untuk membuat prediksi batch untuk eady-to-use model R yang menerima data teks, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih eady-to-usemodel R.
2. Pada halaman eady-to-use model R, pilih eady-to-use model R untuk kasus penggunaan Anda. Untuk data teks, harus salah satu dari yang berikut: Analisis sentimen, ekstraksi Entitas, Deteksi bahasa, atau Deteksi informasi pribadi.
3. Pada halaman Jalankan prediksi untuk eady-to-use model R pilihan Anda, pilih prediksi Batch.
4. Pilih Pilih kumpulan data jika Anda telah mengimpor dataset Anda. Jika tidak, pilih Impor dataset baru, dan kemudian Anda diarahkan melalui alur kerja data impor.
5. Dari daftar kumpulan data yang tersedia, pilih kumpulan data Anda dan pilih Hasilkan prediksi untuk mendapatkan prediksi Anda.

Setelah pekerjaan prediksi selesai berjalan, pada halaman Jalankan prediksi, Anda akan melihat kumpulan data keluaran yang tercantum di bawah Prediksi. Kumpulan data ini berisi hasil Anda, dan jika Anda memilih ikon Opsi lainnya ( ),

Anda dapat Pratinjau data keluaran. Kemudian, Anda dapat memilih Unduh untuk mengunduh hasilnya.

Buat prediksi untuk data gambar

Prosedur berikut menjelaskan cara membuat prediksi tunggal dan batch untuk kumpulan data gambar. Anda dapat menggunakan prosedur untuk jenis eady-to-use model R berikut: gambar deteksi objek dan deteksi teks dalam gambar.

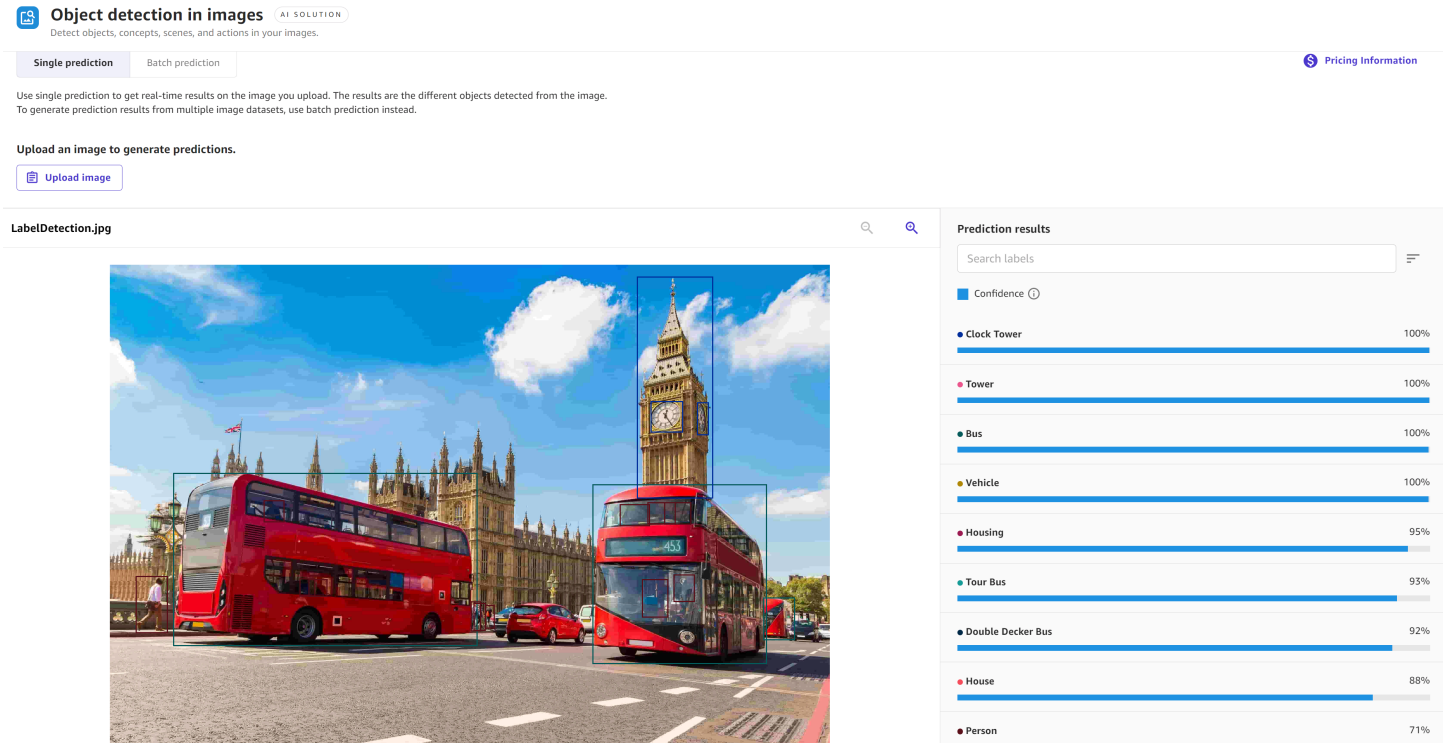
Prediksi tunggal

Untuk membuat prediksi tunggal untuk eady-to-use model R yang menerima data gambar, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih eady-to-usemodel R.
2. Pada halaman eady-to-use model R, pilih eady-to-use model R untuk kasus penggunaan Anda. Untuk data gambar, itu harus salah satu dari yang berikut: Gambar deteksi objek atau Deteksi teks dalam gambar.
3. Pada halaman prediksi Jalankan untuk eady-to-use model R yang Anda pilih, pilih Prediksi tunggal.
4. Pilih Unggah gambar.
5. Anda diminta untuk memilih gambar untuk diunggah dari komputer lokal Anda. Pilih gambar dari file lokal Anda, dan kemudian hasil prediksi dihasilkan.

Di panel kanan hasil Prediksi, Anda menerima analisis gambar Anda selain skor Keyakinan untuk setiap objek atau teks yang terdeteksi. Misalnya, jika Anda memilih deteksi objek dalam gambar, Anda menerima daftar objek dalam gambar bersama dengan skor kepercayaan tentang seberapa pasti model bahwa setiap objek terdeteksi secara akurat, seperti 93%.

Tangkapan layar berikut menunjukkan hasil untuk prediksi tunggal menggunakan deteksi objek dalam solusi gambar, di mana model memprediksi objek seperti menara jam dan bus dengan kepercayaan 100%.



Prediksi Batch

Untuk membuat prediksi batch untuk eady-to-use model R yang menerima data gambar, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih eady-to-usemodel R.
2. Pada halaman eady-to-use model R, pilih eady-to-use model R untuk kasus penggunaan Anda. Untuk data gambar, itu harus salah satu dari yang berikut: Gambar deteksi objek atau Deteksi teks dalam gambar.
3. Pada halaman Jalankan prediksi untuk eady-to-use model R pilihan Anda, pilih prediksi Batch.
4. Pilih Pilih kumpulan data jika Anda telah mengimpor dataset Anda. Jika tidak, pilih Impor dataset baru, dan kemudian Anda diarahkan melalui alur kerja data impor.
5. Dari daftar kumpulan data yang tersedia, pilih kumpulan data Anda dan pilih Hasilkan prediksi untuk mendapatkan prediksi Anda.

Setelah pekerjaan prediksi selesai berjalan, pada halaman Jalankan prediksi, Anda akan melihat kumpulan data keluaran yang tercantum di bawah Prediksi. Kumpulan data ini berisi hasil Anda, dan jika Anda memilih ikon Opsi lainnya ( ),

Anda dapat memilih Lihat hasil prediksi untuk melihat pratinjau data keluaran. Kemudian, Anda dapat memilih Unduh prediksi dan unduh hasilnya sebagai file CSV atau ZIP.

Buat prediksi untuk data dokumen

Prosedur berikut menjelaskan cara membuat prediksi tunggal dan batch untuk kumpulan data dokumen. Anda dapat menggunakan prosedur untuk jenis eady-to-use model R berikut: analisis pengeluaran, analisis dokumen identitas, dan analisis dokumen.

> **Note**
>
> Untuk kueri dokumen, hanya prediksi tunggal yang saat ini didukung.

Prediksi tunggal

Untuk membuat prediksi tunggal untuk eady-to-use model R yang menerima data dokumen, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih eady-to-usemodel R.
2. Pada halaman eady-to-use model R, pilih eady-to-use model R untuk kasus penggunaan Anda. Untuk data dokumen, harus salah satu dari yang berikut: Analisis pengeluaran, analisis dokumen identitas, atau analisis dokumen.
3. Pada halaman prediksi Jalankan untuk eady-to-use model R yang Anda pilih, pilih Prediksi tunggal.
4. Jika eady-to-use model R Anda adalah analisis dokumen identitas atau analisis dokumen, selesaikan tindakan berikut. Jika Anda melakukan analisis pengeluaran atau kueri dokumen, lewati langkah ini dan pergi ke Langkah 5 atau Langkah 6, masing-masing.
    a. Pilih Unggah dokumen.
    b. Anda diminta untuk mengunggah file PDF, JPG, atau PNG dari komputer lokal Anda. Pilih dokumen dari file lokal Anda, dan kemudian hasil prediksi akan dihasilkan.
5. Jika eady-to-use model R Anda adalah analisis pengeluaran, lakukan hal berikut:
    a. Pilih Unggah faktur atau tanda terima.
    b. Anda diminta untuk mengunggah file PDF, JPG, PNG, atau TIFF dari komputer lokal Anda. Pilih dokumen dari file lokal Anda, dan kemudian hasil prediksi akan dihasilkan.
6. Jika eady-to-use model R Anda adalah kueri dokumen, lakukan hal berikut:

a. Pilih Unggah dokumen.

b. Anda diminta untuk mengunggah file PDF dari komputer lokal Anda. Pilih dokumen dari file lokal Anda. PDF Anda harus sepanjang 1—100 halaman.

> **Note**
>
> Jika Anda berada di wilayah Asia Pasifik (Seoul), Asia Pasifik (Singapura), Asia Pasifik (Sydney), atau Eropa (Frankfurt), maka ukuran PDF maksimum untuk kueri dokumen adalah 20 halaman.

c. Di panel sisi kanan, masukkan kueri untuk mencari informasi dalam dokumen. Jumlah karakter yang dapat Anda miliki dalam satu kueri adalah dari 1-200. Anda dapat menambahkan hingga 15 kueri sekaligus.

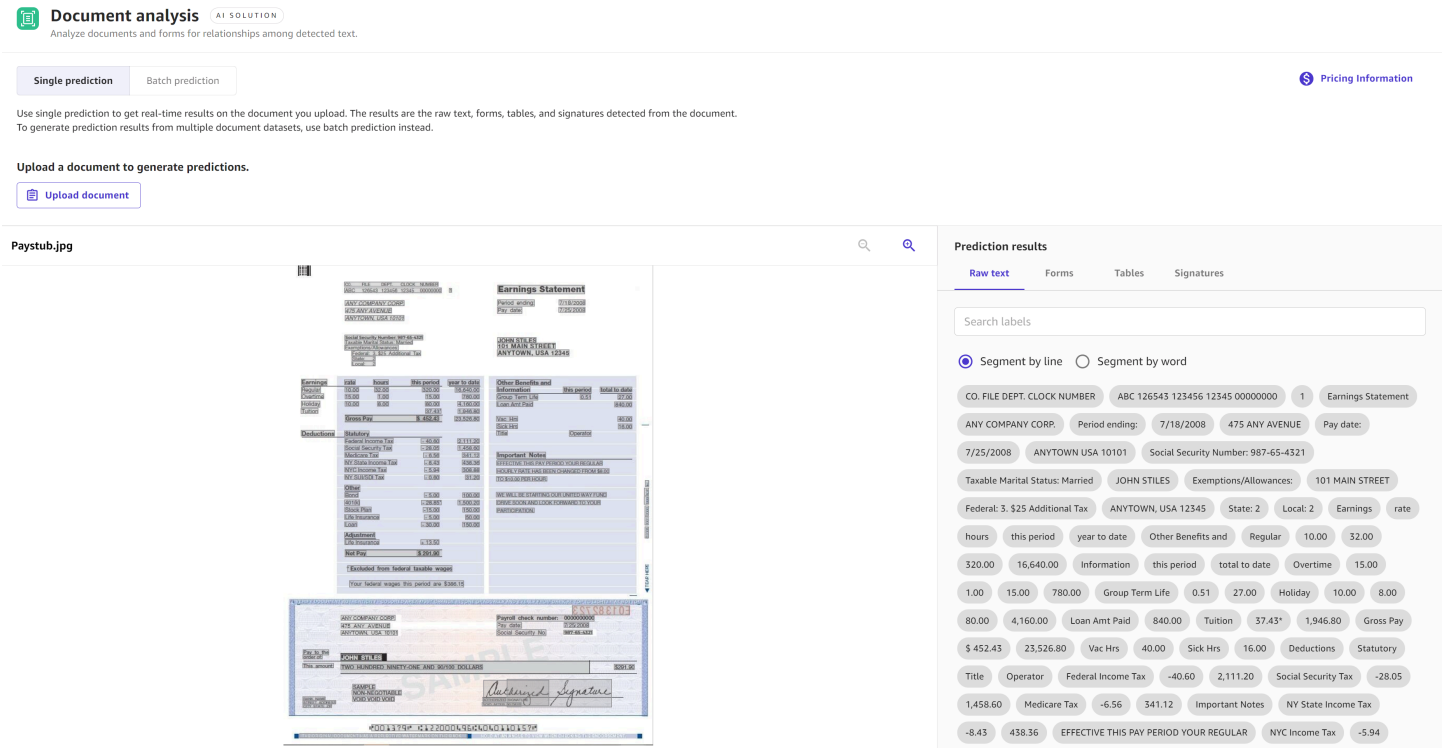
d. Pilih Kirim kueri, lalu hasilnya dihasilkan dengan jawaban atas pertanyaan Anda. Anda ditagih sekali untuk setiap pengiriman kueri yang Anda buat.

Di panel kanan Hasil prediksi, Anda akan menerima analisis dokumen Anda.

Informasi berikut menjelaskan hasil untuk setiap jenis solusi:

- Untuk analisis pengeluaran, hasilnya dikategorikan ke dalam bidang Ringkasan, yang mencakup bidang seperti total pada tanda terima, dan bidang item Baris, yang mencakup bidang seperti item individual pada tanda terima. Bidang yang diidentifikasi disorot pada gambar dokumen dalam output.
- Untuk analisis dokumen identitas, output menunjukkan bidang yang diidentifikasi oleh eady-to-use model R, seperti nama depan dan belakang, alamat, atau tanggal lahir. Bidang yang diidentifikasi disorot pada gambar dokumen dalam output.
- Untuk analisis dokumen, hasilnya dikategorikan ke dalam Teks mentah, Formulir, Tabel, dan Tanda Tangan. Teks mentah mencakup semua teks yang diekstraksi, sedangkan Formulir, Tabel, dan Tanda Tangan hanya menyertakan informasi pada formulir yang termasuk dalam kategori tersebut. Misalnya, Tabel hanya menyertakan informasi yang diekstrak dari tabel dalam dokumen. Bidang yang diidentifikasi disorot pada gambar dokumen dalam output.
- Untuk kueri dokumen, Canvas mengembalikan jawaban untuk setiap kueri Anda. Anda dapat membuka dropdown kueri yang dapat dilipat untuk melihat hasil, bersama dengan skor kepercayaan untuk prediksi. Jika Canvas menemukan beberapa jawaban dalam dokumen, maka Anda mungkin memiliki lebih dari satu hasil untuk setiap kueri.

Tangkapan layar berikut menunjukkan hasil untuk prediksi tunggal menggunakan solusi analisis dokumen.



Prediksi Batch

Untuk membuat prediksi batch untuk eady-to-use model R yang menerima data dokumen, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih eady-to-usemodel R.
2. Pada halaman eady-to-use model R, pilih eady-to-use model R untuk kasus penggunaan Anda. Untuk data gambar, harus salah satu dari yang berikut: Analisis biaya, analisis dokumen identitas, atau analisis dokumen.
3. Pada halaman Jalankan prediksi untuk eady-to-use model R pilihan Anda, pilih prediksi Batch.
4. Pilih Pilih kumpulan data jika Anda telah mengimpor dataset Anda. Jika tidak, pilih Impor dataset baru, dan kemudian Anda diarahkan melalui alur kerja data impor.
5. Dari daftar kumpulan data yang tersedia, pilih kumpulan data Anda dan pilih Hasilkan prediksi. Jika kasus penggunaan Anda adalah analisis dokumen, lanjutkan ke Langkah 6.
6. (Opsional) Jika kasus penggunaan Anda adalah Analisis dokumen, kotak dialog lain yang disebut Pilih fitur untuk disertakan dalam prediksi batch muncul. Anda dapat memilih Formulir, Tabel, dan Tanda Tangan untuk mengelompokkan hasil berdasarkan fitur tersebut. Kemudian, pilih Hasilkan prediksi.

Setelah pekerjaan prediksi selesai berjalan, pada halaman Jalankan prediksi,
Anda akan melihat kumpulan data keluaran yang tercantum di bawah Prediksi.
Kumpulan data ini berisi hasil Anda, dan jika Anda memilih ikon Opsi lainnya
(⋮),
Anda dapat memilih Lihat hasil prediksi untuk melihat pratinjau analisis data dokumen Anda.

Informasi berikut menjelaskan hasil untuk setiap jenis solusi:

- Untuk analisis pengeluaran, hasilnya dikategorikan ke dalam bidang Ringkasan, yang mencakup bidang seperti total pada tanda terima, dan bidang item Baris, yang mencakup bidang seperti item individual pada tanda terima. Bidang yang diidentifikasi disorot pada gambar dokumen dalam output.
- Untuk analisis dokumen identitas, output menunjukkan bidang yang diidentifikasi oleh eady-to-use model R, seperti nama depan dan belakang, alamat, atau tanggal lahir. Bidang yang diidentifikasi disorot pada gambar dokumen dalam output.
- Untuk analisis dokumen, hasilnya dikategorikan ke dalam Teks mentah, Formulir, Tabel, dan Tanda Tangan. Teks mentah mencakup semua teks yang diekstraksi, sedangkan Formulir, Tabel, dan Tanda Tangan hanya menyertakan informasi pada formulir yang termasuk dalam kategori tersebut. Misalnya, Tabel hanya menyertakan informasi yang diekstrak dari tabel dalam dokumen. Bidang yang diidentifikasi disorot pada gambar dokumen dalam output.

Setelah melihat pratinjau hasil, Anda dapat memilih Unduh prediksi dan mengunduh hasilnya sebagai file ZIP.

Gunakan model khusus

Dengan Amazon SageMaker Canvas, Anda dapat membuat model khusus yang dilatih dengan data Anda. Dengan melatih model khusus pada data Anda, Anda dapat menangkap karakteristik dan tren yang spesifik dan paling representatif dari data Anda. Misalnya, Anda mungkin ingin membuat model peramalan deret waktu khusus yang Anda latih pada data inventaris dari gudang Anda yang memungkinkan Anda mengelola operasi logistik Anda.

Anda dapat melatih model kustom Canvas pada jenis kumpulan data berikut:

- Tabular (termasuk data Numerik, Kategoris, Timeseries, dan Teks)
- Citra

Tabel berikut menunjukkan jenis model kustom yang dapat Anda buat di Canvas, bersama dengan tipe data dan sumber data yang didukung

Jenis model	Contoh kasus penggunaan	Jenis data yang didukung	Sumber data yang didukung
Prediksi numerik	Memprediksi harga rumah berdasarkan fitur seperti luas persegi	Numerik	Unggahan lokal, Amazon S3, konektor SaaS
2 kategori prediksi	Memprediksi apakah pelanggan cenderung churn atau tidak	Biner atau Kategoris	Unggahan lokal, Amazon S3, konektor SaaS
3+ prediksi kateogri	Memprediksi hasil pasien setelah keluar dari rumah sakit	Kategoris	Unggahan lokal, Amazon S3, konektor SaaS
Peramalan deret waktu	Memprediksi inventaris Anda untuk kuartal berikutnya	Timeseries	Unggahan lokal, Amazon S3, konektor SaaS
Prediksi gambar label tunggal	Memprediksi jenis cacat manufaktur pada gambar	Gambar (JPG, PNG)	Unggahan lokal, Amazon S3
Prediksi teks multi-kategori	Memprediksi kategori produk, seperti pakaian, elektronik, atau barang rumah tangga, berdasarkan deskripsi produk	Kolom sumber: Teks Kolom target: Biner atau Kategoris	Unggahan lokal, Amazon S3

Memulai

Untuk memulai membangun dan membuat prediksi dari model kustom, lakukan hal berikut:

- Tentukan kasus penggunaan dan jenis model yang ingin Anda bangun. Untuk informasi selengkapnya tentang jenis model kustom, lihatMembangun model kustom. Untuk informasi selengkapnya tentang tipe data dan sumber yang didukung untuk model kustom, lihatImpor data ke Canvas.
- Impor data Anda ke Canvas. Anda dapat membuat model khusus dengan kumpulan data tabel atau gambar apa pun yang memenuhi persyaratan input. Untuk informasi lebih lanjut tentang persyaratan input, lihatBuat kumpulan data.

  Untuk mempelajari lebih lanjut tentang kumpulan data sampel yang SageMaker disediakan yang dapat Anda coba, lihat Menggunakan kumpulan data sampel.
- Bangun model kustom Anda. Anda dapat membuat Quick build untuk mendapatkan model Anda dan mulai membuat prediksi lebih cepat, atau Anda dapat melakukan build Standar untuk akurasi yang lebih besar.

  Untuk jenis model peramalan numerik, kategoris, dan deret waktu, Anda dapat membersihkan dan menyiapkan data Anda dengan fitur-fitur seperti transformasi dan gabungan lanjutan. Untuk model prediksi gambar, Anda dapat Mengedit kumpulan data gambar memperbarui label atau menambah dan menghapus gambar. Perhatikan bahwa Anda tidak dapat menggunakan fitur ini untuk model prediksi teks multi-kategori.
- Evaluasi kinerja model Anda dan tentukan seberapa baik kinerjanya pada data dunia nyata.
- (Opsional) Untuk jenis model tertentu, Anda dapat berkolaborasi dengan ilmuwan data di Amazon SageMaker Studio Classic yang dapat membantu meninjau dan meningkatkan model Anda.
- Buat prediksi tunggal atau batch dengan model Anda.

> **Note**
>
> Jika Anda sudah memiliki model terlatih di Amazon SageMaker Studio Classic yang ingin Anda bagikan dengan Canvas, Anda dapat membawa model Anda sendiri ke SageMaker Canvas. Tinjau prasyarat BYOM untuk menentukan apakah model Anda memenuhi syarat untuk dibagikan.

Membangun model kustom

Gunakan Amazon SageMaker Canvas untuk membuat model kustom pada kumpulan data yang telah Anda impor. Gunakan model yang telah Anda buat untuk membuat prediksi pada data baru.

SageMaker Canvas menggunakan informasi dalam kumpulan data untuk membangun hingga 250 model dan memilih salah satu yang berkinerja terbaik.

Saat Anda mulai membuat model, Canvas secara otomatis merekomendasikan satu atau lebih jenis model. Jenis model termasuk dalam salah satu kategori berikut:

- Prediksi numerik — Ini dikenal sebagai regresi dalam pembelajaran mesin. Gunakan tipe model prediksi numerik saat Anda ingin membuat prediksi untuk data numerik. Misalnya, Anda mungkin ingin memprediksi harga rumah berdasarkan fitur seperti luas persegi rumah.
- Prediksi kategoris — Ini dikenal sebagai klasifikasi dalam pembelajaran mesin. Saat Anda ingin mengkategorikan data ke dalam grup, gunakan jenis model prediksi kategoris:
  - Prediksi kategori 2 — Gunakan tipe model prediksi kategori 2 (juga dikenal sebagai klasifikasi biner dalam pembelajaran mesin) ketika Anda memiliki dua kategori yang ingin Anda prediksi untuk data Anda. Misalnya, Anda mungkin ingin menentukan apakah pelanggan cenderung melakukan churn.
  - Prediksi kategori 3+ — Gunakan tipe model prediksi kategori 3+ (juga dikenal sebagai klasifikasi multi-kelas dalam pembelajaran mesin) ketika Anda memiliki tiga atau lebih kategori yang ingin Anda prediksi untuk data Anda. Misalnya, Anda mungkin ingin memprediksi status pinjaman pelanggan berdasarkan fitur seperti pembayaran sebelumnya.
- Peramalan deret waktu — Gunakan perkiraan deret waktu saat Anda ingin membuat prediksi selama periode waktu tertentu. Misalnya, Anda mungkin ingin memprediksi jumlah barang yang akan Anda jual pada kuartal berikutnya. Untuk informasi tentang prakiraan deret waktu, lihat Prakiraan Deret Waktu di Amazon SageMaker Canvas.
- Prediksi gambar - Gunakan jenis model prediksi gambar label tunggal (juga dikenal sebagai klasifikasi gambar label tunggal dalam pembelajaran mesin) saat Anda ingin menetapkan label ke gambar. Misalnya, Anda mungkin ingin mengklasifikasikan berbagai jenis cacat produksi dalam gambar produk Anda.
- Prediksi teks — Gunakan jenis model prediksi teks multi-kategori (juga dikenal sebagai klasifikasi teks multi-kelas dalam pembelajaran mesin) saat Anda ingin menetapkan label ke bagian teks. Misalnya, Anda mungkin memiliki kumpulan data ulasan pelanggan untuk suatu produk, dan Anda ingin menentukan apakah pelanggan menyukai atau tidak menyukai produk tersebut. Anda mungkin meminta model Anda memprediksi apakah bagian teks tertentu adalah`Positive`,`Negative`, atau`Neutral`.

Untuk tabel tipe data input yang didukung untuk setiap jenis model, lihatGunakan model khusus.

Untuk setiap model data tabular yang Anda buat (yang mencakup model prediksi numerik, kategoris, deret waktu, dan prediksi teks), Anda memilih kolom Target. Kolom Target adalah kolom yang berisi informasi yang ingin Anda prediksi. Misalnya, jika Anda membuat model untuk memprediksi apakah orang telah membatalkan langganan mereka, kolom Target berisi titik data yang merupakan status pembatalan seseorang `yes` atau status pembatalan seseorang. `no`

Untuk model prediksi gambar, Anda membuat model dengan kumpulan data gambar yang telah diberi label. Untuk gambar tak berlabel yang Anda berikan, model memprediksi label. Misalnya, jika Anda membuat model untuk memprediksi apakah gambar itu kucing atau kucing, Anda memberikan gambar berlabel kucing atau kucing saat membuat model. Kemudian, model dapat menerima gambar yang tidak berlabel dan memprediksinya sebagai kucing atau kucing.

Apa yang terjadi ketika Anda membangun model

Untuk membangun model Anda, Anda dapat memilih Quick build atau Standard build. Quick build memiliki waktu pembuatan yang lebih singkat, tetapi build Standar umumnya memiliki akurasi yang lebih tinggi. Tabel berikut menguraikan waktu build rata-rata untuk setiap model dan tipe build, bersama dengan jumlah titik data minimum dan maksimum yang harus Anda miliki untuk setiap tipe build.

Kuota	Prediksi numerik dan kategoris	Peramalan deret waktu	Prediksi gambar	Prediksi teks
Waktu pembuatan cepat	2-20 menit	2-20 menit	15-30 menit	15-30 menit
Waktu pembuatan standar	2-4 jam	2-4 jam	2-5 jam	2-5 jam
Jumlah maksimum entri (baris atau gambar) untuk build Cepat	50.000	50.000	5000	7500

Jika Anda keluar saat menjalankan Quick build, build Anda mungkin akan terganggu hingga Anda masuk lagi. Saat Anda masuk lagi, Canvas melanjutkan build Quick.

Canvas memprediksi nilai dengan menggunakan informasi di sisa kumpulan data, tergantung pada jenis model:

- Untuk prediksi kategoris, Canvas menempatkan setiap baris ke dalam salah satu kategori yang tercantum di kolom Target.
- Untuk prediksi numerik, Canvas menggunakan informasi dalam kumpulan data untuk memprediksi nilai numerik di kolom Target.
- Untuk peramalan deret waktu, Canvas menggunakan data historis untuk memprediksi nilai kolom Target di masa depan.
- Untuk prediksi gambar, Canvas menggunakan gambar yang telah diberi label untuk memprediksi label untuk gambar yang tidak berlabel.
- Untuk prediksi teks, Canvas menganalisis data teks yang telah diberi label untuk memprediksi label untuk bagian teks yang tidak berlabel.

Fitur tambahan untuk membantu Anda membangun model

> **Note**
>
> Fitur-fitur berikut tersedia untuk prediksi numerik dan kategoris dan model peramalan deret waktu.

Sebelum membuat model Anda, Anda dapat memfilter data Anda atau menyiapkannya menggunakan transformasi lanjutan. Untuk informasi selengkapnya tentang menyiapkan data Anda untuk pembuatan model, lihatSiapkan data dengan transformasi lanjutan.

Anda juga dapat menggunakan visualisasi dan analitik untuk menjelajahi data Anda dan menentukan fitur mana yang terbaik untuk disertakan dalam model Anda. Untuk informasi selengkapnya, lihat Menjelajahi dan menganalisis data Anda.

Untuk mempelajari lebih lanjut tentang fitur tambahan seperti melihat pratinjau model, memvalidasi kumpulan data, dan mengubah ukuran sampel acak yang digunakan untuk membuat model, lihat. Pratinjau model Anda

Untuk kumpulan data tabular dengan beberapa kolom (seperti kumpulan data untuk membangun tipe model peramalan kategoris, numerik, atau deret waktu), Anda mungkin memiliki baris dengan titik data yang hilang. Sementara Canvas membangun model, secara otomatis menambahkan nilai yang hilang. Canvas menggunakan nilai dalam kumpulan data Anda untuk melakukan pendekatan matematis untuk nilai yang hilang. Untuk akurasi model tertinggi, kami sarankan menambahkan data yang hilang jika Anda dapat menemukannya. Perhatikan bahwa fitur data yang hilang tidak didukung untuk prediksi teks atau model prediksi gambar.

Memulai

Untuk memulai membangun model kustom, lihat Membangun model dan ikuti prosedur untuk jenis model yang ingin Anda bangun.

Membangun model

Bagian berikut menunjukkan cara membangun model untuk masing-masing jenis utama model kustom.

- Untuk membangun prediksi numerik, prediksi kategori 2, atau 3+ model prediksi kategori, lihat. Membangun model prediksi numerik atau kategoris kustom
- Untuk membuat model prediksi gambar label tunggal, lihat. Membangun model prediksi gambar kustom
- Untuk membuat model prediksi teks multi-kategori, lihat. Membangun model prediksi teks kustom
- Untuk membuat model peramalan deret waktu, lihatBangun model peramalan deret waktu.

> **Note**
>
> Jika Anda menemukan kesalahan selama analisis pasca-pembangunan yang memberi tahu Anda untuk meningkatkan kuota untuk `ml.m5.2xlarge` instans, lihat Meminta Peningkatan Kuota.

Membangun model prediksi numerik atau kategoris kustom

Model prediksi numerik dan kategoris mendukung build Quick dan build Standar.

Untuk membangun model prediksi numerik atau kategoris, gunakan prosedur berikut:

1. Buka aplikasi SageMaker Canvas.

2. Di panel navigasi kiri, pilih Model saya.
3. Pilih Model baru.
4. Dalam kotak dialog Buat model baru, lakukan hal berikut:
    a. Masukkan nama di bidang Nama model.
    b. Pilih jenis masalah analisis prediktif.
    c. Pilih Buat.
5. Untuk Pilih kumpulan data, pilih kumpulan data Anda dari daftar kumpulan data. Jika Anda belum mengimpor data, pilih Impor untuk diarahkan melalui alur kerja data impor.
6. Saat Anda siap untuk mulai membuat model, pilih Pilih kumpulan data.
7. Pada tab Build, untuk daftar dropdown kolom Target, pilih target untuk model yang ingin diprediksi.
8. Untuk tipe Model, Canvas secara otomatis mendeteksi jenis masalah untuk Anda. Jika Anda ingin mengubah jenis atau mengonfigurasi pengaturan model lanjutan, pilih Konfigurasi model.

    Ketika kotak dialog Configure model terbuka, lakukan hal berikut:

    a. Untuk tipe Model, pilih jenis model yang ingin Anda bangun.
    b. Setelah Anda memilih jenis model, ada tambahan Pengaturan lanjutan. Untuk informasi selengkapnya tentang masing-masing pengaturan lanjutan, lihatKonfigurasi bangunan model lanjutan. Untuk mengkonfigurasi pengaturan lanjutan, lakukan hal berikut:
        i. (Opsional) Untuk menu tarik-turun metrik Objective, pilih metrik yang ingin dioptimalkan Canvas saat membuat model Anda. Jika Anda tidak memilih metrik, Canvas memilih satu untuk Anda secara default. Untuk deskripsi metrik yang tersedia, lihat. Referensi metrik
        ii. Untuk metode Pelatihan, pilih mode Auto, Ensemble, atau Hyperparameter optimization (HPO).
        iii. Untuk Algoritma, pilih algoritma yang ingin Anda sertakan untuk membangun kandidat model.
        iv. Untuk Pemisahan data, tentukan dalam persentase bagaimana Anda ingin membagi data antara set Pelatihan dan set Validasi. Set pelatihan digunakan untuk membangun model, sedangkan set validasi digunakan untuk menguji akurasi kandidat model.
        v. Untuk kandidat Max dan runtime, lakukan hal berikut:

A. Tetapkan nilai kandidat Max, atau jumlah maksimum kandidat model yang dapat dihasilkan Canvas. Perhatikan bahwa kandidat Max hanya tersedia dalam mode HPO.

B. Tetapkan nilai jam dan menit untuk runtime pekerjaan Max, atau jumlah waktu maksimum yang dapat dihabiskan Canvas untuk membangun model Anda. Setelah waktu maksimum, Canvas berhenti membangun dan memilih kandidat model terbaik.

c. Setelah mengonfigurasi pengaturan lanjutan, pilih Simpan.

9. Pilih atau batalkan pilihan kolom dalam data Anda untuk menyertakan atau menjatuhkannya dari build Anda.

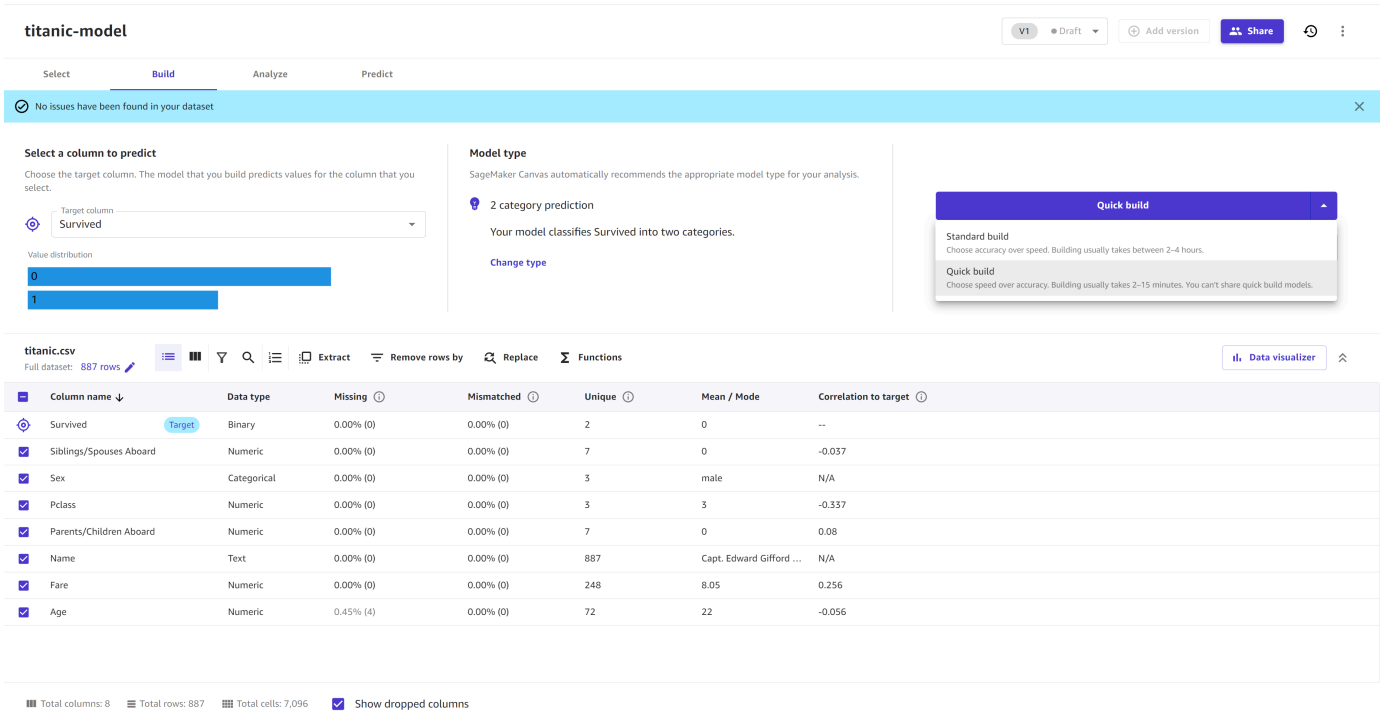
> **Note**
>
> Jika Anda membuat prediksi batch dengan model Anda setelah pembuatan, Canvas menambahkan kolom yang dijatuhkan ke hasil prediksi Anda. Namun, Canvas tidak menambahkan kolom yang dijatuhkan ke prediksi batch Anda untuk model deret waktu.

10. (Opsional) Gunakan alat visualisasi dan analitik yang disediakan Canvas untuk memvisualisasikan data Anda dan menentukan fitur mana yang mungkin ingin Anda sertakan dalam model Anda. Untuk informasi selengkapnya, lihat Menjelajahi dan menganalisis data Anda.

11. (Opsional) Gunakan transformasi data untuk membersihkan, mengubah, dan menyiapkan data Anda untuk pembuatan model. Untuk informasi selengkapnya, lihat Mempersiapkan data Anda dengan transformasi lanjutan. Anda dapat melihat dan menghapus transformasi Anda dengan memilih Resep model untuk membuka panel samping resep Model.

12. (Opsional) Untuk fitur tambahan seperti melihat pratinjau keakuratan model Anda, memvalidasi kumpulan data Anda, dan mengubah ukuran sampel acak yang diambil Canvas dari kumpulan data Anda, lihat. Pratinjau model Anda

13. Setelah meninjau data Anda dan membuat perubahan apa pun pada kumpulan data Anda, pilih Quick build atau Standard build untuk memulai pembuatan model Anda. Tangkapan layar berikut menunjukkan halaman Build dan opsi Quick build dan Standard build.



Setelah model Anda mulai membangun, Anda dapat meninggalkan halaman. Ketika model ditampilkan sebagai Siap di halaman Model saya, model siap untuk analisis dan prediksi.

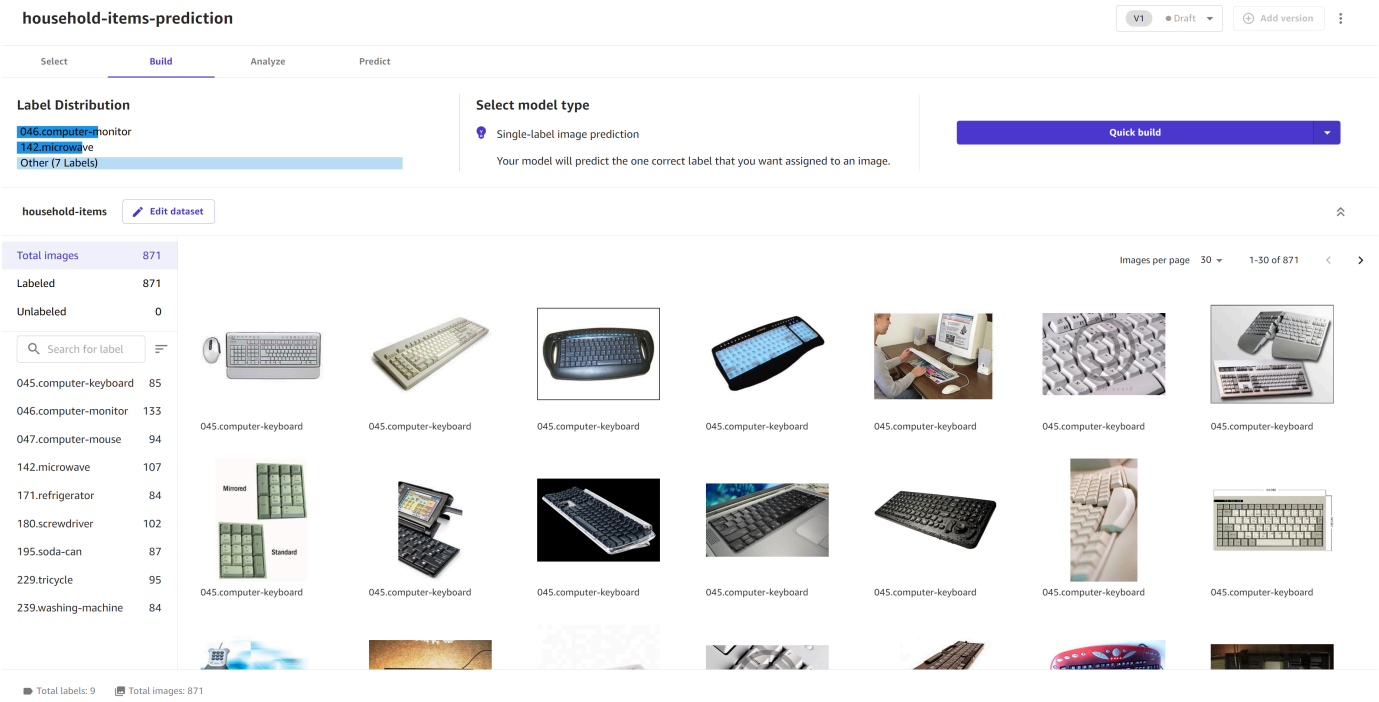
Membangun model prediksi gambar kustom

Model prediksi gambar label tunggal mendukung build Quick dan build Standar.

Untuk membuat model prediksi gambar label tunggal, gunakan prosedur berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.
3. Pilih Model baru.
4. Dalam kotak dialog Buat model baru, lakukan hal berikut:
    a. Masukkan nama di bidang Nama model.
    b. Pilih jenis masalah analisis gambar.
    c. Pilih Buat.
5. Untuk Pilih kumpulan data, pilih kumpulan data Anda dari daftar kumpulan data. Jika Anda belum mengimpor data, pilih Impor untuk diarahkan melalui alur kerja data impor.
6. Saat Anda siap untuk mulai membuat model, pilih Pilih kumpulan data.

7. Pada tab Build, Anda melihat distribusi Label untuk gambar dalam kumpulan data Anda. Tipe Model diatur ke prediksi gambar Single-label.
8. Di halaman ini, Anda dapat melihat pratinjau gambar Anda dan mengedit kumpulan data. Jika Anda memiliki gambar yang tidak berlabel, pilih Edit dataset dan. Tetapkan label ke gambar yang tidak berlabel Anda juga dapat melakukan tugas lain saat AndaMengedit kumpulan data gambar, seperti mengganti nama label dan menambahkan gambar ke kumpulan data.
9. Setelah meninjau data Anda dan membuat perubahan apa pun pada kumpulan data Anda, pilih Quick build atau Standard build untuk memulai pembuatan model Anda. Tangkapan layar berikut menunjukkan halaman Build dari model prediksi gambar yang siap dibuat.



Setelah model Anda mulai membangun, Anda dapat meninggalkan halaman. Ketika model ditampilkan sebagai Siap di halaman Model saya, model siap untuk analisis dan prediksi.

Membangun model prediksi teks kustom

Model prediksi teks multi-kategori mendukung build Cepat dan build Standar.

Untuk membuat model prediksi teks, gunakan prosedur berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.
3. Pilih Model baru.

4. Dalam kotak dialog Buat model baru, lakukan hal berikut:

   a. Masukkan nama di bidang Nama model.

   b. Pilih jenis masalah analisis Teks.

   c. Pilih Buat.

5. Untuk Pilih kumpulan data, pilih kumpulan data Anda dari daftar kumpulan data. Jika Anda belum mengimpor data, pilih Impor untuk diarahkan melalui alur kerja data impor.

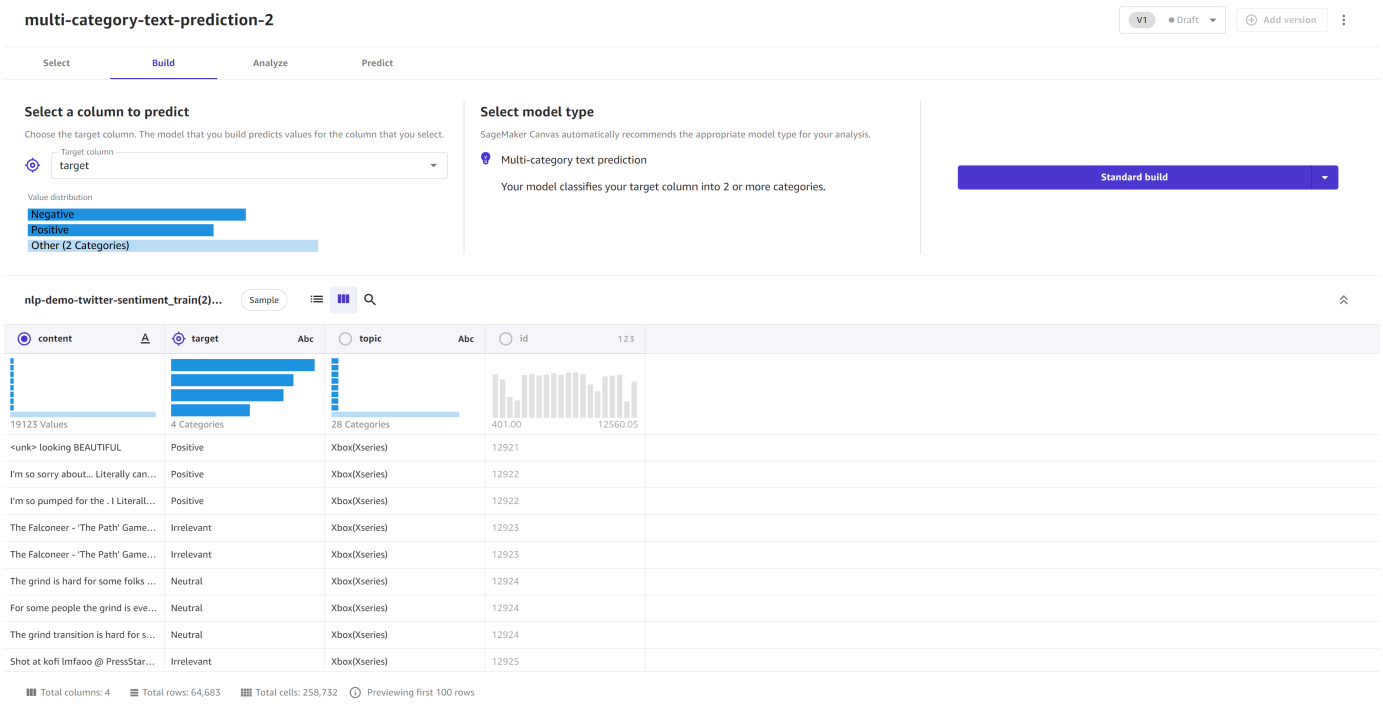
6. Saat Anda siap untuk mulai membuat model, pilih Pilih kumpulan data.

7. Pada tab Build, untuk daftar dropdown kolom Target, pilih target untuk model yang ingin diprediksi. Kolom target harus memiliki tipe data biner atau kategoris, dan harus ada setidaknya 25 entri (atau baris data) untuk setiap label unik di kolom target.

8. Untuk tipe Model, konfirmasikan bahwa jenis model secara otomatis disetel ke prediksi teks Multi-kategori.

9. Untuk kolom pelatihan, pilih kolom sumber data teks Anda. Ini harus berupa kolom yang berisi teks yang ingin Anda analisis.

10. Pilih Quick build atau Standard build untuk mulai membangun model Anda. Screenshot berikut menunjukkan halaman Build dari model prediksi teks yang siap dibuat.



Setelah model Anda mulai membangun, Anda dapat meninggalkan halaman. Ketika model ditampilkan sebagai Siap di halaman Model saya, model siap untuk analisis dan prediksi.

Bangun model peramalan deret waktu

Model peramalan deret waktu mendukung build Cepat dan build Standar.

Untuk membangun model peramalan deret waktu, gunakan prosedur berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.
3. Pilih Model baru.
4. Dalam kotak dialog Buat model baru, lakukan hal berikut:
    a. Masukkan nama di bidang Nama model.
    b. Pilih jenis masalah peramalan deret waktu.
    c. Pilih Buat.
5. Untuk Pilih kumpulan data, pilih kumpulan data Anda dari daftar kumpulan data. Jika Anda belum mengimpor data, pilih Impor untuk diarahkan melalui alur kerja data impor.
6. Saat Anda siap untuk mulai membuat model, pilih Pilih kumpulan data.
7. Pada tab Build, untuk daftar dropdown kolom Target, pilih target untuk model yang ingin diprediksi.
8. Untuk tipe Model, Canvas secara otomatis mendeteksi jenis masalah untuk Anda. Jika Anda ingin mengubah jenis atau mengonfigurasi pengaturan model lanjutan, pilih Konfigurasi model.

    Ketika kotak dialog Configure model terbuka, lakukan hal berikut:

    a. Untuk tipe Model, pilih jenis model yang ingin Anda bangun.
    b. Untuk konfigurasi deret waktu, tentukan nilai berikut:
        i. Untuk dropdown kolom ID Item, pilih kolom di kumpulan data Anda yang secara unik mengidentifikasi setiap baris.
        ii. (Opsional) Untuk dropdown kolom Grup, pilih kolom kategoris yang ingin Anda gunakan untuk mengelompokkan nilai peramalan Anda.
        iii. Untuk dropdown kolom stempel waktu, pilih kolom dengan stempel waktu (dalam format datetime). Untuk informasi selengkapnya tentang format datetime yang diterima, lihat. Prakiraan Deret Waktu di Amazon SageMaker Canvas

iv. Untuk bidang Panjang Forecast, masukkan periode waktu yang ingin Anda prediksi nilainya.

v. (Opsional) Aktifkan sakelar Gunakan jadwal liburan untuk memilih jadwal liburan dari berbagai negara.

vi. Pilih Simpan.

c. Setelah mengonfigurasi pengaturan deret waktu Anda, ada pengaturan lanjutan tambahan. Untuk informasi selengkapnya tentang masing-masing pengaturan lanjutan, lihatKonfigurasi bangunan model lanjutan. Untuk mengkonfigurasi pengaturan lanjutan, lakukan hal berikut:

i. (Opsional) Untuk menu tarik-turun metrik Objective, pilih metrik yang ingin dioptimalkan Canvas saat membuat model Anda. Jika Anda tidak memilih metrik, Canvas memilih satu untuk Anda secara default. Untuk deskripsi metrik yang tersedia, lihat. Referensi metrik

ii. Untuk kuantil Forecast, masukkan hingga 5 nilai kuantil yang dipisahkan koma untuk menentukan batas atas dan bawah perkiraan Anda.

iii. Untuk frekuensi Forecast, tentukan yang berikut ini:

A. Untuk Frekuensi, masukkan Unit dan Nilai untuk menentukan frekuensi untuk nilai yang diperkirakan.

B. Untuk Agregasi, pilih metode agregasi untuk cara Canvas menangani data yang tidak sesuai dengan frekuensi.

iv. Untuk runtime pekerjaan Max, tetapkan nilai jam dan menit untuk jumlah waktu maksimum yang dapat dihabiskan Canvas untuk membangun model Anda. Setelah waktu maksimum, Canvas berhenti membangun dan memilih kandidat model terbaik.

d. Setelah mengonfigurasi pengaturan lanjutan, pilih Simpan.

9. Pilih atau batalkan pilihan kolom dalam data Anda untuk menyertakan atau menjatuhkannya dari build Anda.

> **Note**
>
> Jika Anda membuat prediksi batch dengan model Anda setelah pembuatan, Canvas menambahkan kolom yang dijatuhkan ke hasil prediksi Anda. Namun, Canvas tidak menambahkan kolom yang dijatuhkan ke prediksi batch Anda untuk model deret waktu.

10. (Opsional) Gunakan alat visualisasi dan analitik yang disediakan Canvas untuk memvisualisasikan data Anda dan menentukan fitur mana yang mungkin ingin Anda sertakan

dalam model Anda. Untuk informasi selengkapnya, lihat Menjelajahi dan menganalisis data Anda.

11. (Opsional) Gunakan transformasi data untuk membersihkan, mengubah, dan menyiapkan data Anda untuk pembuatan model. Untuk informasi selengkapnya, lihat Mempersiapkan data Anda dengan transformasi lanjutan. Anda dapat melihat dan menghapus transformasi Anda dengan memilih Resep model untuk membuka panel samping resep Model.
12. (Opsional) Untuk fitur tambahan seperti melihat pratinjau keakuratan model Anda, memvalidasi kumpulan data Anda, dan mengubah ukuran sampel acak yang diambil Canvas dari kumpulan data Anda, lihat. Pratinjau model Anda
13. Setelah meninjau data Anda dan membuat perubahan apa pun pada kumpulan data Anda, pilih Quick build atau Standard build untuk memulai pembuatan model Anda.

Setelah model Anda mulai membangun, Anda dapat meninggalkan halaman. Ketika model ditampilkan sebagai Siap di halaman Model saya, model siap untuk analisis dan prediksi.

Konfigurasi bangunan model lanjutan

Amazon SageMaker Canvas mendukung berbagai pengaturan lanjutan yang dapat Anda konfigurasikan saat membuat model. Halaman berikut mencantumkan semua pengaturan lanjutan bersama dengan informasi tambahan tentang opsi dan konfigurasinya.

> **Note**
>
> Pengaturan lanjutan berikut saat ini hanya didukung untuk jenis model peramalan numerik, kategoris, dan deret waktu.

Pengaturan model prediksi numerik dan kategoris tingkat lanjut

Canvas mendukung pengaturan lanjutan berikut untuk jenis model prediksi numerik dan kategoris.

Metrik obyektif

Metrik objektif adalah metrik yang Anda ingin Canvas optimalkan saat membangun model Anda. Jika Anda tidak memilih metrik, Canvas memilih satu untuk Anda secara default. Untuk deskripsi metrik yang tersedia, lihat. Referensi metrik

Metode pelatihan

Canvas dapat secara otomatis memilih metode pelatihan berdasarkan ukuran dataset, atau Anda dapat memilihnya secara manual. Metode pelatihan berikut tersedia untuk Anda pilih:

- Ensembling — SageMaker memanfaatkan AutoGluon perpustakaan untuk melatih beberapa model dasar. Untuk menemukan kombinasi terbaik untuk kumpulan data Anda, mode ansambel menjalankan 5-10 uji coba dengan model dan pengaturan parameter meta yang berbeda. Kemudian, model ini digabungkan menggunakan metode ansambel susun untuk membuat model prediktif yang optimal. Untuk daftar algoritma yang didukung oleh mode ensemble untuk data tabular, lihat bagian berikut. Algoritma
- Optimasi Hyperparameter (HPO) — SageMaker menemukan versi terbaik dari sebuah model dengan menyetel hyperparameters menggunakan optimasi Bayesian atau optimasi multi-fidelity saat menjalankan pekerjaan pelatihan pada dataset Anda. Mode HPO memilih algoritme yang paling relevan dengan kumpulan data Anda dan memilih rentang hiperparameter terbaik untuk menyetel model Anda. Untuk menyetel model Anda, mode HPO menjalankan hingga 100 uji coba (default) untuk menemukan pengaturan hiperparameter optimal dalam rentang yang dipilih. Jika ukuran dataset Anda kurang dari 100 MB, SageMaker gunakan optimasi Bayesian. SageMaker memilih optimasi multi-fidelity jika dataset Anda lebih besar dari 100 MB.

  Untuk daftar algoritma yang didukung oleh mode HPO untuk data tabular, lihat bagian berikut. Algoritma
- Otomatis — SageMaker secara otomatis memilih mode ensembling atau mode HPO berdasarkan ukuran dataset Anda. Jika dataset Anda lebih besar dari 100 MB, SageMaker pilih mode HPO. Jika tidak, ia memilih mode ansambel.

Algoritma

Dalam mode Ensembling, Canvas mendukung algoritma pembelajaran mesin berikut:

- LightGBM - Kerangka kerja yang dioptimalkan yang menggunakan algoritma berbasis pohon dengan peningkatan gradien. Algoritma ini menggunakan pohon yang tumbuh dalam lebar, bukan kedalaman, dan sangat dioptimalkan untuk kecepatan.
- CatBoost— Kerangka kerja yang menggunakan algoritme berbasis pohon dengan peningkatan gradien. Dioptimalkan untuk menangani variabel kategoris.
- XGBoost — Kerangka kerja yang menggunakan algoritme berbasis pohon dengan peningkatan gradien yang tumbuh secara mendalam, bukan luasnya.

- Random Forest — Algoritma berbasis pohon yang menggunakan beberapa pohon keputusan pada sub-sampel acak data dengan penggantian. Pohon-pohon dibagi menjadi simpul optimal di setiap tingkat. Keputusan setiap pohon dirata-ratakan bersama untuk mencegah overfitting dan meningkatkan prediksi.
- Pohon Ekstra — Algoritma berbasis pohon yang menggunakan beberapa pohon keputusan di seluruh kumpulan data. Pohon-pohon dibelah secara acak di setiap tingkat. Keputusan setiap pohon dirata-ratakan untuk mencegah overfitting dan untuk meningkatkan prediksi. Pohon tambahan menambahkan tingkat pengacakan dibandingkan dengan algoritma hutan acak.
- Model Linear — Kerangka kerja yang menggunakan persamaan linier untuk memodelkan hubungan antara dua variabel dalam data yang diamati.
- Obor jaringan saraf — Model jaringan saraf yang diimplementasikan menggunakan Pytorch.
- Neural network fast.ai — Model jaringan saraf yang diimplementasikan menggunakan fast.ai.

Dalam mode HPO, Canvas mendukung algoritma pembelajaran mesin berikut:

- XGBoost — Algoritma pembelajaran yang diawasi yang mencoba memprediksi variabel target secara akurat dengan menggabungkan ansambel perkiraan dari serangkaian model yang lebih sederhana dan lebih lemah.
- Algoritma pembelajaran mendalam — Perceptron multilayer (MLP) dan jaringan saraf tiruan feedforward. Algoritma ini dapat menangani data yang tidak dapat dipisahkan secara linier.

Pemisahan data

Anda memiliki opsi untuk menentukan bagaimana Anda ingin membagi kumpulan data Anda antara set pelatihan (bagian dari kumpulan data Anda yang digunakan untuk membangun model) dan kumpulan validasi, (bagian dari kumpulan data Anda yang digunakan untuk memverifikasi akurasi model). Misalnya, rasio split umum adalah pelatihan 80% dan validasi 20%, di mana 80% data Anda digunakan untuk membangun model sementara 20% disimpan untuk mengukur kinerja model. Jika Anda tidak menentukan rasio kustom, Canvas membagi dataset Anda secara otomatis.

Kandidat maks

> **Note**
>
> Fitur ini hanya tersedia dalam mode pelatihan HPO.

Anda dapat menentukan jumlah maksimum kandidat model yang dihasilkan Canvas saat membangun model Anda. Kami menyarankan Anda menggunakan jumlah kandidat default, yaitu 100, untuk membangun model yang paling akurat. Jumlah maksimum yang dapat Anda tentukan adalah 250. Mengurangi jumlah kandidat model dapat memengaruhi akurasi model Anda.

Runtime pekerjaan maks

Anda dapat menentukan runtime pekerjaan maksimum, atau jumlah waktu maksimum yang dihabiskan Canvas untuk membangun model Anda. Setelah batas waktu, Canvas berhenti membangun dan memilih kandidat model terbaik.

Waktu maksimum yang dapat Anda tentukan adalah 720 jam. Kami sangat menyarankan agar Anda mempertahankan runtime pekerjaan maksimum lebih dari 30 menit untuk memastikan bahwa Canvas memiliki cukup waktu untuk menghasilkan kandidat model dan menyelesaikan pembuatan model Anda.

Pengaturan model peramalan deret waktu lanjutan

Untuk model peramalan deret waktu, Canvas mendukung metrik berikut yang tercantum di bagian sebelumnya:

- Metrik obyektif
- Algoritma
- Runtime pekerjaan maks

Model peramalan deret waktu juga mendukung pengaturan lanjutan berikut:

Agregasi

Jika Anda mengatur frekuensi peramalan lebih rendah dari frekuensi data yang direkam, Canvas mengumpulkan titik data apa pun yang tidak cocok dengan frekuensi baru. Misalnya, jika Anda memiliki titik data harian tetapi ingin membuat perkiraan mingguan, Anda dapat mengatur frekuensi menjadi mingguan, dan kemudian Canvas menggabungkan semua titik data harian untuk setiap minggu menjadi satu catatan. Agregasi hanya didukung untuk kolom target, dan nilai kolom harus dalam format datetime.

Metode agregasi default adalah dengan `sum` nilai titik data agregat, tetapi Anda juga dapat mengatur metode agregasi ke yang berikut:

- `avg`- Canvas menetapkan nilai catatan ke rata-rata semua titik data agregat.

- `first`- Canvas menetapkan nilai catatan ke nilai pertama dari titik data agregat.
- `min`- Canvas menetapkan nilai catatan ke nilai minimum yang ditemukan di titik data agregat.
- `max`— Canvas menetapkan nilai record ke nilai maksimum yang ditemukan di titik data agregat.

Kuantil Forecast

Untuk peramalan deret waktu, SageMaker latih 6 kandidat model dengan deret waktu target Anda. Kemudian, SageMaker gabungkan model-model ini menggunakan metode ansambel susun untuk membuat model peramalan optimal untuk metrik objektif tertentu. Setiap model peramalan menghasilkan perkiraan probabilistik dengan menghasilkan perkiraan pada kuantil antara P1 dan P99. Kuantil ini digunakan untuk menjelaskan ketidakpastian perkiraan. Secara default, perkiraan dihasilkan untuk 0.1 (`p10`), 0.5 (`p50`), dan 0.9 (`p90`). Anda dapat memilih untuk menentukan hingga lima kuantil Anda sendiri dari 0,01 (`p1`) hingga 0,99 (`p99`), dengan kenaikan 0,01 atau lebih tinggi.

Pratinjau model Anda

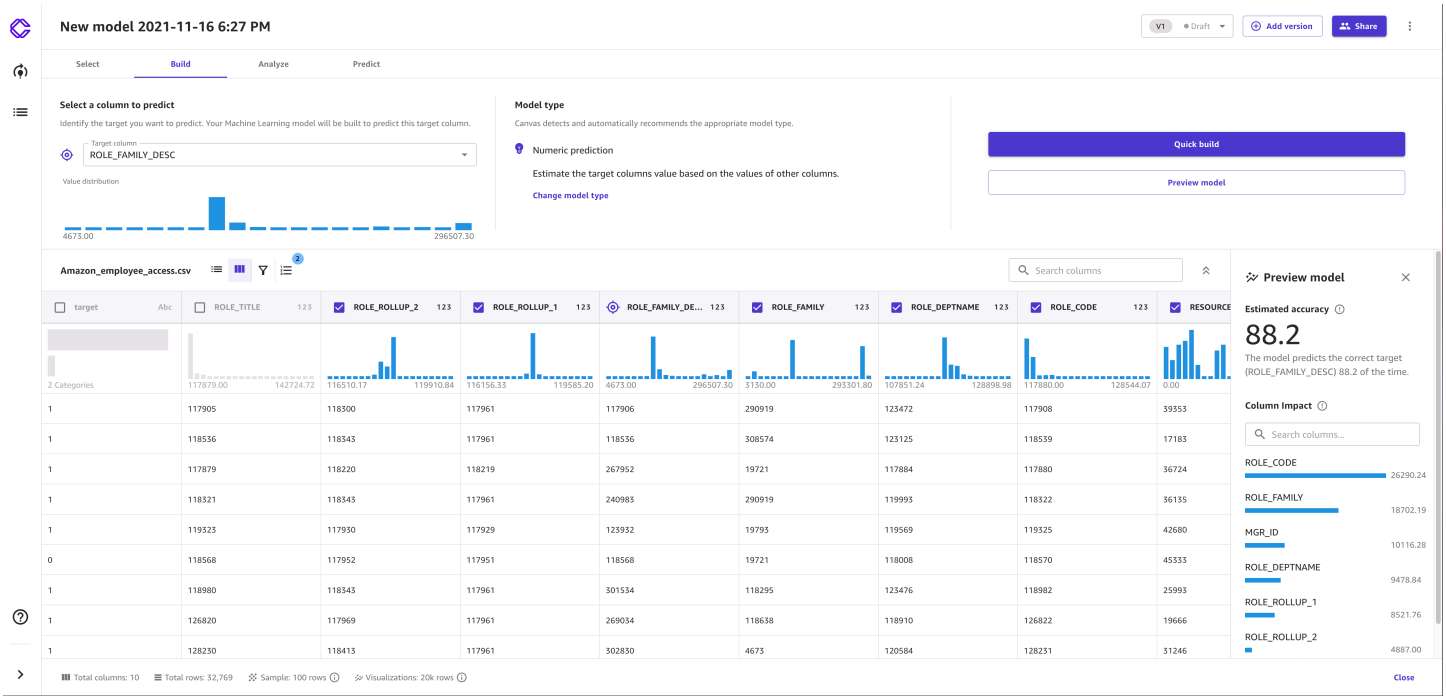
> **Note**
>
> Fungsionalitas berikut hanya tersedia untuk model khusus yang dibuat dengan kumpulan data tabular. Model prediksi teks multi-kategori juga dikecualikan.

SageMaker Canvas memberi Anda alat untuk melihat pratinjau model Anda dan memvalidasi data sebelum Anda mulai membangun. Fungsionalitas berikut termasuk melihat pratinjau keakuratan model Anda, memvalidasi kumpulan data Anda untuk mencegah masalah saat membangun model, dan mengubah ukuran sampel acak untuk model Anda.

Pratinjau model

Dengan Amazon SageMaker Canvas, Anda bisa mendapatkan wawasan dari data sebelum membuat model dengan memilih model Pratinjau. Misalnya, Anda dapat melihat bagaimana data di setiap kolom didistribusikan. Untuk model yang dibuat menggunakan data kategoris, Anda juga dapat memilih model Pratinjau untuk menghasilkan prediksi akurasi Estimasi tentang seberapa baik model dapat menganalisis data Anda. Keakuratan build Cepat atau build Standar menunjukkan seberapa baik kinerja model pada data nyata dan umumnya lebih tinggi daripada akurasi Estimasi.

Amazon SageMaker Canvas secara otomatis menangani nilai yang hilang dalam kumpulan data Anda saat membuat model. Ini menyimpulkan nilai yang hilang dengan menggunakan nilai yang berdekatan yang ada dalam dataset.



Validasi data

Sebelum Anda membuat model, SageMaker Canvas memeriksa kumpulan data Anda untuk masalah yang akan menyebabkan build Anda gagal. Jika SageMaker Canvas menemukan masalah, maka Canvas memperingatkan Anda di halaman Build sebelum Anda mencoba membuat model.

Anda dapat memilih Validasi data untuk melihat daftar masalah dengan kumpulan data Anda. Anda kemudian dapat menggunakan fitur persiapan data SageMaker Canvas, atau alat Anda sendiri, untuk memperbaiki kumpulan data Anda sebelum memulai pembuatan. Jika Anda tidak memperbaiki masalah dengan kumpulan data Anda, build Anda gagal.

Jika Anda membuat perubahan pada kumpulan data untuk memperbaiki masalah, Anda memiliki opsi untuk memvalidasi ulang kumpulan data Anda sebelum mencoba membangun. Kami menyarankan Anda memvalidasi ulang dataset Anda sebelum membangun.

Tabel berikut menunjukkan masalah yang diperiksa SageMaker Canvas dalam kumpulan data Anda dan cara mengatasinya.

Isu	Penyelesaian
Jenis model yang salah untuk data Anda	Coba jenis model lain atau gunakan kumpulan data yang berbeda.

Isu	Penyelesaian
Nilai yang hilang di kolom target Anda	Ganti nilai yang hilang, jatuhkan baris dengan nilai yang hilang, atau gunakan kumpulan data yang berbeda.
Terlalu banyak label unik di kolom target Anda	Verifikasi bahwa Anda telah menggunakan kolom yang benar untuk kolom target Anda, atau gunakan kumpulan data yang berbeda.
Terlalu banyak nilai non-numerik di kolom target Anda	Pilih kolom target yang berbeda, pilih jenis model lain, atau gunakan kumpulan data yang berbeda.
Satu atau beberapa nama kolom berisi garis bawah ganda	Ganti nama kolom untuk menghapus garis bawah ganda, dan coba lagi.
Tak satu pun dari baris dalam dataset Anda yang lengkap	Ganti nilai yang hilang, atau gunakan kumpulan data yang berbeda.
Terlalu banyak label unik untuk jumlah baris dalam data Anda	Periksa apakah Anda menggunakan kolom target kanan, menambah jumlah baris dalam kumpulan data Anda, mengkonsolidasikan label serupa, atau menggunakan kumpulan data yang berbeda.

Sampel acak

SageMaker Canvas menggunakan metode pengambilan sampel acak untuk mengambil sampel kumpulan data Anda. Metode sampel acak berarti bahwa setiap baris memiliki kesempatan yang sama untuk dipilih untuk sampel. Anda dapat memilih kolom di pratinjau untuk mendapatkan statistik ringkasan untuk sampel acak, seperti mean dan mode.

Secara default, SageMaker Canvas menggunakan ukuran sampel acak 20.000 baris dari kumpulan data Anda untuk kumpulan data dengan lebih dari 20.000 baris. Untuk kumpulan data yang lebih kecil dari 20.000 baris, ukuran sampel default adalah jumlah baris dalam kumpulan data Anda. Anda dapat menambah atau mengurangi ukuran sampel dengan memilih Sampel acak di tab Build aplikasi SageMaker Canvas. Anda dapat menggunakan slider untuk memilih ukuran sampel yang

Anda inginkan, dan kemudian memilih Perbarui untuk mengubah ukuran sampel. Ukuran sampel maksimum yang dapat Anda pilih untuk kumpulan data adalah 40.000 baris, dan ukuran sampel minimum adalah 500 baris. Jika Anda memilih ukuran sampel yang besar, pratinjau kumpulan data dan statistik ringkasan mungkin memerlukan beberapa saat untuk dimuat ulang.

Halaman Build menampilkan pratinjau 100 baris dari kumpulan data Anda. Jika ukuran sampel adalah ukuran yang sama dengan dataset Anda, maka pratinjau menggunakan 100 baris pertama dari dataset Anda. Jika tidak, pratinjau menggunakan 100 baris pertama dari sampel acak.

Mengedit kumpulan data gambar

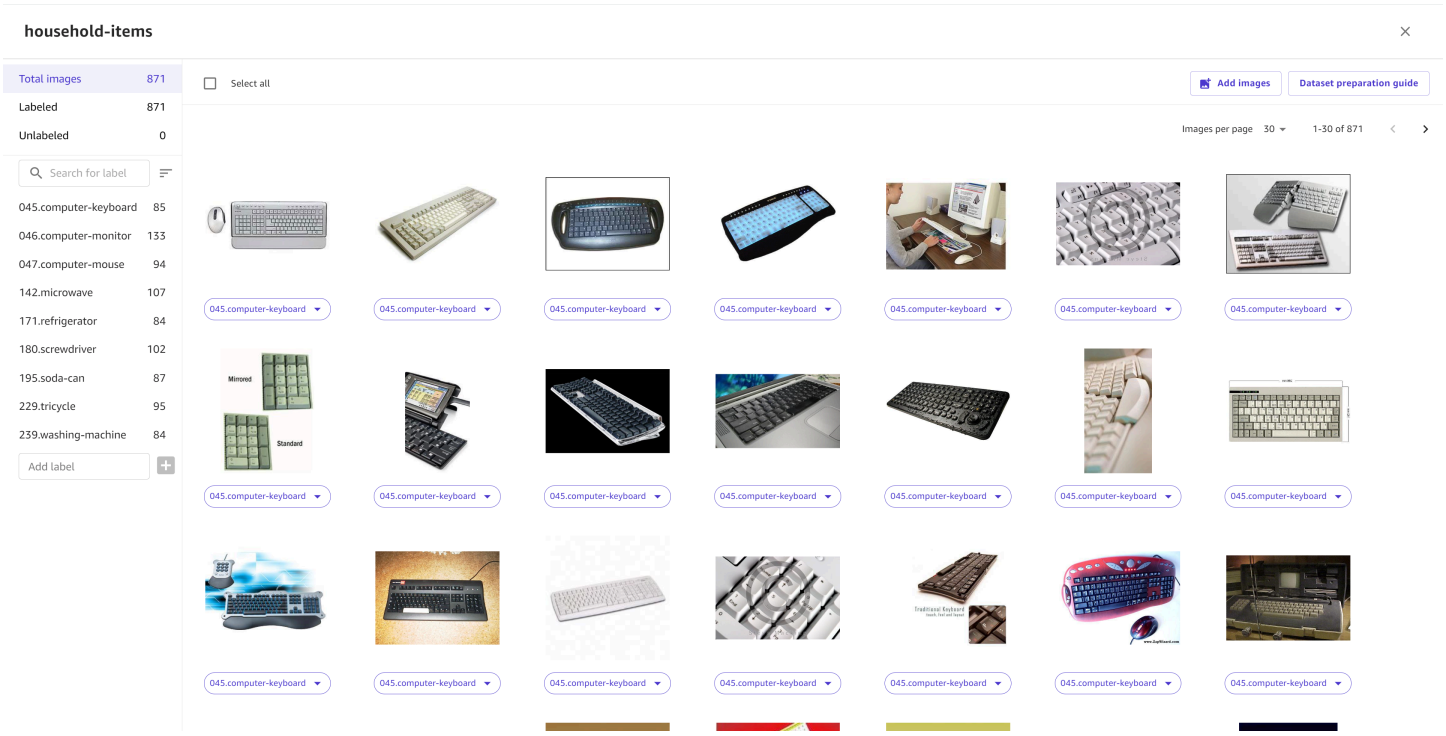
Di Amazon SageMaker Canvas, Anda dapat mengedit kumpulan data gambar dan meninjau label Anda sebelum membuat model. Anda mungkin ingin melakukan tugas seperti menetapkan label ke gambar yang tidak berlabel atau menambahkan lebih banyak gambar ke kumpulan data. Tugas-tugas ini semua dapat dilakukan dalam aplikasi Canvas, memberi Anda satu tempat untuk memodifikasi dataset Anda dan membangun model.

> **Note**
>
> Sebelum membuat model, Anda harus menetapkan label ke semua gambar di kumpulan data Anda. Selain itu, Anda harus memiliki setidaknya 25 gambar per label dan minimal dua label. Untuk informasi selengkapnya tentang menetapkan label, lihat bagian di halaman ini yang disebut Tetapkan label ke gambar yang tidak berlabel. Jika Anda tidak dapat menentukan label untuk gambar, Anda harus menghapusnya dari kumpulan data Anda. Untuk informasi selengkapnya tentang menghapus gambar, lihat bagian di halaman Menambahkan atau menghapus gambar dari dataset ini.

Untuk mulai mengedit kumpulan data gambar, Anda harus berada di tab Build saat membuat model prediksi gambar label tunggal.

Halaman baru terbuka yang menunjukkan gambar dalam kumpulan data Anda bersama dengan labelnya. Halaman ini mengkategorikan kumpulan data gambar Anda ke dalam gambar Total, Gambar berlabel, dan gambar Tidak Berlabel. Anda juga dapat meninjau panduan persiapan Dataset untuk praktik terbaik dalam membangun model prediksi gambar yang lebih akurat.

Tangkapan layar berikut menunjukkan halaman untuk mengedit kumpulan data gambar Anda.



Dari halaman ini, Anda dapat melakukan tindakan berikut.

Lihat properti untuk setiap gambar (label, ukuran, dimensi)

Untuk melihat gambar individual, Anda dapat mencarinya dengan nama file di bilah pencarian. Kemudian, pilih gambar untuk membuka tampilan penuh. Anda dapat melihat properti gambar dan menetapkan kembali label gambar. Pilih Simpan saat Anda melihat gambar.

Menambahkan, mengganti nama, atau menghapus label dalam kumpulan data

Canvas mencantumkan label untuk kumpulan data Anda di panel navigasi kiri. Anda dapat menambahkan label baru ke kumpulan data dengan memasukkan label di bidang Tambahkan teks label.

Untuk mengganti nama atau menghapus label dari kumpulan data Anda, pilih ikon Opsi lainnya (⋮) di sebelah label dan pilih Ganti nama atau Hapus. Jika Anda mengganti nama label, Anda dapat memasukkan nama label baru dan memilih Konfirmasi. Jika Anda menghapus label, label akan dihapus dari semua gambar dalam kumpulan data Anda yang memiliki label tersebut. Gambar apa pun dengan label itu tidak akan diberi label.

Tetapkan label ke gambar yang tidak berlabel

Untuk melihat gambar yang tidak berlabel dalam kumpulan data Anda, pilih Tidak Berlabel di panel navigasi kiri. Untuk setiap gambar, pilih dan buka label berjudul Unlabeled dan pilih label untuk ditetapkan ke gambar dari daftar dropdown. Anda juga dapat memilih lebih dari satu gambar dan melakukan tindakan ini, dan semua gambar yang dipilih diberi label yang Anda pilih.

Tetapkan kembali label ke gambar

Anda dapat menetapkan ulang label ke gambar dengan memilih gambar (atau beberapa gambar sekaligus) dan membuka dropdown berjudul dengan label saat ini. Pilih label yang Anda inginkan, dan gambar atau gambar diperbarui dengan label baru.

Urutkan gambar Anda berdasarkan label

Anda dapat melihat semua gambar untuk label tertentu dengan memilih label di panel navigasi kiri.

Menambahkan atau menghapus gambar dari dataset

Anda dapat menambahkan lebih banyak gambar ke kumpulan data Anda dengan memilih Tambahkan gambar di panel navigasi atas. Anda akan dibawa melalui alur kerja untuk mengimpor lebih banyak gambar. Gambar yang Anda impor ditambahkan ke kumpulan data yang ada.

Anda dapat menghapus gambar dari kumpulan data Anda dengan memilihnya dan kemudian memilih Hapus di panel navigasi atas.

> **Note**
>
> Setelah membuat perubahan apa pun pada kumpulan data Anda, pilih Simpan kumpulan data untuk memastikan bahwa Anda tidak kehilangan perubahan.

Jelajahi dan analisis data Anda

> **Note**
>
> Anda hanya dapat menggunakan visualisasi dan analitik SageMaker Canvas untuk model yang dibangun di atas kumpulan data tabular. Model prediksi teks multi-kategori juga dikecualikan.

Di Amazon SageMaker Canvas, Anda dapat menjelajahi variabel dalam kumpulan data menggunakan visualisasi dan analitik. SageMaker Canvas memberi Anda kemampuan untuk membuat visualisasi dan analitik dalam aplikasi. Anda dapat menggunakan eksplorasi ini untuk mengungkap hubungan antara variabel Anda sebelum membangun model Anda.

Untuk informasi lebih lanjut tentang teknik visualisasi di Canvas, lihatJelajahi data Anda menggunakan teknik visualisasi.

Untuk informasi selengkapnya tentang analitik di Canvas, lihatJelajahi data Anda menggunakan analitik.

Jelajahi data Anda menggunakan teknik visualisasi

> **Note**
>
> Anda hanya dapat menggunakan visualisasi SageMaker Canvas untuk model yang dibangun di atas kumpulan data tabel. Model prediksi teks multi-kategori juga dikecualikan.

Dengan Amazon SageMaker Canvas, Anda dapat menjelajahi dan memvisualisasikan data Anda untuk mendapatkan wawasan lanjutan tentang data Anda sebelum membuat model ML Anda. Anda dapat memvisualisasikan menggunakan plot sebar, diagram batang, dan plot kotak, yang dapat membantu Anda memahami data Anda dan menemukan hubungan antara fitur yang dapat memengaruhi akurasi model.

Di tab Build aplikasi SageMaker Canvas, pilih Visualizer data untuk mulai membuat visualisasi Anda.

Anda dapat mengubah ukuran sampel visualisasi untuk menyesuaikan ukuran sampel acak yang diambil dari kumpulan data Anda. Ukuran sampel yang terlalu besar dapat memengaruhi kinerja visualisasi data Anda, jadi sebaiknya Anda memilih ukuran sampel yang sesuai. Untuk mengubah ukuran sampel, gunakan prosedur berikut.

1. Pilih sampel Visualisasi.
2. Gunakan slider untuk memilih ukuran sampel yang Anda inginkan.
3. Pilih Perbarui untuk mengonfirmasi perubahan pada ukuran sampel Anda.

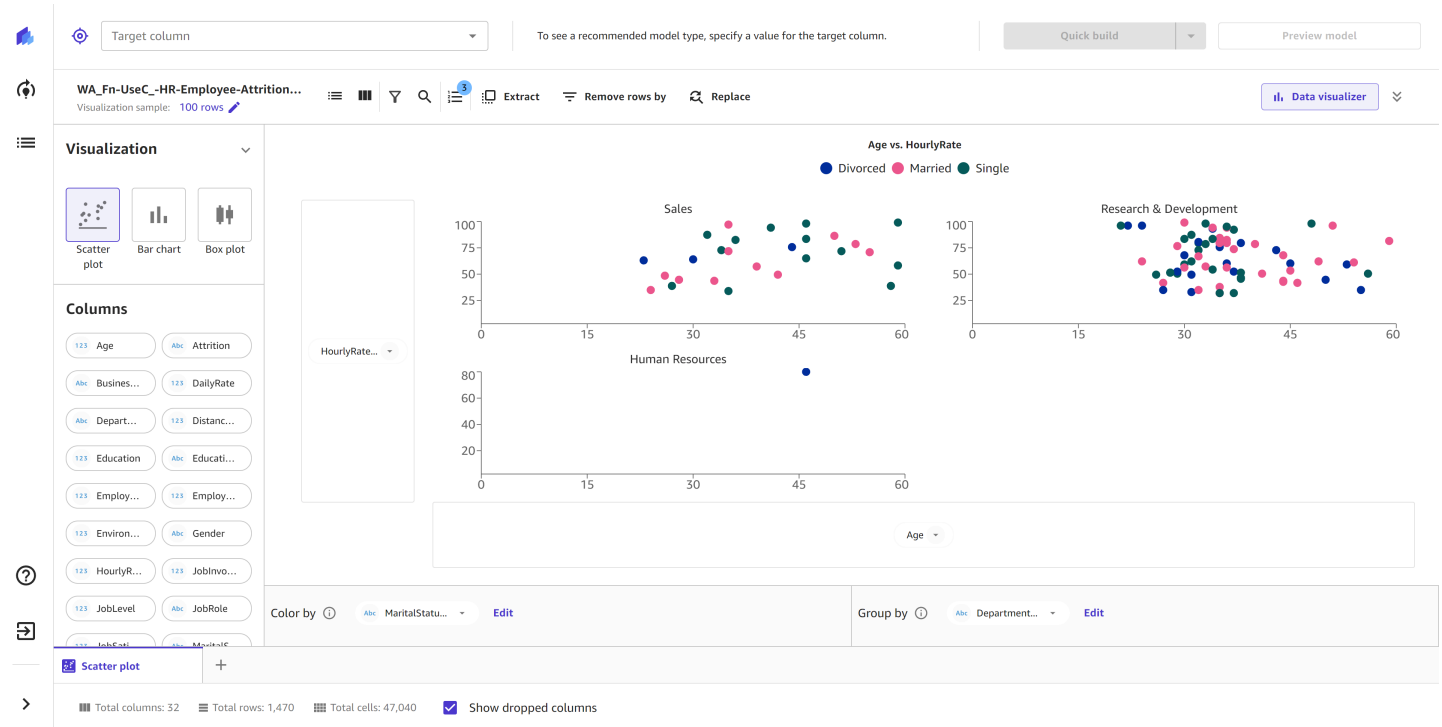
> **Note**
>
> Teknik visualisasi tertentu memerlukan kolom tipe data tertentu. Misalnya, Anda hanya dapat menggunakan kolom numerik untuk sumbu x dan y dari plot pencar.

Plot pencar

Untuk membuat plot pencar dengan dataset Anda, pilih plot Scatter di panel Visualisasi. Kemudian, Anda dapat memilih fitur yang ingin Anda plot pada sumbu x dan y dari bagian Kolom. Anda dapat menarik dan melepas kolom ke sumbu, atau setelah sumbu dijatuhkan, Anda dapat memilih kolom dari daftar kolom yang didukung.

Anda dapat menggunakan Color by untuk mewarnai titik data pada plot dengan fitur ketiga. Anda juga dapat menggunakan Group by untuk mengelompokkan data ke dalam plot terpisah berdasarkan fitur keempat.

Gambar berikut menunjukkan plot pencar yang menggunakan Color by dan Group by. Dalam contoh ini, setiap titik data diwarnai oleh `MaritalStatus` fitur, dan pengelompokan berdasarkan `Department` fitur menghasilkan plot pencar untuk titik data masing-masing departemen.

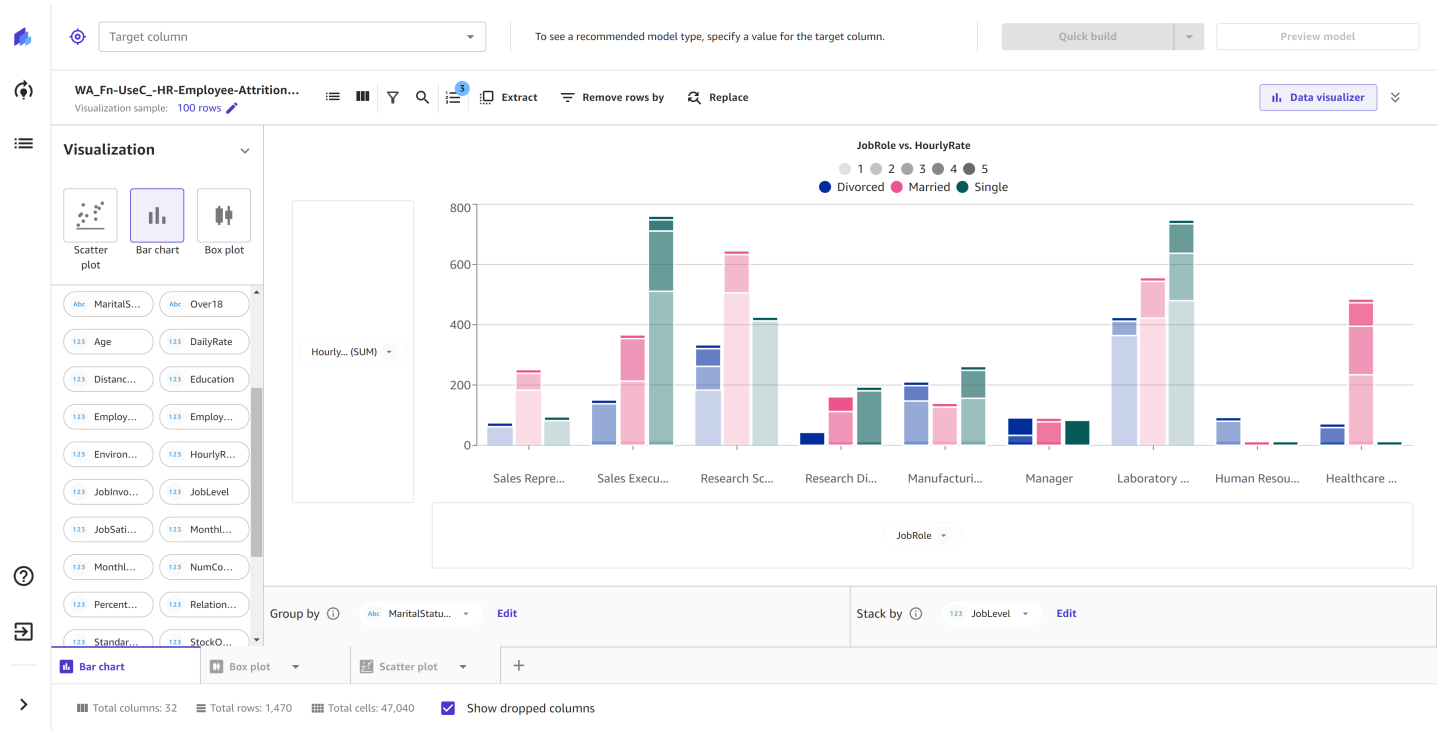


Bagan batang

Untuk membuat diagram batang dengan kumpulan data Anda, pilih Bagan batang di panel Visualisasi. Kemudian, Anda dapat memilih fitur yang ingin Anda plot pada sumbu x dan y dari bagian Kolom. Anda dapat menarik dan melepas kolom ke sumbu, atau setelah sumbu dijatuhkan, Anda dapat memilih kolom dari daftar kolom yang didukung.

Anda dapat menggunakan Group by untuk mengelompokkan diagram batang dengan fitur ketiga. Anda dapat menggunakan Stack by untuk menaungi setiap bilah secara vertikal berdasarkan nilai unik dari fitur keempat.

Gambar berikut menunjukkan diagram batang yang menggunakan Group by dan Stack by. Dalam contoh ini, diagram batang dikelompokkan berdasarkan `MaritalStatus` fitur dan ditumpuk oleh fitur tersebut. `JobLevel` Untuk masing-masing `JobRole` pada sumbu x, ada bilah terpisah untuk kategori unik dalam `MaritalStatus` fitur, dan setiap bilah ditumpuk secara vertikal oleh fitur tersebut`JobLevel`.

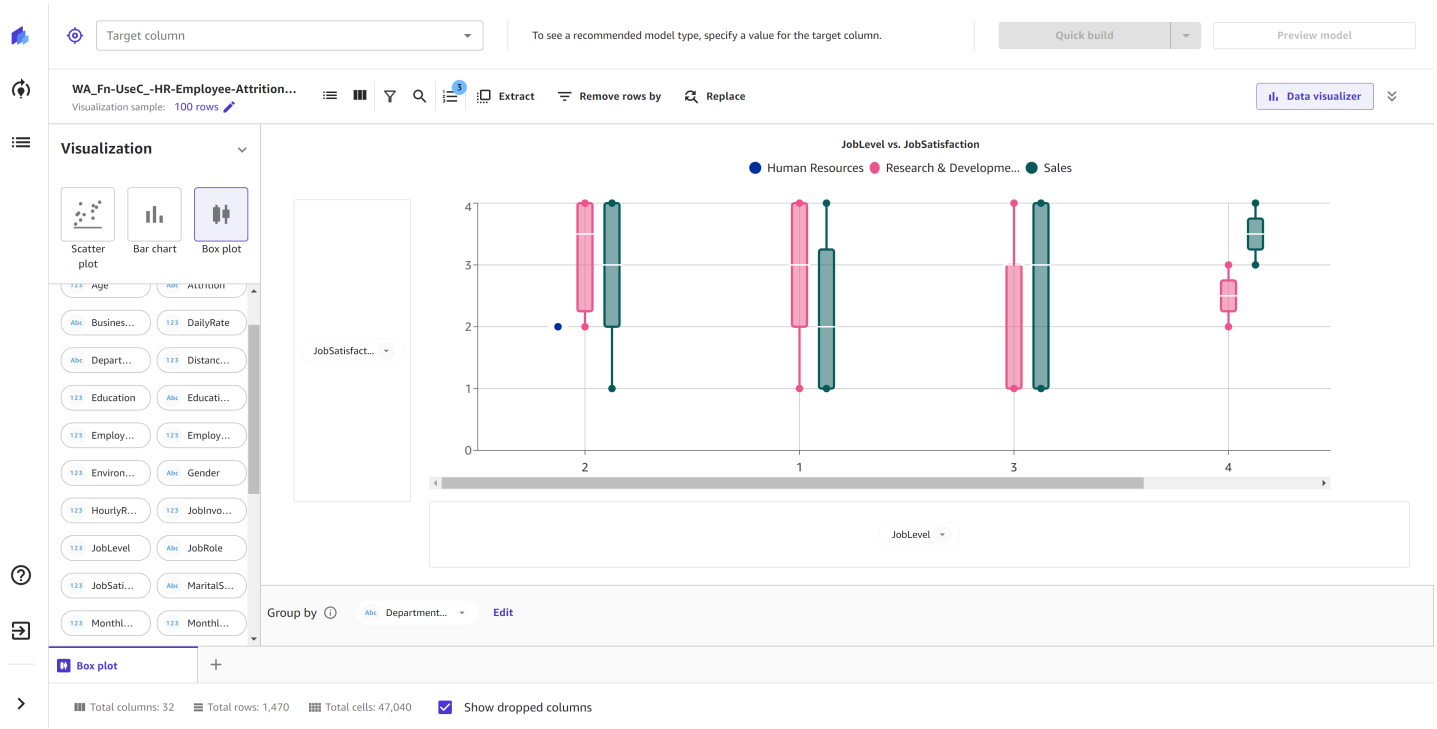


Plot kotak

Untuk membuat plot kotak dengan dataset Anda, pilih Plot kotak di panel Visualisasi. Kemudian, Anda dapat memilih fitur yang ingin Anda plot pada sumbu x dan y dari bagian Kolom. Anda dapat menarik dan melepas kolom ke sumbu, atau setelah sumbu dijatuhkan, Anda dapat memilih kolom dari daftar kolom yang didukung.

Anda dapat menggunakan Group by untuk mengelompokkan plot kotak dengan fitur ketiga.

Gambar berikut menunjukkan plot kotak yang menggunakan Group by. Dalam contoh ini, sumbu x dan y menunjukkan `JobLevel` dan`JobSatisfaction`, masing-masing, dan plot kotak berwarna dikelompokkan berdasarkan fitur. `Department`



Jelajahi data Anda menggunakan analitik

> **Note**
>
> Anda hanya dapat menggunakan analisis SageMaker Canvas untuk model yang dibangun di atas kumpulan data tabular. Model prediksi teks multi-kategori juga dikecualikan.

Dengan analitik di Amazon SageMaker Canvas, Anda dapat menjelajahi kumpulan data dan mendapatkan wawasan tentang semua variabel sebelum membuat model. Anda dapat menentukan hubungan antar fitur dalam kumpulan data Anda menggunakan matriks korelasi. Anda dapat menggunakan teknik ini untuk meringkas kumpulan data Anda ke dalam matriks yang menunjukkan korelasi antara dua atau lebih nilai. Ini membantu Anda mengidentifikasi dan memvisualisasikan pola dalam kumpulan data tertentu untuk analisis data lanjutan.

Matriks menunjukkan korelasi antara setiap fitur sebagai positif, negatif, atau netral. Anda mungkin ingin menyertakan fitur yang memiliki korelasi tinggi satu sama lain saat membangun model Anda.

Fitur yang memiliki sedikit atau tanpa korelasi mungkin tidak relevan dengan model Anda, dan Anda dapat menghapus fitur tersebut saat membuat model Anda.

Untuk memulai dengan matriks korelasi di SageMaker Canvas, lihat bagian berikut.

Buat matriks korelasi

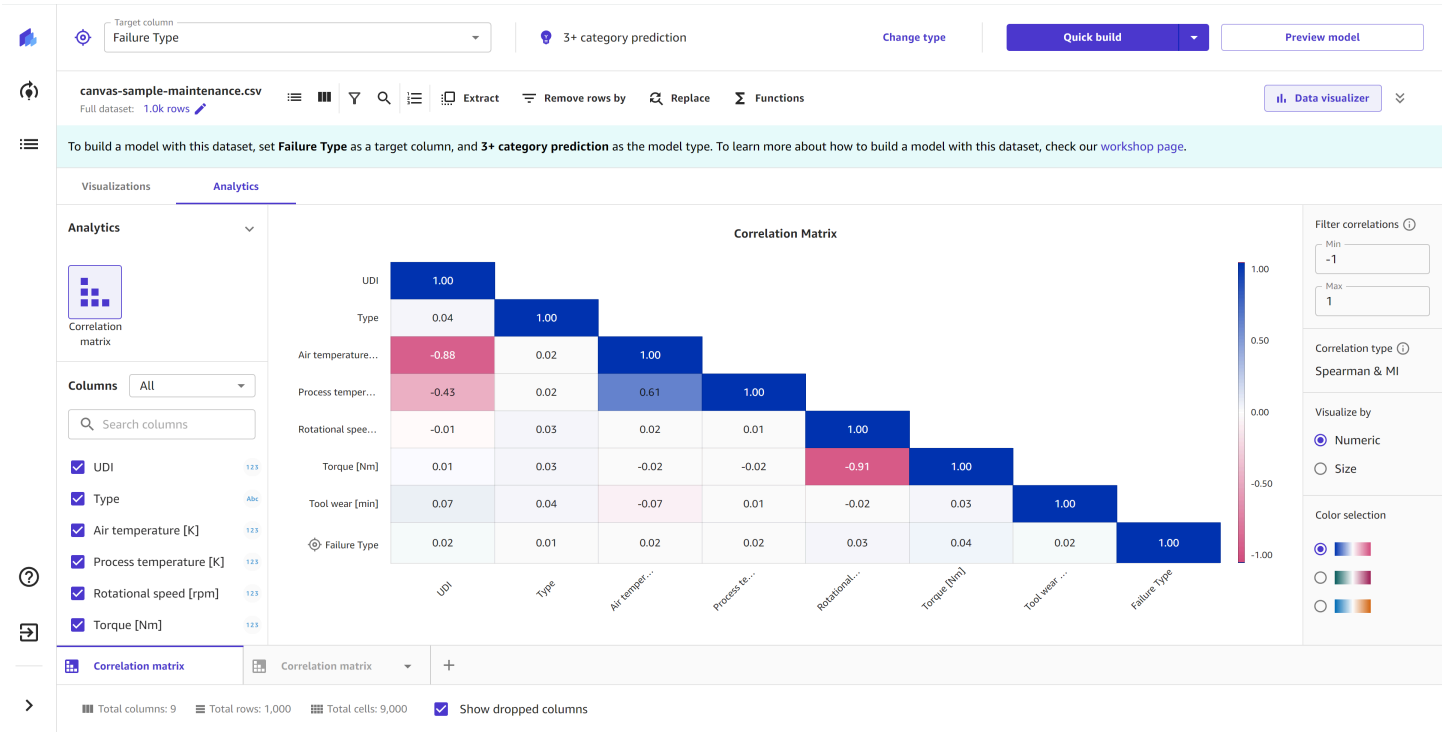
Anda dapat membuat matriks korelasi saat Anda bersiap untuk membangun model di tab Build aplikasi SageMaker Canvas.

Untuk petunjuk tentang cara mulai membuat model, lihatMembangun model.

Setelah Anda mulai menyiapkan model dalam aplikasi SageMaker Canvas, lakukan hal berikut:

1. Di tab Build, pilih Visualizer data.
2. Pilih Analytics.
3. Pilih matriks korelasi.

Anda akan melihat visualisasi yang mirip dengan tangkapan layar berikut, yang menampilkan hingga 15 kolom kumpulan data yang disusun ke dalam matriks korelasi.



Setelah Anda membuat matriks korelasi, Anda dapat menyesuaikannya dengan melakukan hal berikut:

1. Pilih kolom Anda

Untuk Kolom, Anda dapat memilih kolom yang ingin Anda sertakan dalam matriks. Anda dapat membandingkan hingga 15 kolom dari kumpulan data Anda.

> **Note**
>
> Anda dapat menggunakan tipe kolom numerik, kategoris, atau biner untuk matriks korelasi. Matriks korelasi tidak mendukung datetime atau tipe kolom data teks.

Untuk menambah atau menghapus kolom dari matriks korelasi, pilih dan batal pilihan kolom dari panel Kolom. Anda juga dapat menarik dan melepas kolom dari panel langsung ke matriks. Jika kumpulan data Anda memiliki banyak kolom, Anda dapat mencari kolom yang Anda inginkan di bilah kolom Pencarian.

Untuk memfilter kolom berdasarkan tipe data, pilih menu tarik-turun dan pilih Semua, Numerik, atau Kategori. Memilih Semua menampilkan semua kolom dari kumpulan data Anda, sedangkan filter Numerik dan Kategoris hanya menampilkan kolom numerik atau kategoris dalam kumpulan data Anda. Perhatikan bahwa jenis kolom biner disertakan dalam filter numerik atau kategoris.

Untuk wawasan data terbaik, sertakan kolom target Anda dalam matriks korelasi. Saat Anda memasukkan kolom target Anda dalam matriks korelasi, itu muncul sebagai fitur terakhir pada matriks dengan simbol target.

2. Pilih jenis korelasi Anda

SageMaker Canvas mendukung berbagai jenis korelasi, atau metode untuk menghitung korelasi antara kolom Anda.

Untuk mengubah jenis korelasi, gunakan filter Kolom yang disebutkan di bagian sebelumnya untuk memfilter jenis kolom dan kolom yang Anda inginkan. Anda akan melihat tipe Korelasi di panel samping. Untuk perbandingan numerik, Anda memiliki opsi untuk memilih Pearson atau Spearman. Untuk perbandingan kategoris, tipe korelasi ditetapkan sebagai MI. Untuk perbandingan kategoris dan campuran, tipe korelasi ditetapkan sebagai Spearman & MI.

Untuk matriks yang hanya membandingkan kolom numerik, jenis korelasinya adalah Pearson atau Spearman. Ukuran Pearson mengevaluasi hubungan linier antara dua variabel kontinu. Ukuran Spearman mengevaluasi hubungan monotonik antara dua variabel. Untuk Pearson dan Spearman, skala korelasi berkisar antara -1 hingga 1, dengan kedua ujung skala menunjukkan

korelasi sempurna (hubungan langsung 1:1) dan 0 menunjukkan tidak ada korelasi. Anda mungkin ingin memilih Pearson jika data Anda memiliki hubungan yang lebih linier (seperti yang diungkapkan oleh visualisasi plot pencar). Jika data Anda tidak linier, atau berisi campuran hubungan linier dan monotonik, maka Anda mungkin ingin memilih Spearman.

Untuk matriks yang hanya membandingkan kolom kategoris, jenis korelasi diatur ke Klasifikasi Informasi Mutual (MI). Nilai MI adalah ukuran ketergantungan timbal balik antara dua variabel acak. Ukuran MI berada pada skala 0 hingga 1, dengan 0 menunjukkan tidak ada korelasi dan 1 menunjukkan korelasi sempurna.

Untuk matriks yang membandingkan campuran kolom numerik dan kategoris, tipe korelasi Spearman & MI adalah kombinasi dari jenis korelasi Spearman dan MI. Untuk korelasi antara dua kolom numerik, matriks menunjukkan nilai Spearman. Untuk korelasi antara kolom numerik dan kategoris atau dua kolom kategoris, matriks menunjukkan nilai MI.

Terakhir, ingatlah bahwa korelasi tidak selalu menunjukkan sebab-akibat. Nilai korelasi yang kuat hanya menunjukkan bahwa ada hubungan antara dua variabel, tetapi variabel tersebut mungkin tidak memiliki hubungan sebab akibat. Tinjau kolom yang Anda minati dengan cermat untuk menghindari bias saat membangun model Anda.

3. Filter korelasi Anda

Di panel samping, Anda dapat menggunakan Filter korelasi fitur untuk memfilter rentang nilai korelasi yang ingin Anda sertakan dalam matriks. Misalnya, jika Anda ingin memfilter fitur yang hanya memiliki korelasi positif atau netral, Anda dapat mengatur Min ke 0 dan Maks ke 1 (nilai yang valid adalah -1 hingga 1).

Untuk perbandingan Spearman dan Pearson, Anda dapat mengatur rentang korelasi Filter di mana saja dari -1 hingga 1, dengan 0 yang berarti tidak ada korelasi. -1 dan 1 berarti bahwa variabel memiliki korelasi negatif atau positif yang kuat, masing-masing.

Untuk perbandingan MI, rentang korelasi hanya berkisar dari 0 ke 1, dengan 0 berarti tidak ada korelasi dan 1 berarti bahwa variabel memiliki korelasi yang kuat, baik positif maupun negatif.

Setiap fitur memiliki korelasi sempurna (1) dengan dirinya sendiri. Oleh karena itu, Anda mungkin memperhatikan bahwa baris atas matriks korelasi selalu 1. Jika Anda ingin mengecualikan nilai-nilai ini, Anda dapat menggunakan filter untuk mengatur Max kurang dari 1.

Perlu diingat bahwa jika matriks Anda membandingkan campuran kolom numerik dan kategoris dan menggunakan jenis korelasi Spearman & MI, maka korelasi kategoris x numerik dan kategoris

x kategoris (yang menggunakan ukuran MI) berada pada skala 0 hingga 1, sedangkan korelasi numerik x numerik (yang menggunakan ukuran Spearman) berada pada skala -1 hingga 1. Tinjau korelasi minat Anda dengan cermat untuk memastikan bahwa Anda mengetahui jenis korelasi yang digunakan untuk menghitung setiap nilai.

4. Pilih metode visualisasi

Di panel samping, Anda dapat menggunakan Visualize by untuk mengubah metode visualisasi matriks. Pilih metode visualisasi Numerik untuk menunjukkan nilai korelasi (Pearson, Spearman, atau MI), sedangkan memilih metode visualisasi Ukuran memvisualisasikan korelasi dengan titik-titik berukuran dan berwarna yang berbeda. Jika Anda memilih Ukuran, Anda dapat mengarahkan kursor ke titik tertentu pada matriks untuk melihat nilai korelasi yang sebenarnya.

5. Pilih palet warna

Di panel samping, Anda dapat menggunakan Pemilihan warna untuk mengubah palet warna yang digunakan untuk skala korelasi negatif ke positif dalam matriks. Pilih salah satu palet warna alternatif untuk mengubah warna yang digunakan dalam matriks.

Siapkan data dengan transformasi lanjutan

> **Note**
>
> Anda hanya dapat menggunakan transformasi lanjutan untuk model yang dibangun di atas kumpulan data tabel. Model prediksi teks multi-kategori juga dikecualikan.

Dataset pembelajaran mesin Anda mungkin memerlukan persiapan data sebelum Anda membuat model Anda. Anda mungkin ingin membersihkan data Anda karena berbagai masalah, yang mungkin termasuk nilai atau outlier yang hilang, dan melakukan rekayasa fitur untuk meningkatkan akurasi model Anda. Amazon SageMaker Canvas menyediakan transformasi data ML yang dapat digunakan untuk membersihkan, mengubah, dan menyiapkan data untuk pembuatan model. Anda dapat menggunakan transformasi ini pada kumpulan data Anda tanpa kode apa pun. SageMaker Canvas menambahkan transformasi yang Anda gunakan ke resep Model, yang merupakan catatan persiapan data yang dilakukan pada data Anda sebelum membuat model. Transformasi data apa pun yang Anda gunakan hanya memodifikasi data input untuk pembuatan model dan tidak memodifikasi sumber data asli Anda.

Transformasi berikut tersedia di SageMaker Canvas bagi Anda untuk mempersiapkan data Anda untuk membangun.

> **Note**
>
> Pratinjau kumpulan data Anda menunjukkan 100 baris pertama dari kumpulan data. Jika kumpulan data Anda memiliki lebih dari 20.000 baris, Canvas mengambil sampel acak 20.000 baris dan mempratinjau 100 baris pertama dari sampel tersebut. Anda hanya dapat mencari dan menentukan nilai dari baris yang dipratinjau, dan fungsionalitas filter hanya menyaring baris yang dipratinjau dan bukan seluruh kumpulan data.

Jatuhkan kolom

Anda dapat mengecualikan kolom dari build model Anda dengan menjatuhkannya di tab Build aplikasi SageMaker Canvas. Hapus pilihan kolom yang ingin Anda jatuhkan, dan tidak disertakan saat membuat model.

> **Note**
>
> Jika Anda menjatuhkan kolom dan kemudian membuat prediksi batch dengan model Anda, SageMaker Canvas menambahkan kolom yang dijatuhkan kembali ke kumpulan data ouput yang tersedia untuk Anda unduh. Namun, SageMaker Canvas tidak menambahkan kolom yang dijatuhkan kembali untuk model deret waktu.

Filter baris

Fungsionalitas filter menyaring baris yang dipratinjau (100 baris pertama dari kumpulan data Anda) sesuai dengan kondisi yang Anda tentukan. Baris pemfilteran membuat pratinjau sementara data dan tidak memengaruhi pembuatan model. Anda dapat memfilter untuk melihat pratinjau baris yang memiliki nilai yang hilang, berisi outlier, atau memenuhi kondisi khusus di kolom yang Anda pilih.

Filter baris dengan nilai yang hilang

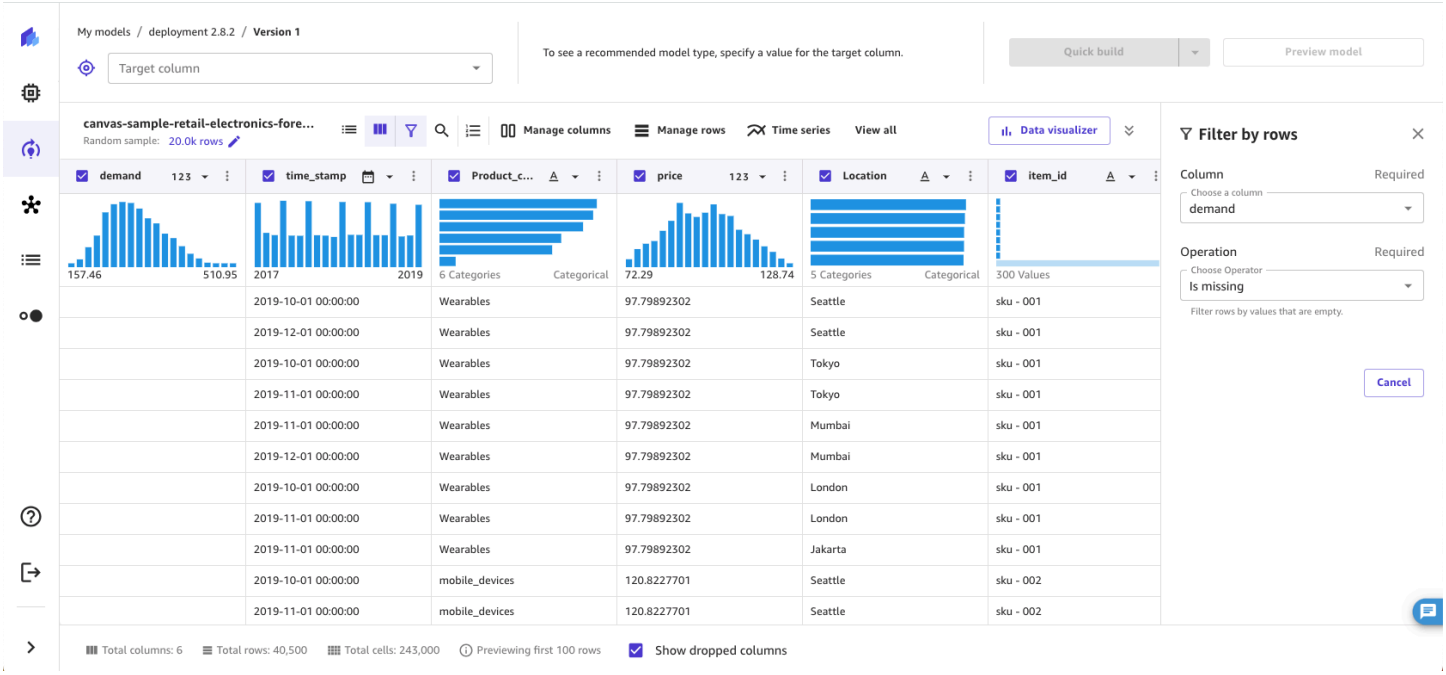
Nilai yang hilang adalah kejadian umum dalam kumpulan data pembelajaran mesin. Jika Anda memiliki baris dengan nilai nol atau kosong di kolom tertentu, Anda mungkin ingin memfilter dan melihat pratinjau baris tersebut.

Untuk memfilter nilai yang hilang dari data pratinjau Anda, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Filter by rows
   ( ).

2. Pilih Kolom yang ingin Anda periksa untuk nilai yang hilang.
3. Untuk Operasi, pilih Tidak ada.

SageMaker Filter kanvas untuk baris yang berisi nilai yang hilang di Kolom yang Anda pilih dan memberikan pratinjau baris yang difilter.



Filter baris dengan outlier

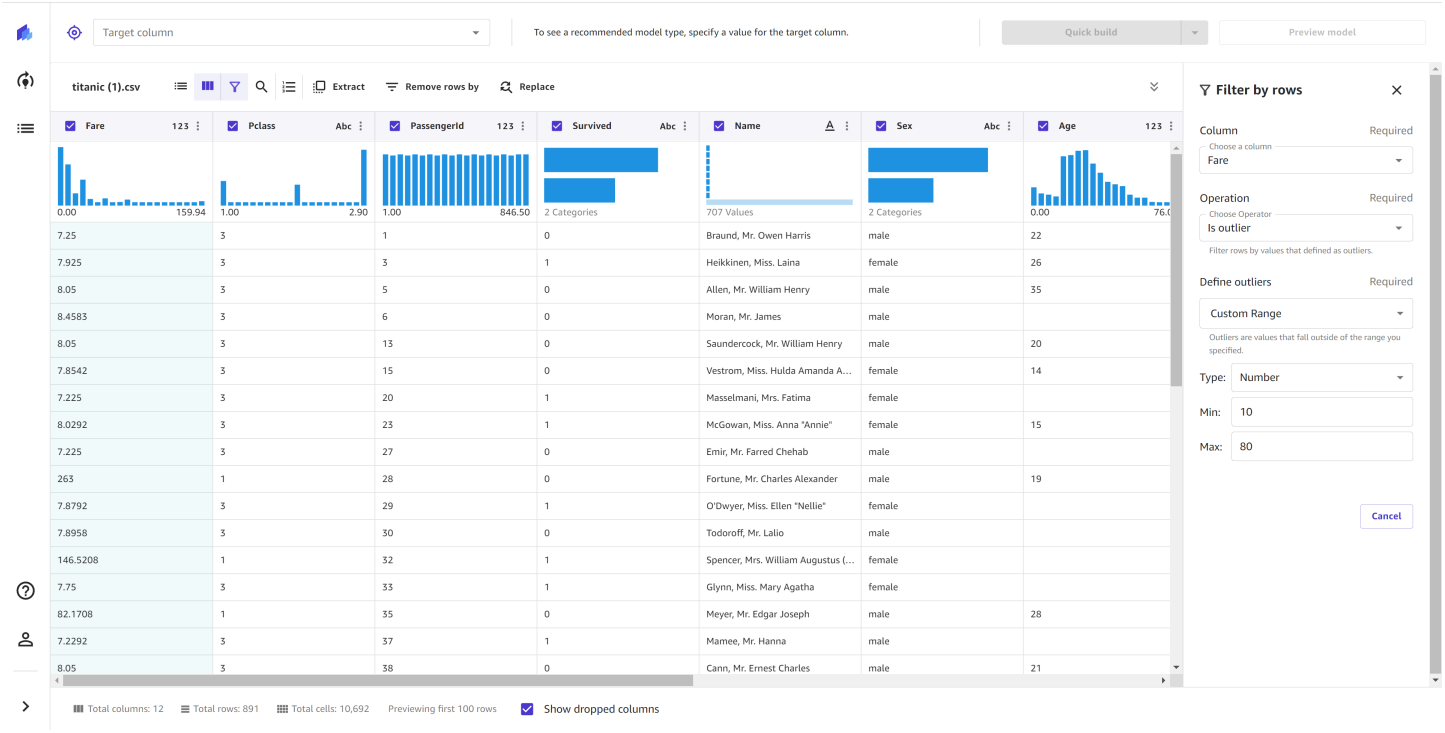
Outlier, atau nilai langka dalam distribusi dan jangkauan data Anda, dapat berdampak negatif pada akurasi model dan menyebabkan waktu pembuatan yang lebih lama. SageMaker Canvas memungkinkan Anda mendeteksi dan memfilter baris yang berisi outlier di kolom numerik. Anda dapat memilih untuk menentukan outlier dengan standar deviasi atau rentang kustom.

Untuk memfilter outlier dalam data Anda, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Filter by rows (▽).
2. Pilih Kolom yang ingin Anda periksa untuk outlier.
3. Untuk Operasi, pilih Is outlier.
4. Atur rentang Outlier ke Deviasi standar atau rentang Kustom.
5. Jika Anda memilih Standar deviasi, tentukan nilai SD (standar deviasi) dari 1-3. Jika Anda memilih Rentang kustom, pilih Persentil atau Angka, lalu tentukan nilai Min dan Maks.

Opsi standar deviasi mendeteksi dan menyaring outlier di kolom numerik menggunakan mean dan standar deviasi. Anda menentukan jumlah standar deviasi suatu nilai harus bervariasi dari rata-rata untuk dianggap sebagai outlier. Misalnya, jika Anda menentukan 3 untuk SD, nilai harus jatuh lebih dari 3 standar deviasi dari mean untuk dianggap sebagai outlier.

Opsi rentang Kustom mendeteksi dan memfilter outlier di kolom numerik menggunakan nilai minimum dan maksimum. Gunakan metode ini jika Anda mengetahui nilai ambang batas yang membatasi outlier. Anda dapat mengatur Jenis rentang ke Persentil atau Angka. Jika Anda memilih Persentil, nilai Min dan Maks harus menjadi minimum dan maksimum rentang persentil (0-100) yang ingin Anda izinkan. Jika Anda memilih Angka, nilai Min dan Maks harus menjadi nilai numerik minimum dan maksimum yang ingin Anda filter dalam data.



Filter baris berdasarkan nilai kustom

Anda dapat memfilter baris dengan nilai yang memenuhi kondisi khusus. Misalnya, Anda mungkin ingin melihat pratinjau baris yang memiliki nilai harga lebih besar dari 100 sebelum menghapusnya. Dengan fungsi ini, Anda dapat memfilter baris yang melebihi ambang batas yang Anda tetapkan dan melihat pratinjau data yang difilter.

Untuk menggunakan fungsionalitas filter khusus, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Filter by rows
   ( ).

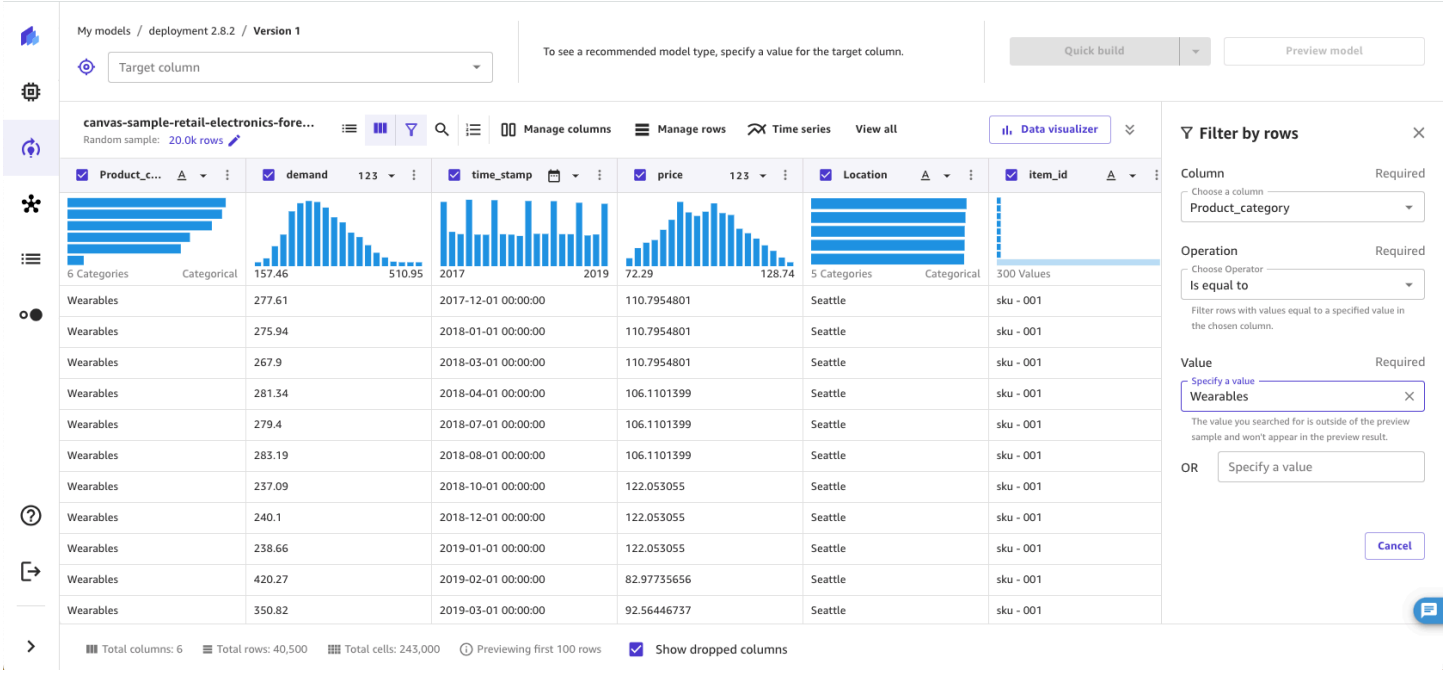
2. Pilih Kolom yang ingin Anda periksa.
3. Pilih jenis Operasi yang ingin Anda gunakan, lalu tentukan nilai untuk kondisi yang dipilih.

Untuk Operasi, Anda dapat memilih salah satu opsi berikut. Perhatikan bahwa operasi yang tersedia bergantung pada tipe data kolom yang Anda pilih. Misalnya, Anda tidak dapat membuat `is greater than` operasi untuk kolom yang berisi nilai teks.

Operasi	Tipe data yang didukung	Jenis fitur yang didukung	Fungsi
Adalah sama dengan	Numerik, Teks	Biner, Kategoris	Filter baris di mana nilai di Kolom sama dengan nilai yang Anda tentukan.
Tidak sama dengan	Numerik, Teks	Biner, Kategoris	Memfilter baris di mana nilai di Kolom tidak sama dengan nilai yang Anda tentukan.
Kurang dari	Numerik	N/A	Filter baris di mana nilai di Kolom kurang dari nilai yang Anda tentukan.
Kurang dari atau sama dengan	Numerik	N/A	Memfilter baris di mana nilai di Kolom kurang dari atau sama dengan nilai yang Anda tentukan.
Lebih besar dari	Numerik	N/A	Filter baris di mana nilai di Kolom lebih besar dari nilai yang Anda tentukan.
Lebih besar dari atau sama dengan	Numerik	N/A	Filter baris di mana nilai di Kolom lebih besar dari atau sama dengan nilai yang Anda tentukan.
Adalah antara	Numerik	N/A	Memfilter baris di mana nilai di Kolom berada di antara atau sama dengan dua nilai yang Anda tentukan.

Operasi	Tipe data yang didukung	Jenis fitur yang didukung	Fungsi
Contains	Teks	Kategoris	Filter baris di mana nilai di Kolom berisi nilai yang Anda tentukan.
Starts with	Teks	Kategoris	Filter baris di mana nilai di Kolom dimulai dengan nilai yang Anda tentukan.
Ends with	Kategoris	Kategoris	Filter baris di mana nilai di Kolom berakhir dengan nilai yang Anda tentukan.

Setelah Anda mengatur operasi filter, SageMaker Canvas memperbarui pratinjau kumpulan data untuk menunjukkan kepada Anda data yang difilter.



Fungsi dan operator

Anda dapat menggunakan fungsi matematika dan operator untuk mengeksplorasi dan mendistribusikan data Anda. Anda dapat menggunakan fungsi yang didukung SageMaker Canvas atau membuat rumus Anda sendiri dengan data yang ada dan membuat kolom baru dengan hasil

rumus. Misalnya, Anda dapat menambahkan nilai yang sesuai dari dua kolom dan menyimpan hasilnya ke kolom baru.

Anda dapat membuat pernyataan sarang untuk membuat fungsi yang lebih kompleks. Berikut ini adalah beberapa contoh fungsi bersarang yang mungkin Anda gunakan.

- Untuk menghitung BMI, Anda bisa menggunakan fungsi `weight / (height ^ 2)` tersebut.
- Untuk mengklasifikasikan usia, Anda dapat menggunakan fungsi `Case(age < 18, 'child', age < 65, 'adult', 'senior')` ini.

Anda dapat menentukan fungsi dalam tahap persiapan data sebelum Anda membangun model Anda. Untuk menggunakan fungsi, lakukan hal berikut.

- Di tab Build aplikasi SageMaker Canvas, pilih Lihat semua dan kemudian pilih Rumus khusus untuk membuka panel rumus Kustom.
- Di panel Formula khusus, Anda dapat memilih Formula untuk ditambahkan ke Resep Model Anda. Setiap rumus diterapkan ke semua nilai di kolom yang Anda tentukan. Untuk rumus yang menerima dua atau lebih kolom sebagai argumen, gunakan kolom dengan tipe data yang cocok; jika tidak, Anda akan mendapatkan kesalahan atau `null` nilai di kolom baru.
- Setelah Anda menentukan Formula, tambahkan nama kolom di bidang Nama Kolom Baru. SageMaker Canvas menggunakan nama ini untuk kolom baru yang dibuat.
- (Opsional) Pilih Pratinjau untuk melihat pratinjau transformasi Anda.
- Untuk menambahkan fungsi ke Resep Model Anda, pilih Tambah.

SageMaker Canvas menyimpan hasil fungsi Anda ke kolom baru menggunakan nama yang Anda tentukan di Nama Kolom Baru. Anda dapat melihat atau menghapus fungsi dari panel Resep Model.

SageMaker Canvas mendukung operator berikut untuk fungsi. Anda dapat menggunakan format teks atau format in-line untuk menentukan fungsi Anda.

Operator	Deskripsi	Jenis data yang didukung	Format teks	Format in-line
Tambahkan	Mengembalikan jumlah nilai	Numerik	Tambahkan (penjualan1, penjualan2)	penjualan1 + penjualan2
Kurangi	Mengembalikan perbedaan antara nilai-nilai	Numerik	Kurangi (penjualan1, penjualan2)	penjualan1 - penjualan2
Lipat gandakan	Mengembalikan produk dari nilai-nilai	Numerik	Kalikan (penjualan1, penjualan2)	penjualan1 * penjualan2
Membagi	Mengembalikan hasil bagi nilai	Numerik	Membagi (penjualan1, penjualan2)	penjualan1/ penjualan2
Mod	Mengembalikan hasil dari operator modulo (sisanya setelah membagi dua nilai)	Numerik	Mod (penjualan1, penjualan2)	penjualan1% penjualan2
Abs	Mengembalikan nilai absolut dari nilai	Numerik	Abs (penjualan1)	N/A
Menegasikan	Mengembalikan nilai negatif	Numerik	Menegasikan (c1)	-c1
Exp	Mengembalikan e (nomor Euler) dinaikkan ke kekuatan nilai	Numerik	Exp (penjualan1)	N/A
Log	Mengembalikan logaritma (basis 10) dari nilai	Numerik	Log (penjualan1)	N/A
PjM	Mengembalikan logaritma natural (basis e) dari nilai	Numerik	Ln (penjualan1)	N/A

Operator	Deskripsi	Jenis data yang didukung	Format teks	Format in-line
Pow	Mengembalikan nilai yang dinaikkan ke daya	Numerik	Pow (penjualan1, 2)	penjualan1 ^ 2
Jika	Mengembalikan label benar atau salah berdasarkan kondisi yang Anda tentukan	Boolean, Numerik, Teks	Jika (penjualan1> 7000, 'truelabel, 'falselabel')	N/A
Atau	Mengembalikan nilai boolean apakah salah satu nilai/kondisi yang ditentukan benar atau tidak	Boolean	Atau (fullprice, discount)	fullprice || discount
Dan	Mengembalikan nilai boolean apakah dua nilai/kondisi yang ditentukan benar atau tidak	Boolean	Dan (penjualan1, penjualan2)	penjualan1 && penjualan 2
Bukan	Mengembalikan nilai boolean yang merupakan kebalikan dari nilai/kondisi tertentu	Boolean	Tidak (penjualan1)	! penjualan1
Kasus	Mengembalikan nilai boolean berdasarkan pernyataan bersyarat (mengembalikan c1 jika cond1 adalah true, mengembalikan c2 jika cond2 adalah true, else mengembalikan c3)	Boolean, Numerik, Teks	Kasus (cond1, c1, cond2, c2, c3)	N/A
Sama	Mengembalikan nilai boolean apakah dua nilai sama	Boolean, Numerik, Teks	N/A	c1 = c2 c1 == c2

Operator	Deskripsi	Jenis data yang didukung	Format teks	Format in-line
Tidak sama	Mengembalikan nilai boolean apakah dua nilai tidak sama	Boolean, Numerik, Teks	N/A	c1! = c2
Kurang dari	Mengembalikan nilai boolean apakah c1 kurang dari c2	Boolean, Numerik, Teks	N/A	c1 <c2
Lebih besar dari	Mengembalikan nilai boolean apakah c1 lebih besar dari c2	Boolean, Numerik, Teks	N/A	c1 > c2
Kurang dari atau sama	Mengembalikan nilai boolean apakah c1 kurang dari atau sama dengan c2	Boolean, Numerik, Teks	N/A	c1 <= c2
Lebih besar dari atau sama	Mengembalikan nilai boolean apakah c1 lebih besar dari atau sama dengan c2	Boolean, Numerik, Teks	N/A	c1 >= c2

SageMaker Canvas juga mendukung operator agregat, yang dapat melakukan operasi seperti menghitung jumlah semua nilai atau menemukan nilai minimum dalam kolom. Anda dapat menggunakan operator agregat dalam kombinasi dengan operator standar dalam fungsi Anda. Misalnya, untuk menghitung selisih nilai dari mean, Anda bisa menggunakan fungsi tersebut`Abs(height - avg(height))`. SageMaker Canvas mendukung operator agregat berikut.

Operator agregat	Deskripsi	format	Contoh
sum	Mengembalikan jumlah semua nilai dalam kolom	sum	jumlah (c1)
minimum	Mengembalikan nilai minimum kolom	min	min (c2)

Operator agregat	Deskripsi	format	Contoh
maksimum	Mengembalikan nilai maksimum kolom	max	maks (c3)
rata-rata	Mengembalikan nilai rata-rata kolom	rata-rata	rata-rata (c4)
std	Mengembalikan standar deviasi sampel kolom	std	std (c1)
stddev	Mengembalikan standar deviasi dari nilai-nilai dalam kolom	stddev	stddev (c1)
perbedaan	Mengembalikan varians nilai yang tidak bias dalam kolom	perbedaan	varians (c1)
kira-kira _count_distinct	Mengembalikan perkiraan jumlah item yang berbeda dalam kolom	kira-kira _count_distinct	kira-kira _count_distinct (c1)
count	Mengembalikan jumlah item dalam kolom	count	menghitung (c1)
first	Mengembalikan nilai pertama dari kolom	first	pertama (c1)
last	Mengembalikan nilai terakhir dari kolom	last	terakhir (c1)
stddev_pop	Mengembalikan standar deviasi populasi kolom	stddev_pop	stddev_pop (c1)
variance_pop	Mengembalikan varians populasi dari nilai-nilai dalam kolom	variance_pop	variance_pop (c1)

Kelola baris

Dengan transformasi Kelola baris, Anda dapat melakukan pengurutan, acak acak, dan menghapus baris data dari kumpulan data.

Urutkan baris

Untuk mengurutkan baris dalam kumpulan data dengan kolom tertentu, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Kelola baris dan kemudian pilih Urutkan baris.
2. Untuk Sort Column, pilih kolom yang ingin Anda urutkan berdasarkan.
3. Untuk Urutan Urutan, pilih Ascending atau Descending.
4. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Baris acak

Untuk mengacak baris dalam kumpulan data secara acak, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Manage rows dan kemudian pilih Shuffle rows.
2. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Jatuhkan baris duplikat

Untuk menghapus baris duplikat dalam kumpulan data, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Kelola baris dan kemudian pilih Jatuhkan baris duplikat.
2. Pilih Tambah untuk menambahkan transformasi ke resep Model.

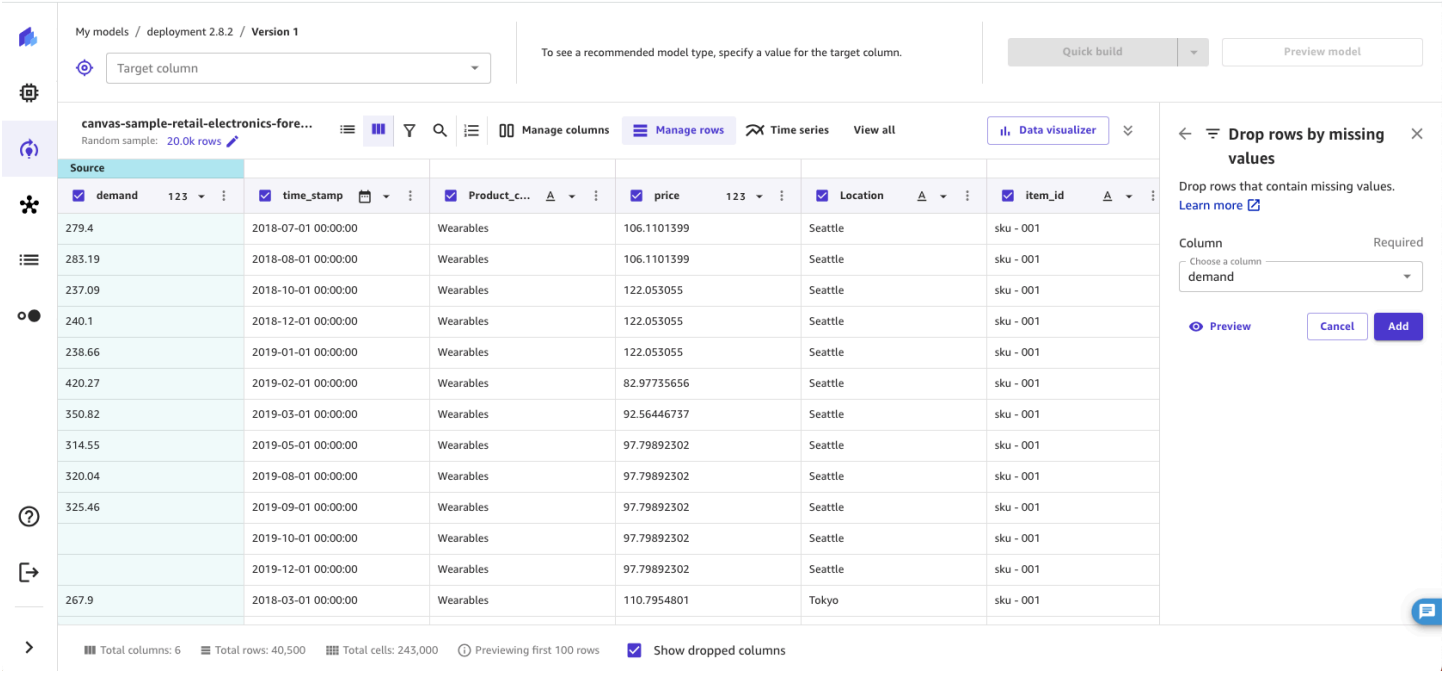
Hapus baris dengan nilai yang hilang

Nilai yang hilang adalah kejadian umum dalam kumpulan data pembelajaran mesin dan dapat memengaruhi akurasi model. Gunakan transformasi ini jika Anda ingin menjatuhkan baris dengan nilai nol atau kosong di kolom tertentu.

Untuk menghapus baris yang berisi nilai yang hilang di kolom tertentu, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Kelola baris.
2. Pilih Jatuhkan baris dengan nilai yang hilang.
3. Pilih Tambah untuk menambahkan transformasi ke resep Model.

SageMaker Kanvas menjatuhkan baris yang berisi nilai yang hilang di Kolom yang Anda pilih. Setelah menghapus baris dari dataset, SageMaker Canvas menambahkan transformasi di bagian resep Model. Jika Anda menghapus transformasi dari bagian Resep model, baris kembali ke kumpulan data Anda.



Hapus baris dengan outlier

Outlier, atau nilai langka dalam distribusi dan jangkauan data Anda, dapat berdampak negatif pada akurasi model dan menyebabkan waktu pembuatan yang lebih lama. Dengan SageMaker Canvas, Anda dapat mendeteksi dan menghapus baris yang berisi outlier di kolom numerik. Anda dapat memilih untuk menentukan outlier dengan standar deviasi atau rentang kustom.

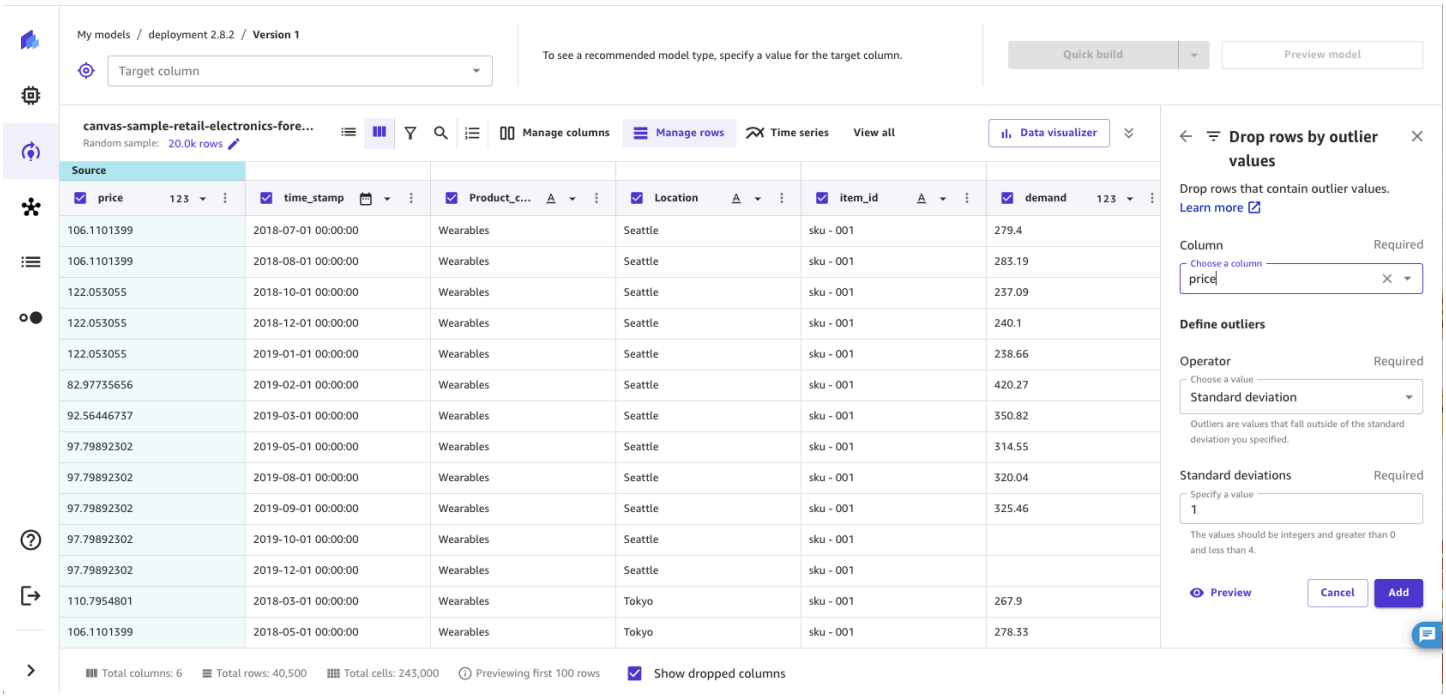
Untuk menghapus outlier dari data Anda, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Kelola baris.
2. Pilih Jatuhkan baris dengan nilai outlier.
3. Pilih Kolom yang ingin Anda periksa untuk outlier.
4. Atur Operator ke Deviasi standar, Rentang numerik kustom, atau Rentang kuantil khusus.
5. Jika Anda memilih Standar deviasi, tentukan Nilai standar deviasi (standar deviasi) dari 1-3. Jika Anda memilih Rentang numerik khusus atau Rentang kuantil khusus, tentukan nilai Min dan Maks (angka untuk rentang numerik, atau persentil antara 0— 100% untuk rentang kuantil).
6. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Opsi standar deviasi mendeteksi dan menghapus outlier dalam kolom numerik menggunakan mean dan standar deviasi. Anda menentukan jumlah standar deviasi suatu nilai harus bervariasi dari rata-rata untuk dianggap sebagai outlier. Misalnya, jika Anda menentukan 3 untuk Standar deviasi, nilai harus jatuh lebih dari 3 standar deviasi dari rata-rata untuk dianggap sebagai outlier.

Rentang numerik kustom dan pilihan rentang kuantil kustom mendeteksi dan menghapus outlier dalam kolom numerik menggunakan nilai minimum dan maksimum. Gunakan metode ini jika Anda mengetahui nilai ambang batas yang membatasi outlier. Jika Anda memilih rentang numerik, nilai Min dan Max harus menjadi nilai numerik minimum dan maksimum yang ingin Anda izinkan dalam data. Jika Anda memilih rentang kuantil, nilai Min dan Maks harus menjadi minimum dan maksimum rentang persentil (0—100) yang ingin Anda izinkan.

Setelah menghapus baris dari dataset, SageMaker Canvas menambahkan transformasi di bagian resep Model. Jika Anda menghapus transformasi dari bagian Resep model, baris kembali ke kumpulan data Anda.



Hapus baris dengan nilai kustom

Anda dapat menghapus baris dengan nilai yang memenuhi kondisi khusus. Misalnya, Anda mungkin ingin mengecualikan semua baris dengan nilai harga lebih dari 100 saat membuat model Anda. Dengan transformasi ini, Anda dapat membuat aturan yang menghapus semua baris yang melebihi ambang batas yang Anda tetapkan.

Untuk menggunakan transformasi hapus kustom, lakukan hal berikut.

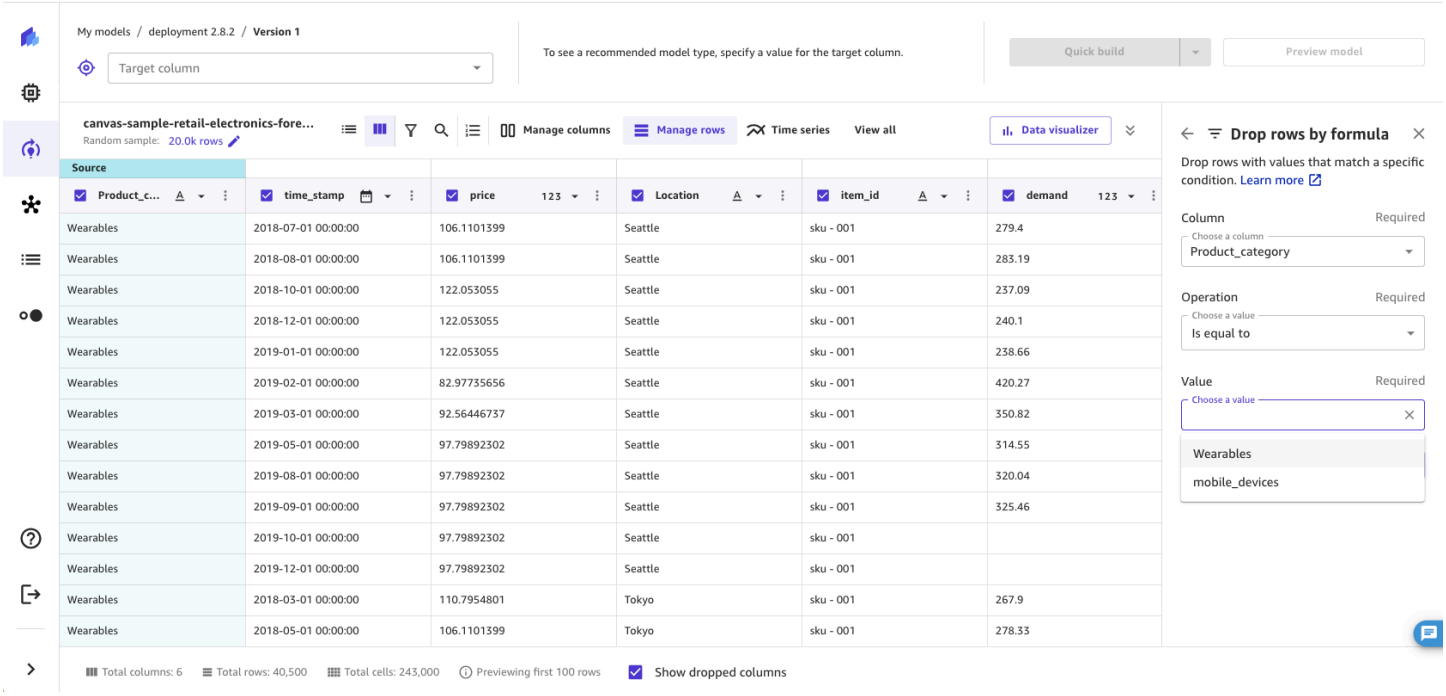
1. Di tab Build aplikasi SageMaker Canvas, pilih Kelola baris.
2. Pilih Jatuhkan baris dengan rumus.
3. Pilih Kolom yang ingin Anda periksa.
4. Pilih jenis Operasi yang ingin Anda gunakan, lalu tentukan nilai untuk kondisi yang dipilih.
5. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Untuk Operasi, Anda dapat memilih salah satu opsi berikut. Perhatikan bahwa operasi yang tersedia bergantung pada tipe data kolom yang Anda pilih. Misalnya, Anda tidak dapat membuat `is greater than` operasi untuk kolom yang berisi nilai teks.

Operasi	Tipe data yang didukung	Jenis fitur yang didukung	Fungsi
Adalah sama dengan	Numerik, Teks	Biner, Kategoris	Menghapus baris di mana nilai di Kolom sama dengan nilai yang Anda tentukan.
Tidak sama dengan	Numerik, Teks	Biner, Kategoris	Menghapus baris di mana nilai di Kolom tidak sama dengan nilai yang Anda tentukan.
Kurang dari	Numerik	N/A	Menghapus baris di mana nilai di Kolom kurang dari nilai yang Anda tentukan.
Kurang dari atau sama dengan	Numerik	N/A	Menghapus baris di mana nilai di Kolom kurang dari atau sama dengan nilai yang Anda tentukan.
Lebih besar dari	Numerik	N/A	Menghapus baris di mana nilai di Kolom lebih besar dari nilai yang Anda tentukan.
Lebih besar dari atau sama dengan	Numerik	N/A	Menghapus baris di mana nilai di Kolom lebih besar dari atau sama dengan nilai yang Anda tentukan.

Operasi	Tipe data yang didukung	Jenis fitur yang didukung	Fungsi
Adalah antara	Numerik	N/A	Menghapus baris di mana nilai di Kolom berada di antara atau sama dengan dua nilai yang Anda tentukan.
Contains	Teks	Kategoris	Menghapus baris di mana nilai di Kolom berisi nilai yang Anda tentukan.
Starts with	Teks	Kategoris	Menghapus baris di mana nilai di Kolom dimulai dengan nilai yang Anda tentukan.
Ends with	Teks	Kategoris	Menghapus baris di mana nilai di Kolom berakhir dengan nilai yang Anda tentukan.

Setelah menghapus baris dari dataset, SageMaker Canvas menambahkan transformasi di bagian resep Model. Jika Anda menghapus transformasi dari bagian Resep model, baris kembali ke kumpulan data Anda.



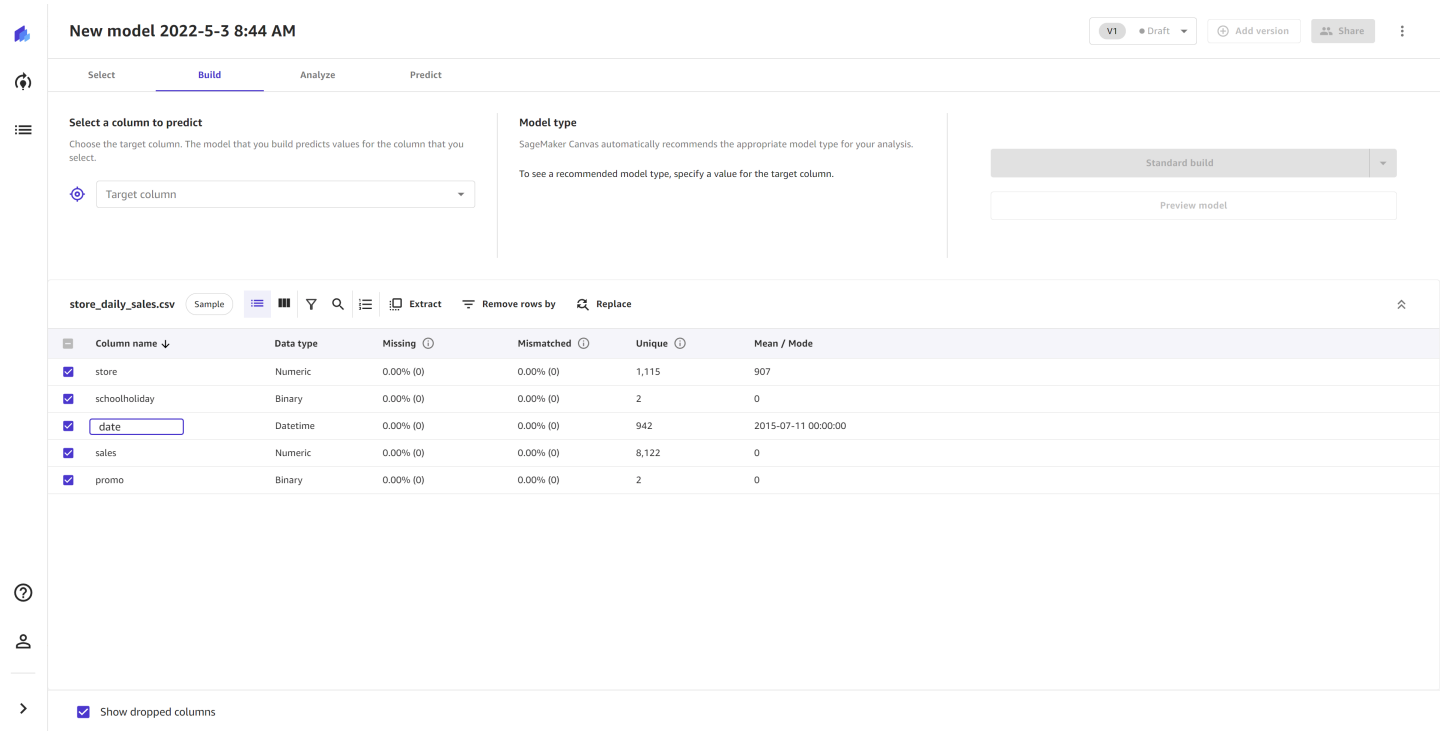
Ubah Nama kolom

Dengan transformasi kolom ganti nama, Anda dapat mengganti nama kolom dalam data Anda. Saat Anda mengganti nama kolom, SageMaker Canvas mengubah nama kolom di input model.

Anda dapat mengganti nama kolom dalam kumpulan data Anda dengan mengklik dua kali pada nama kolom di tab Build aplikasi SageMaker Canvas dan memasukkan nama baru. Menekan tombol Enter mengirimkan perubahan, dan mengklik di mana saja di luar input membatalkan perubahan. Anda juga dapat mengganti nama kolom dengan mengklik ikon Opsi lainnya ( ), terletak di akhir baris dalam tampilan daftar atau di akhir sel header dalam tampilan kisi, dan memilih Ganti nama.

Nama kolom Anda tidak boleh lebih dari 32 karakter atau memiliki garis bawah ganda (__), dan Anda tidak dapat mengganti nama kolom menjadi nama yang sama dengan kolom lain. Anda juga tidak dapat mengganti nama kolom yang dijatuhkan.

Tangkapan layar berikut menunjukkan cara mengganti nama kolom dengan mengklik dua kali nama kolom.



Saat Anda mengganti nama kolom, SageMaker Canvas menambahkan transformasi di bagian Resep Model. Jika Anda menghapus transformasi dari bagian Resep Model, kolom kembali ke nama aslinya.

Kelola kolom

Dengan transformasi berikut, Anda dapat mengubah tipe data kolom dan mengganti nilai atau outlier yang hilang untuk kolom tertentu. SageMaker Canvas menggunakan tipe atau nilai data yang diperbarui saat membuat model Anda tetapi tidak mengubah kumpulan data asli Anda. Perhatikan bahwa jika Anda menjatuhkan kolom dari kumpulan data menggunakan Jatuhkan kolom transformasi, Anda tidak dapat mengganti nilai di kolom itu.

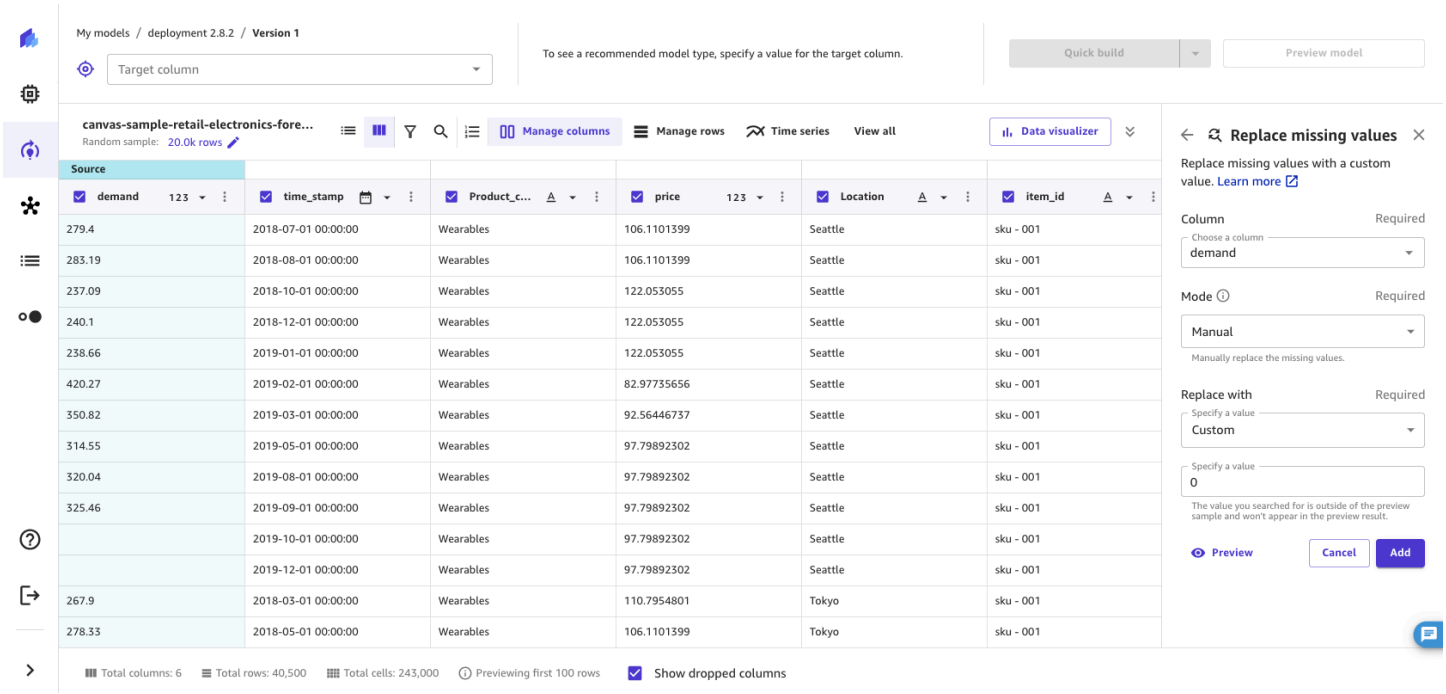
Ganti nilai yang hilang

Nilai yang hilang adalah kejadian umum dalam kumpulan data pembelajaran mesin dan dapat memengaruhi akurasi model. Anda dapat memilih untuk menjatuhkan baris yang memiliki nilai yang hilang, tetapi model Anda lebih akurat jika Anda memilih untuk mengganti nilai yang hilang. Dengan transformasi ini, Anda dapat mengganti nilai yang hilang di kolom numerik dengan rata-rata atau median data dalam kolom, atau Anda juga dapat menentukan nilai khusus untuk mengganti nilai yang hilang. Untuk kolom non-numerik, Anda dapat mengganti nilai yang hilang dengan mode (nilai paling umum) kolom atau nilai khusus.

Gunakan transformasi ini jika Anda ingin mengganti nilai nol atau kosong di kolom tertentu. Untuk mengganti nilai yang hilang di kolom tertentu, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Kelola kolom.
2. Pilih Ganti nilai yang hilang.
3. Pilih Kolom di mana Anda ingin mengganti nilai yang hilang.
4. Atur Mode ke Manual untuk mengganti nilai yang hilang dengan nilai yang Anda tentukan. Dengan pengaturan Otomatis (default), SageMaker Canvas menggantikan nilai yang hilang dengan nilai imputasi yang paling sesuai dengan data Anda. Metode imputasi ini dilakukan secara otomatis untuk setiap model build, kecuali jika Anda menentukan mode Manual.
5. Mengatur Ganti dengan nilai:
   - Jika kolom Anda numerik, pilih Mean, Median, atau Custom. Mean menggantikan nilai yang hilang dengan mean untuk kolom, dan Median menggantikan nilai yang hilang dengan median untuk kolom. Jika Anda memilih Kustom, maka Anda harus menentukan nilai kustom yang ingin Anda gunakan untuk mengganti nilai yang hilang.
   - Jika kolom Anda non-numerik, pilih Mode atau Kustom. Mode menggantikan nilai yang hilang dengan mode, atau nilai yang paling umum, untuk kolom. Untuk Kustom, tentukan nilai kustom. yang ingin Anda gunakan untuk mengganti nilai yang hilang.
6. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Setelah mengganti nilai yang hilang dalam kumpulan data, SageMaker Canvas menambahkan transformasi di bagian resep Model. Jika Anda menghapus transformasi dari bagian Resep model, nilai yang hilang kembali ke kumpulan data.



Ganti outlier

Outlier, atau nilai langka dalam distribusi dan jangkauan data Anda, dapat berdampak negatif pada akurasi model dan menyebabkan waktu pembuatan yang lebih lama. SageMaker Canvas memungkinkan Anda mendeteksi outlier di kolom numerik dan mengganti outlier dengan nilai yang berada dalam rentang yang diterima dalam data Anda. Anda dapat memilih untuk menentukan outlier dengan standar deviasi atau rentang kustom, dan Anda dapat mengganti outlier dengan nilai minimum dan maksimum dalam rentang yang diterima.

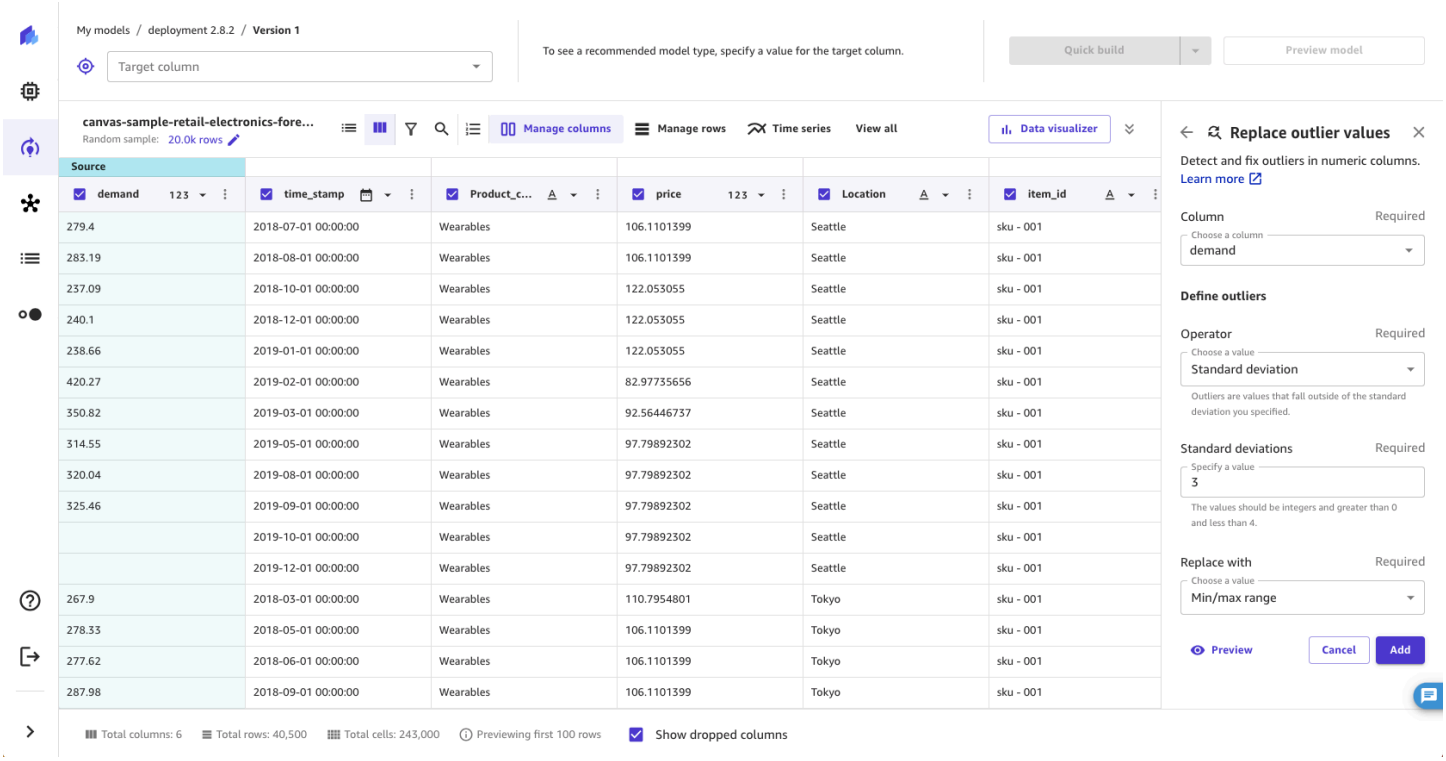
Untuk mengganti outlier dalam data Anda, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Kelola kolom.
2. Pilih Ganti nilai outlier.
3. Pilih Kolom di mana Anda ingin mengganti outlier.
4. Untuk Tentukan pencilan, pilih Deviasi standar, Rentang numerik khusus, atau Rentang kuantil khusus.
5. Jika Anda memilih Standar deviasi, tentukan Nilai standar deviasi (standar deviasi) dari 1-3. Jika Anda memilih Rentang numerik khusus atau Rentang kuantil khusus, tentukan nilai Min dan Maks (angka untuk rentang numerik, atau persentil antara 0— 100% untuk rentang kuantil).
6. Untuk Ganti dengan, pilih rentang Min/maks.
7. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Opsi standar deviasi mendeteksi outlier dalam kolom numerik menggunakan mean dan standar deviasi. Anda menentukan jumlah standar deviasi suatu nilai harus bervariasi dari rata-rata untuk dianggap sebagai outlier. Misalnya, jika Anda menentukan 3 untuk Standar deviasi, nilai harus jatuh lebih dari 3 standar deviasi dari rata-rata untuk dianggap sebagai outlier. SageMaker Canvas menggantikan outlier dengan nilai minimum atau nilai maksimum dalam kisaran yang diterima. Misalnya, jika Anda mengonfigurasi standar deviasi untuk hanya menyertakan nilai dari 200-300, maka SageMaker Canvas mengubah nilai 198 menjadi 200 (minimum).

Rentang numerik kustom dan pilihan rentang kuantil kustom mendeteksi outlier dalam kolom numerik menggunakan nilai minimum dan maksimum. Gunakan metode ini jika Anda mengetahui nilai ambang batas yang membatasi outlier. Jika Anda memilih rentang numerik, nilai Min dan Max harus menjadi nilai numerik minimum dan maksimum yang ingin Anda izinkan. SageMaker Canvas menggantikan nilai apa pun yang berada di luar nilai minimum dan maksimum ke nilai minimum dan maksimum. Misalnya, jika rentang Anda hanya mengizinkan nilai dari 1-100, maka SageMaker Canvas mengubah nilai 102 menjadi 100 (maksimum). Jika Anda memilih rentang kuantil, nilai Min dan Maks harus menjadi minimum dan maksimum rentang persentil (0—100) yang ingin Anda izinkan.

Setelah mengganti nilai dalam kumpulan data, SageMaker Canvas menambahkan transformasi di bagian resep Model. Jika Anda menghapus transformasi dari bagian Resep model, nilai asli kembali ke kumpulan data.



Ubah tipe data

SageMaker Canvas memberi Anda kemampuan untuk mengubah tipe data kolom Anda antara numerik, teks, dan datetime, sementara juga menampilkan tipe fitur terkait untuk tipe data tersebut. Tipe data mengacu pada format data dan cara penyimpanannya, sedangkan tipe fitur mengacu pada karakteristik data yang digunakan dalam algoritma pembelajaran mesin, seperti biner atau kategoris. Ini memberi Anda fleksibilitas untuk secara manual mengubah jenis data di kolom Anda berdasarkan fitur. Kemampuan untuk memilih tipe data yang tepat memastikan integritas dan akurasi data sebelum membangun model. Tipe data ini digunakan saat membuat model.

> **Note**
>
> Saat ini, mengubah jenis fitur (misalnya, dari biner ke kategoris) tidak didukung.

Tabel berikut mencantumkan semua tipe data yang didukung di Canvas.

Jenis data	Deskripsi	Contoh
Numerik	Data numerik mewakili nilai numerik	1, 2, 3 1.1, 1.2. 1.3
Teks	Data teks mewakili urutan karakter, seperti nama atau deskripsi	A, B, C, D apel, pisang, jeruk 1A! , 2A! , 3A!
Datetime	Data datetime mewakili tanggal dan waktu dalam format stempel waktu	2019-07-01 01:00:00, 2019-07-01 02:00:00, 2019-07-01 03:00:00

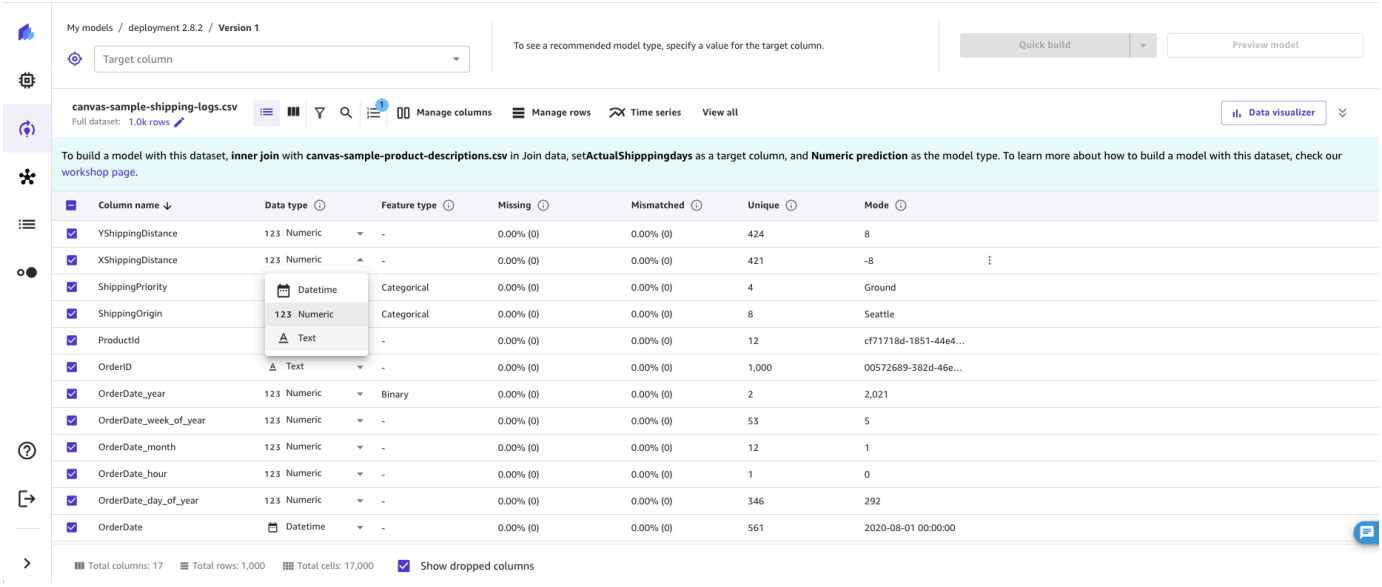
Tabel berikut mencantumkan semua jenis fitur yang didukung di Canvas.

Jenis fitur	Deskripsi	Contoh
Biner	Fitur biner mewakili dua nilai yang mungkin	0, 1, 0, 1, 0 (2 nilai berbeda)

Jenis fitur	Deskripsi	Contoh
		benar, salah, benar (2 nilai berbeda)
Kategoris	Fitur kategoris mewakili kategori atau kelompok yang berbeda	apel, pisang, jeruk, apel (3 nilai berbeda) A, B, C, D, E, A, D, C (5 nilai berbeda)

Untuk mengubah tipe data kolom dalam kumpulan data, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, buka tampilan Kolom atau tampilan Grid dan pilih dropdown tipe Data untuk kolom tertentu.
2. Dalam tarik-turun tipe Data, pilih tipe data yang akan dikonversi. Screenshot berikut menunjukkan menu dropdown.



3. Untuk Kolom, pilih atau verifikasi kolom yang ingin Anda ubah tipe datanya.
4. Untuk tipe data baru, pilih atau verifikasi tipe data baru yang ingin Anda konversi.
5. Jika tipe data baru adalah `Datetime` atau`Numeric`, pilih salah satu opsi berikut di bawah Menangani nilai yang tidak valid:

   a. Ganti dengan nilai kosong - Nilai tidak valid diganti dengan nilai kosong

   b. Hapus baris - Baris dengan nilai tidak valid dihapus dari kumpulan data

c. Ganti dengan nilai kustom - Nilai tidak valid diganti dengan Nilai Kustom yang Anda tentukan.

6. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Tipe data untuk kolom Anda sekarang harus diperbarui.

Siapkan data deret waktu

Gunakan fungsionalitas berikut untuk menyiapkan data deret waktu Anda untuk membangun model peramalan deret waktu.

Sampel ulang data deret waktu

Dengan mengambil sampel ulang data deret waktu, Anda dapat menetapkan interval reguler untuk pengamatan dalam kumpulan data deret waktu Anda. Ini sangat berguna ketika bekerja dengan data deret waktu yang berisi pengamatan dengan jarak yang tidak beraturan. Misalnya, Anda dapat menggunakan resampling untuk mengubah kumpulan data dengan pengamatan yang direkam setiap interval satu jam, dua jam dan tiga jam menjadi interval satu jam reguler di antara pengamatan. Algoritma peramalan membutuhkan pengamatan yang harus dilakukan secara berkala.

Untuk mengambil sampel ulang data deret waktu, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pilih Time series.
2. Pilih Sampel Ulang.
3. Untuk kolom Timestamp, pilih kolom yang ingin Anda terapkan transformasi. Anda hanya dapat memilih kolom dari jenis Datetime.
4. Di bagian Pengaturan frekuensi, pilih Frekuensi dan Tingkat. Frekuensi adalah satuan frekuensi dan Rate adalah interval dari satuan frekuensi yang akan diterapkan pada kolom. Misalnya, memilih `Calendar Day` untuk nilai Frekuensi dan `1` untuk Nilai menetapkan interval untuk meningkat setiap 1 hari kalender, seperti`2023-03-26 00:00:00`,`2023-03-27 00:00:00`,`2023-03-28 00:00:00`. Lihat tabel setelah prosedur ini untuk daftar lengkap nilai Frekuensi.
5. Pilih Tambah untuk menambahkan transformasi ke resep Model.

Tabel berikut mencantumkan semua jenis Frekuensi yang dapat Anda pilih saat pengambilan sampel ulang data deret waktu.

Frekuensi	Deskripsi	Contoh nilai (dengan asumsi Rate adalah 1)
Hari Kerja	Sampel ulang pengamatan di kolom datetime menjadi 5 hari kerja dalam seminggu (Senin, Selasa, Rabu, Kamis, Jumat)	2023-03-24 00:00:00 2023-03-27 00:00:00 2023-03-28 00:00:00 2023-03-29 00:00:00 2023-03-30 00:00:00 2023-03-31 00:00:00 2023-04-03 00:00:00
Hari Kalender	Sampel ulang pengamatan di kolom datetime ke semua 7 hari dalam seminggu (Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, Minggu)	2023-03-26 00:00:00 2023-03-27 00:00:00 2023-03-28 00:00:00 2023-03-29 00:00:00 2023-03-30 00:00:00 2023-03-31 00:00:00 2023-04-01 00:00:00
Minggu	Sampel ulang pengamatan di kolom datetime ke hari pertama setiap minggu	2023-03-13 00:00:00 2023-03-20 00:00:00 2023-03-27 00:00:00 2023-04-03 00:00:00
Bulan	Sampel ulang pengamatan di kolom datetime ke hari pertama setiap bulan	2023-03-01 00:00:00 2023-04-01 00:00:00

Frekuensi	Deskripsi	Contoh nilai (dengan asumsi Rate adalah 1)
		2023-05-01 00:00:00 2023-06-01 00:00:00
Kuartal Tahunan	Sampel ulang pengamatan di kolom datetime ke hari terakhir setiap kuartal	2023-03-31 00:00:00 2023-06-30 00:00:00 2023-09-30 00:00:00 2023-12-31 00:00:00
Tahun	Sampel ulang pengamatan di kolom datetime ke hari terakhir setiap tahun	2022-12-31 0:00:00 2023-12-31 00:00:00 2024-12-31 00:00:00
Jam	Sampel ulang pengamatan di kolom datetime ke setiap jam setiap hari	2023-03-24 00:00:00 2023-03-24 01:00:00 2023-03-24 02:00:00 2023-03-24 03:00:00
Menit	Sampel ulang pengamatan di kolom datetime ke setiap menit setiap jam	2023-03-24 00:00:00 2023-03-24 00:01:00 2023-03-24 00:02:00 2023-03-24 00:03:00

Frekuensi	Deskripsi	Contoh nilai (dengan asumsi Rate adalah 1)
Detik	Sampel ulang pengamatan di kolom datetime ke setiap detik setiap menit	2023-03-24 00:00:00 2023-03-24 00:00:01 2023-03-24 00:00:02 2023-03-24 00:00:03

Saat menerapkan transformasi resampling, Anda dapat menggunakan opsi Lanjutan untuk menentukan bagaimana nilai yang dihasilkan dari kolom lainnya (selain kolom stempel waktu) di kumpulan data Anda dimodifikasi. Ini dapat dicapai dengan menentukan metodologi resampling, yang dapat berupa downsampling atau upsampling untuk kolom numerik dan non-numerik.

Downsampling meningkatkan interval antara pengamatan dalam dataset. Misalnya, jika Anda menurunkan sampel pengamatan yang diambil setiap jam atau setiap dua jam, setiap pengamatan dalam kumpulan data Anda dilakukan setiap dua jam. Nilai kolom lain dari pengamatan per jam digabungkan menjadi satu nilai menggunakan metode kombinasi. Tabel berikut menunjukkan contoh data deret waktu downsampling dengan menggunakan mean sebagai metode kombinasi. Data di-downsample dari setiap dua jam menjadi setiap jam.

Tabel berikut menunjukkan pembacaan suhu per jam selama sehari sebelum downsampling.

Stempel Waktu	Suhu (Celcius)
12:00pm	30
1:00 pagi	32
2:00 pagi	35
3:00 pagi	32
4:00 pagi	30

Tabel berikut menunjukkan pembacaan suhu setelah downsampling untuk setiap dua jam.

Stempel Waktu	Suhu (Celcius)
12:00pm	30
2:00 pagi	33.5
2:00 pagi	35
4:00 pagi	32.5

Untuk menurunkan sampel data deret waktu, lakukan hal berikut:

1. Perluas bagian Advanced di bawah Transformasi Resample.
2. Pilih kombinasi non-numerik untuk menentukan metode kombinasi untuk kolom non-numerik. Lihat tabel di bawah ini untuk daftar lengkap metode kombinasi.
3. Pilih kombinasi numerik untuk menentukan metode kombinasi untuk kolom numerik. Lihat tabel di bawah ini untuk daftar lengkap metode kombinasi.

Jika Anda tidak menentukan metode kombinasi, nilai default adalah `Most Common` untuk kombinasi Non-numerik dan **`Mean`** untuk kombinasi Numerik. Tabel berikut mencantumkan metode untuk kombinasi numerik dan non-numerik.

Metodologi downsampling	Metode kombinasi	Deskripsi
Kombinasi non-numerik	Paling Umum	Nilai agregat di kolom non-numerik dengan nilai yang paling umum terjadi
Kombinasi non-numerik	Terakhir	Nilai agregat di kolom non-numerik dengan nilai terakhir di kolom
Kombinasi non-numerik	Pertama	Nilai agregat di kolom non-numerik dengan nilai pertama di kolom

Metodologi downsampling	Metode kombinasi	Deskripsi
Kombinasi numerik	Berarti	Nilai agregat di kolom numerik dengan mengambil rata-rata semua nilai di kolom
Kombinasi numerik	Median	Nilai agregat di kolom numerik dengan mengambil median semua nilai di kolom
Kombinasi numerik	Min	Nilai agregat di kolom numerik dengan mengambil minimum semua nilai di kolom
Kombinasi numerik	Maks	Nilai agregat di kolom numerik dengan mengambil maksimum semua nilai di kolom
Kombinasi numerik	Jumlah	Nilai agregat di kolom numerik dengan menambahkan semua nilai di kolom
Kombinasi numerik	Kuantil	Nilai agregat di kolom numerik dengan mengambil kuantil semua nilai di kolom

Upsampling mengurangi interval antara pengamatan dalam dataset. Misalnya, jika Anda mengambil sampel pengamatan yang diambil setiap dua jam ke dalam pengamatan per jam, nilai kolom lain dari pengamatan per jam diinterpolasi dari yang telah diambil setiap dua jam.

Untuk meningkatkan data deret waktu, lakukan hal berikut:

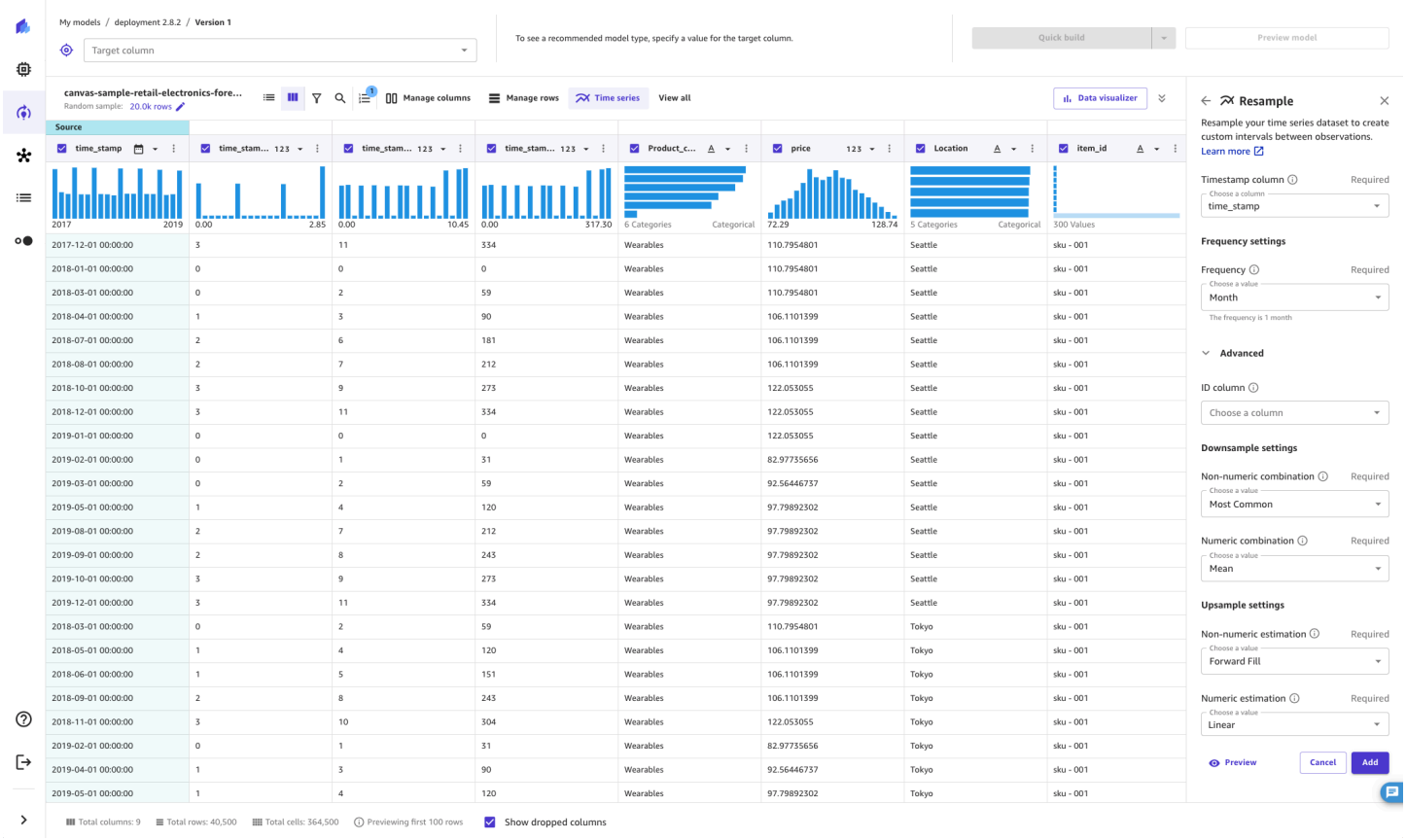
1. Perluas bagian Advanced di bawah Transformasi Resample.
2. Pilih estimasi non-numerik untuk menentukan metode estimasi untuk kolom non-numerik. Lihat tabel setelah prosedur ini untuk daftar lengkap metode.
3. Pilih Estimasi numerik untuk menentukan metode estimasi untuk kolom numerik. Lihat tabel di bawah ini untuk daftar lengkap metode.

4. (Opsional) Pilih Kolom ID untuk menentukan kolom yang memiliki ID pengamatan deret waktu. Tentukan opsi ini jika kumpulan data Anda memiliki dua deret waktu. Jika Anda memiliki kolom yang hanya mewakili satu deret waktu, jangan tentukan nilai untuk bidang ini. Misalnya, Anda dapat memiliki kumpulan data yang memiliki kolom `id` dan`purchase`. `id`Kolom memiliki nilai-nilai berikut:`[1, 2, 2, 1]`. `purchase`Kolom memiliki nilai-nilai berikut`[$2, $3, $4, $1]`. Oleh karena itu, kumpulan data memiliki dua deret waktu—satu deret waktu adalah:`1: [$2, $1]`, dan deret waktu lainnya adalah. `2: [$3, $4]`

Jika Anda tidak menentukan metode estimasi, nilai default adalah **`Forward Fill`** untuk estimasi non-numerik dan **`Linear`** untuk estimasi Numerik. Tabel berikut mencantumkan metode untuk estimasi.

Metodologi upsampling	Metode estimasi	Deskripsi
Estimasi non-numerik	Isi Maju	Interpolasi nilai di kolom non-numerik dengan mengambil nilai berurutan setelah semua nilai di kolom
Estimasi non-numerik	Isi Mundur	Interpolasi nilai di kolom non-numerik dengan mengambil nilai berurutan sebelum semua nilai di kolom
Estimasi non-numerik	Tetap Hilang	Interpolasi nilai di kolom non-numerik dengan menunjukkan nilai kosong
Estimasi numerik	Linear, Waktu, Indeks, Nol, S-Linear, Terdekat, Kuadrat, Kubik, Barycentric, Polinomia l, Krogh, Polinomial Sepotong, Spline, P-chip, Akima, Cubic Spline, Dari Derivatif	Interpolasi nilai dalam kolom numerik dengan menggunak an interpolator specfied. Untuk informasi tentang metode interpolasi, lihat panda. DataFrame.interpolate dalam dokumentasi panda.

Tangkapan layar berikut menunjukkan Pengaturan lanjutan dengan bidang untuk downsampling dan upsampling diisi.



Gunakan ekstraksi datetime

Dengan transformasi ekstraksi datetime, Anda dapat mengekstrak nilai dari kolom datetime ke kolom terpisah. Misalnya, jika Anda memiliki kolom yang berisi tanggal pembelian, Anda dapat mengekstrak nilai bulan ke kolom terpisah dan menggunakan kolom baru saat membuat model Anda. Anda juga dapat mengekstrak beberapa nilai untuk memisahkan kolom dengan satu transformasi.

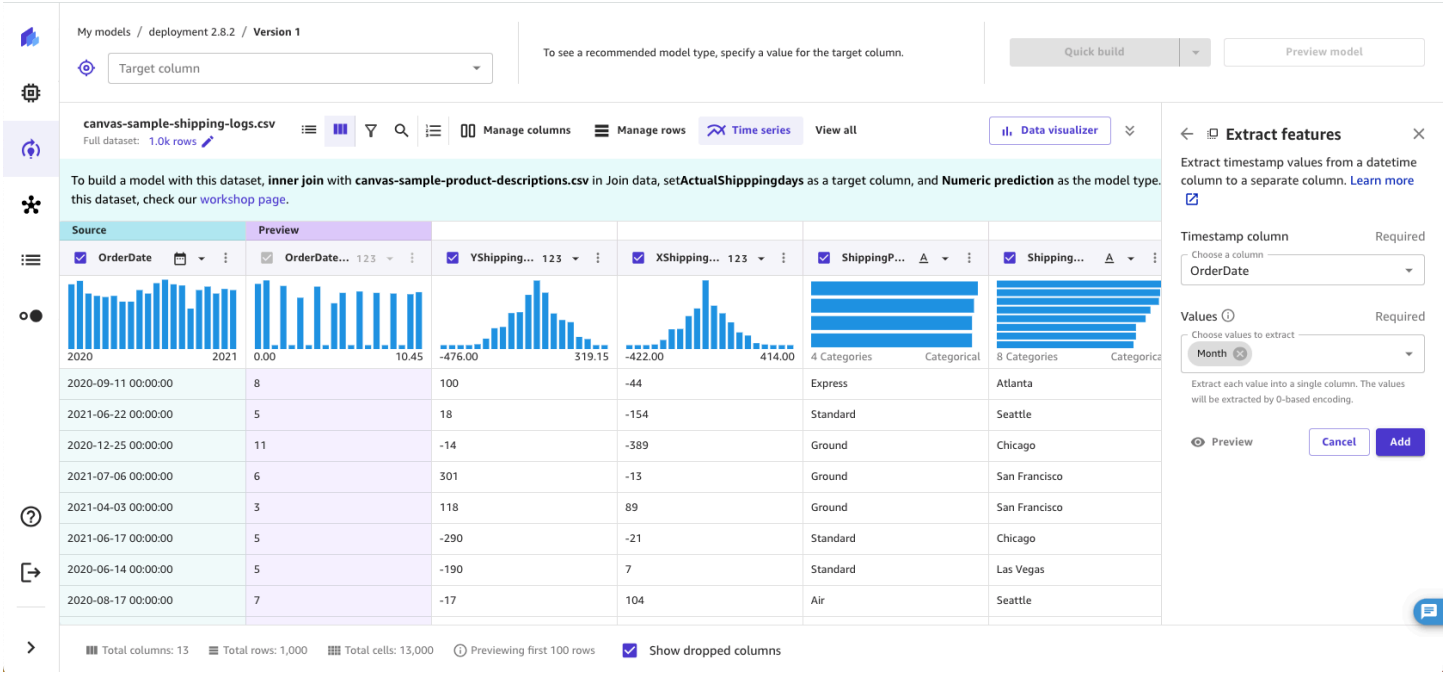
Kolom datetime Anda harus menggunakan format stempel waktu yang didukung. Untuk daftar format yang didukung SageMaker Canvas, lihatPrakiraan Deret Waktu di Amazon SageMaker Canvas. Jika kumpulan data Anda tidak menggunakan salah satu format yang didukung, perbarui kumpulan data Anda untuk menggunakan format stempel waktu yang didukung dan impor ulang ke SageMaker Amazon Canvas sebelum membuat model Anda.

Untuk melakukan ekstraksi datetime, lakukan hal berikut.

1. Di tab Build aplikasi SageMaker Canvas, pada bilah transformasi, pilih Lihat semua.
2. Pilih fitur Ekstrak.

3. Pilih kolom Timestamp dari mana Anda ingin mengekstrak nilai.
4. Untuk Nilai, pilih satu atau beberapa nilai untuk diekstrak dari kolom. Nilai yang dapat Anda ekstrak dari kolom stempel waktu adalah Tahun, Bulan, Hari, Jam, Minggu tahun, Hari tahun, dan Kuartal.
5. (Opsional) Pilih Pratinjau untuk melihat pratinjau hasil transformasi.
6. Pilih Tambah untuk menambahkan transformasi ke resep Model.

SageMaker Canvas membuat kolom baru dalam kumpulan data untuk setiap nilai yang Anda ekstrak. Kecuali untuk nilai Tahun, SageMaker Canvas menggunakan pengkodean berbasis 0 untuk nilai yang diekstraksi. Misalnya, jika Anda mengekstrak nilai Bulan, Januari diekstraksi sebagai 0, dan Februari diekstraksi sebagai 1.



Anda dapat melihat transformasi yang tercantum di bagian Resep model. Jika Anda menghapus transformasi dari bagian Resep model, kolom baru akan dihapus dari kumpulan data.

Evaluasi Kinerja Model Anda di Amazon SageMaker Canvas

Setelah Anda membuat model, Anda dapat mengevaluasi seberapa baik kinerja model Anda pada data Anda sebelum menggunakannya untuk membuat prediksi. Anda dapat menggunakan informasi, seperti akurasi model saat memprediksi label dan metrik lanjutan, untuk menentukan apakah model Anda dapat membuat prediksi yang cukup akurat untuk data Anda.

Pada halaman Analisis untuk model Anda, Amazon SageMaker Canvas menyediakan tiga tab berikut:

- Ikhtisar — Memberi Anda gambaran umum tentang kinerja model, tergantung pada jenis model.
- Skor — Menampilkan visualisasi yang dapat Anda gunakan untuk mendapatkan lebih banyak wawasan tentang kinerja model Anda di luar metrik akurasi keseluruhan.
- Metrik lanjutan - Berisi skor model Anda untuk metrik lanjutan dan informasi tambahan yang dapat memberi Anda pemahaman yang lebih dalam tentang kinerja model Anda. Anda juga dapat melihat informasi seperti dampak kolom.

Bagian ini Evaluasi kinerja model Anda menjelaskan cara melihat dan menafsirkan tab Ikhtisar dan Skor model Anda. Bagian ini Gunakan metrik lanjutan dalam analisis Anda berisi informasi lebih rinci tentang metrik Lanjutan yang digunakan untuk mengukur akurasi model Anda.

Anda juga dapat melihat informasi lebih lanjut untuk kandidat model tertentu, yang merupakan semua iterasi model yang dijalankan Canvas saat membangun model Anda. Berdasarkan metrik lanjutan untuk kandidat model tertentu, Anda dapat memilih kandidat yang berbeda untuk menjadi default, atau versi yang digunakan untuk membuat prediksi dan penerapan. Untuk setiap kandidat model, Anda dapat melihat informasi metrik lanjutan untuk membantu Anda memutuskan kandidat model mana yang ingin Anda pilih sebagai default. Anda dapat melihat informasi ini dengan memilih kandidat model dari papan peringkat Model. Untuk informasi selengkapnya, lihat ???.

Canvas juga menyediakan opsi untuk mengunduh notebook Jupyter sehingga Anda dapat melihat dan menjalankan kode yang digunakan untuk membangun model Anda. Ini berguna jika Anda ingin membuat penyesuaian pada kode atau mempelajari lebih lanjut tentang bagaimana model Anda dibuat. Untuk informasi selengkapnya, lihat Unduh notebook model.

Evaluasi kinerja model Anda

Amazon SageMaker Canvas memberikan ikhtisar dan informasi penilaian untuk berbagai jenis model. Skor model Anda dapat membantu Anda menentukan seberapa akurat model Anda ketika membuat prediksi. Wawasan penilaian tambahan dapat membantu Anda mengukur perbedaan antara nilai aktual dan prediksi.

Untuk melihat analisis model Anda, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.

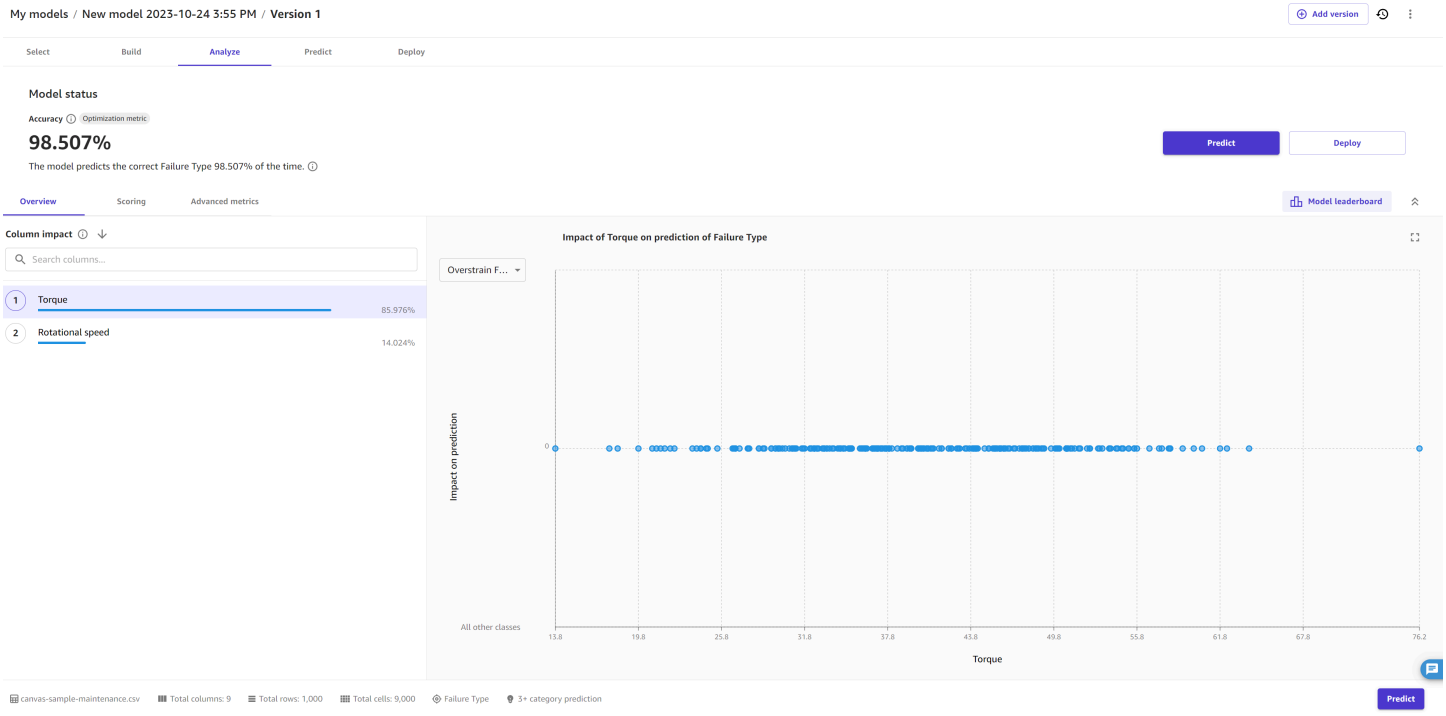
3. Pilih model yang Anda buat.
4. Di panel navigasi atas, pilih tab Analisis.
5. Di dalam tab Analisis, Anda dapat melihat ikhtisar dan informasi penilaian untuk model Anda.

Bagian berikut menjelaskan cara menafsirkan penilaian untuk setiap jenis model.

Mengevaluasi model prediksi kategoris

Tab Ikhtisar menunjukkan dampak kolom untuk setiap kolom. Dampak kolom adalah skor persentase yang menunjukkan berapa banyak bobot kolom dalam membuat prediksi dalam kaitannya dengan kolom lainnya. Untuk dampak kolom 25%, Canvas menimbang prediksi sebagai 25% untuk kolom dan 75% untuk kolom lainnya.

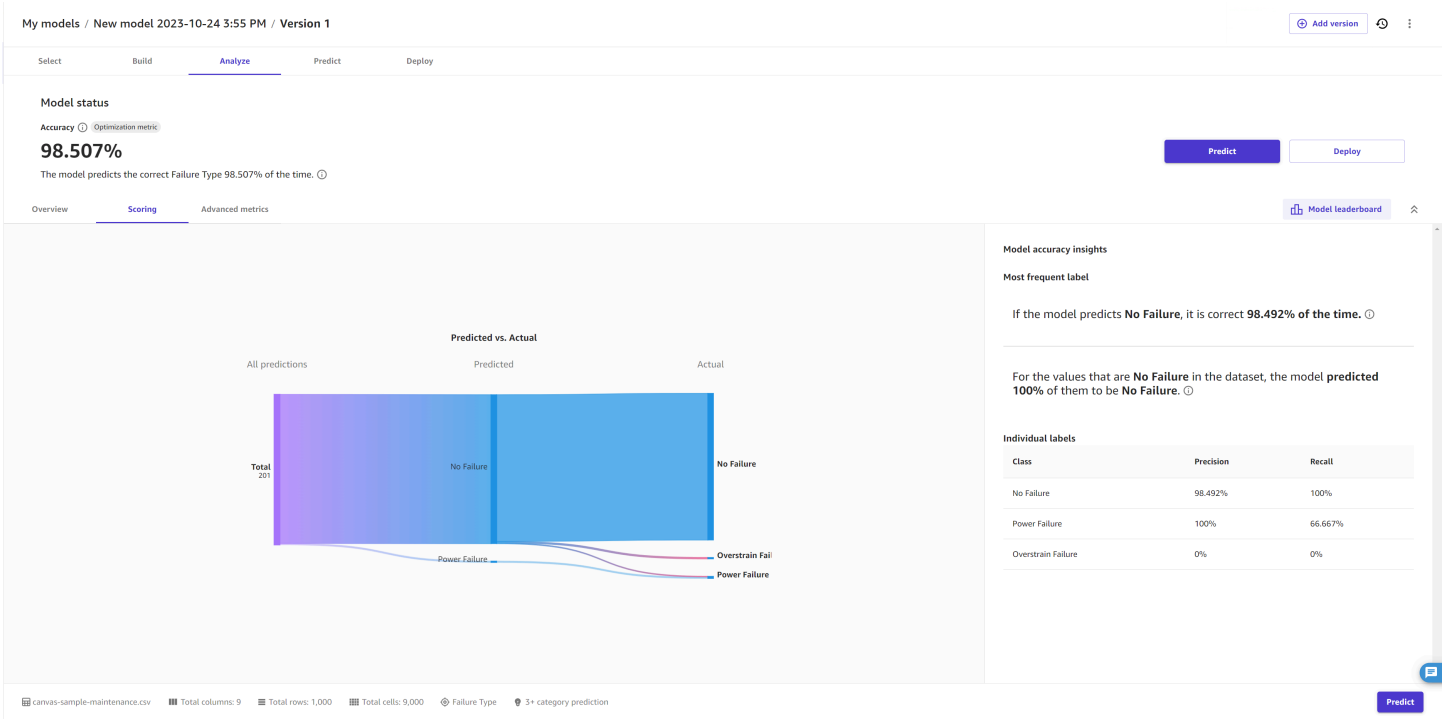
Tangkapan layar berikut menunjukkan skor Akurasi untuk model, bersama dengan metrik Optimasi, yang merupakan metrik yang Anda pilih untuk dioptimalkan saat membuat model. Dalam hal ini, metrik Optimasi adalah Akurasi. Anda dapat menentukan metrik pengoptimalan yang berbeda jika Anda membuat versi baru model Anda.



Tab Skor untuk model prediksi kategoris memberi Anda kemampuan untuk memvisualisasikan semua prediksi. Segmen garis memanjang dari kiri halaman, menunjukkan semua prediksi yang dibuat model. Di tengah halaman, segmen garis bertemu pada segmen tegak lurus untuk menunjukkan proporsi setiap prediksi ke satu kategori. Dari kategori yang diprediksi, segmen

bercabang ke kategori aktual. Anda bisa mendapatkan gambaran visual tentang seberapa akurat prediksi tersebut dengan mengikuti setiap segmen baris dari kategori yang diprediksi ke kategori aktual.

Gambar berikut memberi Anda contoh bagian Penilaian untuk model prediksi kategori 3+.

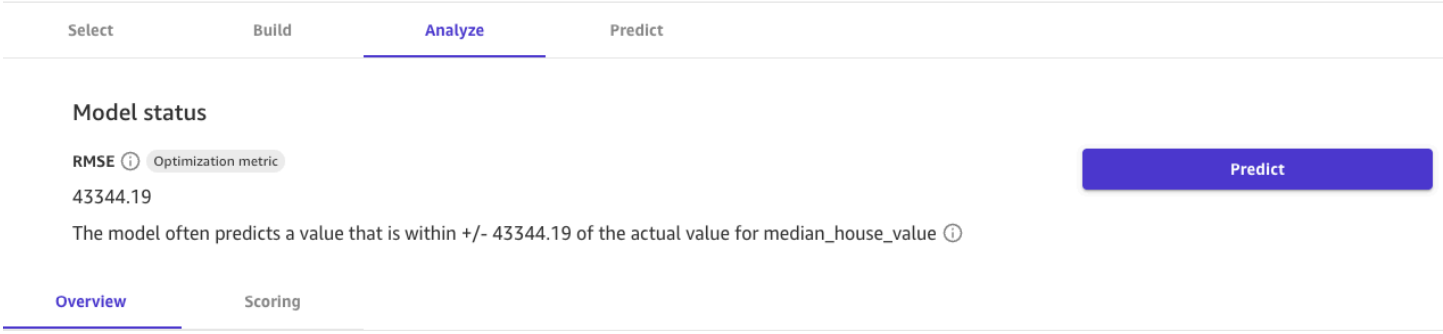


Anda juga dapat melihat tab Metrik lanjutan untuk informasi lebih rinci tentang kinerja model Anda, seperti metrik lanjutan, plot kepadatan kesalahan, atau matriks kebingungan. Untuk mempelajari lebih lanjut tentang tab Metrik lanjutan, lihatGunakan metrik lanjutan dalam analisis Anda.

Mengevaluasi model prediksi numerik

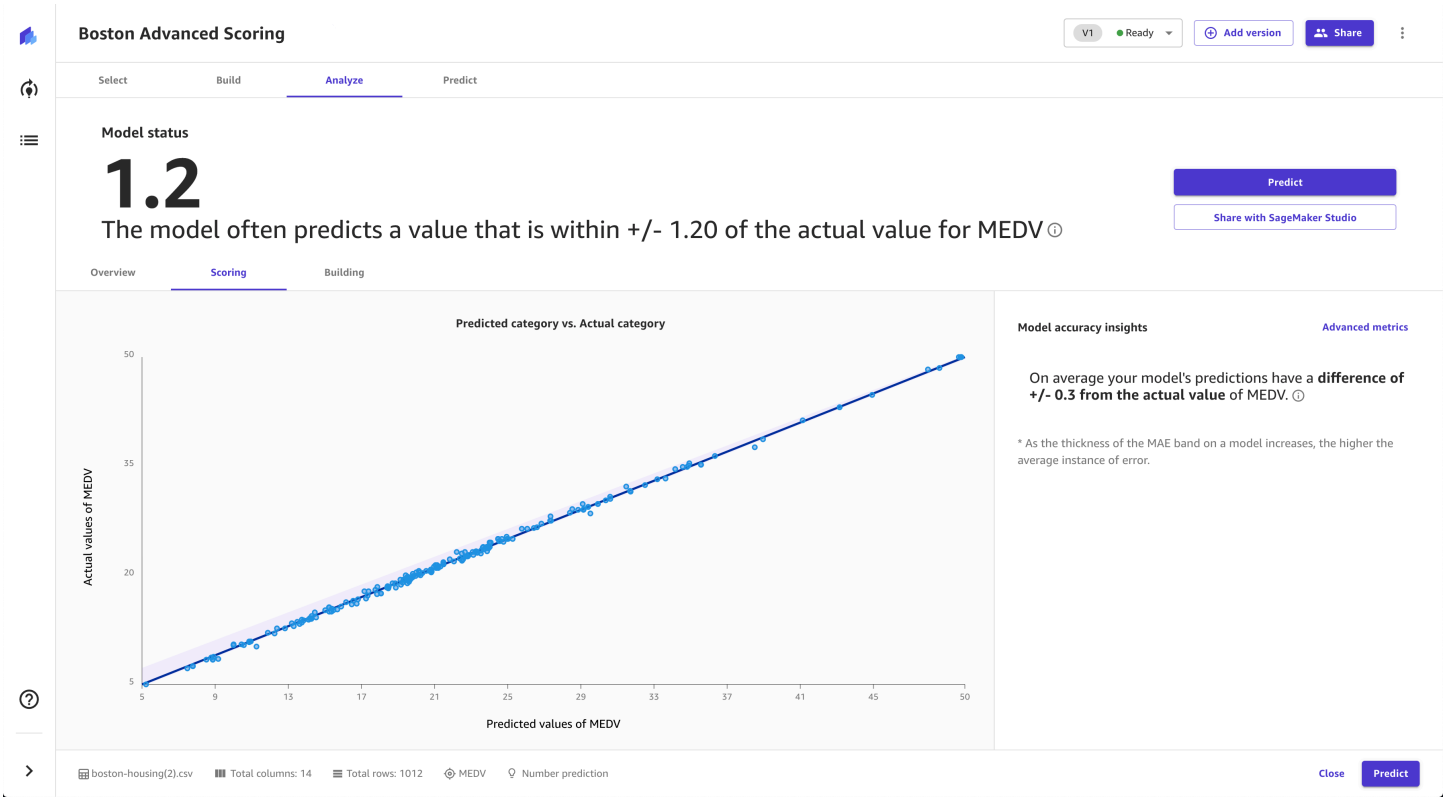
Tab Ikhtisar menunjukkan dampak kolom untuk setiap kolom. Dampak kolom adalah skor persentase yang menunjukkan berapa banyak bobot kolom dalam membuat prediksi dalam kaitannya dengan kolom lainnya. Untuk dampak kolom 25%, Canvas menimbang prediksi sebagai 25% untuk kolom dan 75% untuk kolom lainnya.

Tangkapan layar berikut menunjukkan skor RMSE untuk model pada tab Ikhtisar, yang dalam hal ini adalah metrik Optimasi. Metrik Optimasi adalah metrik yang Anda pilih untuk dioptimalkan saat membuat model. Anda dapat menentukan metrik pengoptimalan yang berbeda jika Anda membuat versi baru model Anda.



Tab Skor untuk prediksi numerik menunjukkan garis untuk menunjukkan nilai prediksi model dalam kaitannya dengan data yang digunakan untuk membuat prediksi. Nilai prediksi numerik sering +/- nilai RMSE (root mean squared error). Nilai yang diprediksi model seringkali berada dalam kisaran RMSE. Lebar pita ungu di sekitar garis menunjukkan kisaran RMSE. Nilai yang diprediksi sering berada dalam kisaran.

Gambar berikut menunjukkan bagian Skor untuk prediksi numerik.



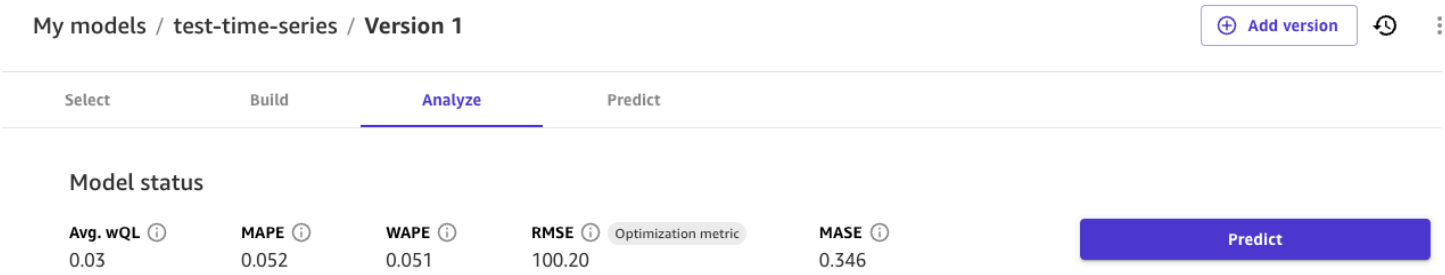
Anda juga dapat melihat tab Metrik lanjutan untuk informasi lebih rinci tentang kinerja model Anda, seperti metrik lanjutan, plot kepadatan kesalahan, atau matriks kebingungan. Untuk mempelajari lebih lanjut tentang tab Metrik lanjutan, lihatGunakan metrik lanjutan dalam analisis Anda.

Mengevaluasi model peramalan deret waktu

Pada halaman Analisis untuk model peramalan deret waktu, Anda dapat melihat ikhtisar metrik model. Anda dapat mengarahkan kursor ke setiap metrik untuk informasi lebih lanjut, atau Anda dapat melihatnyaGunakan metrik lanjutan dalam analisis Anda.

Di bagian Dampak kolom, Anda dapat melihat skor untuk setiap kolom. Dampak kolom adalah skor persentase yang menunjukkan berapa banyak bobot kolom dalam membuat prediksi dalam kaitannya dengan kolom lainnya. Untuk dampak kolom 25%, Canvas menimbang prediksi sebagai 25% untuk kolom dan 75% untuk kolom lainnya.

Tangkapan layar berikut menunjukkan skor metrik deret waktu untuk model, bersama dengan metrik Optimasi, yang merupakan metrik yang Anda pilih untuk dioptimalkan saat membuat model. Dalam hal ini, metrik Optimasi adalah RMSE. Anda dapat menentukan metrik pengoptimalan yang berbeda jika Anda membuat versi baru model Anda.



Evaluasi model prediksi gambar

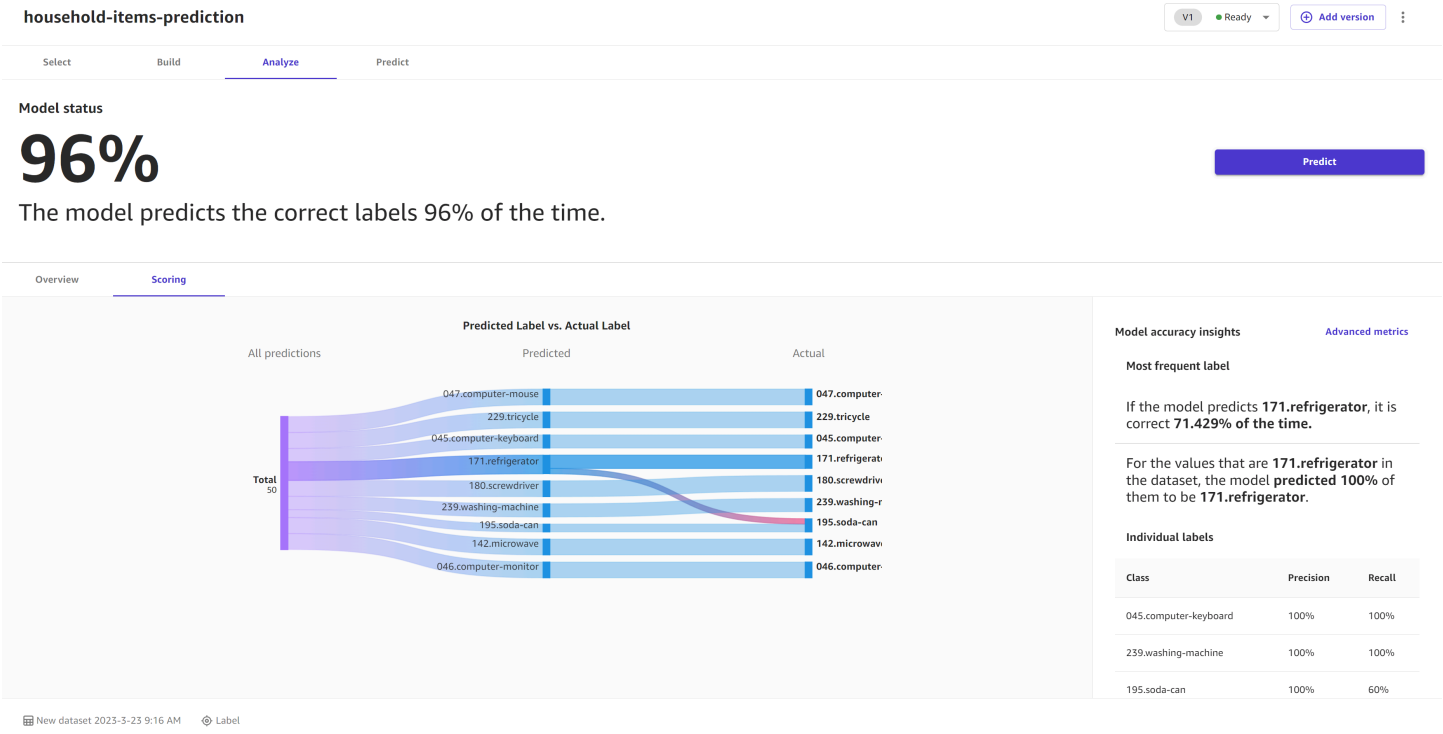
Tab Ikhtisar menunjukkan kinerja Per label, yang memberi Anda skor akurasi keseluruhan untuk gambar yang diprediksi untuk setiap label. Anda dapat memilih label untuk melihat detail yang lebih spesifik, seperti gambar yang diprediksi dengan benar dan diprediksi salah untuk label.

Anda dapat mengaktifkan sakelar Heatmap untuk melihat peta panas untuk setiap gambar. Peta panas menunjukkan kepada Anda bidang minat yang memiliki dampak paling besar saat model Anda membuat prediksi. Untuk informasi selengkapnya tentang heatmap dan cara menggunakannya untuk menyempurnakan model Anda, pilih ikon Info selengkapnya di sebelah Heatmap toggle.

Tab Skor untuk model prediksi gambar label tunggal menunjukkan kepada Anda perbandingan dari apa yang diprediksi model sebagai label versus label sebenarnya. Anda dapat memilih hingga 10 label sekaligus. Anda dapat mengubah label dalam visualisasi dengan memilih menu tarik-turun label dan memilih atau membatalkan pilihan label.

Anda juga dapat melihat wawasan untuk masing-masing label atau grup label, seperti tiga label dengan akurasi tertinggi atau terendah, dengan memilih menu tarik-turun Lihat skor untuk menu tarik-turun di bagian Wawasan akurasi model.

Tangkapan layar berikut menunjukkan Informasi penilaian untuk model prediksi gambar label tunggal.



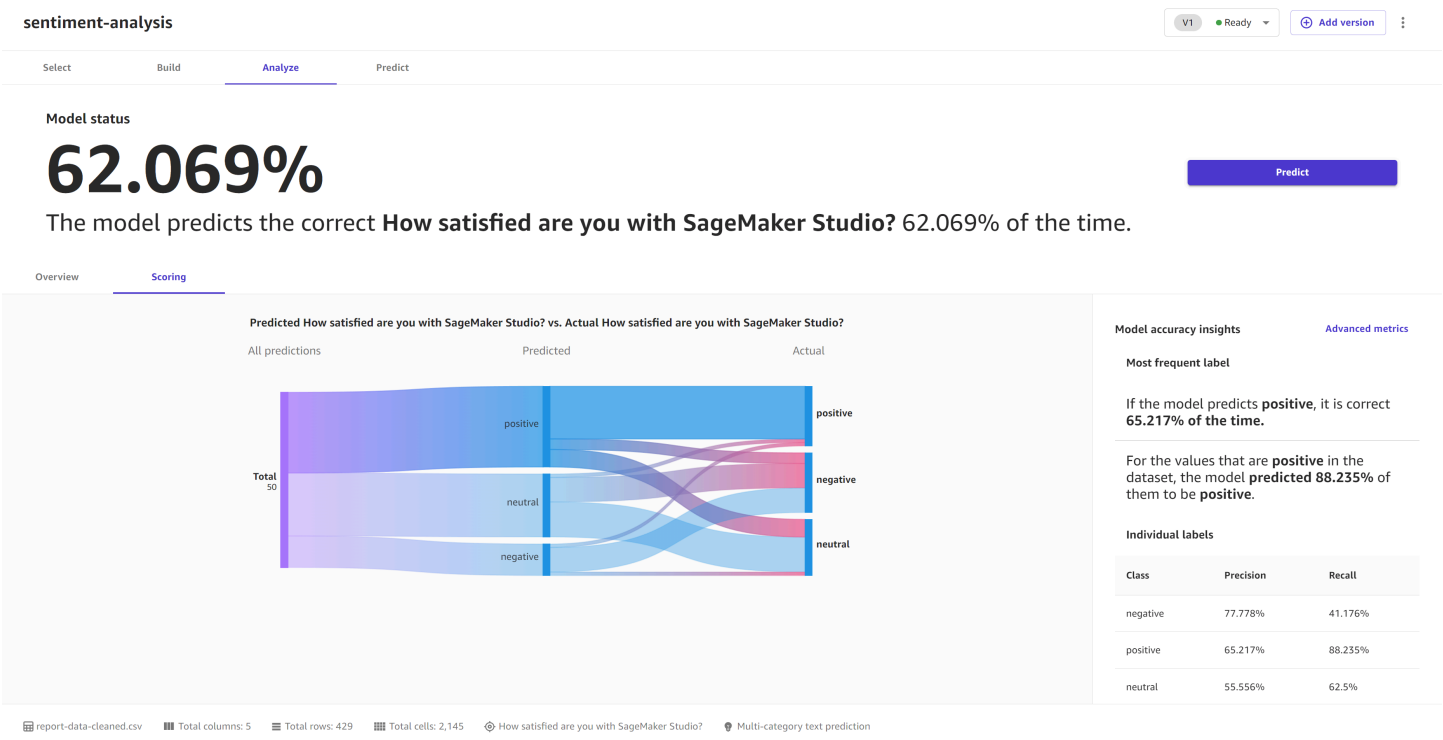
Evaluasi model prediksi teks

Tab Ikhtisar menunjukkan kinerja Per label, yang memberi Anda skor akurasi keseluruhan untuk bagian teks yang diprediksi untuk setiap label. Anda dapat memilih label untuk melihat detail yang lebih spesifik, seperti bagian yang diprediksi dengan benar dan diprediksi salah untuk label.

Tab Skor untuk model prediksi teks multi-kategori menunjukkan kepada Anda perbandingan apa yang diprediksi model sebagai label versus label sebenarnya.

Di bagian Wawasan akurasi model, Anda dapat melihat kategori Paling sering, yang memberi tahu Anda kategori yang paling sering diprediksi model dan seberapa akurat prediksi tersebut. Jika model Anda memprediksi label Positif dengan benar 99% dari waktu, maka Anda dapat cukup yakin bahwa model Anda pandai memprediksi sentimen positif dalam teks.

Tangkapan layar berikut menunjukkan Informasi penilaian untuk model prediksi teks multi-kategori.



Gunakan metrik lanjutan dalam analisis Anda

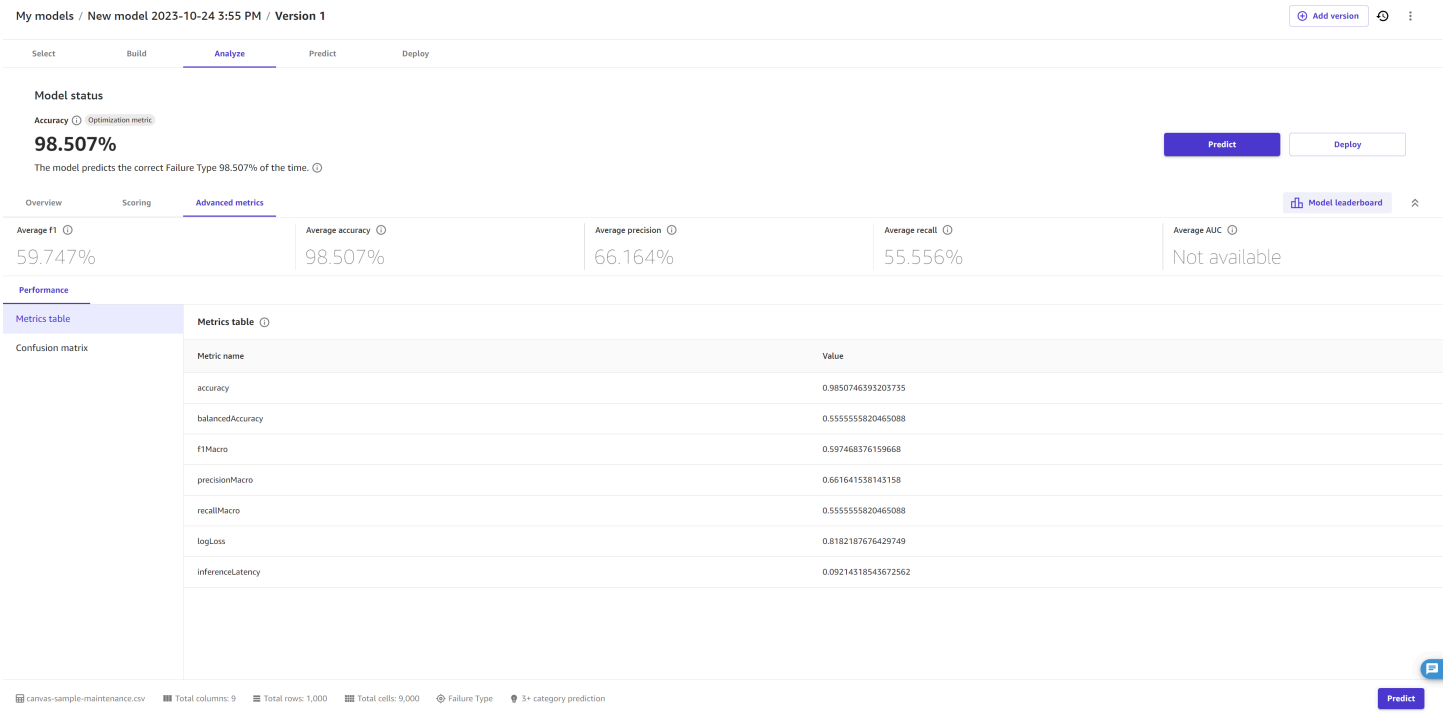
Bagian berikut menjelaskan cara menemukan dan menafsirkan metrik lanjutan untuk model Anda di Amazon SageMaker Canvas.

> **Note**
>
> Metrik lanjutan saat ini hanya tersedia untuk model prediksi numerik dan kategoris.

Untuk menemukan tab Metrik lanjutan, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.
3. Pilih model yang Anda buat.
4. Di panel navigasi atas, pilih tab Analisis.
5. Di dalam tab Analisis, pilih tab Metrik lanjutan.

Di tab Metrik lanjutan, Anda dapat menemukan tab Kinerja. Halaman terlihat seperti tangkapan layar berikut.



Di bagian atas, Anda dapat melihat ikhtisar skor metrik, termasuk metrik Optimasi, yang merupakan metrik yang Anda pilih (atau Canvas yang dipilih secara default) untuk dioptimalkan saat membuat model.

Bagian berikut menjelaskan informasi lebih rinci untuk tab Performance dalam metrik Advanced.

Performa

Di tab Performance, Anda akan melihat tabel Metrik, bersama dengan visualisasi yang dibuat Canvas berdasarkan jenis model Anda. Untuk model prediksi kategoris, Canvas menyediakan matriks kebingungan, sedangkan untuk model prediksi numerik, Canvas memberi Anda residu dan bagan kepadatan kesalahan.

Dalam tabel Metrik, Anda diberikan daftar lengkap skor model Anda untuk setiap metrik lanjutan, yang lebih komprehensif daripada ikhtisar skor di bagian atas halaman. Metrik yang ditampilkan di sini bergantung pada jenis model Anda. Untuk referensi untuk membantu Anda memahami dan menafsirkan setiap metrik, lihatReferensi metrik.

Untuk memahami visualisasi yang mungkin muncul berdasarkan jenis model Anda, lihat opsi berikut:

- Matriks kebingungan - Canvas menggunakan matriks kebingungan untuk membantu Anda memvisualisasikan ketika model membuat prediksi dengan benar. Dalam matriks kebingungan, hasil Anda disusun untuk membandingkan nilai yang diprediksi dengan nilai aktual. Contoh berikut

menjelaskan bagaimana matriks kebingungan bekerja untuk model prediksi kategori 2 yang memprediksi label positif dan negatif:

- Benar positif — Model dengan benar memprediksi positif ketika label sebenarnya positif.
- Benar negatif — Model dengan benar memprediksi negatif ketika label sebenarnya negatif.
- Positif palsu — Model salah memprediksi positif ketika label sebenarnya negatif.
- False negative — Model salah memprediksi negatif ketika label sebenarnya positif.

- Kurva recall presisi — Kurva recall presisi adalah visualisasi skor presisi model yang diplot terhadap skor recall model. Umumnya, model yang dapat membuat prediksi sempurna akan memiliki skor presisi dan ingatan yang keduanya 1. Kurva recall presisi untuk model yang cukup akurat akan cukup tinggi baik dalam presisi maupun recall.
- Residu — Residu adalah perbedaan antara nilai aktual dan nilai yang diprediksi oleh model. Bagan residu memplot residu terhadap nilai yang sesuai untuk memvisualisasikan distribusinya dan pola atau outlier apa pun. Distribusi normal residu di sekitar nol menunjukkan bahwa model tersebut cocok untuk data. Namun, jika residu miring secara signifikan atau memiliki outlier, ini mungkin menunjukkan bahwa model tersebut terlalu sesuai dengan data atau bahwa ada masalah lain yang perlu ditangani.
- Kepadatan kesalahan — Plot kepadatan kesalahan adalah representasi dari distribusi kesalahan yang dibuat oleh model. Ini menunjukkan kepadatan probabilitas kesalahan di setiap titik, membantu Anda mengidentifikasi area mana pun di mana model mungkin terlalu pas atau membuat kesalahan sistematis.

Lihat kandidat model di papan peringkat model

Saat Anda membuat model di Amazon SageMaker Canvas, SageMaker latih beberapa kandidat model, atau iterasi model yang berbeda, dan pilih model dengan nilai tertinggi untuk metrik pengoptimalan secara default. Kandidat model default adalah satu-satunya versi yang dapat Anda gunakan dengan fungsionalitas lain di Canvas seperti membuat prediksi, mendaftar ke registri model, atau menyebarkan ke titik akhir.

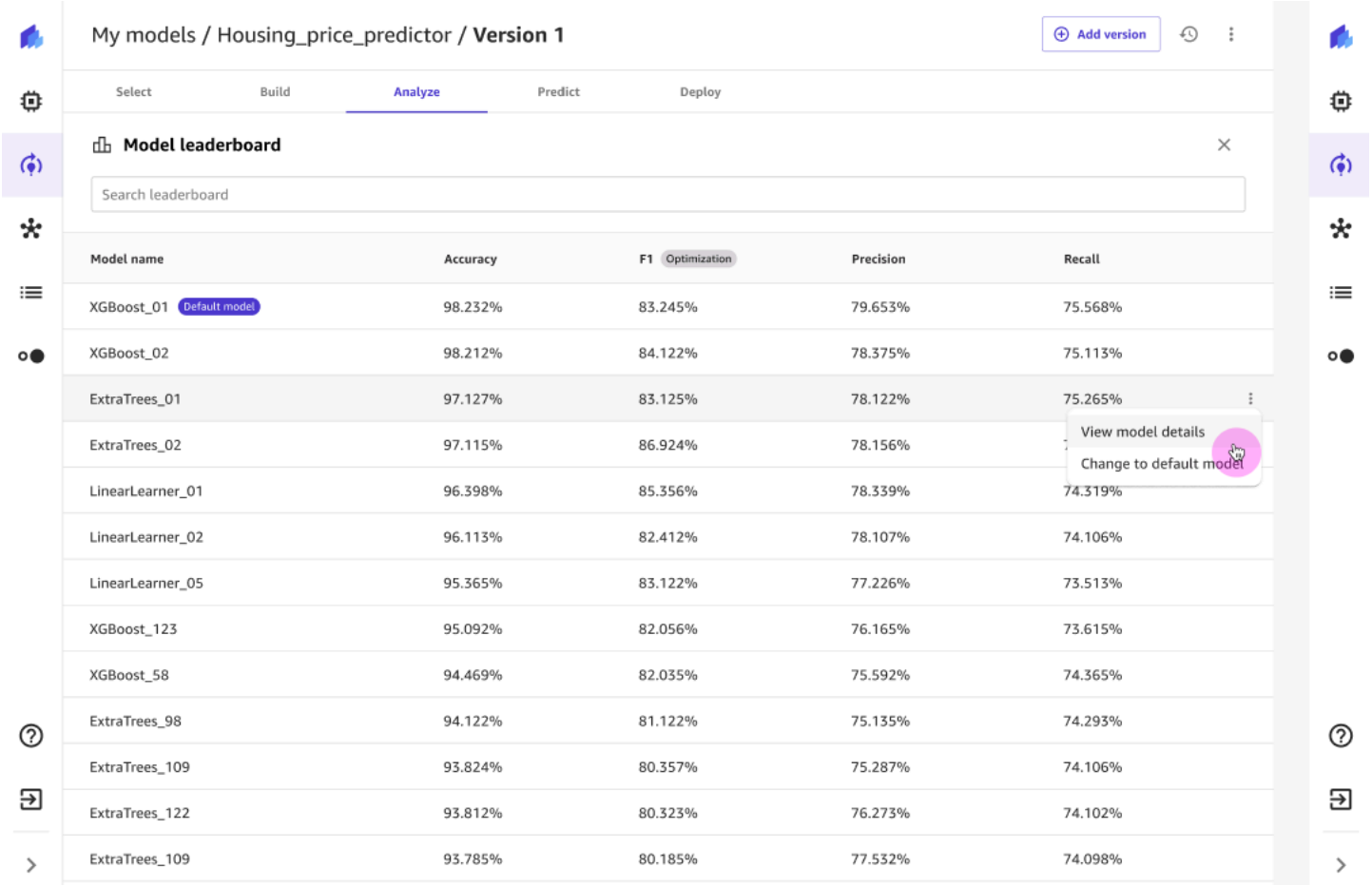
Namun, Anda mungkin ingin meninjau semua kandidat model dan memilih kandidat yang berbeda untuk menjadi model default. Anda dapat melihat semua kandidat model dan detail lebih lanjut tentang setiap kandidat di papan peringkat Model di Canvas.

Untuk melihat papan peringkat Model, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.

2. Di panel navigasi kiri, pilih Model saya.
3. Pilih model yang Anda buat.
4. Di panel navigasi atas, pilih tab Analisis.
5. Di dalam tab Analisis, pilih Papan peringkat Model.

Halaman papan peringkat Model terbuka, yang terlihat seperti tangkapan layar berikut.



Anda dapat melihat bahwa kandidat model pertama yang terdaftar ditandai sebagai model Default. Ini adalah kandidat model yang dengannya Anda dapat membuat prediksi atau menyebarkan ke titik akhir.

Untuk melihat informasi metrik yang lebih rinci tentang kandidat model untuk membandingkannya, Anda dapat memilih ikon Opsi lainnya (⋮) dan memilih Lihat detail model.

> **Important**
>
> Memuat detail model untuk kandidat model non-default mungkin memakan waktu beberapa menit (biasanya kurang dari 10 menit), dan biaya SageMaker Hosting berlaku. Untuk informasi selengkapnya, silakan lihat Harga SageMaker.

Kandidat model terbuka di tab Analisis, dan metrik yang ditampilkan khusus untuk kandidat model tersebut. Setelah selesai meninjau metrik kandidat model, Anda dapat kembali atau keluar dari tampilan untuk kembali ke papan peringkat Model.

Jika Anda ingin menyetel model Default ke kandidat yang berbeda, Anda dapat memilih ikon Opsi lainnya
(⋮)
dan memilih Ubah ke model default. Mengubah model default untuk model yang dilatih menggunakan mode HPO mungkin memakan waktu beberapa menit.

> **Note**
>
> Jika model Anda sudah digunakan dalam produksi, terdaftar ke registri model, atau memiliki otomatisasi yang disiapkan, Anda harus menghapus penerapan, pendaftaran model, atau otomatisasi sebelum mengubah model default.

Referensi metrik

Bagian berikut menjelaskan metrik yang tersedia di Amazon SageMaker Canvas untuk setiap jenis model.

Metrik untuk prediksi numerik

Berikut ini mendefinisikan metrik untuk prediksi numerik di SageMaker Canvas dan memberi Anda informasi tentang bagaimana Anda dapat menggunakannya.

- InferenceLatency — Perkiraan jumlah waktu antara membuat permintaan untuk prediksi model untuk menerimanya dari titik akhir waktu nyata di mana model digunakan. Metrik ini diukur dalam hitungan detik dan hanya tersedia untuk model yang dibuat dengan mode Ensembling.
- MAE - Berarti kesalahan absolut. Rata-rata, prediksi untuk kolom target adalah +/- {MAE} dari nilai sebenarnya.

Mengukur seberapa berbeda nilai prediksi dan aktual, ketika dirata-ratakan di semua nilai. MAE biasanya digunakan dalam prediksi numerik untuk memahami kesalahan prediksi model. Jika prediksi linier, MAE mewakili jarak rata-rata dari garis prediksi ke nilai aktual. MAE didefinisikan sebagai jumlah kesalahan absolut dibagi dengan jumlah pengamatan. Nilai berkisar dari 0 hingga tak terhingga, dengan angka yang lebih kecil menunjukkan kecocokan model yang lebih baik dengan data.

- MAPE — Berarti kesalahan persen absolut. Rata-rata, prediksi untuk kolom target adalah +/- {MAPE}% dari nilai sebenarnya.

  MAPE adalah rata-rata perbedaan absolut antara nilai aktual dan nilai yang diprediksi atau diperkirakan, dibagi dengan nilai aktual dan dinyatakan sebagai persentase. MAPE yang lebih rendah menunjukkan kinerja yang lebih baik, karena itu berarti bahwa nilai yang diprediksi atau diperkirakan lebih dekat dengan nilai aktual.
- MSE — Rata-rata kesalahan kuadrat, atau rata-rata perbedaan kuadrat antara nilai prediksi dan aktual.

  Nilai MSE selalu positif. Semakin baik model dalam memprediksi nilai aktual, semakin kecil nilai MSE.
- R2 — Persentase perbedaan pada kolom target yang dapat dijelaskan oleh kolom input.

  Mengukur seberapa banyak model dapat menjelaskan varians dari variabel dependen. Nilai berkisar dari satu (1) ke negatif (-1). Angka yang lebih tinggi menunjukkan fraksi yang lebih tinggi dari variabilitas yang dijelaskan. Nilai mendekati nol (0) menunjukkan bahwa sangat sedikit variabel dependen yang dapat dijelaskan oleh model. Nilai negatif menunjukkan kecocokan yang buruk dan bahwa model tersebut dikalahkan oleh fungsi konstan (atau garis horizontal).
- RMSE — Root mean kuadrat kesalahan, atau standar deviasi kesalahan.

  Mengukur akar kuadrat dari perbedaan kuadrat antara nilai prediksi dan aktual, dan dirata-ratakan pada semua nilai. Ini digunakan untuk memahami kesalahan prediksi model, dan ini merupakan metrik penting untuk menunjukkan adanya kesalahan model besar dan outlier. Nilai berkisar dari nol (0) hingga tak terhingga, dengan angka yang lebih kecil menunjukkan kecocokan model yang lebih baik dengan data. RMSE tergantung pada skala, dan tidak boleh digunakan untuk membandingkan kumpulan data dari berbagai jenis.

Metrik untuk prediksi kategoris

Bagian ini mendefinisikan metrik untuk prediksi kategoris di SageMaker Canvas dan memberi Anda informasi tentang bagaimana Anda dapat menggunakannya.

Berikut ini adalah daftar metrik yang tersedia untuk prediksi kategori 2:

- Akurasi — Persentase prediksi yang benar.

  Atau, rasio jumlah item yang diprediksi dengan benar dengan jumlah total prediksi. Akurasi mengukur seberapa dekat nilai kelas yang diprediksi dengan nilai aktual. Nilai untuk metrik akurasi bervariasi antara nol (0) dan satu (1). Nilai 1 menunjukkan akurasi sempurna, dan 0 menunjukkan ketidakakuratan total.
- AUC — Nilai antara 0 dan 1 yang menunjukkan seberapa baik model Anda dapat memisahkan kategori dalam kumpulan data Anda. Nilai 1 menunjukkan bahwa ia mampu memisahkan kategori dengan sempurna.
- BalancedAccuracy — Mengukur rasio prediksi yang akurat untuk semua prediksi.

  Rasio ini dihitung setelah menormalkan positif sejati (TP) dan negatif sejati (TN) dengan jumlah total nilai positif (P) dan negatif (N). Ini didefinisikan sebagai berikut:`0.5*((TP/P)+(TN/N))`, dengan nilai mulai dari 0 hingga 1. Metrik akurasi seimbang memberikan ukuran akurasi yang lebih baik ketika jumlah positif atau negatif sangat berbeda satu sama lain dalam kumpulan data yang tidak seimbang, seperti ketika hanya 1% email adalah spam.
- F1 — Ukuran akurasi yang seimbang yang memperhitungkan keseimbangan kelas.

  Ini adalah rata-rata harmonik dari skor presisi dan ingatan, yang didefinisikan sebagai berikut:`F1 = 2 * (precision * recall) / (precision + recall)`. Skor F1 bervariasi antara 0 dan 1. Skor 1 menunjukkan kinerja terbaik, dan 0 menunjukkan yang terburuk.
- InferenceLatency — Perkiraan jumlah waktu antara membuat permintaan untuk prediksi model untuk menerimanya dari titik akhir waktu nyata di mana model digunakan. Metrik ini diukur dalam hitungan detik dan hanya tersedia untuk model yang dibuat dengan mode Ensembling.
- LogLoss Kehilangan log, juga dikenal sebagai kerugian lintas entropi, adalah metrik yang digunakan untuk mengevaluasi kualitas output probabilitas, bukan output itu sendiri. Kehilangan log adalah metrik penting untuk menunjukkan kapan model membuat prediksi yang salah dengan probabilitas tinggi. Nilai berkisar dari 0 hingga tak terbatas. Nilai 0 mewakili model yang memprediksi data dengan sempurna.
- Presisi — Dari semua waktu yang {kategori x} diprediksi, prediksi itu benar {presisi}% dari waktu.

Presisi mengukur seberapa baik suatu algoritma memprediksi positif sejati (TP) dari semua hal positif yang diidentifikasi. Ini didefinisikan sebagai berikut:`Precision = TP/(TP+FP)`, dengan nilai mulai dari nol (0) hingga satu (1). Presisi adalah metrik penting ketika biaya positif palsu tinggi. Misalnya, biaya positif palsu sangat tinggi jika sistem keselamatan pesawat secara keliru dianggap aman untuk terbang. Positif palsu (FP) mencerminkan prediksi positif yang sebenarnya negatif dalam data.

- Ingat — Model dengan benar memprediksi {recall}% menjadi {category x} ketika {target_column} sebenarnya adalah {category x}.

  Ingat mengukur seberapa baik algoritme memprediksi dengan benar semua positif sejati (TP) dalam kumpulan data. Positif sejati adalah prediksi positif yang juga merupakan nilai positif aktual dalam data. Ingat didefinisikan sebagai berikut:`Recall = TP/(TP+FN)`, dengan nilai mulai dari 0 hingga 1. Skor yang lebih tinggi mencerminkan kemampuan model yang lebih baik untuk memprediksi positif sejati (TP) dalam data. Perhatikan bahwa seringkali tidak cukup untuk hanya mengukur ingatan, karena memprediksi setiap output sebagai positif sejati menghasilkan skor ingatan yang sempurna.

Berikut ini adalah daftar metrik yang tersedia untuk prediksi kategori 3+:

- Akurasi — Persentase prediksi yang benar.

  Atau, rasio jumlah item yang diprediksi dengan benar dengan jumlah total prediksi. Akurasi mengukur seberapa dekat nilai kelas yang diprediksi dengan nilai aktual. Nilai untuk metrik akurasi bervariasi antara nol (0) dan satu (1). Nilai 1 menunjukkan akurasi sempurna, dan 0 menunjukkan ketidakakuratan total.

- BalancedAccuracy — Mengukur rasio prediksi yang akurat untuk semua prediksi.

  Rasio ini dihitung setelah menormalkan positif sejati (TP) dan negatif sejati (TN) dengan jumlah total nilai positif (P) dan negatif (N). Ini didefinisikan sebagai berikut:`0.5*((TP/P)+(TN/N))`, dengan nilai mulai dari 0 hingga 1. Metrik akurasi seimbang memberikan ukuran akurasi yang lebih baik ketika jumlah positif atau negatif sangat berbeda satu sama lain dalam kumpulan data yang tidak seimbang, seperti ketika hanya 1% email adalah spam.

- F1makro — Skor F1makro menerapkan penilaian F1 dengan menghitung presisi dan recall, dan kemudian mengambil mean harmoniknya untuk menghitung skor F1 untuk setiap kelas. Kemudian, F1makro rata-rata skor individu untuk mendapatkan skor F1makro. Skor F1makro bervariasi antara 0 dan 1. Skor 1 menunjukkan kinerja terbaik, dan 0 menunjukkan yang terburuk.

- InferenceLatency — Perkiraan jumlah waktu antara membuat permintaan untuk prediksi model untuk menerimanya dari titik akhir waktu nyata di mana model digunakan. Metrik ini diukur dalam hitungan detik dan hanya tersedia untuk model yang dibuat dengan mode Ensembling.
- LogLoss Kehilangan log, juga dikenal sebagai kerugian lintas entropi, adalah metrik yang digunakan untuk mengevaluasi kualitas output probabilitas, bukan output itu sendiri. Kehilangan log adalah metrik penting untuk menunjukkan kapan model membuat prediksi yang salah dengan probabilitas tinggi. Nilai berkisar dari 0 hingga tak terbatas. Nilai 0 mewakili model yang memprediksi data dengan sempurna.
- PrecisionMacro — Mengukur presisi dengan menghitung presisi untuk setiap kelas dan skor rata-rata untuk mendapatkan presisi untuk beberapa kelas. Skor berkisar dari nol (0) hingga satu (1). Skor yang lebih tinggi mencerminkan kemampuan model untuk memprediksi positif sejati (TP) dari semua positif yang diidentifikasi, dirata-ratakan di beberapa kelas.
- RecallMacro — Mengukur penarikan kembali dengan menghitung penarikan untuk setiap kelas dan skor rata-rata untuk mendapatkan penarikan kembali untuk beberapa kelas. Skor berkisar dari 0 hingga 1. Skor yang lebih tinggi mencerminkan kemampuan model untuk memprediksi positif sejati (TP) dalam kumpulan data, sedangkan positif sejati mencerminkan prediksi positif yang juga merupakan nilai positif aktual dalam data. Seringkali tidak cukup untuk mengukur hanya ingatan, karena memprediksi setiap output sebagai positif sejati akan menghasilkan skor ingatan yang sempurna.

Perhatikan bahwa untuk prediksi kategori 3+, Anda juga menerima metrik rata-rata F1, Akurasi, Presisi, dan Ingat. Skor untuk metrik ini hanyalah skor metrik yang dirata-ratakan untuk semua kategori.

Metrik untuk prediksi gambar dan teks

Berikut ini adalah daftar metrik yang tersedia untuk prediksi gambar dan prediksi teks.

- Akurasi — Persentase prediksi yang benar.

  Atau, rasio jumlah item yang diprediksi dengan benar dengan jumlah total prediksi. Akurasi mengukur seberapa dekat nilai kelas yang diprediksi dengan nilai aktual. Nilai untuk metrik akurasi bervariasi antara nol (0) dan satu (1). Nilai 1 menunjukkan akurasi sempurna, dan 0 menunjukkan ketidakakuratan total.
- F1 — Ukuran akurasi yang seimbang yang memperhitungkan keseimbangan kelas.

Ini adalah rata-rata harmonik dari skor presisi dan ingatan, yang didefinisikan sebagai berikut:`F1 = 2 * (precision * recall) / (precision + recall)`. Skor F1 bervariasi antara 0 dan 1. Skor 1 menunjukkan kinerja terbaik, dan 0 menunjukkan yang terburuk.

- Presisi — Dari semua waktu yang {kategori x} diprediksi, prediksi itu benar {presisi}% dari waktu.

  Presisi mengukur seberapa baik suatu algoritma memprediksi positif sejati (TP) dari semua hal positif yang diidentifikasi. Ini didefinisikan sebagai berikut:`Precision = TP/(TP+FP)`, dengan nilai mulai dari nol (0) hingga satu (1). Presisi adalah metrik penting ketika biaya positif palsu tinggi. Misalnya, biaya positif palsu sangat tinggi jika sistem keselamatan pesawat secara keliru dianggap aman untuk terbang. Positif palsu (FP) mencerminkan prediksi positif yang sebenarnya negatif dalam data.
- Ingat — Model dengan benar memprediksi {recall}% menjadi {category x} ketika {target_column} sebenarnya adalah {category x}.

  Ingat mengukur seberapa baik algoritme memprediksi dengan benar semua positif sejati (TP) dalam kumpulan data. Positif sejati adalah prediksi positif yang juga merupakan nilai positif aktual dalam data. Ingat didefinisikan sebagai berikut:`Recall = TP/(TP+FN)`, dengan nilai mulai dari 0 hingga 1. Skor yang lebih tinggi mencerminkan kemampuan model yang lebih baik untuk memprediksi positif sejati (TP) dalam data. Perhatikan bahwa seringkali tidak cukup untuk hanya mengukur ingatan, karena memprediksi setiap output sebagai positif sejati menghasilkan skor ingatan yang sempurna.

Perhatikan bahwa untuk model prediksi gambar dan teks tempat Anda memprediksi 3 kategori atau lebih, Anda juga menerima metrik F1, Akurasi, Presisi, dan Ingat rata-rata. Skor untuk metrik ini hanyalah rata-rata skor metrik untuk semua kategori.

Metrik untuk perkiraan deret waktu

Berikut ini mendefinisikan metrik lanjutan untuk perkiraan deret waktu di Amazon SageMaker Canvas dan memberi Anda informasi tentang bagaimana Anda dapat menggunakannya.

- Average Weighted Quantile Loss (wQL) — Mengevaluasi perkiraan dengan rata-rata akurasi pada kuantil P10, P50, dan P90. Nilai yang lebih rendah menunjukkan model yang lebih akurat.
- Weighted Absolute Percent Error (WAPE) — Jumlah kesalahan absolut yang dinormalisasi dengan jumlah target absolut, yang mengukur deviasi keseluruhan nilai yang diperkirakan dari nilai yang diamati. Nilai yang lebih rendah menunjukkan model yang lebih akurat, di mana WAPE = 0 adalah model tanpa kesalahan.

- Root Mean Square Error (RMSE) - Akar kuadrat dari kesalahan kuadrat rata-rata. RMSE yang lebih rendah menunjukkan model yang lebih akurat, di mana RMSE = 0 adalah model tanpa kesalahan.
- Mean Absolute Percent Error (MAPE) — Persentase kesalahan (perbedaan persen dari nilai perkiraan rata-rata versus nilai aktual) dirata-ratakan pada semua titik waktu. Nilai yang lebih rendah menunjukkan model yang lebih akurat, di mana MAPE = 0 adalah model tanpa kesalahan.
- Mean Absolute Scaled Error (MASE) — Kesalahan absolut rata-rata dari perkiraan dinormalisasi oleh kesalahan absolut rata-rata dari metode peramalan dasar sederhana. Nilai yang lebih rendah menunjukkan model yang lebih akurat, di mana MASE < 1 is estimated to be better than the baseline and MASE > 1 diperkirakan lebih buruk daripada baseline.

Buat prediksi untuk data Anda

Gunakan model kustom yang telah Anda buat di SageMaker Canvas untuk membuat prediksi untuk data Anda. Bagian berikut menunjukkan cara membuat prediksi untuk model prediksi numerik dan kategoris, model prediksi gambar, dan model prediksi teks. Untuk informasi tentang cara membuat prediksi dengan model perkiraan deret waktu, lihat Membuat perkiraan deret waktu.

Prediksi numerik dan kategoris, prediksi gambar, dan model kustom prediksi teks mendukung pembuatan jenis prediksi berikut untuk data Anda:

- Prediksi tunggal — Prediksi tunggal adalah ketika Anda hanya perlu membuat satu prediksi. Misalnya, Anda memiliki satu gambar atau bagian teks yang ingin Anda klasifikasikan.
- Prediksi Batch — Prediksi Batch adalah saat Anda ingin membuat prediksi untuk seluruh kumpulan data. Misalnya, Anda memiliki file CSV ulasan pelanggan yang ingin Anda prediksi sentimen pelanggan, atau Anda memiliki folder file gambar yang ingin Anda klasifikasikan. Anda harus membuat prediksi dengan dataset yang cocok dengan dataset input Anda. Canvas memberi Anda kemampuan untuk melakukan prediksi batch manual, atau Anda dapat mengonfigurasi prediksi batch otomatis yang dimulai setiap kali kumpulan data tertentu diperbarui di Canvas.

Untuk setiap prediksi atau set prediksi, SageMaker Canvas mengembalikan yang berikut:

- Nilai yang diprediksi
- Probabilitas nilai prediksi benar

Memulai

Pilih salah satu alur kerja berikut untuk membuat prediksi dengan model kustom Anda:

- Buat prediksi batch
- Buat prediksi tunggal

Setelah menghasilkan prediksi dengan model Anda, Anda juga dapat melakukan hal berikut:

- Perbarui model Anda dengan membuat versi baru. Jika Anda ingin mencoba meningkatkan akurasi prediksi model Anda, Anda dapat membuat versi baru model Anda. Anda dapat memperbarui data Anda atau mengubah transformasi lanjutan apa pun yang Anda gunakan, dan kemudian Anda dapat meninjau dan membandingkan versi model Anda untuk memilih yang terbaik.
- Daftarkan versi model di registri SageMaker model. Anda dapat mendaftarkan versi model Anda ke registri SageMaker model, yang merupakan fitur untuk melacak dan mengelola status versi model dan pipeline pembelajaran mesin. Seorang ilmuwan data atau pengguna tim MLOP dengan akses ke registri SageMaker model dapat meninjau versi model Anda dan menyetujui atau menolaknya sebelum menerapkannya ke produksi.
- Kirim prediksi batch Anda ke Amazon QuickSight. Di Amazon QuickSight, Anda dapat membuat dan menerbitkan dasbor dengan kumpulan data prediksi batch Anda. Ini dapat membantu Anda menganalisis dan membagikan hasil yang dihasilkan oleh model kustom Anda.

Buat prediksi tunggal

> **Note**
>
> Bagian ini menjelaskan cara mendapatkan prediksi tunggal dari model Anda di dalam aplikasi Canvas. Untuk informasi tentang membuat pemanggilan real-time di lingkungan produksi dengan menerapkan model Anda ke titik akhir, lihat. Menerapkan model Anda ke titik akhir

Buat prediksi tunggal jika Anda ingin mendapatkan prediksi untuk satu titik data. Anda dapat menggunakan fitur ini untuk mendapatkan prediksi waktu nyata atau bereksperimen dengan mengubah nilai individu untuk melihat bagaimana pengaruhnya terhadap hasil prediksi.

Pilih salah satu prosedur berikut berdasarkan jenis model Anda.

Buat prediksi tunggal dengan model prediksi numerik dan kategoris

Untuk membuat prediksi tunggal untuk model prediksi numerik atau kategoris, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih Model saya.

2. Pada halaman Model saya, pilih model Anda.
3. Setelah membuka model Anda, pilih tab Predict.
4. Pada halaman prediksi Jalankan, pilih Prediksi tunggal.
5. Untuk setiap kolom Kolom, yang mewakili kolom data masukan Anda, Anda dapat mengubah Nilai. Pilih menu tarik-turun untuk Nilai yang ingin Anda ubah. Untuk bidang numerik, Anda dapat memasukkan nomor baru. Untuk bidang dengan label, Anda dapat memilih label yang berbeda.
6. Saat Anda siap menghasilkan prediksi, di panel Prediksi kanan, pilih Perbarui.

Di panel Prediksi kanan, Anda akan melihat hasil prediksi. Anda dapat Menyalin bagan hasil prediksi, atau Anda juga dapat memilih Unduh untuk mengunduh bagan hasil prediksi sebagai gambar atau mengunduh nilai dan prediksi sebagai file CSV.

Buat prediksi tunggal dengan model prediksi gambar

Untuk membuat prediksi tunggal untuk model prediksi gambar label tunggal, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih Model saya.
2. Pada halaman Model saya, pilih model Anda.
3. Setelah membuka model Anda, pilih tab Predict.
4. Pada halaman prediksi Jalankan, pilih Prediksi tunggal.
5. Pilih Impor gambar.
6. Anda akan diminta untuk mengunggah gambar. Anda dapat mengunggah gambar dari komputer lokal Anda atau dari bucket Amazon S3.
7. Pilih Impor untuk mengimpor gambar Anda dan menghasilkan prediksi.

Di panel hasil Prediksi kanan, model mencantumkan label yang mungkin untuk gambar bersama dengan skor Keyakinan untuk setiap label. Misalnya, model mungkin memprediksi label Sea untuk gambar, dengan skor kepercayaan 96%. Model tersebut mungkin telah memprediksi gambar sebagai Gletser dengan hanya skor kepercayaan 4%. Karena itu, Anda dapat menentukan bahwa model Anda cukup percaya diri dalam memprediksi gambar laut.

Buat prediksi tunggal dengan model prediksi teks

Untuk membuat prediksi tunggal untuk model prediksi teks multi-kategori, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih Model saya.

2. Pada halaman Model saya, pilih model Anda.
3. Setelah membuka model Anda, pilih tab Predict.
4. Pada halaman prediksi Jalankan, pilih Prediksi tunggal.
5. Untuk bidang Teks, masukkan teks yang ingin Anda prediksi.
6. Pilih Hasilkan hasil prediksi untuk mendapatkan prediksi Anda.

Di panel hasil Prediksi kanan, Anda menerima analisis teks Anda selain skor Keyakinan untuk setiap label yang mungkin. Misalnya, jika Anda memasukkan ulasan yang baik untuk suatu produk, Anda mungkin mendapatkan Positif dengan skor kepercayaan 85%, sedangkan skor kepercayaan untuk Netral mungkin 10% dan skor kepercayaan untuk Negatif hanya 5%.

Buat prediksi batch

Buat prediksi batch saat Anda memiliki seluruh kumpulan data yang ingin Anda hasilkan prediksi.

Ada dua jenis prediksi batch yang dapat Anda buat:

- Prediksi batch manual adalah ketika Anda memiliki kumpulan data yang ingin Anda buat prediksi satu kali.
- Prediksi batch otomatis adalah saat Anda menyiapkan konfigurasi yang menjalankan prediksi batch setiap kali kumpulan data tertentu diperbarui. Misalnya, jika Anda telah mengonfigurasi pembaruan mingguan ke kumpulan data inventaris SageMaker Canvas, Anda dapat mengatur prediksi batch otomatis yang berjalan setiap kali Anda memperbarui kumpulan data. Setelah menyiapkan alur kerja prediksi batch otomatis, lihat Kelola otomatisasi untuk informasi selengkapnya tentang melihat dan mengedit detail konfigurasi Anda. Untuk informasi selengkapnya tentang menyiapkan pembaruan kumpulan data otomatis, lihatKonfigurasikan pembaruan otomatis untuk kumpulan data.

> **Note**
>
> Anda hanya dapat mengatur prediksi batch otomatis untuk kumpulan data yang diimpor melalui unggahan lokal atau Amazon S3. Selain itu, prediksi batch otomatis hanya dapat berjalan saat Anda masuk ke aplikasi Canvas. Jika Anda keluar dari Canvas, pekerjaan prediksi batch otomatis akan dilanjutkan saat Anda masuk kembali.

Untuk memulai, tinjau bagian berikut untuk persyaratan kumpulan data prediksi batch, lalu pilih salah satu alur kerja prediksi batch manual atau otomatis berikut.

Persyaratan dataset prediksi Batch

Untuk prediksi batch, pastikan kumpulan data Anda memenuhi persyaratan yang diuraikan dalam. Buat kumpulan data

Anda mungkin tidak dapat membuat prediksi pada beberapa kumpulan data karena mereka memiliki skema yang tidak kompatibel. Skema adalah struktur organisasi. Untuk kumpulan data tabular, skema adalah nama kolom dan tipe data data di kolom. Skema yang tidak kompatibel mungkin terjadi karena salah satu alasan berikut:

- Dataset yang Anda gunakan untuk membuat prediksi memiliki lebih sedikit kolom daripada kumpulan data yang Anda gunakan untuk membangun model.
- Tipe data di kolom yang Anda gunakan untuk membangun kumpulan data mungkin berbeda dari tipe data dalam kumpulan data yang Anda gunakan untuk membuat prediksi.
- Dataset yang Anda gunakan untuk membuat prediksi dan kumpulan data yang Anda gunakan untuk membangun model memiliki nama kolom yang tidak cocok. Nama kolom peka huruf besar/kecil. `Column1`tidak sama dengan`column1`.

Untuk memastikan bahwa Anda berhasil menghasilkan prediksi batch, cocokkan skema kumpulan data prediksi batch Anda dengan kumpulan data yang Anda gunakan untuk melatih model.

> **Note**
>
> Untuk prediksi batch, jika Anda menjatuhkan kolom apa pun saat membuat model, Canvas menambahkan kolom yang dijatuhkan kembali ke hasil prediksi. Namun, Canvas tidak menambahkan kolom yang dijatuhkan ke prediksi batch Anda untuk model deret waktu.

Buat prediksi batch manual

Pilih salah satu prosedur berikut untuk membuat prediksi batch manual berdasarkan jenis model Anda.

Buat prediksi batch manual dengan model prediksi numerik dan kategoris

Untuk membuat prediksi batch manual untuk model prediksi numerik atau kategoris, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih Model saya.
2. Pada halaman Model saya, pilih model Anda.
3. Setelah membuka model Anda, pilih tab Predict.
4. Pada halaman prediksi Jalankan, pilih prediksi Batch.
5. Pilih Pilih kumpulan data jika Anda telah mengimpor dataset Anda. Jika tidak, pilih Impor dataset baru, dan kemudian Anda akan diarahkan melalui alur kerja data impor.
6. Dari daftar kumpulan data yang tersedia, pilih kumpulan data Anda dan pilih Hasilkan prediksi untuk mendapatkan prediksi Anda.

Setelah pekerjaan prediksi selesai berjalan, pada halaman Jalankan prediksi,
Anda akan melihat kumpulan data keluaran yang tercantum di bawah Prediksi.
Kumpulan data ini berisi hasil Anda, dan jika Anda memilih ikon Opsi lainnya
(⋮),
Anda dapat memilih Pratinjau untuk melihat pratinjau data keluaran. Anda dapat melihat data input yang cocok dengan prediksi dan probabilitas prediksi itu benar. Kemudian, Anda dapat memilih Unduh prediksi untuk mengunduh hasilnya sebagai file.

Buat prediksi batch manual dengan model prediksi gambar

Untuk membuat prediksi batch manual untuk model prediksi gambar label tunggal, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih Model saya.
2. Pada halaman Model saya, pilih model Anda.
3. Setelah membuka model Anda, pilih tab Predict.
4. Pada halaman prediksi Jalankan, pilih prediksi Batch.
5. Pilih Pilih kumpulan data jika Anda telah mengimpor dataset Anda. Jika tidak, pilih Impor dataset baru, dan kemudian Anda akan diarahkan melalui alur kerja data impor.
6. Dari daftar kumpulan data yang tersedia, pilih kumpulan data Anda dan pilih Hasilkan prediksi untuk mendapatkan prediksi Anda.

Setelah pekerjaan prediksi selesai berjalan, pada halaman Jalankan prediksi, Anda akan melihat kumpulan data keluaran yang tercantum di bawah Prediksi. Kumpulan data ini berisi hasil Anda, dan jika Anda memilih ikon Opsi lainnya (⋮), Anda dapat memilih Lihat hasil prediksi untuk melihat data keluaran. Anda dapat melihat gambar bersama dengan label prediksi dan skor kepercayaan mereka. Kemudian, Anda dapat memilih Unduh prediksi untuk mengunduh hasilnya sebagai file CSV atau ZIP.

Buat prediksi batch manual dengan model prediksi teks

Untuk membuat prediksi batch manual untuk model prediksi teks multi-kategori, lakukan hal berikut:

1. Di panel navigasi kiri aplikasi Canvas, pilih Model saya.
2. Pada halaman Model saya, pilih model Anda.
3. Setelah membuka model Anda, pilih tab Predict.
4. Pada halaman prediksi Jalankan, pilih prediksi Batch.
5. Pilih Pilih kumpulan data jika Anda telah mengimpor dataset Anda. Jika tidak, pilih Impor dataset baru, dan kemudian Anda akan diarahkan melalui alur kerja data impor. Dataset yang Anda pilih harus memiliki kolom sumber yang sama dengan kumpulan data yang Anda gunakan untuk membuat model.
6. Dari daftar kumpulan data yang tersedia, pilih kumpulan data Anda dan pilih Hasilkan prediksi untuk mendapatkan prediksi Anda.

Setelah pekerjaan prediksi selesai berjalan, pada halaman Jalankan prediksi, Anda akan melihat kumpulan data keluaran yang tercantum di bawah Prediksi. Kumpulan data ini berisi hasil Anda, dan jika Anda memilih ikon Opsi lainnya (⋮), Anda dapat memilih Pratinjau untuk melihat data keluaran. Anda dapat melihat gambar bersama dengan label prediksi dan skor kepercayaan mereka. Kemudian, Anda dapat memilih Unduh prediksi untuk mengunduh hasilnya.

Buat prediksi batch otomatis

Untuk mengatur jadwal prediksi batch otomatis, lakukan hal berikut:

1. Di panel navigasi kiri Canvas, pilih Model saya.
2. Pilih model Anda.

3. Pilih tab Predict.
4. Pilih prediksi Batch.
5. Untuk Menghasilkan prediksi, pilih Otomatis.
6. Kotak dialog Automate batch predictions muncul. Pilih Pilih kumpulan data dan pilih kumpulan data yang ingin Anda otomatiskan prediksi. Perhatikan bahwa Anda hanya dapat memilih kumpulan data yang diimpor melalui unggahan lokal atau Amazon S3.
7. Setelah memilih kumpulan data, pilih Siapkan.

Canvas menjalankan pekerjaan prediksi batch untuk kumpulan data setelah Anda mengatur konfigurasi. Kemudian, setiap kali AndaMemperbarui kumpulan data, baik secara manual atau otomatis, pekerjaan prediksi batch lain berjalan.

Setelah pekerjaan prediksi selesai berjalan, pada halaman Jalankan prediksi,
Anda akan melihat kumpulan data keluaran yang tercantum di bawah Prediksi.
Kumpulan data ini berisi hasil Anda, dan jika Anda memilih ikon Opsi lainnya
(⋮),
Anda dapat memilih Pratinjau untuk melihat pratinjau data keluaran. Anda dapat melihat data input yang cocok dengan prediksi dan probabilitas prediksi itu benar. Kemudian, Anda dapat memilih Unduh untuk mengunduh hasilnya.

Bagian berikut menjelaskan cara melihat, memperbarui, dan menghapus konfigurasi prediksi batch otomatis Anda melalui halaman Datasets di aplikasi Canvas. Anda hanya dapat mengatur maksimum 20 konfigurasi otomatis di Canvas. Untuk informasi selengkapnya tentang melihat riwayat pekerjaan prediksi batch otomatis atau membuat perubahan pada konfigurasi otomatis melalui halaman Otomasi, lihat. Kelola otomatisasi

Lihat pekerjaan prediksi batch otomatis Anda

Untuk melihat riwayat pekerjaan Anda untuk prediksi batch otomatis Anda, buka tab Predict model Anda.

Setiap pekerjaan prediksi batch otomatis muncul di tab Prediksi model Anda. Di bawah Prediksi, Anda dapat melihat tab Semua pekerjaan dan tab Konfigurasi:

- Semua pekerjaan — Di tab ini, Anda dapat melihat semua pekerjaan prediksi batch untuk model ini. Anda dapat memfilter pekerjaan berdasarkan nama konfigurasi. Untuk setiap pekerjaan, Anda

dapat melihat bidang seperti kumpulan data Input, yang mencakup versi kumpulan data, dan jenis Prediksi, seperti apakah prediksi itu otomatis atau manual. Jika Anda memilih ikon Opsi lainnya ( ), Anda dapat memilih Lihat prediksi atau Unduh prediksi.

- Konfigurasi — Di tab ini, Anda dapat melihat semua konfigurasi prediksi batch otomatis yang Anda buat untuk model ini. Untuk setiap konfigurasi, Anda dapat melihat bidang seperti stempel waktu saat dibuat, kumpulan data Input yang dilacaknya untuk pembaruan, dan pekerjaan Berikutnya dijadwalkan. Jika Anda memilih ikon Opsi lainnya ( ), Anda dapat memilih Lihat semua pekerjaan untuk melihat riwayat pekerjaan dan pekerjaan yang sedang berlangsung untuk konfigurasi.

Edit konfigurasi prediksi batch otomatis Anda

Anda mungkin ingin membuat perubahan pada konfigurasi pembaruan otomatis untuk kumpulan data, seperti mengubah frekuensi pembaruan. Anda mungkin juga ingin menonaktifkan konfigurasi pembaruan otomatis untuk menjeda pembaruan pada kumpulan data Anda.

Saat Anda mengedit konfigurasi prediksi batch, Anda dapat mengubah kumpulan data target tetapi bukan frekuensinya (karena prediksi batch otomatis terjadi setiap kali kumpulan data diperbarui).

Untuk mengedit konfigurasi pembaruan otomatis, lakukan hal berikut:

1. Buka tab Predict model Anda.
2. Di bawah Prediksi, pilih tab Konfigurasi.
3. Temukan konfigurasi Anda dan pilih ikon Opsi lainnya ( ).
4. Dari menu tarik-turun, pilih Perbarui konfigurasi.
5. Kotak dialog Automate batch prediction terbuka. Anda dapat memilih kumpulan data lain dan memilih Mengatur untuk menyimpan perubahan Anda.

Konfigurasi prediksi batch otomatis Anda sekarang diperbarui.

Untuk menjeda prediksi batch otomatis Anda, matikan konfigurasi otomatis Anda dengan melakukan hal berikut:

1. Buka tab Predict model Anda.

2. Di bawah Prediksi, pilih tab Konfigurasi.
3. Temukan konfigurasi Anda dari daftar dan matikan sakelar Pembaruan otomatis.

Prediksi batch otomatis sekarang dijeda. Anda dapat mengaktifkan kembali sakelar kapan saja untuk melanjutkan jadwal pembaruan.

Hapus konfigurasi prediksi batch otomatis Anda

Untuk mempelajari cara menghapus konfigurasi prediksi batch otomatis, lihatHapus konfigurasi otomatis.

Anda juga dapat menghapus konfigurasi Anda dengan melakukan hal berikut:

1. Buka tab Predict model Anda.
2. Di bawah Prediksi, pilih tab Konfigurasi.
3. Temukan konfigurasi Anda dari daftar dan pilih ikon Opsi lainnya (⋮).
4. Dari menu tarik-turun, pilih Hapus konfigurasi.

Konfigurasi Anda sekarang harus dihapus.

Kirim prediksi ke Amazon QuickSight

> **Note**
>
> Anda dapat mengirim prediksi batch ke Amazon QuickSight untuk prediksi numerik dan kategoris serta model peramalan deret waktu. Anda juga dapat mengirim prediksi yang dihasilkan dengan model BYOM. Prediksi gambar label tunggal dan model prediksi teks multi-kategori dikecualikan.

Setelah Anda membuat prediksi batch dengan model tabular khusus di SageMaker Canvas, Anda dapat mengirim prediksi tersebut sebagai file CSV ke Amazon QuickSight, yang merupakan layanan intelijen bisnis (BI) untuk membangun dan menerbitkan dasbor prediktif.

Misalnya, jika Anda membuat model prediksi kategori 2 untuk menentukan apakah pelanggan akan melakukan churn, Anda dapat membuat dasbor visual dan prediktif QuickSight untuk menunjukkan

persentase pelanggan yang diharapkan untuk melakukan churn. Untuk mempelajari lebih lanjut tentang Amazon QuickSight, lihat Panduan QuickSight Pengguna Amazon.

Bagian berikut menunjukkan cara mengirim prediksi batch Anda QuickSight untuk analisis.

Sebelum Anda memulai

Pengguna Anda harus memiliki izin AWS Identity and Access Management (IAM) yang diperlukan untuk mengirim prediksi Anda. QuickSight Administrator Anda dapat mengatur izin IAM untuk pengguna Anda. Untuk informasi selengkapnya, lihat Berikan Izin Pengguna Anda untuk Mengirim Prediksi ke Amazon QuickSight.

QuickSight Akun Anda harus berisi `default` namespace, yang diatur saat Anda pertama kali membuat akun QuickSight . Hubungi administrator Anda untuk membantu Anda mendapatkan akses ke QuickSight. Untuk informasi selengkapnya, lihat Menyiapkan Amazon QuickSight di Panduan QuickSight Pengguna Amazon.

QuickSight Akun Anda harus dibuat di Wilayah yang sama dengan aplikasi Canvas Anda. Jika Wilayah beranda QuickSight akun Anda berbeda dari Wilayah aplikasi Canvas Anda, Anda harus menutup dan membuat ulang QuickSight akun Anda, atau menyiapkan aplikasi Canvas di Wilayah yang sama dengan QuickSight akun Anda. Anda dapat memeriksa Wilayah QuickSight asal Anda dengan melakukan hal berikut (dengan asumsi Anda sudah memiliki QuickSight akun):

1. Buka QuickSight konsol Anda.
2. Saat halaman dimuat, Wilayah QuickSight beranda Anda ditambahkan ke URL dalam format berikut:`https://<your-home-region>.quicksight.aws.amazon.com/`.

Anda harus mengetahui nama pengguna QuickSight pengguna yang ingin Anda kirimi prediksi Anda. Anda dapat mengirim prediksi kepada diri sendiri atau pengguna lain yang memiliki izin yang tepat. Setiap pengguna yang Anda kirimi prediksi harus berada di `default` namespace QuickSight akun Anda dan memiliki peran `Author` atau `Admin` peran di dalamnya. QuickSight

Selain itu, QuickSight harus memiliki akses ke bucket Amazon S3 SageMaker default untuk Domain Anda, yang diberi nama dengan format berikut:. `sagemaker-{REGION}-{ACCOUNT_ID}` Wilayah harus sama dengan Wilayah asal QuickSight akun Anda dan Wilayah aplikasi Canvas Anda. Untuk mempelajari cara memberikan QuickSight akses ke prediksi batch yang disimpan di bucket Amazon S3 Anda, lihat topik yang tidak dapat saya sambungkan ke Amazon S3 di Panduan Pengguna Amazon. QuickSight

Format data yang didukung

Sebelum mengirim prediksi Anda, periksa apakah format data prediksi batch Anda kompatibel dengan. QuickSight

- Untuk mempelajari lebih lanjut tentang format data yang diterima untuk data timeseries, lihat Format tanggal yang didukung di Panduan Pengguna Amazon QuickSight .
- Untuk mempelajari lebih lanjut tentang nilai data yang mungkin mencegah Anda mengirim QuickSight, lihat Nilai yang tidak didukung dalam data di Panduan QuickSight Pengguna Amazon.

Perhatikan juga bahwa Amazon QuickSight menggunakan karakter " sebagai kualifikasi teks, jadi jika data Canvas Anda berisi " karakter apa pun, pastikan Anda menutup semua tanda kutip yang cocok. Setiap kutipan yang tidak cocok dapat menyebabkan masalah dengan pengiriman dataset Anda ke. QuickSight

Kirim prediksi batch Anda ke QuickSight

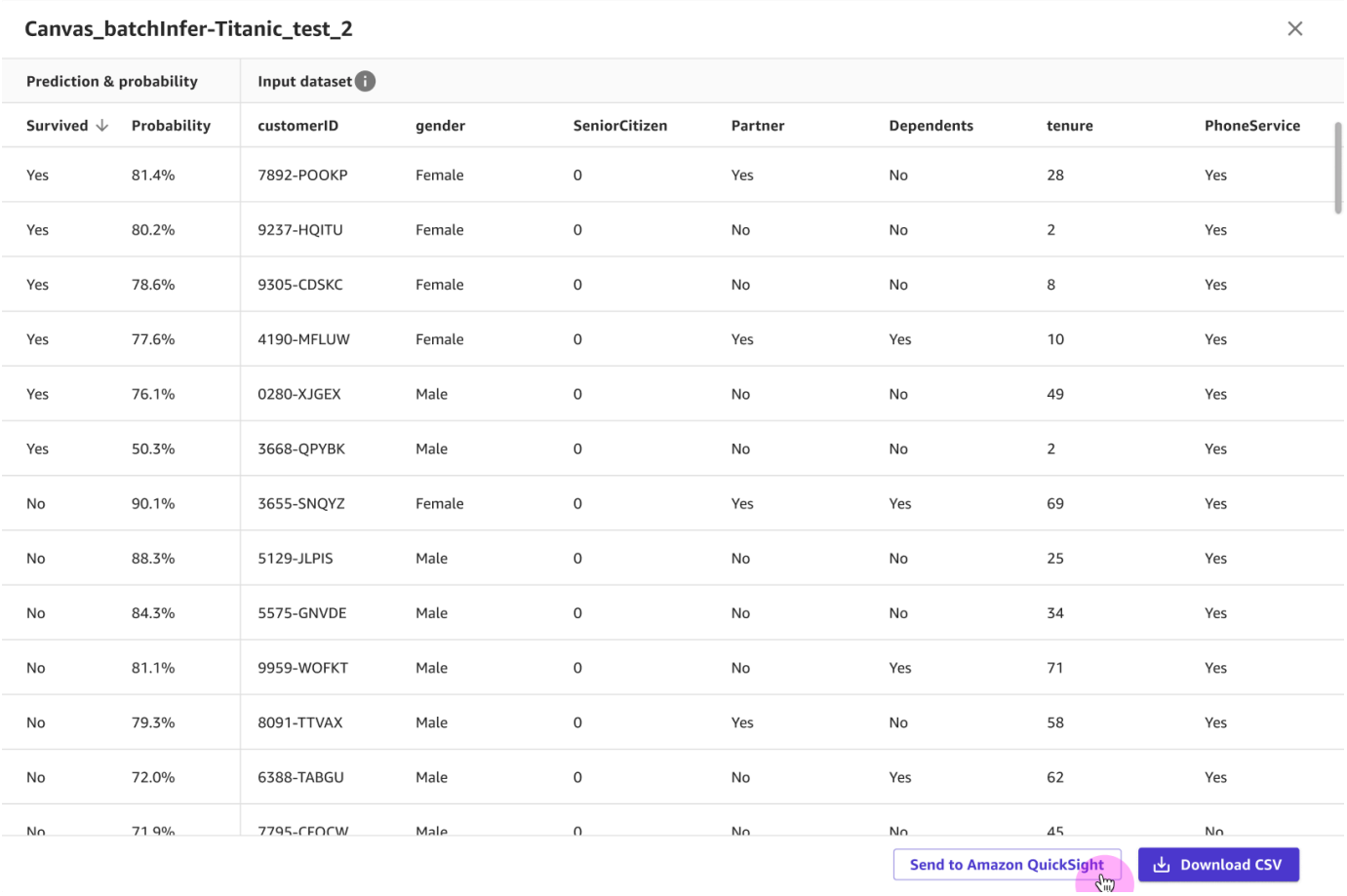
Gunakan prosedur berikut untuk mengirim prediksi Anda ke QuickSight:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.
3. Pada halaman Model saya, pilih model Anda.
4. Pilih tab Predict.
5. Di bawah Prediksi, pilih kumpulan data (atau kumpulan data) prediksi batch yang ingin Anda bagikan. Anda dapat membagikan hingga 5 kumpulan data prediksi batch sekaligus.
6. Setelah memilih kumpulan data, pilih Kirim ke Amazon QuickSight.

> **Note**
>
> QuickSightTombol Kirim ke Amazon tidak aktif kecuali Anda memilih satu atau beberapa kumpulan data.

Atau, Anda dapat melihat pratinjau prediksi Anda dengan memilih ikon Opsi lainnya
(⋮)
dan kemudian Lihat hasil prediksi. Dari pratinjau dataset, Anda dapat memilih Kirim ke Amazon

QuickSight. Tangkapan layar berikut menunjukkan QuickSight tombol Kirim ke Amazon dalam pratinjau dataset.

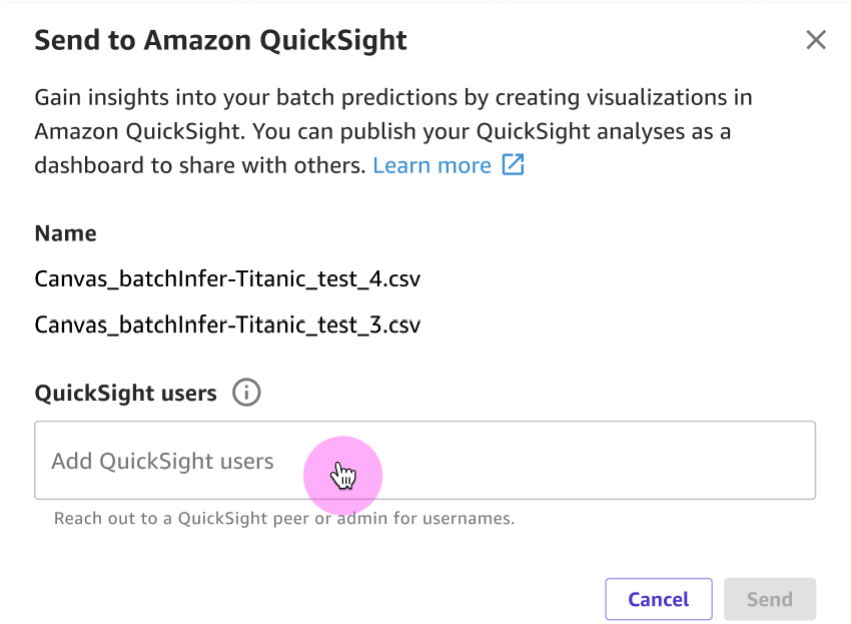


7. Dalam kotak QuickSight dialog Kirim ke Amazon, lakukan hal berikut:

   a. Untuk QuickSight pengguna, masukkan nama QuickSight pengguna yang ingin Anda kirimi prediksi Anda. Jika Anda ingin mengirimnya ke diri sendiri, masukkan nama pengguna Anda sendiri. Anda hanya dapat mengirim prediksi ke pengguna di `default` namespace QuickSight akun, dan pengguna harus memiliki peran `Author` atau`Admin`. QuickSight

   b. Pilih Kirim.

   Tangkapan layar berikut menunjukkan kotak QuickSight dialog Kirim ke Amazon:



Setelah Anda mengirim prediksi batch Anda, QuickSightbidang untuk kumpulan data yang Anda kirim akan ditampilkan sebagai. `Sent` Di kotak konfirmasi yang mengonfirmasi prediksi Anda dikirim, Anda dapat memilih Buka Amazon QuickSight untuk membuka QuickSight aplikasi Anda. Jika Anda selesai menggunakan Canvas, Anda harus keluar dari aplikasi Canvas.

QuickSight Pengguna yang Anda kirimi dataset dapat membuka QuickSight aplikasi mereka dan melihat dataset Canvas yang telah dibagikan dengan mereka. Kemudian, mereka dapat membuat dasbor prediktif dengan data. Untuk informasi selengkapnya, lihat Memulai analisis QuickSight data Amazon di Panduan QuickSight Pengguna Amazon.

Secara default, semua pengguna yang Anda kirimi prediksi memiliki izin pemilik untuk kumpulan data. QuickSight Pemilik dapat membuat analisis, menyegarkan, mengedit, menghapus, dan membagikan kembali kumpulan data. Perubahan yang dilakukan pemilik pada kumpulan data mengubah kumpulan data untuk semua pengguna yang memiliki akses. Untuk mengubah izin, buka kumpulan data QuickSight dan kelola izinnya. Untuk informasi selengkapnya, lihat Melihat dan mengedit izin pengguna yang dibagikan kumpulan data di QuickSight Panduan Pengguna Amazon.

Unduh notebook model

> **Note**
>
> Fitur notebook model hanya tersedia untuk model tabular dan model foundation fine-tuned. Notebook model tidak didukung untuk prediksi gambar, prediksi teks, atau model peramalan deret waktu.
> Jika Anda ingin membuat notebook model untuk model tabular yang dibuat sebelum fitur ini diluncurkan, Anda harus membangun kembali model untuk menghasilkan notebook.

Untuk model yang memenuhi syarat yang berhasil Anda buat di Amazon SageMaker Canvas, buku catatan Jupyter yang berisi laporan semua langkah pembuatan model dibuat. Notebook Jupyter ini berisi kode Python yang dapat Anda jalankan secara lokal atau jalankan di lingkungan seperti Amazon SageMaker Studio Classic untuk mereplikasi langkah-langkah yang diperlukan untuk membangun model Anda. Notebook dapat berguna jika Anda ingin bereksperimen dengan kode atau melihat detail backend tentang bagaimana Canvas membangun model.

Untuk mengakses notebook model, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.
3. Pilih model dan versi yang Anda buat.
4. Pada halaman versi model, pilih ikon Opsi lainnya (⋮) di header.
5. Dari menu tarik-turun, pilih Lihat buku catatan.
6. Sebuah popup muncul dengan konten notebook. Anda dapat memilih Download dan kemudian melakukan salah satu hal berikut:
   a. Pilih Unduh untuk menyimpan konten buku catatan ke perangkat lokal Anda.
   b. Pilih Salin URI S3 untuk menyalin lokasi Amazon S3 tempat notebook disimpan. Notebook disimpan di bucket Amazon S3 yang ditentukan dalam konfigurasi penyimpanan Canvas Anda, yang dikonfigurasi di bagian tersebut[Prasyarat untuk menyiapkan Amazon Canvas SageMaker].

Anda sekarang harus dapat melihat notebook baik secara lokal atau sebagai objek di Amazon S3. Anda dapat mengunggah buku catatan ke IDE untuk mengedit dan menjalankan kode, atau berbagi buku catatan dengan orang lain di organisasi untuk ditinjau.

Kirim model Anda ke Amazon QuickSight

Jika Anda menggunakan Amazon QuickSight dan ingin memanfaatkan SageMaker Canvas dalam QuickSight visualisasi, Anda dapat membuat model Amazon SageMaker Canvas dan menggunakannya sebagai bidang prediktif dalam kumpulan data Anda. QuickSight Bidang prediktif adalah bidang dalam QuickSight kumpulan data Anda yang dapat membuat prediksi untuk kolom tertentu dalam kumpulan data Anda, mirip dengan cara pengguna Canvas membuat prediksi tunggal atau batch dengan model. Untuk mempelajari selengkapnya tentang cara mengintegrasikan kemampuan prediksi Canvas ke dalam QuickSight kumpulan data Amazon Anda, lihat Integrasi SageMaker kanvas di Panduan Pengguna Amazon QuickSight .

Langkah-langkah berikut menjelaskan bagaimana Anda dapat menambahkan bidang prediktif ke QuickSight kumpulan data Anda menggunakan model Canvas:

1. Buka aplikasi Canvas dan buat model dengan dataset Anda.
2. Setelah membangun model di Canvas, kirim model ke QuickSight. File skema secara otomatis diunduh ke mesin lokal Anda saat Anda mengirim model ke QuickSight. Anda mengunggah file skema ini ke QuickSight langkah berikutnya.
3. Buka QuickSight dan pilih kumpulan data dengan skema yang sama dengan kumpulan data yang Anda gunakan untuk membangun model Anda. Tambahkan bidang prediktif ke kumpulan data dan lakukan hal berikut:
    a. Tentukan model yang dikirim dari kanvas.
    b. Unggah file skema yang telah diunduh di Langkah 2.
4. Simpan dan publikasikan perubahan Anda, lalu buat prediksi untuk kumpulan data baru. QuickSight menggunakan model untuk mengisi kolom target dengan prediksi.

Untuk mengirim model dari Canvas ke QuickSight, Anda harus memenuhi prasyarat berikut:

- Anda harus memiliki Canvas dan QuickSight mengatur. QuickSightAkun Anda harus dibuat Wilayah AWS sama dengan aplikasi Canvas Anda. Jika Wilayah beranda QuickSight akun Anda berbeda dari Wilayah aplikasi Canvas Anda, Anda harus menutup dan membuat ulang QuickSight akun Anda, atau menyiapkan aplikasi Canvas di Wilayah yang sama dengan QuickSight akun

Anda. QuickSight Akun Anda juga harus berisi namespace default, yang Anda atur saat pertama kali membuat akun QuickSight . Hubungi administrator Anda untuk membantu Anda mendapatkan akses ke QuickSight. Untuk informasi selengkapnya, lihat Menyiapkan Amazon QuickSight di Panduan QuickSight Pengguna Amazon.

- Pengguna Anda harus memiliki izin AWS Identity and Access Management (IAM) yang diperlukan untuk mengirim prediksi Anda. QuickSight Administrator Anda dapat mengatur izin IAM untuk pengguna Anda. Untuk informasi selengkapnya, lihat Memberi Izin Pengguna Anda untuk Mengirim Prediksi ke Amazon. QuickSight
- QuickSight harus memiliki akses ke bucket Amazon S3 yang telah Anda tentukan untuk penyimpanan aplikasi Canvas. Untuk informasi selengkapnya, lihat Konfigurasikan penyimpanan Amazon S3 Anda.

Prakiraan Deret Waktu di Amazon SageMaker Canvas

> **Note**
>
> Model peramalan deret waktu hanya didukung untuk kumpulan data tabel.

Amazon SageMaker Canvas memberi Anda kemampuan untuk menggunakan perkiraan deret waktu pembelajaran mesin. Prakiraan deret waktu memberi Anda kemampuan untuk membuat prediksi yang dapat bervariasi seiring waktu.

Anda dapat membuat perkiraan deret waktu untuk contoh berikut:

- Perkiraan inventaris Anda dalam beberapa bulan mendatang.
- Jumlah barang yang terjual dalam empat bulan ke depan.
- Pengaruh penurunan harga pada penjualan selama musim liburan.
- Persediaan barang dalam 12 bulan ke depan.
- Jumlah pelanggan yang memasuki toko dalam beberapa jam ke depan.
- Meramalkan bagaimana penurunan 10% dalam harga suatu produk mempengaruhi penjualan selama periode waktu tertentu.

Untuk membuat perkiraan deret waktu, kumpulan data Anda harus memiliki yang berikut:

- Kolom stempel waktu dengan semua nilai yang memiliki tipe. `datetime`

- Kolom target yang memiliki nilai yang Anda gunakan untuk memperkirakan nilai masa depan.
- Kolom ID item yang berisi pengidentifikasi unik untuk setiap item dalam kumpulan data Anda, seperti nomor SKU.

`datetime`Nilai dalam kolom stempel waktu harus menggunakan salah satu format berikut:

- `YYYY-MM-DD HH:MM:SS`
- `YYYY-MM-DDTHH:MM:SSZ`
- `YYYY-MM-DD`
- `MM/DD/YY`
- `MM/DD/YY HH:MM`
- `MM/DD/YYYY`
- `YYYY/MM/DD HH:MM:SS`
- `YYYY/MM/DD`
- `DD/MM/YYYY`
- `DD/MM/YY`
- `DD-MM-YY`
- `DD-MM-YYYY`

Anda dapat membuat perkiraan untuk interval berikut:

- 1 menit
- 5 menit
- 15 menit
- 30 menit
- 1 Jam
- 1 hari
- 1 minggu
- 1 bulan
- 1 tahun

Nilai masa depan dalam kumpulan data masukan Anda

Canvas secara otomatis mendeteksi kolom dalam kumpulan data Anda yang berpotensi mengandung nilai future. Jika ada, nilai-nilai ini dapat meningkatkan akurasi prediksi. Kanvas menandai kolom khusus ini dengan `Future values` label. Canvas menyimpulkan hubungan antara data di kolom ini dan kolom target yang Anda coba prediksi, dan memanfaatkan hubungan itu untuk menghasilkan perkiraan yang lebih akurat.

Misalnya, Anda dapat memperkirakan jumlah es krim yang dijual oleh toko kelontong. Untuk membuat ramalan, Anda harus memiliki kolom stempel waktu dan kolom yang menunjukkan berapa banyak es krim yang dijual toko kelontong. Untuk perkiraan yang lebih akurat, dataset Anda juga dapat mencakup harga, suhu lingkungan, rasa es krim, atau pengenal unik untuk es krim.

Penjualan es krim mungkin meningkat ketika cuaca lebih hangat. Penurunan harga es krim dapat mengakibatkan lebih banyak unit terjual. Memiliki kolom dengan data suhu sekitar dan kolom dengan data harga dapat meningkatkan kemampuan Anda untuk memperkirakan jumlah unit es krim yang dijual toko kelontong.

Meskipun memberikan nilai future bersifat opsional, ini membantu Anda melakukan analisis bagaimana-jika secara langsung di aplikasi Canvas, menunjukkan kepada Anda bagaimana perubahan nilai future dapat mengubah prediksi Anda.

Menangani nilai yang hilang

Anda mungkin memiliki data yang hilang karena berbagai alasan. Alasan data Anda yang hilang mungkin menginformasikan bagaimana Anda ingin Canvas mengimplikasikannya. Misalnya, organisasi Anda mungkin menggunakan sistem otomatis yang hanya melacak saat penjualan terjadi. Jika Anda menggunakan kumpulan data yang berasal dari jenis sistem otomatis ini, Anda memiliki nilai yang hilang di kolom target.

> **Important**
>
> Jika Anda memiliki nilai yang hilang di kolom target, sebaiknya gunakan kumpulan data yang tidak memilikinya. SageMaker Canvas menggunakan kolom target untuk memperkirakan nilai masa depan. Nilai yang hilang di kolom target dapat sangat mengurangi keakuratan perkiraan.

Untuk nilai yang hilang dalam kumpulan data, Canvas secara otomatis memasukkan nilai yang hilang untuk Anda dengan mengisi kolom target dengan `0` dan kolom numerik lainnya dengan nilai median kolom.

Namun, Anda dapat memilih logika pengisian Anda sendiri untuk kolom target dan kolom numerik lainnya di kumpulan data Anda. Kolom target memiliki pedoman dan batasan pengisian yang berbeda dari kolom numerik lainnya. Kolom target diisi hingga akhir periode historis, sedangkan kolom numerik diisi di periode historis dan masa depan hingga akhir cakrawala perkiraan. Canvas hanya mengisi nilai future dalam kolom numerik jika data Anda memiliki setidaknya satu record dengan stempel waktu future dan nilai untuk kolom tertentu.

Anda dapat memilih salah satu opsi logika pengisian berikut untuk memasukkan nilai yang hilang dalam data Anda:

- `zero`— Isi dengan`0`.
- `NaN`— Isi dengan NaN, atau bukan nomor. Ini hanya didukung untuk kolom target.
- `mean`— Isi dengan nilai rata-rata dari seri data.
- `median`— Isi dengan nilai median dari seri data.
- `min`— Isi dengan nilai minimum dari seri data.
- `max`— Isi dengan nilai maksimum dari seri data.

Saat memilih logika pengisian, Anda harus mempertimbangkan bagaimana model Anda menafsirkan logika. Misalnya, dalam skenario ritel, mencatat penjualan nol dari item yang tersedia berbeda dengan mencatat penjualan nol dari item yang tidak tersedia, karena skenario terakhir tidak selalu menyiratkan kurangnya minat pelanggan pada item yang tidak tersedia. Dalam hal ini, mengisi kolom target dari kumpulan data dapat menyebabkan model menjadi kurang bias dalam prediksinya dan menyimpulkan kurangnya minat pelanggan pada item yang tidak tersedia. `0` Sebaliknya, mengisi dengan `NaN` dapat menyebabkan model mengabaikan kejadian sebenarnya dari nol item yang dijual dari item yang tersedia.

Jenis prakiraan

Anda dapat membuat salah satu dari jenis prakiraan berikut:

- Item tunggal
- Semua item

Untuk perkiraan semua item dalam kumpulan data Anda, SageMaker Canvas mengembalikan perkiraan untuk nilai masa depan untuk setiap item dalam kumpulan data Anda.

Untuk perkiraan item tunggal, Anda menentukan item dan SageMaker Canvas mengembalikan perkiraan untuk nilai masa depan. Perkiraan mencakup grafik garis yang memplot nilai prediksi dari waktu ke waktu.

Topik

Dapatkan wawasan tambahan dari perkiraan Anda

Di Amazon SageMaker Canvas, Anda dapat menggunakan metode opsional berikut untuk mendapatkan lebih banyak wawasan dari perkiraan Anda:

- Kolom grup
- Jadwal liburan
- Skenario bagaimana-jika

Anda dapat menentukan kolom dalam kumpulan data Anda sebagai kolom Grup. Amazon SageMaker Canvas mengelompokkan perkiraan berdasarkan setiap nilai di kolom. Misalnya, Anda dapat mengelompokkan perkiraan pada kolom yang berisi data harga atau pengidentifikasi item unik. Mengelompokkan perkiraan berdasarkan kolom memungkinkan Anda membuat perkiraan yang lebih spesifik. Misalnya, jika Anda mengelompokkan perkiraan pada kolom yang berisi pengidentifikasi item, Anda dapat melihat perkiraan untuk setiap item.

Penjualan barang secara keseluruhan mungkin dipengaruhi oleh kehadiran hari libur. Misalnya, di Amerika Serikat, jumlah barang yang terjual pada bulan November dan Desember mungkin sangat berbeda dari jumlah barang yang terjual pada bulan Januari. Jika Anda menggunakan data dari November dan Desember untuk memperkirakan penjualan pada bulan Januari, hasil Anda mungkin tidak akurat. Menggunakan jadwal liburan mencegah Anda mendapatkan hasil yang tidak akurat. Anda dapat menggunakan jadwal liburan untuk 251 negara.

Untuk perkiraan pada satu item dalam kumpulan data Anda, Anda dapat menggunakan skenario bagaimana-jika. Skenario bagaimana-jika memberi Anda kemampuan untuk mengubah nilai dalam data Anda dan mengubah perkiraan. Misalnya, Anda dapat menjawab pertanyaan-pertanyaan

berikut dengan menggunakan skenario bagaimana-jika, “Bagaimana jika saya menurunkan harga? Bagaimana itu akan mempengaruhi jumlah barang yang terjual?”

Buat perkiraan deret waktu

Untuk membuat perkiraan deret waktu, Anda memilih kolom target. Kolom target berisi data yang ingin Anda prediksi. Misalnya, kolom target Anda mungkin memiliki data tentang jumlah item yang terjual. Setelah Anda memilih kolom target, Amazon SageMaker Canvas memilih jenis Model. SageMaker Canvas menggunakan data deret waktu untuk secara otomatis memilih model deret waktu yang dapat Anda gunakan untuk membuat prediksi pada data Anda. Setelah Anda membangun model, Anda dapat mengevaluasi kinerjanya dan menggunakannya untuk membuat prediksi pada data baru.

Gunakan prosedur berikut untuk membuat perkiraan deret waktu.

Untuk membuat perkiraan deret waktu, lakukan hal berikut.

1. Impor data.
2. Pilih kolom target dalam kumpulan data Anda.
3. SageMaker Canvas secara otomatis memilih peramalan deret Waktu sebagai jenis model. Pilih Setel konfigurasi untuk mengonfirmasi bahwa Anda melakukan perkiraan deret waktu.
4. Tentukan bidang berikut:

    - Kolom ID Item - Kolom yang berisi pengidentifikasi unik untuk setiap item dalam kumpulan data Anda. Misalnya, nomor SKU secara unik mengidentifikasi item.
    - Opsional: Kolom grup - Kelompokkan perkiraan deret waktu berdasarkan nilai di kolom. Misalnya, Anda dapat mengelompokkan perkiraan Anda untuk suatu item berdasarkan toko.
    - Kolom stempel waktu - Kolom yang berisi stempel waktu dalam kumpulan data Anda. Untuk daftar `datetime` format yang didukung untuk kolom ini, lihatPrakiraan Deret Waktu di Amazon SageMaker Canvas.
    - Timestamp masa depan — Stempel waktu yang menunjukkan waktu perkiraan masa depan. SageMaker Kanvas memperkirakan nilai hingga titik waktu yang ditentukan oleh stempel waktu.
    - Opsional: Jadwal liburan — Aktifkan jadwal liburan untuk menggunakan jadwal liburan suatu negara. Gunakan untuk membuat perkiraan Anda dengan data liburan lebih akurat.

Anda dapat memiliki salah satu dari jenis nilai yang hilang berikut:

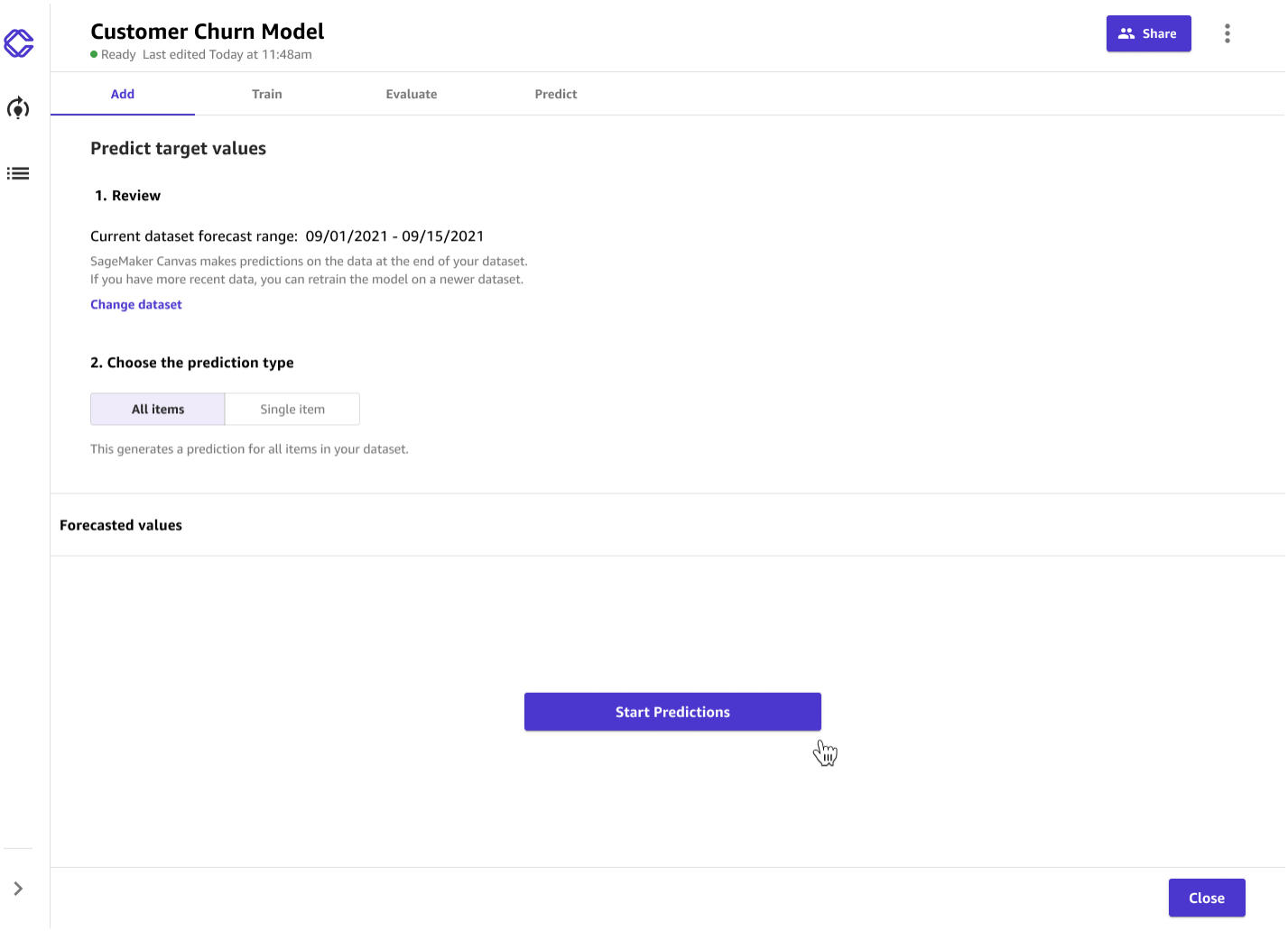
- Nilai-nilai future yang hilang
- Nilai yang hilang

Nilai future yang hilang adalah nilai yang hilang di kolom target. SageMaker Canvas menggunakan nilai di kolom target untuk memperkirakan nilai di masa depan. Jika Anda memiliki nilai yang hilang di kolom target, perkiraan Anda mungkin kurang akurat. Kami sangat menyarankan untuk memperbarui dataset.

Nilai yang hilang adalah nilai yang hilang di kolom apa pun selain kolom target. Dengan nilai yang hilang yang tidak ada di kolom target, akan sangat membantu untuk mencatat hal berikut:

- Mereka umumnya tidak mengurangi keakuratan perkiraan Anda sebanyak kehilangan nilai future.
- SageMaker Canvas secara otomatis menyiratkan nilai yang hilang.

Anda dapat mengevaluasi model dengan melihat seberapa dekat prediksi dalam nilai aktual. Anda juga dapat menggunakan metrik Dampak Kolom untuk menentukan arah dan besarnya dampak kolom pada prediksi model. Misalnya, pada gambar berikut, liburan memiliki dampak positif terbesar pada perkiraan permintaan. Harga memiliki dampak negatif terbesar pada permintaan.

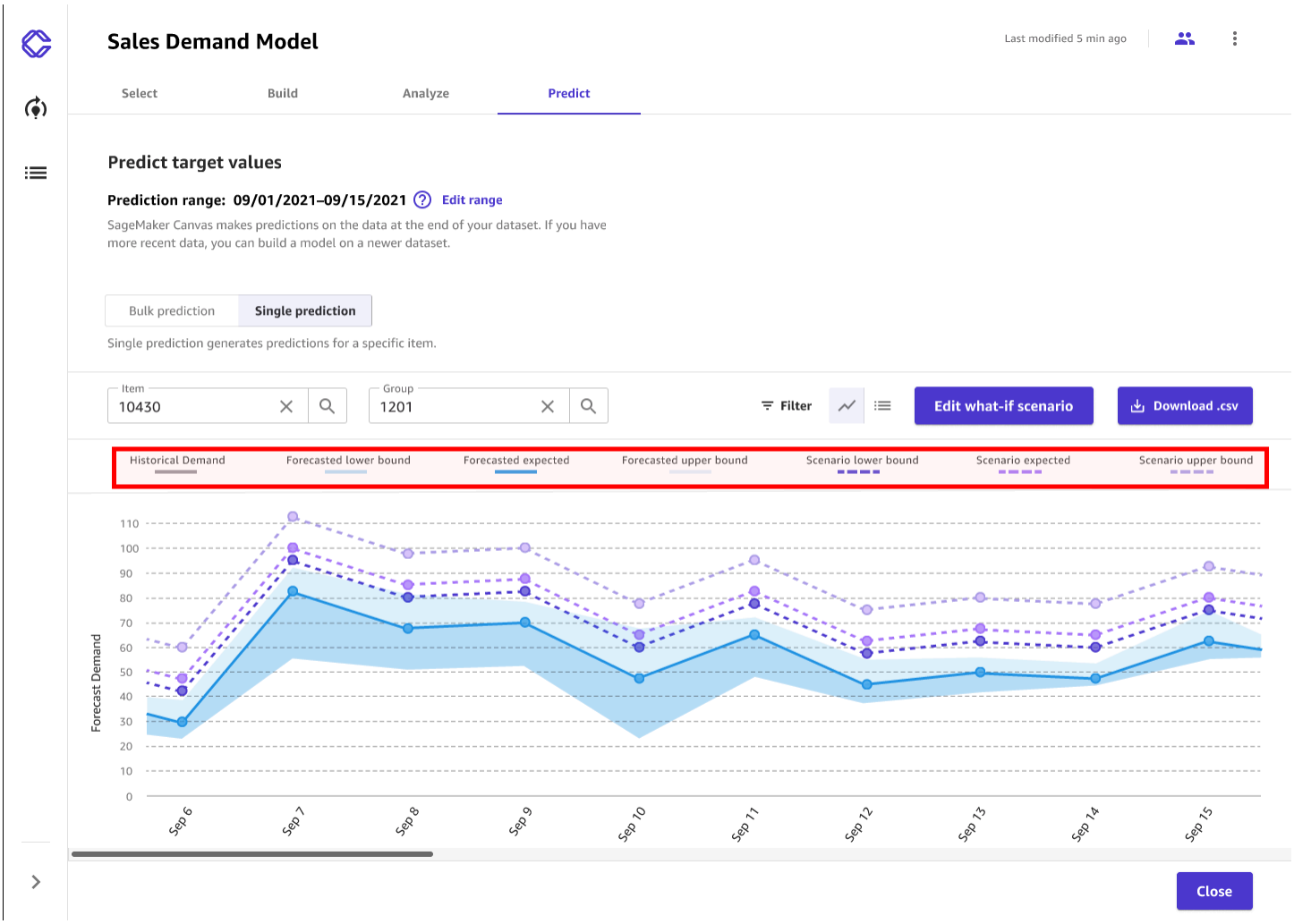


Setelah Anda membuat model, Anda dapat membuat jenis prakiraan berikut:

- Item tunggal — Buat perkiraan untuk satu item dalam kumpulan data dan grafik garis dari nilai yang diperkirakan SageMaker Canvas. Misalnya, Anda dapat melihat bagaimana penjualan suatu barang bervariasi dari waktu ke waktu.
- Semua item — Buat perkiraan untuk semua item dalam kumpulan data.
- Skenario bagaimana-jika - Lihat bagaimana perubahan nilai dalam kumpulan data dapat memengaruhi perkiraan keseluruhan untuk satu item.

Gambar berikut menunjukkan perkiraan item tunggal dengan skenario bagaimana-jika. Dalam skenario bagaimana-jika, Anda memiliki kemampuan untuk mengubah nilai yang dapat bervariasi seiring waktu. Anda dapat melihat bagaimana mengubah nilai memengaruhi perkiraan.

Titik-titik yang dihubungkan oleh garis biru solid adalah nilai-nilai yang diperkirakan model. Titik-titik yang dihubungkan oleh garis putus-putus menunjukkan skenario bagaimana-jika.



Memperbarui Model di Amazon SageMaker Canvas

Amazon SageMaker Canvas memberi Anda kemampuan untuk memperbarui model yang telah Anda buat menggunakan data baru. SageMaker Canvas menunjukkan riwayat model, sehingga Anda dapat membandingkan model yang Anda buat baru-baru ini dengan model yang telah Anda buat di masa lalu.

Setiap model yang Anda buat memiliki nomor versi. Model pertama adalah Versi 1, atauV1. Anda dapat menggunakan versi model untuk melihat perubahan akurasi prediksi saat memperbarui data atau menggunakan transformasi lanjutan.

> **Note**
>
> Model prediksi teks dan prediksi gambar hanya mendukung satu versi model.

Untuk versi baru model, Anda hanya dapat memilih kumpulan data yang memiliki kolom target yang sama dengan kolom target di Versi 1. Anda harus membuat setidaknya satu versi model untuk menambahkan versi baru, dan Anda dapat menghapus versi yang tidak berguna bagi Anda lagi.

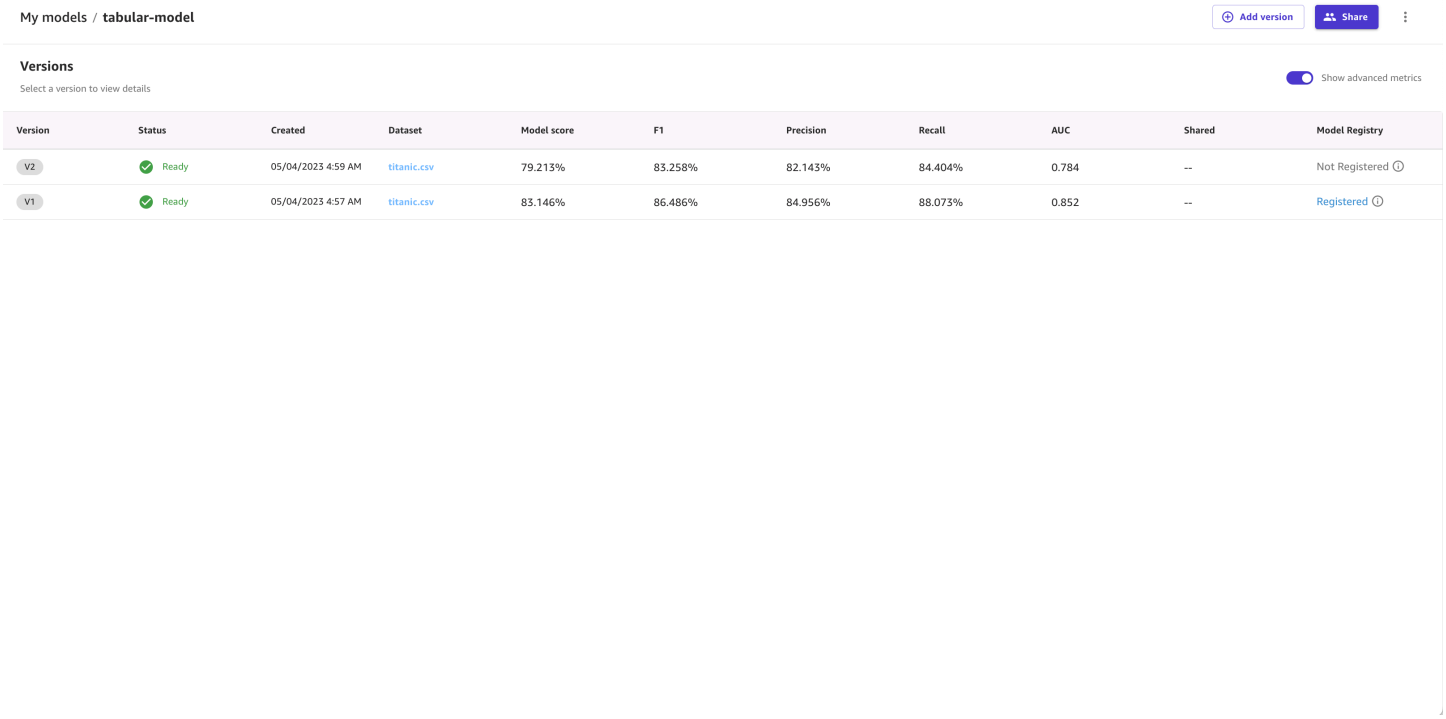
Anda juga dapat melihat Daftarkan versi model di registri SageMaker model untuk membantu melacak versi Anda dari waktu ke waktu dan berkolaborasi dengan pengguna Studio Classic yang dapat menyetujui atau menolak versi model Anda.

Gunakan prosedur berikut untuk menambahkan versi model baru atau untuk melihat semua versi untuk model Anda.

Untuk menambahkan versi model baru, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas Anda.
2. Di panel navigasi kiri, pilih Model saya.
3. Pada halaman Model saya, pilih model Anda. Anda dapat Memfilter berdasarkan jenis masalah untuk menemukan model Anda dengan lebih mudah.
4. Setelah memilih model Anda, halaman Versi terbuka, mencantumkan semua versi model Anda.
5. Pilih Tambahkan versi.

Gambar berikut menunjukkan halaman Versi untuk model, di mana Anda dapat melihat versi model Anda dan menambahkan versi baru.



Pada halaman Versi, Anda dapat melihat informasi berikut untuk setiap versi model Anda:

- Status — Bidang ini memberi tahu Anda apakah model Anda sedang build (`In building`), done building (`Ready`), gagal membangun (`Failed`), atau masih diedit (`In draft`).
- Skor model, F1, Presisi, Ingat, dan AUC — Jika Anda mengaktifkan sakelar Tampilkan metrik lanjutan di halaman ini, Anda dapat melihat metrik model ini. Metrik ini menunjukkan keakuratan dan kinerja model Anda. Untuk informasi selengkapnya, lihat Mengevaluasi model Anda.
- Dibagikan - Bidang ini memberi tahu Anda apakah Anda telah membagikan versi model dengan pengguna SageMaker Studio Classic atau tidak.
- Registri model — Bidang ini memberi tahu Anda apakah Anda telah mendaftarkan versi ke registri model atau tidak. Untuk informasi selengkapnya, lihat Daftarkan versi model di registri SageMaker model.

Setelah Anda memilih versi baru, Anda memulai proses membangun model lain. Proses untuk membangun versi baru model hampir sama dengan proses untuk membangun model untuk pertama kalinya. Untuk versi baru model, Anda hanya dapat memilih kumpulan data yang memiliki kolom target yang sama dengan kolom target di Versi 1. Untuk informasi lebih lanjut tentang membangun model, lihatMembangun model kustom.

Operasionalkan model Anda

Setelah membuat model di SageMaker Canvas yang Anda rasa percaya diri, Anda mungkin ingin mengintegrasikan model Anda dengan proses operasi pembelajaran mesin (MLOP) di organisasi Anda. MLOP mencakup tugas-tugas umum seperti menyebarkan model untuk digunakan dalam produksi atau menyiapkan pipeline integrasi berkelanjutan dan penyebaran berkelanjutan (CI/CD).

Topik berikut menjelaskan bagaimana Anda dapat menggunakan fitur dalam Canvas untuk menggunakan model buatan Canvas dalam produksi.

Topik

Daftarkan versi model di registri SageMaker model

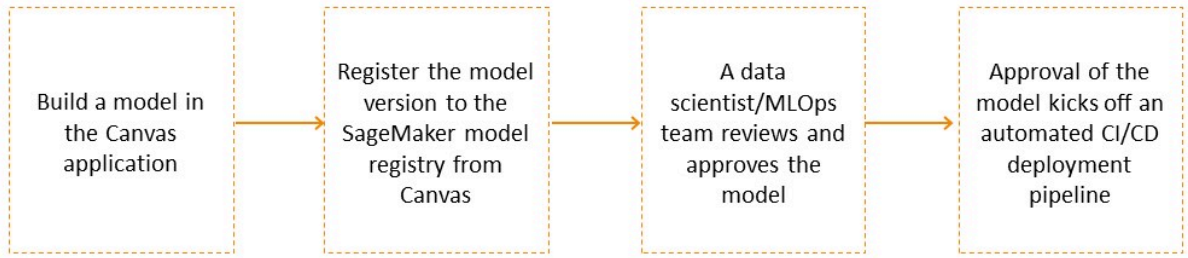
Dengan SageMaker Canvas, Anda dapat membangun beberapa iterasi, atau versi, model Anda untuk memperbaikinya dari waktu ke waktu. Anda mungkin ingin membuat versi baru model Anda jika Anda memperoleh data pelatihan yang lebih baik atau jika Anda ingin mencoba meningkatkan akurasi model. Untuk informasi selengkapnya tentang menambahkan versi ke model Anda, lihat Memperbarui model.

Setelah Anda membangun model yang Anda rasa percaya diri, Anda mungkin ingin mengevaluasi kinerjanya dan meninjaunya oleh ilmuwan data atau insinyur MLOP di organisasi Anda sebelum menggunakannya dalam produksi. Untuk melakukan ini, Anda dapat mendaftarkan versi model Anda ke registri SageMaker model. Registri SageMaker model adalah repositori yang dapat digunakan oleh ilmuwan atau insinyur data untuk membuat katalog model pembelajaran mesin (ML) dan mengelola versi model dan metadata terkait, seperti metrik pelatihan. Mereka juga dapat mengelola dan mencatat status persetujuan model.

Setelah Anda mendaftarkan versi model Anda ke registri SageMaker model, ilmuwan data atau tim MLOPS Anda dapat mengakses registri SageMaker model melalui SageMaker Studio Classic, yang merupakan lingkungan pengembangan terintegrasi berbasis web (IDE) untuk bekerja dengan model pembelajaran mesin. Di antarmuka registri SageMaker model di Studio Classic, ilmuwan data atau tim MLOPs dapat mengevaluasi model Anda dan memperbarui status persetujuannya. Jika model tidak sesuai dengan persyaratan mereka, ilmuwan data atau tim MLOPs dapat memperbarui statusnya. `Rejected` Jika model melakukan sesuai dengan kebutuhan mereka, maka ilmuwan data atau tim MLOPs dapat memperbarui status ke. `Approved` Kemudian, mereka dapat menerapkan

model Anda ke titik akhir atau mengotomatiskan penerapan model dengan pipeline CI/CD. Anda dapat menggunakan fitur registri SageMaker model untuk mengintegrasikan model Canvas bawaan dengan mulus dengan proses MLOPs di organisasi Anda.

Diagram berikut merangkum contoh mendaftarkan versi model yang dibangun di Canvas ke registri SageMaker model untuk diintegrasikan ke dalam alur kerja MLOPS.



Anda dapat mendaftarkan versi model tabel, gambar, dan teks ke registri SageMaker model.

> **Note**
>
> Saat ini, pendaftaran peramalan deret waktu atau versi model BYOM yang dibangun di Canvas ke registri SageMaker model tidak didukung.

Bagian berikut menunjukkan cara mendaftarkan versi model ke registri SageMaker model dari Canvas.

Manajemen izin

Secara default, Anda memiliki izin untuk mendaftarkan versi model ke registri SageMaker model. SageMaker memberikan izin ini untuk semua profil pengguna Canvas baru dan yang sudah ada melalui AmazonSageMakerCanvasFullAccesskebijakan, yang dilampirkan ke peran eksekusi AWS IAM untuk SageMaker Domain yang menghosting aplikasi Canvas Anda.

Jika administrator Canvas Anda menyiapkan Domain atau profil pengguna baru, saat mereka menyiapkan Domain dan mengikuti petunjuk prasyarat dalam panduan Memulai, SageMaker mengaktifkan izin pendaftaran model melalui opsi konfigurasi izin Ops, yang diaktifkan secara default.

Administrator Canvas dapat mengelola izin pendaftaran model di tingkat profil pengguna juga. Misalnya, jika administrator ingin memberikan izin pendaftaran model ke beberapa profil pengguna tetapi menghapus izin untuk orang lain, mereka dapat mengedit izin untuk pengguna tertentu. Prosedur berikut menunjukkan cara mematikan izin pendaftaran model untuk profil pengguna tertentu:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain profil pengguna.
5. Pada halaman Detail Domain, pilih profil Pengguna yang izinnya ingin Anda edit.
6. Pada halaman Detail Pengguna, pilih Edit.
7. Di panel navigasi kiri, pilih Pengaturan kanvas.
8. Di bagian konfigurasi izin Ops ML, matikan sakelar Aktifkan izin pendaftaran Registrasi Model.
9. Pilih Kirim untuk menyimpan perubahan pada pengaturan Domain Anda.

Profil pengguna seharusnya tidak lagi memiliki izin pendaftaran model.

Daftarkan versi model ke registri SageMaker model

SageMaker registri model melacak semua versi model yang Anda buat untuk memecahkan masalah tertentu dalam grup model. Saat Anda membuat model SageMaker Canvas dan mendaftarkannya ke registri SageMaker model, itu akan ditambahkan ke grup model sebagai versi model baru. Misalnya, jika Anda membangun dan mendaftarkan empat versi model Anda, maka ilmuwan data atau tim MLOPS yang bekerja di antarmuka registri SageMaker model dapat melihat grup model dan meninjau keempat versi model di satu tempat.

Saat mendaftarkan model Canvas ke registri SageMaker model, grup model secara otomatis dibuat dan dinamai sesuai model Canvas Anda. Secara opsional, Anda dapat mengganti namanya menjadi nama pilihan Anda, atau menggunakan grup model yang ada di registri SageMaker model. Untuk informasi selengkapnya tentang membuat grup model, lihat Membuat Grup Model.

> **Note**
>
> Saat ini, Anda hanya dapat mendaftarkan model bawaan Canvas ke registri SageMaker model di akun yang sama.

Untuk mendaftarkan versi model ke registri SageMaker model dari aplikasi Canvas, gunakan prosedur berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Model saya.
3. Pada halaman Model saya, pilih model Anda. Anda dapat Memfilter berdasarkan jenis masalah untuk menemukan model Anda dengan lebih mudah.
4. Setelah memilih model Anda, halaman Versi terbuka, mencantumkan semua versi model Anda. Anda dapat mengaktifkan sakelar Tampilkan metrik lanjutan untuk melihat metrik lanjutan, seperti Recall dan Precision, untuk membandingkan versi model Anda dan menentukan mana yang ingin Anda daftarkan.
5. Dari daftar versi model, untuk versi yang ingin Anda daftarkan, pilih ikon Opsi lainnya (⁞).
   Atau, Anda dapat mengklik dua kali pada versi yang perlu Anda daftarkan, dan kemudian pada halaman detail versi, pilih ikon Opsi lainnya (⁞).
6. Dalam daftar dropdown, pilih Tambahkan ke Registri Model. Kotak dialog Add to Model Registry terbuka.
7. Dalam kotak dialog Add to Model Registry, lakukan hal berikut:

   a. (Opsional) Di bagian grup model SageMaker Studio Classic, untuk bidang Nama grup model, masukkan nama grup model tempat Anda ingin mendaftarkan versi Anda. Anda dapat menentukan nama untuk grup model baru yang SageMaker dibuat untuk Anda, atau Anda dapat menentukan grup model yang ada. Jika Anda tidak menentukan bidang ini, Canvas mendaftarkan versi Anda ke grup model default dengan nama yang sama dengan model Anda.

   b. Pilih Tambahkan.

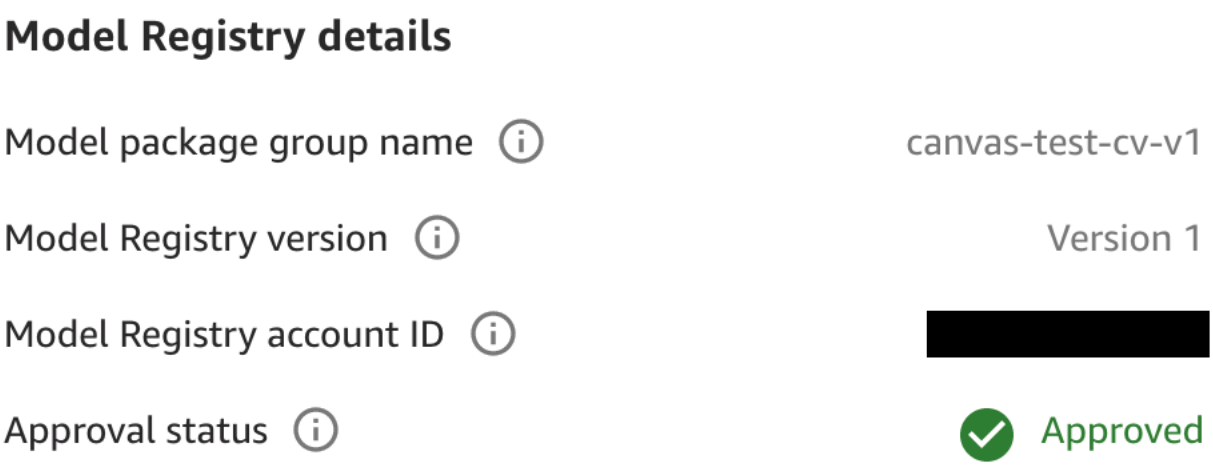
Versi model Anda sekarang harus terdaftar ke grup model di registri SageMaker model. Saat Anda mendaftarkan versi model ke grup model di registri SageMaker model, semua versi model Canvas berikutnya terdaftar ke grup model yang sama (jika Anda memilih untuk mendaftarkannya). Jika Anda ingin mendaftarkan versi Anda ke grup model yang berbeda, Anda harus pergi ke registri SageMaker model dan menghapus grup model. Kemudian, Anda dapat mendaftarkan ulang versi model Anda ke grup model baru.

Untuk melihat status model Anda, Anda dapat kembali ke halaman Versi untuk model Anda di aplikasi Canvas. Halaman ini menunjukkan status Registri Model dari setiap versi. Jika statusnya`Registered`, maka model telah berhasil didaftarkan.

Jika Anda ingin melihat detail versi model terdaftar Anda, untuk status Registri Model, Anda dapat mengarahkan kursor ke bidang Terdaftar untuk melihat kotak pop-up Detail registri Model. Detail ini berisi info lebih lanjut, seperti berikut ini:

- Nama grup paket Model adalah grup model tempat versi Anda terdaftar di registri SageMaker model.
- Status Persetujuan, yang dapat berupa `Pending ApprovalApproved`, atau`Rejected`. Jika pengguna Studio Classic menyetujui atau menolak versi Anda di registri SageMaker model, status ini diperbarui pada halaman versi model saat Anda me-refresh halaman.

Tangkapan layar berikut menunjukkan kotak Detail registri Model, bersama dengan status Persetujuan `Approved` untuk versi model khusus ini.



Menerapkan model Anda ke titik akhir

Di Amazon SageMaker Canvas, Anda dapat menerapkan model Anda ke titik akhir untuk membuat prediksi. SageMaker menyediakan infrastruktur ML bagi Anda untuk meng-host model Anda pada titik akhir dengan instance komputasi yang Anda pilih. Kemudian, Anda dapat memanggil titik akhir (mengirim permintaan prediksi) dan mendapatkan prediksi waktu nyata dari model Anda. Dengan fungsi ini, Anda dapat menggunakan model Anda dalam produksi untuk menanggapi permintaan yang masuk, dan Anda dapat mengintegrasikan model Anda dengan aplikasi dan alur kerja yang ada.

Untuk memulai, Anda harus memiliki versi model yang siap digunakan. Untuk informasi selengkapnya tentang membuat model di Canvas, lihatMembangun model kustom.

> **Important**
>
> Anda dapat menerapkan jenis model apa pun di SageMaker Canvas kecuali untuk model peramalan deret waktu.

Tinjau bagian Manajemen izin berikut, lalu mulailah membuat penerapan baru di bagian Menerapkan model.

Manajemen izin

Secara default, Anda memiliki izin untuk menerapkan versi model ke titik akhir SageMaker Hosting. SageMaker memberikan izin ini untuk semua profil pengguna Canvas baru dan yang sudah ada melalui AmazonSageMakerCanvasFullAccesskebijakan, yang dilampirkan ke peran eksekusi AWS IAM untuk SageMaker Domain yang menghosting aplikasi Canvas Anda.

Jika administrator Canvas Anda menyiapkan Domain atau profil pengguna baru, saat mereka menyiapkan Domain dan mengikuti petunjuk prasyarat diPrasyarat untuk menyiapkan Amazon Canvas SageMaker , SageMaker mengaktifkan izin penerapan model melalui opsi Aktifkan penerapan langsung model Canvas, yang diaktifkan secara default.

Administrator Canvas dapat mengelola izin penerapan model di tingkat profil pengguna juga. Misalnya, jika administrator ingin memberikan izin penerapan model ke beberapa profil pengguna tetapi menghapus izin untuk orang lain, mereka dapat mengedit izin untuk pengguna tertentu.

Prosedur berikut menunjukkan cara menonaktifkan izin penerapan model untuk profil pengguna tertentu:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain profil pengguna.
5. Pada halaman Detail Domain, pilih profil Pengguna yang izinnya ingin Anda edit.
6. Pada halaman Detail Pengguna, pilih Edit.
7. Di panel navigasi kiri, pilih Pengaturan kanvas.

8. Di bagian konfigurasi izin Ops ML, matikan sakelar Aktifkan penerapan langsung model Canvas.
9. Pilih Kirim untuk menyimpan perubahan pada pengaturan Domain Anda.

Profil pengguna seharusnya tidak lagi memiliki izin penerapan model.

Menyebarkan model

Untuk memulai penerapan model Anda, Anda membuat penerapan baru di Canvas dan menentukan versi model yang ingin Anda terapkan bersama dengan infrastruktur ML, seperti jenis dan jumlah instance komputasi yang ingin Anda gunakan untuk hosting model.

Canvas menyarankan jenis default dan jumlah instans berdasarkan jenis model Anda, atau Anda dapat mempelajari lebih lanjut tentang berbagai jenis SageMaker instans di halaman SageMaker harga Amazon. Anda dikenakan biaya berdasarkan harga SageMaker instans saat titik akhir Anda aktif.

Setelah model Anda diterapkan ke titik akhir inferensi real-time SageMaker Hosting, Anda dapat mulai membuat prediksi dengan menjalankan titik akhir.

Ada beberapa cara berbeda bagi Anda untuk menerapkan versi model dari aplikasi Canvas. Anda dapat mengakses opsi penerapan model melalui salah satu metode berikut:

- Pada halaman Model saya dari aplikasi Canvas, Anda dapat memilih model yang ingin Anda terapkan. Kemudian, dari halaman Versi model, Anda dapat memilih ikon Opsi lainnya (⋮) di sebelah versi model dan pilih Deploy.
- Saat berada di halaman detail untuk versi model, pada tab Analisis, Anda dapat memilih opsi Deploy.
- Saat berada di halaman detail untuk versi model, pada tab Predict, Anda dapat memilih ikon Opsi lainnya (⋮) di bagian atas halaman dan pilih Terapkan.
- Pada halaman Operasi aplikasi Canvas, Anda dapat memilih tab Deployments dan kemudian memilih Create deployment.

Semua metode ini membuka panel samping model Deploy, tempat Anda menentukan konfigurasi penerapan untuk model Anda. Untuk menerapkan model dari panel ini, lakukan hal berikut:

1. (Opsional) Jika Anda membuat penerapan dari halaman Operasi, Anda akan memiliki opsi untuk Pilih model dan versi. Gunakan menu tarik-turun untuk memilih model dan versi model yang ingin Anda terapkan.
2. Masukkan nama di bidang Nama Deployment.
3. Untuk tipe Instance, SageMaker mendeteksi tipe dan nomor instans default yang sesuai untuk model Anda. Namun, Anda dapat mengubah jenis instance yang ingin Anda gunakan untuk menghosting model Anda.

   > **Note**
   >
   > Jika Anda kehabisan kuota instans untuk jenis instans yang dipilih di AWS akun Anda, Anda dapat meminta peningkatan kuota. Untuk informasi selengkapnya tentang kuota default dan cara meminta peningkatan, lihat SageMaker titik akhir dan kuota Amazon di panduan Referensi AWS Umum.

4. Untuk hitungan Instance, Anda dapat mengatur jumlah instans aktif yang digunakan untuk titik akhir Anda. SageMaker mendeteksi nomor default yang cocok untuk model Anda, tetapi Anda dapat mengubah nomor ini.
5. Saat Anda siap untuk menerapkan model Anda, pilih Deploy.

Model Anda sekarang harus diterapkan ke titik akhir. Untuk informasi tentang cara melihat detail penerapan atau melakukan berbagai tindakan, lihat bagian berikut.

Melihat penerapan Anda

Anda mungkin ingin memeriksa status atau detail penerapan model di Canvas. Misalnya, jika penerapan gagal, Anda mungkin ingin memeriksa detailnya untuk memecahkan masalah.

Anda dapat melihat penerapan model Canvas Anda dari aplikasi Canvas atau dari konsol Amazon SageMaker .

Untuk melihat detail penerapan dari Canvas, pilih salah satu prosedur berikut:

Untuk melihat detail penerapan Anda dari halaman Operasi, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Operasi.
3. Pilih tab Deployment.

4. Pilih penyebaran Anda berdasarkan nama dari daftar.

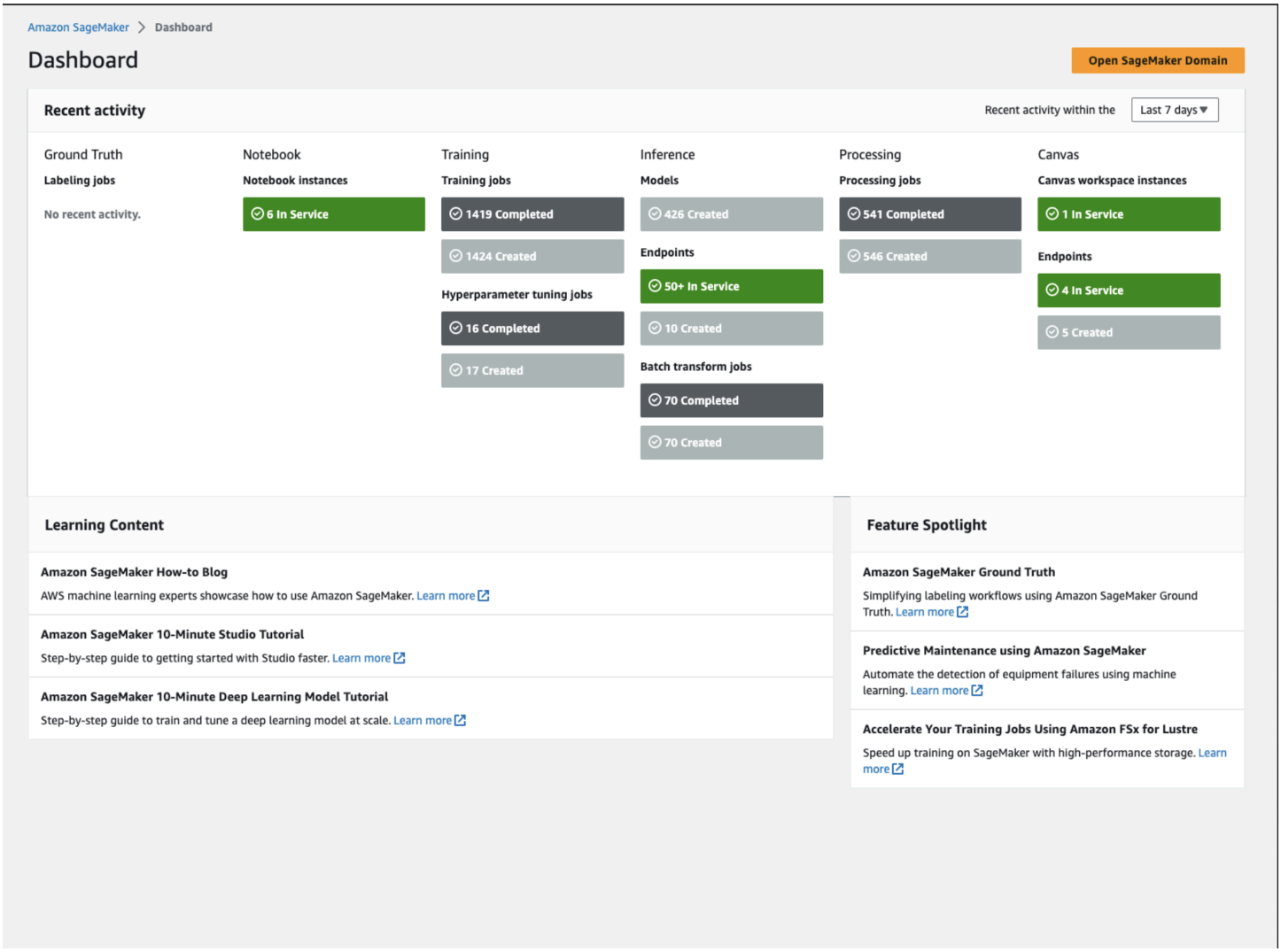
Untuk melihat detail penerapan Anda dari halaman versi model, lakukan hal berikut:

1. Di aplikasi SageMaker Canvas, buka halaman detail versi model Anda.
2. Pilih tab Deploy.
3. Di bagian Deployment yang mencantumkan semua konfigurasi penerapan yang terkait dengan versi model tersebut, temukan penerapan Anda.
4. Pilih ikon Opsi lainnya
   (⋮),
   lalu pilih Lihat detail untuk membuka halaman detail.

Halaman detail untuk penerapan Anda terbuka, dan Anda dapat melihat informasi seperti waktu prediksi terbaru, status dan konfigurasi titik akhir, dan versi model yang saat ini digunakan ke titik akhir.

Anda juga dapat melihat instance ruang kerja Canvas yang saat ini aktif dan titik akhir aktif dari SageMaker dasbor di konsol. SageMaker Endpoint Canvas Anda terdaftar di samping titik akhir SageMaker Hosting lain yang telah Anda buat, dan Anda dapat memfilternya dengan mencari titik akhir dengan tag Canvas.

Tangkapan layar berikut menunjukkan SageMaker dasbor. Di bagian Canvas, Anda dapat melihat bahwa satu instance ruang kerja dalam layanan dan empat titik akhir aktif.



Perbarui konfigurasi penerapan

Anda juga dapat memperbarui konfigurasi penerapan Anda. Misalnya, Anda dapat menerapkan versi model yang diperbarui ke titik akhir, atau Anda dapat memperbarui jenis instans atau jumlah instance di belakang titik akhir berdasarkan kebutuhan kapasitas Anda.

Ada beberapa cara berbeda bagi Anda untuk memperbarui penerapan Anda dari aplikasi Canvas. Anda dapat menggunakan salah satu metode berikut:

- Pada halaman Operasi aplikasi Canvas, Anda dapat memilih tab Deployments dan memilih penyebaran yang ingin Anda perbarui. Kemudian, pilih Perbarui konfigurasi.
- Saat berada di halaman detail untuk versi model, pada tab Deploy, Anda dapat melihat penerapan untuk versi tersebut. Di samping penerapan, pilih ikon Opsi lainnya

(⋮)
dan kemudian pilih Perbarui konfigurasi.

Kedua metode sebelumnya membuka panel sisi konfigurasi Perbarui, tempat Anda dapat membuat perubahan pada konfigurasi penerapan Anda. Untuk memperbarui konfigurasi, lakukan hal berikut:

1. Untuk menu tarik-turun Pilih versi, Anda dapat memilih versi model yang berbeda untuk diterapkan ke titik akhir.

   > **Note**
   >
   > Saat memperbarui konfigurasi penerapan, Anda hanya dapat memilih versi model yang berbeda untuk diterapkan. Untuk menerapkan model yang berbeda, buat penerapan baru.

2. Untuk tipe Instance, Anda dapat memilih jenis instans yang berbeda untuk menghosting model Anda.
3. Untuk jumlah Instance, Anda dapat mengubah jumlah instans aktif yang digunakan untuk titik akhir Anda.
4. Pilih Simpan.

Konfigurasi penerapan Anda sekarang harus diperbarui.

Uji penerapan Anda

Anda dapat menguji penerapan Anda dengan memanggil titik akhir, atau membuat permintaan prediksi tunggal, melalui aplikasi Canvas. Titik akhir mengembalikan respons dengan prediksi bersama dengan probabilitas prediksi itu benar. Anda dapat menggunakan fungsi ini untuk mengonfirmasi bahwa titik akhir Anda merespons permintaan sebelum menjalankan titik akhir Anda secara terprogram di lingkungan produksi.

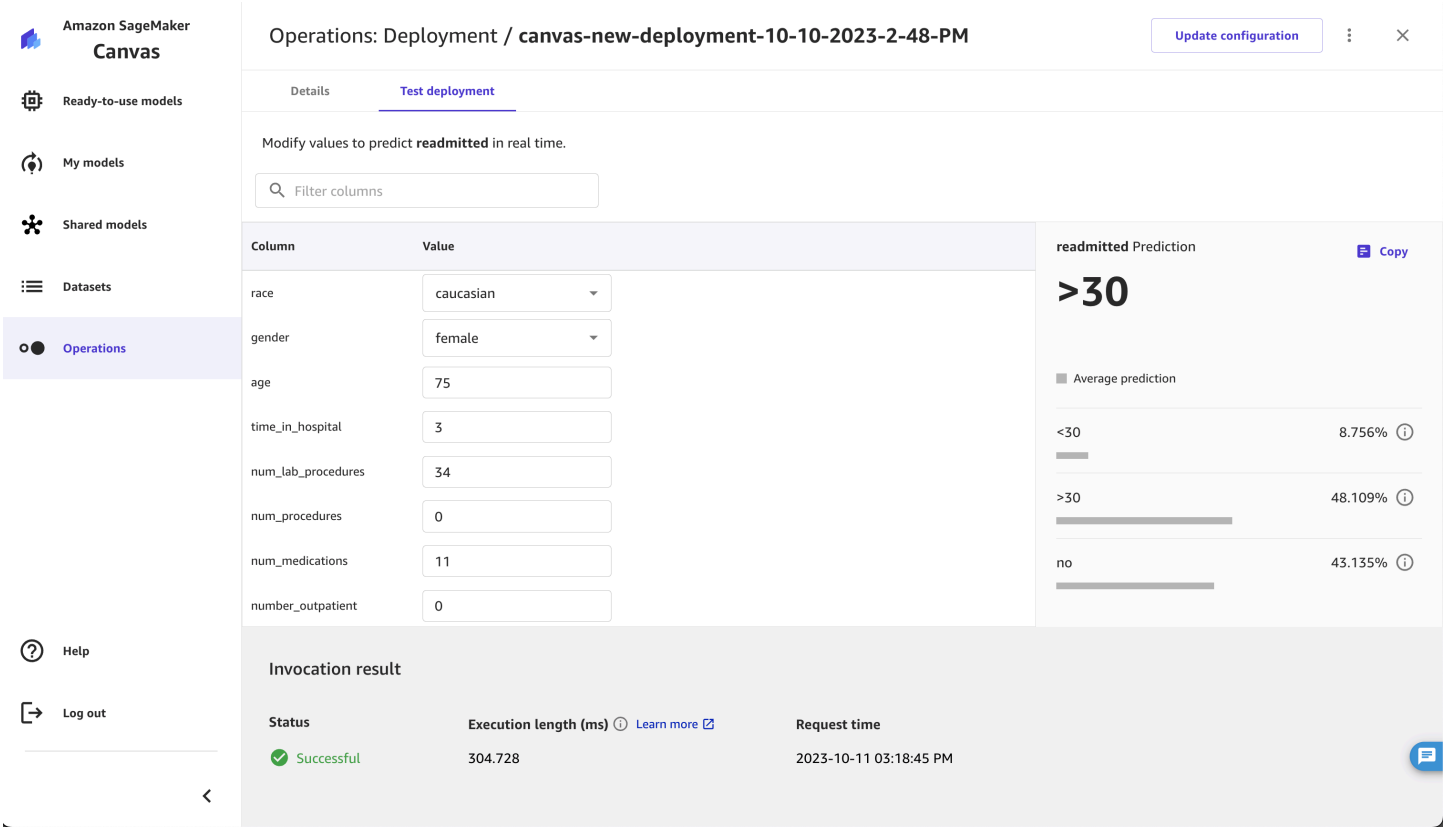
> **Note**
>
> Panjang eksekusi adalah perkiraan waktu yang dibutuhkan untuk memanggil dan mendapatkan respons dari titik akhir di Canvas. Untuk metrik latensi mendetail, lihat Metrik Pemanggilan SageMaker Titik Akhir.

Untuk menguji titik akhir Anda melalui aplikasi Canvas, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Operasi.
3. Pilih tab Deployment.
4. Dari daftar penerapan, pilih salah satu dengan titik akhir yang ingin Anda panggil.
5. Pada halaman detail penerapan, pilih tab Test deployment.
6. Pada halaman pengujian penerapan, Anda dapat memodifikasi bidang Nilai untuk menentukan titik data baru.
7. Setelah memodifikasi nilai, pilih Perbarui untuk mendapatkan hasil prediksi.

Prediksi dimuat, bersama dengan bidang hasil Pemanggilan yang menunjukkan apakah pemanggilan berhasil atau tidak dan berapa lama permintaan untuk diproses.

Screenshot berikut menunjukkan prediksi yang dilakukan dalam aplikasi Canvas pada tab Test deployment.



Untuk semua jenis model kecuali prediksi numerik, prediksi mengembalikan bidang berikut:

- predicted_label — output yang diprediksi
- probabilitas — probabilitas bahwa label yang diprediksi benar
- label — daftar semua label yang mungkin
- probabilitas — probabilitas yang sesuai dengan setiap label (urutan daftar ini cocok dengan urutan label)

Untuk model prediksi numerik, prediksi hanya berisi bidang skor, yang merupakan output prediksi dari model, seperti harga prediksi sebuah rumah.

Anda dapat terus membuat prediksi tunggal melalui halaman pengujian penerapan, atau Anda dapat melihat bagian berikut Panggil titik akhir Anda untuk mempelajari cara memanggil titik akhir Anda secara terprogram dari aplikasi.

Panggil titik akhir Anda

Setelah menguji penerapan, Anda dapat menggunakan titik akhir dalam produksi dengan aplikasi Anda dengan menjalankan titik akhir secara terprogram dengan cara yang sama seperti Anda dapat memanggil titik akhir real-time lainnya. SageMaker Memanggil endpoint secara terprogram mengembalikan objek respons yang berisi bidang yang sama seperti yang disebutkan di bagian sebelumnya. Uji penerapan Anda

Untuk informasi lebih rinci tentang cara memanggil titik akhir secara terprogram, lihat. Memanggil model untuk inferensi waktu nyata

Contoh Python berikut menunjukkan cara memanggil endpoint Anda berdasarkan jenis model.

Model prediksi numerik dan kategoris

Contoh berikut menunjukkan cara memanggil model prediksi numerik atau kategoris.

```
import boto3
import pandas as pd

client = boto3.client("runtime.sagemaker")
body = pd.DataFrame(['feature_column1', 'feature_column2'], ['feature_column1',
 'feature_column2']).to_csv(header=False, index=False).encode("utf-8")

response = client.invoke_endpoint(
    EndpointName="endpoint_name",
```

```
    ContentType="text/csv",
    Body=body,
    Accept="application/json"
)
```

Model prediksi gambar

Contoh berikut menunjukkan cara memanggil model prediksi gambar.

```
import boto3
client = boto3.client("runtime.sagemaker")
with open("example_image.jpg", "rb") as file:
    body = file.read()
    response = client.invoke_endpoint(
        EndpointName="endpoint_name",
        ContentType="application/x-image",
        Body=body,
        Accept="application/json"
    )
```

Model prediksi teks

Contoh berikut menunjukkan cara memanggil model prediksi teks.

```
import boto3
import pandas as pd

client = boto3.client("runtime.sagemaker")
body = pd.DataFrame([["Example text 1"], ["Example text 2"]]).to_csv(header=False,
 index=False).encode("utf-8")

response = client.invoke_endpoint(
    EndpointName="endpoint_name",
    ContentType="text/csv",
    Body=body,
    Accept="application/json"
)
```

Hapus penerapan model

Anda dapat menghapus penerapan model Anda dari aplikasi Canvas. Tindakan ini juga menghapus titik akhir dari SageMaker konsol dan mematikan sumber daya terkait titik akhir apa pun.

> **Note**
>
> Secara opsional, Anda dapat menghapus titik akhir Anda melalui SageMaker konsol atau menggunakan API. SageMaker `DeleteEndpoint` Untuk informasi selengkapnya, lihat Hapus Titik Akhir dan Sumber Daya. Namun, saat Anda menghapus titik akhir melalui SageMaker konsol atau API alih-alih aplikasi Canvas, daftar penerapan di Canvas tidak diperbarui secara otomatis. Anda juga harus menghapus penyebaran dari aplikasi Canvas untuk menghapusnya dari daftar.

Untuk menghapus penerapan di Canvas, lakukan hal berikut:

1. Buka aplikasi SageMaker Canvas.
2. Di panel navigasi kiri, pilih Operasi.
3. Pilih tab Deployment.
4. Dari daftar penerapan, pilih salah satu yang ingin Anda hapus.
5. Di bagian atas halaman detail penyebaran, pilih ikon Opsi lainnya ( ).
6. Pilih Hapus penerapan.
7. Di kotak dialog Hapus penyebaran, pilih Hapus.

Deployment dan titik akhir SageMaker Hosting Anda sekarang harus dihapus dari Canvas dan konsol. SageMaker

Kelola otomatisasi

Di SageMaker Canvas, Anda dapat membuat otomatisasi yang memperbarui kumpulan data atau menghasilkan prediksi dari model sesuai jadwal. Misalnya, Anda mungkin menerima data pengiriman baru setiap hari. Anda dapat mengatur pembaruan otomatis untuk kumpulan data dan prediksi batch otomatis yang berjalan setiap kali kumpulan data diperbarui. Dengan menggunakan fitur-fitur ini, Anda dapat mengatur alur kerja otomatis dan mengurangi jumlah waktu yang Anda habiskan untuk memperbarui kumpulan data secara manual dan membuat prediksi.

> **Note**
>
> Anda hanya dapat mengatur maksimum 20 konfigurasi otomatis di aplikasi Canvas Anda. Otomatisasi hanya aktif saat Anda masuk ke aplikasi Canvas. Jika Anda keluar dari Canvas, pekerjaan otomatis Anda berhenti sampai Anda masuk kembali.

Bagian berikut menjelaskan cara melihat, mengedit, dan menghapus konfigurasi untuk otomatisasi yang ada. Untuk mempelajari cara mengatur otomatisasi, lihat topik berikut:

- Untuk mengatur pembaruan kumpulan data otomatis, lihatMemperbarui kumpulan data.
- Untuk mengatur prediksi batch otomatis, lihatBuat prediksi batch.

Lihat otomatisasi Anda

Anda juga dapat melihat semua pekerjaan pembaruan otomatis Anda dengan membuka panel navigasi kiri Canvas dan memilih Otomasi. Halaman Otomasi menggabungkan otomatisasi untuk pembaruan dataset otomatis dan prediksi batch otomatis. Dari halaman Otomasi, Anda dapat melihat tab berikut:

- Semua pekerjaan — Anda dapat melihat setiap instance pembaruan Dataset atau pekerjaan prediksi Batch yang telah dilakukan Canvas. Untuk setiap pekerjaan, Anda dapat melihat bidang seperti kumpulan data Input terkait, nama Konfigurasi konfigurasi pembaruan otomatis terkait, dan Status yang menunjukkan apakah pekerjaan berhasil atau tidak. Anda dapat memfilter pekerjaan berdasarkan nama konfigurasi:
  - Untuk pekerjaan pembaruan kumpulan data, Anda dapat memilih versi terbaru dari kumpulan data, atau pekerjaan terbaru, untuk melihat pratinjau kumpulan data.
  - Untuk pekerjaan prediksi batch, Anda dapat memilih ikon Opsi lainnya (⋮) untuk melihat atau mengunduh prediksi untuk pekerjaan itu.
- Konfigurasi - Anda dapat melihat semua pembaruan Dataset dan konfigurasi prediksi Batch yang telah Anda buat. Untuk setiap konfigurasi, Anda dapat melihat bidang seperti dataset Input terkait dan Frekuensi pekerjaan. Anda juga dapat mematikan atau mengaktifkan sakelar Pembaruan otomatis untuk menjeda atau melanjutkan pembaruan otomatis. Jika Anda memilih ikon Opsi lainnya (⋮)

untuk konfigurasi tertentu, Anda dapat memilih untuk Melihat semua pekerjaan untuk konfigurasi, Perbarui konfigurasi, atau Hapus konfigurasi.

Edit konfigurasi otomatis Anda

Setelah menyiapkan konfigurasi, Anda mungkin ingin mengubahnya. Untuk pembaruan kumpulan data otomatis, Anda dapat memperbarui lokasi Amazon S3 untuk Canvas untuk mengimpor data, frekuensi pembaruan, dan waktu mulai. Untuk prediksi batch otomatis, Anda dapat mengubah kumpulan data yang dilacak konfigurasi untuk pembaruan. Anda juga dapat mematikan otomatisasi untuk menghentikan sementara pembaruan hingga Anda memilih untuk melanjutkannya.

Bagian berikut menunjukkan cara memperbarui setiap jenis konfigurasi.

> **Note**
>
> Anda tidak dapat mengubah frekuensi untuk prediksi batch otomatis karena prediksi batch otomatis berjalan setiap kali kumpulan data target diperbarui.

Edit konfigurasi pembaruan kumpulan data otomatis Anda

Anda mungkin ingin membuat perubahan pada konfigurasi pembaruan otomatis untuk kumpulan data, seperti mengubah frekuensi pembaruan. Anda mungkin juga ingin menonaktifkan konfigurasi pembaruan otomatis untuk menjeda pembaruan pada kumpulan data Anda.

Untuk membuat perubahan pada konfigurasi pembaruan otomatis untuk kumpulan data, lakukan hal berikut:

1. Di panel navigasi kiri Canvas, pilih Automations.
2. Pilih tab Konfigurasi.
3. Untuk konfigurasi pembaruan otomatis Anda, pilih ikon Opsi lainnya (⁞).
4. Di menu tarik-turun, pilih Perbarui konfigurasi. Anda akan dibawa ke tab Pembaruan otomatis dari kumpulan data.
5. Buat perubahan Anda pada konfigurasi. Setelah selesai membuat perubahan, pilih Simpan.

Untuk menjeda pembaruan kumpulan data Anda, matikan konfigurasi otomatis Anda. Salah satu cara untuk mematikan auto update adalah dengan melakukan hal berikut:

1. Di panel navigasi kiri Canvas, pilih Automations.
2. Pilih tab Konfigurasi.
3. Temukan konfigurasi Anda dari daftar dan matikan sakelar Pembaruan otomatis.

Pembaruan otomatis untuk kumpulan data Anda sekarang dijeda. Anda dapat mengaktifkan kembali sakelar ini kapan saja untuk melanjutkan jadwal pembaruan.

Edit konfigurasi prediksi batch otomatis Anda

Saat Anda mengedit konfigurasi prediksi batch, Anda dapat mengubah kumpulan data target tetapi bukan frekuensinya (karena prediksi batch otomatis terjadi setiap kali kumpulan data diperbarui).

Untuk membuat perubahan pada konfigurasi prediksi batch otomatis Anda, lakukan hal berikut:

1. Di panel navigasi kiri Canvas, pilih Automations.
2. Pilih tab Konfigurasi.
3. Untuk konfigurasi pembaruan otomatis Anda, pilih ikon Opsi lainnya (⁞).
4. Di menu tarik-turun, pilih Perbarui konfigurasi. Anda akan dibawa ke tab Pembaruan otomatis dari kumpulan data.
5. Kotak dialog Automate batch prediction terbuka. Anda dapat memilih kumpulan data lain dan memilih Mengatur untuk menyimpan perubahan Anda.

Konfigurasi prediksi batch otomatis Anda sekarang diperbarui.

Untuk menjeda prediksi batch otomatis Anda, matikan konfigurasi otomatis Anda. Gunakan prosedur berikut untuk mematikan konfigurasi Anda:

1. Di panel navigasi kiri Canvas, pilih Automations.
2. Pilih tab Konfigurasi.
3. Temukan konfigurasi Anda dari daftar dan matikan sakelar Pembaruan otomatis.

Prediksi batch otomatis untuk kumpulan data Anda sekarang dijeda. Anda dapat mengaktifkan kembali sakelar ini kapan saja untuk melanjutkan jadwal pembaruan.

Hapus konfigurasi otomatis

Anda mungkin ingin menghapus konfigurasi untuk menghentikan alur kerja otomatis Anda di SageMaker Canvas.

Untuk menghapus konfigurasi untuk pembaruan dataset otomatis atau prediksi batch otomatis, lakukan hal berikut:

1. Di panel navigasi kiri Canvas, pilih Automations.
2. Pilih tab Konfigurasi.
3. Temukan konfigurasi pembaruan otomatis Anda, dan pilih ikon Opsi lainnya (⋮).
4. Pilih Hapus konfigurasi.
5. Di kotak dialog yang muncul, pilih Hapus.

Konfigurasi pembaruan otomatis Anda sekarang dihapus.

Berkolaborasi dengan ilmuwan data

> **Note**
>
> Fungsionalitas yang dijelaskan di halaman ini hanya berlaku untuk Amazon SageMaker Studio Classic. Saat ini, Anda hanya dapat berbagi model ke Canvas (atau melihat model Canvas bersama) di Studio Classic. Jika saat ini Anda menggunakan Studio versi terbaru, Anda harus menjalankan Studio Classic dari dalam versi terbaru Studio untuk berbagi model ke Canvas atau melihat model yang dibagikan dari Canvas. Untuk informasi selengkapnya tentang mengakses Studio Classic, lihat dokumentasi Studio Classic.

Dengan Amazon SageMaker Canvas, analis bisnis yang menggunakan Canvas dan ilmuwan data yang menggunakan Amazon SageMaker Studio Classic dapat berbagi model ML dan berkolaborasi satu sama lain sambil bekerja di lingkungan mereka sendiri untuk berbagi pengetahuan domain dan memberikan masukan ahli untuk meningkatkan model.

Menggunakan kolaborasi SageMaker Canvas, Anda dapat berbagi model build Standar dari Canvas dengan ilmuwan data di Studio Classic untuk meninjau, memperbarui, dan berbagi kembali dengan pengguna Canvas. Pengguna di Canvas dapat berbagi satu versi model dengan hingga 23 pengguna Studio Classic.

> **Note**
>
> Kolaborasi pada model dengan pengguna Studio Classic tidak didukung untuk prediksi gambar label tunggal, prediksi teks multi-kategori, atau jenis model peramalan deret waktu. Selain itu, SageMaker Canvas tidak mendukung berbagi model Anda ke profil pengguna yang sama dengan yang membuat model. Anda harus memiliki dua profil pengguna terpisah untuk berbagi model.

Bagian berikut menjelaskan langkah-langkah untuk kolaborasi:

- Dalam aplikasi Canvas, seorang analis bisnis berbagi model mereka dengan pengguna Studio Classic.
- Pengguna Studio Classic menerima model bersama dalam aplikasi Studio Classic. Mereka dapat memilih untuk berbagi umpan balik dengan analis, membuat pembaruan pada model, atau berbagi versi model alternatif.
- Analis bisnis menerima umpan balik atau model yang diperbarui di Canvas dan dapat menghasilkan prediksi dalam mode hanya lihat.

Untuk berkolaborasi, pengguna Canvas dan pengguna Studio Classic harus berada di SageMaker Domain Amazon yang sama. Untuk informasi selengkapnya tentang pengaturan Domain dan pengguna, lihat Prasyarat SageMaker Kanvas.

> **Note**
>
> Kolaborasi model berbeda dariBawa model Anda sendiri ke SageMaker Canvas, di mana Anda dapat membawa model yang telah Anda latih di mana saja dan mengimpornya ke Canvas untuk menghasilkan prediksi.

Prasyarat

Sebelum pengguna Canvas dan pengguna Studio Classic dapat berkolaborasi pada model, peran IAM pengguna harus memiliki izin AWS Identity and Access Management (IAM) untuk berbagi model. Jika Anda belum menyiapkan izin, lihatBerikan Izin Pengguna untuk Berkolaborasi dengan Studio Classic.

Pengguna Canvas juga harus memiliki model build Standar yang dilatih di Canvas dan siap untuk dibagikan.

> **Note**
>
> Kolaborasi tidak mendukung model Quick build.

Anda juga harus memiliki nama profil pengguna pengguna Studio Classic dengan siapa Anda ingin berkolaborasi. Pengguna Studio Classic harus berada di SageMaker Domain Amazon yang sama dengan pengguna Canvas Anda. Anda dapat menemukan nama profil pengguna dengan menggunakan prosedur berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilih Domain.
3. Dari daftar Domain, pilih Domain Anda. Ini membuka halaman detail Domain, di mana Anda dapat menemukan semua profil Pengguna untuk Domain.

Siapkan nama profil pengguna untuk langkah pertama dari tutorial berikut.

Pengguna kanvas: Bagikan model dengan pengguna Studio Classic

Dalam aplikasi Canvas, bagikan versi model Anda dengan pengguna Studio Classic atau minta umpan balik dari mereka. Anda harus menggunakan versi model yang telah dibuat; Anda tidak dapat membagikan versi model yang merupakan draf atau sedang dibangun. Anda hanya dapat berbagi satu versi per model.

Untuk membagikan model Canvas Anda dengan pengguna Studio Classic, gunakan prosedur berikut.

1. Buka aplikasi SageMaker Canvas.
2. Dari halaman Model, pilih model yang ingin Anda bagikan. Anda hanya dapat berbagi model build Standar.
3. Di header, pilih Bagikan.
4. Dalam kotak dialog Share Model, lakukan hal berikut:
    a. Dari daftar tarik-turun Pilih versi model untuk dibagikan, pilih versi model yang Anda inginkan umpan baliknya.

b. Dari daftar dropdown pengguna SageMaker Studio, pilih pengguna Studio Classic berdasarkan nama profilnya. Anda dapat menambahkan hingga 23 pengguna Studio Classic.

c. Untuk bidang Tambahkan catatan, Anda dapat memasukkan catatan singkat yang menyertai model Anda saat mengirimkannya ke pengguna Studio Classic.

d. Pilih Bagikan.

e. Di kotak konfirmasi Bagikan Model yang muncul, pilih Bagikan.

Anda sekarang telah membagikan model Anda dengan pengguna Studio Classic, dan pengguna menerima pemberitahuan di Studio Classic bahwa model telah dibagikan dengan mereka.

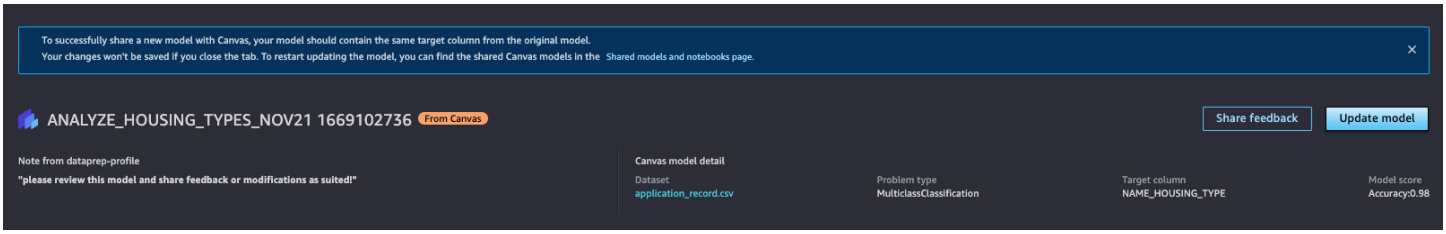
Pengguna Studio Classic: Menerima model di Studio Classic dari pengguna Canvas

Di Studio Classic, jika model telah dibagikan dengan Anda, Anda menerima pemberitahuan yang mirip dengan yang berikut ini saat Anda membuka aplikasi Studio Classic.

Pilih Lihat model bersama untuk membuka halaman Model bersama dan buku catatan di Studio Classic. Jika Anda melewatkan notifikasi, Anda dapat menemukan halaman Model dan buku catatan bersama dengan melakukan hal berikut:

1. Buka aplikasi Amazon SageMaker Studio Classic Anda.
2. Di panel navigasi samping, pilih ikon Beranda ().
3. Di bilah navigasi samping yang terbuka, pilih Model.
4. Dalam daftar tarik-turun, pilih Model bersama untuk membuka halaman Model bersama dan buku catatan.

Pada halaman Model bersama dan buku catatan, pilih filter Dibagikan dengan saya. Anda akan melihat model Canvas yang telah dibagikan dengan Anda dalam daftar model bersama. Pilih Lihat model pada model bersama, yang membuka halaman detail model di Autopilot. Model yang dibuka harus memiliki spanduk di bagian atas yang terlihat mirip dengan tangkapan layar berikut.



Dari halaman ini, Anda dapat melihat detail model, serta catatan tentang model yang dibagikan kepada Anda oleh pengguna Canvas. Di spanduk kanvas di bagian atas, Anda dapat memilih tindakan berikut:

- Bagikan umpan balik dengan pengguna Canvas.
- Buat pembaruan pada model bersama dan bagikan pembaruan dengan pengguna Canvas.
- Bagikan versi alternatif model dengan pengguna Canvas. Canvas menggunakan Autopilot untuk melatih beberapa versi model dan memilih versi terbaik. Anda dapat memilih versi yang berbeda jika Anda memutuskan bahwa itu lebih baik untuk kasus penggunaan Anda.

Untuk informasi selengkapnya tentang tindakan sebelumnya, lihat bagian berikut.

Bagikan umpan balik

Anda mungkin ingin mengirim komentar atau umpan balik ke pengguna Canvas tanpa membuat perubahan apa pun pada model.

Untuk berbagi umpan balik tentang model bersama, gunakan prosedur berikut:

1. Pada halaman detail model, pilih Bagikan umpan balik.
2. Di kotak dialog Bagikan umpan balik, tambahkan catatan di bidang Tambahkan umpan balik.
3. Pilih Bagikan untuk mengirim umpan balik ke pengguna Canvas.

Setelah memberikan umpan balik, Anda dapat melihat umpan balik yang Anda kirim di spanduk Canvas di bagian atas halaman detail model. Pengguna Canvas menerima umpan balik dalam aplikasi Canvas dan dapat membuat perubahan berdasarkan umpan balik Anda.

Bagikan model yang diperbarui dengan pengguna Canvas

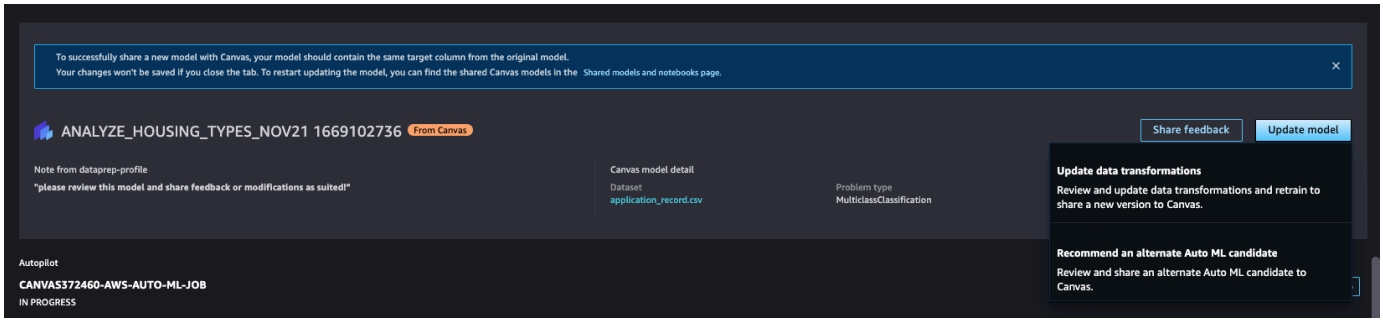
Anda mungkin ingin membuat perubahan pada model yang dibagikan pengguna Canvas dengan Anda. Misalnya, Anda mungkin ingin menggunakan transformasi data lanjutan seperti pengkodean satu panas untuk meningkatkan akurasi model. Anda dapat memperbarui model dengan Amazon

SageMaker Data Wrangler dan Amazon SageMaker Autopilot di Studio Classic, yang merupakan fitur yang membantu Anda melakukan transformasi data dan melatih model Anda.

> **Warning**
>
> Jika Anda keluar dari alur kerja berikut kapan saja, pembaruan model Anda tidak disimpan, dan Anda harus memulai ulang alur kerja.

Untuk memperbarui model dan mengirim model yang diperbarui ke pengguna Canvas, gunakan prosedur berikut:

1. Pada halaman detail model, di spanduk Canvas, pilih Perbarui model.
2. Dalam daftar dropdown banner, pilih Perbarui transformasi data.



3. Alur kerja membuka model Anda di Amazon SageMaker Data Wrangler, di mana Anda dapat memilih untuk mengedit transformasi data yang digunakan untuk model. Buat transformasi data Anda di antarmuka Data Wrangler. Untuk informasi selengkapnya tentang Data Wrangler dan transformasi data yang dapat Anda gunakan, lihat dokumentasi Data Wrangler.
4. Setelah Anda menyelesaikan transformasi data Anda, pilih Retrain model pada banner Canvas untuk membuka Ekspor data dan melatih model dengan halaman SageMaker Autopilot di antarmuka Data Wrangler.
5. Verifikasi bidang pada data Ekspor dan latih model dengan halaman SageMaker Autopilot, lalu pilih Ekspor dan latih untuk mengekspor transformasi data Anda ke Amazon Autopilot. SageMaker
6. Alur kerja membuka halaman eksperimen Create an Autopilot di Autopilot, tempat Anda dapat membuat eksperimen Autopilot dan melatih ulang model dengan transformasi data yang diperbarui. Isi kolom untuk masing-masing halaman percobaan Create an Autopilot.

   Untuk informasi selengkapnya tentang eksperimen Autopilot dan Autopilot, lihat Membuat eksperimen dalam dokumentasi Autopilot.

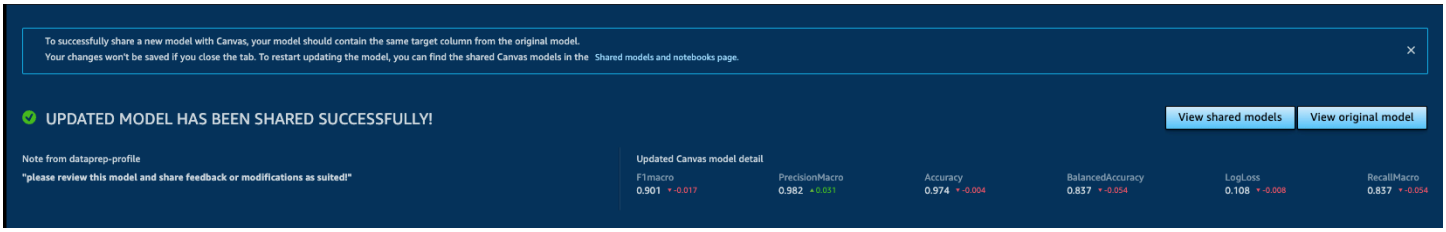
7. Setelah selesai mengonfigurasi eksperimen Autopilot dan meninjau pengaturan akhir, pilih Buat eksperimen di antarmuka Autopilot untuk mulai melatih model. Model kereta, di mana Anda dapat memilih Hentikan pelatihan di antarmuka Autopilot kapan saja.
8. Setelah model dilatih, spanduk Canvas di bagian atas halaman membandingkan metrik model lama dengan model yang diperbarui. Ringkasan model Terbaik mencantumkan metrik, seperti Recall dan Precision, dan apakah model baru meningkatkan metrik atau tidak. Tinjau metrik dan putuskan apakah Anda ingin membagikan model yang diperbarui atau tidak. Untuk informasi selengkapnya tentang metrik Autopilot, lihat Metrik dan validasi.
9. Jika Anda memutuskan ingin membagikan model yang diperbarui dengan pengguna Canvas, pilih Bagikan di spanduk.
10. Di kotak dialog Bagikan, lakukan hal berikut:

    a. Untuk daftar tarik-turun Pilih model untuk dibagikan, model terbaik dari eksperimen Autopilot Anda harus sudah dipilih dan ditandai dengan label Kandidat Terbaik. Jika versi model yang ingin Anda bagikan tidak dipilih, buka dropdown dan pilih versi yang benar.

    b. Untuk bidang Tambahkan umpan balik, Anda dapat memasukkan catatan untuk pengguna Canvas.

    c. Pilih Bagikan untuk membagikan model yang diperbarui dan catat dengan pengguna Canvas.

Setelah membagikan model, Anda menerima pemberitahuan bahwa model Anda berhasil dibagikan mirip dengan tangkapan layar berikut.



Anda dapat memilih Lihat model bersama di spanduk untuk kembali ke halaman Model bersama dan buku catatan. Dari halaman ini, Anda dapat melihat model terbaru yang Anda bagikan dengan pengguna Canvas di bawah label Shared by me.

Bagikan model alternatif dengan pengguna Canvas

Saat SageMaker Canvas membuat model, Amazon SageMaker Autopilot melatih beberapa versi model dan memilih yang terbaik. Anda mungkin memutuskan bahwa versi alternatif dari model lebih baik sesuai dengan kebutuhan Anda. Anda dapat membagikan versi Autopilot alternatif model

dengan pengguna Canvas alih-alih membuat perubahan pada yang mereka kirim. Untuk informasi selengkapnya tentang Autopilot, lihat dokumentasi Autopilot.

Untuk berbagi model alternatif, gunakan prosedur berikut:

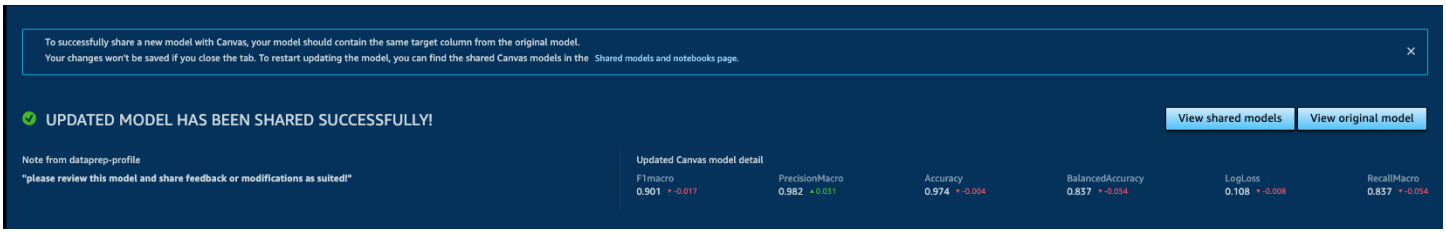
1. Pada halaman detail model, di spanduk Canvas, pilih Perbarui model.
2. Dalam daftar dropdown banner, pilih Rekomendasikan kandidat Auto ML alternatif.
3. Halaman untuk pekerjaan Autopilot terbuka di mana Anda dapat meninjau semua versi model terlatih. Saat Anda siap untuk membagikan versi alternatif, di spanduk Canvas di bagian atas halaman, pilih Bagikan.
4. Di kotak dialog Bagikan, lakukan hal berikut:

    a. Untuk daftar tarik-turun Pilih model untuk dibagikan, model terbaik dari eksperimen Autopilot dipilih dan ditandai dengan label Kandidat Terbaik. Buka dropdown dan pilih versi model alternatif yang ingin Anda bagikan.

    b. Untuk bidang Tambahkan umpan balik, Anda dapat memasukkan catatan untuk pengguna Canvas.

    c. Pilih Bagikan untuk berbagi versi model alternatif dan catat dengan pengguna Canvas.

Setelah membagikan model, Anda menerima pemberitahuan bahwa model alternatif Anda berhasil dibagikan mirip dengan tangkapan layar berikut.



Anda dapat memilih Lihat model bersama di spanduk untuk kembali ke halaman Model bersama dan buku catatan. Dari halaman ini, Anda dapat melihat model terbaru yang Anda bagikan dengan pengguna Canvas di bawah label Shared by me.

Pengguna kanvas: Menerima pembaruan model dari pengguna Studio Classic

Saat pengguna Studio Classic membagikan model yang diperbarui atau alternatif dengan pengguna Canvas, pengguna Canvas akan menerima notifikasi.

Di aplikasi Canvas, notifikasi terlihat seperti tangkapan layar berikut.

Anda dapat memilih Lihat pembaruan untuk melihat model yang diperbarui, atau Anda dapat pergi ke halaman Model di aplikasi Canvas dan memilih model bersama untuk melihatnya.

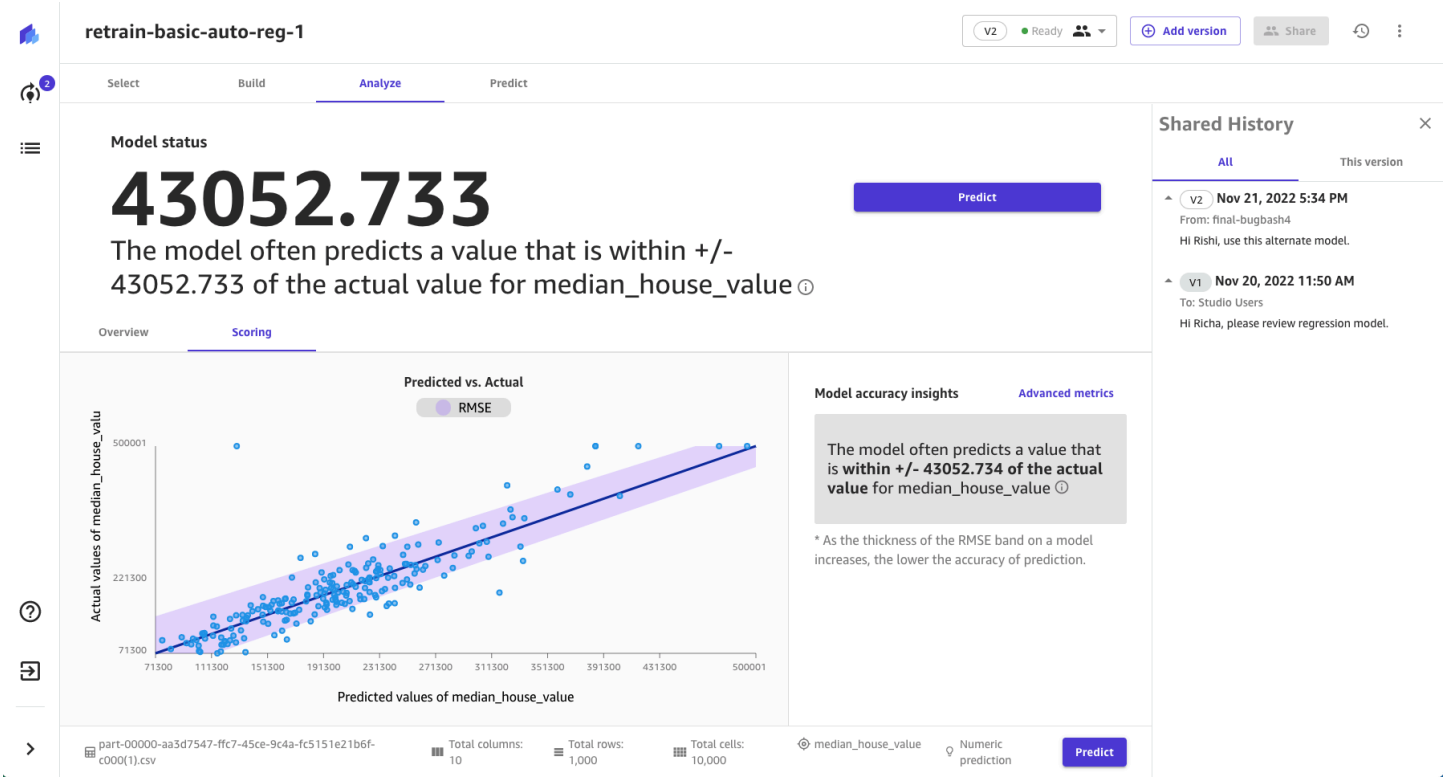
> **Note**
>
> Pengguna Canvas tidak dapat mengedit model yang telah dibagikan dengan mereka oleh pengguna Studio Classic. Model yang diimpor dari Studio Classic hanya dilihat dan diprediksi.

Model di mana pengguna Studio Classic telah berkolaborasi terlihat seperti kartu berikut pada halaman Model.

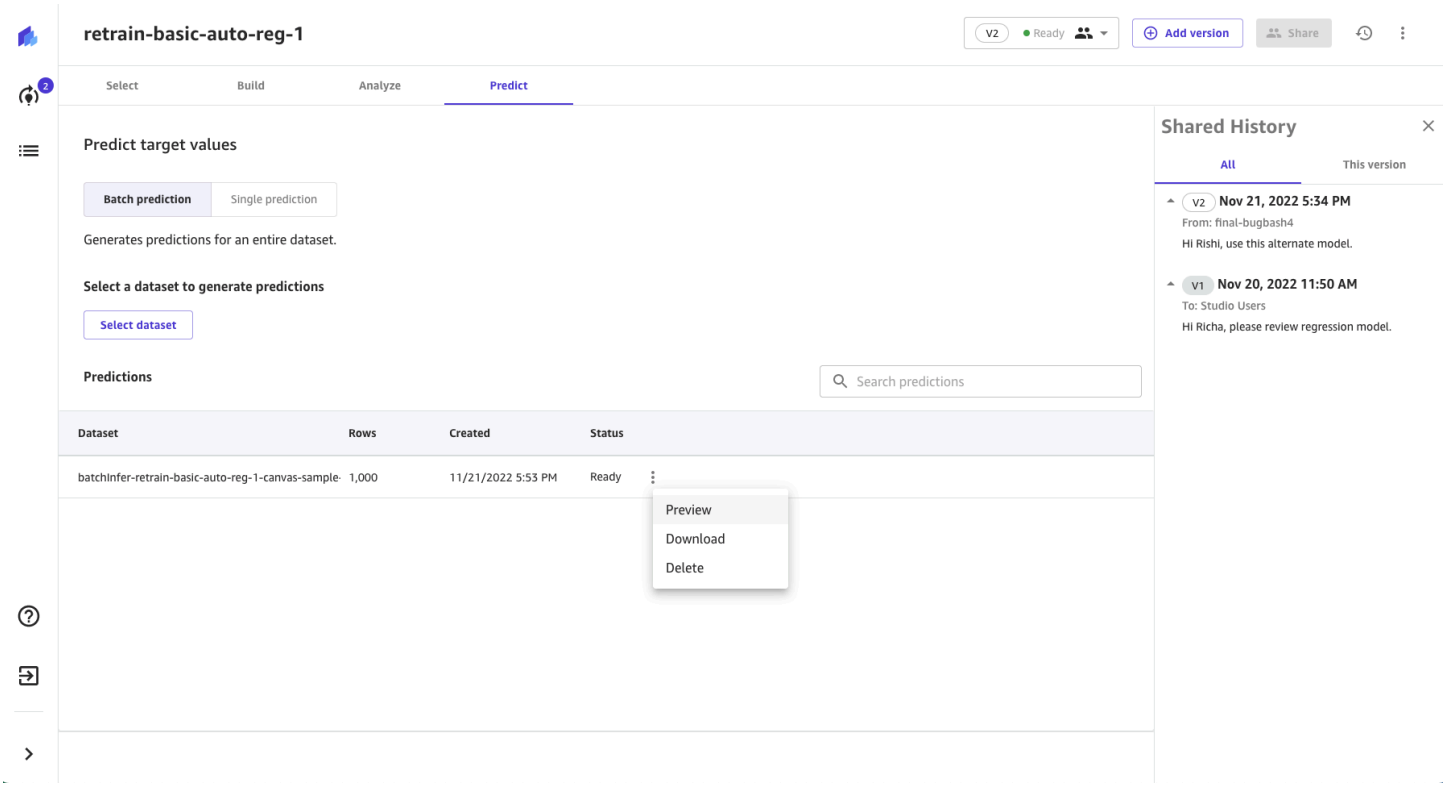
Impor model dari Studio Classic dapat memakan waktu hingga 20 menit, di mana model ditampilkan sebagai Mengimpor.

Setelah mengimpor model, Anda dapat melihat metriknya dan menghasilkan prediksi dengannya.

Tangkapan layar berikut menunjukkan tab Analisis, di mana Anda dapat mengevaluasi akurasi dan metrik model. Untuk informasi selengkapnya, lihat Evaluasi Kinerja Model Anda di Amazon SageMaker Canvas.



Tangkapan layar berikut menunjukkan Predict tab, di mana Anda dapat menghasilkan prediksi dengan model. Untuk informasi selengkapnya tentang menghasilkan prediksi di Canvas, lihatBuat prediksi untuk data Anda.



Pada tab Analisis dan Prediksi, Anda dapat melihat panel Riwayat Bersama, yang menampilkan versi model dan komentar yang dibagikan kepada Anda oleh pengguna Studio Classic.

Bawa model Anda sendiri ke SageMaker Canvas

> **Note**
>
> Fungsionalitas yang dijelaskan di halaman ini hanya berlaku untuk Amazon SageMaker Studio Classic. Saat ini, Anda hanya dapat berbagi model ke Canvas (atau melihat model Canvas bersama) di Studio Classic. Jika saat ini Anda menggunakan Studio versi terbaru, Anda harus menjalankan Studio Classic dari dalam versi terbaru Studio untuk berbagi model ke Canvas atau melihat model yang dibagikan dari Canvas. Untuk informasi selengkapnya tentang mengakses Studio Classic, lihat dokumentasi Studio Classic.

Analis bisnis dapat mengambil manfaat dari model ML yang sudah dibangun oleh ilmuwan data untuk memecahkan masalah bisnis alih-alih membuat model baru di Amazon SageMaker Canvas. Namun, mungkin sulit untuk menggunakan model ini di luar lingkungan di mana mereka dibangun karena persyaratan teknis, kekakuan alat, dan proses manual untuk mengimpor model. Hal ini sering

memaksa pengguna untuk membangun kembali model ML, menghasilkan duplikasi upaya dan waktu dan sumber daya tambahan.

SageMaker Canvas menghapus batasan ini sehingga Anda dapat menghasilkan prediksi di Canvas dengan model yang telah Anda latih di mana saja. Anda dapat mendaftarkan model ML di SageMaker Model Registry, yang merupakan penyimpanan metadata untuk model ML, dan mengimpornya ke SageMaker Canvas. Selain itu, Anda dapat menghasilkan prediksi dengan model yang telah dilatih oleh ilmuwan data di Amazon SageMaker Autopilot atau. SageMaker JumpStart Pengguna Canvas kemudian dapat menganalisis dan menghasilkan prediksi dari model apa pun yang telah dibagikan dengan mereka.

Setelah Anda puasPrasyarat, lihat bagian berikut untuk petunjuk tentang cara membawa model Anda sendiri ke Canvas dan menghasilkan prediksi. Alur kerja dimulai di Studio Classic, di mana pengguna Studio Classic berbagi model dengan pengguna Canvas. Kemudian, pengguna Canvas masuk ke aplikasi Canvas mereka untuk menerima model bersama dan menghasilkan prediksi dengannya.

> **Note**
>
> Anda dapat berbagi model yang dilatih dengan data tabel, teks, dan gambar ke Canvas. Anda tidak dapat berbagi model deret waktu. Selain itu, Canvas membawa model Anda sendiri (BYOM) hanya mendukung model berbasis CPU (atau model yang menggunakan instance CPU untuk membuat prediksi).

Prasyarat

Untuk membawa model Anda ke SageMaker Canvas, lengkapi prasyarat berikut:

- Anda harus memiliki pengguna Amazon SageMaker Studio Classic yang telah onboard ke Domain Amazon SageMaker . Pengguna Studio Classic harus berada di Domain yang sama dengan pengguna Canvas. Berbagi model terjadi ketika pengguna Studio Classic berbagi model dengan pengguna Canvas dari dalam Studio Classic. Jika Anda belum menyiapkan pengguna Studio Classic, lihat dokumentasi Studio Classic dan Onboard to Amazon SageMaker Domain.
- Anda harus memiliki model terlatih dari SageMaker Autopilot, SageMaker JumpStart, atau SageMaker Model Registry. Untuk model apa pun yang Anda buat di luar SageMaker, Anda harus mendaftarkan model Anda di Model Registry sebelum mengimpornya ke Canvas. Untuk informasi selengkapnya, lihat dokumentasi Model Registry.

- Pengguna Canvas yang ingin Anda bagikan model Anda harus memiliki izin untuk mengakses bucket Amazon S3 tempat Anda menyimpan kumpulan data dan artefak model. Untuk petunjuk tentang bagaimana admin dapat memberi pengguna Canvas izin yang mereka butuhkan, lihat. Berikan Izin Pengguna untuk Berkolaborasi dengan Studio Classic
- Anda juga harus memiliki nama profil pengguna dari pengguna Canvas dengan siapa Anda ingin berkolaborasi. Pengguna Canvas harus berada di SageMaker Domain Amazon yang sama dengan pengguna Studio Classic Anda. Anda dapat menemukan nama profil pengguna dengan menggunakan prosedur berikut:

  1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
  2. Di panel navigasi, pilih Domain.
  3. Dari daftar Domain, pilih Domain Anda. Ini membuka halaman detail Domain, di mana Anda dapat menemukan semua profil Pengguna untuk Domain.

  Siapkan nama profil pengguna untuk langkah pertama dari tutorial berikut.

Jika aplikasi SageMaker Canvas Anda berjalan di VPC pelanggan pribadi, model Autopilot apa pun yang dibagikan dari Studio Classic harus menggunakan mode Autopilot HPO untuk mendukung pembuatan prediksi di Canvas. Untuk informasi selengkapnya tentang mode HPO, lihat Mode pelatihan dan dukungan algoritme dalam dokumentasi Autopilot.

> **Note**
>
> Jika Anda menginginkan umpan balik dari ilmuwan data tentang model yang dibangun di dalam CanvasBerkolaborasi dengan ilmuwan data, lihat, di mana pengguna Canvas berbagi model dengan pengguna Studio Classic, dan pengguna Studio Classic membagikan umpan balik atau pembaruan model.

Pengguna Studio Classic: Bagikan model ke SageMaker Canvas

Anda harus memiliki model yang dilatih dengan data tabular yang siap Anda bagikan dengan pengguna Canvas. Lihat bagian berikut untuk informasi tentang cara membagikan model Anda dari fitur dalam Studio Classic.

Autopilot

Anda dapat membagikan model ke Canvas dari Amazon SageMaker Autopilot di Studio Classic. Autopilot adalah fitur yang memungkinkan Anda untuk melatih dan menerapkan model Anda. SageMaker

Anda harus memiliki pengguna Studio Classic dan model terlatih yang siap dibagikan dari Autopilot. Untuk informasi selengkapnya tentang cara mengatur Studio Classic, lihat dokumentasi Studio Classic. Untuk informasi selengkapnya tentang Autopilot, lihat dokumentasi Autopilot.

Untuk berbagi model dari Autopilot ke Canvas, gunakan prosedur berikut.

1. Buka aplikasi Amazon SageMaker Studio Classic Anda.
2. Di panel navigasi samping, pilih ikon Beranda ().
3. Di bilah navigasi samping Studio Classic, pilih AutoML untuk membuka Autopilot.
4. Pada halaman Autopilot, pilih model Autopilot yang ingin Anda bagikan dengan pengguna Canvas. Anda hanya dapat berbagi satu model pada satu waktu.
5. Dari halaman detail pekerjaan Autopilot, di tab Model, pilih versi model yang ingin Anda bagikan.
6. Pilih Bagikan.
7. Dalam kotak dialog Bagikan model, lakukan hal berikut:

    a. Untuk bidang Add Canvas users, masukkan nama profil pengguna Canvas. Anda dapat memasukkan hingga 23 pengguna Canvas. Jika profil pengguna yang Anda tentukan tidak memiliki aplikasi Canvas yang terkait dengannya, Anda tidak dapat memasukkan nama profil.

    b. Untuk bidang Tambahkan catatan, tambahkan deskripsi atau catatan untuk pengguna Canvas saat mereka menerima model.

    c. Pilih Bagikan untuk membagikan model.

Anda sekarang telah berbagi model dengan pengguna Canvas.

JumpStart

Anda dapat membagikan model ke Canvas dari SageMaker JumpStart Studio Classic. Dengan JumpStart, Anda dapat mengakses dan menyetel model yang telah dilatih sebelumnya sebelum menerapkannya.

Anda harus memiliki pengguna Studio Classic dan pekerjaan pelatihan yang berhasil diselesaikan di JumpStart. Untuk informasi selengkapnya tentang cara mengatur Studio Classic, lihat dokumentasi Studio Classic. Untuk informasi selengkapnya JumpStart, lihat JumpStart dokumentasi.

Untuk berbagi model dari JumpStart ke Canvas, gunakan prosedur berikut.

1. Buka aplikasi Amazon SageMaker Studio Classic Anda.
2. Di panel navigasi samping, pilih ikon Beranda
   ().
3. Di bilah navigasi samping yang terbuka, pilih SageMaker JumpStart.
4. Pilih JumpStart Aset yang diluncurkan untuk membuka halaman yang mencantumkan pekerjaan, model, dan titik akhir JumpStart pelatihan Anda.
5. Pilih tab Pekerjaan pelatihan untuk melihat daftar pekerjaan pelatihan model Anda.
6. Dari daftar pekerjaan Pelatihan, pilih pekerjaan pelatihan yang ingin Anda bagikan dengan pengguna Canvas. Anda hanya dapat berbagi satu pekerjaan dalam satu waktu. Ini membuka halaman detail pekerjaan pelatihan.
7. Di header untuk pekerjaan pelatihan, pilih Bagikan, dan pilih Bagikan ke Kanvas.

   > **Note**
   >
   > Anda hanya dapat berbagi model tabular ke Canvas. Mencoba membagikan model yang tidak berbentuk tabel menimbulkan kesalahan`Unsupported data type`.

8. Dalam kotak dialog Bagikan ke Kanvas, lakukan hal berikut:

   a. Untuk bidang Tambahkan pengguna Canvas untuk berbagi, masukkan nama profil pengguna Canvas. Anda dapat memasukkan hingga 23 pengguna Canvas. Jika profil pengguna yang Anda tentukan tidak memiliki aplikasi Canvas yang terkait dengannya, Anda tidak dapat memasukkan nama profil.

   b. Untuk bidang Tambahkan catatan, tambahkan deskripsi atau catatan untuk pengguna Canvas saat mereka menerima model.

   c. Pilih Bagikan untuk membagikan model.

Anda sekarang telah berbagi model dengan pengguna Canvas.

Registri Model

Anda dapat berbagi model ke Canvas dari SageMaker Model Registry di Studio Classic. Dengan Model Registry, Anda dapat mendaftarkan model yang Anda bawa dari luar SageMaker dan mengintegrasikannya dengan pipeline ML Anda.

Anda harus memiliki pengguna Studio Classic dan versi model yang disimpan di Registry Model. Untuk informasi selengkapnya tentang cara mengatur Studio Classic, lihat dokumentasi Studio Classic. Jika Anda tidak memiliki versi model di Registri Model, buat grup model dan daftarkan versinya. Untuk informasi selengkapnya tentang Model Registry, lihat dokumentasi Model Registry.

Untuk berbagi versi model dari Model Registry ke Canvas, gunakan prosedur berikut.

1. Buka aplikasi Amazon SageMaker Studio Classic Anda.
2. Di panel navigasi samping, pilih ikon Beranda
   ( ).
3. Di bilah navigasi samping yang terbuka, pilih Model.
4. Pilih Model Registry dari daftar dropdown untuk membuka halaman Model Registry dan menampilkan semua grup model yang terdaftar di akun Anda.
5. Pilih grup model yang memiliki versi model yang ingin Anda bagikan.
6. Anda dapat membagikan versi model baik dari halaman grup model atau halaman versi model.
   - Untuk membagikan versi model dari halaman grup model, selesaikan langkah-langkah berikut:
     1. Pilih Versi, dan centang kotak di sebelah versi model yang ingin Anda bagikan dengan pengguna Canvas. Anda hanya dapat berbagi satu versi model pada satu waktu.
     2. Di menu tarik-turun Tindakan, pilih Bagikan artefak model.
   - Untuk membagikan versi model dari halaman versi model, selesaikan langkah-langkah berikut:
     1. Pilih Versi, dan pilih nama versi model yang ingin Anda bagikan dengan pengguna Canvas. Anda hanya dapat berbagi satu versi model pada satu waktu.
     2. Di menu tarik-turun Tindakan, pilih Bagikan artefak model.
7. Dalam kotak dialog Bagikan model, lakukan hal berikut:
   a. Untuk bidang Tambahkan pengguna Canvas untuk berbagi, masukkan nama profil pengguna Canvas. Anda dapat memasukkan hingga 23 pengguna Canvas. Jika profil

pengguna yang Anda tentukan tidak memiliki aplikasi Canvas yang terkait dengannya, Anda tidak dapat memasukkan nama profil.

b. Untuk Tambahkan detail model, lakukan hal berikut:

   i. Untuk bidang Dataset pelatihan, masukkan jalur Amazon S3 untuk kumpulan data pelatihan Anda.

   ii. Untuk bidang kumpulan data Validasi, masukkan jalur Amazon S3 untuk kumpulan data validasi Anda.

   iii. Untuk kolom Target, pilih Gunakan kolom pertama jika kolom pertama dalam kumpulan data Anda adalah target, atau pilih Tentukan nama kolom target untuk menetapkan target sebagai kolom yang berbeda di kumpulan data Anda.

   iv. Untuk header Kolom, pilih salah satu opsi berikut:

      A. Pilih Gunakan baris pertama jika baris pertama kumpulan data Anda berisi header kolom.

      B. Pilih Tentukan kumpulan data yang berbeda di S3 untuk header kolom jika Anda memiliki file yang disimpan di Amazon S3 yang berisi header yang dapat dipetakan ke kumpulan data Anda. File header harus memiliki jumlah kolom yang sama dengan kumpulan data Anda.

      C. Pilih Buat secara otomatis jika Anda belum memiliki header kolom dan ingin SageMaker menghasilkan nama kolom generik untuk kumpulan data Anda.

   v. Dari daftar dropdown tipe Masalah, pilih jenis model Anda.

   vi. Jika Anda memilih jenis masalah klasifikasi biner atau Multi-class, opsi Configure model output akan muncul.

      Jika Anda sudah memiliki file yang disimpan di Amazon S3 yang memetakan nama kelas kolom target default ke nama kelas yang Anda inginkan, kemudian aktifkan Nama keluaran Model dan masukkan jalur Amazon S3 ke file pemetaan. Jika Anda tidak memiliki file pemetaan, matikan nama keluaran Model dan masukkan Numer output model secara manual (jumlah kelas kolom target dalam data Anda). Kemudian, masukkan nama kelas yang Anda inginkan untuk mengganti nama kelas default.

c. (Opsional) Untuk bidang Tambahkan catatan, tambahkan deskripsi atau catatan untuk pengguna Canvas saat mereka menerima model.

d. Pilih Bagikan untuk membagikan versi model.

Anda sekarang telah berbagi model dengan pengguna Canvas.

Model dan notebook bersama

Pada halaman Model bersama dan buku catatan di Amazon SageMaker Studio Classic, Anda dapat melihat model yang telah Anda bagikan dan yang telah dibagikan kepada Anda. Halaman ini memberi Anda tempat sentral untuk melihat dan mengelola semua model Anda di Studio Classic.

Anda harus memiliki pengguna Studio Classic dan model yang siap dibagikan dari Autopilot, JumpStart, atau Model Registry. Untuk informasi selengkapnya tentang cara mengatur Studio Classic, lihat dokumentasi Studio Classic. Untuk informasi selengkapnya tentang halaman Model bersama dan buku catatan, lihat dokumentasi Model bersama dan buku catatan.

Contoh berikut memandu Anda melalui berbagi model SageMaker Autopilot Amazon, tetapi Anda dapat menggunakan fitur berbagi pada halaman Model bersama dan notebook untuk berbagi model dari salah satu fitur lain di bagian sebelumnya, seperti Jumpstart dan Model Registry.

Untuk membagikan model Autopilot dari halaman Model bersama dan buku catatan, gunakan prosedur berikut.

1. Buka aplikasi Amazon SageMaker Studio Classic Anda.
2. Di panel navigasi samping, pilih ikon Beranda
   ( ).
3. Di bilah navigasi samping Studio Classic, pilih Model.
4. Dalam daftar tarik-turun, pilih Model bersama untuk membuka halaman Model bersama dan buku catatan.
5. Pilih ikon filter, dan dalam daftar dropdown Shared from, pilih Autopilot.
6. Pilih model Autopilot dari daftar yang ingin Anda bagikan dengan pengguna Canvas. Anda hanya dapat berbagi satu model pada satu waktu. Atau, Anda dapat memilih model untuk membuka halaman detail model.
7. Dari halaman pekerjaan Autopilot atau halaman detail model, pilih Bagikan.
8. Dalam kotak dialog Bagikan model, lakukan hal berikut:

   a. Untuk bidang Tambahkan pengguna Canvas untuk berbagi, masukkan nama profil pengguna Canvas. Anda dapat memasukkan hingga 23 pengguna Canvas. Jika profil pengguna yang Anda tentukan tidak memiliki aplikasi Canvas yang terkait dengannya, Anda tidak dapat memasukkan nama profil.

b. Untuk bidang Tambahkan catatan, tambahkan deskripsi atau catatan untuk pengguna Canvas saat mereka menerima model.

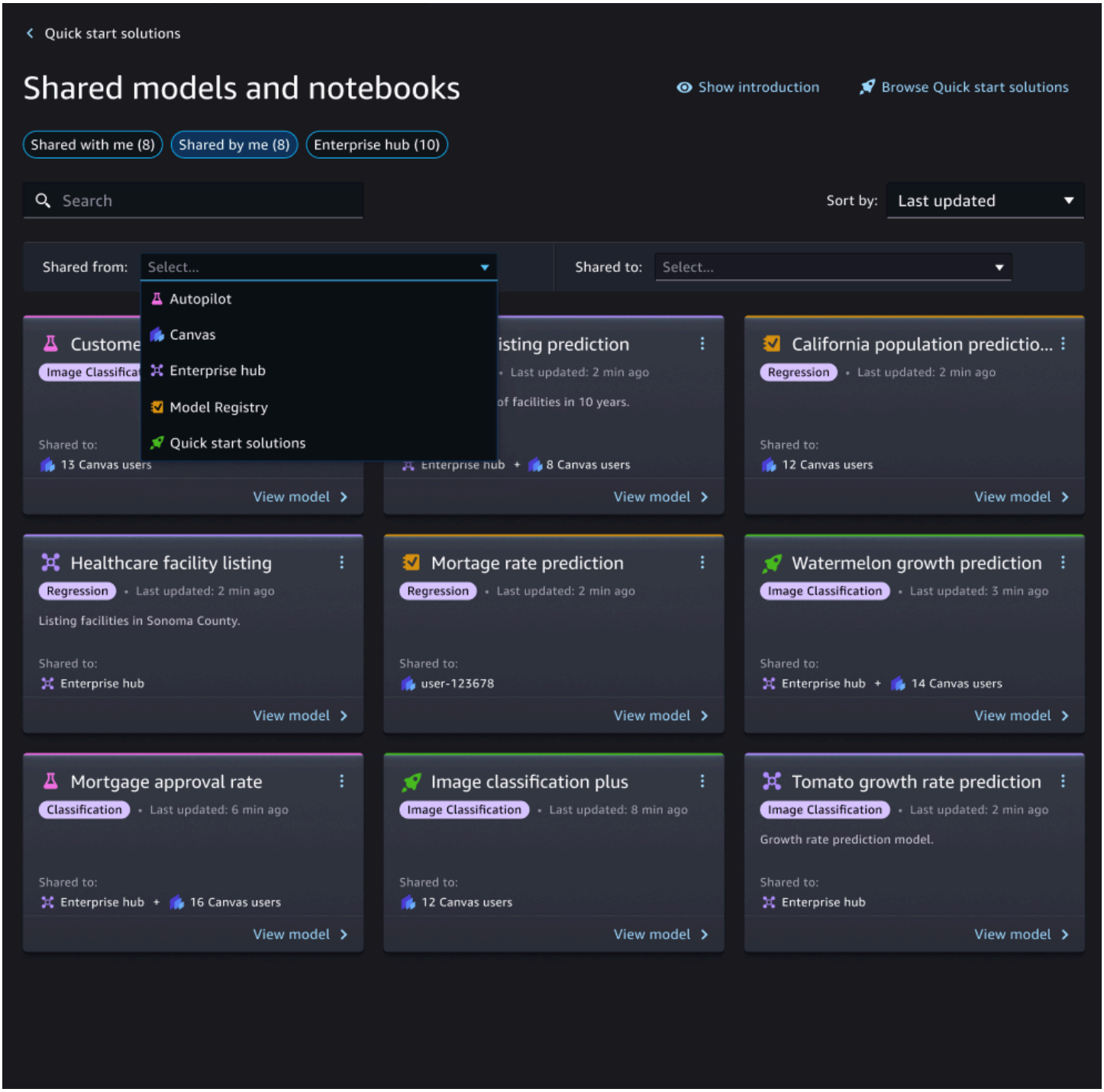
c. Pilih Bagikan untuk membagikan model.

Anda sekarang telah berbagi model dengan pengguna Canvas.

Setelah Anda membagikan model, Anda menerima popup notifikasi di Studio Classic mirip dengan tangkapan layar berikut.

Anda dapat memilih Lihat model untuk membuka halaman Shared models dan notebook di Studio Classic. Anda juga dapat melihat model bersama kapan saja dari halaman Model bersama dan buku catatan.

Dari halaman ini, Anda dapat melihat model yang telah Anda bagikan dengan pengguna Canvas di bawah label Shared by me, seperti yang ditunjukkan pada gambar berikut.

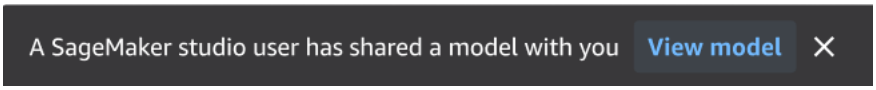


Model yang telah Anda bagikan ke Canvas memiliki teks pada kartu yang mirip dengan contoh berikut:`Shared to: 12 Canvas users`.

Pengguna kanvas: Menerima model bersama di SageMaker Canvas

Ketika pengguna Studio Classic berbagi model dengan pengguna Canvas, Anda menerima pemberitahuan dalam aplikasi Canvas bahwa pengguna Studio Classic telah berbagi model dengan Anda.

Dalam aplikasi Canvas, notifikasi mirip dengan tangkapan layar berikut.



Anda dapat memilih Lihat pembaruan untuk melihat model bersama, atau Anda dapat pergi ke halaman Model di aplikasi Canvas untuk menemukan semua model yang telah dibagikan dengan Anda.

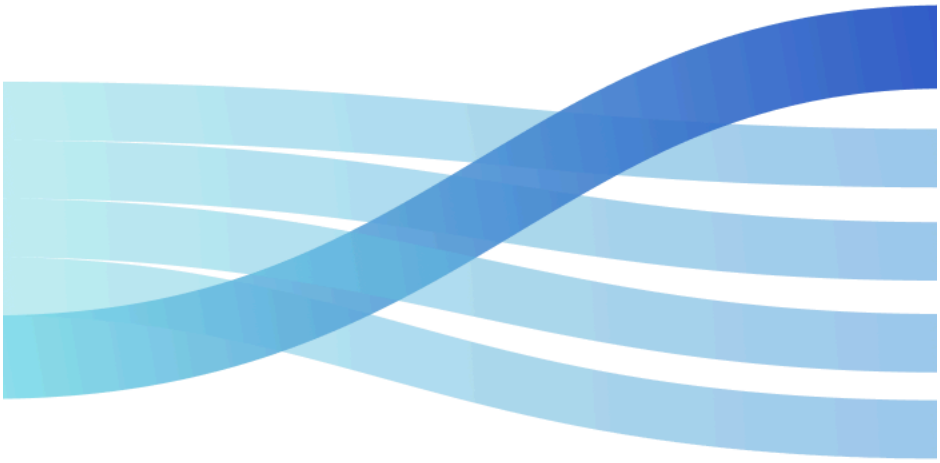
> **Note**
>
> Pengguna Canvas tidak dapat mengedit model yang telah dibagikan dengan mereka oleh pengguna Studio Classic. Model yang diimpor dari Studio Classic hanya dilihat dan diprediksi.

Model yang telah dibagikan oleh pengguna Studio Classic terlihat seperti kartu berikut di halaman Model. Ini berbeda dariBerkolaborasi dengan ilmuwan data, di mana pengguna Canvas berbagi model dan pengguna Studio Classic berbagi pembaruan atau umpan balik dengan pengguna Canvas.

Importing

Studio

Customer Churn Model



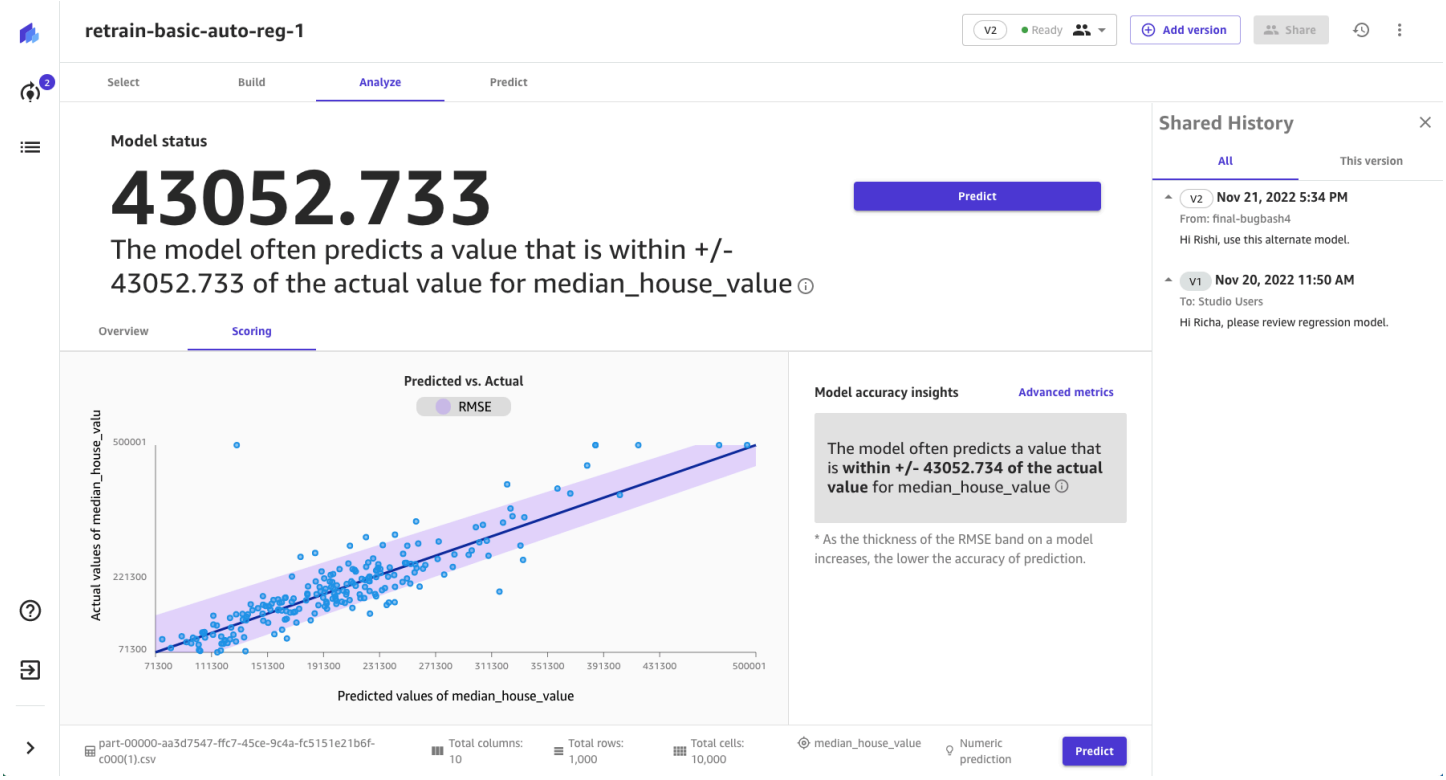
Accuracy	--
Dataset	--
Target	Plan
Problem type	Multiclass
Received	7/3/2021 18:11

View

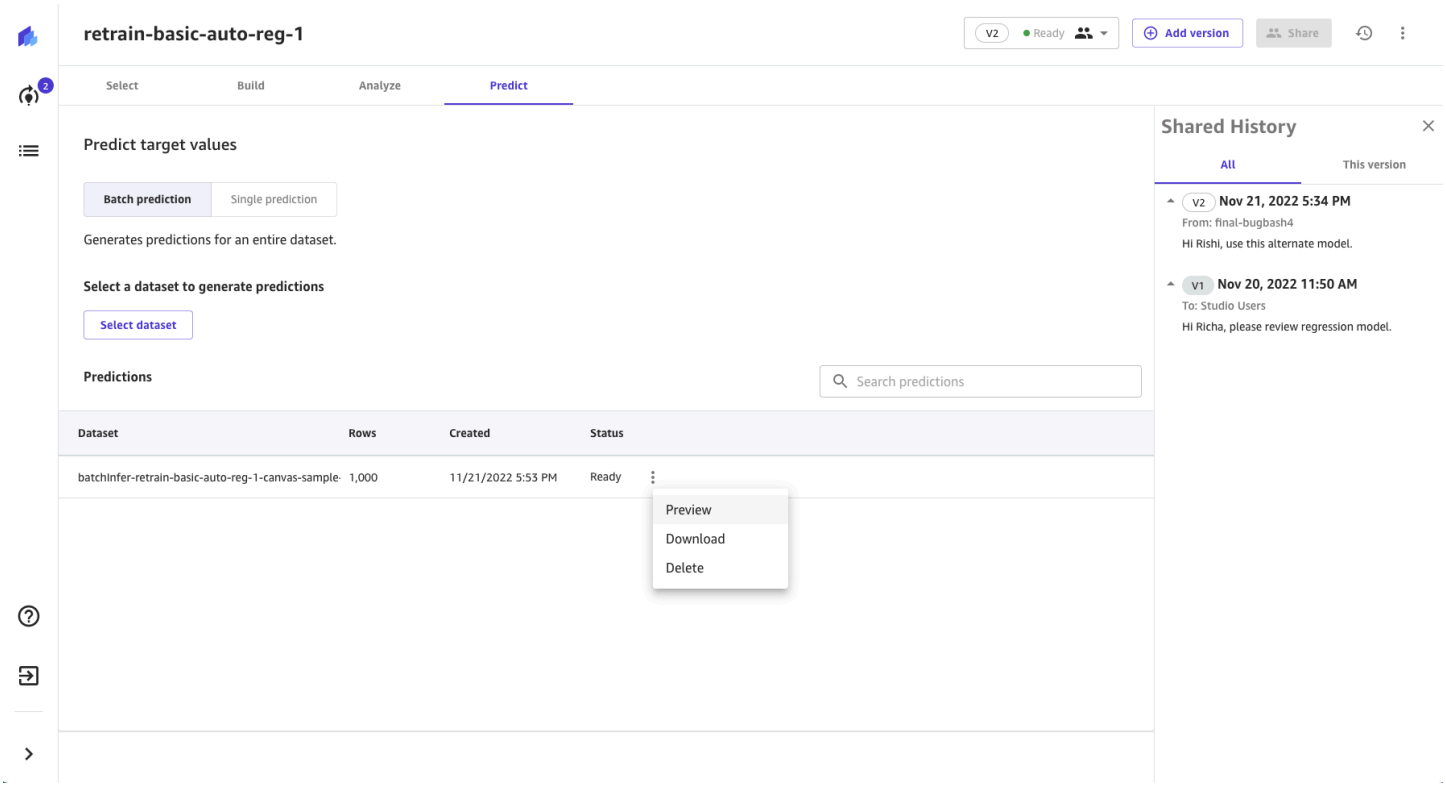
Impor model dari Studio Classic dapat memakan waktu hingga 20 menit, di mana model ditampilkan sebagai Mengimpor.

Setelah mengimpor model, Anda dapat melihat metriknya dan menghasilkan prediksi dengannya. SageMaker Canvas menggunakan sumber daya Inferensi SageMaker Tanpa Server Amazon untuk menghasilkan analisis model dan prediksi untuk model bersama. Anda mungkin melihat biaya yang terkait dengan Inferensi Tanpa Server di akun Anda. AWS

Tangkapan layar berikut menunjukkan tab Analisis di aplikasi Canvas untuk model bersama, di mana Anda dapat mengevaluasi akurasi dan metrik model. Untuk informasi selengkapnya, lihat Evaluasi Kinerja Model Anda di Amazon SageMaker Canvas.



Tangkapan layar berikut menunjukkan Predict tab, di mana Anda dapat menghasilkan prediksi dengan model. Untuk informasi selengkapnya tentang menghasilkan prediksi di Canvas, lihatBuat prediksi untuk data Anda.



Pada tab Analisis dan Prediksi, Anda dapat melihat panel Riwayat Bersama, yang menampilkan versi model dan komentar yang dibagikan kepada Anda oleh pengguna Studio Classic.

Keluar dari Amazon SageMaker Canvas

Jika Anda tidak aktif menggunakan Amazon SageMaker Canvas, kami sarankan Anda keluar atau mengonfigurasi aplikasi Anda untuk dimatikan secara otomatis. Segera setelah Anda meluncurkan aplikasi Canvas, instance ruang kerja didedikasikan untuk Anda gunakan, dan Anda akan ditagih selama Anda masuk. Keluar atau shutdown otomatis mengakhiri instance ruang kerja. Untuk informasi lebih lanjut, lihat SageMaker Harga.

Bagian berikut menjelaskan cara keluar dari aplikasi Canvas Anda dan cara mengonfigurasi aplikasi Anda untuk mematikan jadwal secara otomatis.

Keluar dari Canvas

Saat Anda keluar dari Canvas, model dan kumpulan data Anda tidak terpengaruh, tetapi SageMaker Canvas membatalkan tugas pembuatan Cepat apa pun. Jika Anda keluar dari SageMaker Canvas saat menjalankan Quick build, build Anda mungkin akan terganggu hingga Anda masuk kembali. Saat Anda masuk kembali, SageMaker Canvas secara otomatis memulai ulang build. Build standar berlanjut bahkan jika Anda keluar.

Untuk keluar, pilih tombol Keluar
( )
di panel kiri aplikasi SageMaker Canvas.

Anda juga dapat keluar dari aplikasi SageMaker Canvas dengan menutup tab browser Anda dan kemudian menghapus aplikasi di konsol.

Setelah Anda keluar, SageMaker Canvas memberitahu Anda untuk meluncurkan kembali di tab yang berbeda. Login membutuhkan waktu antara 3 menit dan 8 menit. Jika Anda memiliki administrator yang mengatur SageMaker Canvas untuk Anda, gunakan instruksi yang mereka berikan kepada Anda untuk masuk kembali. Jika tidak memiliki administrator, lihat prosedur untuk mengakses SageMaker Canvas diPrasyarat untuk menyiapkan Amazon Canvas SageMaker .

Matikan Canvas secara otomatis

Jika Anda seorang administrator Canvas, Anda mungkin ingin secara teratur mematikan aplikasi untuk mengurangi biaya. Anda dapat membuat jadwal untuk mematikan aplikasi Canvas aktif, atau Anda dapat membuat otomatisasi untuk mematikan aplikasi Canvas segera setelah mereka menganggur (artinya pengguna belum aktif selama 2 jam).

Anda dapat membuat solusi ini menggunakan AWS Lambda fungsi yang memanggil `DeleteApp` API dan menghapus aplikasi Canvas dengan kondisi tertentu. Untuk informasi selengkapnya tentang solusi ini dan akses ke AWS CloudFormation template yang dapat Anda gunakan, lihat blog Mengoptimalkan biaya untuk Amazon SageMaker Canvas dengan shutdown otomatis aplikasi idle.

> **Note**
>
> Anda mungkin mengalami CloudWatch metrik Amazon yang hilang jika ada kesalahan saat mengatur jadwal penutupan idle atau kesalahan. CloudWatch Kami menyarankan Anda menambahkan CloudWatch alarm yang memantau metrik yang hilang. Jika Anda mengalami masalah ini, hubungi AWS Support bantuan.

Keterbatasan dan pemecahan masalah

Bagian berikut menguraikan bantuan pemecahan masalah dan batasan yang berlaku saat menggunakan Amazon Canvas. SageMaker Anda dapat menggunakan topik ini untuk membantu memecahkan masalah apa pun yang Anda temui.

Memecahkan masalah dengan pemberian izin melalui konsol SageMaker

Jika Anda mengalami masalah dalam memberikan izin dasar Canvas atau izin eady-to-use model R kepada pengguna Anda, pengguna Anda mungkin memiliki peran eksekusi AWS IAM dengan lebih dari satu hubungan kepercayaan dengan layanan lain. AWS Hubungan kepercayaan adalah kebijakan yang melekat pada peran Anda yang menentukan prinsip mana (pengguna, peran, akun, atau layanan) yang dapat mengambil peran tersebut. Misalnya, Anda mungkin mengalami masalah saat memberikan izin Canvas tambahan kepada pengguna jika peran eksekusi mereka memiliki hubungan kepercayaan dengan Amazon SageMaker dan Amazon Forecast.

Anda dapat memperbaiki masalah ini dengan memilih salah satu opsi berikut.

1. Hapus semua kecuali satu layanan tepercaya dari peran.

Solusi ini mengharuskan Anda untuk mengedit hubungan kepercayaan untuk peran IAM profil pengguna Anda dan menghapus semua AWS layanan kecuali SageMaker.

Untuk mengedit hubungan kepercayaan untuk peran eksekusi IAM Anda, lakukan hal berikut:

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi konsol IAM, pilih Peran. Konsol tersebut menampilkan peran di akun Anda.
3. Pilih nama peran yang ingin Anda ubah, dan pilih tab Hubungan kepercayaan pada halaman detail.
4. Pilih Edit kebijakan kepercayaan.
5. Di editor kebijakan Edit kepercayaan, tempel yang berikut ini, lalu pilih Perbarui Kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Anda juga dapat memperbarui dokumen kebijakan ini menggunakan IAM CLI. Untuk informasi selengkapnya, lihat update-trust di Referensi Baris Perintah IAM.

Sekarang Anda dapat mencoba lagi memberikan izin dasar Canvas atau izin eady-to-use model R kepada pengguna Anda.

2. Gunakan peran yang berbeda dengan satu atau lebih sedikit layanan tepercaya.

Solusi ini mengharuskan Anda untuk menentukan peran IAM yang berbeda untuk profil pengguna Anda. Gunakan opsi ini jika Anda sudah memiliki peran IAM yang dapat Anda ganti.

Untuk menentukan peran eksekusi yang berbeda bagi pengguna Anda, lakukan hal berikut:

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain yang ingin Anda lihat daftar profil pengguna.
5. Pada halaman Detail domain, pilih tab Profil pengguna.
6. Pilih pengguna yang izinnya ingin Anda edit. Pada halaman Detail pengguna, pilih Edit.
7. Pada halaman Pengaturan umum, pilih daftar tarik-turun peran Eksekusi dan pilih peran yang ingin Anda gunakan.
8. Pilih Kirim untuk menyimpan perubahan Anda ke profil pengguna.

Pengguna Anda sekarang harus menggunakan peran eksekusi dengan hanya satu layanan tepercaya (SageMaker).

Anda dapat mencoba lagi memberikan izin dasar Canvas atau izin eady-to-use model R kepada pengguna Anda.

3. Lampirkan kebijakan AWS terkelola secara manual ke peran eksekusi alih-alih menggunakan sakelar di setelan SageMaker Domain.

Alih-alih menggunakan sakelar di setelan Domain atau profil pengguna, Anda dapat melampirkan kebijakan AWS terkelola secara manual yang memberikan izin yang benar kepada pengguna.

Untuk memberikan izin dasar Canvas pengguna, lampirkan AmazonSageMakerCanvasFullAccesskebijakan. Untuk memberikan izin eady-to-use model R pengguna, lampirkan ServicesAccess kebijakan AmazonSageMakerCanvasAI.

Gunakan prosedur berikut untuk melampirkan kebijakan AWS terkelola ke peran Anda:

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pilih Peran.
3. Di kotak pencarian, cari peran IAM pengguna berdasarkan nama dan pilih.
4. Pada halaman untuk peran pengguna, di bawah Izin, pilih Tambahkan izin.
5. Dari menu tarik-turun, pilih Lampirkan kebijakan.
6. Cari dan pilih kebijakan atau kebijakan yang ingin dilampirkan ke peran eksekusi pengguna:
   a. Untuk memberikan izin dasar Canvas, cari dan pilih AmazonSageMakerCanvasFullAccesskebijakan.
   b. Untuk memberikan izin eady-to-use model R, cari dan pilih ServicesAccess kebijakan AmazonSageMakerCanvasAI.
7. Pilih Tambahkan izin untuk melampirkan kebijakan ke peran.

Setelah melampirkan kebijakan AWS terkelola ke peran pengguna melalui konsol IAM, pengguna Anda sekarang harus memiliki izin dasar Canvas atau izin model Ready-to-use .

Keterbatasan untuk kolaborasi

Keterbatasan umum berikut berlaku saat Anda berkolaborasi dengan ilmuwan data di Amazon SageMaker Studio Classic.

- Anda hanya dapat berbagi model yang berhasil dilatih dari Canvas ke Studio Classic. Demikian pula, Anda hanya dapat berbagi model yang telah berhasil dilatih di Studio Classic kembali ke Canvas.
- Anda tidak dapat membagikan model Quick build dari Canvas ke Studio Classic. Anda hanya dapat berbagi model build Standar.
- Anda hanya dapat membagikan satu versi model build Standar yang dilatih di Canvas. Anda dapat melatih versi tambahan model Anda dalam Canvas, tetapi Anda tidak dapat membagikannya ke Studio Classic.
- Dari Studio Classic, Anda hanya dapat berbagi umpan balik atau berbagi model yang diperbarui dengan Canvas. Anda tidak dapat melakukan kedua tindakan pada saat yang sama.
- Batasan panjang untuk komentar yang dibagikan dari Studio Classic ke Canvas dan Canvas ke Studio Classic adalah 1024 karakter.

- Anda hanya dapat membagikan model Canvas atau Studio Classic Anda dengan profil pengguna yang berbeda. Anda tidak dapat berbagi model antara Canvas dan Studio Classic dalam profil pengguna Anda sendiri.
- Anda tidak dapat berbagi dari pengguna Canvas ke pengguna Canvas, atau dari pengguna Studio Classic ke pengguna Studio Classic.

Ada juga batasan yang berlaku tergantung pada jenis model yang ingin Anda bagikan. Lihat bagian berikut untuk batasan model peramalan deret waktu dan model prediksi numerik dan kategoris.

Keterbatasan untuk berkolaborasi pada model peramalan deret waktu

Batasan berikut berlaku saat Anda berkolaborasi pada model peramalan deret waktu antara Canvas dan Studio Classic.

- Anda tidak dapat membuat prediksi dengan model peramalan deret waktu di Studio Classic melalui tombol Bagikan otomatis. Namun, Anda dapat membuat notebook Jupyter dan menulis kode Anda sendiri.
- Untuk model peramalan deret waktu, Anda tidak dapat mengubah resep model atau transformasi data di Studio Classic. Anda hanya dapat melakukan pembaruan berikut untuk model peramalan deret waktu di Studio Classic:
  - Anda dapat memperbarui panjang cakrawala perkiraan.
  - Anda dapat memperbarui bidang metadata item, yang mengelompokkan data Anda berdasarkan kolom tertentu.
  - Anda dapat memperbarui bidang dimensi lainnya, seperti menentukan jadwal liburan.

Keterbatasan untuk berkolaborasi pada model prediksi numerik dan kategoris

Batasan berikut berlaku saat Anda berkolaborasi pada jenis model prediksi numerik dan kategoris antara Canvas dan Studio Classic.

- Saat memperbarui atau melatih model di Studio Classic, jika Anda menutup tab dengan spanduk kolaborasi di bagian atas, itu mengakhiri alur kerja model berbagi dan Anda kehilangan kemajuan. Dalam hal ini, Anda harus memulai ulang alur kerja model berbagi dari bagian Shared With Me di halaman Model Bersama. Untuk informasi selengkapnya, lihat Berkolaborasi dengan ilmuwan data.
- Saat memperbarui model di Studio Classic, Anda tidak dapat mengubah kolom target jika ingin membagikan pembaruan model kembali ke Canvas. Jika Anda ingin mengubah kolom target dan melatih kembali model, latih model dan kemudian gunakan tombol Bagikan untuk berbagi ke

Canvas. Untuk informasi selengkapnya tentang berbagi model baru ke Canvas, lihat Membawa model Anda sendiri ke SageMaker Canvas.

- Saat memperbarui model di antarmuka Amazon SageMaker Data Wrangler Recipe di Studio Classic, ada batasan perubahan yang dapat diterapkan pengguna Studio Classic yang didukung Canvas:
  - Anda hanya dapat membagikan model ke Canvas yang telah dilatih dari node terakhir dalam aliran data linier Data Wrangler.
  - Hanya node transformasi yang didukung.
  - Anda tidak dapat melakukan operasi pada kolom Target.
  - Anda tidak dapat memperbarui tipe data kolom.
  - Anda tidak dapat memperbarui sumber data atau menambahkan sumber data baru.
- Saat membagikan kandidat alternatif ke Canvas dari halaman Studio Classic Autopilot, Anda tidak dapat memilih model dari papan peringkat. Anda harus memilih model bersama dari spanduk dan kemudian memilih alternatif dari daftar. Untuk informasi selengkapnya, lihat Berbagi model alternatif dengan pengguna Canvas di dokumentasi Canvas.
- Hanya model yang kompatibel dengan SageMaker Neo yang dapat dibagikan kembali ke Canvas dengan sukses. Model yang kompatibel adalah model Autopilot yang menggunakan algoritma XGBoost atau MLP. Model yang tidak kompatibel termasuk model Autopilot yang menggunakan algoritma pembelajar linier.
- Untuk transformasi formula khusus menggunakan Spark SQL, Canvas hanya mendukung operasi Unary, fungsi Agregat, operasi penggabungan String, dan operasi Power. Operasi lain tidak didukung.

Batasan untuk membawa model Anda sendiri (BYOM)

Keterbatasan umum berikut berlaku ketika Anda ingin membawa model Anda sendiri ke SageMaker Canvas.

- Saat model dibagikan dari Studio Classic ke Canvas, pengguna Canvas tidak dapat memperbarui atau melihat detail pada kumpulan data yang digunakan untuk membuat model.
- Saat pengguna Canvas ingin menjalankan prediksi tunggal pada model yang diimpor, tidak ada batasan tipe data saat memperbarui nilai kolom. Anda harus memastikan secara manual bahwa saat memperbarui nilai untuk prediksi tunggal, Anda mencocokkan tipe data dari nilai yang ada.
- Ketika pengguna Canvas ingin menjalankan prediksi batch pada model yang diimpor, Canvas mengasumsikan bahwa Anda (pengguna Canvas) tahu seperti apa dataset input yang diharapkan.

Anda harus memiliki kumpulan data dengan kolom dan tipe data yang cocok dengan kumpulan data yang digunakan untuk melatih model. Jika tidak, konsultasikan dengan pengguna yang berbagi model dengan Anda dan impor kumpulan data yang dapat Anda gunakan untuk menjalankan prediksi batch.

- Aplikasi Canvas secara internal menggunakan titik akhir tanpa server untuk menjalankan prediksi dan menghasilkan metrik model. Model yang dibagikan ke Canvas harus kompatibel dengan titik akhir tanpa server:
  - Ukuran memori maksimum adalah 6144 MB.
  - Saat mengonfigurasi kunci respons input inferensi dalam wadah Anda, gunakan konfigurasi berikut:

```
INFERENCE_INPUT_RESPONSE_KEYS = {
 "BINARY": ["predicted_label", "probability"],
 "MULTI_CLASS": ["predicted_label", "probability", "probabilities", "labels"],
}
```

  - Anda dapat memilih wadah inferensi SageMaker yang disediakan atau membawa wadah inferensi gambar Anda sendiri untuk digunakan untuk titik akhir. SageMaker menyediakan wadah untuk algoritme bawaan dan gambar Docker bawaan untuk beberapa kerangka kerja pembelajaran mesin yang paling umum. Jika Anda membawa wadah Anda sendiri, Anda harus memodifikasinya agar dapat digunakan SageMaker. Untuk informasi selengkapnya tentang membawa wadah Anda sendiri, lihat Mengadaptasi Wadah Inferensi Anda Sendiri.
  - Pengecualian Fitur untuk titik akhir tanpa server juga berlaku.
- Agar berhasil membagikan model dari Studio Classic ke Canvas, Canvas menerima output inferensi model dalam format di bawah ini:

  TEKS/CSV

  - Regresi: Respons inferensi model harus berupa string byte di mana setiap prediksi output dipisahkan oleh: \n

```
b'-0.0007884334772825241\n-0.015136942267417908\n0.05006366223096847 5\n0.02891816757619381\
```

  - Klasifikasi: Respon inferensi model harus berupa string byte di mana masing-masing`predicted_label`,`predicted_probability`,`probabilities`, dan `labels` dipisahkan oleh\n. Contoh berikut adalah untuk klasifikasi biner:

```
b'no,0.9967488050460815,"[0.9967488050460815, 0.003251201706007123]","[\'no
\', \'yes\']"\nno,0.9999420642852783,"[0.9999420642852783,
 5.793538366560824e-05]","[\'no\', \'yes
\']"\nno,0.9999846816062927,"[0.9999846816062927, 1.5326571883633733e-05]","[\'no
\', \'yes\']"\nno,0.9999727606773376,"[0.9999727606773376,
 2.7267418772680685e-05]","[\'no\', \'yes\']"\n'
```

Contoh berikut adalah untuk klasifikasi multi-kelas:

```
b'Iris-setosa,1.0,"[1.0, 0.0, 0.0]","[\'Iris-setosa\', \'Iris-versicolor\',
 \'Iris-virginica\']"\nIris-setosa,1.0,"[1.0, 0.0, 0.0]","[\'Iris-setosa\', \'Iris-
versicolor\', \'Iris-virginica\']"\nIris-setosa,1.0,"[1.0, 0.0, 0.0]","[\'Iris-
setosa\', \'Iris-versicolor\', \'Iris-virginica\']"\nIris-setosa,1.0,"[1.0, 0.0,
 0.0]","[\'Iris-setosa\', \'Iris-versicolor\', \'Iris-virginica\']"\n'
```

APLIKASI/JSON

- Regresi: Respons inferensi model harus berupa string JSON yang berisi `prediction` kunci, dan nilainya harus berupa daftar prediksi keluaran:

```
let response = {
    "predictions": [
        // First instance prediction.
        1.75
        // Second instance prediction.
        3.25
    ]
}
```

- Klasifikasi: Respon inferensi model harus berupa string JSON yang berisi `probabilities` kunci, dan nilainya harus menjadi daftar probabilitas.

  Contoh berikut adalah untuk klasifikasi biner:

```
let response = {
    "probabilities": [
        // First instance prediction.
        [0.9, 0.1]
        // Second instance prediction.
        [0.2, 0.8]
    ]
```

```
}
```

Contoh berikut adalah untuk klasifikasi multi-kelas:

```
let response = {
    "probabilities": [
        // First instance prediction.
        [0.7, 0.2, 0.1]
        // Second instance prediction.
        [0.2, 0.5, 0.3]
    ]
}
```

Ada juga batasan yang berlaku tergantung pada jenis model yang ingin Anda bawa:

Bawa model Anda sendiri dari SageMaker JumpStart

Tinjau informasi dan batasan berikut saat berbagi SageMaker JumpStart model dengan Canvas.

- Berikut ini adalah algoritma yang didukung yang dapat Anda impor model ke Canvas. Untuk detail selengkapnya, lihat Dokumentasi SageMaker JumpStart .
  - Klasifikasi tabel: LightGBM,, XGBoost, -Tabular CatBoost,, Linear AutoGluon Learner TabTransformer
  - Regresi tabel: LightGBM,, XGBoost, -Tabular, CatBoost, Linear Learner AutoGluon TabTransformer
- Di SageMaker JumpStart, tombol Bagikan hanya dihidupkan jika model siap dibagikan ke Canvas. Jika model terlatih Anda tidak memiliki tombol Bagikan ke SageMaker Kanvas, model Anda tidak didukung untuk BYOM.
- Anda harus memberikan kumpulan data pelatihan dan validasi saat melatih model. SageMaker JumpStart Kumpulan data harus disimpan di Amazon S3, dan peran eksekusi pengguna Studio Classic dan Canvas Anda harus memiliki akses ke lokasi Amazon S3. Anda dapat menggunakan URI Amazon S3 yang sama untuk berbagi kumpulan data pelatihan dan validasi dengan Canvas, atau Anda dapat berbagi kumpulan data yang berbeda dengan skema data yang sama.

  File data pelatihan atau validasi Anda akan terlihat seperti berikut (dalam format CSV). Anda harus mengindeks file Anda dengan kolom pertama sebagai target.

```
3 1 22 1 1 0 4 4
```

```
0 0 38 0 0 1 3 4
1 0 67 0 1 0 1 6
1 0 67 0 0 2 2 6
0 0 40 0 0 2 6 6
2 0 56 1 0 1 2 6
```

- Secara default, SageMaker JumpStart gunakan kolom pertama dari kumpulan data pelatihan dan validasi sebagai target saat melatih model. Kolom target (atau secara default, kolom pertama) dari kumpulan data dibagikan ke Canvas.
- Anda harus memberikan header kolom dari kumpulan data pelatihan dan validasi saat melatih model. SageMaker JumpStart Secara default, SageMaker JumpStart hanya menerima kumpulan data tanpa header kolom, jadi Anda harus menambahkan header kolom sebagai file saat melatih model Anda. URI Amazon S3 untuk file header kolom juga dibagikan ke Canvas. File header kolom Anda akan terlihat seperti contoh berikut (dalam format CSV). Kolom pertama harus menjadi target.

  ```
  Segmentation EverMarried Age Graduated WorkExperience SpendingScore FamilySize Var1
  ```

- Pekerjaan pelatihan SageMaker JumpStart harus `Complete` sebelum Anda dapat berbagi dengan Canvas.
- Untuk masalah klasifikasi (atau prediksi kategoris di Canvas), nama kelas asli perlu disediakan di bagian Configure model output saat berbagi ke Canvas. Urutan nama kelas harus cocok dengan pengindeksan yang digunakan dalam model. File relasi pemetaan Anda akan terlihat seperti contoh berikut dalam format CSV, di mana indeks 0 (indeks pertama) dipetakan ke nama kelas: `A`

  ```
  A B C D
  ```

  Ketika pengguna Canvas melihat metrik model dalam aplikasi Canvas, mereka hanya dapat melihat indeks setiap kelas (0, 1, 2). Namun, pengguna dapat melihat nama kelas saat melihat hasil untuk prediksi tunggal.

Bawa model Anda sendiri dari Autopilot

Tinjau informasi dan batasan berikut saat berbagi model dari Autopilot ke Canvas.

- Anda hanya dapat membagikan model ke Canvas yang telah berhasil Anda latih dari pekerjaan AutoML dengan mode Ensembling, HPO, atau Auto (untuk mode Otomatis, Autopilot memilih mode Ensembling atau HPO berdasarkan ukuran kumpulan data pelatihan). Jenis masalah Autopilot yang saat ini didukung adalah Regresi, Klasifikasi multi-kelas, klasifikasi biner.

- Untuk setiap pekerjaan Autopilot, Anda dapat memilih model apa saja (model Terbaik atau kandidat lainnya) untuk dibagikan ke Canvas satu per satu. Anda hanya perlu memilih tombol Bagikan model dan kemudian menentukan pengguna Canvas dengan siapa Anda ingin berbagi model dan catatan.
- AutoGluon-Model tabular yang menggunakan transformator Data Wrangler untuk inferensi tidak dapat dibagikan ke Canvas. Ini karena transformator Data Wrangler menyebabkan model menggunakan lebih dari satu wadah.
- Model HPO yang tidak kompatibel dengan SageMaker Neo tidak dapat dibagikan ke Canvas dengan sukses. Model yang kompatibel adalah model Autopilot yang menggunakan algoritma XGBoost atau MLP. Model yang tidak kompatibel termasuk model Autopilot yang menggunakan algoritma pembelajar linier.

Bawa model Anda sendiri dari Model Registry

Tinjau informasi dan batasan berikut saat berbagi model dari Model Registry ke Canvas.

- Tidak seperti tombol Bagikan yang disediakan oleh SageMaker JumpStart, Model Registry tidak menyediakan validasi model, jadi ada kemungkinan model terdaftar yang berhasil dibagikan dari Studio Classic dapat gagal saat mengimpor ke Canvas karena ketidakcocokan model. Tinjau tips berikut sebelum berbagi ke Canvas dari Model Registry:
  - Gunakan wadah inferensi tunggal untuk model Anda. Anda dapat mendaftarkan model dengan beberapa kontainer di dalam AdditionalInferenceSpecificationsbidang, tetapi Canvas hanya dioptimalkan untuk satu wadah inferensi per model. Misalnya, saat Anda menggunakan pipeline inferensi dan mendaftarkan beberapa kontainer di `AdditionalInferenceSpecifications` lapangan dengan beberapa kontainer pra-pemrosesan data dan wadah inferensi, secara default wadah pertama dipilih untuk inferensi model di Canvas. Evaluasi apakah ini berfungsi untuk kasus penggunaan Anda jika Anda menggunakan pipeline pembelajaran mesin.
  - Gunakan algoritma tabular SageMaker bawaan dengan format inferensi yang kompatibel. Algoritma sampel yang diuji dengan output inferensi yang kompatibel adalah Autogluon-Tabular,, LightGBM, dan XGBoost. CatBoost TabTransformer Algoritma seperti Factorization Machines tidak menerima CSV sebagai input file, dan format output inferensi untuk algoritma seperti Linear Learner dan K-NN tidak didukung oleh Canvas.
  - Anda juga dapat membawa wadah gambar Anda sendiri dan membagikannya ke Canvas, atau memodifikasi SageMaker wadah yang sudah dibuat sebelumnya.

    - Jika Anda membawa wadah Anda sendiri, Anda harus memodifikasinya agar berfungsi SageMaker. Untuk informasi selengkapnya tentang membawa wadah Anda sendiri, lihat Mengadaptasi Wadah Inferensi Anda Sendiri.
    - Untuk pemformatan terperinci untuk format keluaran inferensi Anda, lihat. Batasan untuk membawa model Anda sendiri (BYOM)
- Saat mendaftarkan model Anda dalam grup paket model, ingatlah untuk memberikan atribut berikut dengan wadah inferensi Anda:
    - Lingkungan:

```
"{\"SAGEMAKER_CONTAINER_LOG_LEVEL\": \"20\", \"SAGEMAKER_PROGRAM\": \"inference.py
\", \"SAGEMAKER_REGION\": \"us-west-2\", \"SAGEMAKER_SUBMIT_DIRECTORY\": \"/opt/ml/
model/code\"}"
```

    - Gambar:

```
"s3://sagemaker-us-west-2-<account-id>/model-regression-
abalone-2022-10-14-23-02-45/model.tar.gz"
```

    - ModelDataUrl

```
"<account-id>.dkr.ecr.us-west-2.amazonaws.com/sagemaker-xgboost:1.3-1"
```

- Anda harus memberikan kumpulan data pelatihan dan validasi saat berbagi model dari Model Registry ke Canvas. Kumpulan data harus disimpan di Amazon S3, dan peran eksekusi pengguna Studio Classic dan Canvas harus memiliki akses ke lokasi Amazon S3. Anda dapat menggunakan URI Amazon S3 yang sama untuk berbagi kumpulan data pelatihan dan validasi dengan Canvas, atau Anda dapat berbagi kumpulan data yang berbeda dengan skema data yang sama. Kumpulan data harus memiliki format input yang tepat yang memberi makan wadah inferensi model Anda.
- Anda harus memberikan kolom target ke Canvas, atau kolom pertama dari kumpulan data pelatihan/validasi Anda digunakan secara default.
- Di bagian Tambahkan detail model saat berbagi ke Canvas, Anda dapat memberikan baris pertama kumpulan data pelatihan dan validasi sebagai header, atau Anda dapat menentukan header sebagai file yang berbeda.
- Untuk masalah klasifikasi (atau prediksi kategoris di Canvas), nama kelas asli perlu disediakan saat berbagi ke SageMaker Canvas melalui opsi Konfigurasi keluaran model. Urutan nama kelas harus cocok dengan pengindeksan yang digunakan dengan model bersama. Pemetaan dapat berupa file CSV di Amazon S3, atau Anda dapat memasukkan nama kelas secara manual.

Kelola penagihan dan biaya di Canvas SageMaker

Untuk melacak biaya yang terkait dengan aplikasi SageMaker Canvas Anda, Anda dapat menggunakan AWS Billing and Cost Management layanan ini. Billing and Cost Management menyediakan alat untuk membantu Anda mengumpulkan informasi terkait biaya dan penggunaan Anda, menganalisis driver biaya dan tren penggunaan Anda, dan mengambil tindakan untuk menganggarkan pengeluaran Anda. Lihat informasi yang lebih lengkap di Apakah AWS Billing and Cost Management?

Penagihan di SageMaker Canvas terdiri dari komponen-komponen berikut:

- Biaya instans ruang kerja - Anda dikenakan biaya untuk jumlah jam masuk atau menggunakan SageMaker Canvas. Kami menyarankan Anda keluar atau membuat jadwal untuk mematikan aplikasi Canvas apa pun yang tidak Anda gunakan secara aktif untuk mengurangi biaya. Untuk informasi selengkapnya, lihat Keluar dari Amazon SageMaker Canvas.
- AWSbiaya layanan — Anda dikenakan biaya untuk membangun dan membuat prediksi dengan model khusus, atau untuk membuat prediksi dengan model Ready-to-use :
  - Biaya pelatihan — Anda dikenakan biaya untuk sumber daya yang digunakan untuk membangun model khusus.
  - Biaya prediksi — Anda dikenakan biaya untuk sumber daya yang digunakan untuk menghasilkan prediksi, tergantung pada jenis model kustom yang Anda buat atau jenis eady-to-use model R yang Anda gunakan.

eady-to-use Model R di Canvas memanfaatkan AWS layanan lain untuk menghasilkan prediksi. Bila Anda menggunakan eady-to-use model R, Anda dikenakan biaya oleh layanan masing-masing, dan ketentuan harga mereka berlaku:

- Untuk analisis sentimen, ekstraksi entitas, deteksi bahasa, dan deteksi informasi pribadi, Anda dikenakan biaya dengan harga Amazon Comprehend.
- Untuk deteksi objek dalam gambar dan deteksi teks dalam gambar, Anda dikenakan biaya dengan harga Amazon Rekognition.
- Untuk analisis pengeluaran, analisis dokumen identitas, dan analisis dokumen, Anda dikenakan biaya dengan harga Amazon Texttract.

Untuk informasi lebih lanjut, lihat harga SageMaker Canvas.

Untuk membantu melacak biaya di Billing and Cost Management, Anda dapat menetapkan tag kustom ke aplikasi dan pengguna Canvas SageMaker Anda. Anda dapat melacak biaya yang dikeluarkan aplikasi Anda, dan dengan menandai profil pengguna individu, Anda dapat melacak biaya berdasarkan profil pengguna. Untuk informasi selengkapnya tentang tag, lihat Menggunakan Tag Alokasi Biaya.

Anda dapat menambahkan tag ke aplikasi SageMaker Canvas dan pengguna dengan melakukan hal berikut:

- Jika Anda menyiapkan SageMaker Domain dan SageMaker Kanvas Amazon untuk pertama kalinya, ikuti petunjuk Memulai dan menambahkan tag saat membuat Domain atau pengguna Anda. Anda dapat menambahkan tag baik melalui pengaturan Umum di pengaturan konsol Domain, atau melalui API (CreateDomainatau CreateUserProfile). SageMaker menambahkan tag yang ditentukan dalam Domain Anda atau UserProfile ke aplikasi SageMaker Canvas atau pengguna yang Anda buat setelah Anda membuat Domain.
- Jika Anda ingin menambahkan tag ke aplikasi di Domain yang ada, Anda harus menambahkan tag ke Domain atau UserProfile. Anda dapat menambahkan tag melalui konsol atau AddTagsAPI. Jika Anda menambahkan tag melalui konsol, Anda harus menghapus dan meluncurkan kembali aplikasi SageMaker Canvas Anda agar tag dapat menyebar ke aplikasi. Jika Anda menggunakan API, tag akan ditambahkan langsung ke aplikasi. Untuk informasi selengkapnya tentang menghapus dan meluncurkan kembali aplikasi SageMaker Canvas, lihat Mengelola aplikasi.

Setelah Anda menambahkan tag ke Domain Anda, mungkin diperlukan waktu hingga 24 jam agar tag muncul di AWS Billing and Cost Management konsol untuk aktivasi. Setelah mereka muncul di konsol, dibutuhkan 24 jam lagi agar tag diaktifkan.

Pada halaman Penjelajah biaya, Anda dapat mengelompokkan dan memfilter biaya berdasarkan tag dan jenis penggunaan untuk memisahkan biaya instans Ruang Kerja (Session-Hrs) dari biaya Pelatihan Anda. Nama-nama jenis penggunaan adalah sebagai berikut:

- Biaya instans ruang kerja (Session-Hrs): `REGION-Canvas:Session-Hrs (Hrs)`
- Biaya pelatihan:
  - `REGION-Canvas:CreateModelRequest-Tier0 (CreateModelRequest)`
  - `REGION-Canvas:MillionCells-Tier1 (MillionCells)`

Kemampuan SageMaker geospasial Amazon

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Jika sebelum 30 November 2023 Anda membuat SageMaker Domain Amazon, Studio Classic tetap menjadi pengalaman default. Domain yang dibuat setelah 30 November 2023 default ke pengalaman Studio baru. Fitur dan sumber daya SageMaker geospasial Amazon hanya tersedia di Studio Classic. Untuk mempelajari lebih lanjut tentang menyiapkan domain dan memulai dengan Studio, lihatMemulai dengan SageMaker geospasial Amazon .

Kemampuan SageMaker geospasial Amazon memudahkan ilmuwan data dan insinyur pembelajaran mesin (ML) untuk membangun, melatih, dan menerapkan model ML lebih cepat menggunakan data geospasial. Anda memiliki akses ke alat data, pemrosesan, dan visualisasi sumber terbuka dan pihak ketiga untuk membuatnya lebih efisien dalam menyiapkan data geospasial untuk ML. Anda dapat meningkatkan produktivitas Anda dengan menggunakan algoritme yang dibuat khusus dan model ML yang telah dilatih sebelumnya untuk mempercepat pembuatan dan pelatihan model, serta menggunakan alat visualisasi bawaan untuk mengeksplorasi output prediksi pada peta interaktif dan kemudian berkolaborasi di seluruh tim dalam wawasan dan hasil.

> **Note**
>
> Saat ini, kemampuan SageMaker geospasial hanya didukung di Wilayah Barat AS (Oregon). Jika Anda tidak melihat UI SageMaker geospasial yang tersedia di instans Studio Classic saat ini, periksa untuk memastikan Anda saat ini berada di Wilayah AS Barat (Oregon).

Mengapa menggunakan kemampuan SageMaker geospasial?

Anda dapat menggunakan kemampuan SageMaker geospasial untuk membuat prediksi pada data geospasial lebih cepat daripada solusi. do-it-yourself SageMaker Kemampuan geospasial memudahkan untuk mengakses data geospasial dari danau data pelanggan yang ada, kumpulan data sumber terbuka, dan penyedia data geospasial lainnya. SageMaker SageMaker Kemampuan geospasial meminimalkan kebutuhan untuk membangun infrastruktur kustom dan fungsi pra-pemrosesan data dengan menawarkan algoritme yang dibuat khusus untuk persiapan data yang

efisien, pelatihan model, dan inferensi. Anda juga dapat membuat dan berbagi visualisasi dan data khusus dengan perusahaan Anda dari Amazon SageMaker Studio Classic. SageMaker Kemampuan geospasial menawarkan model pra-terlatih untuk penggunaan umum di bidang pertanian, real estat, asuransi, dan layanan keuangan.

Bagaimana saya bisa menggunakan kemampuan SageMaker geospasial?

Anda dapat menggunakan kemampuan SageMaker geospasial dengan dua cara.

- Melalui UI SageMaker geospasial, sebagai bagian dari Amazon SageMaker Studio Classic UI.
- Melalui contoh notebook Studio Classic yang menggunakan gambar Geospatial 1.0.

SageMaker memiliki kemampuan geospasial sebagai berikut

- Gunakan gambar SageMaker geospasial yang dibuat khusus yang mendukung instance notebook berbasis CPU dan GPU, dan juga menyertakan pustaka sumber terbuka yang umum digunakan yang ditemukan dalam alur kerja pembelajaran mesin geospasial.
- Gunakan Amazon SageMaker Processing dan wadah SageMaker geospasial untuk menjalankan beban kerja skala besar dengan kumpulan data Anda sendiri, termasuk tanah, cuaca, iklim, LiDAR, dan citra udara dan satelit komersial.
- Jalankan pekerjaan Earth Observation untuk pemrosesan data raster.
- Jalankan pekerjaan Pengayaan Vektor untuk mengubah garis lintang dan bujur menjadi alamat yang dapat dibaca manusia, dan cocokkan jejak GPS yang bising ke jalan tertentu.
- Gunakan alat visualisasi bawaan langsung di Studio Classic untuk melihat data geospasial atau prediksi model secara interaktif di peta.

Anda juga dapat menggunakan data dari kumpulan penyedia data geospasial. Saat ini, pengumpulan data yang tersedia meliputi:

- USGS Landsat
- Sentinel-1
- Sentinel-2
- Copernicus DEM
- National Agriculture Imagery Program

Apakah Anda pengguna SageMaker geospasial pertama kali?

Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Domain baru yang dibuat setelah 30 November 2023 default ke pengalaman Studio. Akses ke SageMaker geospasial terbatas pada Studio Classic, untuk mempelajari lebih lanjut lihatMengakses geospasial SageMaker .

Jika Anda pengguna pertama kali AWS atau Amazon SageMaker, kami sarankan Anda melakukan hal berikut:

1. Buat sebuahAkun AWS.

   Untuk mempelajari cara menyiapkan AWS akun dan memulai SageMaker, lihatMengatur SageMaker Prasyarat Amazon.
2. Buat peran pengguna dan peran eksekusi yang bekerja dengan SageMaker geospasial.

   Sebagai layanan terkelola, kemampuan SageMaker geospasial Amazon melakukan operasi atas nama Anda pada AWS perangkat keras yang SageMaker mengelola. Peran SageMaker eksekusi dan hanya melakukan operasi yang diberikan pengguna. Untuk bekerja dengan kemampuan SageMaker geospasial, Anda harus mengatur peran pengguna dan peran eksekusi. Untuk informasi selengkapnya, lihat SageMaker peran kemampuan geospasial.
3. Perbarui kebijakan kepercayaan Anda untuk memasukkan SageMaker geospasial.

   SageMaker geospasial mendefinisikan prinsip layanan tambahan. Untuk mempelajari cara membuat atau memperbarui kebijakan kepercayaan peran SageMaker eksekusi Anda, lihatMenambahkan prinsip layanan SageMaker geospasial ke peran SageMaker eksekusi yang ada.
4. Siapkan SageMaker Domain Amazon untuk mengakses Amazon SageMaker Studio Classic.

   Untuk menggunakan SageMaker geospasial, Domain diperlukan. Untuk Domain yang dibuat sebelum 30 November 2023, pengalaman defaultnya adalah Studio Classic. Domain yang dibuat setelah 30 November 2023 default ke pengalaman Studio. Untuk mempelajari selengkapnya tentang mengakses Studio Classic dari Studio, lihatMengakses geospasial SageMaker .
5. Ingat, matikan sumber daya.

   Setelah Anda selesai menggunakan kemampuan SageMaker geospasial, matikan instance yang dijalankannya untuk menghindari biaya tambahan. Untuk informasi selengkapnya, lihat Matikan Sumber Daya.

Topik

Memulai dengan SageMaker geospasial Amazon

SageMaker geospasial menyediakan tipe Image dan Instance yang dibuat khusus untuk notebook Amazon SageMaker Studio Classic. Anda dapat menggunakan notebook berkemampuan CPU atau GPU dengan Gambar SageMaker geospasial. Anda juga dapat memvisualisasikan data geospasial Anda menggunakan visualizer yang dibuat khusus. Selain itu, SageMaker geospasial juga menyediakan API yang memungkinkan Anda untuk meminta pengumpulan data raster. Anda juga dapat menggunakan model pra-terlatih untuk menganalisis data geospasial, membalikkan geocoding, dan pencocokan peta.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Jika sebelum 30 November 2023 Anda membuat SageMaker Domain Amazon, Studio Classic tetap menjadi pengalaman default. Domain yang dibuat setelah 30 November 2023 default ke pengalaman Studio baru.

Untuk mengakses dan mulai menggunakan SageMaker geospasial Amazon, lakukan hal berikut:

Topik

- Akses pengumpulan data raster Sentinel-2 dan buat pekerjaan observasi bumi untuk melakukan segmentasi lahan

Mengakses geospasial SageMaker

> **Note**
>
> Saat ini, kemampuan SageMaker geospasial hanya didukung di Wilayah AS Barat (Oregon) dan di Studio Classic.
> Jika Anda tidak melihat UI SageMaker geospasial yang tersedia di instans Studio Classic saat ini, periksa untuk memastikan Anda saat ini berada di Wilayah AS Barat (Oregon).

Domain diperlukan untuk mengakses SageMaker geospasial. Jika Anda membuat Domain sebelum 30 November 2023, pengalaman defaultnya adalah Studio Classic.

Jika Anda membuat Domain setelah 30 November 2023 atau jika Anda telah bermigrasi ke Studio, Anda dapat menggunakan prosedur berikut untuk mengaktifkan Studio Classic dari dalam Studio untuk menggunakan fitur SageMaker geospasial.

Untuk mempelajari lebih lanjut tentang membuat Domain, lihat Onboard to Amazon SageMaker Domain.

Untuk mengakses Studio Classic dari Studio

1. Luncurkan Amazon SageMaker Studio.
2. Di bawah Aplikasi, pilih Studio Classic.
3. Kemudian, pilih Create Studio Classic space.
4. Pada halaman ruang Create Studio Classic, masukkan Nama.
5. Nonaktifkan opsi Bagikan dengan domain saya. SageMaker geospasial tidak tersedia di domain bersama.
6. Kemudian pilih Buat ruang.

Saat berhasil, Status berubah menjadi Pembaruan. Ketika aplikasi Studio Classic Anda siap digunakan, status berubah menjadi Berhenti.

Untuk memulai aplikasi Studio Classic, pilih Run.

Membuat notebook Amazon SageMaker Studio Classic menggunakan gambar geospasial

Important

Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

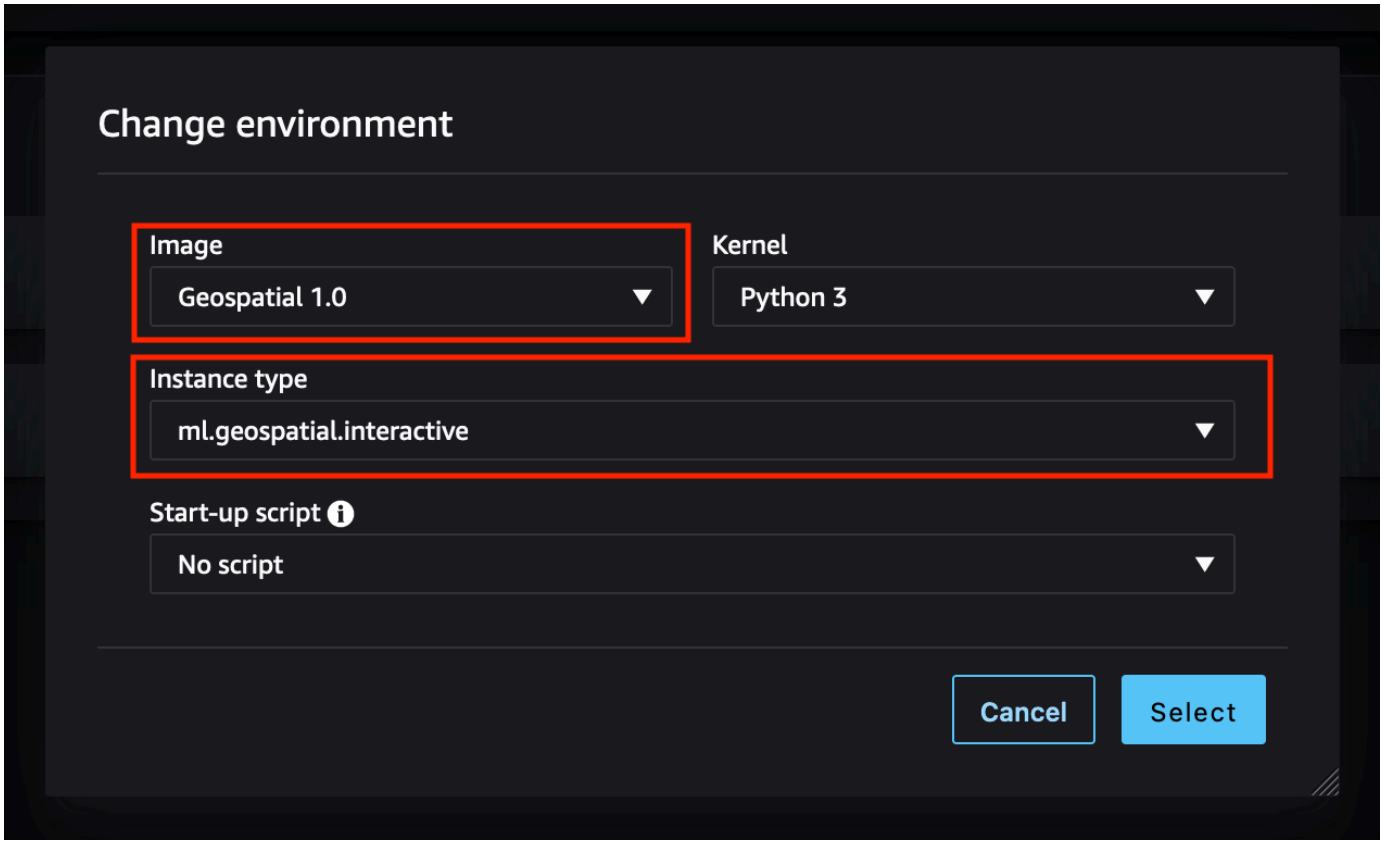
Note

Saat ini, SageMaker geospasial hanya didukung di Wilayah Barat AS (Oregon).
Jika Anda tidak melihat SageMaker geospasial tersedia di instance domain atau notebook Anda saat ini, pastikan Anda saat ini berada di Wilayah AS Barat (Oregon).

Gunakan prosedur berikut untuk membuat notebook Studio Classic dengan gambar SageMaker geospasial. Jika pengalaman studio default Anda adalah Studio, lihat Mengakses geospasial SageMaker untuk mempelajari cara memulai aplikasi Studio Classic.

Untuk membuat notebook Studio Classic dengan gambar SageMaker geospasial

1. Luncurkan Studio Klasik
2. Pilih Beranda di bilah menu.
3. Di bawah Tindakan cepat, pilih Buka Peluncur.
4. Saat kotak dialog Launcher terbuka. Pilih Ubah lingkungan di bawah Notebook dan hitung sumber daya.
5. Kapan, kotak dialog Ubah lingkungan terbuka. Pilih dropdown Gambar dan pilih atau ketik Geospatial 1.0.



6. Selanjutnya, pilih jenis Instance dari dropdown.

   SageMaker geospasial mendukung dua jenis instance notebook: CPU dan GPU. Instance CPU yang didukung disebut ml.geospatial.interactive. Setiap contoh GPU keluarga G5 dapat digunakan dengan gambar Geospasial 1.0.

   > **Note**
   >
   > Jika Anda menerima ResourceLimitExceededkesalahan saat mencoba memulai instance berbasis GPU, Anda perlu meminta peningkatan kuota. Untuk memulai permintaan peningkatan kuota Service Quotas, lihat Meminta peningkatan kuota pada Panduan Pengguna Service Quotas

7. Pilih Pilih.
8. Pilih Buat Notebook.

Setelah membuat buku catatan, untuk mempelajari lebih lanjut tentang SageMaker geospasial, coba tutorial SageMaker geospasial. Ini menunjukkan kepada Anda bagaimana memproses data gambar

Sentinel-2 dan melakukan segmentasi lahan di atasnya menggunakan API pekerjaan pengamatan bumi.

Akses pengumpulan data raster Sentinel-2 dan buat pekerjaan observasi bumi untuk melakukan segmentasi lahan

Tutorial berbasis Python ini menggunakan SDK for Python (Boto3) dan notebook Amazon Studio Classic. SageMaker Untuk menyelesaikan demo ini dengan sukses, pastikan Anda memiliki izin AWS Identity and Access Management (IAM) yang diperlukan untuk menggunakan SageMaker geospasial dan Studio Classic. SageMaker geospasial mengharuskan Anda memiliki pengguna, grup, atau peran yang dapat mengakses Studio Classic. Anda juga harus memiliki peran SageMaker eksekusi yang menentukan prinsip layanan SageMaker geospasial, `sagemaker-geospatial.amazonaws.com` dalam kebijakan kepercayaannya.

Untuk mempelajari lebih lanjut tentang persyaratan ini, lihat peran IAM SageMaker geospasial.

Tutorial ini menunjukkan cara menggunakan API SageMaker geospasial untuk menyelesaikan tugas-tugas berikut:

- Temukan koleksi data raster yang tersedia dengan`list_raster_data_collections`.
- Cari pengumpulan data raster tertentu dengan menggunakan`search_raster_data_collection`.
- Buat pekerjaan pengamatan bumi (EOJ) dengan menggunakan`start_earth_observation_job`.

Menggunakan **`list_raster_data_collections`** untuk menemukan koleksi data yang tersedia

SageMaker geospasial mendukung beberapa pengumpulan data raster. Untuk mempelajari lebih lanjut tentang pengumpulan data yang tersedia, lihatPengumpulan data.

Demo ini menggunakan data satelit yang dikumpulkan dari satelit GeoTIFF yang Sentinel-2 Dioptimalkan di Cloud. Satelit-satelit ini menyediakan cakupan global permukaan bumi setiap lima hari. Selain mengumpulkan gambar permukaan Bumi, satelit Sentinel-2 juga mengumpulkan data di berbagai spektrum.

Untuk mencari area minat (AOI), Anda memerlukan ARN yang terkait dengan data satelit Sentinel-2. Untuk menemukan koleksi data yang tersedia dan ARN terkait di AndaWilayah AWS, gunakan operasi `list_raster_data_collections` API.

Karena respons dapat diberi paginasi, Anda harus menggunakan `get_paginator` operasi untuk mengembalikan semua data yang relevan:

```
import boto3
import sagemaker
import sagemaker_geospatial_map
import json

## SageMaker Geospatial  is currently only avaialable in US-WEST-2
session = boto3.Session(region_name='us-west-2')
execution_role = sagemaker.get_execution_role()

## Creates a SageMaker Geospatial client instance
geospatial_client = session.client(service_name="sagemaker-geospatial")

# Creates a resusable Paginator for the list_raster_data_collections API operation
paginator = geospatial_client.get_paginator("list_raster_data_collections")

# Create a PageIterator from the paginator class
page_iterator = paginator.paginate()

# Use the iterator to iterate throught the results of list_raster_data_collections
results = []
for page in page_iterator:
    results.append(page['RasterDataCollectionSummaries'])

print(results)
```

Ini adalah contoh respons JSON dari operasi `list_raster_data_collections` API. Itu terpotong untuk menyertakan hanya pengumpulan data (Sentinel-2) yang digunakan dalam contoh kodenya. Untuk detail selengkapnya tentang pengumpulan data raster tertentu, gunakan`get_raster_data_collection`:

```
{
    "Arn": "arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/
public/nmqj48dcu3g7ayw8",
    "Description": "Sentinel-2a and Sentinel-2b imagery, processed to Level 2A (Surface
 Reflectance) and converted to Cloud-Optimized GeoTIFFs",
    "DescriptionPageUrl": "https://registry.opendata.aws/sentinel-2-l2a-cogs",
    "Name": "Sentinel 2 L2A COGs",
    "SupportedFilters": [
        {
```

```
            "Maximum": 100,
            "Minimum": 0,
            "Name": "EoCloudCover",
            "Type": "number"
        },
        {
            "Maximum": 90,
            "Minimum": 0,
            "Name": "ViewOffNadir",
            "Type": "number"
        },
        {
            "Name": "Platform",
            "Type": "string"
        }
    ],
    "Tags": {},
    "Type": "PUBLIC"
}
```

Setelah menjalankan contoh kode sebelumnya, Anda mendapatkan ARN dari pengumpulan data raster Sentinel-2,. `arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/public/nmqj48dcu3g7ayw8` Di bagian selanjutnya, Anda dapat menanyakan pengumpulan data Sentinel-2 menggunakan API. `search_raster_data_collection`

Mencari pengumpulan data Sentinel-2 raster menggunakan **`search_raster_data_collection`**

Di bagian sebelumnya, Anda biasa mendapatkan ARN `list_raster_data_collections` untuk pengumpulan data. Sentinel-2 Sekarang Anda dapat menggunakan ARN itu untuk mencari pengumpulan data di area minat tertentu (AOI), rentang waktu, properti, dan pita UV yang tersedia.

Untuk memanggil `search_raster_data_collection` API, Anda harus meneruskan Python dic tionary ke `RasterDataCollectionQuery` parameter. Contoh ini menggunakan`AreaOfInterest`,`TimeRangeFilter`,`PropertyFilters`, dan`BandFilter`. Untuk memudahkan, Anda dapat menentukan kamus Python menggunakan variabel **`search_rdc_query`** untuk menahan parameter permintaan pencarian:

```
search_rdc_query = {
    "AreaOfInterest": {
        "AreaOfInterestGeometry": {
            "PolygonGeometry": {
                "Coordinates": [
```

```
                    [
                        # coordinates are input as longitute followed by latitude
                        [-114.529, 36.142],
                        [-114.373, 36.142],
                        [-114.373, 36.411],
                        [-114.529, 36.411],
                        [-114.529, 36.142],
                    ]
                ]
            }
        }
    },
    "TimeRangeFilter": {
        "StartTime": "2022-01-01T00:00:00Z",
        "EndTime": "2022-07-10T23:59:59Z"
    },
    "PropertyFilters": {
        "Properties": [
            {
                "Property": {
                    "EoCloudCover": {
                        "LowerBound": 0,
                        "UpperBound": 1
                    }
                }
            }
        ],
        "LogicalOperator": "AND"
    },
    "BandFilter": [
        "visual"
    ]
}
```

Dalam contoh ini, Anda menanyakan `AreaOfInterest` yang menyertakan Danau Mead di Utah. Selain itu, Sentinel-2 mendukung beberapa jenis pita gambar. Untuk mengukur perubahan permukaan air, Anda hanya perlu `visual` pita.

Setelah membuat parameter kueri, Anda dapat menggunakan `search_raster_data_collection` API untuk membuat permintaan.

Contoh kode berikut mengimplementasikan permintaan `search_raster_data_collection` API. API ini tidak mendukung pagination menggunakan `get_paginator` API. Untuk memastikan bahwa

respons API lengkap telah dikumpulkan, sampel kode menggunakan `while` loop untuk memeriksa apakah `NextToken` ada. Contoh kode kemudian digunakan `.extend()` untuk menambahkan URL citra satelit dan metadata respons lainnya ke file. `items_list`

Untuk mempelajari selengkapnya`search_raster_data_collection`, lihat SearchRasterDataCollectiondi Referensi Amazon SageMaker API.

```
search_rdc_response = sm_geo_client.search_raster_data_collection(
    Arn='arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/
public/nmqj48dcu3g7ayw8',
    RasterDataCollectionQuery=search_rdc_query
)


## items_list is the response from the API request.
items_list = []

## Use the python .get() method to check that the 'NextToken' exists, if null returns
 None breaking the while loop
while search_rdc_response.get('NextToken'):
    items_list.extend(search_rdc_response['Items'])
    search_rdc_response = sm_geo_client.search_raster_data_collection(
        Arn='arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-
collection/public/nmqj48dcu3g7ayw8',
        RasterDataCollectionQuery=search_rdc_query,
 NextToken=search_rdc_response['NextToken']
    )

## Print the number of observation return based on the query
print (len(items_list))
```

Berikut ini adalah respons JSON dari kueri Anda. Itu telah dipotong untuk kejelasan. Hanya yang **`"BandFilter": ["visual"]`** ditentukan dalam permintaan yang dikembalikan dalam pasangan `Assets` kunci-nilai:

```
{
    'Assets': {
        'visual': {
            'Href': 'https://sentinel-cogs.s3.us-west-2.amazonaws.com/sentinel-s2-l2a-
cogs/15/T/UH/2022/6/S2A_15TUH_20220623_0_L2A/TCI.tif'
        }
    },
```

```
    'DateTime': datetime.datetime(2022, 6, 23, 17, 22, 5, 926000, tzinfo = tzlocal()),
    'Geometry': {
        'Coordinates': [
            [
                [-114.529, 36.142],
                [-114.373, 36.142],
                [-114.373, 36.411],
                [-114.529, 36.411],
                [-114.529, 36.142],
            ]
        ],
        'Type': 'Polygon'
    },
    'Id': 'S2A_15TUH_20220623_0_L2A',
    'Properties': {
        'EoCloudCover': 0.046519,
        'Platform': 'sentinel-2a'
    }
}
```

Sekarang setelah Anda memiliki hasil kueri, di bagian selanjutnya Anda dapat memvisualisasikan hasilnya dengan menggunakan`matplotlib`. Ini untuk memverifikasi bahwa hasil berasal dari wilayah geografis yang benar.

Memvisualisasikan penggunaan Anda **`search_raster_data_collection`**`matplotlib`

Sebelum Anda memulai pekerjaan pengamatan bumi (EOJ), Anda dapat memvisualisasikan hasil dari pertanyaan kami dengan. `matplotlib` Contoh kode berikut mengambil item pertama,`items_list[0]["Assets"]["visual"]["Href"]`, dari `items_list` variabel yang dibuat dalam sampel kode sebelumnya dan mencetak gambar menggunakan`matplotlib`.

```
# Visualize an example image.
import os
from urllib import request
import tifffile
import matplotlib.pyplot as plt

image_dir = "./images/lake_mead"
os.makedirs(image_dir, exist_ok=True)

image_dir = "./images/lake_mead"
os.makedirs(image_dir, exist_ok=True)
```

```
image_url = items_list[0]["Assets"]["visual"]["Href"]
img_id = image_url.split("/")[-2]
path_to_image = image_dir + "/" + img_id + "_TCI.tif"
response = request.urlretrieve(image_url, path_to_image)
print("Downloaded image: " + img_id)

tci = tifffile.imread(path_to_image)
plt.figure(figsize=(6, 6))
plt.imshow(tci)
plt.show()
```

Setelah memeriksa apakah hasilnya berada di wilayah geografis yang benar, Anda dapat memulai Earth Observation Job (EOJ) pada langkah berikutnya. Anda menggunakan EOJ untuk mengidentifikasi badan air dari citra satelit dengan menggunakan proses yang disebut segmentasi tanah.

Memulai pekerjaan pengamatan bumi (EOJ) yang melakukan segmentasi tanah pada serangkaian citra satelit

SageMaker geospasial menyediakan beberapa model pra-terlatih yang dapat Anda gunakan untuk memproses data geospasial dari pengumpulan data raster. Untuk mempelajari lebih lanjut tentang model pra-terlatih yang tersedia dan operasi kustom, lihatJenis Operasi.

Untuk menghitung perubahan luas permukaan air, Anda perlu mengidentifikasi piksel mana dalam gambar yang sesuai dengan air. Segmentasi tutupan lahan adalah model segmentasi semantik yang didukung oleh API. `start_earth_observation_job` Model segmentasi semantik mengaitkan label dengan setiap piksel di setiap gambar. Dalam hasilnya, setiap piksel diberi label yang didasarkan pada peta kelas untuk model. Berikut ini adalah peta kelas untuk model segmentasi lahan:

```
{
    0: "No_data",
    1: "Saturated_or_defective",
    2: "Dark_area_pixels",
    3: "Cloud_shadows",
    4: "Vegetation",
    5: "Not_vegetated",
    6: "Water",
    7: "Unclassified",
    8: "Cloud_medium_probability",
    9: "Cloud_high_probability",
    10: "Thin_cirrus",
```

```
    11: "Snow_ice"
}
```

Untuk memulai pekerjaan pengamatan bumi, gunakan `start_earth_observation_job` API. Ketika Anda mengirimkan permintaan Anda, Anda harus menentukan yang berikut:

- `InputConfig`(dict) - Digunakan untuk menentukan koordinat area yang ingin Anda cari, dan metadata lain yang terkait dengan pencarian Anda.
- `JobConfig`(dict) - Digunakan untuk menentukan jenis operasi EOJ yang Anda lakukan pada data. Contoh ini menggunakan **`LandCoverSegmentationConfig`**.
- `ExecutionRoleArn`(string) — ARN dari peran SageMaker eksekusi dengan izin yang diperlukan untuk menjalankan pekerjaan.
- `Name`(string) —Nama untuk pekerjaan pengamatan bumi.

`InputConfig`Ini adalah Python kamus. Gunakan variabel berikut **`eoj_input_config`** untuk menahan parameter permintaan pencarian. Gunakan variabel ini saat Anda membuat permintaan `start_earth_observation_job` API. w.

```
# Perform land cover segmentation on images returned from the Sentinel-2 dataset.
eoj_input_config = {
    "RasterDataCollectionQuery": {
        "RasterDataCollectionArn": "arn:aws:sagemaker-geospatial:us-
west-2:378778860802:raster-data-collection/public/nmqj48dcu3g7ayw8",
        "AreaOfInterest": {
            "AreaOfInterestGeometry": {
                "PolygonGeometry": {
                    "Coordinates":[
                        [
                            [-114.529, 36.142],
                            [-114.373, 36.142],
                            [-114.373, 36.411],
                            [-114.529, 36.411],
                            [-114.529, 36.142],
                        ]
                    ]
                }
            }
        },
        "TimeRangeFilter": {
            "StartTime": "2021-01-01T00:00:00Z",
```

```
            "EndTime": "2022-07-10T23:59:59Z",
        },
        "PropertyFilters": {
            "Properties": [{"Property": {"EoCloudCover": {"LowerBound": 0,
 "UpperBound": 1}}}],
            "LogicalOperator": "AND",
        },
    }
}
```

`JobConfig`Ini adalah Python kamus yang digunakan untuk menentukan operasi EOJ yang ingin Anda lakukan pada data Anda:

```
eoj_config = {"LandCoverSegmentationConfig": {}}
```

Dengan elemen kamus yang sekarang ditentukan, Anda dapat mengirimkan permintaan `start_earth_observation_job` API Anda menggunakan contoh kode berikut:

```
# Gets the execution role arn associated with current notebook instance
execution_role_arn = sagemaker.get_execution_role()

# Starts an earth observation job
response = sm_geo_client.start_earth_observation_job(
    Name="lake-mead-landcover",
    InputConfig=eoj_input_config,
    JobConfig=eoj_config,
    ExecutionRoleArn=execution_role_arn,
)

print(response)
```

Awal pekerjaan pengamatan bumi mengembalikan ARN bersama dengan metadata lainnya.

Untuk mendapatkan daftar semua pekerjaan pengamatan bumi yang sedang berlangsung dan saat ini, gunakan `list_earth_observation_jobs` API. Untuk memantau status pekerjaan pengamatan bumi tunggal, gunakan `get_earth_observation_job` API. Untuk membuat permintaan ini, gunakan ARN yang dibuat setelah mengirimkan permintaan EOJ Anda. Untuk mempelajari selengkapnya, lihat GetEarthObservationJobdi Referensi Amazon SageMaker API.

Untuk menemukan ARN yang terkait dengan EOJ Anda, gunakan operasi `list_earth_observation_jobs` API. Untuk mempelajari selengkapnya, lihat ListEarthObservationJobsdi Referensi Amazon SageMaker API.

```
# List all jobs in the account
sg_client.list_earth_observation_jobs()["EarthObservationJobSummaries"]
```

Berikut ini adalah contoh respon JSON:

```
{
    'Arn': 'arn:aws:sagemaker-geospatial:us-west-2:111122223333:earth-observation-job/
futg3vuq935t',
    'CreationTime': datetime.datetime(2023, 10, 19, 4, 33, 54, 21481, tzinfo =
 tzlocal()),
    'DurationInSeconds': 3493,
    'Name': 'lake-mead-landcover',
    'OperationType': 'LAND_COVER_SEGMENTATION',
    'Status': 'COMPLETED',
    'Tags': {}
}, {
    'Arn': 'arn:aws:sagemaker-geospatial:us-west-2:111122223333:earth-observation-job/
wu8j9x42zw3d',
    'CreationTime': datetime.datetime(2023, 10, 20, 0, 3, 27, 270920, tzinfo =
 tzlocal()),
    'DurationInSeconds': 1,
    'Name': 'mt-shasta-landcover',
    'OperationType': 'LAND_COVER_SEGMENTATION',
    'Status': 'INITIALIZING',
    'Tags': {}
}
```

Setelah status pekerjaan EOJ Anda berubah menjadi`COMPLETED`, lanjutkan ke bagian berikutnya untuk menghitung perubahan luas Mead's permukaan Danau.

Menghitung perubahan luas Mead permukaan Danau

Untuk menghitung perubahan luas permukaan Danau Mead, pertama-tama ekspor hasil EOJ ke Amazon S3 dengan menggunakan: `export_earth_observation_job`

```
sagemaker_session = sagemaker.Session()
s3_bucket_name = sagemaker_session.default_bucket()  # Replace with your own bucket if
 needed
```

```
s3_bucket = session.resource("s3").Bucket(s3_bucket_name)
prefix = "export-lake-mead-eoj"  # Replace with the S3 prefix desired
export_bucket_and_key = f"s3://{s3_bucket_name}/{prefix}/"

eoj_output_config = {"S3Data": {"S3Uri": export_bucket_and_key}}
export_response = sm_geo_client.export_earth_observation_job(
    Arn="arn:aws:sagemaker-geospatial:us-west-2:111122223333:earth-observation-
job/7xgwzijebynp",
    ExecutionRoleArn=execution_role_arn,
    OutputConfig=eoj_output_config,
    ExportSourceImages=False,
)
```

Untuk melihat status pekerjaan ekspor Anda, gunakan`get_earth_observation_job`:

```
export_job_details =
 sm_geo_client.get_earth_observation_job(Arn=export_response["Arn"])
```

Untuk menghitung perubahan ketinggian air Danau Mead, unduh masker tutupan lahan ke instance SageMaker notebook lokal dan unduh gambar sumber dari kueri kami sebelumnya. Dalam peta kelas untuk model segmentasi tanah, indeks kelas air adalah 6.

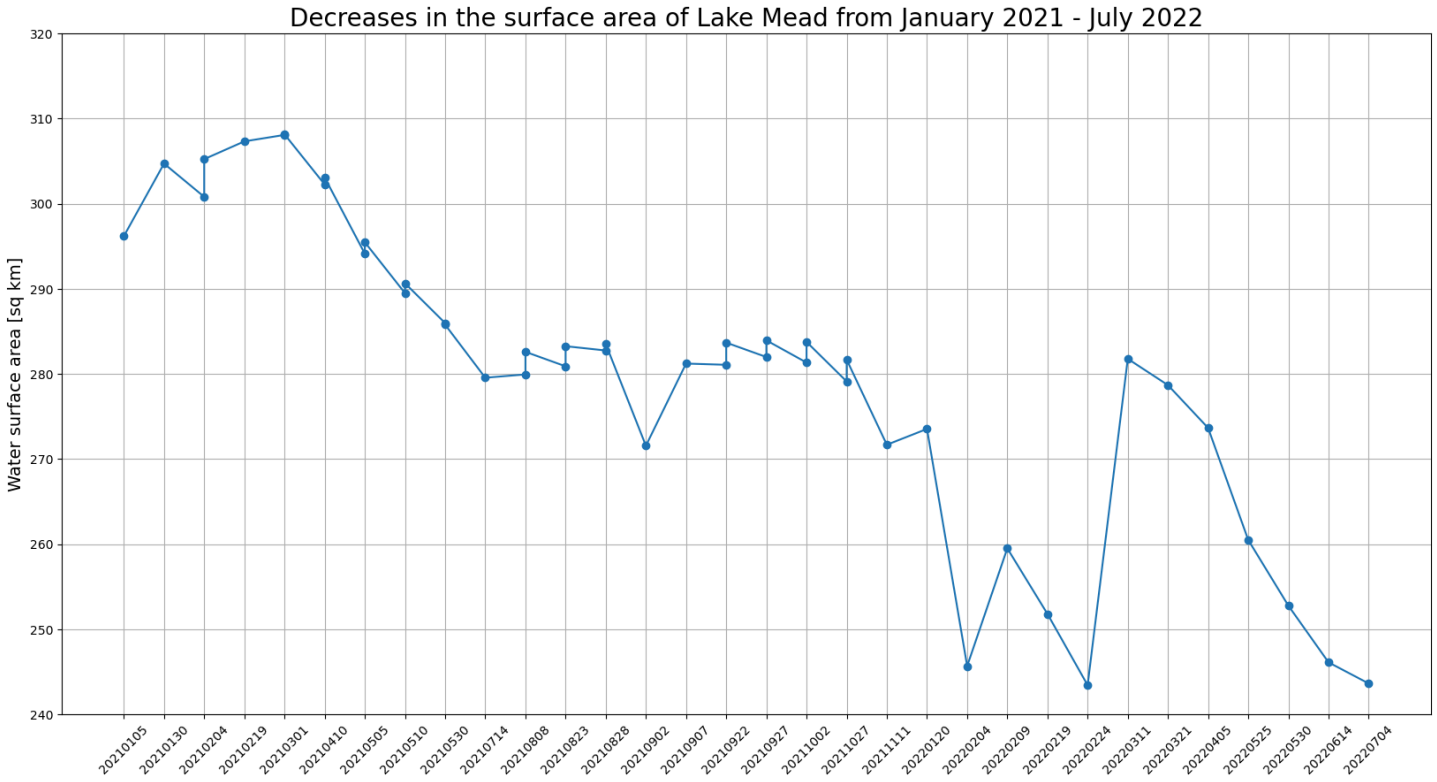
Untuk mengekstrak masker air dari Sentinel-2 gambar, ikuti langkah-langkah ini. Pertama, hitung jumlah piksel yang ditandai sebagai air (indeks kelas 6) pada gambar. Kedua, kalikan hitungan dengan area yang dicakup setiap piksel. Band dapat berbeda dalam resolusi spasialnya. Untuk model segmentasi tutupan lahan, semua pita diambil sampelnya ke resolusi spasial sebesar 60 meter.

```
import os
from glob import glob
import cv2
import numpy as np
import tifffile
import matplotlib.pyplot as plt
from urllib.parse import urlparse
from botocore import UNSIGNED
from botocore.config import Config

# Download land cover masks
mask_dir = "./masks/lake_mead"
os.makedirs(mask_dir, exist_ok=True)
image_paths = []
```

```
for s3_object in s3_bucket.objects.filter(Prefix=prefix).all():
    path, filename = os.path.split(s3_object.key)
    if "output" in path:
        mask_name = mask_dir + "/" + filename
        s3_bucket.download_file(s3_object.key, mask_name)
        print("Downloaded mask: " + mask_name)

# Download source images for visualization
for tci_url in tci_urls:
    url_parts = urlparse(tci_url)
    img_id = url_parts.path.split("/")[-2]
    tci_download_path = image_dir + "/" + img_id + "_TCI.tif"
    cogs_bucket = session.resource(
        "s3", config=Config(signature_version=UNSIGNED, region_name="us-west-2")
    ).Bucket(url_parts.hostname.split(".")[0])
    cogs_bucket.download_file(url_parts.path[1:], tci_download_path)
    print("Downloaded image: " + img_id)

print("Downloads complete.")

image_files = glob("images/lake_mead/*.tif")
mask_files = glob("masks/lake_mead/*.tif")
image_files.sort(key=lambda x: x.split("SQA_")[1])
mask_files.sort(key=lambda x: x.split("SQA_")[1])
overlay_dir = "./masks/lake_mead_overlay"
os.makedirs(overlay_dir, exist_ok=True)
lake_areas = []
mask_dates = []

for image_file, mask_file in zip(image_files, mask_files):
    image_id = image_file.split("/")[-1].split("_TCI")[0]
    mask_id = mask_file.split("/")[-1].split(".tif")[0]
    mask_date = mask_id.split("_")[2]
    mask_dates.append(mask_date)
    assert image_id == mask_id
    image = tifffile.imread(image_file)
    image_ds = cv2.resize(image, (1830, 1830), interpolation=cv2.INTER_LINEAR)
    mask = tifffile.imread(mask_file)
    water_mask = np.isin(mask, [6]).astype(np.uint8)  # water has a class index 6
    lake_mask = water_mask[1000:, :1100]
    lake_area = lake_mask.sum() * 60 * 60 / (1000 * 1000)  # calculate the surface area
    lake_areas.append(lake_area)
    contour, _ = cv2.findContours(water_mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    combined = cv2.drawContours(image_ds, contour, -1, (255, 0, 0), 4)
```

```
    lake_crop = combined[1000:, :1100]
    cv2.putText(lake_crop, f"{mask_date}", (10,50), cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0,
 0, 0), 3, cv2.LINE_AA)
    cv2.putText(lake_crop, f"{lake_area} [sq km]", (10,100), cv2.FONT_HERSHEY_SIMPLEX,
 1.5, (0, 0, 0), 3, cv2.LINE_AA)
    overlay_file = overlay_dir + '/' + mask_date + '.png'
    cv2.imwrite(overlay_file, cv2.cvtColor(lake_crop, cv2.COLOR_RGB2BGR))

# Plot water surface area vs. time.
plt.figure(figsize=(20,10))
plt.title('Lake Mead surface area for the 2021.02 - 2022.07 period.', fontsize=20)
plt.xticks(rotation=45)
plt.ylabel('Water surface area [sq km]', fontsize=14)
plt.plot(mask_dates, lake_areas, marker='o')
plt.grid('on')
plt.ylim(240, 320)
for i, v in enumerate(lake_areas):
    plt.text(i, v+2, "%d" %v, ha='center')
plt.show()
```

Dengan menggunakan`matplotlib`, Anda dapat memvisualisasikan hasilnya dengan grafik. Grafik menunjukkan bahwa luas permukaan Danau Mead menurun dari Januari 2021—Juli 2022.



Menggunakan pekerjaan pemrosesan untuk beban kerja geospasial kustom

Dengan Amazon SageMaker Processing, Anda dapat menggunakan pengalaman terkelola yang disederhanakan SageMaker untuk menjalankan beban kerja pemrosesan data dengan wadah geospasial yang dibuat khusus.

Infrastruktur dasar untuk pekerjaan Amazon SageMaker Processing sepenuhnya dikelola oleh SageMaker. Selama pekerjaan pemrosesan, sumber daya klaster disediakan selama durasi pekerjaan Anda, dan dibersihkan saat pekerjaan selesai.

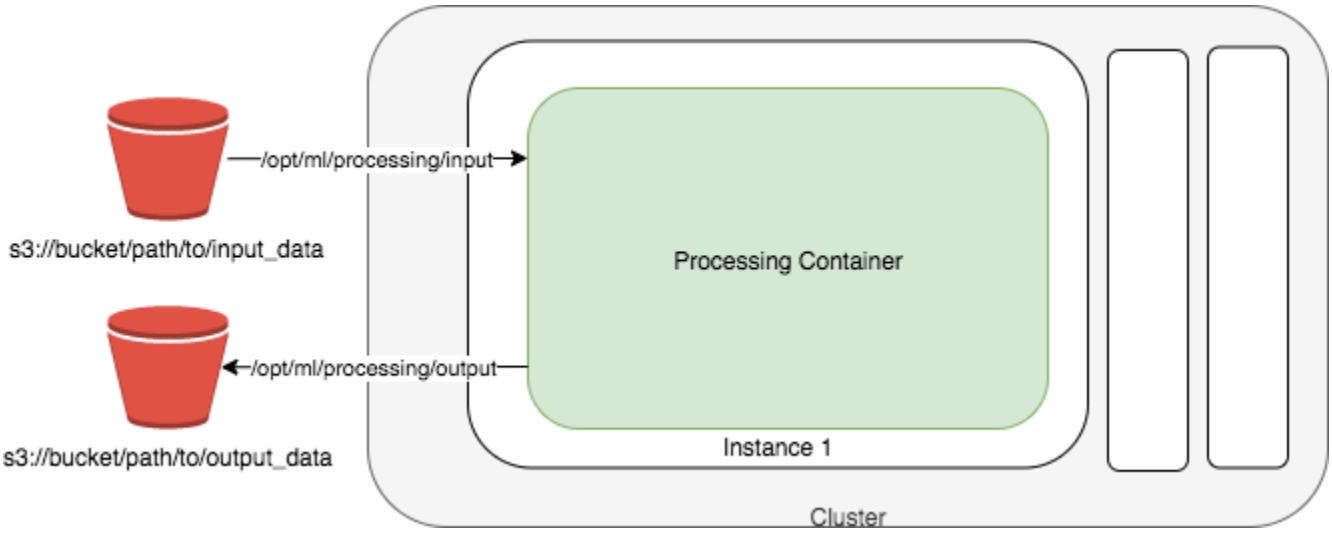


Diagram sebelumnya menunjukkan bagaimana SageMaker memutar pekerjaan pemrosesan geospasial. SageMaker mengambil skrip beban kerja geospasial Anda, menyalin data geospasial Anda dari Amazon Simple Storage Service (Amazon S3), dan kemudian menarik wadah geospasial yang ditentukan. Infrastruktur yang mendasari untuk pekerjaan pemrosesan sepenuhnya dikelola oleh SageMaker. Sumber daya cluster disediakan selama durasi pekerjaan Anda, dan dibersihkan saat pekerjaan selesai. Output dari pekerjaan pemrosesan disimpan dalam ember yang Anda tentukan.

⚠ Kendala penamaan jalur

Jalur lokal di dalam wadah pekerjaan Pemrosesan harus dimulai dengan**`/opt/ml/processing/`**.

SageMaker geospasial menyediakan wadah yang dibuat khusus, `081189585635.dkr.ecr.us-west-2.amazonaws.com/sagemaker-geospatial-v1-0:latest` yang dapat ditentukan saat menjalankan pekerjaan pemrosesan.

Topik

Ikhtisar: Jalankan pekerjaan pemrosesan menggunakan **`ScriptProcessor`** dan wadah SageMaker geospasial

SageMaker geospasial menyediakan wadah pemrosesan yang dibangun khusus,. `081189585635.dkr.ecr.us-west-2.amazonaws.com/sagemaker-geospatial-v1-0:latest` Anda dapat menggunakan wadah ini saat menjalankan pekerjaan dengan Amazon SageMaker Processing. Saat Anda membuat instance `ScriptProcessor`kelas yang tersedia melalui Amazon SageMaker Python SDK for Processing, tentukan ini. `image_uri`

> **Note**
>
> Jika Anda menerima ResourceLimitExceededkesalahan saat mencoba memulai pekerjaan pemrosesan, Anda perlu meminta peningkatan kuota. Untuk memulai permintaan peningkatan kuota Service Quotas, lihat Meminta peningkatan kuota pada Panduan Pengguna Service Quotas

Prasyarat untuk menggunakan **`ScriptProcessor`**

1. Anda telah membuat Python skrip yang menentukan beban kerja MS geospasial Anda.
2. Anda telah memberikan akses peran SageMaker eksekusi ke bucket Amazon S3 apa pun yang diperlukan.
3. Siapkan data Anda untuk diimpor ke dalam wadah. Pekerjaan Amazon SageMaker Processing mendukung pengaturan `s3_data_type` sama dengan `"ManifestFile"` atau ke`"S3Prefix"`.

Prosedur berikut menunjukkan cara membuat instance `ScriptProcessor` dan mengirimkan pekerjaan Amazon SageMaker Processing menggunakan wadah SageMaker geospasial.

Untuk membuat **`ScriptProcessor`** instance dan mengirimkan pekerjaan Amazon SageMaker Processing menggunakan wadah SageMaker geospasial

1. Buat instance `ScriptProcessor` kelas menggunakan gambar SageMaker geospasial:

```
from sagemaker.processing import ScriptProcessor, ProcessingInput, ProcessingOutput

sm_session = sagemaker.session.Session()
execution_role_arn = sagemaker.get_execution_role()

# purpose-built geospatial container
image_uri = '081189585635.dkr.ecr.us-west-2.amazonaws.com/sagemaker-geospatial-v1-0:latest'

script_processor = ScriptProcessor(
 command=['python3'],
 image_uri=image_uri,
 role=execution_role_arn,
 instance_count=4,
 instance_type='ml.m5.4xlarge',
 sagemaker_session=sm_session
)
```

   Ganti *`execution_role_arn`* dengan ARN dari peran eksekusi yang memiliki SageMaker akses ke data input yang disimpan di Amazon S3 dan layanan AWS lain yang ingin Anda panggil dalam pekerjaan pemrosesan Anda. Anda dapat memperbarui `instance_count` dan `instance_type` untuk mencocokkan persyaratan pekerjaan pemrosesan Anda.

2. Untuk memulai pekerjaan pemrosesan, gunakan `.run()` metode ini:

```
# Can be replaced with any S3 compliant string for the name of the folder.
s3_folder = geospatial-data-analysis

# Use .default_bucket() to get the name of the S3 bucket associated with your current
 SageMaker session
s3_bucket = sm_session.default_bucket()

s3_manifest_uri = f's3://{s3_bucket}/{s3_folder}/manifest.json'
s3_prefix_uri =  f's3://{s3_bucket}/{s3_folder}/image-prefix

script_processor.run(
 code='preprocessing.py',
```

```
 inputs=[
  ProcessingInput(
   source=s3_manifest_uri | s3_prefix_uri ,
   destination='/opt/ml/processing/input_data/',
   s3_data_type= "ManifestFile" | "S3Prefix",
   s3_data_distribution_type= "ShardedByS3Key" | "FullyReplicated"
  )
 ],
 outputs=[
        ProcessingOutput(
            source='/opt/ml/processing/output_data/',
            destination=s3_output_prefix_url
        )
    ]
)
```

- Ganti *preprocessing.py* dengan nama skrip pemrosesan data Python Anda sendiri.
- Pekerjaan pemrosesan mendukung dua metode untuk memformat data input Anda. Anda dapat membuat file manifes yang menunjuk ke semua data input untuk pekerjaan pemrosesan Anda, atau Anda dapat menggunakan awalan umum pada setiap input data individual. Jika Anda membuat set file manifes `s3_manifest_uri` sama dengan`"ManifestFile"`. Jika Anda menggunakan awalan file yang disetel `s3_manifest_uri` sama dengan`"S3Prefix"`. Anda menentukan jalur ke data Anda menggunakan`source`.
- Anda dapat mendistribusikan data pekerjaan pemrosesan Anda dengan dua cara:
  - Mendistribusikan data Anda ke semua instance pemrosesan dengan menyetel `s3_data_distribution_type` sama dengan`FullyReplicated`.
  - Mendistribusikan data Anda dalam pecahan berdasarkan kunci Amazon S3 dengan `s3_data_distribution_type` menyetel sama dengan. `ShardedByS3Key` Bila Anda menggunakan `ShardedByS3Key` satu pecahan data dikirim ke setiap instance pemrosesan.

Anda dapat menggunakan skrip untuk memproses data SageMaker geospasial. Skrip itu dapat ditemukan di Langkah 3: Menulis skrip yang dapat menghitung NDVI. Untuk mempelajari lebih lanjut tentang operasi `.run()` API, lihat `run`di Amazon SageMaker Python SDK for Processing.

Untuk memantau kemajuan pekerjaan pemrosesan Anda, `ProcessingJobs` kelas mendukung `describe`metode. Metode ini mengembalikan respons dari panggilan `DescribeProcessingJob` API. Untuk mempelajari selengkapnya, lihat `DescribeProcessingJob`di Referensi Amazon SageMaker API.

Topik berikutnya menunjukkan cara membuat instance `ScriptProcessor` kelas menggunakan wadah SageMaker geospasial, dan kemudian bagaimana menggunakannya untuk menghitung Normalized Difference Vegetation Index (NDVI) dengan gambar. Sentinel-2

Menggunakan **`ScriptProcessor`** untuk menghitung Normalized Difference Vegetation Index (NDVI) menggunakan data satelit Sentinel-2

Contoh kode berikut menunjukkan kepada Anda cara menghitung indeks vegetasi perbedaan yang dinormalisasi dari area geografis tertentu menggunakan gambar geospasial yang dibuat khusus dalam notebook Studio Classic dan menjalankan beban kerja skala besar dengan Amazon Processing SageMaker menggunakan dari Python SDK. `ScriptProcessor` SageMaker

Demo ini juga menggunakan instance notebook Amazon SageMaker Studio Classic yang menggunakan kernel geospasial dan jenis instans. Untuk mempelajari cara membuat instance notebook geospasial Studio Classic, lihatMembuat notebook Amazon SageMaker Studio Classic menggunakan gambar geospasial.

Anda dapat mengikuti demo ini di instance notebook Anda sendiri dengan menyalin dan menempelkan cuplikan kode berikut:

1. Gunakan `search_raster_data_collection` untuk menanyakan area minat tertentu (AOI) selama rentang waktu tertentu menggunakan pengumpulan data raster tertentu,. Sentinel-2
2. Buat file manifes yang menentukan data apa yang akan diproses selama pekerjaan pemrosesan.
3. Tulis skrip Python pemrosesan data yang menghitung NDVI.
4. Buat `ScriptProcessor` instance dan mulai pekerjaan Amazon SageMaker Processing.
5. Memvisualisasikan hasil pekerjaan pemrosesan Anda.

Kueri pengumpulan data Sentinel-2 raster menggunakan **`SearchRasterDataCollection`**

Dengan `search_raster_data_collection` Anda dapat menanyakan koleksi data raster yang didukung. Contoh ini menggunakan data yang diambil dari Sentinel-2 satelit. Area minat (`AreaOfInterest`) yang ditentukan adalah pedesaan Iowa utara, dan rentang waktu (`TimeRangeFilter`) adalah 1 Januari 2022 hingga 30 Desember 2022. Untuk melihat koleksi data raster yang tersedia dalam Wilayah AWS penggunaan `list_raster_data_collections` Anda. Untuk melihat contoh kode menggunakan API ini, lihat `ListRasterDataCollections`di Panduan SageMaker Pengembang Amazon.

Dalam contoh kode berikut Anda menggunakan ARN yang terkait dengan pengumpulan data Sentinel-2 raster,. `arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/public/nmqj48dcu3g7ayw8`

Permintaan `search_raster_data_collection` API membutuhkan dua parameter:

- Anda perlu menentukan `Arn` parameter yang sesuai dengan pengumpulan data raster yang ingin Anda kueri.
- Anda juga perlu menentukan `RasterDataCollectionQuery` parameter, yang mengambil Python kamus.

Contoh kode berikut berisi pasangan kunci-nilai yang diperlukan untuk `RasterDataCollectionQuery` parameter yang disimpan ke variabel. `search_rdc_query`

```
search_rdc_query = {
    "AreaOfInterest": {
        "AreaOfInterestGeometry": {
            "PolygonGeometry": {
                "Coordinates": [[
                    [
              -94.50938680498298,
              43.22487436936203
            ],
            [
              -94.50938680498298,
              42.843474642037194
            ],
            [
              -93.86520004156142,
              42.843474642037194
            ],
            [
              -93.86520004156142,
              43.22487436936203
            ],
            [
              -94.50938680498298,
              43.22487436936203
            ]
               ]]
            }
```

```
        }
    },
    "TimeRangeFilter": {"StartTime": "2022-01-01T00:00:00Z", "EndTime":
 "2022-12-30T23:59:59Z"}
}
```

Untuk membuat `search_raster_data_collection` permintaan, Anda harus menentukan ARN dari pengumpulan data Sentinel-2 raster:. `arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/public/nmqj48dcu3g7ayw8` Anda juga harus meneruskan kamus Python yang telah ditentukan sebelumnya, yang menentukan parameter kueri.

```
## Creates a SageMaker Geospatial client instance
sm_geo_client= session.create_client(service_name="sagemaker-geospatial")

search_rdc_response1 = sm_geo_client.search_raster_data_collection(
    Arn='arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/
public/nmqj48dcu3g7ayw8',
    RasterDataCollectionQuery=search_rdc_query
)
```

Hasil API ini tidak dapat dipaginasi. Untuk mengumpulkan semua gambar satelit yang dikembalikan oleh `search_raster_data_collection` operasi, Anda dapat menerapkan `while` loop. Ini memeriksa `NextToken` dalam respons API:

```
## Holds the list of API responses from search_raster_data_collection
items_list = []
while search_rdc_response1.get('NextToken') and search_rdc_response1['NextToken'] !=
 None:
    items_list.extend(search_rdc_response1['Items'])

    search_rdc_response1 = sm_geo_client.search_raster_data_collection(
     Arn='arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/
public/nmqj48dcu3g7ayw8',
        RasterDataCollectionQuery=search_rdc_query,
        NextToken=search_rdc_response1['NextToken']
    )
```

Respons API menampilkan daftar URL di bawah `Assets` kunci yang sesuai dengan pita gambar tertentu. Berikut ini adalah versi respons API yang terpotong. Beberapa pita gambar telah dihapus untuk kejelasan.

```
{
 'Assets': {
        'aot': {
            'Href': 'https://sentinel-cogs.s3.us-west-2.amazonaws.com/sentinel-s2-l2a-
cogs/15/T/UH/2022/12/S2A_15TUH_20221230_0_L2A/AOT.tif'
        },
        'blue': {
            'Href': 'https://sentinel-cogs.s3.us-west-2.amazonaws.com/sentinel-s2-l2a-
cogs/15/T/UH/2022/12/S2A_15TUH_20221230_0_L2A/B02.tif'
        },
        'swir22-jp2': {
            'Href': 's3://sentinel-s2-l2a/tiles/15/T/UH/2022/12/30/0/B12.jp2'
        },
        'visual-jp2': {
            'Href': 's3://sentinel-s2-l2a/tiles/15/T/UH/2022/12/30/0/TCI.jp2'
        },
        'wvp-jp2': {
            'Href': 's3://sentinel-s2-l2a/tiles/15/T/UH/2022/12/30/0/WVP.jp2'
        }
    },
    'DateTime': datetime.datetime(2022, 12, 30, 17, 21, 52, 469000, tzinfo =
 tzlocal()),
    'Geometry': {
        'Coordinates': [
            [
                [-95.46676936182894, 43.32623760511659],
                [-94.11293433656887, 43.347431265475954],
                [-94.09532154452742, 42.35884880571144],
                [-95.42776890002203, 42.3383710796791],
                [-95.46676936182894, 43.32623760511659]
            ]
        ],
        'Type': 'Polygon'
    },
    'Id': 'S2A_15TUH_20221230_0_L2A',
    'Properties': {
        'EoCloudCover': 62.384969,
        'Platform': 'sentinel-2a'
    }
}
```

Di bagian berikutnya, Anda membuat file manifes menggunakan `'Id'` kunci dari respons API.

Membuat file manifes masukan menggunakan **`Id`** kunci dari respons **`search_raster_data_collection`** API

Saat menjalankan pekerjaan pemrosesan, Anda harus menentukan input data dari Amazon S3. Tipe data input dapat berupa file manifes, yang kemudian menunjuk ke file data individual. Anda juga dapat menambahkan awalan ke setiap file yang ingin diproses. Contoh kode berikut mendefinisikan folder tempat file manifes Anda akan dihasilkan.

Gunakan SDK for Python (Boto3) untuk mendapatkan bucket default dan ARN dari peran eksekusi yang terkait dengan instance notebook Studio Classic Anda:

```
sm_session = sagemaker.session.Session()
s3 = boto3.resource('s3')
# Gets the default excution role associated with the notebook
execution_role_arn = sagemaker.get_execution_role()

# Gets the default bucket associated with the notebook
s3_bucket = sm_session.default_bucket()

# Can be replaced with any name
s3_folder = "script-processor-input-manifest"
```

Selanjutnya, Anda membuat file manifes. Ini akan menyimpan URL gambar satelit yang ingin Anda proses ketika Anda menjalankan pekerjaan pemrosesan Anda nanti di langkah 4.

```
# Format of a manifest file
manifest_prefix = {}
manifest_prefix['prefix'] = 's3://' + s3_bucket + '/' + s3_folder + '/'
manifest = [manifest_prefix]

print(manifest)
```

Contoh kode berikut mengembalikan URI S3 tempat file manifes Anda akan dibuat.

```
[{'prefix': 's3://sagemaker-us-west-2-111122223333/script-processor-input-manifest/'}]
```

Semua elemen respons dari respons search_raster_data_collection tidak diperlukan untuk menjalankan pekerjaan pemrosesan.

Cuplikan kode berikut menghapus elemen yang tidak perlu`'Properties'`,`'Geometry'`, dan. `'DateTime'` Pasangan `'Id'` kunci-nilai,`'Id': 'S2A_15TUH_20221230_0_L2A'`,

berisi tahun dan bulan. Contoh kode berikut mem-parsing data tersebut untuk membuat kunci baru dalam Python **`dict_month_items`** kamus. Nilai adalah aset yang dikembalikan dari `SearchRasterDataCollection` kueri.

```
# For each response get the month and year, and then remove the metadata not related to
 the satelite images.
dict_month_items = {}
for item in items_list:
    # Example ID being split: 'S2A_15TUH_20221230_0_L2A'
    yyyymm = item['Id'].split("_")[2][:6]
    if yyyymm not in dict_month_items:
        dict_month_items[yyyymm] = []

    # Removes uneeded metadata elements for this demo
    item.pop('Properties', None)
    item.pop('Geometry', None)
    item.pop('DateTime', None)

    # Appends the response from search_raster_data_collection to newly created key
 above
    dict_month_items[yyyymm].append(item)
```

Contoh kode ini mengunggah `dict_month_items` ke Amazon S3 sebagai objek JSON menggunakan [`.upload_file()`](#)operasi API:

```
## key_ is the yyyymm timestamp formatted above
## value_ is the reference to all the satellite images collected via our searchRDC
 query
for key_, value_ in dict_month_items.items():
    filename = f'manifest_{key_}.json'
    with open(filename, 'w') as fp:
        json.dump(value_, fp)
    s3.meta.client.upload_file(filename, s3_bucket, s3_folder + '/' + filename)
    manifest.append(filename)
    os.remove(filename)
```

Contoh kode ini mengunggah `manifest.json` file induk yang menunjuk ke semua manifes lain yang diunggah ke Amazon S3. Ini juga menyimpan jalur ke variabel lokal:**`s3_manifest_uri`**. Anda akan menggunakan variabel itu lagi untuk menentukan sumber data input saat Anda menjalankan pekerjaan pemrosesan di langkah 4.

```
with open('manifest.json', 'w') as fp:
    json.dump(manifest, fp)
s3.meta.client.upload_file('manifest.json', s3_bucket, s3_folder + '/' +
 'manifest.json')
os.remove('manifest.json')

s3_manifest_uri = f's3://{s3_bucket}/{s3_folder}/manifest.json'
```

Sekarang setelah Anda membuat file manifes masukan dan mengunggahnya, Anda dapat menulis skrip yang memproses data Anda dalam pekerjaan pemrosesan. Ini memproses data dari citra satelit, menghitung NDVI, dan kemudian mengembalikan hasilnya ke lokasi Amazon S3 yang berbeda.

Tulis skrip yang menghitung NDVI

Amazon SageMaker Studio Classic mendukung penggunaan perintah sihir `%%writefile` sel. Setelah menjalankan sel dengan perintah ini, isinya akan disimpan ke direktori Studio Classic lokal Anda. Ini adalah kode khusus untuk menghitung NDVI. Namun, berikut ini dapat berguna ketika Anda menulis skrip Anda sendiri untuk pekerjaan pemrosesan:

- Dalam wadah pekerjaan pemrosesan Anda, jalur lokal di dalam wadah harus dimulai`/opt/ml/processing/`. Dalam contoh ini, **`input_data_path = '/opt/ml/processing/input_data/'`** dan **`processed_data_path = '/opt/ml/processing/output_data/'`** ditentukan dengan cara itu.
- Dengan Amazon SageMaker Processing, skrip yang menjalankan pekerjaan pemrosesan dapat mengunggah data yang diproses langsung ke Amazon S3. Untuk melakukannya, pastikan peran eksekusi yang terkait dengan `ScriptProcessor` instans Anda memiliki persyaratan yang diperlukan untuk mengakses bucket S3. Anda juga dapat menentukan parameter output saat menjalankan pekerjaan pemrosesan. Untuk mempelajari lebih lanjut, lihat operasi `.run()` API di Amazon SageMaker Python SDK. Dalam contoh kode ini, hasil pemrosesan data diunggah langsung ke Amazon S3.
- Untuk mengelola ukuran Amazon EBSContainer yang dilampirkan ke pemrosesan Anda, gunakan parameternya. `volume_size_in_gb` Ukuran default kontainer adalah 30 GB. Anda juga dapat menggunakan library Python Garbage Collector secara opsional untuk mengelola penyimpanan di wadah Amazon EBS Anda.

  Contoh kode berikut memuat array ke dalam wadah pekerjaan pemrosesan. Saat array membangun dan mengisi memori, pekerjaan pemrosesan macet. Untuk mencegah kerusakan ini, contoh berikut berisi perintah yang menghapus array dari wadah pekerjaan pemrosesan.

```
%%writefile compute_ndvi.py

import os
import pickle
import sys
import subprocess
import json
import rioxarray

if __name__ == "__main__":
    print("Starting processing")

    input_data_path = '/opt/ml/processing/input_data/'
    input_files = []

    for current_path, sub_dirs, files in os.walk(input_data_path):
        for file in files:
            if file.endswith(".json"):
                input_files.append(os.path.join(current_path, file))

    print("Received {} input_files: {}".format(len(input_files), input_files))

    items = []
    for input_file in input_files:
        full_file_path = os.path.join(input_data_path, input_file)
        print(full_file_path)
        with open(full_file_path, 'r') as f:
            items.append(json.load(f))

    items = [item for sub_items in items for item in sub_items]

    for item in items:
        red_uri = item["Assets"]["red"]["Href"]
        nir_uri = item["Assets"]["nir"]["Href"]

        red = rioxarray.open_rasterio(red_uri, masked=True)
        nir = rioxarray.open_rasterio(nir_uri, masked=True)

        ndvi = (nir - red)/ (nir + red)

        file_name = 'ndvi_' + item["Id"] + '.tif'
        output_path = '/opt/ml/processing/output_data'
        output_file_path = f"{output_path}/{file_name}"
```

```
        ndvi.rio.to_raster(output_file_path)
        print("Written output:", output_file_path)
```

Anda sekarang memiliki skrip yang dapat menghitung NDVI. Selanjutnya, Anda dapat membuat instance dari ScriptProcessor dan menjalankan pekerjaan Processing Anda.

Membuat sebuah instance dari **`ScriptProcessor`** kelas

Demo ini menggunakan ScriptProcessorkelas yang tersedia melalui Amazon SageMaker Python SDK. Pertama, Anda perlu membuat instance dari kelas, dan kemudian Anda dapat memulai pekerjaan Processing Anda dengan menggunakan `.run()` metode.

```
from sagemaker.processing import ScriptProcessor, ProcessingInput, ProcessingOutput

image_uri = '081189585635.dkr.ecr.us-west-2.amazonaws.com/sagemaker-geospatial-v1-0:latest'

processor = ScriptProcessor(
 command=['python3'],
 image_uri=image_uri,
 role=execution_role_arn,
 instance_count=4,
 instance_type='ml.m5.4xlarge',
 sagemaker_session=sm_session
)

print('Starting processing job.')
```

Ketika Anda memulai pekerjaan Processing Anda, Anda perlu menentukan `ProcessingInput`objek. Dalam objek itu, Anda menentukan yang berikut:

- Jalur ke file manifes yang Anda buat di langkah 2,**`s3_manifest_uri`**. Ini adalah sumber data input ke wadah.
- Jalur ke tempat Anda ingin data input disimpan dalam wadah. Ini harus sesuai dengan jalur yang Anda tentukan dalam skrip Anda.
- Gunakan `s3_data_type` parameter untuk menentukan input sebagai`"ManifestFile"`.

```
s3_output_prefix_url = f"s3://{s3_bucket}/{s3_folder}/output"
```

```
processor.run(
    code='compute_ndvi.py',
    inputs=[
        ProcessingInput(
            source=s3_manifest_uri,
            destination='/opt/ml/processing/input_data/',
            s3_data_type="ManifestFile",
            s3_data_distribution_type="ShardedByS3Key"
        ),
    ],
    outputs=[
        ProcessingOutput(
            source='/opt/ml/processing/output_data/',
            destination=s3_output_prefix_url,
            s3_upload_mode="Continuous"
        )
    ]
)
```

Contoh kode berikut menggunakan `.describe()`metode untuk mendapatkan rincian pekerjaan Processing Anda.

```
preprocessing_job_descriptor = processor.jobs[-1].describe()
s3_output_uri = preprocessing_job_descriptor["ProcessingOutputConfig"]["Outputs"][0]
["S3Output"]["S3Uri"]
print(s3_output_uri)
```

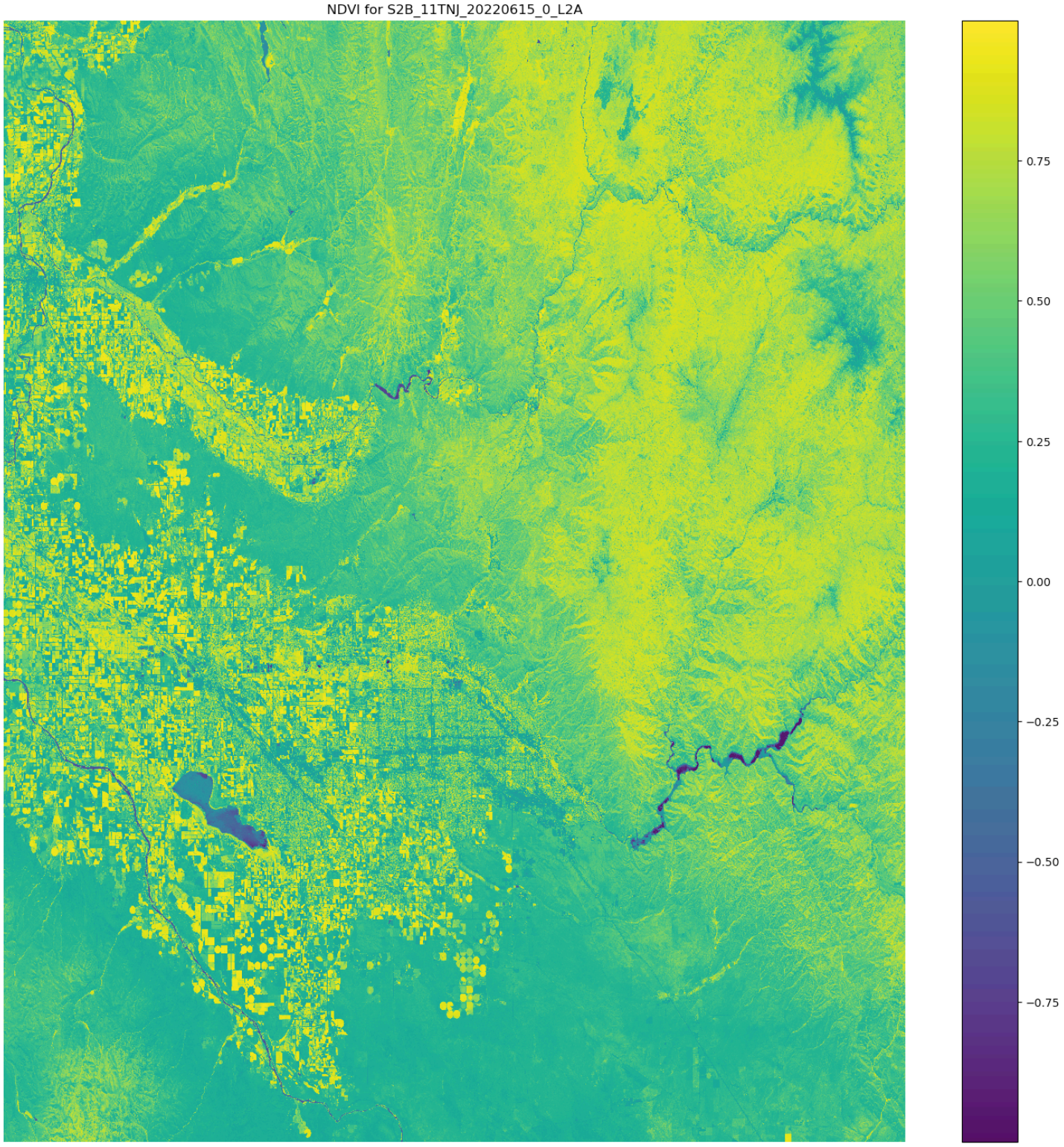
Memvisualisasikan hasil Anda menggunakan **`matplotlib`**

Dengan pustaka Python Matplotlib, Anda dapat memplot data raster. Sebelum Anda memplot data, Anda perlu menghitung NDVI menggunakan gambar sampel dari satelit. Sentinel-2 Contoh kode berikut membuka array gambar menggunakan operasi `.open_rasterio()` API, dan kemudian menghitung NDVI menggunakan `nir` dan pita `red` gambar dari data satelit. Sentinel-2

```
# Opens the python arrays
import rioxarray

red_uri = items[25]["Assets"]["red"]["Href"]
nir_uri = items[25]["Assets"]["nir"]["Href"]

red = rioxarray.open_rasterio(red_uri, masked=True)
```

```
nir = rioxarray.open_rasterio(nir_uri, masked=True)

# Calculates the NDVI
ndvi = (nir - red)/ (nir + red)

# Common plotting library in Python
import matplotlib.pyplot as plt

f, ax = plt.subplots(figsize=(18, 18))
ndvi.plot(cmap='viridis', ax=ax)
ax.set_title("NDVI for {}".format(items[25]["Id"]))
ax.set_axis_off()
plt.show()
```

Output dari contoh kode sebelumnya adalah citra satelit dengan nilai NDVI yang dilapis di atasnya. Nilai NDVI mendekati 1 menunjukkan banyak vegetasi yang ada, dan nilai mendekati 0 menunjukkan tidak ada vegetasi yang ditampilkan.



Ini melengkapi demo penggunaan`ScriptProcessor`.

Pekerjaan Pengamatan Bumi

Menggunakan pekerjaan Observasi Bumi (EOJ), Anda dapat memperoleh, mengubah, dan memvisualisasikan data geospasial untuk membuat prediksi. Anda dapat memilih operasi berdasarkan kasus penggunaan Anda dari berbagai operasi dan model. Anda mendapatkan fleksibilitas dalam memilih bidang minat Anda, memilih penyedia data, dan mengatur berbasis rentang waktu dan cloud-cover-percentage-based filter. Setelah SageMaker membuat EOJ untuk Anda, Anda dapat memvisualisasikan input dan output pekerjaan menggunakan fungsionalitas visualisasi. EOJ memiliki berbagai kasus penggunaan yang mencakup membandingkan deforestasi dari waktu ke waktu dan mendiagnosis kesehatan tanaman. Anda dapat membuat EOJ dengan menggunakan SageMaker notebook dengan gambar SageMaker geospasial. Anda juga dapat mengakses UI SageMaker geospasial sebagai bagian dari Amazon SageMaker Studio Classic UI untuk melihat daftar semua pekerjaan Anda. Anda juga dapat menggunakan UI untuk menjeda atau menghentikan pekerjaan yang sedang berlangsung. Anda dapat memilih pekerjaan dari daftar EOJ yang tersedia untuk melihat ringkasan Job, rincian Job serta memvisualisasikan output Job.

Topik

Buat Pekerjaan Pengamatan Bumi Menggunakan Notebook Amazon SageMaker Studio Classic dengan Gambar SageMaker geospasial

Untuk menggunakan notebook SageMaker Studio Classic dengan gambar SageMaker geospasial:

1. Dari Peluncur, pilih Ubah lingkungan di bawah Notebook dan hitung sumber daya.
2. Selanjutnya, dialog Ubah lingkungan terbuka.
3. Pilih dropdown Gambar dan pilih Geospatial 1.0. Jenis Instance harus ml.geospatial.interactive. Jangan mengubah nilai default untuk pengaturan lain.
4. Pilih Pilih.
5. Pilih Buat Notebook.

Anda dapat memulai EOJ menggunakan notebook Amazon SageMaker Studio Classic dengan gambar SageMaker geospasial menggunakan kode yang disediakan di bawah ini.

```
import boto3
import sagemaker
import sagemaker_geospatial_map

session = boto3.Session()
execution_role = sagemaker.get_execution_role()
sg_client = session.client(service_name="sagemaker-geospatial")
```

Berikut ini adalah contoh yang menunjukkan cara membuat EOJ di Wilayah Barat AS (Oregon).

```
#Query and Access Data
search_rdc_args = {
    "Arn": "arn:aws:sagemaker-geospatial:us-west-2:378778860802:raster-data-collection/
public/nmqj48dcu3g7ayw8",  # sentinel-2 L2A COG
    "RasterDataCollectionQuery": {
        "AreaOfInterest": {
            "AreaOfInterestGeometry": {
                "PolygonGeometry": {
                    "Coordinates": [
                        [
                            [-114.529, 36.142],
                            [-114.373, 36.142],
                            [-114.373, 36.411],
                            [-114.529, 36.411],
                            [-114.529, 36.142],
                        ]
                    ]
                }
            }
        },
        "TimeRangeFilter": {
            "StartTime": "2021-01-01T00:00:00Z",
            "EndTime": "2022-07-10T23:59:59Z",
        },
        "PropertyFilters": {
            "Properties": [{"Property": {"EoCloudCover": {"LowerBound": 0,
 "UpperBound": 1}}}],
            "LogicalOperator": "AND",
        },
        "BandFilter": ["visual"],
    },
}
```

```
tci_urls = []
data_manifests = []
while search_rdc_args.get("NextToken", True):
    search_result = sg_client.search_raster_data_collection(**search_rdc_args)
    if search_result.get("NextToken"):
        data_manifests.append(search_result)
    for item in search_result["Items"]:
        tci_url = item["Assets"]["visual"]["Href"]
        print(tci_url)
        tci_urls.append(tci_url)

    search_rdc_args["NextToken"] = search_result.get("NextToken")

# Perform land cover segmentation on images returned from the sentinel dataset.
eoj_input_config = {
    "RasterDataCollectionQuery": {
        "RasterDataCollectionArn": "arn:aws:sagemaker-geospatial:us-
west-2:378778860802:raster-data-collection/public/nmqj48dcu3g7ayw8",
        "AreaOfInterest": {
            "AreaOfInterestGeometry": {
                "PolygonGeometry": {
                    "Coordinates": [
                        [
                            [-114.529, 36.142],
                            [-114.373, 36.142],
                            [-114.373, 36.411],
                            [-114.529, 36.411],
                            [-114.529, 36.142],
                        ]
                    ]
                }
            }
        },
        "TimeRangeFilter": {
            "StartTime": "2021-01-01T00:00:00Z",
            "EndTime": "2022-07-10T23:59:59Z",
        },
        "PropertyFilters": {
            "Properties": [{"Property": {"EoCloudCover": {"LowerBound": 0,
 "UpperBound": 1}}}],
            "LogicalOperator": "AND",
        },
    }
}
```

```
eoj_config = {"LandCoverSegmentationConfig": {}}

response = sg_client.start_earth_observation_job(
    Name="lake-mead-landcover",
    InputConfig=eoj_input_config,
    JobConfig=eoj_config,
    ExecutionRoleArn=execution_role,
)
```

Setelah EOJ Anda dibuat, EOJ `Arn` dikembalikan kepada Anda. Anda menggunakan `Arn` untuk mengidentifikasi pekerjaan dan melakukan operasi lebih lanjut. Untuk mendapatkan status pekerjaan, Anda bisa lari`sg_client.get_earth_observation_job(Arn = response['Arn'])`.

Contoh berikut menunjukkan bagaimana untuk query status EOJ sampai selesai.

```
eoj_arn = response["Arn"]
job_details = sg_client.get_earth_observation_job(Arn=eoj_arn)
{k: v for k, v in job_details.items() if k in ["Arn", "Status", "DurationInSeconds"]}
# List all jobs in the account
sg_client.list_earth_observation_jobs()["EarthObservationJobSummaries"]
```

Setelah EOJ selesai, Anda dapat memvisualisasikan output EOJ langsung di notebook. Contoh berikut menunjukkan kepada Anda bagaimana peta interaktif dapat dirender.

```
map = sagemaker_geospatial_map.create_map({
'is_raster': True
})
map.set_sagemaker_geospatial_client(sg_client)
# render the map
map.render()
```

Contoh berikut menunjukkan bagaimana peta dapat dipusatkan pada area yang diinginkan dan input dan output dari EOJ dapat dirender sebagai lapisan terpisah dalam peta.

```
# visualize the area of interest
config = {"label": "Lake Mead AOI"}
aoi_layer = map.visualize_eoj_aoi(Arn=eoj_arn, config=config)

# Visualize input.
time_range_filter = {
    "start_date": "2022-07-01T00:00:00Z",
```

```
    "end_date": "2022-07-10T23:59:59Z",
}
config = {"label": "Input"}

input_layer = map.visualize_eoj_input(
    Arn=eoj_arn, config=config, time_range_filter=time_range_filter
)
# Visualize output, EOJ needs to be in completed status.
time_range_filter = {
    "start_date": "2022-07-01T00:00:00Z",
    "end_date": "2022-07-10T23:59:59Z",
}
config = {"preset": "singleBand", "band_name": "mask"}
output_layer = map.visualize_eoj_output(
    Arn=eoj_arn, config=config, time_range_filter=time_range_filter
)
```

Anda dapat menggunakan `export_earth_observation_job` fungsi untuk mengekspor hasil EOJ ke bucket Amazon S3 Anda. Fungsi ekspor membuatnya nyaman untuk berbagi hasil di seluruh tim. SageMaker juga menyederhanakan manajemen dataset. Kami cukup membagikan hasil EOJ menggunakan pekerjaan ARN, alih-alih merayapi ribuan file di bucket S3. Setiap EOJ menjadi aset dalam katalog data, karena hasilnya dapat dikelompokkan berdasarkan pekerjaan ARN. Contoh berikut menunjukkan bagaimana Anda dapat mengekspor hasil EOJ.

```
sagemaker_session = sagemaker.Session()
s3_bucket_name = sagemaker_session.default_bucket()  # Replace with your own bucket if
 needed
s3_bucket = session.resource("s3").Bucket(s3_bucket_name)
prefix = "eoj_lakemead"  # Replace with the S3 prefix desired
export_bucket_and_key = f"s3://{s3_bucket_name}/{prefix}/"

eoj_output_config = {"S3Data": {"S3Uri": export_bucket_and_key}}
export_response = sg_client.export_earth_observation_job(
    Arn=eoj_arn,
    ExecutionRoleArn=execution_role,
    OutputConfig=eoj_output_config,
    ExportSourceImages=False,
)
```

Anda dapat memantau status pekerjaan ekspor Anda menggunakan cuplikan berikut.

```
# Monitor the export job status
```

```
export_job_details = sg_client.get_earth_observation_job(Arn=export_response["Arn"])
{k: v for k, v in export_job_details.items() if k in ["Arn", "Status",
 "DurationInSeconds"]}
```

Anda tidak dikenakan biaya penyimpanan setelah Anda menghapus EOJ.

Untuk contoh yang menampilkan cara menjalankan EOJ, lihat posting blog ini.

Untuk contoh notebook lainnya tentang kemampuan SageMaker geospasial, lihat repositori iniGitHub .

Jenis Operasi

Saat Anda membuat EOJ, Anda memilih operasi berdasarkan kasus penggunaan Anda. Kemampuan SageMaker geospasial Amazon menyediakan kombinasi operasi yang dibuat khusus dan model yang telah dilatih sebelumnya. Anda dapat menggunakan operasi ini untuk memahami dampak perubahan lingkungan dan aktivitas manusia dari waktu ke waktu atau mengidentifikasi piksel cloud dan bebas awan.

Penutupan Awan

Mengidentifikasi awan dalam citra satelit merupakan langkah pra-pemrosesan penting dalam menghasilkan data geospasial berkualitas tinggi. Mengabaikan piksel awan dapat menyebabkan kesalahan dalam analisis, dan deteksi piksel awan yang berlebihan dapat mengurangi jumlah pengamatan yang valid. Cloud masking memiliki kemampuan untuk mengidentifikasi piksel berawan dan bebas awan dalam citra satelit. Masker awan yang akurat membantu mendapatkan citra satelit untuk diproses dan meningkatkan pembuatan data. Berikut ini adalah peta kelas untuk cloud masking.

```
{
0: "No_cloud",
1: "cloud"
}
```

Penghapusan Awan

Penghapusan cloud untuk data Sentinel-2 menggunakan model segmentasi semantik berbasis ML untuk mengidentifikasi awan dalam gambar. Piksel berawan dapat diganti dengan piksel dari stempel waktu lainnya. USGS Landsatdata berisi metadata landsat yang digunakan untuk penghapusan cloud.

Statistik Temporal

Statistik temporal menghitung statistik untuk data geospasial melalui waktu. Statistik temporal yang saat ini didukung meliputi mean, median, dan standar deviasi. Anda dapat menghitung statistik ini dengan menggunakan `GROUPBY` dan mengaturnya ke salah satu `all` atau `yearly`. Anda juga dapat menyebutkan `TargetBands`.

Statistik Zonal

Statistik zona melakukan operasi statistik pada area tertentu pada gambar.

Resampling

Resampling digunakan untuk meningkatkan dan menurunkan resolusi gambar geospasial. `value` Atribut dalam resampling mewakili panjang sisi piksel.

Geomosaik

Geomosaic memungkinkan Anda menjahit gambar yang lebih kecil menjadi gambar besar.

Penumpukan Band

Penumpukan pita membutuhkan lebih dari satu pita gambar sebagai input dan menumpuknya menjadi satu GeoTIFF. `OutputResolution` Atribut menentukan resolusi gambar output. Berdasarkan resolusi gambar input, Anda dapat mengaturnya ke `lowest`, `highest` atau `average`.

Band Matematika

Band Math, juga dikenal sebagai Indeks Spektral, adalah proses mengubah pengamatan dari beberapa pita spektral menjadi satu pita, yang menunjukkan kelimpahan relatif fitur minat. Misalnya, Normalized Difference Vegetation Index (NDVI) dan Enhanced Vegetation Index (EVI) sangat membantu untuk mengamati keberadaan fitur vegetasi hijau.

Segmentasi Tutupan Lahan

Segmentasi Tutupan Lahan adalah model segmentasi semantik yang memiliki kemampuan untuk mengidentifikasi material fisik, seperti vegetasi, air, dan tanah kosong, di permukaan bumi. Memiliki cara yang akurat untuk memetakan pola tutupan lahan membantu Anda memahami dampak perubahan lingkungan dan aktivitas manusia dari waktu ke waktu. Segmentasi Tutupan Lahan sering digunakan untuk perencanaan wilayah, tanggap bencana, pengelolaan ekologi, dan penilaian dampak lingkungan. Berikut ini adalah peta kelas untuk segmentasi Tutupan Tanah.

```
{
0: "No_data",
1: "Saturated_or_defective",
2: "Dark_area_pixels",
3: "Cloud_shadows",
4: "Vegetation",
5: "Not_vegetated",
6: "Water",
7: "Unclassified",
8: "Cloud_medium_probability",
9: "Cloud_high_probability",
10: "Thin_cirrus",
11: "Snow_ice"
}
```

Ketersediaan Operasi EOJ

Ketersediaan operasi tergantung pada apakah Anda menggunakan UI SageMaker geospasial atau notebook Amazon SageMaker Studio Classic dengan gambar SageMaker geospasial. Saat ini, notebook mendukung semua fungsi. Untuk meringkas, operasi geospasial berikut didukung oleh: SageMaker

Operasi	Deskripsi	Ketersediaan
Penutupan Awan	Identifikasi piksel cloud dan bebas awan untuk mendapatk an citra satelit yang lebih baik dan akurat.	UI, Notebook
Penghapusan Awan	Hapus piksel yang berisi bagian awan dari citra satelit.	Buku catatan
Statistik Temporal	Hitung statistik melalui waktu untuk GeoTIFF tertentu.	Buku catatan
Statistik Zonal	Hitung statistik pada wilayah yang ditentukan pengguna.	Buku catatan
Pengambilan Sampel Ulang	Skala gambar ke resolusi yang berbeda.	Buku catatan

Operasi	Deskripsi	Ketersediaan
Geomosaik	Gabungkan beberapa gambar untuk kesetiaan yang lebih besar.	Buku catatan
Penumpukan Band	Gabungkan beberapa pita spektral untuk membuat satu gambar.	Buku catatan
Matematika Band/Indeks Spektral	Dapatkan kombinasi pita spektral yang menunjukkan banyaknya fitur yang menarik.	UI, Notebook
Segmentasi Tutupan Lahan	Mengidentifikasi jenis tutupan lahan seperti vegetasi dan air dalam citra satelit.	UI, Notebook

Lowongan Vector Enrichment

Vector Enrichment Job (VEJ) melakukan operasi pada data vektor Anda. Saat ini, Anda dapat menggunakan VEJ untuk melakukan geocoding terbalik atau pencocokan peta.

Geocoding Terbalik

Dengan VEJ geocoding terbalik, Anda dapat mengonversi koordinat geografis (lintang, bujur) menjadi alamat yang dapat dibaca manusia yang didukung oleh Amazon Location Service. Saat Anda mengunggah file CSV yang berisi koordinat bujur dan lintang, file tersebut mengembalikan nomor alamat, negara, label, kotamadya, lingkungan, kode pos, dan wilayah lokasi tersebut. File output terdiri dari data masukan Anda bersama dengan kolom yang berisi nilai-nilai yang ditambahkan di akhir. Pekerjaan ini dioptimalkan untuk menerima puluhan ribu jejak GPS.

Pencocokan Peta

Pencocokan peta memungkinkan Anda untuk mengambil koordinat GPS ke segmen jalan. Input harus berupa file CSV yang berisi ID jejak (rute), bujur, garis lintang, dan atribut stempel waktu. Mungkin ada beberapa koordinat GPS per rute. Input dapat berisi beberapa rute juga. Outputnya adalah file GeoJSON yang berisi tautan dari rute yang diprediksi. Ini juga memiliki titik jepret yang

disediakan dalam input. Pekerjaan ini dioptimalkan untuk menerima puluhan ribu drive dalam satu permintaan. Pencocokan peta didukung oleh OpenStreetMap. Pencocokan peta gagal jika nama di bidang sumber input tidak cocok dengan yang ada di`MapMatchingConfig`. Pesan galat yang Anda terima berisi nama bidang yang ada di file input dan nama bidang yang diharapkan yang tidak ditemukan`MapMatchingConfig`.

File CSV masukan untuk VEJ harus berisi yang berikut:

- Baris header
- Garis lintang dan bujur di kolom terpisah
- Kolom ID dan Timestamp dapat dalam format numerik atau string. Semua data kolom lainnya harus dalam format numerik saja
- Tidak ketinggalan kutipan yang cocok

Untuk kolom stempel waktu, kemampuan SageMaker geospasial mendukung waktu epoch dalam detik dan milidetik (bilangan bulat panjang). Format string yang didukung adalah sebagai berikut:

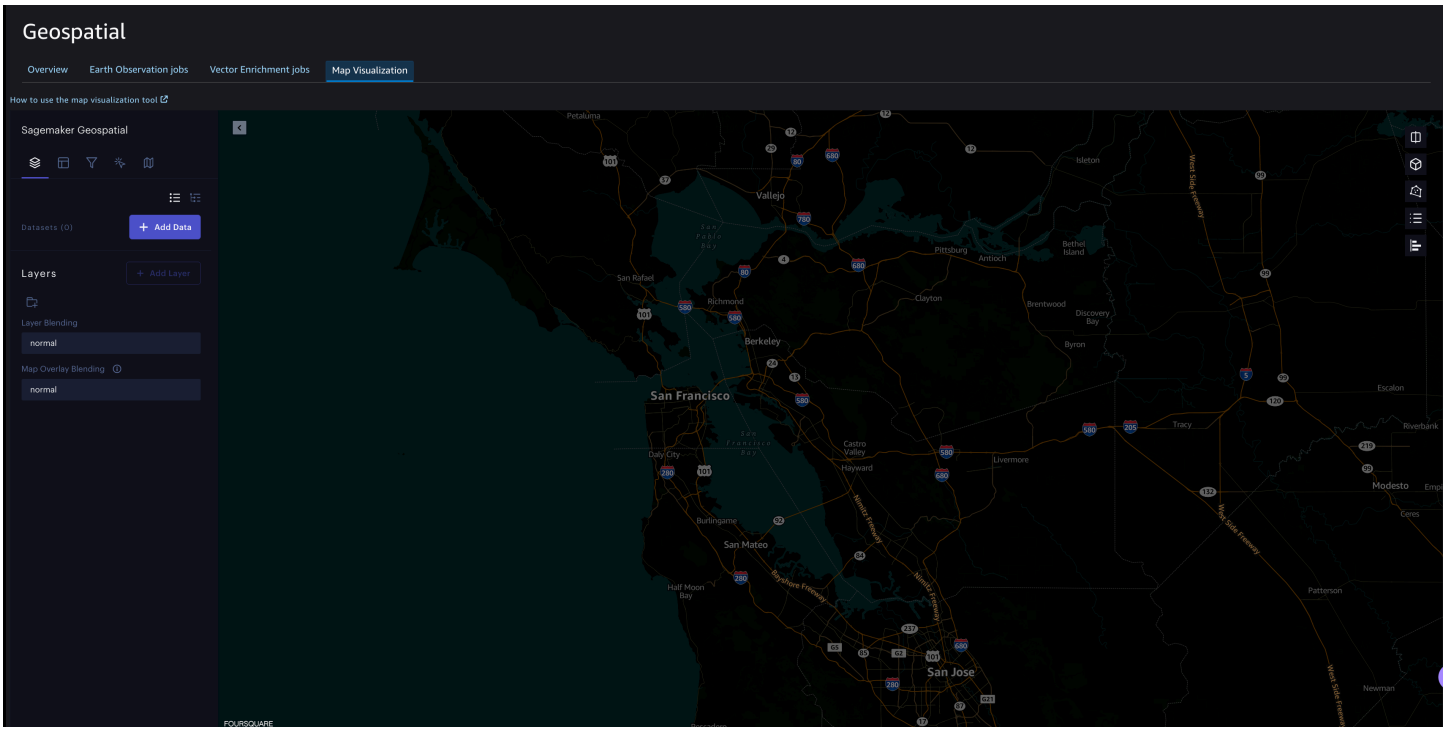
- “DD.mm.yyyy HH: mm: ss z”
- “Yyyy-mm-dd't'hh:mm: ss.sss'z '”
- “YYYY-MM-DD't'hh:mm: SS”
- “YYYY-MM-DD hh:mm: ss a”
- “YYYY-MM-DD HH: mm: SS”
- “YYYYMMDDhhmmss”

Meskipun Anda perlu menggunakan notebook Amazon SageMaker Studio Classic untuk menjalankan VEJ, Anda dapat melihat semua pekerjaan yang Anda buat menggunakan UI. Untuk menggunakan visualisasi di notebook, Anda harus terlebih dahulu mengekspor output Anda ke bucket S3 Anda. Tindakan VEJ yang dapat Anda lakukan adalah sebagai berikut.

- StartVectorEnrichmentJob
- GetVectorEnrichmentJob
- ListVectorEnrichmentJobs
- StopVectorEnrichmentJob
- DeleteVectorEnrichmentJob

Visualisasi Menggunakan kemampuan SageMaker geospasial

Dengan menggunakan fungsi visualisasi yang disediakan oleh SageMaker geospasial Amazon, Anda dapat memvisualisasikan data geospasial, input ke pekerjaan EOJ atau VEJ Anda serta output yang diekspor dari bucket Amazon S3 Anda. Alat visualisasi ini didukung oleh Foursquare Studio. Gambar berikut menggambarkan alat visualisasi yang didukung oleh kemampuan SageMaker geospasial.



Anda dapat menggunakan panel navigasi kiri untuk menambahkan data, lapisan, filter, dan kolom. Anda juga dapat membuat modifikasi pada bagaimana Anda berinteraksi dengan peta.

Dataset

Sumber data yang digunakan untuk visualisasi disebut Dataset. Untuk menambahkan data untuk visualisasi, pilih Tambahkan Data di panel navigasi kiri. Anda dapat mengunggah data dari bucket Amazon S3 atau mesin lokal Anda. Format data yang didukung adalah CSV, JSON dan GeoJSON. Anda dapat menambahkan beberapa kumpulan data ke peta Anda. Setelah Anda mengunggah kumpulan data, Anda dapat melihatnya dimuat di layar peta.

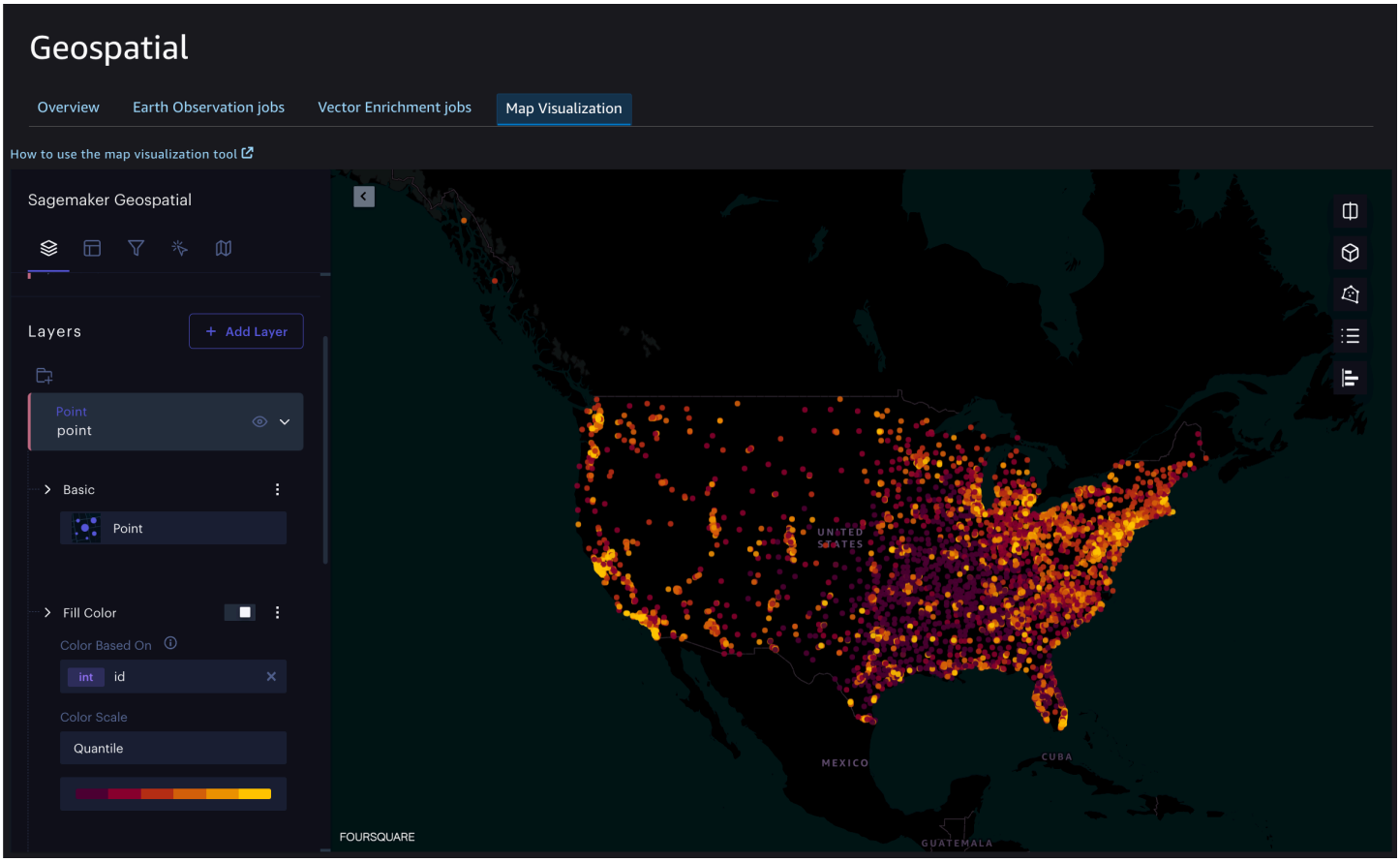
Lapisan

Di panel layer, layer dibuat dan diisi secara otomatis saat Anda menambahkan dataset. Jika peta Anda terdiri dari lebih dari satu kumpulan data, Anda dapat memilih kumpulan data mana yang termasuk dalam lapisan. Anda dapat membuat layer baru dan mengelompokkannya. SageMaker

SageMaker Kemampuan geospasial mendukung berbagai jenis lapisan, termasuk titik, busur, ikon, dan poligon.

Anda dapat memilih titik data apa pun di lapisan untuk memiliki Outline. Anda juga dapat menyesuaikan titik data lebih lanjut. Misalnya, Anda dapat memilih jenis layer sebagai Point dan kemudian Fill Color berdasarkan kolom mana pun dari dataset Anda. Anda juga dapat mengubah radius titik.

Gambar berikut menunjukkan panel lapisan yang didukung oleh kemampuan SageMaker geospasial.



Kolom

Anda dapat melihat kolom yang ada di kumpulan data Anda dengan menggunakan tab Kolom di panel navigasi kiri.

Filter

Anda dapat menggunakan filter untuk membatasi titik data yang ditampilkan di peta.

Interaksi

Di panel Interaksi, Anda dapat menyesuaikan cara Anda berinteraksi dengan peta. Misalnya, Anda dapat memilih metrik apa yang akan ditampilkan saat mengarahkan tooltip ke titik data.

Peta dasar

Saat ini, SageMaker hanya mendukung peta dasar Amazon Dark.

Mode Peta Pisahkan

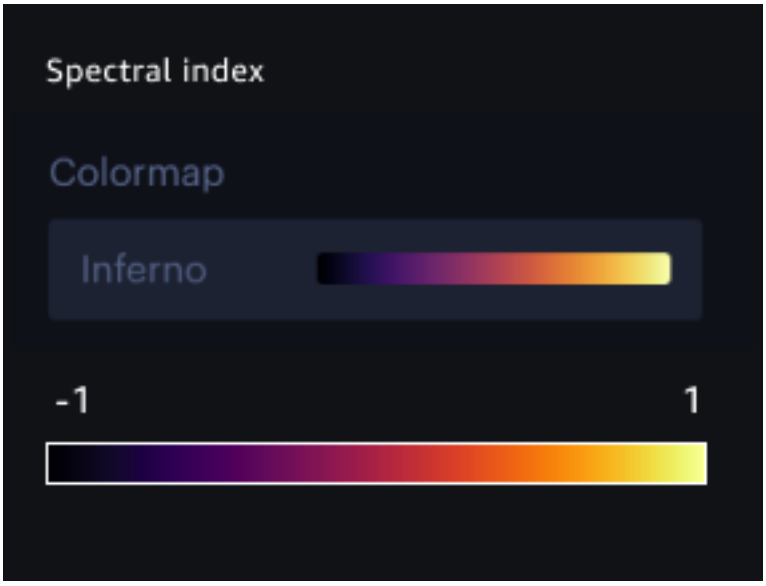
Anda dapat memiliki Peta Tunggal, Peta Ganda, atau Peta Gesek. Dengan Dual Maps, Anda dapat membandingkan peta yang sama side-by-side menggunakan lapisan yang berbeda. Gunakan Swipe Maps untuk melapisi dua peta satu sama lain dan gunakan pemisah geser untuk membandingkannya. Anda dapat memilih mode peta terpisah dengan memilih tombol Split Mode di sudut kanan atas peta Anda.

Legenda untuk EOJ di UI SageMaker geospasial

Visualisasi output dari EOJ tergantung pada operasi yang Anda pilih untuk membuatnya. Legenda didasarkan pada skala warna default. Anda dapat melihat legenda dengan memilih tombol Tampilkan legenda di sudut kanan atas peta Anda.

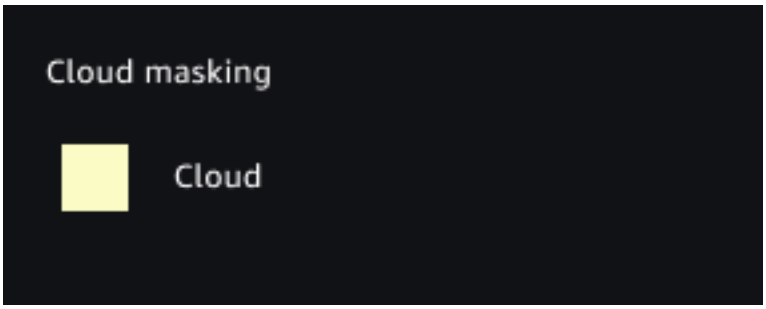
Indeks Spektral

Saat Anda memvisualisasikan output untuk EOJ yang menggunakan operasi indeks spektral, Anda dapat memetakan kategori berdasarkan warna dari legenda seperti yang ditunjukkan.



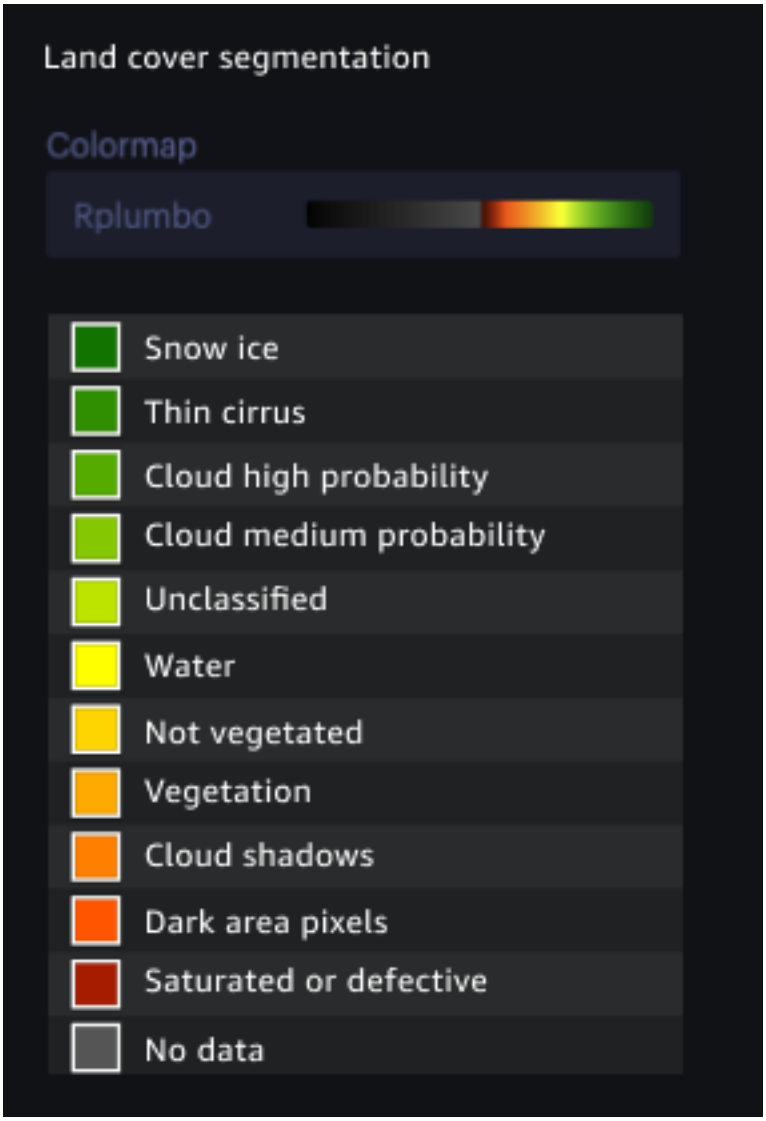
Penutupan Awan

Saat Anda memvisualisasikan output untuk EOJ yang menggunakan operasi cloud masking, Anda dapat memetakan kategori berdasarkan warna dari legenda seperti yang ditunjukkan.



Segmentasi Tutupan Lahan

Saat Anda memvisualisasikan output untuk EOJ yang menggunakan operasi Segmentasi Tutupan Tanah, Anda dapat memetakan kategori berdasarkan warna dari legenda seperti yang ditunjukkan.



SDK Peta SageMaker geospasial Amazon

Anda dapat menggunakan kemampuan SageMaker geospasial Amazon untuk memvisualisasikan peta dalam UI SageMaker geospasial serta SageMaker notebook dengan gambar geospasial. Visualisasi ini didukung oleh perpustakaan visualisasi peta yang disebut Foursquare Studio

Anda dapat menggunakan API yang disediakan oleh SDK peta SageMaker geospasial untuk memvisualisasikan data geospasial Anda, termasuk input, output, dan AoI untuk EOJ.

Topik

add_dataset API

Menambahkan objek dataset raster atau vektor ke peta.

Permintaan sintaks

```
Request =
    add_dataset(
      self,
      dataset: Union[Dataset, Dict, None] = None,
      *,
      auto_create_layers: bool = True,
      center_map: bool = True,
      **kwargs: Any,
    ) -> Optional[Dataset]
```

Parameter permintaan

Permintaan menerima parameter berikut.

Argumen posisi

Pendapat	Tipe	Deskripsi
`dataset`	Serikat [Dataset, Dict, Tidak Ada]	Data yang digunakan untuk membuat kumpulan data, dalam format CSV, JSON, atau GeoJSON (untuk kumpulan data lokal) atau string UUID.

Argumen kata kunci

Pendapat	Tipe	Deskripsi
`auto_create_layers`	Boolean	Apakah akan mencoba membuat lapisan baru saat menambahkan kumpulan data. Nilai default-nya adalah `False`.
`center_map`	Boolean	Apakah akan memusatkan peta pada kumpulan data yang dibuat. Nilai default-nya adalah `True`.
`id`	String	Pengidentifikasi unik dari kumpulan data. Jika Anda tidak memberikannya, ID acak dihasilkan.
`label`	String	Label dataset yang ditampilkan.
`color`	Tupel [mengapung, mengapung, mengapung]	Label warna dari kumpulan data.

Pendapat	Tipe	Deskripsi
`metadata`	Kamus	Objek yang berisi metadata tileset (untuk kumpulan data ubin).

Respons

API ini mengembalikan objek Dataset yang ditambahkan ke peta.

update_dataset API

Memperbarui pengaturan kumpulan data yang ada.

Permintaan sintaks

```
Request =
    update_dataset(
    self,
    dataset_id: str,
    values: Union[_DatasetUpdateProps, dict, None] = None,
    **kwargs: Any,
) -> Dataset
```

Parameter permintaan

Permintaan menerima parameter berikut.

Argumen posisi

Pendapat	Tipe	Deskripsi
`dataset_id`	String	Pengidentifikasi dataset yang akan diperbarui.
`values`	Serikat [_ DatasetUpdateProps, dict, Tidak ada]	Nilai untuk memperbarui.

Argumen kata kunci

Pendapat	Tipe	Deskripsi
`label`	String	Label dataset yang ditampilkan.
`color`	RGBColor	Label warna dari kumpulan data.

Respons

API ini mengembalikan objek dataset yang diperbarui untuk peta interaktif, atau `None` untuk lingkungan HTML non-interaktif.

add_layer API

Menambahkan layer baru ke peta. Fungsi ini membutuhkan setidaknya satu konfigurasi lapisan yang valid.

Permintaan sintaks

```
Request =
    add_layer(
    self,
    layer: Union[LayerCreationProps, dict, None] = None,
    **kwargs: Any
) -> Layer
```

Parameter permintaan

Permintaan menerima parameter berikut.

Argumen

Pendapat	Tipe	Deskripsi
`layer`	Serikat [LayerCreationProps, dict, Tidak ada]	Satu set properti yang digunakan untuk membuat layer.

Respons

Objek layer yang ditambahkan ke peta.

update_layer API

Perbarui lapisan yang ada dengan nilai yang diberikan.

Minta sintaks

```
Request =
    update_layer(
  self,
  layer_id: str,
  values: Union[LayerUpdateProps, dict, None],
  **kwargs: Any
) -> Layer
```

Parameter permintaan

Permintaan menerima parameter berikut.

Argumen

Argumen posisi	Tipe	Deskripsi
`layer_id`	String	ID dari layer yang akan diperbarui.
`values`	Serikat [LayerUpdateProps, dict, Tidak ada]	Nilai untuk memperbarui.

Argumen kata kunci

Pendapat	Tipe	Deskripsi
`type`	LayerType	Jenis lapisan.

Pendapat	Tipe	Deskripsi
`data_id`	String	Pengidentifikasi unik dari kumpulan data yang divisualisasikan oleh lapisan ini.
`fields`	Dict [string, Opsional [string]]	Kamus yang memetakan bidang yang diperlukan lapisan untuk visualisasi ke bidang kumpulan data yang sesuai.
`label`	String	Label kanonik dari lapisan ini.
`is_visible`	Boolean	Apakah lapisan terlihat atau tidak.
`config`	LayerConfig	Konfigurasi lapisan khusus untuk tipenya.

Respons

Mengembalikan objek layer diperbarui.

visualize_eoj_aoi API

Visualisasikan AoI dari pekerjaan yang diberikan ARN.

Parameter permintaan

Permintaan menerima parameter berikut.

Argumen

Pendapat	Tipe	Deskripsi
`Arn`	String	ARN dari pekerjaan itu.
`config`	Kamus	Opsi untuk melewatkan properti lapisan.

Pendapat	Tipe	Deskripsi
	config = {label: <string>label khusus dari lapisan AoI yang ditambahkan, default AoI}	

Respons

Referensi dari objek lapisan masukan yang ditambahkan.

visualize_eoj_input API

Visualisasikan masukan dari EOJ ARN yang diberikan.

Parameter permintaan

Permintaan menerima parameter berikut.

Argumen

Pendapat	Tipe	Deskripsi
`Arn`	String	ARN dari pekerjaan itu.

`time_range_filter`	Kamus
	time_range_filter = {
	start_date: <string>tanggal dalam format ISO
	end_date: <string>tanggal dalam format ISO
	}
		Opsi untuk menyediakan waktu mulai dan berakhir. Default untuk pencarian pengumpulan data raster tanggal mulai dan berakhir.

`config`	Kamus config = {label: <string>label khusus dari lapisan keluaran	Opsi untuk melewatkan properti lapisan.

Pendapat	Tipe	Deskripsi
	yang ditambahkan, Input default}	

Respons

Referensi dari objek lapisan masukan yang ditambahkan.

visualisasika_eoj_output API

Visualisasikan output dari EOJ ARN yang diberikan.

Parameter permintaan

Permintaan menerima parameter berikut.

Argumen

Pendapat	Tipe	Deskripsi
`Arn`	String	ARN dari pekerjaan itu.

`time_range_filter`	Kamus
	time_range_filter = {
	start_date: <string>tanggal dalam format ISO
	end_date: <string>tanggal dalam format ISO
	}
		Opsi untuk menyediakan waktu mulai dan berakhir. Default untuk pencarian pengumpulan data raster tanggal mulai dan berakhir.

`config`	Kamus konfigurasi = {	Opsi untuk melewatkan properti lapisan.

Pendapat	Tipe	Deskripsi
	label: <string>label khusus dari lapisan keluaran yang ditambahkan, Output default preset: <string>SingleBand atau TrueColor, band_name:<string>, hanya diperlukan untuk preset 'SingleBand'. Band yang diizinkan untuk EOJ }	

Respons

Referensi dari objek Layer keluaran yang ditambahkan.

Untuk mempelajari lebih lanjut tentang memvisualisasikan data geospasial Anda, lihat Visualisasi Menggunakan geospasial Amazon. SageMaker

SageMaker FAQ kemampuan geospasial

Gunakan item FAQ berikut untuk menemukan jawaban atas pertanyaan umum tentang kemampuan SageMaker geospasial.

1. Di wilayah mana kemampuan SageMaker geospasial Amazon tersedia?

   Saat ini, kemampuan SageMaker geospasial hanya didukung di Wilayah Barat AS (Oregon). Untuk melihat SageMaker geospasial, pilih nama Wilayah yang saat ini ditampilkan di bilah navigasi konsol. Kemudian pilih Wilayah Barat AS (Oregon).

2. AWS Identity and Access ManagementIzin dan kebijakan apa yang diperlukan untuk menggunakan SageMaker geospasial?

   Untuk menggunakan SageMaker geospasial, Anda memerlukan pengguna, grup, atau peran yang dapat diakses SageMaker. Anda juga perlu membuat peran SageMaker eksekusi sehingga

SageMaker geospasial dapat melakukan operasi atas nama Anda. Untuk mempelajari lebih lanjut, lihat peran kemampuan SageMaker geospasial.

3. Saya memiliki peran SageMaker eksekusi yang ada. Apakah saya perlu memperbaruinya?

   Ya. Untuk menggunakan SageMaker geospasial, Anda harus menentukan prinsip layanan tambahan dalam kebijakan kepercayaan IAM Anda:. `sagemaker-geospatial.amazonaws.com` Untuk mempelajari tentang menentukan prinsip layanan dalam hubungan kepercayaan, lihat Menambahkan prinsip layanan SageMaker geospasial ke peran SageMaker eksekusi yang ada di Panduan SageMaker Pengembang Amazon.

4. Dapatkah saya menggunakan kemampuan SageMaker geospasial melalui lingkungan VPC saya?

   Ya, Anda dapat menggunakan SageMaker geospasial melalui VPN. Untuk mempelajari selengkapnya, lihat Gunakan kemampuan SageMaker geospasial Amazon di Amazon Virtual Private Cloud.

5. Mengapa saya tidak dapat melihat visualizer peta SageMaker geospasial, gambar, atau jenis instance saat saya menavigasi ke Amazon SageMaker Studio Classic?

   Verifikasi bahwa Anda meluncurkan Amazon SageMaker Studio Classic di Wilayah AS Barat (Oregon) dan bahwa Anda tidak menggunakan ruang bersama.

6. Mengapa saya tidak dapat melihat gambar SageMaker geospasial atau tipe instance ketika saya mencoba membuat instance notebook di Studio Classic?

   Verifikasi bahwa Anda meluncurkan Amazon SageMaker Studio Classic di Wilayah AS Barat (Oregon) dan bahwa Anda tidak menggunakan ruang bersama. Untuk mempelajari selengkapnya, lihat Membuat notebook Amazon SageMaker Studio Classic menggunakan gambar geospasial.

7. Band apa yang didukung untuk berbagai pengumpulan data raster?

   Gunakan respons `GetRasterDataCollection` API dan lihat `ImageSourceBands` bidang untuk menemukan band yang didukung untuk pengumpulan data tertentu.

SageMaker Keamanan dan Izin geospasial

Gunakan topik di halaman ini untuk mempelajari tentang fitur keamanan kemampuan SageMaker geospasial. Selain itu, pelajari cara menggunakan kemampuan SageMaker geospasial di Amazon Virtual Private Cloud serta melindungi data Anda saat istirahat menggunakan enkripsi.

Untuk informasi selengkapnya tentang pengguna dan peran IAM, lihat Identitas (Pengguna, Grup, dan Peran) di Panduan Pengguna IAM.

Untuk mempelajari lebih lanjut tentang menggunakan IAM dengan SageMaker, lihatIdentity and Access Management untuk Amazon SageMaker.

Topik

- Analisis Konfigurasi dan Kerentanan dalam SageMaker geospasial
- Praktik Terbaik Keamanan untuk kemampuan SageMaker geospasial
- Gunakan kemampuan SageMaker geospasial Amazon di Amazon Virtual Private Cloud
- Gunakan AWS KMS Izin untuk kemampuan SageMaker geospasial Amazon

Analisis Konfigurasi dan Kerentanan dalam SageMaker geospasial

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara AWS dan Anda, pelanggan kami. AWSmenangani tugas-tugas keamanan dasar seperti sistem operasi tamu (OS) dan patching database, konfigurasi firewall, dan pemulihan bencana. Prosedur ini telah ditinjau dan disertifikasi oleh pihak ketiga yang sesuai. Untuk detail selengkapnya, lihat sumber daya berikut:

- Model Tanggung Jawab Bersama.
- Amazon Web Services: Ikhtisar Proses Keamanan.

Praktik Terbaik Keamanan untuk kemampuan SageMaker geospasial

Kemampuan SageMaker geospasial Amazon menyediakan sejumlah fitur keamanan untuk dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau cukup untuk lingkungan Anda, anggap sebagai pertimbangan yang membantu dan bukan sebagai resep.

Terapkan prinsip hak istimewa paling rendah

Kemampuan SageMaker geospasial Amazon menyediakan kebijakan akses terperinci untuk aplikasi yang menggunakan peran IAM. Kami merekomendasikan bahwa peran hanya diberikan set minimum hak istimewa yang diperlukan oleh pekerjaan. Kami juga merekomendasikan untuk mengaudit pekerjaan untuk izin secara teratur dan setiap perubahan pada aplikasi Anda.

Izin kontrol akses berbasis peran (RBAC)

Administrator harus secara ketat mengontrol izin kontrol akses berbasis peran (RBAC) untuk kemampuan geospasial Amazon. SageMaker

Gunakan kredensi sementara bila memungkinkan

Jika memungkinkan, gunakan kredensil sementara alih-alih kredensil jangka panjang, seperti kunci akses. Untuk skenario di mana Anda memerlukan pengguna IAM dengan akses terprogram dan kredensyal jangka panjang, kami sarankan Anda memutar kunci akses. Memutar kredensyal jangka panjang secara teratur membantu Anda membiasakan diri dengan prosesnya. Ini berguna jika Anda pernah berada dalam situasi di mana Anda harus memutar kredensyal, seperti ketika seorang karyawan meninggalkan perusahaan Anda. Kami menyarankan Anda menggunakan akses IAM informasi yang terakhir digunakan untuk memutar dan menghapus kunci akses dengan aman. Untuk informasi selengkapnya, lihat Memutar kunci akses dan Praktik terbaik keamanan di IAM.

Gunakan AWS CloudTrail untuk melihat dan mencatat panggilan API

AWS CloudTrailmelacak siapa pun yang melakukan panggilan API di AWS akun Anda. Panggilan API dicatat setiap kali ada yang menggunakan API kemampuan SageMaker geospasial Amazon, konsol kemampuan SageMaker geospasial Amazon, atau perintah CLI SageMaker AWS kemampuan geospasial Amazon. Aktifkan logging dan tentukan bucket Amazon S3 untuk menyimpan log.

Kepercayaan, privasi, dan keamanan konten Anda adalah prioritas tertinggi kami. Kami menerapkan kontrol teknis dan fisik yang bertanggung jawab dan canggih yang dirancang untuk mencegah akses tidak sah ke, atau pengungkapan, konten Anda dan memastikan bahwa penggunaan kami sesuai dengan komitmen kami kepada Anda. Untuk informasi selengkapnya, lihat FAQ Privasi AWS Data.

Gunakan kemampuan SageMaker geospasial Amazon di Amazon Virtual Private Cloud

Topik berikut memberikan informasi tentang cara menggunakan SageMaker notebook dengan gambar SageMaker geospasial di SageMaker Domain Amazon dengan mode VPC saja. Untuk informasi selengkapnya tentang VPC di Amazon SageMaker Studio Classic, lihat Memilih VPC Amazon.

`VPC only`komunikasi dengan internet

Secara default, SageMaker Domain menggunakan dua Amazon VPC. Salah satu VPC Amazon dikelola oleh Amazon SageMaker dan menyediakan akses internet langsung. Anda menentukan VPC

Amazon lainnya, yang menyediakan lalu lintas terenkripsi antara Domain dan volume Amazon Elastic File System (Amazon EFS) Anda.

Anda dapat mengubah perilaku ini sehingga SageMaker mengirimkan semua lalu lintas melalui VPC Amazon yang Anda tentukan. Jika `VPC only` telah dipilih sebagai mode akses jaringan selama pembuatan SageMaker Domain, persyaratan berikut harus dipertimbangkan untuk tetap mengizinkan penggunaan notebook SageMaker Studio Classic dalam Domain yang dibuat SageMaker .

Persyaratan untuk menggunakan **`VPC only`** mode

> **Note**
>
> Untuk menggunakan komponen visualisasi kemampuan SageMaker geospasial, browser yang Anda gunakan untuk mengakses UI SageMaker Studio Classic harus terhubung ke internet.

Saat Anda memilih`VpcOnly`, ikuti langkah-langkah ini:

1. Anda harus menggunakan subnet pribadi saja. Anda tidak dapat menggunakan subnet publik dalam `VpcOnly` mode.
2. Pastikan subnet Anda memiliki jumlah alamat IP yang diperlukan. Jumlah yang diharapkan dari alamat IP yang dibutuhkan per pengguna dapat bervariasi berdasarkan kasus penggunaan. Kami merekomendasikan antara 2 dan 4 alamat IP per pengguna. Total kapasitas alamat IP untuk domain Studio Classic adalah jumlah alamat IP yang tersedia untuk setiap subnet yang disediakan saat domain dibuat. Pastikan perkiraan penggunaan alamat IP Anda tidak melebihi kapasitas yang didukung oleh jumlah subnet yang Anda berikan. Selain itu, menggunakan subnet yang didistribusikan di banyak zona ketersediaan dapat membantu ketersediaan alamat IP. Untuk informasi selengkapnya, lihat ukuran VPC dan subnet untuk IPv4.

   > **Note**
   >
   > Anda hanya dapat mengonfigurasi subnet dengan VPC penyewaan default di mana instans Anda berjalan pada perangkat keras bersama. Untuk informasi selengkapnya tentang atribut tenancy untuk VPC, lihat Instans Khusus.

3. Siapkan satu atau beberapa grup keamanan dengan aturan masuk dan keluar yang bersama-sama memungkinkan lalu lintas berikut:

- Lalu lintas NFS melalui TCP pada port 2049 antara domain dan volume Amazon EFS.
- Lalu lintas TCP dalam grup keamanan. Ini diperlukan untuk konektivitas antara JupyterServer aplikasi dan KernelGateway aplikasi. Anda harus mengizinkan akses ke setidaknya port dalam jangkauan`8192-65535`.

4. Jika Anda ingin mengizinkan akses internet, Anda harus menggunakan gateway NAT dengan akses ke internet, misalnya melalui gateway internet.
5. Jika Anda tidak ingin mengizinkan akses internet, buat antarmuka VPC endpoint (AWS PrivateLink) untuk memungkinkan Studio Classic mengakses layanan berikut dengan nama layanan yang sesuai. Anda juga harus mengaitkan grup keamanan untuk VPC Anda dengan titik akhir ini.

> **Note**
>
> Saat ini, kemampuan SageMaker geospasial hanya didukung di Wilayah Barat AS (Oregon).

- SageMaker API: `com.amazonaws.us-west-2.sagemaker.api`
- SageMaker runtime:`com.amazonaws.us-west-2.sagemaker.runtime`. Ini diperlukan untuk menjalankan notebook Studio Classic dengan gambar SageMaker geospasial.
- Amazon S3:. `com.amazonaws.us-west-2.s3`
- Untuk menggunakan SageMaker Proyek:`com.amazonaws.us-west-2.servicecatalog`.
- SageMaker kemampuan geospasial: `com.amazonaws.us-west-2.sagemaker-geospatial`

Jika Anda menggunakan SageMaker Python SDK untuk menjalankan pekerjaan pelatihan jarak jauh, Anda juga harus membuat endpoint Amazon VPC berikut.

- AWS Security Token Service: `com.amazonaws.`*`region`*`.sts`
- Amazon CloudWatch:`com.amazonaws.`*`region`*`.logs`. Ini diperlukan untuk memungkinkan SageMaker Python SDK mendapatkan status pekerjaan pelatihan jarak jauh. Amazon CloudWatch

> **Note**
>
> Untuk pelanggan yang bekerja dalam mode VPC, firewall perusahaan dapat menyebabkan masalah koneksi dengan SageMaker Studio Classic atau antara JupyterServer dan. KernelGateway Lakukan pemeriksaan berikut jika Anda mengalami salah satu masalah ini saat menggunakan SageMaker Studio Classic dari belakang firewall.
>
> - Periksa apakah URL Studio Classic ada di daftar yang diizinkan jaringan Anda.
> - Periksa apakah koneksi websocket tidak diblokir. Jupyter menggunakan websocket di bawah tenda. Jika KernelGateway aplikasi ini InService, JupyterServer mungkin tidak dapat terhubung ke KernelGateway. Anda akan melihat masalah ini saat membuka Terminal Sistem juga.

Gunakan AWS KMS Izin untuk kemampuan SageMaker geospasial Amazon

Anda dapat melindungi data Anda saat istirahat menggunakan enkripsi untuk kemampuan SageMaker geospasial. Secara default, ia menggunakan enkripsi sisi server dengan kunci milik SageMaker geospasial Amazon. SageMaker Kemampuan geospasial juga mendukung opsi untuk enkripsi sisi server dengan kunci KMS yang dikelola pelanggan.

Enkripsi Sisi Server dengan kunci terkelola SageMaker geospasial Amazon (Default)

SageMaker Kemampuan geospasial mengenkripsi semua data Anda, termasuk hasil komputasi dari pekerjaan Pengamatan Bumi (EOJ) dan pekerjaan Pengayaan Vektor (VEJ) bersama dengan semua metadata layanan Anda. Tidak ada data yang disimpan dalam kemampuan SageMaker geospasial yang tidak terenkripsi. Ini menggunakan kunci yang AWS dimiliki default untuk mengenkripsi semua data Anda.

Enkripsi Sisi Server dengan kunci KMS yang dikelola pelanggan (Opsional)

SageMaker Kemampuan geospasial mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola untuk menambahkan lapisan enkripsi kedua di atas enkripsi AWS milik yang ada. Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti:

- Menetapkan dan memelihara kebijakan utama
- Menetapkan dan memelihara kebijakan dan hibah IAM

- Mengaktifkan dan menonaktifkan kebijakan utama
- Memutar bahan kriptografi kunci
- Menambahkan tanda
- Membuat alias kunci
- Kunci penjadwalan untuk penghapusan

Untuk informasi selengkapnya, lihat Kunci terkelola pelanggan di Panduan AWS Key Management Service Pengembang.

Bagaimana kemampuan SageMaker geospasial menggunakan hibah di AWS KMS

SageMaker Kemampuan geospasial memerlukan hibah untuk menggunakan kunci yang dikelola pelanggan Anda. Saat Anda membuat EOJ atau VEJ yang dienkripsi dengan kunci yang dikelola pelanggan, kemampuan SageMaker geospasial menciptakan hibah atas nama Anda dengan mengirimkan permintaan ke. `CreateGrant` AWS KMS Hibah AWS KMS digunakan untuk memberikan akses kemampuan SageMaker geospasial ke kunci KMS di akun pelanggan. Anda dapat mencabut akses ke hibah, atau menghapus akses layanan ke kunci yang dikelola pelanggan kapan saja. Jika Anda melakukannya, kemampuan SageMaker geospasial tidak akan dapat mengakses data apa pun yang dienkripsi oleh kunci yang dikelola pelanggan, yang memengaruhi operasi yang bergantung pada data tersebut.

Buat kunci terkelola pelanggan

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakan AWS Management Console, atau AWS KMS API.

Untuk membuat kunci terkelola pelanggan simetris

Ikuti langkah-langkah untuk Membuat kunci KMS enkripsi simetris di Panduan AWS Key Management Service Pengembang.

Kebijakan utama

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi

selengkapnya, lihat Menentukan akses ke AWS KMS kunci di Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci terkelola pelanggan dengan sumber daya kemampuan SageMaker geospasial Anda, operasi API berikut harus diizinkan dalam kebijakan utama. Prinsip untuk operasi ini harus Peran Eksekusi yang Anda berikan dalam permintaan kemampuan SageMaker geospasial. SageMaker kemampuan geospasial mengasumsikan Peran Eksekusi yang disediakan dalam permintaan untuk melakukan operasi KMS ini.

- `kms:CreateGrant`
- `kms:GenerateDataKey`
- `kms:Decrypt`
- `kms:GenerateDataKeyWithoutPlaintext`

Berikut ini adalah contoh pernyataan kebijakan yang dapat Anda tambahkan untuk kemampuan SageMaker geospasial:

CreateGrant

```
"Statement" : [
    {
      "Sid" : "Allow access to Amazon SageMaker geospatial capabilities",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "<Customer provided Execution Role ARN>"
      },
      "Action" : [
          "kms:CreateGrant",
           "kms:Decrypt",
           "kms:GenerateDataKey",
           "kms:GenerateDataKeyWithoutPlaintext"
      ],
      "Resource" : "*",
    },
 ]
```

Untuk informasi selengkapnya tentang menentukan izin dalam kebijakan, lihat AWS KMSizin di Panduan Pengembang. AWS Key Management Service Untuk informasi selengkapnya tentang pemecahan masalah, lihat Memecahkan masalah akses kunci di Panduan Pengembang. AWS Key Management Service

Jika kebijakan kunci Anda tidak memiliki root akun sebagai administrator kunci, Anda perlu menambahkan izin KMS yang sama pada ARN peran eksekusi Anda. Berikut adalah contoh kebijakan yang dapat Anda tambahkan ke peran eksekusi:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "kms:CreateGrant",
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": [
              "<KMS key Arn>"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Memantau kunci enkripsi Anda untuk kemampuan SageMaker geospasial

Bila Anda menggunakan kunci terkelola AWS KMS pelanggan dengan sumber daya kemampuan SageMaker geospasial, Anda dapat menggunakan AWS CloudTrail atau Amazon CloudWatch Logs untuk melacak permintaan yang dikirimkan SageMaker geospasial. AWS KMS

Pilih tab di tabel berikut untuk melihat contoh AWS CloudTrail peristiwa untuk memantau operasi KMS yang dipanggil oleh kemampuan SageMaker geospasial untuk mengakses data yang dienkripsi oleh kunci terkelola pelanggan Anda.

CreateGrant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SageMaker-Geospatial-StartEOJ-
KMSAccess",
        "arn": "arn:aws:sts::111122223333:assumed-role/
SageMakerGeospatialCustomerRole/SageMaker-Geospatial-StartEOJ-KMSAccess",
```

```
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE3",
                "arn": "arn:aws:sts::111122223333:assumed-role/
SageMakerGeospatialCustomerRole",
                "accountId": "111122223333",
                "userName": "SageMakerGeospatialCustomerRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-17T18:02:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "arn:aws:iam::111122223333:root"
    },
    "eventTime": "2023-03-17T18:02:06Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "retiringPrincipal": "sagemaker-geospatial.us-west-2.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "operations": [
            "Decrypt"
        ],
        "granteePrincipal": "sagemaker-geospatial.us-west-2.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
```

```
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateDataKey

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "sagemaker-geospatial.amazonaws.com"
    },
    "eventTime": "2023-03-24T00:29:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "sagemaker-geospatial.amazonaws.com",
    "userAgent": "sagemaker-geospatial.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:s3:arn": "arn:aws:s3:::axis-earth-observation-
job-378778860802/111122223333/napy9eintp64/output/
consolidated/32PPR/2022-01-04T09:58:03Z/S2B_32PPR_20220104_0_L2A_msavi.tif"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "keySpec": "AES_256"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
```

```
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "sagemaker-geospatial.amazonaws.com"
    },
    "eventTime": "2023-03-28T22:04:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "sagemaker-geospatial.amazonaws.com",
    "userAgent": "sagemaker-geospatial.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "aws:s3:arn": "arn:aws:s3:::axis-earth-observation-
job-378778860802/111122223333/napy9eintp64/output/
consolidated/32PPR/2022-01-04T09:58:03Z/S2B_32PPR_20220104_0_L2A_msavi.tif"
        },
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateDataKeyWithoutPlainText

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SageMaker-Geospatial-StartEOJ-
KMSAccess",
        "arn": "arn:aws:sts::111122223333:assumed-role/
SageMakerGeospatialCustomerRole/SageMaker-Geospatial-StartEOJ-KMSAccess",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE3",
                "arn": "arn:aws:sts::111122223333:assumed-role/
SageMakerGeospatialCustomerRole",
                "accountId": "111122223333",
                "userName": "SageMakerGeospatialCustomerRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-17T18:02:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "arn:aws:iam::111122223333:root"
    },
    "eventTime": "2023-03-28T22:09:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-west-2",
```

```
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Jenis instance komputasi

SageMaker Kemampuan geospasial menawarkan tiga jenis contoh komputasi.

- SageMaker Instans notebook geospasial Studio Classic — SageMaker geospasial mendukung instans notebook berbasis CPU dan GPU di Studio Classic. Instance notebook digunakan untuk membangun, melatih, dan menerapkan model ML. Untuk daftar jenis instans notebook yang tersedia yang berfungsi dengan gambar geospasial, lihat Jenis instans notebook yang didukung.
- SageMaker contoh pekerjaan geospasial - Jalankan pekerjaan pemrosesan untuk mengubah data citra satelit.
- SageMaker jenis inferensi model geospasial — Buat prediksi dengan menggunakan model ML yang telah dilatih sebelumnya pada citra satelit.

Jenis instans ditentukan oleh operasi yang Anda jalankan.

Tabel berikut menunjukkan operasi spesifik SageMaker geospasial yang tersedia dan jenis instance yang dapat Anda gunakan.

Operasi	Instans
Statistik Temporal	ml.geospatial.jobs
Statistik Zonal	ml.geospatial.jobs
Pengambilan Sampel Ulang	ml.geospatial.jobs
Geomosaik	ml.geospatial.jobs
Penumpukan Band	ml.geospatial.jobs
Band Matematika	ml.geospatial.jobs
Penghapusan Cloud dengan Landsat8	ml.geospatial.jobs
Penghapusan Cloud dengan Sentinel-2	ml.geospatial.models
Penutupan Awan	ml.geospatial.models
Segmentasi Tutupan Lahan	ml.geospatial.models

SageMaker jenis instance notebook yang didukung geospasial

SageMaker geospasial mendukung instance notebook berbasis CPU dan GPU di Studio Classic. Jika saat memulai instance notebook yang diaktifkan GPU Anda menerima ResourceLimitExceededkesalahan, Anda perlu meminta peningkatan kuota. Untuk memulai permintaan peningkatan kuota Service Quotas, lihat Meminta peningkatan kuota pada Panduan Pengguna Service Quotas.

Jenis instans notebook Studio Classic yang didukung

Nama	Jenis instans
ml.geospatial.interactive	CPU
ml.g5.xlarge	GPU

Nama	Jenis instans
ml.g5.2xbesar	GPU
ml.g5.4xbesar	GPU
ml.g5.8xbesar	GPU
ml.g5.16xbesar	GPU
ml.g5.12xbesar	GPU
ml.g5.24xbesar	GPU
ml.g5.48xbesar	GPU

Anda dikenakan tarif yang berbeda untuk setiap jenis instans komputasi yang Anda gunakan. Untuk informasi selengkapnya tentang harga, lihat Geospasial MS dengan Amazon SageMaker.

SageMaker perpustakaan geospasial

Jenis Instance khusus SageMaker geospasial, **`ml.geospatial.interactive`** berisi pustaka Python berikut.

Pustaka geospasial tersedia pada jenis instance geospasial

Nama perpustakaan	Versi tersedia
numpy	1.23.4
scipy	1.11.2
panda	1.4.4
gdal	3.2.2
fiona	1.8.22
geopanda	0.11.1
indah	1.8.4

Nama perpustakaan	Versi tersedia
seaborn	0.11.2
buku catatan	1.8.22
gambar scikit	0.11.2
rasterio	6.4.12
scikit-belajar	0.19.2
ipyleaflet	1.0.1
rpohon	0.17.2
opencv	4.6.0.66
supy	2022.4.7
Kotak peralatan SNAP	9.0
cdsapi	0.6.1
arosik	1.8.1
rasterstats	0.18.0
rioxarray	0.14.1
Pyrosar	0.20.0
eo-belajar	1.4.1
deepforest	1.2.7
gesek	2.8.0
NetCDF4	1.6.3
xarray [lengkap]	0.20.1

Nama perpustakaan	Versi tersedia
Orfeotoolbox	OTB-8.1.1
pytorch	2.0.1
pytorch-cuda	11.8
torchvision	0.15.2
torchaudio	2.0.2
pytorch-petir	2.0.6
tensorflow	2.13.0

Pengumpulan data

Amazon SageMaker geospasial mendukung pengumpulan data raster berikut. Dari pengumpulan data berikut, Anda dapat menggunakan USGS Landsat dan pengumpulan data yang Sentinel-2 Dioptimalkan di Cloud GeoTIFF saat memulai Earth Observation Job (EOJ). Untuk mempelajari lebih lanjut tentang EOJ, lihatPekerjaan Pengamatan Bumi.

- Copernicus Digital Elevation Model (DEM)— GLO-30
- Copernicus Digital Elevation Model (DEM)— GLO-90
- Sentinel-2 Cloud-Optimized GeoTIFFs
- Sentinel-1
- National Agriculture Imagery Program (NAIP) pada AWS
- USGS Landsat 8

Untuk menemukan daftar koleksi data raster yang tersedia di AndaWilayah AWS, gunakan`ListRasterDataCollections`. `ListRasterDataCollections`Sebagai tanggapan, Anda mendapatkan `RasterDataCollectionMetadata`objek yang berisi detail tentang koleksi data raster yang tersedia.

Example Contoh - Memanggil **`ListRasterDataCollections`** API menggunakan AWS SDK for Python (Boto3)

Bila Anda menggunakan SDK untuk Python (SageMaker Boto3) dan geospasial, Anda harus membuat klien geospasial,. `geospatial_client` Gunakan Python cuplikan berikut untuk melakukan panggilan ke API: `list_raster_data_collections`

```
import boto3
import sagemaker
import sagemaker_geospatial_map
import json

## SageMaker Geospatial Capabilities is currently only avaialable in US-WEST-2
session = boto3.Session(region_name='us-west-2')
execution_role = sagemaker.get_execution_role()

## Creates a SageMaker Geospatial client instance
geospatial_client = session.client(service_name="sagemaker-geospatial")

# Creates a resusable Paginator for the list_raster_data_collections API operation
paginator = geospatial_client.get_paginator("list_raster_data_collections")

# Create a PageIterator from the Paginator
page_iterator = paginator.paginate()

# Use the iterator to iterate throught the results of list_raster_data_collections
results = []
for page in page_iterator:
 results.append(page['RasterDataCollectionSummaries'])

print (results)
```

Dalam respons JSON, Anda akan menerima yang berikut ini, yang telah dipotong untuk kejelasan:

```
{
    "Arn": "arn:aws:sagemaker-geospatial:us-west-2:555555555555:raster-data-collection/
public/dxxbpqwvu9041ny8",
    "Description": "Copernicus DEM is a Digital Surface Model which represents the
 surface of the Earth including buildings, infrastructure, and vegetation. GLO-30 is
 instance of Copernicus DEM that provides limited worldwide coverage at 30 meters.",
    "DescriptionPageUrl": "https://registry.opendata.aws/copernicus-dem/",
    "Name": "Copernicus DEM GLO-30",
    "Tags": {},
```

```
    "Type": "PUBLIC"
}
```

Informasi pita gambar dari USGS Landsat dan pengumpulan Sentinel-2 data

Informasi pita gambar dari USGS Landsat 8 dan pengumpulan Sentinel-2 data disediakan dalam tabel berikut.

USGS Landsat

Nama band	Rentang panjang gelombang (nm)	Unit	Rentang yang valid	Isi nilai	Resolusi spasial
pesisir	435 - 451	Tak Bersatu	1 - 65455	0 (Tidak Ada Data)	30m
biru	452 - 512	Tak Bersatu	1 - 65455	0 (Tidak Ada Data)	30m
hijau	533 - 590	Tak Bersatu	1 - 65455	0 (Tidak Ada Data)	30m
merah	636 - 673	Tak Bersatu	1 - 65455	0 (Tidak Ada Data)	30m
nir	851 - 879	Tak Bersatu	1 - 65455	0 (Tidak Ada Data)	30m
pusar16	1566 - 1651	Tak Bersatu	1 - 65455	0 (Tidak Ada Data)	30m
pusar22	2107 - 2294	Tak Bersatu	1 - 65455	0 (Tidak Ada Data)	30m
qa_aerosol	TA	Indeks Bit	0 - 255	1	30m
qa_pixel	TA	Indeks Bit	1 - 65455	1 (bit 0)	30m
qa_radsat	TA	Indeks Bit	1 - 65455	TA	30m

Nama band	Rentang panjang gelombang (nm)	Unit	Rentang yang valid	Isi nilai	Resolusi spasial
t	10600 - 11190	Kelvin Berskala	1 - 65455	0 (Tidak Ada Data)	30m (diskalakan dari 100m)
atran	TA	Tak Bersatu	0 - 10000	-9999 (Tidak Ada Data)	30m
cdist	TA	Kilometer	0 - 24000	-9999 (Tidak Ada Data)	30m
drad	TA	W/ (m ^ 2 sr μm) /DN	0 - 28000	-9999 (Tidak Ada Data)	30m
urad	TA	W/ (m ^ 2 sr μm) /DN	0 - 28000	-9999 (Tidak Ada Data)	30m
trad	TA	W/ (m ^ 2 sr μm) /DN	0 - 28000	-9999 (Tidak Ada Data)	30m
emis	TA	Koefisien emisivitas	1 - 10000	-9999 (Tidak Ada Data)	30m
emsd	TA	Koefisien emisivitas	1 - 10000	-9999 (Tidak Ada Data)	30m

Sentinel-2

Nama band	Rentang panjang gelombang (nm)	Penskalaan	Rentang yang valid	Isi nilai	Resolusi spasial
pesisir	443	0,0001	TA	0 (Tidak Ada Data)	60m

Nama band	Rentang panjang gelombang (nm)	Penskalaan	Rentang yang valid	Isi nilai	Resolusi spasial
biru	490	0,0001	TA	0 (Tidak Ada Data)	10m
hijau	560	0,0001	TA	0 (Tidak Ada Data)	10m
merah	665	0,0001	TA	0 (Tidak Ada Data)	10m
menebus 1	705	0,0001	TA	0 (Tidak Ada Data)	20m
menebus 2	740	0,0001	TA	0 (Tidak Ada Data)	20m
menebus 3	783	0,0001	TA	0 (Tidak Ada Data)	20m
nir	842	0,0001	TA	0 (Tidak Ada Data)	10m
nir08	865	0,0001	TA	0 (Tidak Ada Data)	20m
nir08	865	0,0001	TA	0 (Tidak Ada Data)	20m
nir09	940	0,0001	TA	0 (Tidak Ada Data)	60m
pusar16	1610	0,0001	TA	0 (Tidak Ada Data)	20m

Nama band	Rentang panjang gelombang (nm)	Penskalaan	Rentang yang valid	Isi nilai	Resolusi spasial
pusar22	2190	0,0001	TA	0 (Tidak Ada Data)	20m
aot	Ketebalan optik aerosol	0,001	TA	0 (Tidak Ada Data)	10m
wvp	Uap air rata-rata pemandangan	0,001	TA	0 (Tidak Ada Data)	10m
scl	Data klasifikasi adegan	TA	1 - 11	0 (Tidak Ada Data)	20m

RStudio di Amazon SageMaker

RStudio adalah lingkungan pengembangan terintegrasi untuk R, dengan konsol, editor sintaks-menyoroti yang mendukung eksekusi kode langsung, dan alat-alat untuk merencanakan, sejarah, debugging dan manajemen ruang kerja. Amazon SageMaker mendukung RStudio sebagai lingkungan pengembangan terintegrasi (IDE) yang dikelola sepenuhnya yang terintegrasi dengan Amazon SageMaker Domain melalui Posit Workbench. Untuk informasi lebih lanjut tentang Posit Workbench, lihat situs web Posit.

RStudio memungkinkan pelanggan untuk membuat wawasan ilmu data menggunakan lingkungan R. Dengan integrasi RStudio, Anda dapat meluncurkan lingkungan RStudio di Domain untuk menjalankan alur kerja RStudio Anda pada sumber daya. SageMaker

SageMakermengintegrasikan RStudio melalui pembuatan aplikasi RStudioServerPro.

Berikut ini didukung oleh RStudio padaSageMaker.

- R pengembang menggunakan antarmuka RStudio IDE dengan alat pengembang populer dari ekosistem R. Pengguna dapat meluncurkan sesi RStudio baru, menulis kode R, menginstal

dependensi dari RStudio Package Manager, dan mempublikasikan aplikasi Shiny menggunakan RStudio Connect.

- Pengembang R dapat dengan cepat menskalakan sumber daya komputasi yang mendasari untuk menjalankan pemrosesan data skala besar dan analisis statistik.
- Administrator platform dapat mengatur identitas pengguna, otorisasi, jaringan, penyimpanan, dan keamanan untuk tim sains data mereka melalui AWS IAM Identity Center dan AWS Identity and Access Management integrasi. Ini termasuk koneksi ke sumber daya Amazon Virtual Private Cloud (Amazon VPC) pribadi dan mode bebas internet dengan. AWS PrivateLink
- Integrasi denganAWS License Manager.

Untuk informasi tentang langkah-langkah orientasi untuk membuat Domain dengan RStudio diaktifkan, lihat. Ikhtisar SageMaker Domain Amazon

Ketersediaan Region

Tabel berikut memberikan informasi tentang RStudio Wilayah AWS yang SageMaker didukung di.

Nama wilayah	Wilayah
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-sentral-1

Nama wilayah	Wilayah
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
South America (Sao Paulo)	sa-east-1

Komponen RStudio

- R StudioServerPro: StudioServerPro Aplikasi R adalah aplikasi multiuser yang merupakan sumber daya bersama di antara semua profil pengguna di Domain. Setelah aplikasi RStudio dibuat di Domain, admin dapat memberikan izin kepada pengguna di Domain.
- Pengguna RStudio: pengguna RStudio adalah pengguna dalam Domain yang berwenang untuk menggunakan lisensi RStudio.
- Admin RStudio: Admin RStudio di Amazon dapat mengakses SageMaker dasbor administratif RStudio. SageMakerAdmin RStudio di Amazon berbeda dari admin Posit Workbench “stok” karena mereka tidak memiliki akses root ke instans yang menjalankan StudioServerPro aplikasi R dan tidak dapat memodifikasi file konfigurasi RStudio.
- Server RStudio: Instans Server RStudio bertanggung jawab untuk melayani UI RStudio ke semua Pengguna yang berwenang. Instance ini diluncurkan pada SageMaker instans Amazon.
- RSession: RSession adalah antarmuka berbasis browser ke RStudio IDE yang berjalan pada instans Amazon. SageMaker Pengguna dapat membuat dan berinteraksi dengan proyek RStudio mereka melalui RSession.
- R SessionGateway: SessionGateway Aplikasi R digunakan untuk mendukung RSession.
- Dasbor administratif RStudio: Dasbor ini memberikan informasi tentang pengguna RStudio di SageMaker Domain Amazon dan sesi mereka. Dasbor ini hanya dapat diakses oleh pengguna yang memiliki otorisasi admin RStudio.

Perbedaan dari Posit Workbench

RStudio di Amazon SageMaker memiliki beberapa perbedaan yang signifikan dari Posit Workbench.

- Saat menggunakan RStudio aktifSageMaker, pengguna tidak memiliki akses ke file konfigurasi RStudio. Amazon SageMaker mengelola file konfigurasi dan menetapkan default. Anda dapat memodifikasi URL Manajer Paket RStudio Connect dan RStudio saat membuat Domain Amazon berkemampuan RStudio. SageMaker
- Berbagi proyek, kolaborasi waktu nyata, dan Job Launcher saat ini tidak didukung saat menggunakan RStudio di Amazon. SageMaker
- Saat menggunakan RStudio aktifSageMaker, RStudio IDE berjalan di SageMaker instans Amazon untuk sumber daya komputasi kontainer sesuai permintaan.
- RStudio SageMaker hanya mendukung RStudio IDE dan tidak mendukung IDE lain yang didukung oleh instalasi Posit Workbench.
- RStudio SageMaker hanya mendukung versi RStudio yang ditentukan dalam. Tingkatkan Versi RStudio

Kelola RStudio di Amazon SageMaker

Topik berikut memberikan informasi tentang mengelola RStudio di AmazonSageMaker. Ini termasuk informasi tentang konfigurasi lingkungan RStudio Anda, sesi pengguna, dan sumber daya yang diperlukan. Untuk informasi tentang cara menggunakan RStudio padaSageMaker, lihatMenggunakan RStudio di Amazon SageMaker.

Untuk informasi tentang membuat SageMaker Domain Amazon dengan RStudio diaktifkan, lihatIkhtisar SageMaker Domain Amazon.

Untuk informasi tentang AWS Wilayah yang didukung oleh RStudio, lihatWilayah dan kuota yang didukung. SageMaker

Topik

- Lisensi RStudio
- Tingkatkan Versi RStudio
- Jaringan dan Penyimpanan
- Jenis StudioServerPro contoh R
- URL RStudio Connect

- Manajer Package RStudio
- Buat Amazon SageMaker Domain dengan RStudio menggunakanAWS CLI
- Tambahkan dukungan RStudio ke Domain yang ada
- Bawa gambar Anda sendiri ke RStudio SageMaker
- Mengelola pengguna
- Dasbor administratif RStudio
- Matikan dan restart RStudio
- Manajemen penagihan dan biaya
- Mendiagnosis masalah dan mendapatkan dukungan

Lisensi RStudio

RStudio di Amazon SageMaker adalah produk berbayar dan mengharuskan setiap pengguna dilisensikan dengan tepat. Lisensi untuk RStudio di Amazon SageMaker dapat diperoleh dari RStudio PBC secara langsung, atau dengan membeli langganan Posit Workbench di Marketplace. AWS Untuk pelanggan Posit Workbench Enterprise yang ada, lisensi dikeluarkan tanpa biaya tambahan.

Untuk menggunakan lisensi RStudio dengan AmazonSageMaker, Anda harus terlebih dahulu memiliki lisensi RStudio yang valid terdaftar. AWS License Manager Langganan ke Posit Workbench di AWS Marketplace secara otomatis memicu pembuatan lisensi dengan. AWS License Manager Untuk lisensi yang dibeli langsung melalui Rstudio PBC, pemberian lisensi untuk AWS Akun Anda harus dibuat. Hubungi RStudio untuk pembelian lisensi langsung atau untuk mengaktifkan lisensi yang ada di. AWS License Manager Untuk informasi selengkapnya tentang mendaftarkan lisensiAWS License Manager, lihat Penjual mengeluarkan lisensi di. AWS License Manager

Topik berikut menunjukkan cara memperoleh dan memvalidasi lisensi yang diberikan oleh RStudio PBC.

Dapatkan lisensi RStudio

1. Jika Anda tidak memiliki lisensi RStudio, Anda dapat membeli satu dari AWS Marketplace atau dari RStudio PBC secara langsung.

   - Untuk membeli langganan dari AWS Marketplace, selesaikan langkah-langkah dalam Berlangganan produk AMI dengan harga kontrak penawaran umum dengan mencari Posit Workbench.

- Untuk membeli langsung dari RStudio PBC, buka Harga RStudio atau hubungi sales@rstudio.com. Saat membeli atau memperbarui lisensi RStudio, Anda harus memberikan AWS Akun yang akan menjadi tuan rumah SageMaker Domain Amazon Anda.

Jika Anda memiliki lisensi RStudio yang ada, hubungi perwakilan Penjualan RStudio Anda atau sales@rstudio.com untuk menambahkan RStudio di Amazon SageMaker ke lisensi Posit Workbench Enterprise yang ada, atau untuk mengonversi lisensi Posit Workbench Standard Anda. Perwakilan Penjualan RStudio akan mengirimkan formulir pesanan elektronik yang sesuai.

2. RStudio memberikan lisensi Posit Workbench ke AWS Akun Anda melalui Wilayah AWS License Manager AS Timur (Virginia Utara). Meskipun lisensi RStudio diberikan di Wilayah AS Timur (Virginia Utara), lisensi Anda dapat dikonsumsi di AWS Wilayah mana pun yang didukung oleh RStudio di Amazon. SageMaker Anda dapat mengharapkan proses pemberian lisensi selesai dalam tiga hari kerja setelah membagikan ID AWS akun Anda dengan RStudio.
3. Ketika lisensi ini diberikan, Anda menerima email dari perwakilan Penjualan RStudio Anda dengan instruksi untuk menerima pemberian lisensi Anda.

Validasi lisensi RStudio Anda untuk digunakan dengan Amazon SageMaker

1. Masuk ke AWS License Manager konsol di wilayah yang sama dengan SageMaker Domain Amazon Anda. Jika Anda menggunakan AWS License Manager untuk pertama kalinya, AWS License Manager meminta Anda untuk memberikan izin untuk menggunakanAWS License Manager.
2. Pilih Mulai menggunakan Manajer AWS lisensi.
3. Pilih `I grant AWS License Manager the required permissions` dan pilih Izin Hibah.
4. Arahkan ke Lisensi yang Diberikan di panel kiri.
5. Pilih hibah lisensi dengan `RSW-SageMaker` sebagai `Product name` dan pilih View.
6. Dari halaman detail lisensi, pilih Terima & aktifkan lisensi.

Dasbor administratif RStudio

Anda dapat menggunakan dasbor administratif RStudio untuk melihat jumlah pengguna pada lisensi mengikuti langkah-langkah diDasbor administratif RStudio.

Tingkatkan Versi RStudio

Panduan ini memberikan informasi tentang`2023.03.2-454.pro2`pembaruan versi untuk RStudio pada SageMaker. Mulai 27 Juni 2023, domain baru dengan dukungan RStudio dibuat dengan`Posit Workbench`versi`2023.03.2-454.pro2`. Hal ini berlaku untuk`RStudioServerPro`aplikasi dan default`RSessionGateway`aplikasi.

Selama masa tenggang hingga 31 Agustus 2023, Anda dapat menguji versi terbaru di domain yang ada. Selama periode ini, domain yang ada terus menggunakan domain sebelumnya`2022.02.2-485.pro2`versi secara default. Untuk mengubah versi default ini, buat ulang`RStudioServerPro`aplikasi mengikuti langkah-langkah di`Tingkatkan ke versi baru`. Setelah 31 Agustus 2023, semua domain yang ada secara otomatis diluncurkan`RStudioServerPro`dan`RSessionGateway`aplikasi dengan versi`2023.03.2-454.pro2`.

Bagian berikut memberikan informasi tentang`2023.03.2-454.pro2`rilis dan cara menguji versi terbaru.

Update versi terbaru

Versi RStudio terbaru adalah`2023.03.2-454.pro2`. Versi ini mencakup perubahan berikut:

- Ditambahkan`RTools`4.3 dukungan
- Ditambahkan dukungan untuk R 4.3
- Upgrade`Quarto`ke 1.2.335
- Peningkatan manajemen sesi

Untuk informasi selengkapnya tentang perubahan dalam rilis ini, lihat`https://docs.posit.co/ide/news/`.

Versioning

Saat ini ada dua versi`Posit Workbench`didukung oleh SageMaker.

- Versi terbaru yang didukung:`2023.03.2-454.pro2`
- Versi sebelumnya didukung:`2022.02.2-485.pro2`

Selama masa tenggang, default`Posit Workbench`versi yang dipilih oleh SageMaker tergantung pada tanggal pembuatan domain.

- Untuk domain yang dibuat setelah 27 Juni 2023 (domain yang baru dibuat), versi`2023.03.2-454.pro2`adalah versi default yang dipilih.
- Untuk domain yang dibuat sebelum 27 Juni 2023 (domain yang ada), versi`2022.02.2-485.pro2`adalah versi default yang dipilih. Anda dapat memperbarui domain Anda ke versi terbaru (`2023.03.2-454.pro2`) dengan mengaturnya sebagai versi default untuk domain. Untuk informasi selengkapnya, lihat Tingkatkan ke versi baru.

> **Note**
>
> Default`RSessionGateway`versi aplikasi cocok dengan versi saat ini`RStudioServerPro`aplikasi.

Tabel berikut mencantumkan ARN gambar untuk kedua versi untuk masing-masingWilayah AWS. ARN ini dilewatkan sebagai bagian dari`update-domain`perintah untuk mengatur versi yang diinginkan.

Region	`2022.02.2-485.pro2` Gambar ARN	`2023.03.2-454.pro2` Gambar ARN
us-east-1	arn:aws:sagemaker:us-east-1:081325390199:image/rstudio-workbench-2021.08	arn:aws:sagemaker:us-east-1:081325390199:image/rstudio-workbench-2023.03
us–east-2	arn:aws:sagemaker:us-east-2:429704687514:image/rstudio-workbench-2021.08	arn:aws:sagemaker:us-east-2:429704687514:image/rstudio-workbench-2023.03
us-west-1	arn:aws:sagemaker:us-west-1:742091327244:image/rstudio-workbench-2021.08	arn:aws:sagemaker:us-west-1:742091327244:image/rstudio-workbench-2023.03
us-west-2	arn:aws:sagemaker:us-west-2:236514542706:image/rstudio-workbench-2021.08	arn:aws:sagemaker:us-west-2:236514542706:image/rstudio-workbench-2023.03

af-south-1	arn:aws:sagemaker:af-south-1:559312083959:image/rstudio-workbench-2021.08	arn:aws:sagemaker:af-south-1:559312083959:image/rstudio-workbench-2023.03
ap-east-1	arn:aws:sagemaker:ap-east-1:493642496378:image/rstudio-workbench-2021.08	arn:aws:sagemaker:ap-east-1:493642496378:image/rstudio-workbench-2023.03
ap-south-1	arn:aws:sagemaker:ap-south-1:394103062818:image/rstudio-workbench-2021.08	arn:aws:sagemaker:ap-south-1:394103062818:image/rstudio-workbench-2023.03
ap-northeast-2	arn:aws:sagemaker:ap-northeast-2:806072073708:image/rstudio-workbench-2021.08	arn:aws:sagemaker:ap-northeast-2:806072073708:image/rstudio-workbench-2023.03
ap-southeast-1	arn:aws:sagemaker:ap-southeast-1:492261229750:image/rstudio-workbench-2021.08	arn:aws:sagemaker:ap-southeast-1:492261229750:image/rstudio-workbench-2023.03
ap-southeast-2	arn:aws:sagemaker:ap-southeast-2:452832661640:image/rstudio-workbench-2021.08	arn:aws:sagemaker:ap-southeast-2:452832661640:image/rstudio-workbench-2023.03
ap-northeast-1	arn:aws:sagemaker:ap-northeast-1:102112518831:image/rstudio-workbench-2021.08	arn:aws:sagemaker:ap-northeast-1:102112518831:image/rstudio-workbench-2023.03
ca-central-1	arn:aws:sagemaker:ca-central-1:310906938811:image/rstudio-workbench-2021.08	arn:aws:sagemaker:ca-central-1:310906938811:image/rstudio-workbench-2023.03
eu-central-1	arn:aws:sagemaker:eu-central-1:936697816551:image/rstudio-workbench-2021.08	arn:aws:sagemaker:eu-central-1:936697816551:image/rstudio-workbench-2023.03
eu-west-1	arn:aws:sagemaker:eu-west-1:470317259841:image/rstudio-workbench-2021.08	arn:aws:sagemaker:eu-west-1:470317259841:image/rstudio-workbench-2023.03

eu-west-2	arn:aws:sagemaker:eu-west-2:712779665605:image/rstudio-workbench-2021.08	arn:aws:sagemaker:eu-west-2:712779665605:image/rstudio-workbench-2023.03
eu-west-3	arn:aws:sagemaker:eu-west-3:615547856133:image/rstudio-workbench-2021.08	arn:aws:sagemaker:eu-west-3:615547856133:image/rstudio-workbench-2023.03
eu-north-1	arn:aws:sagemaker:eu-north-1:243637512696:image/rstudio-workbench-2021.08	arn:aws:sagemaker:eu-north-1:243637512696:image/rstudio-workbench-2023.03
eu-south-1	arn:aws:sagemaker:eu-south-1:592751261982:image/rstudio-workbench-2021.08	arn:aws:sagemaker:eu-south-1:592751261982:image/rstudio-workbench-2023.03
sa-east-1	arn:aws:sagemaker:sa-east-1:782484402741:image/rstudio-workbench-2021.08	arn:aws:sagemaker:sa-east-1:782484402741:image/rstudio-workbench-2023.03

Tingkatkan ke versi baru

Domain yang ada terus menggunakan versi`2022.02.2-485.pro2`selama masa tenggang. Selama masa tenggang, Anda dapat menggunakan yang terbaru`2023.03.2-454.pro2`versi dalam salah satu dari dua cara:

- Membuat domain baru dariAWS CLIdengan RStudio diaktifkan.
- Perbarui domain yang ada untuk menggunakan`2023.03.2-454.pro2`versi.

Prosedur berikut menunjukkan cara menghapus aplikasi RStudio untuk domain yang ada, mengatur versi default ke`2023.03.2-454.pro2`, dan kemudian membuat aplikasi RStudio. Prosedur ini hanya diperlukan selama masa tenggang.

1. Hapus`RStudioServerPro`aplikasi dan semua`RSessionGateway`aplikasi yang terkait dengan domain Anda yang ada. Untuk informasi tentang cara menemukan ID domain, lihatLihat Domain. Untuk informasi selengkapnya tentang menghapus aplikasi, lihatMatikan dan restart RStudio.

```
aws sagemaker delete-app \
    --region region \
    --domain-id domainId \
    --user-profile-name domain-shared \
    --app-type RStudioServerPro \
    --app-name default
```

2. Perbarui domain yang akan disetel`2023.03.2-454.pro2`sebagai defaultPosit Workbenchversi. The`SageMakerImageArn`nilai berikut`update-domain`perintah menentukan RStudio`2023.03.2-454.pro2`versi sebagai default. ARN ini harus cocok dengan`Region`bahwa domain Anda berada di. Untuk daftar semua ARN yang tersedia, lihatVersioning.

   Lulus peran eksekusi ARN untuk domain yang menyediakan izin untuk memperbarui domain.

```
aws sagemaker update-domain \
    --region region \
    --domain-id domainId \
    --domain-settings-for-update "{\"RStudioServerProDomainSettingsForUpdate\":
{\"DefaultResourceSpec\": {\"SageMakerImageArn\": \"arn-for-2023.03.2-454.pro2-
version\", \"InstanceType\": \"system\"}, \"DomainExecutionRoleArn\": \"execution-
role-arn\"}}"
```

3. Buat yang baru`RStudioServerPro`aplikasi di domain yang ada.

```
aws sagemaker create-app \
    --region region
    --domain-id domainId \
    --user-profile-name domain-shared \
    --app-type RStudioServerPro \
    --app-name default
```

Anda`RStudioServerPro`aplikasi sekarang diperbarui ke versi`2023.03.2-454.pro2`. Anda sekarang dapat meluncurkan kembali`RSessionGateway`aplikasi.

Turunkan versi ke versi yang ada

Selama masa tenggang, Anda dapat secara manual menurunkan versi aplikasi RStudio yang ada ke`2022.02.2-485.pro2`versi.

Untuk menurunkan versi ke versi yang ada

1. Hapus`RStudioServerPro`aplikasi yang terkait dengan domain Anda yang ada. Untuk informasi tentang cara menemukan ID domain, lihatLihat Domain.

```
aws sagemaker delete-app \
    --domain-id domainId \
    --user-profile-name domain-shared \
    --app-type RStudioServerPro \
    --app-name default
```

2. Lewati yang sesuai`2022.02.2-485.pro2`ARN untuk AndaRegionsebagai bagian dari`update-domain`perintah. Untuk daftar semua ARN yang tersedia, lihatVersioning. Anda juga harus melewati peran eksekusi ARN untuk domain yang menyediakan izin untuk memperbarui domain.

```
aws sagemaker update-domain \
    --region region \
    --domain-id domainId \
    --domain-settings-for-update "{\"RStudioServerProDomainSettingsForUpdate\":
{\"DefaultResourceSpec\": {\"SageMakerImageArn\": \"arn-for-2022.02.2+485.pro2-
version\", \"InstanceType\": \"system\"}, \"DomainExecutionRoleArn\": \"execution-
role-arn\"}}"
```

3. Buat yang baru`RStudioServerPro`aplikasi di domain yang ada. Versi RStudio default ke`2022.02.2-485.pro2`.

```
aws sagemaker create-app \
    --domain-id domainId \
    --user-profile-name domain-shared \
    --app-type RStudioServerPro \
    --app-name default
```

Anda`RStudioServerPro`aplikasi sekarang diturunkan ke versi`2022.02.2-485.pro2`.

Perubahan pada Gambar BYOI

Jika Anda menggunakan gambar BYOI dengan RStudio dan memperbarui`RStudioServerPro`versi ke`2023.03.2-454.pro2`, Anda harus memutakhirkan gambar khusus Anda untuk menggunakan`2023.03.2-454.pro2`rilis dan terapkan ulang RSessions Anda yang ada. Jika Anda mencoba memuat gambar yang tidak kompatibel di RSession domain

menggunakan`2023.03.2-454.pro2`versi, RSession gagal karena tidak dapat mengurai parameter yang diterimanya. Untuk mencegah kegagalan, perbarui semua gambar kustom yang diterapkan di yang sudah ada`RStudioServerPro`aplikasi.

The`RSW_VERSION`diDockerfileharus konsisten denganPosit Workbenchversi yang digunakan di RStudio pada SageMaker. Anda dapat memvalidasi versi saat ini diPosit Workbench. Untuk melakukannya, gunakan nama versi yang terletak di sudut kiri bawahPosit Workbenchhalaman peluncur.

```
...
ARG RSW_VERSION=2023.03.2+454.pro2
ENV RSTUDIO_FORCE_NON_ZERO_EXIT_CODE="1"
ARG RSW_NAME=rstudio-workbench-rhel
ARG RSW_DOWNLOAD_URL=https://s3.amazonaws.com/rstudio-ide-build/server/${OS}/x86_64
RUN RSW_VERSION_URL=`echo -n "${RSW_VERSION}" | sed 's/+/-/g'` \
    && curl -o rstudio-workbench.rpm ${RSW_DOWNLOAD_URL}/${RSW_NAME}-
${RSW_VERSION_URL}-x86_64.rpm \
    && yum install -y rstudio-workbench.rpm
```

Jaringan dan Penyimpanan

Topik berikut menjelaskan akses jaringan dan pertimbangan penyimpanan data untuk instans RStudio Anda. Untuk informasi umum tentang akses jaringan dan penyimpanan data saat menggunakan Amazon SageMaker, lihatPerlindungan Data di Amazon SageMaker.

Volume Amazon EFS

RStudio di Amazon SageMaker membagikan volume Amazon EFS dengan aplikasi Amazon SageMaker Studio Classic di domain. Saat aplikasi RStudio ditambahkan ke domain, SageMaker buat folder bernama `shared` di direktori Amazon EFS. Jika `shared` folder ini dihapus atau diubah secara manual, maka aplikasi RStudio mungkin tidak lagi berfungsi. Untuk informasi selengkapnya tentang volume Amazon EFS, lihatKelola Volume Penyimpanan Amazon EFS Anda di SageMaker Studio Classic.

Paket dan skrip yang diinstal

Paket yang Anda instal dari dalam RStudio dicakup ke tingkat profil pengguna. Ini berarti bahwa paket yang diinstal tetap ada melalui RSession shutdown, restart, dan di seluruh RSessions untuk setiap profil pengguna tempat mereka diinstal. R Script yang disimpan di RSessions berperilaku dengan cara yang sama. Baik paket dan R Script disimpan dalam volume Amazon EFS pengguna.

Enkripsi

RStudio di Amazon SageMaker mendukung enkripsi saat istirahat.

Gunakan RStudio dalam mode VPC saja

RStudio di Amazon SageMaker mendukung AWS PrivateLinkintegrasi. Dengan integrasi ini, Anda dapat menggunakan RStudio aktif SageMaker dalam mode VPC saja tanpa akses langsung ke internet. Bila Anda menggunakan RStudio dalam mode VPC saja, grup keamanan Anda secara otomatis dikelola oleh layanan. Ini termasuk konektivitas antara RServer Anda dan RSessions Anda.

Berikut ini diperlukan untuk menggunakan RStudio dalam mode VPC saja. Untuk informasi selengkapnya tentang memilih VPC, lihat. Pilih VPC Amazon

- Subnet pribadi dengan akses internet untuk melakukan panggilan ke Amazon SageMaker & License Manager, atau titik akhir Amazon Virtual Private Cloud (Amazon VPC) untuk Amazon & SageMaker License Manager.
- Domain tidak dapat memiliki lebih dari dua Grup Keamanan terkait.
- ID Grup Keamanan untuk digunakan dengan Domain di Pengaturan Domain. Ini harus memungkinkan semua akses keluar.
- ID Grup Keamanan untuk digunakan dengan titik akhir VPC Amazon. Grup keamanan ini harus mengizinkan lalu lintas masuk dari ID Grup Keamanan Domain.
- Titik Akhir VPC Amazon untuk dan. `sagemaker.api` AWS License Manager Ini harus dalam VPC Amazon yang sama dengan subnet pribadi.

Jenis StudioServerPro contoh R

Saat memutuskan jenis instans Amazon EC2 yang akan digunakan untuk StudioServerPro aplikasi R Anda, faktor utama yang perlu dipertimbangkan adalah bandwidth. Bandwidth penting karena StudioServerPro instans R bertanggung jawab untuk melayani UI RStudio ke semua pengguna. Ini termasuk alur kerja UI yang berat, seperti menghasilkan angka, animasi, dan menampilkan banyak baris data. Oleh karena itu, mungkin ada beberapa penurunan kinerja UI tergantung pada beban kerja di semua pengguna. Berikut ini adalah jenis instance yang tersedia untuk digunakan untuk R AndaStudioServerPro. Untuk informasi harga tentang instans ini, lihat SageMakerHarga Amazon.

- `ml.t3.medium`: Jenis instance ini direkomendasikan untuk Domain dengan penggunaan UI rendah dan gratis untuk digunakan.

> **Note**
>
> `system`Nilai diterjemahkan ke`ml.t3.medium`.

- `ml.c5.4xlarge`: Jenis instance ini direkomendasikan untuk Domain dengan penggunaan UI moderat.
- `ml.c5.9xlarge`: Jenis instance ini direkomendasikan untuk Domain dengan penggunaan UI berat.

Mengubah tipe instans RStudio

Untuk mengubah tipe instance R AndaStudioServerPro, teruskan tipe instance baru sebagai bagian dari panggilan ke perintah `update-domain` CLI. Anda kemudian perlu menghapus StudioServerPro aplikasi R yang ada menggunakan perintah `delete-app` CLI dan membuat StudioServerPro aplikasi R baru menggunakan perintah CLI`create-app`.

URL RStudio Connect

RStudio Connect adalah platform penerbitan untuk aplikasi Shiny, laporan R penurunan harga, dashboard, plot, dan banyak lagi. RStudio Connect memudahkan untuk memunculkan pembelajaran mesin dan wawasan ilmu data dengan membuat konten hosting menjadi sederhana dan terukur. Jika Anda memiliki server RStudio Connect, maka Anda dapat mengatur server sebagai tempat default di mana aplikasi dipublikasikan. Untuk informasi selengkapnya tentang RStudio Connect, lihat RStudio Connect.

Saat Anda bergabung ke RStudio di Amazon SageMaker Domain, server RStudio Connect tidak dibuat. Anda dapat membuat server RStudio Connect pada instans Amazon EC2 untuk menggunakan Connect with Amazon SageMaker Domain. Untuk informasi tentang cara menyiapkan server RStudio Connect Anda, lihat Host RStudio Connect dan Package Manager untuk pengembangan MLdi RStudio di Amazon SageMaker.

Menambahkan URL RStudio Connect

Jika Anda memiliki URL RStudio Connect, Anda dapat memperbarui URL default sehingga Pengguna RStudio Anda dapat mempublikasikannya.

1. Arahkan ke halaman Domain.
2. Pilih Domain yang diinginkan.

3. Pilih Pengaturan Domain.
4. Di bawah Pengaturan Umum, pilih Edit.
5. Dari halaman baru, pilih Pengaturan RStudio di sisi kiri.
6. Di bawah URL RStudio Connect, masukkan URL RStudio Connect untuk ditambahkan.
7. Pilih Kirim.

CLI

Anda dapat mengatur URL RStudio Connect default saat membuat Domain. Satu-satunya cara untuk memperbarui URL RStudio Connect Anda dariAWS CLI adalah dengan menghapus Domain Anda dan membuat yang baru dengan URL RStudio Connect yang diperbarui.

Manajer Package RStudio

RStudio Package Manager adalah server manajemen repositori yang digunakan untuk mengatur dan memusatkan paket di seluruh organisasi Anda. Untuk informasi selengkapnya tentang Manajer Package RStudio, lihat Manajer Package RStudio. Jika Anda tidak menyediakan URL Manajer Package Anda sendiri, Amazon SageMaker Domain menggunakan repositori Package Manager default ketika Anda onboard RStudio mengikuti langkah-langkah dilkhtisar SageMaker Domain Amazon. Untuk informasi selengkapnya, lihat Host RStudio Connect dan Package Manager untuk pengembangan MLdi RStudio di Amazon SageMaker.

Perbarui URL Manajer Package

Anda dapat memperbarui URL Manajer Package yang digunakan untuk Domain berkemampuan RStudio Anda sebagai berikut.

1. Arahkan ke halaman Domain.
2. Pilih Domain yang diinginkan.
3. Pilih Pengaturan Domain.
4. Di bawah Pengaturan Umum, pilih Edit.
5. Dari halaman baru, pilih Pengaturan RStudio di sisi kiri.
6. Di bawah Manajer Package RStudio, masukkan URL Manajer Package RStudio Anda.
7. Pilih Kirim.

CLI

Satu-satunya cara untuk memperbarui URL Manajer Package Anda dariAWS CLI adalah dengan menghapus Domain Anda dan membuat yang baru dengan URL Manajer Package yang diperbarui.

Buat Amazon SageMaker Domain dengan RStudio menggunakanAWS CLI

Topik berikut menunjukkan cara bergabung ke Amazon SageMaker Domain dengan RStudio diaktifkan menggunakanAWS CLI. Untuk onboard menggunakanAWS Management Console, LihatIkhtisar SageMaker Domain Amazon.

Prasyarat

- Instal dan konfigurasiAWS CLIversi 2
- MengonfigurasiAWS CLIdengan kredensi IAM

Buat**`DomainExecution`**peran

Untuk meluncurkan Aplikasi RStudio, Anda harus menyediakan`DomainExecution`peran. Peran ini digunakan untuk menentukan apakah RStudio perlu diluncurkan sebagai bagian dari Amazon SageMaker Pembuatan domain. Peran ini juga digunakan oleh Amazon SageMaker untuk mengakses Lisensi RStudio dan mendorong log RStudio.

> **Note**
>
> Klaster`DomainExecution`peran harus memiliki setidaknyaAWS License Managerizin untuk mengakses Lisensi RStudio, dan CloudWatch izin untuk mendorong log di akun Anda.

Prosedur berikut menunjukkan cara menciptakan`DomainExecution`peran denganAWS CLI.

1. Buat file dengan nama `assume-role-policy.json` dengan konten berikut ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com"
```

```
                ]
            }
        }
    ]
}
```

2. BuatDomainExecutionperan.<REGION>harus menjadiAWSWilayah untuk meluncurkan Domain Anda di.

```
aws iam create-role --region <REGION> --role-name DomainExecution --assume-role-
policy-document file://assume-role-policy.json
```

3. Buat file dengan nama domain-setting-policy.json dengan konten berikut ini. Kebijakan ini memungkinkan RStudioServerPro aplikasi untuk mengakses sumber daya yang diperlukan dan memungkinkan Amazon SageMaker untuk secara otomatis meluncurkan RStudioServerPro aplikasi saat R yang adaStudioServerPro aplikasi ada diDeletedatauFailedstatus.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "license-manager:ExtendLicenseConsumption",
                "license-manager:ListReceivedLicenses",
                "license-manager:GetLicense",
                "license-manager:CheckoutLicense",
                "license-manager:CheckInLicense",
                "logs:CreateLogDelivery",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DeleteLogDelivery",
                "logs:Describe*",
                "logs:GetLogDelivery",
                "logs:GetLogEvents",
                "logs:ListLogDeliveries",
                "logs:PutLogEvents",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery",
                "sagemaker:CreateApp"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

4. Buat kebijakan pengaturan Domain yang dilampirkan ke`DomainExecution`peran. Ketahui`PolicyArn`dari respon, Anda harus memasukkan ARN itu dalam langkah-langkah berikut.

```
aws iam create-policy --region <REGION> --policy-name domain-setting-policy --
policy-document file://domain-setting-policy.json
```

5. Melampirkan`domain-setting-policy`kepada`DomainExecution`peran. Gunakan`PolicyArn`kembali pada langkah sebelumnya.

```
aws iam attach-role-policy --role-name DomainExecution --policy-arn <POLICY_ARN>
```

Buat Amazon SageMaker Domain dengan Aplikasi RStudio

RStudioServerPro Aplikasi diluncurkan secara otomatis saat Anda membuat Amazon SageMaker Domain menggunakan`create-domain`Perintah CLI dengan`RStudioServerProDomainSettings`parameter yang ditentukan. Saat meluncurkan RStudioServerPro Aplikasi, Amazon SageMaker memeriksa lisensi RStudio valid di akun dan gagal pembuatan Domain jika lisensi tidak ditemukan.

Penciptaan Amazon SageMaker Domain berbeda berdasarkan metode otentikasi dan jenis jaringan. Opsi ini harus digunakan bersama, dengan satu metode otentikasi dan satu jenis koneksi jaringan dipilih. Untuk informasi lebih lanjut tentang persyaratan untuk membuat Domain baru, lihatCreateDomain.

Metode Autentikasi berikut didukung.

- `IAM Auth`
- `SSO Auth`

Tipe koneksi jaringan berikut ini didukung:

- `PublicInternet`
- `VPCOnly`

Metode Autentikasi

Mode Autentikasi IAM

Berikut ini menunjukkan cara membuat Amazon SageMaker Domain dengan RStudio diaktifkan dan`IAM Auth`Jenis jaringan. Untuk informasi lebih lanjut tentangAWS Identity and Access Management, LihatApa itu IAM?.

- `DomainExecutionRoleArn`harus menjadi ARN untuk peran yang dibuat di langkah sebelumnya.
- `ExecutionRole`adalah ARN dari peran yang diberikan kepada pengguna di Amazon SageMaker Domain.
- `vpc-id`harus menjadi ID Amazon Virtual Private Cloud.`subnet-ids`harus berupa daftar ID subnet yang dipisahkan oleh spasi. Untuk informasi tentang`vpc-id`dan`subnet-ids`, LihatVPC dan subnet.
- `RStudioPackageManagerUrl`dan`RStudioConnectUrl`bersifat opsional dan harus diatur ke URL Manajer Package RStudio dan server RStudio Connect Anda, masing-masing.
- `app-network-access-type`harus baik`PublicInternetOnly`atau`VPCOnly`.

```
aws sagemaker create-domain --region <REGION> --domain-name <DOMAIN_NAME> \
    --auth-mode IAM \
    --default-user-settings ExecutionRole=<DEFAULT_USER_EXECUTIONROLE> \
    --domain-settings
 RStudioServerProDomainSettings={RStudioPackageManagerUrl=<<PACKAGE_MANAGER_URL>,RStudioConnect
 \
    --vpc-id <VPC_ID> \
    --subnet-ids <SUBNET_IDS> \
    --app-network-access-type <NETWORK_ACCESS_TYPE>
```

Otentikasi menggunakan IAM Identity Center

Berikut ini menunjukkan cara membuat Amazon SageMaker Domain dengan RStudio diaktifkan dan`SSO Auth`Jenis jaringan.AWS IAM Identity Centerharus diaktifkan untuk wilayah tempat domain diluncurkan. Untuk informasi selengkapnya tentang IAM Identity Center, lihatApaAWS IAM Identity Center?.

- `DomainExecutionRoleArn`harus menjadi ARN untuk peran yang dibuat di langkah sebelumnya.
- `ExecutionRole`adalah ARN dari peran yang diberikan kepada pengguna di Amazon SageMaker Domain.

- `vpc-id`harus menjadi ID Amazon Virtual Private Cloud.`subnet-ids`harus berupa daftar ID subnet yang dipisahkan oleh spasi. Untuk informasi tentang`vpc-id`dan`subnet-ids`, LihatVPC dan subnet.
- `RStudioPackageManagerUrl`dan`RStudioConnectUrl`bersifat opsional dan harus diatur ke URL Manajer Package RStudio dan server RStudio Connect Anda, masing-masing.
- `app-network-access-type`harus baik`PublicInternetOnly`atau`VPCOnly`.

```
aws sagemaker create-domain --region <REGION> --domain-name <DOMAIN_NAME> \
    --auth-mode SSO \
    --default-user-settings ExecutionRole=<DEFAULT_USER_EXECUTIONROLE> \
    --domain-settings
 RStudioServerProDomainSettings={RStudioPackageManagerUrl=<<PACKAGE_MANAGER_URL>,RStudioConnect
 \
    --vpc-id <VPC_ID> \
    --subnet-ids <SUBNET_IDS> \
    --app-network-access-type <NETWORK_ACCESS_TYPE>
```

Tipe koneksi

PublicInternet/Jenis jaringan Internet langsung

Berikut ini menunjukkan cara membuat Amazon SageMaker Domain dengan RStudio diaktifkan dan`PublicInternet`Jenis jaringan.

- `DomainExecutionRoleArn`harus menjadi ARN untuk peran yang dibuat di langkah sebelumnya.
- `ExecutionRole`adalah ARN dari peran yang diberikan kepada pengguna di Amazon SageMaker Domain.
- `vpc-id`harus menjadi ID Amazon Virtual Private Cloud.`subnet-ids`harus berupa daftar ID subnet yang dipisahkan oleh spasi. Untuk informasi tentang`vpc-id`dan`subnet-ids`, LihatVPC dan subnet.
- `RStudioPackageManagerUrl`dan`RStudioConnectUrl`bersifat opsional dan harus diatur ke URL Manajer Package RStudio dan server RStudio Connect Anda, masing-masing.
- `auth-mode`harus baik`SSO`atau`IAM`.

```
aws sagemaker create-domain --region <REGION> --domain-name <DOMAIN_NAME> \
    --auth-mode <AUTH_MODE> \
```

```
    --default-user-settings ExecutionRole=<DEFAULT_USER_EXECUTIONROLE> \
    --domain-settings
RStudioServerProDomainSettings={RStudioPackageManagerUrl=<<PACKAGE_MANAGER_URL>,RStudioConnect
\
    --vpc-id <VPC_ID> \
    --subnet-ids <SUBNET_IDS> \
    --app-network-access-type PublicInternetOnly
```

Mode VPConly

Berikut ini menunjukkan cara meluncurkan Amazon SageMaker Domain dengan RStudio diaktifkan dan`VPCOnly`Jenis jaringan. Untuk informasi lebih lanjut tentang penggunaan`VPCOnly`jenis akses jaringan, lihatConnect SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal.

- `DomainExecutionRoleArn`harus menjadi ARN untuk peran yang dibuat di langkah sebelumnya.
- `ExecutionRole`adalah ARN dari peran yang diberikan kepada pengguna di Amazon SageMaker Domain.
- `vpc-id`harus menjadi ID Amazon Virtual Private Cloud.`subnet-ids`harus berupa daftar ID subnet yang dipisahkan oleh spasi. Subnet pribadi Anda harus dapat mengakses internet untuk melakukan panggilan ke Amazon SageMaker, danAWS License Manageratau memiliki Amazon VPC endpoint untuk Amazon SageMaker danAWS License Manager. Untuk informasi tentang Amazon VPC endpoint, lihatTitik akhir VPC antarmukaUntuk informasi tentang`vpc-id`dan`subnet-ids`, LihatVPC dan subnet.
- `SecurityGroups`harus mengizinkan akses outbound ke Amazon SageMaker danAWS License Managertitik akhir.
- `auth-mode`harus baik`SSO`atau`IAM`.

> **Note**
>
> Saat menggunakan titik akhir Amazon Virtual Private Cloud, grup keamanan yang dilampirkan ke titik akhir Amazon Virtual Private Cloud Anda harus mengizinkan lalu lintas masuk dari grup keamanan yang Anda lewati sebagai bagian dari`domain-setting`parameter dari`create-domain`Panggilan CLI.

Dengan RStudio, Amazon SageMaker mengelola grup keamanan untuk Anda. Ini artinya bahwa Amazon SageMaker mengelola aturan grup keamanan untuk memastikan RSessions dapat

mengakses RStudioServerPro Aplikasi. Amazon SageMaker membuat satu aturan grup keamanan per profil pengguna.

```
aws sagemaker create-domain --region <REGION> --domain-name <DOMAIN_NAME> \
    --auth-mode <AUTH_MODE> \
    --default-user-settings
 SecurityGroups=<USER_SECURITY_GROUP>,ExecutionRole=<DEFAULT_USER_EXECUTIONROLE> \
    --domain-settings
 SecurityGroupIds=<DOMAIN_SECURITY_GROUP>,RStudioServerProDomainSettings={DomainExecutionRoleAr
 \
    --vpc-id <VPC_ID> \
    --subnet-ids "<SUBNET_IDS>" \
    --app-network-access-type VPCOnly --app-security-group-management Service
```

Catatan: RStudioServerPro aplikasi diluncurkan oleh profil pengguna khusus bernama`domain-shared`. Akibatnya, aplikasi ini tidak dikembalikan sebagai bagian dari`list-app`Panggilan API oleh profil pengguna lainnya.

Anda mungkin harus menambah kuota Amazon VPC di akun Anda untuk meningkatkan jumlah pengguna. Untuk informasi selengkapnya, lihat kuota Amazon VPC.

Memverifikasi pembuatan domain

Gunakan perintah berikut untuk memastikan Domain Anda telah dibuat dengan a`Status`dari`InService`. Klasterdomain-iddditambahkan ke Domain ARN. Misalnya, `arn:aws:sagemaker:`*<REGION>*`:`*<ACCOUNT_ID>*`:domain/`*<DOMAIN_ID>*.

```
aws sagemaker describe-domain --domain-id <DOMAIN_ID> --region <REGION>
```

Tambahkan dukungan RStudio ke Domain yang ada

Jika Anda telah menambahkan Lisensi RStudio melaluiAWS License Manager, Anda dapat membuat SageMaker Domain Amazon baru dengan dukungan untuk RStudio aktif. SageMaker Jika Anda memiliki Domain yang sudah ada yang tidak mendukung RStudio, Anda dapat menambahkan dukungan RStudio ke Domain tersebut tanpa harus menghapus dan membuat ulang Domain.

Topik berikut menguraikan cara menambahkan dukungan ini.

Prasyarat

Anda harus menyelesaikan langkah-langkah berikut sebelum memperbarui Domain Anda saat ini untuk menambahkan dukungan untuk RStudio. SageMaker

- Instal dan konfigurasikan AWS CLIversi 2
- Konfigurasikan AWS CLIkredenal dengan IAM
- Buat peran eksekusi Domain mengikuti langkah-langkah dalam Membuat SageMaker Domain dengan RStudio menggunakan. AWS CLI Peran IAM tingkat Domain ini diperlukan oleh aplikasi R. StudioServerPro Peran tersebut memerlukan akses AWS License Manager untuk memverifikasi lisensi Posit Workbench yang valid dan Amazon CloudWatch Logs untuk menerbitkan log server.
- Bawa lisensi RStudio Anda untuk AWS License Manager mengikuti langkah-langkah dalam lisensi RStudio.
- (Opsional) Jika Anda ingin menggunakan RStudio dalam `VPCOnly` mode, selesaikan langkah-langkah di RStudio di VPC saja.
- Pastikan bahwa grup keamanan yang telah Anda konfigurasikan untuk masing-masing UserProfiledi Domain Anda memenuhi kuota tingkat akun. Saat mengonfigurasi profil pengguna default selama pembuatan Domain, Anda dapat menggunakan `DefaultUserSettings` parameter CreateDomainAPI untuk menambahkan `SecurityGroups` yang diwarisi oleh semua profil pengguna yang dibuat di Domain. Anda juga dapat menyediakan grup keamanan tambahan untuk pengguna tertentu sebagai bagian dari `UserSettings` parameter CreateUserProfileAPI. Jika Anda telah menambahkan grup keamanan dengan cara ini, Anda harus memastikan bahwa jumlah grup keamanan per profil pengguna tidak melebihi kuota maksimum 2 dalam `VPCOnly` mode dan 4 dalam `PublicInternetOnly` mode. Jika jumlah grup keamanan yang dihasilkan untuk profil pengguna melebihi kuota, Anda dapat menggabungkan beberapa aturan grup keamanan menjadi satu grup keamanan.

Tambahkan dukungan RStudio ke Domain yang ada

Setelah menyelesaikan prasyarat, Anda dapat menambahkan dukungan RStudio ke Domain yang ada. Langkah-langkah berikut menguraikan cara memperbarui Domain yang ada untuk menambahkan dukungan untuk RStudio.

Langkah 1: Hapus semua aplikasi di Domain

Untuk menambahkan dukungan untuk RStudio di Domain Anda, SageMaker harus memperbarui grup keamanan yang mendasarinya untuk semua profil pengguna yang ada. Untuk menyelesaikan ini, Anda harus menghapus dan membuat ulang semua aplikasi yang ada di Domain. Prosedur berikut menunjukkan cara menghapus semua aplikasi.

1. Daftar semua aplikasi di Domain.

```
aws sagemaker \
   list-apps \
   --domain-id-equals <DOMAIN_ID>
```

2. Hapus setiap aplikasi untuk setiap profil pengguna di Domain.

```
// JupyterServer apps
aws sagemaker \
    delete-app \
    --domain-id <DOMAIN_ID> \
    --user-profile-name <USER_PROFILE> \
    --app-type JupyterServer \
    --app-name <APP_NAME>

// KernelGateway apps
aws sagemaker \
    delete-app \
    --domain-id <DOMAIN_ID> \
    --user-profile-name <USER_PROFILE> \
    --app-type KernelGateway \
    --app-name <APP_NAME>
```

Langkah 2 - Perbarui semua profil pengguna dengan daftar baru grup keamanan

Ini adalah tindakan satu kali yang harus Anda selesaikan untuk semua profil pengguna yang ada di Domain Anda ketika Anda telah memfaktorkan ulang grup keamanan yang ada. Ini mencegah Anda mencapai kuota untuk jumlah maksimum grup keamanan. Panggilan `UpdateUserProfile` API gagal jika pengguna memiliki aplikasi yang InServiceberstatus. Hapus semua aplikasi, lalu panggil `UpdateUserProfile` API untuk memperbarui grup keamanan.

> **Note**
>
> Persyaratan berikut untuk `VPCOnly` mode yang diuraikan dalam Connect Amazon SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal tidak lagi diperlukan saat menambahkan dukungan RStudio karena `AppSecurityGroupManagement` dikelola oleh layanan: SageMaker

"Lalu lintas TCP dalam grup keamanan. Ini diperlukan untuk konektivitas antara JupyterServer aplikasi dan KernelGateway aplikasi. Anda harus mengizinkan akses ke setidaknya port dalam jangkauan8192-65535."

```
aws sagemaker \
    update-user-profile \
    --domain-id <DOMAIN_ID>\
    --user-profile-name <USER_PROFILE> \
    --user-settings "{\"SecurityGroups\": [\"<SECURITY_GROUP>\",
 \"<SECURITY_GROUP>\"]}"
```

Langkah 3 - Aktifkan RStudio dengan memanggil API UpdateDomain

1. Panggil UpdateDomainAPI untuk menambahkan dukungan untuk RStudio aktif. SageMaker `defaultusersettings`Parameter hanya diperlukan jika Anda telah memfaktorkan ulang grup keamanan default untuk profil pengguna Anda.

   - Untuk `VPCOnly` mode:

     ```
     aws sagemaker \
         update-domain \
         --domain-id <DOMAIN_ID> \
         --app-security-group-management Service \
         --domain-settings-for-update
      RStudioServerProDomainSettingsForUpdate={DomainExecutionRoleArn=<DOMAIN_EXECUTION_ROLE_A
      \
         --default-user-settings "{\"SecurityGroups\": [\"<SECURITY_GROUP>\",
      \"<SECURITY_GROUP>\"]}"
     ```

   - Untuk `PublicInternetOnly` mode:

     ```
     aws sagemaker \
         update-domain \
         --domain-id <DOMAIN_ID> \
         --domain-settings-for-update
      RStudioServerProDomainSettingsForUpdate={DomainExecutionRoleArn=<DOMAIN_EXECUTION_ROLE_A
         --default-user-settings "{\"SecurityGroups\": [\"<SECURITY_GROUP>\",
      \"<SECURITY_GROUP>\"]}"
     ```

2. Verifikasi bahwa status Domain adalah`InService`. Setelah status Domain`InService`, dukungan untuk RStudio on SageMaker ditambahkan.

```
aws sagemaker \
    describe-domain \
    --domain-id <DOMAIN_ID>
```

3. Verifikasi bahwa status StudioServerPro aplikasi R `InService` menggunakan perintah berikut.

```
aws sagemaker list-apps --user-profile-name domain-shared
```

Langkah 4 - Tambahkan akses RStudio untuk pengguna yang ada

Sebagai bagian dari pembaruan di Langkah 3, SageMaker tandai RStudio AccessStatusdari semua profil pengguna yang ada di Domain sebagai `DISABLED` default. Ini mencegah melebihi jumlah pengguna yang diizinkan oleh lisensi Anda saat ini. Untuk menambahkan akses bagi pengguna yang ada, ada langkah keikutsertaan satu kali. Lakukan opt-in dengan memanggil UpdateUserProfileAPI dengan R StudioServerProAppSettings berikut:

- `AccessStatus` = `ENABLED`
- Opsional - `UserGroup` = `R_STUDIO_USER` atau `R_STUDIO_ADMIN`

```
aws sagemaker \
    update-user-profile \
    --domain-id <DOMAIN_ID>\
    --user-profile-name <USER_PROFILE> \
    --user-settings "{\"RStudioServerProAppSettings\": {\"AccessStatus\": \"ENABLED
\"}}"
```

> **Note**
>
> Secara default, jumlah pengguna yang dapat memiliki akses ke RStudio adalah 60.

Langkah 5 - Nonaktifkan akses RStudio untuk pengguna baru

Kecuali ditentukan lain saat memanggil`UpdateDomain`, dukungan RStudio ditambahkan secara default untuk semua profil pengguna baru yang dibuat setelah Anda menambahkan dukungan untuk

RStudio aktif. SageMaker Untuk menonaktifkan akses untuk profil pengguna baru, Anda harus secara eksplisit mengatur `AccessStatus` parameter `DISABLED` sebagai bagian dari panggilan API. `CreateUserProfile` Jika `AccessStatus` parameter tidak ditentukan sebagai bagian dari `CreateUserProfile` API, status akses default adalah`ENABLED`.

```
aws sagemaker \
    create-user-profile \
    --domain-id <DOMAIN_ID>\
    --user-profile-name <USER_PROFILE> \
    --user-settings "{\"RStudioServerProAppSettings\": {\"AccessStatus\": \"DISABLED
\"}}"
```

Bawa gambar Anda sendiri ke RStudio SageMaker

SEBUAH SageMaker image adalah file yang mengidentifikasi paket bahasa dan dependensi lain yang diperlukan untuk menjalankan RStudio di Amazon SageMaker. SageMaker menggunakan gambar-gambar ini untuk menciptakan lingkungan tempat Anda menjalankan RStudio. Amazon SageMaker menyediakan gambar RStudio bawaan untuk Anda gunakan. Jika Anda membutuhkan fungsionalitas yang berbeda, Anda dapat membawa gambar kustom Anda sendiri.

Proses untuk membawa gambar Anda sendiri untuk digunakan dengan RStudio aktif SageMaker mengambil tiga langkah:

1. Buat gambar khusus dari Dockerfile dan dorong gambar khusus ke repositori Amazon Elastic Container Registry (Amazon ECR).
2. Buat SageMaker gambar yang menunjuk ke gambar kontainer di Amazon ECR dan melampirkannya ke Amazon Anda SageMaker Domain.
3. Luncurkan sesi baru di RStudio dengan gambar kustom Anda.

Anda dapat membuat gambar dan versi gambar, dan melampirkan versi gambar ke Domain Anda, menggunakan SageMaker panel kontrol,AWS SDK for Python (Boto3), danAWS Command Line Interface(AWS CLI). Anda juga dapat membuat gambar dan versi gambar menggunakan SageMaker konsol, bahkan jika Anda belum onboard ke Domain.

Topik berikut menunjukkan cara membawa gambar Anda sendiri ke RStudio di SageMaker dengan membuat, melampirkan, dan meluncurkan gambar khusus.

Terminologi kunci

Bagian berikut mendefinisikan istilah kunci untuk membawa gambar Anda sendiri untuk digunakan dengan RStudio pada SageMaker.

- Dockerfile:Dockerfile adalah file yang mengidentifikasi paket bahasa dan dependensi lain untuk image Docker Anda.
- Gambar Docker:Gambar Docker adalah Dockerfile yang dibangun. Gambar ini diperiksa ke Amazon ECR dan berfungsi sebagai dasar SageMakergambar.
- SageMaker gambar:SEBUAH SageMaker gambar adalah pemegang untuk satu set SageMaker versi gambar berdasarkan gambar Docker.
- Versi gambar:Versi gambar dari SageMaker image mewakili image Docker yang kompatibel dengan RStudio dan disimpan dalam repositori Amazon ECR. Setiap versi gambar tidak dapat diubah. Versi gambar ini dapat dilampirkan ke domain dan digunakan dengan RStudio di SageMaker.

Prasyarat

Anda harus menyelesaikan prasyarat berikut sebelum membawa gambar Anda sendiri untuk digunakan dengan RStudio di Amazon SageMaker.

- Jika Anda memiliki Domain yang sudah ada dengan RStudio yang dibuat sebelum 7 April 2022, Anda harus menghapus RStudioServerPro aplikasi dan buat ulang. Untuk informasi tentang cara menghapus aplikasi, lihatMatikan dan Perbarui SageMaker Studio Classic.
- Instal aplikasi Docker. Untuk informasi tentang pengaturan Docker, lihatOrientasi dan pengaturan.
- Buat salinan lokal dari Dockerfile yang kompatibel dengan RStudio yang berfungsi SageMaker. Untuk informasi tentang membuat contoh RStudio dockerfile, lihatGunakan gambar khusus untuk membawa lingkungan pengembangan Anda sendiri ke RStudio di Amazon SageMaker.
- Gunakan sebuahAWS Identity and Access ManagementPeran eksekusi yang memilikiAmazonSageMakerFullAccesskebijakan terlampir. Jika Anda telah onboard ke Domain, Anda bisa mendapatkan peran dariRingkasan Domainbagian dari SageMaker panel kontrol.

  Tambahkan izin berikut untuk mengakses layanan Amazon Elastic Container Registry (Amazon ECR) ke peran eksekusi.

```
{
    "Version":"2012-10-17",
```

```
    "Statement":[
        {
            "Sid": "VisualEditor0",
            "Effect":"Allow",
            "Action":[
                "ecr:CreateRepository",
                "ecr:BatchGetImage",
                "ecr:CompleteLayerUpload",
                "ecr:DescribeImages",
                "ecr:DescribeRepositories",
                "ecr:UploadLayerPart",
                "ecr:ListImages",
                "ecr:InitiateLayerUpload",
                "ecr:BatchCheckLayerAvailability",
                "ecr:PutImage"
            ],
            "Resource": "*"
        }
    ]
}
```

- Instal dan konfigurasiAWS CLIdengan versi berikut (atau lebih tinggi). Untuk informasi tentang menginstalAWS CLI, lihatMenginstal atau memperbarui versi terbaru dariAWS CLI.

```
AWS CLI v1 >= 1.23.6
AWS CLI v2 >= 2.6.2
```

Spesifikasi gambar RStudio kustom

Dalam panduan ini, Anda akan mempelajari spesifikasi gambar RStudio khusus untuk digunakan saat Anda membawa gambar Anda sendiri. Ada dua set persyaratan yang harus Anda penuhi dengan gambar RStudio kustom Anda untuk menggunakannya dengan Amazon SageMaker. Persyaratan ini diberlakukan oleh RStudio PBC dan platform Amazon SageMaker Studio Classic. Jika salah satu dari rangkaian persyaratan ini tidak terpenuhi, maka gambar kustom Anda tidak akan berfungsi dengan baik.

Persyaratan PBC RStudio

Persyaratan PBC RStudio ditata dalam artikel Menggunakan gambar Docker dengan RStudio Workbench /RStudio Server Pro, Launcher, dan Kubernetes. Ikuti petunjuk dalam artikel ini untuk membuat dasar gambar RStudio kustom Anda.

Untuk petunjuk tentang cara menginstal beberapa versi R dalam gambar kustom Anda, lihat Menginstal beberapa versi R di Linux.

Persyaratan Amazon SageMaker Studio Classic

Amazon SageMaker Studio Classic memberlakukan serangkaian persyaratan instalasi berikut untuk gambar RStudio Anda.

- Anda harus menggunakan gambar dasar RStudio setidaknya`2023.03.2-454.pro2`. Untuk informasi selengkapnya, lihat Tingkatkan Versi RStudio.
- Anda harus menginstal paket-paket berikut:

```
yum install -y sudo \
openjdk-11-jdk \
libpng-dev \
&& yum clean all \
&& /opt/R/${R_VERSION}/bin/R -e "install.packages('reticulate', repos='https://
packagemanager.rstudio.com/cran/__linux__/centos7/latest')" \
&& /opt/python/${PYTHON_VERSION}/bin/pip install --upgrade \
    'boto3>1.0<2.0' \
    'awscli>1.0<2.0' \
    'sagemaker[local]<3'
```

- Anda harus memberikan nilai default untuk nilai `RSTUDIO_CONNECT_URL` dan `RSTUDIO_PACKAGE_MANAGER_URL` lingkungan.

```
ENV RSTUDIO_CONNECT_URL "YOUR_CONNECT_URL"
ENV RSTUDIO_PACKAGE_MANAGER_URL "YOUR_PACKAGE_MANAGER_URL"
```

Spesifikasi umum berikut berlaku untuk gambar yang diwakili oleh versi gambar RStudio.

Menjalankan gambar

`ENTRYPOINT`dan `CMD` instruksi diganti sehingga gambar dijalankan sebagai aplikasi RSession.

Menghentikan gambar

`DeleteApp`API mengeluarkan `docker stop` perintah yang setara. Proses lain dalam wadah tidak akan mendapatkan sinyal SIGKILL/SIGTERM.

Sistem file

`/opt/ml`Direktori `/opt/.sagemakerinternal` dan dicadangkan. Data apa pun di direktori ini mungkin tidak terlihat saat runtime.

Data pengguna

Setiap pengguna dalam SageMaker domain mendapatkan direktori pengguna pada volume Amazon Elastic File System bersama dalam gambar. Lokasi direktori pengguna saat ini pada volume Amazon Elastic File System adalah`/home/sagemaker-user`.

Metadata

File metadata terletak di. `/opt/ml/metadata/resource-metadata.json` Tidak ada variabel lingkungan tambahan yang ditambahkan ke variabel yang ditentukan dalam gambar. Untuk informasi selengkapnya, lihat Dapatkan Metadata Aplikasi.

GPU

Pada instance GPU, gambar dijalankan dengan `--gpus` opsi. Hanya toolkit CUDA yang harus disertakan dalam gambar, bukan driver NVIDIA. Untuk informasi selengkapnya, lihat Panduan Pengguna NVIDIA.

Metrik dan pencatatan

Log dari proses RSession dikirim ke Amazon CloudWatch di akun pelanggan. Nama grup log adalah`/aws/sagemaker/studio`. Nama aliran log adalah`$domainID/$userProfileName/RSession/$appName`.

Ukuran gambar

Ukuran gambar dibatasi hingga 25 GB. Untuk melihat ukuran gambar Anda, jalankan`docker image ls`.

Buat gambar RStudio kustom

Topik ini menjelaskan bagaimana Anda dapat membuat gambar RStudio kustom menggunakan SageMaker konsol dan. AWS CLI Jika Anda menggunakanAWS CLI, Anda harus menjalankan langkah-langkah dari mesin lokal Anda. Langkah-langkah berikut tidak berfungsi dari dalam Amazon SageMaker Studio Classic.

Saat Anda membuat gambar, SageMaker juga membuat versi gambar awal. Versi gambar mewakili gambar kontainer di Amazon Elastic Container Registry (ECR). Gambar kontainer harus memenuhi

persyaratan yang akan digunakan di RStudio. Untuk informasi selengkapnya, lihat Spesifikasi gambar RStudio kustom.

Untuk informasi tentang menguji gambar Anda secara lokal dan menyelesaikan masalah umum, lihat repo Sampel Gambar Kustom SageMaker Studio.

Topik

- Tambahkan image container RStudio Docker yang SageMaker kompatibel ke Amazon ECR
- Buat SageMaker gambar dari konsol
- Buat gambar dari AWS CLI

Tambahkan image container RStudio Docker yang SageMaker kompatibel ke Amazon ECR

Gunakan langkah-langkah berikut untuk menambahkan image container Docker ke Amazon ECR:

- Buat repositori Amazon ECR.
- Otentikasi ke Amazon ECR.
- Buat gambar RStudio Docker yang SageMaker kompatibel.
- Dorong gambar ke repositori Amazon ECR.

> **Note**
>
> Repositori Amazon ECR harus Wilayah AWS sama dengan domain Anda.

Untuk membangun dan menambahkan image Docker ke Amazon ECR

1. Buat repositori Amazon ECR menggunakan file. AWS CLI Untuk membuat repositori menggunakan konsol Amazon ECR, lihat Membuat repositori.

   ```
   aws ecr create-repository \
       --repository-name rstudio-custom \
       --image-scanning-configuration scanOnPush=true
   ```

   Respons:

   ```
   {
   ```

```
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-2:acct-id:repository/rstudio-custom",
        "registryId": "acct-id",
        "repositoryName": "rstudio-custom",
        "repositoryUri": "acct-id.dkr.ecr.us-east-2.amazonaws.com/rstudio-custom",
        ...
    }
}
```

2. Otentikasi ke Amazon ECR menggunakan URI repositori yang dikembalikan sebagai respons dari perintah. `create-repository` Pastikan aplikasi Docker berjalan. Untuk informasi selengkapnya, lihat Otentikasi Registri.

```
aws ecr get-login-password | \
    docker login --username AWS --password-stdin <repository-uri>
```

Respons:

```
Login Succeeded
```

3. membuat gambar Docker. Jalankan perintah berikut dari direktori yang menyertakan Dockerfile Anda.

```
docker build .
```

4. Tandai gambar buatan Anda dengan tag unik.

```
docker tag <image-id> "<repository-uri>:<tag>"
```

5. Dorong gambar kontainer ke repositori Amazon ECR. Untuk informasi selengkapnya, lihat ImagePushdan Mendorong gambar.

```
docker push <repository-uri>:<tag>
```

Respons:

```
The push refers to repository [<account-id>.dkr.ecr.us-east-2.amazonaws.com/
rstudio-custom]
r: digest: <digest> size: 3066
```

Buat SageMaker gambar dari konsol

Untuk membuat gambar

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Gambar.
4. Pada halaman Custom images, pilih Create image.
5. Untuk Sumber gambar, masukkan jalur registri ke gambar kontainer di Amazon ECR. Path dalam format berikut:

   ```
   acct-id.dkr.ecr.region.amazonaws.com/repo-name[:tag] or [@digest]
   ```
6. Pilih Berikutnya.
7. Di bawah properti Gambar, masukkan yang berikut ini:

   - Nama gambar — Nama harus unik untuk akun Anda saat iniWilayah AWS.
   - (Opsional) Nama tampilan gambar - Nama yang ditampilkan di antarmuka pengguna domain. Ketika tidak disediakan, `Image name` ditampilkan.
   - (Opsional) Deskripsi — Deskripsi gambar.
   - Peran IAM — Peran harus memiliki AmazonSageMakerFullAccesskebijakan yang dilampirkan. Gunakan menu tarik-turun untuk memilih salah satu opsi berikut:
     - Buat peran baru — Tentukan bucket Amazon Simple Storage Service (Amazon S3) tambahan yang ingin diakses oleh pengguna notebook. Jika Anda tidak ingin mengizinkan akses ke bucket tambahan, pilih Tidak Ada.

       SageMaker melekatkan `AmazonSageMakerFullAccess` kebijakan pada peran. Peran ini memungkinkan pengguna notebook Anda mengakses bucket Amazon S3 yang tercantum di sebelah tanda centang.
     - Masukkan peran IAM khusus ARN — Masukkan Nama Sumber Daya Amazon (ARN) peran IAM Anda.
     - Gunakan peran yang ada — Pilih salah satu peran yang ada dari daftar.
   - (Opsional) Tag gambar - Pilih Tambahkan tag baru. Anda dapat menambahkan hingga 50 tanda. Tag dapat dicari menggunakan SageMaker konsol atau API. SageMaker `Search`
8. Di bawah Jenis gambar, pilih gambar RStudio.
9. Pilih Kirim.

Gambar baru ditampilkan dalam daftar gambar Kustom dan disorot secara singkat. Setelah gambar berhasil dibuat, Anda dapat memilih nama gambar untuk melihat propertinya atau memilih Buat versi untuk membuat versi lain.

Untuk membuat versi gambar lain

1. Pilih Buat versi pada baris yang sama dengan gambar.
2. Untuk Sumber gambar, masukkan jalur registri ke gambar Amazon ECR. Gambar tidak boleh menjadi gambar yang sama seperti yang digunakan dalam versi SageMaker gambar sebelumnya.

Untuk menggunakan gambar kustom di RStudio, Anda harus melampirkannya ke domain Anda. Untuk informasi selengkapnya, lihat Lampirkan SageMaker gambar khusus.

Buat gambar dari AWS CLI

Bagian ini menunjukkan cara membuat SageMaker gambar Amazon khusus menggunakan fileAWS CLI.

Gunakan langkah-langkah berikut untuk membuat SageMaker gambar:

- Buat `Image`.
- Buat `ImageVersion`.
- Buat file konfigurasi.
- Buat `AppImageConfig`.

Untuk membuat entitas SageMaker gambar

1. Buat SageMaker gambar. Peran ARN harus memiliki setidaknya `AmazonSageMakerFullAccessPolicy` kebijakan terlampir.

   ```
   aws sagemaker create-image \
       --image-name rstudio-custom-image \
       --role-arn arn:aws:iam::<acct-id>:role/service-role/<execution-role>
   ```

   Respons:

   ```
   {
   ```

```
    "ImageArn": "arn:aws:sagemaker:us-east-2:acct-id:image/rstudio-custom-image"
}
```

2. Buat versi SageMaker gambar dari gambar. Berikan nilai tag unik yang Anda pilih saat Anda mendorong gambar ke Amazon ECR.

```
aws sagemaker create-image-version \
    --image-name rstudio-custom-image \
    --base-image <repository-uri>:<tag>
```

Respons:

```
{
    "ImageVersionArn": "arn:aws:sagemaker:us-east-2:acct-id:image-version/rstudio-
image/1"
}
```

3. Periksa apakah versi gambar berhasil dibuat.

```
aws sagemaker describe-image-version \
    --image-name rstudio-custom-image \
    --version 1
```

Respons:

```
{
    "ImageVersionArn": "arn:aws:sagemaker:us-east-2:acct-id:image-version/rstudio-
custom-image/1",
    "ImageVersionStatus": "CREATED"
}
```

> **Note**
>
> Jika responsnya`"ImageVersionStatus": "CREATED_FAILED"`, responsnya juga mencakup alasan kegagalan. Masalah izin adalah penyebab umum kegagalan. Anda juga dapat memeriksa CloudWatch Log Amazon Anda. Nama grup log adalah`/aws/sagemaker/studio`. Nama aliran log adalah`$domainID/$userProfileName/KernelGateway/$appName`.

4. Buat file konfigurasi, bernama`app-image-config-input.json`. Konfigurasi gambar aplikasi digunakan untuk konfigurasi untuk menjalankan SageMaker gambar sebagai aplikasi Kernel Gateway.

```
{
    "AppImageConfigName": "rstudio-custom-config"
}
```

5. Buat AppImageConfig menggunakan file yang Anda buat di langkah sebelumnya.

```
aws sagemaker create-app-image-config \
    --cli-input-json file://app-image-config-input.json
```

Respons:

```
{
    "AppImageConfigArn": "arn:aws:sagemaker:us-east-2:acct-id:app-image-config/r-
image-config"
}
```

Lampirkan SageMaker gambar khusus

Panduan ini menunjukkan cara melampirkan gambar RStudio kustom ke SageMaker Domain Amazon Anda menggunakan SageMaker konsol atau AWS Command Line Interface (AWS CLI).

Untuk menggunakan SageMaker gambar kustom, Anda harus melampirkan gambar RStudio kustom ke Domain Anda. Saat Anda melampirkan versi gambar, itu muncul di Peluncur RStudio dan tersedia di daftar dropdown Pilih gambar. Anda menggunakan dropdown untuk mengubah gambar yang digunakan oleh RStudio.

Ada batasan jumlah versi gambar yang dapat Anda lampirkan. Setelah Anda mencapai batas, Anda harus terlebih dahulu melepaskan versi sehingga Anda dapat melampirkan versi gambar yang berbeda.

Topik

- Melampirkan versi gambar ke Domain Anda menggunakan konsol
- Lampirkan versi gambar yang ada ke Domain Anda menggunakan AWS CLI

Melampirkan versi gambar ke Domain Anda menggunakan konsol

Anda dapat melampirkan versi SageMaker gambar kustom ke Domain menggunakan panel kontrol SageMaker konsol. Anda juga dapat membuat SageMaker gambar kustom, dan versi gambar, lalu melampirkan versi itu ke Domain Anda.

Untuk melampirkan gambar yang ada

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih domain yang diinginkan.
5. Pilih Lingkungan.
6. Di bawah gambar Custom SageMaker Studio Classic yang dilampirkan ke domain, pilih Lampirkan gambar.
7. Untuk Sumber gambar, pilih Gambar yang ada atau Gambar baru.

   Jika Anda memilih Gambar yang ada, pilih gambar dari toko SageMaker gambar Amazon.

   Jika Anda memilih Gambar baru, berikan jalur registri Amazon ECR untuk gambar Docker Anda. Jalur harus Wilayah AWS sama dengan Domain. Repo Amazon ECR harus berada di akun yang sama dengan Domain Anda, atau izin lintas akun untuk SageMaker harus diaktifkan.
8. Pilih gambar yang ada dari daftar.
9. Pilih versi gambar dari daftar.
10. Pilih Berikutnya.
11. Masukkan nilai untuk nama Gambar, Nama tampilan gambar, dan Deskripsi.
12. Pilih peran IAM. Untuk informasi selengkapnya, lihat Buat gambar RStudio kustom.
13. (Opsional) Tambahkan tag untuk gambar.
14. (Opsional) Pilih Tambahkan tag baru, lalu tambahkan tag konfigurasi.
15. Untuk jenis Gambar, pilih RStudio Image.
16. Pilih Kirim.

Tunggu hingga versi gambar dilampirkan ke Domain. Setelah versi dilampirkan, itu muncul di daftar Gambar khusus dan disorot secara singkat.

Lampirkan versi gambar yang ada ke Domain Anda menggunakan AWS CLI

Dua metode disajikan untuk melampirkan versi gambar ke Domain Anda menggunakanAWS CLI. Pada metode pertama, Anda membuat Domain baru dengan versi terlampir. Metode ini lebih sederhana tetapi Anda harus menentukan informasi Amazon Virtual Private Cloud (Amazon VPC) dan peran eksekusi yang diperlukan untuk membuat Domain.

Jika Anda sudah onboard ke Domain, Anda dapat menggunakan metode kedua untuk melampirkan versi gambar ke Domain Anda saat ini. Dalam hal ini, Anda tidak perlu menentukan informasi VPC Amazon dan peran eksekusi. Setelah Anda melampirkan versi, hapus semua aplikasi di Domain Anda dan luncurkan kembali RStudio.

Lampirkan SageMaker gambar ke Domain baru

Untuk menggunakan metode ini, Anda harus menentukan peran eksekusi yang memiliki AmazonSageMakerFullAccesskebijakan yang dilampirkan.

Gunakan langkah-langkah berikut untuk membuat Domain dan melampirkan SageMaker gambar kustom:

- Dapatkan ID VPC default dan ID subnet Anda.
- Buat file konfigurasi untuk Domain, yang menentukan gambar.
- Buat Domain dengan file konfigurasi.

Untuk menambahkan SageMaker gambar kustom ke Domain Anda

1. Dapatkan ID VPC default Anda.

```
aws ec2 describe-vpcs \
    --filters Name=isDefault,Values=true \
    --query "Vpcs[0].VpcId" --output text
```

   Respons:

```
vpc-xxxxxxxx
```

2. Dapatkan ID subnet default Anda menggunakan ID VPC dari langkah sebelumnya.

```
aws ec2 describe-subnets \
    --filters Name=vpc-id,Values=<vpc-id> \
```

```
    --query "Subnets[*].SubnetId" --output json
```

Respons:

```
[
    "subnet-b55171dd",
    "subnet-8a5f99c6",
    "subnet-e88d1392"
]
```

3. Buat file konfigurasi bernama`create-domain-input.json`. Masukkan ID VPC, ID subnet`ImageName`, dan `AppImageConfigName` dari langkah sebelumnya. Karena `ImageVersionNumber` tidak ditentukan, versi terbaru dari gambar digunakan, yang merupakan satu-satunya versi dalam kasus ini. Peran eksekusi Anda harus memenuhi persyaratan di[Prasyarat](#).

```
{
  "DomainName": "domain-with-custom-r-image",
  "VpcId": "<vpc-id>",
  "SubnetIds": [
    "<subnet-ids>"
  ],
  "DomainSettings": {
    "RStudioServerProDomainSettings": {
      "DomainExecutionRoleArn": "<execution-role>"
    }
  },
  "DefaultUserSettings": {
    "ExecutionRole": "<execution-role>",
    "RSessionAppSettings": {
      "CustomImages": [
        {
         "AppImageConfigName": "rstudio-custom-config",
         "ImageName": "rstudio-custom-image"
        }
      ]
     }
  },
  "AuthMode": "IAM"
}
```

4. Buat Domain dengan SageMaker gambar kustom terlampir.

```
aws sagemaker create-domain \
    --cli-input-json file://create-domain-input.json
```

Respons:

```
{
    "DomainArn": "arn:aws:sagemaker:region:acct-id:domain/domain-id",
    "Url": "https://domain-id.studio.region.sagemaker.aws/..."
}
```

Lampirkan SageMaker gambar ke Domain yang ada

Metode ini mengasumsikan bahwa Anda sudah onboard ke Domain. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.

> **Note**
>
> Anda harus menghapus semua aplikasi di Domain Anda untuk memperbarui Domain dengan versi gambar baru. Untuk informasi tentang menghapus aplikasi ini, lihatHapus SageMaker Domain Amazon.

Gunakan langkah-langkah berikut untuk menambahkan SageMaker gambar ke Domain Anda saat ini.

- Dapatkan `DomainID` dari SageMaker konsol.
- Gunakan `DomainID` untuk mendapatkan Domain. `DefaultUserSettings`
- Tambahkan `ImageName` dan `AppImageConfig` sebagai a `CustomImage` ke`DefaultUserSettings`.
- Perbarui Domain Anda untuk menyertakan gambar kustom.

Untuk menambahkan SageMaker gambar kustom ke Domain Anda

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.

4. Pilih domain yang diinginkan.
5. Pilih Setelan domain.
6. Di bawah Pengaturan Umum, temukan ID Domain. ID dalam format berikut:`d-xxxxxxxxxxxx`.
7. Gunakan ID Domain untuk mendapatkan deskripsi Domain.

```
aws sagemaker describe-domain \
    --domain-id <d-xxxxxxxxxxxx>
```

Respons:

```
{
    "DomainId": "d-xxxxxxxxxxxx",
    "DefaultUserSettings": {
      "KernelGatewayAppSettings": {
        "CustomImages": [
        ],
        ...
      }
    }
}
```

8. Simpan `DefaultUserSettings` bagian respons ke file bernama`update-domain-input.json`.
9. Masukkan `ImageName` dan `AppImageConfigName` dari langkah sebelumnya sebagai gambar khusus. Karena `ImageVersionNumber` tidak ditentukan, versi terbaru dari gambar digunakan, yang merupakan satu-satunya versi dalam kasus ini.

```
{
    "DefaultUserSettings": {
        "RSessionAppSettings": {
           "CustomImages": [
              {
                 "ImageName": "rstudio-custom-image",
                 "AppImageConfigName": "rstudio-custom-config"
              }
           ]
        }
    }
}
```

10. Gunakan ID Domain dan file pengaturan pengguna default untuk memperbarui Domain Anda.

```
aws sagemaker update-domain \
    --domain-id <d-xxxxxxxxxxxx> \
    --cli-input-json file://update-domain-input.json
```

Respons:

```
{
    "DomainArn": "arn:aws:sagemaker:region:acct-id:domain/domain-id"
}
```

11. Hapus `RStudioServerPro` aplikasi. Anda harus memulai ulang aplikasi `RStudioServerPro` bersama domain untuk UI Peluncur RStudio untuk mengambil perubahan terbaru.

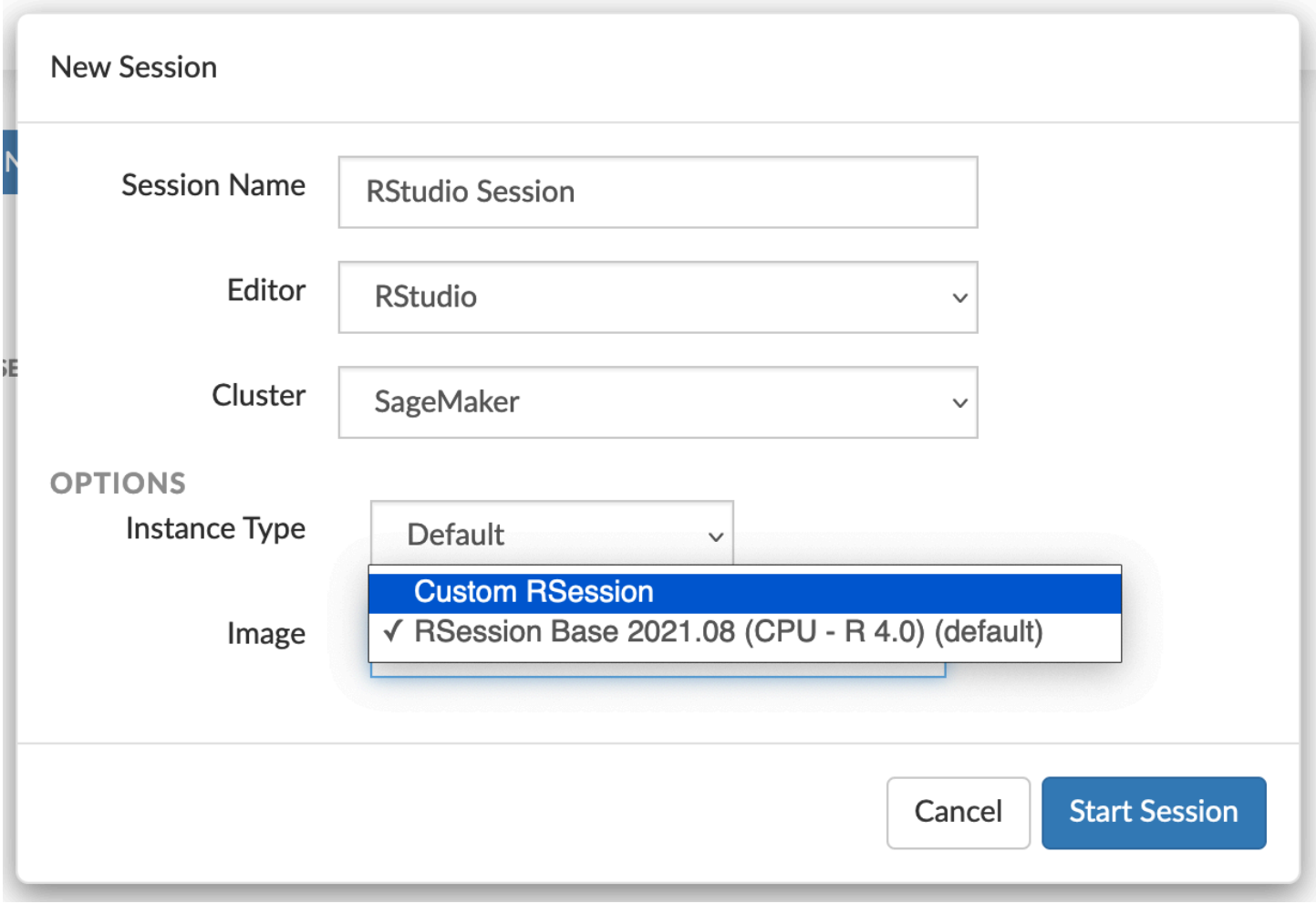
```
aws sagemaker delete-app \
    --domain-id <d-xxxxxxxxxxxx> --user-profile-name domain-shared \
    --app-type RStudioServerPro --app-name default
```

12. Buat `RStudioServerPro` aplikasi baru. Anda harus membuat aplikasi ini menggunakan fileAWS CLI.

```
aws sagemaker create-app \
    --domain-id <d-xxxxxxxxxxxx> --user-profile-name domain-shared \
    --app-type RStudioServerPro --app-name default
```

Luncurkan kustom SageMaker citra di RStudio

Anda dapat menggunakan gambar kustom Anda saat meluncurkan aplikasi RStudio dari konsol. Setelah Anda membuat kustom SageMaker gambar dan melampirkannya ke domain Anda, gambar muncul di kotak dialog pemilih gambar dari RStudio Launcher. Untuk meluncurkan aplikasi RStudio baru, ikuti langkah-langkah di[Buka RStudio Launcher dan luncurkan RSessions](#)dan pilih citra kustom Anda seperti yang ditunjukkan pada citra berikut.



Bersihkan sumber gambar

Panduan ini menunjukkan cara membersihkan sumber daya gambar RStudio yang Anda buat di bagian sebelumnya. Untuk menghapus gambar, selesaikan langkah-langkah berikut menggunakan salah satu SageMaker konsol atauAWS CLI, seperti yang ditunjukkan dalam panduan ini.

- Lepaskan versi gambar dan gambar dari Amazon Anda SageMaker Domain.
- Hapus gambar, versi gambar, dan konfigurasi gambar aplikasi.

Setelah Anda menyelesaikan langkah-langkah ini, Anda dapat menghapus gambar kontainer dan repositori dari Amazon ECR. Untuk informasi selengkapnya tentang cara menghapus gambar kontainer dan repositori, lihatMenghapus repositori.

Membersihkan sumber daya dari SageMakerkonsol

Saat Anda melepaskan gambar dari Domain, semua versi gambar akan terlepas. Ketika gambar terlepas, semua pengguna Domain kehilangan akses ke versi gambar.

Untuk melepaskan gambar

1. Buka Amazon SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilihKonfigurasi admin.
3. Di bawahKonfigurasi admin, pilihDomain.
4. Pilih domain yang diinginkan.
5. PilihLingkungan.
6. Di bawahGambar kustom yang dilampirkan ke domain, pilih gambar dan kemudian pilihLepaskan.
7. (Opsional) Untuk menghapus gambar dan semua versi SageMaker, pilihHapus juga gambar yang dipilih.... Ini tidak menghapus gambar terkait dari Amazon ECR.
8. Pilih Lepaskan.

Membersihkan sumber daya dariAWS CLI

Untuk membersihkan sumber daya

1. Lepaskan versi gambar dan gambar dari Domain Anda dengan meneruskan daftar gambar kustom kosong ke Domain. Bukaupdate-domain-input.jsonfile yang Anda buat diLampirkan SageMaker gambar ke domain Anda saat ini.
2. HapusRSessionAppSettingsgambar khusus dan kemudian simpan file. Jangan memodifikasiKernelGatewayAppSettingsgambar khusus.

```
{
    "DomainId": "d-xxxxxxxxxxxx",
    "DefaultUserSettings": {
      "KernelGatewayAppSettings": {
         "CustomImages": [
         ],
         ...
      },
      "RSessionAppSettings": {
        "CustomImages": [
```

```
        ],
        "DefaultResourceSpec": {
        }
        ...
      }
    }
}
```

3. Gunakan ID Domain dan file pengaturan pengguna default untuk memperbarui Domain Anda.

```
aws sagemaker update-domain \
    --domain-id <d-xxxxxxxxxxxx> \
    --cli-input-json file://update-domain-input.json
```

Jawaban:

```
{
    "DomainArn": "arn:aws:sagemaker:us-east-2:acct-id:domain/d-xxxxxxxxxxxx"
}
```

4. Hapus konfigurasi gambar aplikasi.

```
aws sagemaker delete-app-image-config \
    --app-image-config-name rstudio-image-config
```

5. Hapus SageMaker image, yang juga menghapus semua versi gambar. Gambar kontainer di Amazon ECR yang diwakili oleh versi gambar tidak dihapus.

```
aws sagemaker delete-image \
    --image-name rstudio-image
```

Mengelola pengguna

Setelah Domain SageMaker Amazon yang diaktifkan RStudio berjalan, Anda dapat menambahkan profil pengguna UserProfiles () ke Domain. Topik berikut menunjukkan cara membuat profil pengguna yang diizinkan untuk menggunakan RStudio, serta memperbarui profil pengguna yang ada. Untuk informasi tentang cara menghapus Aplikasi RStudio, UserProfile, atau Domain, ikuti langkah-langkah di Menghapus SageMaker Domain Amazon.

> **Note**
>
> Batas untuk jumlah total UserProfiles dalam SageMaker Domain Amazon adalah 60.

Ada dua jenis pengguna:

- Tidak sah: Pengguna ini tidak dapat mengakses aplikasi RStudio. Secara default, pengguna baru adalah `Unauthorized` jika Domain diaktifkan untuk RStudio.
- Resmi: Pengguna ini dapat mengakses aplikasi RStudio dan menggunakan salah satu kursi lisensi RStudio.

Jika pengguna diotorisasi, mereka dapat diberikan salah satu tingkat akses berikut ke RStudio.

- Pengguna RStudio: Ini adalah pengguna RStudio standar dan dapat mengakses RStudio.
- Admin RStudio: Admin SageMaker Domain Amazon Anda memiliki kemampuan untuk membuat pengguna, menambahkan pengguna yang ada, dan memperbarui izin pengguna yang ada. Admin juga dapat mengakses dasbor Administratif RStudio. Namun, admin ini tidak dapat memperbarui parameter yang dikelola oleh Amazon SageMaker.

Metode untuk membuat pengguna

Topik berikut menunjukkan cara membuat pengguna di Domain Amazon SageMaker berkemampuan RStudio Anda.

Buat konsol pengguna

Untuk membuat pengguna di Domain SageMaker Amazon berkemampuan RStudio dari konsol, selesaikan langkah-langkahnya. Pilih nama pengguna

Buat CLI pengguna

Perintah berikut menunjukkan cara menambahkan pengguna ke SageMaker Domain Amazon dengan otentikasi IAM. Pengguna dapat menjadi bagian dari grup `R_STUDIO_USER` atau `R_STUDIO_ADMIN` Pengguna.

```
aws sagemaker create-user-profile --region <REGION> \
    --domain-id <DOMAIN-ID> \
    --user-profile-name <USER_PROFILE_NAME-ID> \
```

```
    --user-settings RStudioServerProAppSettings={UserGroup=<USER-GROUP>}
```

Perintah berikut menunjukkan cara menambahkan pengguna ke SageMaker Domain Amazon dengan otentikasi menggunakan IAM Identity Center. Pengguna dapat menjadi bagian dari grup `R_STUDIO_USER` atau `R_STUDIO_ADMIN` Pengguna.

```
aws sagemaker create-user-profile --region <REGION> \
    --domain-id <DOMAIN-ID> \
    --user-profile-name <USER_PROFILE_NAME-ID> \
    --user-settings RStudioServerProAppSettings={UserGroup=<USER-GROUP>} \
    --single-sign-on-user-identifier UserName \
    --single-sign-on-user-value <USER-NAME>
```

Perbarui pengguna yang ada

Anda tidak dapat memperbarui otorisasi pengguna yang ada. Anda harus menghapus pengguna yang ada dan membuat yang baru dengan otorisasi yang diperbarui.

Masuk ke RStudio sebagai pengguna lain

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih domain yang berisi profil pengguna.
5. Pilih nama pengguna dari daftar pengguna. Ini membuka halaman baru dengan detail tentang profil pengguna dan aplikasi yang sedang berjalan.
6. Pilih Luncurkan.
7. Dari dropdown, pilih RStudio untuk meluncurkan instance RStudio.

Mengakhiri sesi untuk pengguna lain

1. Dari daftar aplikasi yang sedang berjalan, identifikasi aplikasi yang ingin Anda hapus.
2. Klik tombol Hapus aplikasi masing-masing untuk aplikasi yang Anda hapus.

Hapus pengguna lain

Anda tidak dapat menghapus pengguna jika pengguna menjalankan aplikasi apa pun. Hapus semua aplikasi sebelum mencoba menghapus pengguna.

1. Dari halaman Profil Pengguna, pilih Edit. Ini membuka halaman pengaturan Umum baru.
2. Di bawah Hapus pengguna, pilih Hapus pengguna.

Dasbor administratif RStudio

Topik ini menunjukkan cara mengakses dan menggunakan dasbor administratif RStudio. Dengan dasbor administratif RStudio, admin dapat mengelola pengguna dan RSessions, serta melihat informasi tentang pemanfaatan instans Server RStudio dan Log Amazon. CloudWatch

Luncurkan dasbor administratif RStudio

`R_STUDIO_ADMIN`Otorisasi memungkinkan pengguna untuk mengakses dasbor administratif RStudio. `R_STUDIO_ADMIN`Pengguna dapat mengakses dasbor administratif RStudio `workspaces` dengan mengganti dengan URL RStudio mereka `admin` secara manual. Berikut ini menunjukkan cara memodifikasi URL untuk mengakses dasbor administratif RStudio.

Misalnya, URL RStudio berikut:

```
https://<DOMAIN-ID>.studio.us-east-2.sagemaker.aws/rstudio/default/s/<SESSION-ID>/
workspaces
```

Dapat dikonversi ke:

```
https://<DOMAIN-ID>.studio.us-east-2.sagemaker.aws/rstudio/default/s/<SESSION-ID>/admin
```

Tab dasbor

Tab ini memberikan gambaran umum tentang pemanfaatan instans Server RStudio Anda, serta informasi tentang jumlah RSessions aktif.

Tab sesi

Tab ini memberikan informasi tentang RSessions aktif, seperti pengguna yang meluncurkan RSessions, waktu RSessions telah berjalan, dan pemanfaatan sumber daya mereka.

Tab pengguna

Tab ini memberikan informasi tentang pengguna resmi RStudio di Domain, seperti waktu RSession terakhir diluncurkan dan pemanfaatan sumber daya mereka.

Tab Statistik

Tab ini memberikan informasi tentang pemanfaatan instance Server RStudio Anda.

Tab log

Tab ini menampilkan CloudWatch Log Amazon untuk instance Server RStudio. Untuk informasi selengkapnya tentang peristiwa logging dengan Amazon CloudWatch Logs, lihat Apa itu Amazon CloudWatch Logs? .

Matikan dan restart RStudio

Untuk mematikan dan memulai ulang Posit Workbench Anda dan StudioServerPro aplikasi R terkait, Anda harus terlebih dahulu mematikan semua RSession yang ada. Anda dapat mematikan SessionGateway aplikasi R dari dalam RStudio. Anda kemudian dapat mematikan StudioServerPro aplikasi R menggunakanAWS CLI. Setelah StudioServerPro aplikasi R dimatikan, Anda harus membuka kembali RStudio melalui konsol. SageMaker

Setiap informasi notebook yang belum disimpan hilang dalam proses. Data pengguna dalam volume Amazon EFS tidak terpengaruh.

> **Note**
>
> Jika Anda menggunakan gambar kustom dengan RStudio, pastikan bahwa gambar buruh pelabuhan Anda menggunakan versi RStudio yang kompatibel dengan versi Posit Workbench yang digunakan oleh SageMaker setelah Anda me-restart aplikasi R Anda. StudioServerPro

Topik berikut menunjukkan cara mematikan aplikasi R SessionGateway dan R dan memulai ulang StudioServerPro aplikasi.

Tangguhkan RSessions Anda

Lengkapi prosedur berikut untuk menangguhkan semua RSession Anda.

1. Dari RStudio Launcher, identifikasi RSession yang ingin Anda tunda.
2. Pilih Suspend untuk sesi.
3. Ulangi ini untuk semua RSessions.

Hapus RSessions Anda

Lengkapi prosedur berikut untuk mematikan semua RSession Anda.

1. Dari RStudio Launcher, identifikasi RSession yang ingin Anda hapus.
2. Pilih Keluar untuk sesi. Ini akan membuka jendela Quit Session baru.
3. Dari jendela Keluar Sesi, pilih Paksa Keluar, untuk mengakhiri semua proses turunan dalam sesi.
4. Pilih Keluar Sesi untuk mengonfirmasi penghapusan sesi.
5. Ulangi ini untuk semua RSessions.

Menghapus StudioServerPro aplikasi R

Jalankan perintah berikut dari AWS CLI untuk menghapus dan memulai ulang StudioServerPro aplikasi R Anda.

1. Hapus StudioServerPro aplikasi R dengan menggunakan id domain Anda saat ini.

```
aws sagemaker delete-app \
    --domain-id <domainId> \
    --user-profile-name domain-shared \
    --app-type RStudioServerPro \
    --app-name default
```

2. Buat kembali StudioServerPro aplikasi R.

```
aws sagemaker create-app \
    --domain-id <domainId> \
    --user-profile-name domain-shared \
    --app-type RStudioServerPro \
    --app-name default
```

Manajemen penagihan dan biaya

Untuk melacak biaya yang terkait dengan lingkungan RStudio Anda, Anda dapat menggunakanAWS Billing and Cost Management layanan ini. AWS Billing and Cost Managementmenyediakan alat yang berguna untuk membantu Anda mengumpulkan informasi terkait dengan biaya dan penggunaan Anda, menganalisis pemicu biaya dan tren penggunaan Anda, serta mengambil tindakan untuk

menganggarkan pengeluaran Anda. Untuk informasi selengkapnya, lihatAWS Billing and Cost Management? .

Berikut ini menjelaskan komponen yang diperlukan untuk menjalankan RStudio di Amazon SageMaker dan bagaimana setiap faktor komponen menjadi penagihan untuk instans RStudio Anda.

- Lisensi RStudio -Anda harus membeli lisensi RStudio. Tidak ada biaya tambahan untuk menggunakan lisensi RStudio Anda dengan Amazon SageMaker. Untuk informasi selengkapnya tentang lisensi RStudio Anda, lihatLisensi RStudio.
- RSession - Ini adalah sesi kerja RStudio diluncurkan oleh pengguna akhir. Anda dikenakan biaya saat RSession sedang berjalan.
- RStudio Server - Sebuah server multi-penyewa mengelola semua RSessions. Anda dapat memilih jenis instans untuk menjalankan Server RStudio, dan membayar biaya terkait. Contoh default, “sistem”, gratis, tetapi Anda dapat memilih untuk membayar tingkatan yang lebih tinggi. Untuk informasi selengkapnya tentang tipe instans yang tersedia untuk Server RStudio Anda, lihatJenis StudioServerPro contoh R.

Pelacakan penagihan di tingkat pengguna

Untuk melacak penagihan di tingkat pengguna menggunakan Tag Alokasi Biaya, lihat Menggunakan Tag Alokasi Biaya.

Mendiagnosis masalah dan mendapatkan dukungan

Bagian berikut menjelaskan cara mendiagnosis masalah dengan RStudio di Amazon SageMaker. Untuk mendapatkan dukungan untuk RStudio di Amazon SageMaker, hubungi SageMaker dukungan Amazon. Untuk bantuan dengan membeli lisensi RStudio atau memodifikasi jumlah kursi lisensi, hubungi sales@rstudio.com.

Peningkatan versi Anda

Jika Anda menerima peringatan bahwa ada ketidakcocokan versi antaraStudioServerPro aplikasi RSession dan R, maka Anda harus meningkatkan versiStudioServerPro aplikasi R Anda. Untuk informasi selengkapnya, lihat Tingkatkan Versi RStudio.

Melihat Metrik

Anda dapat memantau kinerja alur kerja Anda saat menggunakan RStudio di Amazon SageMaker. Lihat log data dan informasi tentang metrik dengan dasbor administratif RStudio atau Amazon CloudWatch.

Melihat log RStudio Anda dari dasbor administratif RStudio

Anda dapat melihat metrik dan log langsung dari dasbor administratif RStudio.

1. Masuk ke SageMaker Domain Amazon Anda.
2. Arahkan ke dasbor administratif RStudio mengikuti langkah-langkah diDasbor administratif RStudio.
3. Pilih tab Log.

Melihat log RStudio Anda dari Amazon CloudWatch Logs

Amazon CloudWatch memantauAWS sumber daya Anda dan aplikasi yang Anda jalankanAWS di secara langsung. Anda dapat menggunakan Amazon CloudWatch untuk mengumpulkan dan menelusuri metrik, yang merupakan variabel yang dapat diukur untuk sumber daya dan aplikasi Anda. Untuk memastikan bahwa aplikasi RStudio Anda memiliki izin untuk Amazon CloudWatch, Anda harus menyertakan izin yang dijelaskan dalamIkhtisar SageMaker Domain Amazon. Anda tidak perlu melakukan pengaturan apa pun untuk mengumpulkan Amazon CloudWatch Logs.

Langkah-langkah berikut menunjukkan cara melihat Amazon CloudWatch Logs untuk RSession Anda.

Log ini dapat ditemukan di aliran`/aws/sagemaker/studio` log dariAWS CloudWatch konsol.

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pilih`Logs` dari sisi kiri. Dari menu pilihan menurun, pilih`Log groups`.
3. Di`Log groups` layar, cari`aws/sagemaker/studio`. Memilih grup Log.
4. Di`aws/sagemaker/studio` `Log group` layar, arahkan ke`Log streams` tab.
5. Untuk menemukan log untuk Domain Anda, cari`Log streams` menggunakan format berikut:

```
<DomainId>/domain-shared/rstudioserverpro/default
```

Menggunakan RStudio di Amazon SageMaker

Dengan dukungan RStudio di AmazonSageMaker, Anda dapat menempatkan alur kerja produksi Anda dan memanfaatkan fiturSageMaker. Topik berikut menunjukkan cara meluncurkan sesi RStudio dan menyelesaikan alur kerja kunci. Untuk informasi tentang mengelola RStudioSageMaker, lihatKelola RStudio di Amazon SageMaker.

Untuk informasi tentang langkah-langkah orientasi untuk membuat SageMaker Domain Amazon dengan RStudio diaktifkan, lihat. Ikhtisar SageMaker Domain Amazon

Untuk informasi tentang AWS Wilayah yang didukung oleh RStudio, lihatWilayah dan kuota yang didukung. SageMaker

Topik

- Berkolaborasi di RStudio
- Gambar dasar R
- Kolokasi aplikasi RSession
- Buka RStudio Launcher dan luncurkan RSessions
- Publikasikan ke RStudio Connect
- Akses SageMaker fitur Amazon dengan RStudio di Amazon SageMaker

Berkolaborasi di RStudio

Untuk berbagi proyek RStudio Anda, Anda dapat menghubungkan RStudio ke repo Git Anda. Untuk informasi tentang pengaturan ini, lihat Kontrol Versi dengan Git dan SVN.

Catatan: Berbagi proyek dan kolaborasi waktu nyata saat ini tidak didukung saat menggunakan RStudio di Amazon. SageMaker

Gambar dasar R

Saat meluncurkan instance RStudio Anda, image Base R berfungsi sebagai dasar instance Anda. Gambar ini memperluas gambar r-session-completeDocker.

Gambar Basis R ini meliputi:

- R v4.0 atau lebih tinggi
- `awscli`,`sagemaker`, dan paket `boto3` Python
- Paket reticulate untuk integrasi R SDK

Kolokasi aplikasi RSession

Pengguna dapat membuat beberapa aplikasi RSession pada contoh yang sama. Setiap jenis instans mendukung hingga empat aplikasi RSession yang terkolokasi. Ini berlaku untuk setiap

pengguna secara independen. Misalnya, jika dua pengguna membuat aplikasi, maka SageMaker mengalokasikan instance dasar yang berbeda untuk setiap pengguna. Masing-masing contoh ini akan mendukung 4 aplikasi RSession.

Pelanggan hanya membayar untuk jenis instans yang digunakan terlepas dari berapa banyak aplikasi Rsession yang berjalan pada instance. Jika pengguna membuat RSession dengan jenis contoh terkait yang berbeda, maka contoh yang mendasari baru dibuat.

Buka RStudio Launcher dan luncurkan RSessions

Topik berikut menunjukkan cara menggunakan RStudio Launcher untuk meluncurkan RSessions.

Buka Peluncur RStudio

Buka peluncur RStudio menggunakan serangkaian prosedur berikut yang cocok dengan lingkungan Anda.

Buka RStudio Launcher dari Amazon Console SageMaker

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Dari navigasi kiri, pilih RStudio.
3. Di bawah Memulai, pilih domain dan profil pengguna yang akan diluncurkan.
4. Pilih Luncurkan RStudio.

Buka RStudio Launcher dari Amazon Studio SageMaker

1. Arahkan ke Studio mengikuti langkah-langkah diLuncurkan Amazon SageMaker Studio.
2. Di bawah Aplikasi, pilih RStudio.
3. Dari halaman landing RStudio, pilih Luncurkan aplikasi.

Buka RStudio Launcher dari AWS CLI

Prosedur untuk membuka Peluncur RStudio menggunakan AWS CLI perbedaan tergantung pada metode yang digunakan untuk mengelola pengguna Anda.

Pusat Identitas IAM

1. Gunakan portal AWS akses untuk membuka SageMaker Domain Amazon Anda.

2. Ubah path URL ke “/rstudio/default” sebagai berikut.

```
#Studio URL
https://<domain-id>.studio.<region>.sagemaker.aws/jupyter/default/lab

#modified URL
https://<domain-id>.studio.<region>.sagemaker.aws/rstudio/default
```

IAM

Untuk membuka RStudio Launcher dari AWS CLI dalam mode IAM, selesaikan prosedur berikut.

1. Buat URL presigned menggunakan perintah berikut.

```
aws sagemaker create-presigned-domain-url --region <REGION> \
    --domain-id <DOMAIN-ID> \
    --user-profile-name <USER-PROFILE-NAME>
```

2. Tambahkan &redirect=R StudioServerPro ke URL yang dihasilkan.
3. Arahkan ke URL yang diperbarui.

Luncurkan RSessions

Setelah meluncurkan RStudio Launcher, Anda dapat membuat RSession baru.

1. Pilih Sesi Baru.
2. Masukkan Nama Sesi.
3. Pilih jenis instans yang dijalankan RSession Anda. Ini default ke. `ml.t3.medium`
4. Pilih Gambar yang digunakan RSession Anda sebagai kernel.
5. Pilih Mulai Sesi.
6. Setelah sesi Anda dibuat, Anda dapat memulainya dengan memilih nama.

> **Note**
>
> Jika Anda menerima peringatan bahwa ada ketidakcocokan versi antara aplikasi RSession dan R Anda, maka Anda harus memutakhirkan versi StudioServerPro aplikasi R StudioServerPro Anda. Untuk informasi selengkapnya, lihat Tingkatkan Versi RStudio.

Tangguhkan RSessions Anda

1. Dari RStudio Launcher, identifikasi RSession yang ingin Anda tangguhkan.
2. Pilih Tangguhkan untuk sesi.

Hapus RSessions Anda

1. Dari RStudio Launcher, identifikasi RSession yang ingin Anda hapus.
2. Pilih Keluar untuk sesi. Ini membuka jendela Sesi Keluar baru.
3. Dari jendela Keluar Sesi, pilih Paksa Keluar, untuk mengakhiri semua proses anak dalam sesi.
4. Pilih Keluar Sesi untuk mengonfirmasi penghapusan sesi.

Publikasikan ke RStudio Connect

RStudio Connect memungkinkan ilmuwan data untuk mempublikasikan wawasan, dasbor, dan aplikasi web dari RStudio di Amazon SageMaker. Untuk informasi selengkapnya, lihatHost RStudio Connect dan Package Manager untuk pengembangan ML di RStudio di Amazon SageMaker.

Untuk informasi selengkapnya tentang RStudio Connect, lihatPanduan Pengguna RStudio Connect.

Akses SageMaker fitur Amazon dengan RStudio di Amazon SageMaker

Salah satu manfaat menggunakan RStudio di Amazon SageMaker adalah integrasi SageMaker fitur Amazon. Ini termasuk integrasi dengan Amazon SageMaker Studio Classic dan Reticulate.

Gunakan Amazon SageMaker Studio Classic dan RStudio di Amazon SageMaker

Instans Amazon SageMaker Studio Classic dan RStudio Anda berbagi sistem file Amazon EFS yang sama. Ini berarti file yang Anda impor dan buat menggunakan Studio Classic dapat diakses menggunakan RStudio dan sebaliknya. Ini memungkinkan Anda untuk bekerja pada file yang sama menggunakan Studio Classic dan RStudio tanpa harus memindahkan file Anda di antara keduanya. Untuk informasi selengkapnya tentang alur kerja ini, lihat blog Mengumumkan RStudio yang Dikelola Sepenuhnya di Amazon SageMaker untuk Ilmuwan Data.

Gunakan Amazon SageMaker SDK dengan retikulasi

Paket reticulate digunakan sebagai antarmuka R ke Amazon SageMaker Python SDK untuk melakukan panggilan API ke Amazon. SageMaker Paket reticulate menerjemahkan antara objek R

dan Python, dan Amazon SageMaker menyediakan lingkungan ilmu data tanpa server untuk melatih dan menerapkan model Machine Learning (ML) dalam skala besar. Untuk informasi umum tentang paket retikulasi, lihat R Interface to Python.

Untuk blog yang menguraikan cara menggunakan paket retikulasi dengan Amazon SageMaker, lihat Menggunakan R dengan Amazon. SageMaker

Contoh berikut menunjukkan cara menggunakan retikulat untuk kasus penggunaan tertentu.

- Untuk buku catatan yang menjelaskan cara menggunakan retikulat untuk melakukan transformasi batch guna membuat prediksi, lihat Transformasi Batch Menggunakan R dengan Amazon. SageMaker
- Untuk buku catatan yang menjelaskan cara menggunakan retikulat untuk melakukan penyetelan hiperparameter dan menghasilkan prediksi, lihat Optimasi Hyperparameter Menggunakan R dengan Amazon. SageMaker

Memulai Editor Kode di Amazon SageMaker Studio

Code Editor, berdasarkan Code-OSS, Visual Studio Code - Open Source, membantu Anda menulis, menguji, men-debug, dan menjalankan analisis dan kode pembelajaran mesin Anda. Editor Kode meluas dan terintegrasi penuh dengan Amazon SageMaker Studio. Ini juga mendukung ekstensi lingkungan pengembangan terintegrasi (IDE) yang tersedia di Open VSX Registry.

Code Editor memiliki ekstensi AWSToolkit for VS Code yang sudah diinstal sebelumnya, yang memungkinkan koneksi Layanan AWS ke Amazon CodeWhispererseperti, tujuan umum, generator kode bertenaga pembelajaran mesin yang menyediakan rekomendasi kode secara real time. Untuk informasi selengkapnya tentang ekstensi, lihatKoneksi dan Ekstensi Editor Kode.

> **Important**
>
> Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Untuk meluncurkan Editor Kode, buat ruang pribadi Editor Kode. Ruang Editor Kode menggunakan satu instans Amazon Elastic Compute Cloud (Amazon EC2) untuk komputasi Anda dan satu volume

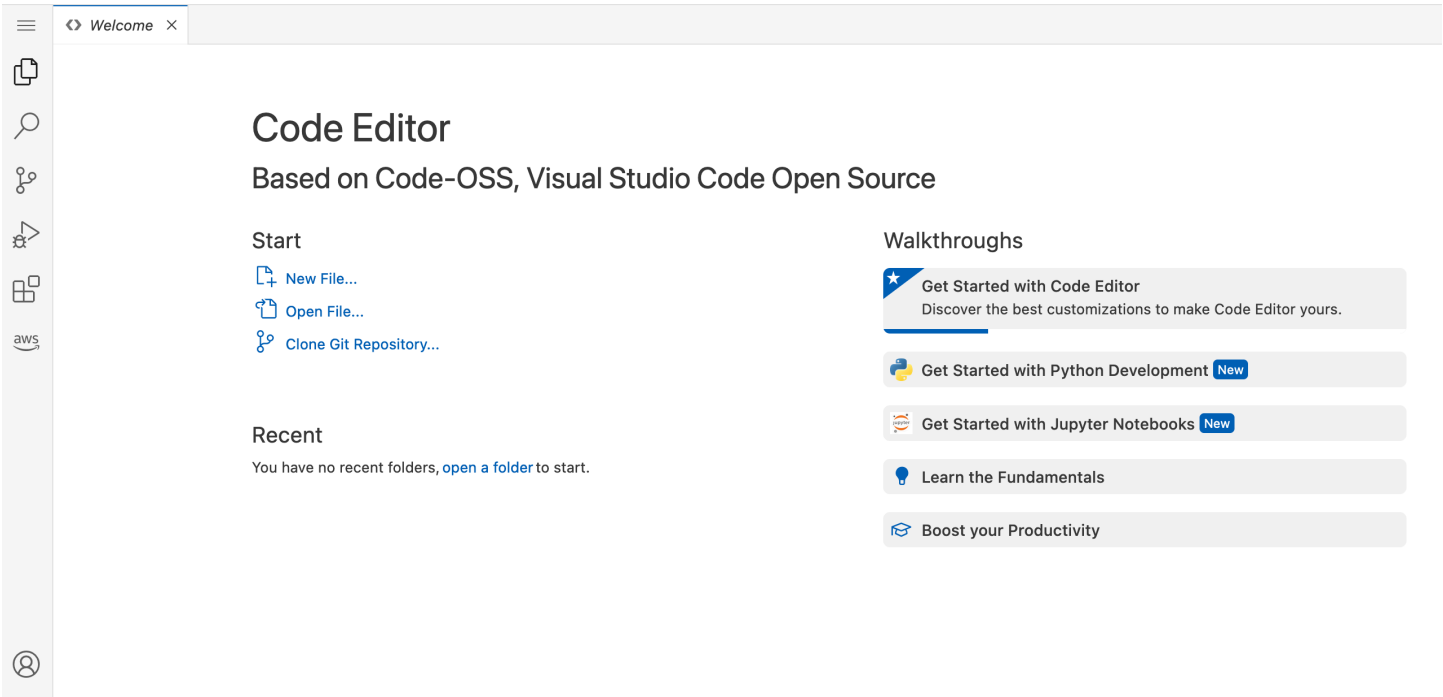
Amazon Elastic Block Store (Amazon EBS) untuk penyimpanan Anda. Segala sesuatu di ruang Anda seperti kode, profil Git, dan variabel lingkungan disimpan pada volume Amazon EBS yang sama. Volume memiliki 3000 IOPS dan throughput 125 MBps. Administrator Anda telah mengonfigurasi pengaturan penyimpanan Amazon EBS default untuk ruang Anda.

Ukuran penyimpanan default adalah 5 GB, tetapi administrator Anda dapat meningkatkan jumlah ruang yang Anda dapatkan. Untuk informasi selengkapnya, lihat Ubah ukuran penyimpanan default.

Anda dapat menskalakan komputasi Anda ke atas atau ke bawah dengan mengubah jenis instans Amazon EC2 yang menjalankan aplikasi Editor Kode Anda. Sebelum Anda mengubah jenis instance terkait, Anda harus terlebih dahulu menghentikan ruang Editor Kode Anda. Untuk informasi selengkapnya, lihat Contoh dan gambar aplikasi Editor Kode.

Administrator Anda mungkin memberi Anda konfigurasi siklus hidup untuk menyesuaikan lingkungan Anda. Anda dapat menentukan konfigurasi siklus hidup saat Anda membuat ruang. Untuk informasi selengkapnya, lihat Konfigurasi siklus hidup Editor Kode.

Anda juga dapat membawa sistem penyimpanan file Anda sendiri jika Anda memiliki volume Amazon EFS.



Topik

- Panduan pengguna Editor Kode
- Panduan adminstrator Editor Kode

Panduan pengguna Editor Kode

Topik di bagian ini menyediakan panduan untuk menggunakan Editor Kode, termasuk cara meluncurkan, menambahkan koneksiLayanan AWS, mematikan sumber daya, dan banyak lagi. Setelah membuat ruang Editor Kode, Anda dapat mengakses sesi Editor Kode Anda langsung melalui browser.

Dalam lingkungan Editor Kode Anda, Anda dapat melakukan hal berikut:

- Akses semua artefak yang ada di direktori home Anda
- Kloning GitHub repositori Anda dan komit perubahan
- Akses SageMaker Python SDK

Anda dapat kembali ke Studio untuk meninjau aset apa pun yang dibuat di lingkungan Editor Kode seperti eksperimen, pipeline, atau pekerjaan pelatihan.

Topik

Periksa versi Editor Kode

Langkah-langkah berikut menunjukkan cara memeriksa versi aplikasi Editor Kode Anda.

Untuk memeriksa versi aplikasi Editor Kode

1. Luncurkan dan jalankan ruang Editor Kode dan arahkan ke UI aplikasi Editor Kode. Untuk informasi selengkapnya, lihat Luncurkan aplikasi Editor Kode di Studio.

2. Di sudut kiri atas UI Editor Kode, pilih tombol menu ().

Kemudian, pilih Bantuan. Kemudian, pilih Tentang.

> **Note**
>
> Rilis SageMaker Code Editor saat ini didasarkan pada versi 1.83.1 dariCode-OSS, Visual Studio Code -. Open Source

Contoh dan gambar aplikasi Editor Kode

Hanya beberapa contoh yang kompatibel dengan aplikasi Editor Kode. Anda dapat memilih jenis instance yang kompatibel dengan kasus penggunaan Anda dari menu tarik-turun Instance.

Instans peluncuran cepat dimulai jauh lebih cepat daripada instance lainnya. Untuk informasi selengkapnya tentang jenis instans peluncuran cepat di Studio,Jenis Instans Studio Klasik yang Tersedia.

> **Note**
>
> Jika Anda menggunakan tipe instans GPU saat mengonfigurasi aplikasi Editor Kode, Anda juga harus menggunakan gambar berbasis GPU. UI ruang Editor Kode secara otomatis memilih gambar yang kompatibel saat Anda memilih jenis instans.

Dalam suatu ruang, data Anda disimpan dalam volume Amazon EBS yang bertahan secara independen dari masa pakai instans. Anda tidak akan kehilangan data Anda ketika Anda mengubah instance. Jika ruang Editor Kode Anda`Running`, Anda harus menghentikan ruang Anda sebelum mengubah jenis instance.

Tabel berikut mencantumkan ARN dari CPU Editor Kode dan gambar GPU yang tersedia untuk setiap Wilayah.

Wilayah	CPU	GPU

us-east-1	arn:aws:sagemaker:us-east-1:885854791233:image/sagemaker-distribution-cpu	arn:aws:sagemaker:us-east-1:885854791233:image/sagemaker-distribution-gpu
us-east-2	arn:aws:sagemaker:us-east-2:37914896644:image/sagemaker-distribution-cpu	arn:aws:sagemaker:us-east-2:37914896644:image/sagemaker-distribution-gpu
us-west-1	arn:aws:sagemaker:us-west-1:053634841547:image/sagemaker-distribution-cpu	arn:aws:sagemaker:us-west-1:053634841547:image/sagemaker-distribution-gpu
us-west-2	arn:aws:sagemaker:us-west-2:542918446943:image/sagemaker-distribution-cpu	arn:aws:sagemaker:us-west-2:542918446943:image/sagemaker-distribution-gpu
af-south-1	arn:aws:sagemaker:af-south-1:238384257742:image/sagemaker-distribution-cpu	arn:aws:sagemaker:af-south-1:238384257742:image/sagemaker-distribution-gpu
ap-east-1	arn:aws:sagemaker:ap-east-1:523751269255:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-east-1:523751269255:image/sagemaker-distribution-gpu
ap-south-1	arn:aws:sagemaker:ap-south-1:245090515133:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-south-1:245090515133:image/sagemaker-distribution-gpu
ap-northeast-2	arn:aws:sagemaker:ap-northeast-2:064688005998:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-northeast-2:064688005998:image/sagemaker-distribution-gpu
ap-southeast-1	arn:aws:sagemaker:ap-southeast-1:022667117163:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-southeast-1:022667117163:image/sagemaker-distribution-gpu
ap-southeast-2	arn:aws:sagemaker:ap-southeast-2:648430277019:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-southeast-2:648430277019:image/sagemaker-distribution-gpu

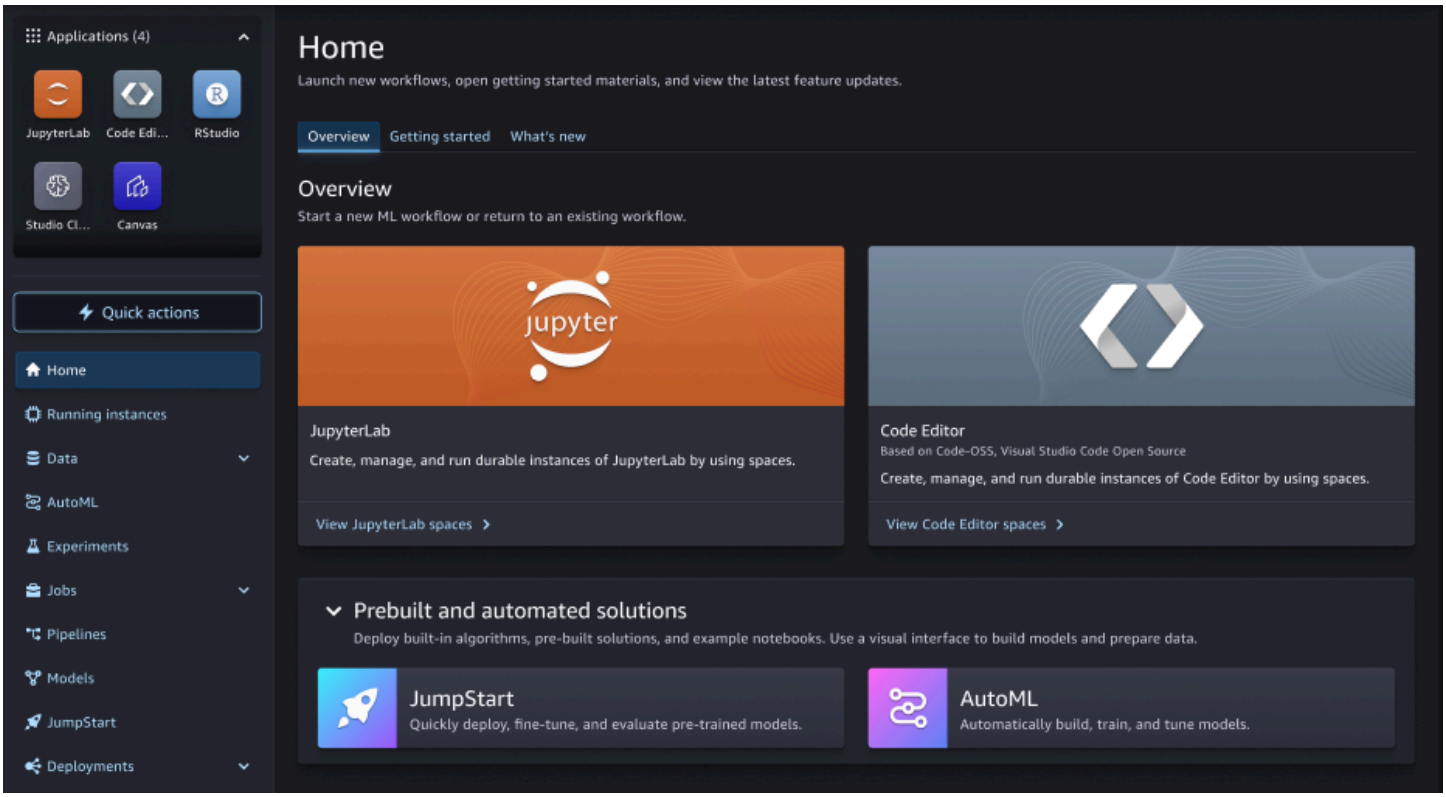
ap-northeast-1	arn:aws:sagemaker:ap-northeast-1:010972774902:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-northeast-1:010972774902:image/sagemaker-distribution-gpu
ca-central-1	arn:aws:sagemaker:ca-central-1:481561238223:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ca-central-1:481561238223:image/sagemaker-distribution-gpu
eu-central-1	arn:aws:sagemaker:eu-central-1:545423591354:image/sagemaker-distribution-cpu	arn:aws:sagemaker:eu-central-1:545423591354:image/sagemaker-distribution-gpu
eu-west-1	arn:aws:sagemaker:eu-west-1:819792524951:image/sagemaker-distribution-cpu	arn:aws:sagemaker:eu-west-1:819792524951:image/sagemaker-distribution-gpu
eu-west-2	arn:aws:sagemaker:eu-west-2:021081402939:image/sagemaker-distribution-cpu	arn:aws:sagemaker:eu-west-2:021081402939:image/sagemaker-distribution-gpu
eu-west-3	arn:aws:sagemaker:eu-west-3:856416204555:image/sagemaker-distribution-cpu	arn:aws:sagemaker:eu-west-3:856416204555:image/sagemaker-distribution-gpu
eu-north-1	arn:aws:sagemaker:eu-north-1:175620155138:image/sagemaker-distribution-cpu	arn:aws:sagemaker:eu-north-1:175620155138:image/sagemaker-distribution-gpu
eu-south-1	arn:aws:sagemaker:eu-south-1:810671768855:image/sagemaker-distribution-cpu	arn:aws:sagemaker:eu-south-1:810671768855:image/sagemaker-distribution-gpu
sa-east-1	arn:aws:sagemaker:sa-east-1:567556641782:image/sagemaker-distribution-cpu	arn:aws:sagemaker:sa-east-1:567556641782:image/sagemaker-distribution-gpu
ap-northeast-3	arn:aws:sagemaker:ap-northeast-3:564864627153:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-northeast-3:564864627153:image/sagemaker-distribution-gpu

ap-southeast-3	arn:aws:sagemaker:ap-southeast-3:370607712162:image/sagemaker-distribution-cpu	arn:aws:sagemaker:ap-southeast-3:370607712162:image/sagemaker-distribution-gpu
me-south-1	arn:aws:sagemaker:me-south-1:523774347010:image/sagemaker-distribution-cpu	arn:aws:sagemaker:me-south-1:523774347010:image/sagemaker-distribution-gpu
me-central-1	arn:aws:sagemaker:me-central-1:358593528301:image/sagemaker-distribution-cpu	arn:aws:sagemaker:me-central-1:358593528301:image/sagemaker-distribution-gpu
il-central-1	arn:aws:sagemaker:il-central-1:080319125002:image/sagemaker-distribution-cpu	arn:aws:sagemaker:il-central-1:080319125002:image/sagemaker-distribution-gpu
cn-north-1	arn:aws:sagemaker:cn-north-1:674439102856:image/sagemaker-distribution-cpu	arn:aws:sagemaker:cn-north-1:674439102856:image/sagemaker-distribution-gpu
cn-northwest-1	arn:aws:sagemaker:cn-northwest-1:651871951035:image/sagemaker-distribution-cpu	arn:aws:sagemaker:cn-northwest-1:651871951035:image/sagemaker-distribution-gpu
us-gov-west-1	arn:aws:sagemaker:us-gov-west-1:300992924816:image/sagemaker-distribution-cpu	arn:aws:sagemaker:us-gov-west-1:300992924816:image/sagemaker-distribution-gpu
us-gov-east-1	arn:aws:sagemaker:us-gov-east-1:300993876623:image/sagemaker-distribution-cpu	arn:aws:sagemaker:us-gov-east-1:300993876623:image/sagemaker-distribution-gpu

Jika Anda menemukan batas instans, hubungi administrator Anda. Untuk mendapatkan lebih banyak penyimpanan dan komputasi untuk pengguna, administrator dapat meminta peningkatan kuota penggunaAWS. Untuk informasi selengkapnya tentang meminta peningkatan kuota, lihat SageMaker titik akhir dan kuota Amazon.

Luncurkan aplikasi Editor Kode di Studio

Untuk mengonfigurasi dan mengakses lingkungan pengembangan terintegrasi Editor Kode Anda melalui Studio, Anda harus membuat ruang Editor Kode. Untuk informasi selengkapnya tentang spasi di Studio, lihatRuang Amazon SageMaker Studio.



Prosedur berikut menunjukkan cara membuat dan menjalankan ruang Editor Kode.

Untuk membuat dan menjalankan ruang Editor Kode

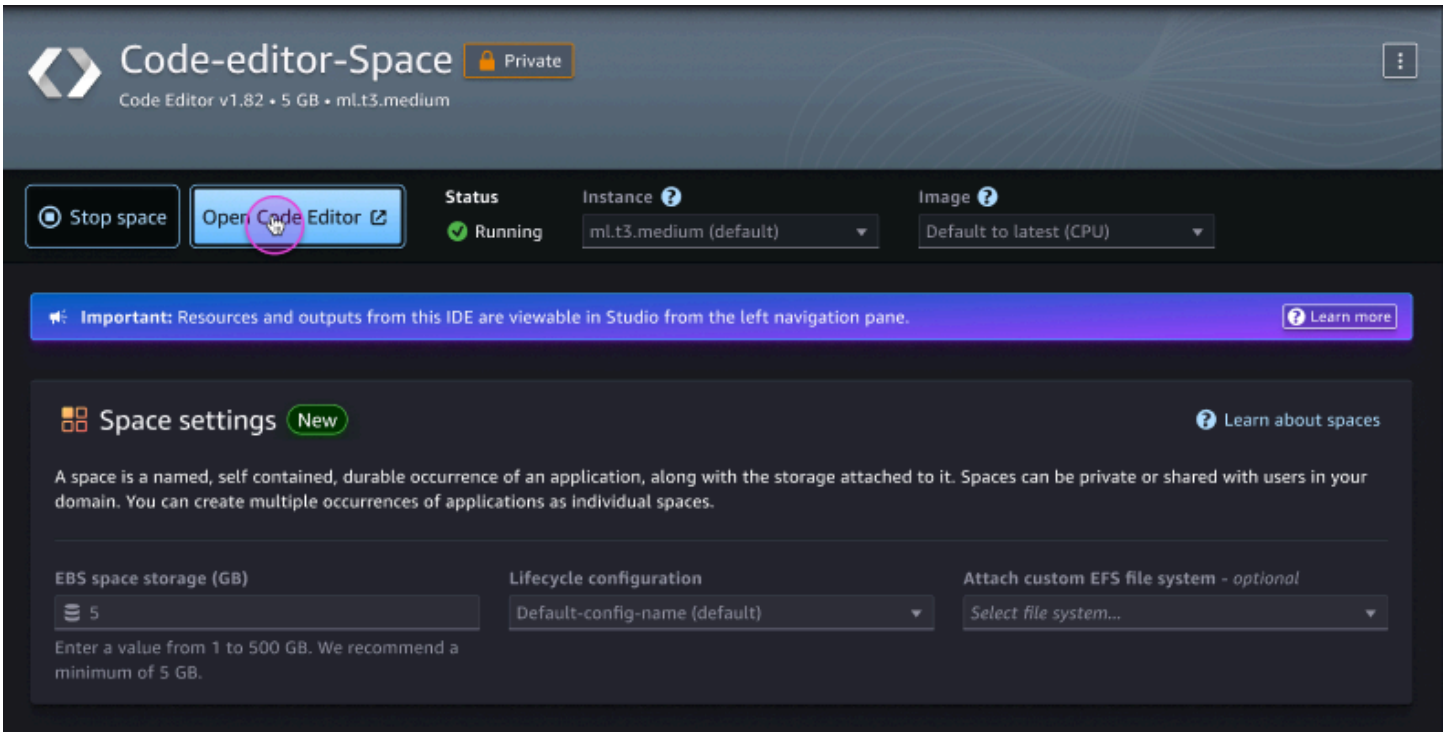
1. Luncurkan pengalaman Studio yang diperbarui. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio.
2. Lakukan salah satu dari cara berikut:
   - Dalam UI Amazon SageMaker Studio yang diperbarui, pilih Editor Kode dari menu Aplikasi.
   - Dalam UI Amazon SageMaker Studio yang diperbarui, pilih Lihat spasi Editor Kode di bagian Ikhtisar beranda Studio.
3. Di sudut kanan atas halaman landing Code Editor, pilih ruang Create Code Editor.
4. Masukkan nama untuk ruang Editor Kode Anda. Nama harus 1—62 karakter dengan menggunakan huruf, angka, dan tanda hubung saja.
5. Pilih Buat ruang.

6. Setelah ruang dibuat, Anda memiliki beberapa opsi sebelum Anda memilih untuk menjalankan ruang:
   - Anda dapat mengedit pengaturan sistem file Storage (GB), Lifecycle Configuration, atau Attach custom EFS filesystem. Opsi untuk pengaturan ini tersedia berdasarkan spesifikasi administrator.
   - Dari menu tarik-turun Instance, Anda dapat memilih jenis instance yang paling kompatibel dengan kasus penggunaan Anda.

     Jika Anda menggunakan tipe instans GPU saat mengonfigurasi aplikasi Editor Kode, Anda juga harus menggunakan gambar berbasis GPU. Dalam suatu ruang, data Anda disimpan dalam volume Amazon EBS yang bertahan secara independen dari masa pakai instans. Anda tidak akan kehilangan data Anda ketika Anda mengubah instance.

   > **Note**
   >
   > Untuk memperbarui pengaturan ruang, Anda harus terlebih dahulu menghentikan ruang Anda.

7. Setelah memperbarui pengaturan Anda, pilih Jalankan Spasi di halaman detail spasi.
8. Setelah status spasi`Running`, pilih Buka Editor Kode untuk pergi ke sesi Editor Kode Anda.



Dalam halaman landing Code Editor Studio, Anda dapat memfilter dan mengelola ruang yang ada.

Untuk mengelola ruang Editor Kode

1. Arahkan ke halaman landing Code Editor Studio dan filter ruang Editor Kode Anda dengan Private to me atau Running.
2. Lakukan salah satu dari cara berikut:
   - Di halaman landing Code Editor Studio, di baris nama spasi pilihan Anda, Anda dapat Berhenti, Mulai, atau Buka ruang itu di kolom Tindakan.
   - Pilih nama spasi di halaman landing Code Editor Studio. Ini membawa Anda ke halaman detail ruang di mana Anda juga dapat Berhenti, Mulai, atau Buka ruang itu atau perbarui pengaturan ruang.

Luncurkan aplikasi Editor Kode menggunakan AWS CLI

Untuk mengonfigurasi dan mengakses lingkungan pengembangan terintegrasi Editor Kode Anda melalui AWS Command Line Interface (AWS CLI), Anda harus membuat ruang Editor Kode. Pastikan untuk memenuhi Prasyarat sebelum melalui langkah-langkah berikut. Gunakan prosedur berikut untuk membuat dan menjalankan ruang Editor Kode.

Untuk membuat dan menjalankan ruang Editor Kode

1. Akses ruang menggunakan AWS Identity and Access Management (IAM) atau AWS IAM Identity Center otentikasi. Untuk informasi selengkapnya tentang mengakses spasi menggunakanAWS CLI, lihat Mengakses spasi menggunakan in. AWS Command Line Interface Ruang Amazon SageMaker Studio
2. Buat aplikasi dan tentukan `CodeEditor` sebagai `app-type` menggunakan perintah berikut.

   Jika Anda menggunakan jenis instans GPU saat membuat aplikasi Editor Kode, Anda juga harus menggunakan gambar berbasis GPU.

```
aws sagemaker create-app \
--domain-id domain-id \
--space-name space-name \
--app-type CodeEditor \
--app-name default \
--resource-spec "SageMakerImageArn=arn:aws:sagemaker:region:account-
id:image/sagemaker-distribution-cpu"
```

Untuk informasi selengkapnya tentang ARN gambar Editor Kode yang tersedia, lihatContoh dan gambar aplikasi Editor Kode.

3. Setelah aplikasi Code Editor dalam layanan, luncurkan aplikasi menggunakan URL yang telah ditentukan sebelumnya. Anda dapat menggunakan `describe-app` API untuk memeriksa apakah aplikasi Anda dalam layanan. Gunakan `create-presigned-domain-url` API untuk membuat URL presigned:

```
aws sagemaker create-presigned-domain-url \
--domain-id domain-id \
--space-name space-name \
--user-profile-name user-profile-name \
--session-expiration-duration-in-seconds 43200 \
--landing-uri app:CodeEditor:
```

4. Buka URL yang dihasilkan untuk mulai bekerja di aplikasi Editor Kode Anda.

Mengkloning repositori di Editor Kode

Anda dapat menavigasi melalui folder dan mengkloning repositori di jendela Explorer UI aplikasi Editor Kode.

Untuk mengkloning repositori, lakukan langkah-langkah berikut:

Untuk mengkloning repositori

1. Buka aplikasi Editor Kode Anda di browser, dan pilih tombol Eksplorasi () di panel navigasi kiri.
2. Pilih Clone Repository di jendela Explorer. Kemudian, berikan URL repositori atau pilih sumber repositori di prompt.
3. Pilih folder untuk mengkloning repositori Anda. Perhatikan bahwa folder Editor Kode default adalah`/home/sagemaker-user/`. Kloning repositori Anda mungkin membutuhkan waktu.
4. Untuk membuka repositori kloning, pilih Open in New Window atau Open.
5. Untuk kembali ke beranda UI aplikasi Editor Kode, pilih Batal.
6. Dalam repositori, prompt menanyakan apakah Anda mempercayai penulis file di repositori baru Anda. Anda memiliki dua pilihan:

a. Untuk mempercayai folder dan mengaktifkan semua fitur, pilih Ya, saya percaya penulis.

b. Untuk menelusuri konten repositori dalam mode terbatas, pilih Tidak, saya tidak mempercayai penulisnya.

   Dalam mode terbatas, tugas tidak diizinkan untuk dijalankan, debugging dinonaktifkan, pengaturan ruang kerja tidak diterapkan, dan ekstensi memiliki fungsionalitas terbatas.

   Untuk keluar dari mode terbatas, percayai penulis semua file di folder Anda saat ini atau folder induknya, dan aktifkan semua fitur, pilih Kelola di spanduk Mode Terbatas.

Koneksi dan Ekstensi Editor Kode

Code Editor mendukung koneksi IDE Layanan AWS serta ekstensi yang tersedia di Open VSX Registry.

Koneksi ke AWS

Lingkungan Code Editor terintegrasi dengan AWSToolkit for VS Code untuk menambahkan koneksi Layanan AWS ke. Untuk memulai dengan koneksi keLayanan AWS, Anda harus memiliki kredenal AWS Identity and Access Management (IAM) yang valid. Untuk informasi selengkapnya, lihat Otentikasi dan akses untuk Kode AWS Toolkit for Visual Studio.

Dalam lingkungan Editor Kode Anda, Anda dapat menambahkan koneksi ke:

- AWSExplorer — Melihat, memodifikasi, dan menerapkan AWS sumber daya di Amazon S3 CloudWatch, dan banyak lagi.

  Mengakses fitur tertentu dalam AWS Explorer memerlukan AWS izin tertentu. Untuk informasi selengkapnya, lihat Otentikasi dan akses untuk Kode AWS Toolkit for Visual Studio.
- Amazon CodeWhisperer— Bangun aplikasi lebih cepat dengan saran kode yang didukung AI.

  Untuk menggunakan Amazon CodeWhisperer Editor Kode, Anda harus menambahkan izin berikut ke peran SageMaker eksekusi Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeWhispererPermissions",
```

```
      "Effect": "Allow",
      "Action": ["codewhisperer:GenerateRecommendations"],
      "Resource": "*"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat Membuat kebijakan IAM dan Menambahkan serta menghapus izin identitas IAM di Panduan Pengguna IAM.

Ekstensi

Editor Kode mendukung ekstensi IDE yang tersedia di Open VSX Registry.

Untuk memulai dengan ekstensi di lingkungan Editor Kode Anda, pilih ikon Ekstensi () di panel navigasi kiri. Di sini, Anda dapat mengonfigurasi koneksi AWS dengan menginstal fileAWS Toolkit. Untuk informasi selengkapnya, silakan lihat Menginstal AWS Toolkit for Visual Studio Code.

Di bilah pencarian, Anda dapat mencari langsung ekstensi tambahan melalui Open VSX Registry, seperti, JupyterAWS Toolkit, Python dan banyak lagi.

Keluar dan matikan sumber daya

Di sudut kiri atas lingkungan Editor Kode, pilih ikon menu ().

Kemudian, pilih SageMaker: Keluar.

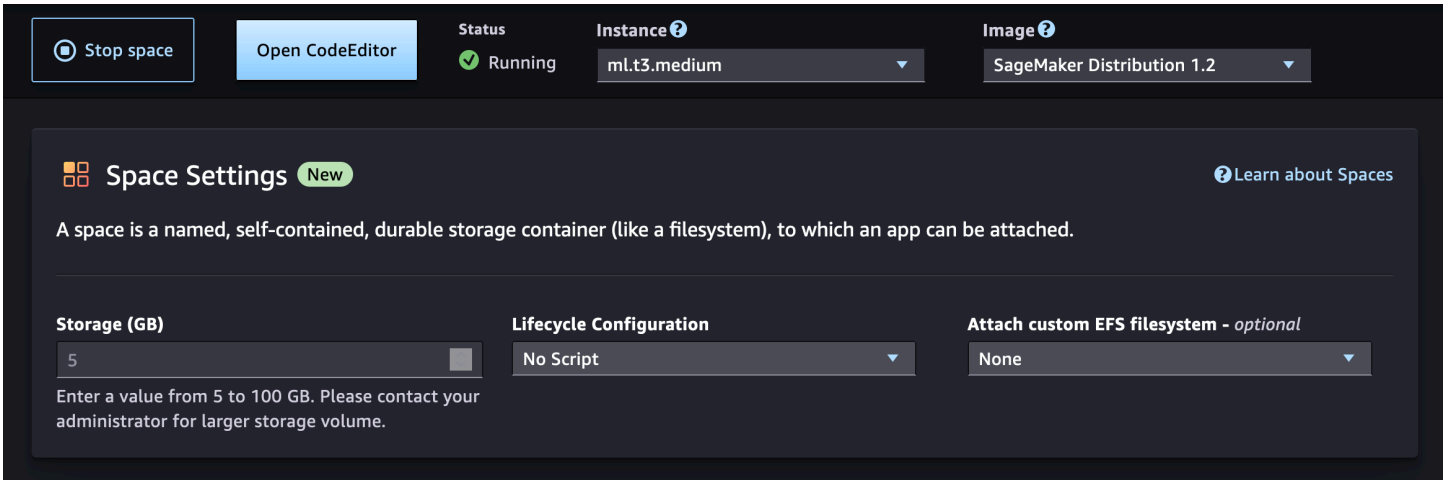
Hentikan ruang Anda melalui Studio

Untuk menghentikan ruang Editor Kode di Studio, gunakan langkah-langkah berikut:

Untuk menghentikan ruang Editor Kode Anda di Studio

1. Kembali ke halaman landing Code Editor dengan melakukan salah satu hal berikut:
   a. Di bilah navigasi di sudut kiri atas, pilih Editor Kode.
   b. Atau, di panel navigasi kiri, pilih Editor Kode di menu Aplikasi.

2. Temukan nama ruang Editor Kode yang Anda buat. Jika status ruang Anda Berjalan, pilih Berhenti di kolom Tindakan. Anda juga dapat menghentikan ruang Anda secara langsung di halaman detail spasi dengan memilih Stop space. Ruang mungkin membutuhkan waktu untuk berhenti.



Sumber daya tambahan seperti SageMaker endpoint, cluster Amazon EMR (Amazon EMR) dan Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) bucket yang dibuat dari Studio tidak otomatis dihapus saat instance spasi dimatikan. Untuk berhenti mengumpulkan biaya dari sumber daya, hapus sumber daya tambahan apa pun. Untuk informasi selengkapnya, lihat Menghapus sumber daya yang tidak digunakan.

Matikan sumber daya menggunakan AWS CLI

Anda dapat menghapus aplikasi dan spasi Editor Kode Anda menggunakan AWS Command Line Interface (AWS CLI).

- DeleteApp
- DeleteSpace

Panduan adminstrator Editor Kode

Anda dapat menggunakan Editor Kode dengan Instans Sesuai Permintaan untuk waktu startup yang lebih cepat, dan penyimpanan yang dapat dikonfigurasi. Anda dapat meluncurkan aplikasi Editor Kode melalui Amazon SageMaker Studio atau melalui fileAWS CLI. Anda juga dapat mengedit pengaturan default Editor Kode dalam konsol Domain. Untuk informasi selengkapnya, lihat Lihat dan Edit Domain.

Topik

Prasyarat

Untuk menggunakan Editor Kode, berdasarkan Code-OSS, Visual Studio Code - Open Source, pertama onboard ke SageMaker Amazon Domain dan buat profil pengguna. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.

Jika Anda berinteraksi dengan aplikasi Editor Kode Anda menggunakanAWS CLI, Anda juga harus menyelesaikan prasyarat berikut.

- Perbarui AWS CLI dengan mengikuti langkah-langkah dalam Menginstal AWS CLI Versi saat ini.
- Dari mesin lokal Anda, jalankan `aws configure` dan berikan AWS kredensyal Anda. Untuk informasi tentang AWS kredensyal, lihat Memahami dan mendapatkan kredensyal Anda AWS.

Untuk mendapatkan lebih banyak penyimpanan dan komputasi untuk aplikasi Anda, Anda dapat meminta peningkatan AWS kuota Anda. Untuk informasi selengkapnya tentang meminta peningkatan kuota, lihat SageMaker titik akhir dan kuota Amazon.

Berikan pengguna Anda akses ke ruang pribadi

Bagian ini menyediakan kebijakan yang memberikan akses pengguna ke ruang pribadi. Anda juga dapat menggunakan kebijakan untuk membatasi ruang pribadi dan aplikasi yang terkait dengannya kepada pemilik yang terkait dengan profil pengguna.

Anda harus memberikan izin kepada pengguna Anda untuk hal-hal berikut:

- Ruang pribadi
- Profil pengguna yang diperlukan untuk akses ke ruang pribadi

Untuk memberikan izin, lampirkan kebijakan berikut ke peran IAM pengguna Anda.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {

      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateApp",
        "sagemaker:DeleteApp"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:app/*",
      "Condition": {
        "Null": {
          "sagemaker:OwnerUserProfileArn": "true"
        }
      }
    },
    {
      "Sid": "SMStudioCreatePresignedDomainUrlForUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreatePresignedDomainUrl"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:user-profile/
${sagemaker:DomainId}/${sagemaker:UserProfileName}"
    },
    {
      "Sid": "SMStudioAppPermissionsListAndDescribe",
      "Effect": "Allow",
      "Action": [
        "sagemaker:ListApps",
        "sagemaker:ListDomains",
        "sagemaker:ListUserProfiles",
        "sagemaker:ListSpaces",
        "sagemaker:DescribeApp",
        "sagemaker:DescribeDomain",
        "sagemaker:DescribeUserProfile",
        "sagemaker:DescribeSpace"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SMStudioAppPermissionsTagOnCreate",
      "Effect": "Allow",
      "Action": [
```

```
        "sagemaker:AddTags"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:*/*",
      "Condition": {
        "Null": {
          "sagemaker:TaggingAction": "false"
        }
      }
    },
    {
      "Sid": "SMStudioRestrictSharedSpacesWithoutOwners",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateSpace",
        "sagemaker:UpdateSpace",
        "sagemaker:DeleteSpace"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:space/
${sagemaker:DomainId}/*",
      "Condition": {
        "Null": {
          "sagemaker:OwnerUserProfileArn": "true"
        }
      }
    },
    {
      "Sid": "SMStudioRestrictSpacesToOwnerUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateSpace",
        "sagemaker:UpdateSpace",
        "sagemaker:DeleteSpace"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:space/
${sagemaker:DomainId}/*",
      "Condition": {
        "ArnLike": {
          "sagemaker:OwnerUserProfileArn": "arn:aws:sagemaker:$Wilayah AWS:
$111122223333:user-profile/${sagemaker:DomainId}/${sagemaker:UserProfileName}"
        },
        "StringEquals": {
          "sagemaker:SpaceSharingType": [
            "Private",
            "Shared"
```

```
          ]
        }
      }
    },
    {
      "Sid": "SMStudioRestrictCreatePrivateSpaceAppsToOwnerUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateApp",
        "sagemaker:DeleteApp"
      ],
      "Resource": "arn:aws:sagemaker:{{Region}}:{{AccountId}}:app/
${sagemaker:DomainId}/*",
      "Condition": {
        "ArnLike": {
          "sagemaker:OwnerUserProfileArn": "arn:aws:sagemaker:
${aws:Region}:${aws:PrincipalAccount}:user-profile/${sagemaker:DomainId}/
${sagemaker:UserProfileName}"
        },
        "StringEquals": {
          "sagemaker:SpaceSharingType": [
            "Private"
          ]
        }
      }
    },
  ]
}
```

Ubah ukuran penyimpanan default

Anda dapat mengubah pengaturan penyimpanan default pengguna Anda. Anda juga dapat mengubah pengaturan penyimpanan default berdasarkan kebutuhan organisasi Anda dan kebutuhan pengguna Anda.

Untuk mengubah ukuran penyimpanan pengguna Anda, lakukan hal berikut:

1. Perbarui pengaturan penyimpanan Amazon EBS di Domain.
2. Buat profil pengguna dan tentukan pengaturan penyimpanan di dalamnya.

Gunakan perintah berikut AWS Command Line Interface (AWS CLI) untuk memperbarui domain.

```
aws --region $REGION sagemaker update-domain \
--domain-id $DOMAIN_ID \
--default-user-settings '{
    "SpaceStorageSettings": {
        "DefaultEbsStorageSettings":{
            "DefaultEbsVolumeSizeInGb":5,
            "MaximumEbsVolumeSizeInGb":100
        }
    }
}'
```

Gunakan AWS CLI perintah berikut untuk membuat profil pengguna dan menentukan pengaturan penyimpanan default.

```
aws --region $REGION sagemaker create-user-profile \
--domain-id $DOMAIN_ID \
--user-profile-name $USER_PROFILE_NAME \
--user-settings '{
    "SpaceStorageSettings": {
        "DefaultEbsStorageSettings":{
            "DefaultEbsVolumeSizeInGb":5,
            "MaximumEbsVolumeSizeInGb":100
        }
    }
}'
```

Gunakan AWS CLI perintah berikut untuk memperbarui pengaturan penyimpanan default di profil pengguna.

```
aws --region $REGION sagemaker update-user-profile \
--domain-id $DOMAIN_ID \
--user-profile-name $USER_PROFILE_NAME \
--user-settings '{
    "SpaceStorageSettings": {
        "DefaultEbsStorageSettings":{
            "DefaultEbsVolumeSizeInGb":25,
            "MaximumEbsVolumeSizeInGb":200
        }
    }
}'
```

Konfigurasi siklus hidup Editor Kode

Anda dapat menggunakan konfigurasi siklus hidup Editor Kode untuk mengotomatiskan penyesuaian lingkungan Studio Anda. Kustomisasi ini termasuk menginstal paket kustom, mengkonfigurasi ekstensi, preloading dataset, dan menyiapkan repositori kode sumber.

Instruksi berikut menggunakan AWS Command Line Interface (AWS CLI) untuk membuat, melampirkan, men-debug, dan melepaskan konfigurasi siklus hidup untuk jenis aplikasi: `CodeEditor`

- Membuat dan melampirkan konfigurasi siklus hidup di Studio
- Debug konfigurasi siklus hidup di Studio
- Lepaskan konfigurasi siklus hidup di Studio

Membuat dan melampirkan konfigurasi siklus hidup di Studio

Bagian berikut menyediakan AWS CLI perintah untuk membuat konfigurasi siklus hidup, melampirkan konfigurasi siklus hidup saat membuat profil pengguna baru, dan melampirkan konfigurasi siklus hidup saat memperbarui profil pengguna. Untuk prasyarat dan langkah-langkah umum dalam membuat dan melampirkan konfigurasi siklus hidup di Studio, lihat. Membuat dan mengaitkan konfigurasi siklus hidup

Saat membuat konfigurasi siklus hidup Studio Anda dengan `create-studio-lifecycle-config` perintah, pastikan untuk menentukan apakah ada. `studio-lifecycle-config-app-type` *`CodeEditor`* Contoh berikut menunjukkan cara membuat konfigurasi siklus hidup Studio baru untuk aplikasi Editor Kode Anda.

```
aws sagemaker create-studio-lifecycle-config \
--studio-lifecycle-config-name my-code-editor-lcc \
--studio-lifecycle-config-content $LCC_CONTENT \
--studio-lifecycle-config-app-type CodeEditor
```

Perhatikan ARN dari konfigurasi siklus hidup yang baru dibuat yang dikembalikan. Saat melampirkan konfigurasi siklus hidup, berikan ARN ini dalam daftar. `LifecycleConfigArns` `CodeEditorAppSettings`

Anda dapat melampirkan konfigurasi siklus hidup saat membuat profil pengguna atau domain. Contoh berikut menunjukkan cara membuat profil pengguna baru dengan konfigurasi siklus hidup

terlampir. Anda juga dapat membuat domain baru dengan konfigurasi siklus hidup yang dilampirkan dengan menggunakan perintah create-domain.

```
# Create a new UserProfile
aws sagemaker create-user-profile \
--domain-id domain-id \
--user-profile-name user-profile-name \
--user-settings '{
"CodeEditorAppSettings": {
  "LifecycleConfigArns":
    [lifecycle-configuration-arn-list]
  }
}'
```

Anda juga dapat melampirkan konfigurasi siklus hidup saat memperbarui profil pengguna atau domain. Contoh berikut menunjukkan cara memperbarui profil pengguna dengan konfigurasi siklus hidup terlampir. Anda juga dapat memperbarui domain baru dengan konfigurasi siklus hidup yang dilampirkan menggunakan perintah update-domain.

```
# Update a UserProfile
aws sagemaker update-user-profile \
--domain-id domain-id \
--user-profile-name user-profile-name \
--user-settings '{
"CodeEditorAppSettings": {
  "LifecycleConfigArns":
    [lifecycle-configuration-arn-list]
  }
}'
```

Debug konfigurasi siklus hidup di Studio

Untuk petunjuk tentang men-debug konfigurasi siklus hidup di Studio, lihat. Debug konfigurasi siklus hidup

Untuk menemukan log untuk aplikasi tertentu, cari aliran log menggunakan format berikut:

```
domain-id/space-name/CodeEditor/default/LifecycleConfigOnStart
```

Lepaskan konfigurasi siklus hidup di Studio

Untuk langkah-langkah melepaskan konfigurasi siklus hidup di Studio, lihat. Lepaskan konfigurasi siklus hidup

Untuk melepaskan konfigurasi siklus hidup menggunakanAWS CLI, hapus konfigurasi siklus hidup yang diinginkan dari daftar konfigurasi siklus hidup yang dilampirkan ke sumber daya. Kemudian lewati daftar sebagai bagian dari perintah masing-masing:

- update-user-profile
- pembaruan-domain

Misalnya, perintah berikut menghapus semua konfigurasi siklus hidup untuk aplikasi Editor Kode yang dilampirkan ke domain.

```
aws sagemaker update-domain --domain-id domain-id \
--default-user-settings '{
"CodeEditorAppSettings": {
  "LifecycleConfigArns":
    []
  }
}'
```

Buat konfigurasi siklus hidup untuk mengkloning repositori ke dalam aplikasi Editor Kode

Bagian ini menunjukkan cara mengkloning repositori dan membuat aplikasi Editor Kode dengan konfigurasi siklus hidup terlampir.

1. Dari mesin lokal Anda, buat file bernama `my-script.sh` dengan konten berikut:

   ```
   #!/bin/bash
   set -eux
   ```

2. Kloning repositori pilihan Anda dalam skrip konfigurasi siklus hidup Anda.

   ```
   export REPOSITORY_URL="https://github.com/aws-samples/sagemaker-studio-lifecycle-config-examples.git"
   git -C /home/sagemaker-user clone $REPOSITORY_URL
   ```

3. Setelah menyelesaikan skrip Anda, buat dan lampirkan konfigurasi siklus hidup Anda. Untuk informasi selengkapnya, lihat Membuat dan melampirkan konfigurasi siklus hidup di Studio.

4. Buat aplikasi Editor Kode Anda dengan konfigurasi siklus hidup terlampir.

```
aws sagemaker create-app \
--domain-id domain-id \
--space-name space-name \
--app-type CodeEditor \
--app-name default \
--resource-spec "SageMakerImageArn=arn:aws:sagemaker:region:image-account-
id:image/sagemaker-distribution-
cpu,LifecycleConfigArn=arn:aws:sagemaker:region:user-account-id:studio-lifecycle-
config/my-code-editor-lcc,InstanceType=ml.t3.large"
```

   Untuk informasi selengkapnya tentang ARN gambar Editor Kode yang tersedia, lihatContoh dan gambar aplikasi Editor Kode.

Buat konfigurasi siklus hidup untuk menginstal ekstensi Editor Kode

Bagian ini menunjukkan cara membuat konfigurasi lifecyctle untuk menginstal ekstensi dari Open VSX Registry di lingkungan Editor Kode Anda.

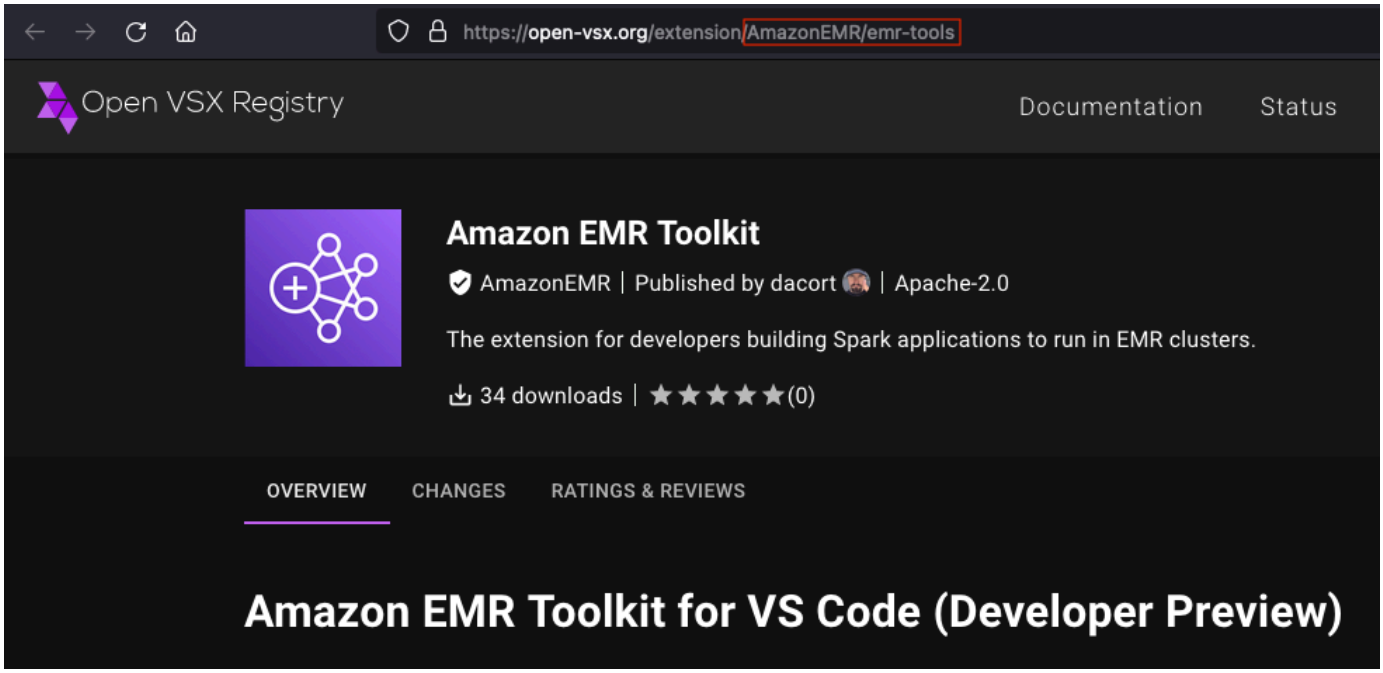
1. Dari mesin lokal Anda, buat file bernama `my-script.sh` dengan konten berikut:

```
#!/bin/bash
set -eux
```

2. Dalam skrip, instal ekstensi Open VSX Registry pilihan Anda:

```
sagemaker-code-editor --install-extension AmazonEMR.emr-tools --extensions-dir /
opt/amazon/sagemaker/sagemaker-code-editor-server-data/extensions
```

   Anda dapat mengambil nama ekstensi dari URL ekstensi di Open VSX Registry. Nama ekstensi yang digunakan dalam `sagemaker-code-editor` perintah harus berisi semua teks `https://open-vsx.org/extension/` yang mengikuti URL. Ganti semua contoh garis miring (`/`) dengan periode (`.`). Misalnya, `AmazonEMR/emr-tools` seharusnya`AmazonEMR.emr-tools`.



3. Setelah menyelesaikan skrip Anda, buat dan lampirkan konfigurasi siklus hidup Anda. Untuk informasi selengkapnya, lihat Membuat dan melampirkan konfigurasi siklus hidup di Studio.
4. Buat aplikasi Editor Kode Anda dengan konfigurasi siklus hidup terlampir:

```
aws sagemaker create-app \
--domain-id domain-id \
--space-name space-name \
--app-type CodeEditor \
--app-name default \
--resource-spec "SageMakerImageArn=arn:aws:sagemaker:region:image-account-
id:image/sagemaker-distribution-
cpu,LifecycleConfigArn=arn:aws:sagemaker:region:user-account-id:studio-lifecycle-
config/my-code-editor-lcc,InstanceType=ml.t3.large"
```

Untuk informasi selengkapnya tentang ARN gambar Editor Kode yang tersedia, lihatContoh dan gambar aplikasi Editor Kode. Untuk informasi selengkapnya tentang koneksi dan ekstensi, lihatKoneksi dan Ekstensi Editor Kode.

SageMaker HyperPod

SageMaker HyperPod membantu Anda menyediakan cluster tangguh untuk menjalankan beban kerja machine learning (ML) dan mengembangkan state-of-the-art model seperti model bahasa besar (LLM), model difusi, dan model dasar (FM). Ini mempercepat pengembangan FM dengan

menghilangkan angkat berat yang tidak berdiferensiasi yang terlibat dalam membangun dan memelihara cluster komputasi skala besar yang ditenagai oleh ribuan akselerator seperti AWS Trainium dan NVIDIA A100 dan H100 Graphical Processing Unit (GPU). Ketika akselerator gagal, cluster penyembuhan diri secara otomatis mendeteksi dan mengganti perangkat keras yang rusak dengan cepat sehingga Anda dapat fokus menjalankan beban kerja ML selama berminggu-minggu dan berbulan-bulan tanpa gangguan. Selain itu, dengan SageMaker HyperPod, Anda dapat menyesuaikan lingkungan komputasi agar sesuai dengan kebutuhan Anda dan mengonfigurasinya dengan perpustakaan pelatihan SageMaker terdistribusi Amazon untuk mencapai kinerja optimal. AWS

Cluster operasi

Anda dapat membuat, mengonfigurasi, dan memelihara SageMaker HyperPod cluster secara grafis melalui antarmuka pengguna konsol (UI) dan secara terprogram melalui antarmuka AWS baris perintah (CLI) atau. AWS SDK for Python (Boto3) Dengan Amazon VPC, Anda dapat mengamankan jaringan cluster dan juga memanfaatkan konfigurasi cluster Anda dengan sumber daya di VPC Anda, seperti Amazon FSx for Lustre, yang menawarkan throughput tercepat. Anda juga dapat memberikan peran IAM yang berbeda ke grup instans klaster, dan membatasi tindakan yang dapat dioperasikan oleh sumber daya klaster dan pengguna Anda. Untuk mempelajari selengkapnya, lihat the section called “Beroperasi SageMaker HyperPod”.

Mengonfigurasi lingkungan ML Anda

SageMaker HyperPod berjalanthe section called “SageMaker HyperPod DLAMI”, yang mengatur lingkungan ML pada HyperPod cluster. Anda dapat mengonfigurasi penyesuaian tambahan ke DLAMI dengan menyediakan skrip siklus hidup untuk mendukung kasus penggunaan Anda. Untuk mempelajari lebih lanjut tentang cara mengatur skrip siklus hidup, lihat dan. the section called “Mulai menggunakan SageMaker HyperPod” the section called “Templat skrip konfigurasi siklus hidup”

Penjadwalan pekerjaan

Setelah Anda berhasil membuat HyperPod cluster, pengguna cluster dapat masuk ke node cluster (seperti node head atau controller, log-in node, dan worker node) dan menjadwalkan pekerjaan untuk menjalankan beban kerja machine learning. Untuk mempelajari selengkapnya, lihat the section called “Jalankan pekerjaan di HyperPod cluster”.

Ketahanan terhadap kegagalan perangkat keras

SageMaker HyperPod menjalankan pemeriksaan kesehatan pada node cluster dan menyediakan fungsionalitas auto-resume beban kerja. Dengan fitur ketahanan klaster HyperPod, Anda dapat

melanjutkan beban kerja dari pos pemeriksaan terakhir yang Anda simpan, setelah node yang salah diganti dengan node yang sehat di cluster dengan lebih dari 16 node. Untuk mempelajari selengkapnya, lihat the section called “Ketahanan klaster”.

Pencatatan dan pengelolaan klaster

Anda dapat menemukan metrik pemanfaatan SageMaker HyperPod sumber daya dan log siklus hidup di Amazon CloudWatch, dan mengelola SageMaker HyperPod sumber daya dengan menandainya. Setiap `CreateCluster` API yang dijalankan membuat aliran log yang berbeda, dinamai dalam `<cluster-name>-<timestamp>` format. Di aliran log, Anda dapat memeriksa nama host, nama skrip siklus hidup yang gagal, dan output dari skrip yang gagal seperti dan. `stdout` `stderr` Untuk informasi selengkapnya, lihat the section called “Manajemen klaster”.

Kompatibel dengan SageMaker alat

Dengan menggunakan SageMaker HyperPod, Anda dapat mengonfigurasi cluster dengan pustaka komunikasi kolektif yang AWS dioptimalkan yang ditawarkan oleh SageMaker, seperti perpustakaan paralelisme data SageMaker terdistribusi (SMDDP). Pustaka SMDDP mengimplementasikan `AllGather` operasi yang dioptimalkan ke AWS komputasi dan infrastruktur jaringan untuk instans pembelajaran SageMaker mesin berkinerja terbaik yang didukung oleh GPU NVIDIA A100. Untuk mempelajari informasi lebih lanjut, lihat the section called “Jadwalkan pekerjaan untuk beban kerja pelatihan terdistribusi pada SageMaker HyperPod”.

Topik

- SageMaker HyperPod prasyarat
- Memulai dengan SageMaker HyperPod
- Beroperasi SageMaker HyperPod
- Jalankan pekerjaan di SageMaker HyperPod cluster
- SageMaker HyperPod ketahanan klaster
- SageMaker HyperPod manajemen klaster
- SageMaker HyperPod referensi
- Catatan SageMaker HyperPod rilis Amazon

SageMaker HyperPod prasyarat

Bagian berikut memandu Anda melalui prasyarat yang perlu Anda persiapkan sebelum memulai. SageMaker HyperPod

Topik

SageMaker HyperPod kuota

Anda dapat membuat SageMaker HyperPod cluster yang diberikan kuota untuk penggunaan cluster di akun AndaAWS.

> **Important**
>
> Untuk mempelajari lebih lanjut tentang SageMaker HyperPod harga, lihat the section called "SageMaker HyperPod harga" dan SageMaker Harga Amazon.

Melihat SageMaker HyperPod kuota Amazon menggunakan AWS Management Console

Cari nilai default dan terapan kuota, juga disebut sebagai batas, untuk penggunaan klaster, yang digunakan untuk SageMaker HyperPod.

1. Buka konsol Service Quotas.
2. Di panel navigasi kiri, pilih AWSlayanan.
3. Dari daftar AWSlayanan, cari dan pilih Amazon SageMaker.
4. Dalam daftar Kuota layanan, Anda dapat melihat nama kuota layanan, nilai yang diterapkan (jika tersedia), kuota AWS default, dan apakah nilai kuota dapat disesuaikan.
5. Di bilah pencarian, ketik penggunaan cluster. Ini menunjukkan kuota untuk penggunaan cluster, kuota yang diterapkan, dan kuota default.

Untuk meningkatkan SageMaker HyperPod kuota Amazon menggunakan AWS Management Console

Tingkatkan kuota Anda di tingkat akun atau sumber daya.

1. Untuk menambah kuota instans untuk penggunaan klaster, pilih kuota yang ingin Anda tingkatkan.
2. Jika kuota dapat disesuaikan, Anda dapat meminta peningkatan kuota di tingkat akun atau tingkat sumber daya berdasarkan nilai yang tercantum di kolom Adjustability.
3. Untuk Meningkatkan nilai kuota, masukkan nilai baru. Nilai baru lebih besar dari nilai saat ini.
4. Pilih Permintaan.
5. Untuk melihat permintaan yang tertunda atau yang baru saja diselesaikan di konsol, navigasikan ke tab Riwayat permintaan dari halaman detail layanan, atau pilih Dasbor dari panel navigasi. Untuk permintaan yang tertunda, pilih status permintaan untuk membuka penerimaan permintaan. Status awal dari permintaan adalah Tertunda. Setelah status berubah menjadi Kuota yang diminta, Anda melihat nomor kasus denganAWS Support. Pilih nomor kasus untuk membuka tiket untuk permintaan Anda.

Untuk mempelajari lebih lanjut tentang meminta peningkatan kuota secara umum, lihat Meminta Peningkatan Kuota dalam Panduan Pengguna Service QuotasAWS.

Menyiapkan pengguna dan peran IAM untuk SageMaker HyperPod pengguna dan sumber daya

Ada tiga lapisan utama SageMaker HyperPod pengguna: admin AWS akun, administrator cluster (seperti arsitek cloud), dan pengguna cluster (seperti ilmuwan pembelajaran mesin). Admin AWS akun harus menyiapkan pengguna IAM dengan melampirkan izin atau kebijakan yang tepat untuk administrator klaster. Untuk administrator klaster, admin AWS akun juga harus membuat peran IAM yang dapat digunakan oleh administrator SageMaker HyperPod klaster untuk menjalankan dan berkomunikasi dengan AWS sumber daya yang diperlukan, seperti Amazon S3, Amazon, dan (SSM). CloudWatch AWS Systems Manager Terakhir, administrator klaster dapat memberikan izin kepada pengguna klaster untuk masuk ke SageMaker HyperPod cluster melalui Agen SSM.

Topik

- Menyiapkan pengguna IAM untuk administrator klaster
- Mengatur pengguna IAM untuk pengguna cluster
- Peran IAM untuk SageMaker HyperPod

Menyiapkan pengguna IAM untuk administrator klaster

Administrator cluster adalah arsitek cloud yang mengoperasikan dan mengkonfigurasi SageMaker HyperPod cluster, melakukan tugas di. the section called “Beroperasi SageMaker HyperPod” Contoh

kebijakan berikut mencakup kumpulan izin minimum bagi administrator klaster untuk menjalankan API SageMaker HyperPod inti dan mengelola klaster apa pun di dalam akun AndaAWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateCluster",
                "sagemaker:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:DeleteCluster",
                "sagemaker:DescribeCluster",
                "sagemaker:DescribeClusterNode",
                "sagemaker:ListClusterNodes",
                "sagemaker:UpdateCluster"
            ],
            "Resource": "arn:aws:sagemaker:region:account-id:cluster/*"
        }
    ]
}
```

Untuk memberikan izin untuk mengakses SageMaker konsol, gunakan kebijakan sampel yang disediakan di Izin yang Diperlukan untuk Menggunakan Konsol Amazon SageMaker .

Untuk memberikan izin untuk mengakses konsol SSM, gunakan kebijakan sampel yang disediakan di Menggunakan AWS Systems Manager konsol di AWS Systems Manager Panduan Pengguna.

Anda juga dapat mempertimbangkan untuk melampirkan `AmazonSageMakerFullAccess`kebijakan ke pengguna IAM; namun, perhatikan bahwa `AmazonSageMakerFullAccess` kebijakan tersebut memberikan izin ke seluruh panggilan, fitur, dan SageMaker sumber daya API.

Untuk panduan tentang pengguna IAM secara umum, lihat pengguna IAM di AWS Identity and Access ManagementPanduan Pengguna.

Mengatur pengguna IAM untuk pengguna cluster

Pengguna cluster adalah insinyur pembelajaran mesin yang masuk dan menjalankan beban kerja ML pada node SageMaker HyperPod cluster yang disediakan oleh administrator klaster. Untuk pengguna cluster di AWS akun Anda, Anda harus memberikan izin `"ssm:StartSession"` untuk menjalankan `start-session` perintah SSM. Berikut ini adalah contoh kebijakan untuk pengguna IAM.

Izin IAM untuk semua sumber daya

Tambahkan kebijakan berikut untuk memberikan izin sesi SSM pengguna IAM untuk terhubung ke target SSM untuk semua sumber daya.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:TerminateSession"
            ],
            "Resource": "*"
        }
    ]
}
```

Peran IAM untuk SageMaker HyperPod

Agar SageMaker HyperPod cluster dapat berjalan dan berkomunikasi dengan AWS sumber daya yang diperlukan, Anda perlu melampirkan yang dikelola `AmazonSageMakerClusterInstanceRolePolicy` ke grup instance cluster. Dengan kebijakan AWS terkelola ini, grup instans SageMaker HyperPod klaster berperan untuk berkomunikasi dengan Amazon CloudWatch, Amazon S3, dan AWS Systems Manager Agen (Agen SSM). Kebijakan terkelola ini adalah persyaratan minimum agar SageMaker HyperPod sumber daya dapat berjalan dengan benar, jadi Anda harus memberikan peran IAM dengan kebijakan ini ke semua grup instans. Ini `AmazonSageMakerClusterInstanceRolePolicy` memiliki izin berikut:

- log - Diperlukan untuk memungkinkan SageMaker HyperPod untuk mempublikasikan aliran log.
- cloudwatch — Diperlukan untuk memungkinkan SageMaker HyperPod untuk memposting CloudWatch metrik.

- s3 - Diperlukan SageMaker HyperPod untuk memungkinkan daftar dan mengambil file dari bucket Amazon S3 di akun Anda dengan awalan. `sagemaker-`
- ssmmessages - Diperlukan untuk memungkinkan Agen SSM berkomunikasi dengan layanan backend SSM. Prinsipal dapat menggunakan Agen SSM untuk membuat dan membuka saluran kontrol dan data. SageMaker memulai dan mengelola Agen SSM saat memulai instance cluster.

> **Tip**
>
> Bergantung pada preferensi Anda dalam mendesain tingkat izin untuk beberapa grup instans, Anda juga dapat mengatur beberapa peran IAM dan melampirkannya ke grup instans yang berbeda. Saat Anda mengatur akses pengguna klaster ke node SageMaker HyperPod cluster tertentu, node mengambil peran dengan izin selektif yang Anda lampirkan secara manual. Ketika Anda, sebagai admin AWS akun atau administrator klaster, mengatur akses pengguna cluster ke node cluster tertentu melalui AWS Systems Manager(lihat jugathe section called “Siapkan AWS Systems Manager dan Jalankan Sebagai untuk kontrol akses pengguna cluster”), node cluster mengambil peran dengan izin selektif yang Anda lampirkan secara manual.

Setelah Anda selesai membuat peran IAM, buat catatan nama dan ARN mereka. Anda menggunakan peran saat membuat SageMaker HyperPod klaster, memberikan izin yang benar yang diperlukan untuk setiap grup instans untuk berkomunikasi dengan sumber daya yang diperlukanAWS.

(Opsional) Izin tambahan untuk digunakan SageMaker HyperPod dengan Amazon Virtual Private Cloud

Jika Anda ingin menggunakan Amazon Virtual Private Cloud (VPC) Anda sendiri alih-alih SageMaker VPC default, Anda harus menambahkan izin tambahan berikut ke peran IAM untuk. SageMaker HyperPod

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
```

```
            "ec2:DescribeVpcs",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups",
            "ec2:DetachNetworkInterface"
        ],
        "Resource": "*"
    }
    {
        "Effect": "Allow",
        "Action": "ec2:CreateTags",
        "Resource": [
            "arn:aws:ec2:*:*:network-interface/*"
        ]
    }
```

Daftar berikut menguraikan izin mana yang diperlukan untuk mengaktifkan fungsionalitas SageMaker HyperPod klaster saat Anda mengonfigurasi klaster dengan VPC Amazon Anda sendiri.

- `ec2`Izin berikut diperlukan untuk mengaktifkan konfigurasi SageMaker HyperPod cluster dengan VPC Anda.

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
}
```

- `ec2`Izin berikut diperlukan untuk mengaktifkan fungsi SageMaker HyperPod auto-resume.

```
{
    "Effect": "Allow",
    "Action": [
```

```
            "ec2:DetachNetworkInterface"
        ],
        "Resource": "*"
}
```

- ec2Izin berikut memungkinkan SageMaker HyperPod untuk membuat tag pada antarmuka jaringan dalam akun Anda.

```
{
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
    ]
}
```

Siapkan AWS Systems Manager dan Jalankan Sebagai untuk kontrol akses pengguna cluster

the section called “SageMaker HyperPod DLAMI”dilengkapi dengan AWS Systems Manager(SSM) di luar kotak untuk membantu Anda mengelola akses ke grup instans SageMaker HyperPod cluster Anda. Bagian ini menjelaskan cara membuat pengguna sistem operasi (OS) di SageMaker HyperPod cluster Anda dan mengaitkannya dengan pengguna dan peran IAM. Ini berguna untuk mengautentikasi sesi SSM menggunakan kredensil akun pengguna OS.

Aktifkan Run As di AWS akun Anda

Sebagai admin AWS akun atau administrator cloud, Anda dapat mengelola akses ke SageMaker HyperPod cluster pada peran IAM atau tingkat pengguna dengan menggunakan fitur Run As di SSM. Dengan fitur ini, Anda dapat memulai setiap sesi SSM menggunakan pengguna OS yang terkait dengan peran IAM atau pengguna.

Untuk mengaktifkan Run As di AWS akun Anda, ikuti langkah-langkah di Turn on Run As support for Linux dan macOS managed node. Jika Anda sudah membuat pengguna OS di klaster Anda, pastikan Anda mengaitkannya dengan peran IAM atau pengguna dengan menandai mereka sebagai dipandu di Opsi 2 dari langkah 5 di bawah Untuk mengaktifkan dukungan Run As untuk Linux dan node yang dikelola macOS.

Siapkan skrip untuk menyiapkan pengguna Linux

Untuk menyelesaikan pengaturan pengguna cluster untuk mengakses HyperPod kluster melalui SSM, Anda perlu mengonfigurasi skrip untuk menambahkan pengguna sambil menyiapkan skrip konfigurasi siklus hidup untuk membuat klaster. HyperPod Dalam repositori template yang disediakan dithe section called “Templat skrip konfigurasi siklus hidup”, ada skrip bernama `add_users.sh` yang mengkonsumsi data pengguna dari. `shared_users.txt` Ada juga template bernama`shared_users_sample.txt`, jadi lihat contoh file teks untuk mengisi dan membuat `shared_users.txt` file. Perhatikan bahwa Anda harus mengunggah dua file sebagai bagian dari persiapan dan pengunggahan skrip siklus hidup ke bucket S3, yang akan Anda pelajari. the section called “Mulai menggunakan SageMaker HyperPod”

(Opsional) Siapkan SageMaker HyperPod dengan VPC Amazon Anda

Jika Anda tidak menyediakan VPC, SageMaker HyperPod gunakan VPC default. SageMaker Jika Anda ingin menggunakan VPC Anda sendiri untuk terhubung SageMaker HyperPod dengan AWS sumber daya di VPC Anda, Anda harus memberikan nama VPC, ID, ID subnet, dan ID Wilayah AWS grup keamanan saat Anda membuat. SageMaker HyperPod Jika Anda ingin membuat VPC baru, lihat Membuat VPC default atau Membuat VPC di Panduan Pengguna Amazon Virtual Private Cloud.

Penting bagi Anda untuk membuat semua sumber daya di Zona Ketersediaan yang samaWilayah AWS, dan mengonfigurasi aturan grup keamanan untuk memungkinkan koneksi antar sumber daya.

Misalnya, asumsikan bahwa Anda membuat VPC di Wilayah. `us-west-2` Kemudian, Anda harus membuat subnet di VPC ini di Availability `us-west-2a` Zone, dan Anda harus membuat grup keamanan, yang memungkinkan semua lalu lintas masuk (masuk) dari dalam grup keamanan, dan memungkinkan semua lalu lintas keluar.

Anda juga perlu memastikan bahwa VPC memiliki koneksi ke. Amazon Simple Storage Service Jika Anda mengonfigurasi VPC Anda, maka grup instans tidak memiliki akses ke internet. Mereka tidak dapat terhubung ke bucket Amazon S3 untuk mengakses atau menyimpan file seperti skrip siklus hidup, data pelatihan, dan artefak model, kecuali Anda membuat titik akhir VPC yang memungkinkan akses. Dengan membuat titik akhir VPC, Anda mengizinkan grup instans mengakses bucket S3 dalam VPC yang sama. Kami menyarankan Anda juga membuat kebijakan khusus yang hanya mengizinkan permintaan dari VPC pribadi Anda untuk mengakses bucket S3 Anda. Untuk informasi selengkapnya, lihat Titik Akhir untuk Amazon S3.

(Opsional) Siapkan SageMaker HyperPod dengan Amazon FSx for Lustre

Untuk mulai menggunakan SageMaker HyperPod dan memetakan jalur data antara cluster dan FSx for Lustre fille system, pilih salah satu yang didukung oleh. Wilayah AWS SageMaker HyperPod Setelah memilih yang Wilayah AWS Anda inginkan, Anda juga harus menentukan Availability Zone (AZ) mana yang akan digunakan. Jika Anda menggunakan node SageMaker HyperPod komputasi di AZ yang berbeda dari AZ di mana sistem fsX for Lustre file Anda diatur dalam hal yang sama, mungkin ada komunikasi dan overhead jaringan. Wilayah AWS Kami menyarankan Anda untuk menggunakan AZ fisik yang sama dengan AZ untuk akun SageMaker HyperPod layanan untuk menghindari lalu lintas lintas lintas AZ antara SageMaker HyperPod cluster dan FSx for Lustre file system Anda. Juga, pastikan Anda telah mengonfigurasinya dengan VPC Anda. Jika Anda ingin menggunakan Amazon FSx sebagai sistem file utama untuk penyimpanan, Anda harus mengonfigurasi SageMaker HyperPod cluster dengan VPC.

Memulai dengan SageMaker HyperPod

Mulailah dengan membuat SageMaker HyperPod klaster pertama Anda dan pelajari fungsionalitas operasi klaster. SageMaker HyperPod

Anda dapat membuat SageMaker HyperPod cluster melalui UI SageMaker konsol atau AWS CLI perintah. Tutorial ini menunjukkan cara membuat SageMaker HyperPod cluster baru dengan Slurm, yang merupakan perangkat lunak penjadwal beban kerja yang populer. Setelah Anda melalui tutorial ini, Anda akan tahu cara masuk ke node cluster menggunakan AWS Systems Manager perintah (`aws ssm`). Setelah Anda menyelesaikan tutorial ini, lihat juga the section called “Beroperasi SageMaker HyperPod” untuk mempelajari lebih lanjut tentang oparasi SageMaker HyperPod dasar, dan the section called “Jalankan pekerjaan di HyperPod cluster” untuk mempelajari cara menjadwalkan pekerjaan di klaster yang disediakan.

Topik

- Menggunakan UI SageMaker HyperPod konsol
- Menggunakan AWS CLI perintah untuk SageMaker HyperPod API

Menggunakan UI SageMaker HyperPod konsol

Buat SageMaker HyperPod cluster pertama Anda menggunakan UI SageMaker HyperPod konsol.

Buat SageMaker HyperPod klaster pertama Anda

Petunjuk berikut menunjukkan cara membuat SageMaker HyperPod cluster baru menggunakan UI SageMaker konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih HyperPod Cluster di panel navigasi kiri.
3. Di halaman SageMaker HyperPod Clusters, pilih Create cluster.
4. Di Langkah 1: Pengaturan cluster, tentukan nama untuk cluster baru. Lewati bagian Tag.
5. Pada Langkah 2: Grup instance, tambahkan grup instance. Setiap grup instans dapat dikonfigurasi secara berbeda, dan Anda dapat membuat klaster heterogen yang terdiri dari beberapa grup instans dengan berbagai jenis instance. Agar skrip konfigurasi siklus hidup berjalan di grup instans selama pembuatan klaster, Anda dapat memulai dengan menggunakan contoh skrip siklus hidup yang disediakan di repositori Pelatihan Terdistribusi Awsome. GitHub

   a. Untuk nama grup Instance, tentukan nama untuk grup instance.
   b. Untuk jenis instance Select, pilih instance untuk grup instance.
   c. Untuk Kuantitas, tentukan bilangan bulat yang tidak melebihi kuota instance untuk penggunaan klaster.
   d. Untuk jalur S3 ke file skrip siklus hidup, masukkan jalur Amazon S3 tempat skrip siklus hidup Anda disimpan.

      i. Unduh salinan skrip siklus hidup ke direktori di komputer lokal Anda.

      ```
      git clone https://github.com/aws-samples/awsome-distributed-training/
      ```

      ii. Di bawah `1.architectures/5.sagemaker_hyperpods/LifecycleScripts/base-config`, Anda dapat menemukan satu set lengkap skrip siklus hidup. Untuk mempelajari lebih lanjut tentang skrip siklus hidup, lihat juga. the section called “Templat skrip konfigurasi siklus hidup”

      iii. Buka dan edit file bernama `provisioning_params.json` untuk menentukan parameter dasar untuk penamaan sumber daya cluster. Misalnya, jika Anda membuat cluster dengan konfigurasi Slurm, `provisioning_params.json` harus mirip dengan berikut ini.

      ```
      {
          "workload_manager": "slurm",
      ```

```
    "controller_group": "my-controller-group",
    "login_group": "my-login-group", // Optional
    "worker_groups": [
        {
            "instance_group_name": "worker-group-1",
            "partition_name": "partition-1"
        }
    ]
}
```

Anda dapat mengedit nilai kunci untuk`controller_group`,`login_group`, `instance_group_name` dan `partition_name` kunci ke nilai masing-masing.

iv. Unggah skrip ke bucket Amazon S3 Anda. Buat bucket S3 yang jalurnya dalam format berikut:`s3://sagemaker-<unique-s3-bucket-name>/<lifecycle-script-directory>/src`. Anda dapat membuat bucket ini menggunakan konsol Amazon S3.

> **Note**
>
> Anda harus `sagemaker-` mengawali jalur bucket S3, karena ??? with `AmazonSageMakerClusterInstanceRolePolicy` memungkinkan prinsipal untuk hanya mengakses bucket S3 dengan awalan khusus ini.

e. Untuk jalur Direktori ke skrip siklus hidup yang sedang dibuat, masukkan nama file skrip siklus hidup di bawah jalur S3 ke file skrip siklus hidup.

f. Untuk peran IAM, pilih peran IAM yang Anda buat menggunakan `AmazonSageMakerClusterInstanceRolePolicy` dari bagian. the section called “Peran IAM untuk SageMaker HyperPod”

g. Untuk Thread per inti di bawah Konfigurasi lanjutan, tentukan 1 untuk menonaktifkan multi-threading dan 2 untuk mengaktifkan multi-threading. Untuk mengetahui jenis instans mana yang mendukung multi-threading, lihat tabel referensi inti CPU dan utas per inti CPU per jenis instans di Panduan Pengguna Amazon Elastic Compute Cloud.

6. Pada Langkah 3: Konfigurasi lanjutan, atur pengaturan jaringan di dalam dan in-and-out dari cluster. Pilih VPC Anda sendiri jika Anda sudah memiliki VPC yang memberikan SageMaker akses ke VPC Anda. Jika Anda tidak memilikinya tetapi ingin membuat VPC baru, ikuti petunjuk di Buat VPC di Panduan Pengguna Amazon Virtual Private Cloud. Anda dapat membiarkannya sebagai tidak ada VPC untuk menggunakan SageMaker VPC default.

7. Pada Langkah 4: Tinjau dan buat, tinjau konfigurasi yang telah Anda tetapkan dari langkah 1 hingga 3 dan selesaikan pengiriman permintaan pembuatan cluster.
8. Cluster baru akan muncul di bawah Cluster di panel utama konsol. SageMaker HyperPod Anda dapat memeriksa statusnya yang ditampilkan di bawah kolom Status.
9. Setelah status cluster berubah`InService`, pengguna cluster dapat mulai masuk ke node cluster. Untuk informasi selengkapnya tentang mengakses node cluster dan menjalankan beban kerja ML, lihat. the section called “Jalankan pekerjaan di HyperPod cluster”

Hapus cluster dan sumber daya bersih

Setelah Anda berhasil menguji pembuatan SageMaker HyperPod klaster, klaster terus berjalan di **`InService`** status hingga Anda menghapus klaster. Sebaiknya hapus klaster apa pun yang dibuat menggunakan SageMaker instans sesuai permintaan saat tidak digunakan untuk menghindari biaya layanan lanjutan berdasarkan harga sesuai permintaan. Dalam tutorial ini, Anda telah membuat sebuah cluster yang terdiri dari dua kelompok instance. Salah satunya menggunakan instance C5, jadi pastikan Anda menghapus cluster dengan mengikuti instruksi dithe section called “Hapus SageMaker HyperPod klaster”.

Namun, jika Anda telah membuat klaster dengan kapasitas komputasi cadangan, status klaster tidak memengaruhi penagihan layanan.

Untuk membersihkan skrip siklus hidup dari bucket S3 yang digunakan untuk tutorial ini, buka bucket S3 yang Anda gunakan selama pembuatan cluster dan hapus file seluruhnya.

Jika Anda telah menguji menjalankan beban kerja apa pun di klaster, pastikan apakah Anda telah mengunggah data apa pun atau jika pekerjaan Anda menyimpan artefak apa pun ke bucket S3 atau layanan sistem file yang berbeda seperti Amazon FSx for Lustre dan Amazon Elastic File System. Untuk mencegah biaya yang timbul, hapus semua artefak dan data dari penyimpanan atau sistem file.

Menggunakan AWS CLI perintah untuk SageMaker HyperPod API

Buat SageMaker HyperPod cluster pertama Anda menggunakan AWS CLI perintah untuk HyperPod.

Buat SageMaker HyperPod klaster pertama Anda

Lihat petunjuk berikut tentang membuat SageMaker HyperPod cluster baru menggunakan AWS CLI command (`aws sagemaker create-cluster`), yang mengoperasikan SageMaker HyperPod `CreateCluster` API di backend.

1. Siapkan dan unggah skrip siklus hidup ke bucket S3 Anda untuk dijalankan di setiap grup instans selama pembuatan klaster.

```
s3://sagemaker-<unique-s3-bucket-name>/<lifecycle-script-directory>/src/
on_create.sh
```

> **Note**
>
> Anda harus `sagemaker-` mengawali jalur bucket S3, karena ??? with `AmazonSageMakerClusterInstanceRolePolicy` memungkinkan prinsipal untuk hanya mengakses bucket S3 dengan awalan tertentu.

Jika Anda tidak memiliki satu set skrip konfigurasi siklus hidup, gunakan skrip contoh yang disediakan di repositori Pelatihan Terdistribusi Awsome. GitHub Sub-langkah berikut menunjukkan cara mengunduh, apa yang harus dimodifikasi, dan cara mengunggah ke bucket S3.

a. Unduh salinan sampel skrip siklus hidup ke direktori di komputer lokal Anda.

```
git clone https://github.com/aws-samples/awsome-distributed-training/
```

b. Di bawah `1.architectures/5.sagemaker_hyperpods/LifecycleScripts/base-config`, Anda dapat menemukan satu set lengkap skrip siklus hidup. Untuk mempelajari lebih lanjut tentang sampel skrip siklus hidup, lihat juga. the section called “Templat skrip konfigurasi siklus hidup”

c. Buka dan edit file bernama `provisioning_params.json` untuk menentukan parameter dasar untuk penamaan sumber daya cluster. Misalnya, jika Anda membuat cluster dengan konfigurasi Slurm, `provisioning_params.json` harus mirip dengan berikut ini.

```
{
    "workload_manager": "slurm",
    "controller_group": "my-controller-group",
    "login_group": "my-login-group", // Optional
    "worker_groups": [
        {
            "instance_group_name": "worker-group-1",
            "partition_name": "partition-1"
        }
```

```
    ]
}
```

Anda dapat mengedit nilai kunci untuk`controller_group`,`login_group`, `instance_group_name` dan `partition_name` kunci ke nilai masing-masing.

d. Unggah skrip kes`3://sagemaker-<unique-s3-bucket-name>/<lifecycle-script-directory>/src`. Anda dapat melakukannya dengan menggunakan konsol S3, atau dengan menjalankan perintah AWS CLI S3 berikut.

```
aws s3 sync \
    ~/local-dir-for-lifecycle-scripts/* \
    s3://sagemaker-<unique-s3-bucket-name>/<lifecycle-script-directory>/src
```

2. Siapkan file CreateClusterpermintaan dalam format JSON. Contoh permintaan berikut didasarkan pada node yang diperlukan untuk grup yang didefinisikan `provisioning_params.json` dalam Langkah 1.2. Persyaratan minimum untuk Slurm adalah satu pengontrol dan satu kelompok pekerja. UntukExecutionRole, berikan ARN dari peran IAM yang Anda buat dengan Prasyarat yang dikelolaAmazonSageMakerClusterInstanceRolePolicy.

```
// create_cluster.json
{
   "ClusterName": "cluster-slurm-default-vpc",
   "InstanceGroups": [
       {
         "ExecutionRole": "string",
         "InstanceCount": number,
         "InstanceGroupName": "my-controller-group",
         "InstanceType": "string",
         "LifeCycleConfig": {
            "OnCreate": "on_create_lifecycle_script.sh",
            "SourceS3Uri": "s3://<your-s3-bucket>/<lifecycle-script-directory>/src"
         },
         "ThreadsPerCore": number
      },
      {
         "ExecutionRole": "string",
         "InstanceCount": number,
         "InstanceGroupName": "my-login-group",
         "InstanceType": "string",
         "LifeCycleConfig": {
```

```
            "OnCreate": "on_create_lifecycle_script.sh",
            "SourceS3Uri": "s3://<your-s3-bucket>/<lifecycle-script-directory>/src"
        },
        "ThreadsPerCore": number
      }
   ]
}
```

3. Jalankan perintah berikut untuk membuat cluster.

```
aws sagemaker create-cluster --cli-input-json file://complete/path/to/
create_cluster.json
```

Ini harus mengembalikan ARN dari cluster yang dibuat.

4. Jalankan `describe-cluster` untuk memeriksa status cluster.

```
aws sagemaker describe-cluster --cluster-name cluster-slurm-default-vpc
```

Setelah status cluster berubah**`InService`**, lanjutkan ke langkah berikutnya.

5. Jalankan `list-cluster-nodes` untuk memeriksa detail node cluster.

```
aws sagemaker list-cluster-nodes --cluster-name cluster-slurm-default-vpc
```

Ini mengembalikan respons, dan `InstanceId` itulah yang dibutuhkan pengguna klaster Anda untuk logging (`aws ssm`) ke dalamnya. Untuk informasi selengkapnya tentang masuk ke node cluster dan menjalankan beban kerja ML, lihatthe section called “Jalankan pekerjaan di HyperPod cluster”.

Hapus cluster dan sumber daya bersih

Setelah Anda berhasil menguji pembuatan SageMaker HyperPod klaster, klaster terus berjalan di **`InService`** status hingga Anda menghapus klaster. Kami menyarankan Anda menghapus klaster apa pun yang dibuat menggunakan SageMaker kapasitas sesuai permintaan saat tidak digunakan untuk menghindari biaya layanan lanjutan berdasarkan harga sesuai permintaan. Dalam tutorial ini, Anda telah membuat sebuah cluster yang terdiri dari dua kelompok instance. Salah satunya menggunakan instance C5, jadi pastikan Anda menghapus cluster dengan menjalankan perintah berikut.

```
aws sagemaker delete-cluster --cluster-name cluster-slurm-default-vpc
```

Untuk membersihkan skrip siklus hidup dari bucket S3 yang digunakan untuk tutorial ini, buka bucket S3 yang Anda gunakan selama pembuatan cluster dan hapus file seluruhnya.

Jika Anda telah menguji menjalankan beban kerja apa pun di klaster, pastikan apakah Anda telah mengunggah data apa pun atau jika pekerjaan Anda menyimpan artefak apa pun ke bucket S3 atau layanan sistem file yang berbeda seperti Amazon FSx for Lustre dan Amazon Elastic File System. Untuk mencegah biaya yang timbul, hapus semua artefak dan data dari penyimpanan atau sistem file.

Beroperasi SageMaker HyperPod

Bagian ini memandu Anda melalui cara mengoperasikan SageMaker HyperPod cluster melalui UI konsol atauAWS CLI.

Topik

- Menggunakan UI SageMaker HyperPod konsol
- Menggunakan CLI AWS

Menggunakan UI SageMaker HyperPod konsol

Topik berikut memberikan panduan tentang cara beroperasi SageMaker HyperPod melalui UI konsol.

Topik

- Buat SageMaker HyperPod cluster
- Jelajahi SageMaker HyperPod cluster Anda
- Lihat detail setiap SageMaker HyperPod cluster
- Mengedit SageMaker HyperPod klaster
- Hapus SageMaker HyperPod klaster

Buat SageMaker HyperPod cluster

Lihat petunjuk berikut tentang membuat SageMaker HyperPod cluster baru melalui UI SageMaker HyperPod konsol.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih HyperPod Cluster di panel navigasi kiri.
3. Di SageMaker HyperPod landing page, pilih Create cluster.
4. Pada Langkah 1: Pengaturan cluster, atur informasi dasar untuk cluster.
   a. Untuk nama Cluster, tentukan nama untuk cluster baru.
   b. Untuk Tag, tambahkan pasangan kunci dan nilai ke cluster baru dan kelola cluster sebagai AWS sumber daya. Untuk mempelajari lebih lanjut, lihat Menandai AWS sumber daya Anda.
5. Pada Langkah 2: Grup instance, pilih Buat grup instance. Setiap grup instans dapat dikonfigurasi secara berbeda, dan Anda dapat membuat klaster heterogen yang terdiri dari beberapa grup instans dengan berbagai jenis instance. Di jendela pop-up Create an instance group configuration, isi informasi konfigurasi grup instance.
   a. Untuk nama grup Instance, tentukan nama untuk grup instance.
   b. Untuk jenis instance Select, pilih instance untuk grup instance.
   c. Untuk Kuantitas, tentukan bilangan bulat yang tidak melebihi kuota instance untuk penggunaan klaster.
   d. Untuk jalur Amazon S3 ke file skrip siklus hidup, masukkan jalur S3 tempat skrip siklus hidup Anda disimpan.
   e. Untuk jalur Direktori ke skrip siklus hidup yang sedang dibuat, masukkan nama file skrip siklus hidup di bawah jalur S3 ke file skrip siklus hidup.
   f. Untuk peran IAM, pilih peran IAM yang telah Anda buat untuk SageMaker HyperPod sumber daya, mengikuti bagian tersebut. the section called “Menyiapkan pengguna dan peran IAM untuk SageMaker HyperPod pengguna dan sumber daya”
   g. Untuk konfigurasi Lanjutan, Anda dapat mengatur Thread per inti, tentukan 1 untuk menonaktifkan multithreading dan 2 untuk mengaktifkan multithreading. Untuk mengetahui jenis instans mana yang mendukung multithreading, lihat tabel referensi inti CPU dan utas per inti CPU per jenis instans di Panduan Pengguna Amazon EC2.
6. Pada Langkah 3: Konfigurasi lanjutan, konfigurasikan pengaturan jaringan opsional dalam cluster dan in-and-out cluster. Pilih VPC Anda sendiri jika Anda sudah memiliki VPC yang memberikan SageMaker akses ke VPC Anda. Jika Anda ingin membuat VPC baru, lihat Membuat VPC default atau Membuat VPC di Panduan Pengguna Amazon Virtual Private Cloud. Anda dapat membiarkannya sebagai tidak ada VPC untuk menggunakan SageMaker VPC default.

> **Note**
>
> Jika Anda ingin menggunakan VPC Anda sendiri, Anda harus menambahkan izin tambahan ke peran IAM untuk cluster. SageMaker HyperPod Untuk mempelajari selengkapnya, lihat the section called “(Opsional) Siapkan SageMaker HyperPod dengan VPC Amazon Anda”.

7. Pada Langkah 4: Tinjau dan buat, tinjau konfigurasi yang telah Anda tetapkan dari langkah 1 hingga 3 dan selesaikan pengiriman permintaan pembuatan cluster.

Jelajahi SageMaker HyperPod cluster Anda

Di bawah Cluster di halaman utama SageMaker HyperPod konsol, semua cluster yang dibuat akan muncul terdaftar di bawah bagian Cluster, yang menyediakan tampilan ringkasan cluster, ARN, status, dan waktu pembuatannya.

Lihat detail setiap SageMaker HyperPod cluster

Di bawah Cluster di halaman utama konsol, Nama cluster diaktifkan sebagai tautan. Pilih tautan nama cluster untuk melihat detail setiap cluster.

Mengedit SageMaker HyperPod klaster

1. Di bawah Cluster, pilih cluster yang ingin Anda perbarui.
2. Pilih tombol Tindakan, dan pilih Edit cluster.
3. Di <your-cluster>halaman Edit, Anda dapat mengedit konfigurasi dan tag grup instance. Setelah melakukan perubahan, pilih Kirim.
   a. Di bagian Konfigurasi grup instans, Anda dapat menambahkan lebih banyak grup instance dengan memilih Buat grup cluster.
   b. Di bagian Konfigurasi grup instans, Anda dapat memilih salah satu grup instans, dan memilih Edit untuk mengubah konfigurasinya.
   c. Di bagian Tag, Anda dapat memperbarui tag untuk cluster.

Hapus SageMaker HyperPod klaster

1. Di bawah Cluster, pilih cluster yang ingin Anda hapus.
2. Pilih Tindakan, dan pilih Hapus klaster.

3. Di jendela pop-up untuk penghapusan klaster, tinjau informasi klaster dengan hati-hati untuk mengonfirmasi bahwa Anda memilih klaster yang tepat untuk dihapus.
4. Setelah Anda meninjau informasi klaster, pilih Ya, hapus klaster.
5. Di bidang teks untuk mengonfirmasi penghapusan ini, ketik. **`delete`**
6. Pilih Hapus di sudut kanan bawah jendela pop-up untuk menyelesaikan pengiriman permintaan penghapusan cluster.

Menggunakan CLI AWS

Topik berikut memberikan panduan tentang menulis file permintaan SageMaker HyperPod API dalam format JSON dan menjalankannya menggunakan AWS CLI perintah.

Topik

- Buat cluster baru
- Jelaskan sebuah cluster
- Daftar rincian node cluster
- Jelaskan detail node cluster
- Daftar cluster
- Perbarui klaster
- Hapus klaster

Buat cluster baru

1. Siapkan skrip konfigurasi siklus hidup dan unggah ke bucket S3, seperti. `s3://sagemaker-<your-s3-bucket>/<lifecycle-script-directory>/src/` Langkah 2 berikut mengasumsikan bahwa ada skrip titik masuk yang dinamai `on_create.sh` dalam bucket S3 yang ditentukan.

   > **Important**
   >
   > Pastikan Anda mengatur jalur S3 untuk memulai`s3://sagemaker-`. the section called “Peran IAM untuk SageMaker HyperPod”Memiliki `AmazonSageMakerClusterInstanceRolePolicy`terlampir terkelola, yang memungkinkan akses ke bucket S3 dengan awalan tertentu. `sagemaker-`

2. Siapkan file permintaan CreateClusterAPI dalam format JSON. Contoh permintaan berikut didasarkan pada node yang diperlukan untuk grup yang didefinisikan `provisioning_params.json` dalam Langkah 1.2. Persyaratan minimum untuk Slurm adalah satu pengontrol dan satu kelompok pekerja. Untuk`ExecutionRole`, berikan ARN dari peran IAM yang Anda buat dengan yang dikelola `AmazonSageMakerClusterInstanceRolePolicy` dari bagian. the section called “Peran IAM untuk SageMaker HyperPod”

```
// create_cluster.json
{
    // Required
    "ClusterName": "your-hyperpod-cluster",
    // Required
    "InstanceGroups": [
        {
            "InstanceGroupName": "controller-group",
            "InstanceType": "ml.m5.xlarge",
            "InstanceCount": 1,
            "LifeCycleConfig": {
                "SourceS3Uri": "s3://sagemaker-<your-s3-bucket>/<lifecycle-script-
directory>/src/",
                "OnCreate": "on_create.sh"
            },
            "ExecutionRole": "arn:aws:iam::111122223333:role/iam-role-for-cluster",
            "ThreadsPerCore": 1
        },
        {
            "InstanceGroupName": "worker-group-1",
            "InstanceType": "ml.p4d.xlarge",
            "InstanceCount": 1,
            "LifeCycleConfig": {
                "SourceS3Uri": "s3://sagemaker-<your-s3-bucket>/<lifecycle-script-
directory>/src/",
                "OnCreate": "on_create.sh"
            },
            "ExecutionRole": "arn:aws:iam::111122223333:role/iam-role-for-cluster",
            "ThreadsPerCore": 1
        }
    ],
    // Optional
    "Tags": [
        {
```

```
            "Key": "string",
            "Value": "string"
        }
    ],
    // Optional
    "VpcConfig": {
        "SecurityGroupIds": [ "string" ],
        "Subnets": [ "string" ]
    }
}
```

Bergantung pada bagaimana Anda mendesain struktur klaster melalui skrip siklus hidup, Anda dapat menambahkan dan mengonfigurasi beberapa grup instance di bawah parameter permintaan. `InstanceGroups`

Untuk parameter `Tags` permintaan, Anda dapat menambahkan tag khusus untuk mengelola SageMaker HyperPod cluster sebagai AWS sumber daya. Anda dapat menambahkan tag ke klaster Anda dengan cara yang sama seperti Anda menambahkannya di AWS layanan lain yang mendukung penandaan. Untuk mempelajari selengkapnya tentang menandai AWS sumber daya secara umum, lihat Panduan Pengguna AWS Sumber Daya Tag.

Untuk parameter `VpcConfig` permintaan, Anda dapat menentukan VPC Anda sendiri. Untuk informasi selengkapnya, lihat the section called “(Opsional) Siapkan SageMaker HyperPod dengan VPC Amazon Anda”.

3. Jalankan perintah berikut untuk mengirimkan permintaan `CreateCluster` API.

```
aws sagemaker create-cluster --cli-input-json file://complete/path/to/
create_cluster.json
```

Ini harus mengembalikan ARN dari cluster baru.

Jelaskan sebuah cluster

Jalankan `describe-cluster` untuk memeriksa status cluster. Anda dapat menentukan nama atau ARN cluster.

```
aws sagemaker describe-cluster --cluster-name your-hyperpod-cluster
```

Setelah status cluster berubah**`InService`**, lanjutkan ke langkah berikutnya.

Daftar rincian node cluster

Jalankan `list-cluster-nodes` untuk memeriksa informasi kunci dari node cluster.

```
aws sagemaker list-cluster-nodes --cluster-name your-hyperpod-cluster
```

Ini mengembalikan respons, dan `InstanceId` itulah yang perlu Anda gunakan untuk logging (using`aws ssm`) ke dalamnya.

Jelaskan detail node cluster

Jalankan `describe-cluster-node` untuk mengambil rincian node cluster. Anda bisa mendapatkan ID node cluster dari list-cluster-nodes output. Anda dapat menentukan nama atau ARN cluster.

```
aws sagemaker describe-cluster-node \
    --cluster-name your-hyperpod-cluster \
    --node-id i-111222333444555aa
```

Daftar cluster

Jalankan `list-clusters` untuk mencantumkan semua cluster di akun Anda.

```
aws sagemaker list-clusters
```

Anda juga dapat menambahkan bendera tambahan untuk memfilter daftar cluster ke bawah. Untuk mempelajari selengkapnya tentang apa yang dijalankan perintah ini pada level rendah dan flag tambahan untuk pemfilteran, lihat referensi ListClustersAPI.

Perbarui klaster

Jalankan `update-cluster` untuk memperbarui konfigurasi cluster.

```
aws sagemaker update-cluster --cluster-name your-hyperpod-cluster
```

Untuk mempelajari lebih lanjut tentang perintah ini dan flag yang tersedia, lihat referensi UpdateClusterAPI.

Hapus klaster

Jalankan `delete-cluster` untuk menghapus cluster. Anda dapat menentukan nama atau ARN cluster.

```
aws sagemaker delete-cluster --cluster-name your-hyperpod-cluster
```

Jalankan pekerjaan di SageMaker HyperPod cluster

Bagian ini untuk pengguna cluster seperti ilmuwan machine learning (ML). Ini memberikan contoh menjalankan beban kerja MLpada cluster yang disediakan SageMaker HyperPod . Bergantung pada cara Anda mengatur lingkungan di HyperPod cluster Anda, ada banyak cara untuk menjalankan beban kerja ML pada HyperPod cluster. Contoh menjalankan beban kerja ML pada HyperPod cluster disediakan di repositori Pelatihan Terdistribusi Awsome. GitHub Topik berikut memandu Anda melalui cara masuk ke HyperPod kluster yang disediakan dan membantu Anda memulai menjalankan beban kerja MS sampel.

Topik

- Akses node SageMaker HyperPod cluster
- Jadwalkan pekerjaan Slurm di cluster SageMaker HyperPod
- Jadwalkan pekerjaan untuk beban kerja pelatihan terdistribusi pada SageMaker HyperPod

Akses node SageMaker HyperPod cluster

Anda dapat mengakses InServicecluster Anda melalui AWS Systems Manager (SSM) dengan menjalankan AWS CLI perintah `aws ssm start-session` dengan nama host SageMaker HyperPod cluster dalam format. `sagemaker-cluster:[cluster-id]_[instance-grou-name]-[instance-id]` Anda dapat mengambil ID cluster dan ID instance dari SageMaker HyperPod konsol atau dengan menjalankan AWS CLIperintah untuk SageMaker HyperPod. Misalnya, jika ID cluster Anda`aaa11bbbbb222`, nama node cluster adalah`controller-group`, dan ID node cluster adalah`i-111222333444555aa`, `start-session` perintah SSM harus sebagai berikut.

> **Note**
>
> Jika Anda belum menyiapkanAWS Systems Manager, ikuti instruksi yang diberikan dithe section called “Siapkan AWS Systems Manager dan Jalankan Sebagai untuk kontrol akses pengguna cluster”.

```
aws ssm start-session \
    --target sagemaker-cluster:aa11bbbbb222_controller-group-i-111222333444555aa \
```

```
    --region us-west-2
```

Jadwalkan pekerjaan Slurm di cluster SageMaker HyperPod

Anda dapat meluncurkan pekerjaan pelatihan menggunakan Slurm standar `sbatch` atau `srun` perintah. Misalnya, untuk meluncurkan pekerjaan pelatihan 8-node, Anda dapat menjalankan pelatihan `srun -N 8 --exclusive train.sh` SageMaker HyperPod pendukung di berbagai lingkungan, termasuk,`conda`, `venv``docker`, dan`enroot`. Anda dapat mengonfigurasi lingkungan ML dengan menjalankan skrip siklus hidup di kluster Anda. SageMaker HyperPod Anda juga memiliki opsi untuk melampirkan sistem file bersama seperti FSx, yang juga dapat digunakan sebagai lingkungan virtual.

Contoh berikut menunjukkan cara menjalankan pekerjaan untuk melatih Llama-2 dengan teknik Fully Sharded Data Parallelism (FSDP) pada cluster dengan sistem file Amazon FSx bersama SageMaker HyperPod . Anda juga dapat menemukan lebih banyak contoh dari GitHub repositori Pelatihan Terdistribusi Awsome.

> **Tip**
>
> Semua SageMaker HyperPod contoh tersedia di `3.test_cases` folder GitHub repositori Pelatihan Terdistribusi Awsome.

1. Kloning GitHub repositori Pelatihan Terdistribusi Awsome, dan salin contoh pekerjaan pelatihan ke sistem file Amazon FSx bersama Anda.

   ```
   TRAINING_DIR=/fsx/users/my-user/fsdp
   git clone https://github.com/aws-samples/awsome-distributed-training/
   cp -R awsome-distributed-training/3.test_cases/10.FSDP $TRAINING_DIR
   ```

2. Buat `conda` lingkungan di sistem file Amazon FSx bersama Anda. Pastikan bahwa sistem file dapat diakses oleh semua node di cluster.

   ```
   #!/usr/bin/env bash
   set -ex

   wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
   chmod +x Miniconda3-latest-Linux-x86_64.sh
   ./Miniconda3-latest-Linux-x86_64.sh -b -f -p ./miniconda3
   ```

```
source ./miniconda3/bin/activate

conda create -y -n pt_fsdp python=3.10

source activate pt_fsdp

# Install PyTorch
pip install torch torchvision torchaudio
pip install packaging transformers accelerate ninja tensorboard h5py datasets

# create output dir
mkdir tensorboard
mkdir checkpoints
```

Simpan skrip ini ke volume bersama. Tutorial ini mengasumsikan bahwa itu disimpan sebagai/`fsx/users/my_user/create_env.sh`.

3. Bangun lingkungan virtual dengan meluncurkan pekerjaan slurm node tunggal.

```
srun -N 1 /fsx/users/my_user/create_env.sh
```

4. Setelah lingkungan dibangun, Anda dapat meluncurkan pekerjaan pelatihan dengan menunjuk ke jalur lingkungan pada volume bersama. Anda dapat meluncurkan pekerjaan pelatihan single-node dan multi-node dengan pengaturan yang sama. Untuk meluncurkan pekerjaan, buat skrip peluncur pekerjaan (juga disebut skrip titik masuk) sebagai berikut.

```
#!/usr/bin/env bash
set -ex

ENV_PATH=/fsx/users/my_user/pytorch_env
TORCHRUN=$ENV_PATH/bin/torchrun
TRAINING_SCRIPT=/fsx/users/my_user/pt_train.py

WORLD_SIZE_JOB=$SLURM_NTASKS
RANK_NODE=$SLURM_NODEID
PROC_PER_NODE=8
MASTER_ADDR=(`scontrol show hostnames \$SLURM_JOB_NODELIST | head -n 1`)
MASTER_PORT=$(expr 10000 + $(echo -n $SLURM_JOBID | tail -c 4))

DIST_ARGS="--nproc_per_node=$PROC_PER_NODE \
           --nnodes=$WORLD_SIZE_JOB \
           --node_rank=$RANK_NODE \
           --master_addr=$MASTER_ADDR \
```

```
            --master_port=$MASTER_PORT \
          "

$TORCHRUN $DIST_ARGS $TRAINING_SCRIPT
```

> **Tip**
>
> Jika Anda ingin membuat pekerjaan pelatihan Anda lebih tangguh terhadap kegagalan perangkat keras dengan menggunakan kemampuan resume otomatis SageMaker HyperPod, Anda perlu mengatur variabel lingkungan dengan benar `MASTER_ADDR` di skrip entrypoint. Untuk mempelajari selengkapnya, lihat the section called “Lanjutkan otomatis”.

Tutorial ini mengasumsikan bahwa script ini disimpan sebagai`/fsx/users/my_user/train.sh`.

5. Dengan skrip ini dalam volume bersama di`/fsx/users/my_user/train.sh`, jalankan `srun` perintah berikut untuk menjadwalkan pekerjaan Slurm.

```
cd /fsx/users/my_user/
srun -N 8 train.sh
```

Jadwalkan pekerjaan untuk beban kerja pelatihan terdistribusi pada SageMaker HyperPod

Perpustakaan paralelisme data SageMaker terdistribusi (SMDDP) adalah perpustakaan komunikasi kolektif yang meningkatkan kinerja komputasi pelatihan paralel data terdistribusi. Perpustakaan SMDDP menangani overhead komunikasi dari operasi komunikasi kolektif utama dengan menawarkan yang berikut ini untuk. SageMaker HyperPod

1. Perpustakaan menawarkan `AllGather` dioptimalkan untukAWS. `AllGather`adalah operasi kunci yang digunakan dalam pelatihan paralelisme data sharded, yang merupakan teknik paralelisme data hemat memori yang ditawarkan oleh perpustakaan populer seperti perpustakaan SageMaker model paralelisme (SMP), DeepSpeed Zero Redundancy Optimizer (Zero), dan Fully Sharded Data Parallelism (FSDP). PyTorch
2. Pustaka melakukan node-to-node komunikasi yang dioptimalkan dengan sepenuhnya memanfaatkan infrastruktur AWS jaringan dan topologi instance SageMaker ML.

Menggunakan SMDDP pada SageMaker HyperPod

Berikut ini adalah persyaratan lingkungan pelatihan untuk menggunakan perpustakaan SMDDP pada. SageMaker HyperPod

- `libstdc++`versi runtime lebih besar dari 3.
- PyTorch 2.0.1 dengan cuda 11.8
- Python 3.10.x
- `ml.p4d.24xlarge`dan`ml.p4de.24xlarge`, yang didukung jenis instance oleh perpustakaan SMDDP
- `imdsv2`diaktifkan pada host pelatihan

Tergantung di mana Anda menjalankan pekerjaan pelatihan terdistribusi, Anda dapat menggunakan perpustakaan SMDDP sebagai berikut.

Untuk menginstal perpustakaan SMDDP di DLAMI SageMaker HyperPod

- `pip install --no-cache-dir https://smdataparallel.s3.amazonaws.com/binary/pytorch/2.0.1/cu118/2023-11-17/smdistributed_dataparallel-2.0.2-cp310-cp310-linux_x86_64.whl`

> **Note**
>
> Jika bekerja di lingkungan conda, pastikan PyTorch itu diinstal melalui `conda install` bukan`pip`.
>
> ```
> conda install pytorch==2.0.1  torchvision==0.15.2 torchaudio==2.0.2 pytorch-cuda=11.8 -c pytorch -c nvidia
> ```

Untuk menggunakan perpustakaan SMDDP pada wadah Docker

- Pustaka SMDDP sudah diinstal sebelumnya pada SageMaker Deep Learning Containers (DLC). Untuk menemukan daftar DLC SageMaker kerangka kerja PyTorch dengan pustaka SMDDP, lihat Kerangka Kerja yang Didukung dalam dokumentasi pustaka paralelisme data. SageMaker Anda juga dapat membawa wadah Docker Anda sendiri dengan dependensi yang diperlukan diinstal untuk menggunakan perpustakaan SMDDP. Untuk mempelajari lebih lanjut tentang menyiapkan

kontainer Docker khusus untuk menggunakan pustaka SMDDP, lihat juga. the section called “Buat wadah docker Anda sendiri dengan perpustakaan”

> **Important**
>
> Untuk menggunakan pustaka SMDDP dalam wadah Docker, Anda harus mengikat mount `/var/log` direktori dari mesin host ke `/var/log` dalam wadah. Ini dapat dilakukan dengan menambahkan opsi berikut saat menjalankan wadah Anda.
>
> ```
> docker run <OTHER_OPTIONS> -v /var/log:/var/log ...
> ```

Sesuaikan skrip PyTorch pelatihan Anda untuk memanfaatkan perpustakaan SMDDP

Mulai dari perpustakaan paralelisme data SageMaker terdistribusi (SMDDP) v1.4.0, Anda dapat menggunakan pustaka sebagai opsi backend untuk paket terdistribusi. PyTorch Untuk menggunakan SMDDP `AllReduce` dan operasi `AllGather` kolektif, Anda hanya perlu mengimpor perpustakaan SMDDP di awal skrip pelatihan Anda dan menetapkan SMDDP sebagai backend modul terdistribusi selama inisialisasi grup proses. PyTorch Dengan satu baris spesifikasi backend, Anda dapat menyimpan semua modul PyTorch terdistribusi asli dan seluruh skrip pelatihan tidak berubah. Cuplikan kode berikut menunjukkan cara menggunakan pustaka SMDDP sebagai backend paket pelatihan terdistribusi PyTorch berbasis: distributed PyTorchdata parallel (DDP), PyTorch full sharded data parallelism (FSDP), dan Megatron-. DeepSpeedDeepSpeed

Untuk PyTorch DDP atau FSDP

Inisialisasi kelompok proses sebagai berikut.

```
import torch.distributed as dist
import smdistributed.dataparallel.torch.torch_smddp

dist.init_process_group(backend="smddp")
```

> **Note**
>
> (Hanya untuk pekerjaan PyTorch DDP) `smddp` Backend saat ini tidak mendukung pembuatan grup subproses dengan API. `torch.distributed.new_group()` Anda juga tidak dapat

menggunakan `smddp` backend secara bersamaan dengan backend grup proses lainnya seperti dan. `NCCL` `Gloo`

Untuk DeepSpeed atau Megatron- DeepSpeed

Inisialisasi kelompok proses sebagai berikut.

```
import deepspeed
import smdistributed.dataparallel.torch.torch_smddp

deepspeed.init_distributed(dist_backend="smddp")
```

Note

Untuk menggunakan SMDDP `AllGather` dengan peluncur `mpirun` berbasis (`smdistributeddanpytorchddp`) dithe section called “Langkah 2: Luncurkan pekerjaan pelatihan terdistribusi”, Anda juga perlu mengatur variabel lingkungan berikut dalam skrip pelatihan Anda.

```
export SMDATAPARALLEL_OPTIMIZE_SDP=true
```

Untuk panduan umum tentang menulis skrip pelatihan PyTorch FSDP, lihat Pelatihan Model Lanjutan dengan Paralel Data Berbagi Penuh (FSDP) dalam dokumentasi. PyTorch

Untuk panduan umum tentang menulis skrip pelatihan PyTorch DDP, lihat Memulai dengan data terdistribusi paralel dalam PyTorch dokumentasi.

SageMaker HyperPod ketahanan klaster

SageMaker HyperPod menyediakan fitur ketahanan cluster berikut.

Topik

- Pemeriksaan kesehatan cluster
- Lanjutkan otomatis
- Cara mengganti instance yang salah di luar resume otomatis SageMaker HyperPod

Pemeriksaan kesehatan cluster

Bagian ini menjelaskan serangkaian pemeriksaan kesehatan yang SageMaker HyperPod digunakan untuk secara teratur memantau kesehatan instance cluster untuk masalah dengan perangkat seperti akselerator (inti GPU dan Trainium) dan jaringan (EFA).

Kategori	Nama utilitas	Kompatibilitas tipe instans	Deskripsi
Akselerator	Kebijakan DCGM	GPU	Setiap instance di cluster terus memantau semua kebijakan terkait GPU termasuk kesalahan XID dengan NVIDIA DCGM.
	NVIDIA SMI	GPU	utilitas nvidia-smi adalah CLI terkenal untuk mengelola dan memantau GPU. Pemeriksa kesehatan bawaan mem-parsing output dari `nvidia-smi` untuk menentukan kesehatan instance.
	Sysfs neuron	Trainium	Untuk instance yang didukung Trainium, kesehatan perangkat Neuron ditentukan dengan membaca penghitung dari Sysf Neuron yang disebarkan langsung oleh driver Neuron.

Jaringan	EFA	GPU dan Trainium	Untuk membantu diagnostik perangkat Elastic Fabric Adapter (EFA), pemeriksa kesehatan EFA menjalank an serangkaian tes konektivitas menggunakan semua kartu EFA yang tersedia dalam instans.
Stres	Diagnostik DCGM	GPU	Diagnostik DCGM level 2 digunakan untuk melatih GPU dalam sistem dan menempatkan mereka di bawah tekanan untuk mendapatkan wawasan menyeluruh tentang kesehatan.
	Stres CPU	GPU dan Trainium	Kesehatan CPU ditentuka n menggunakan alat stress Linux, yang menjalankan beberapa utas untuk mencapai pemanfaat an CPU 100% dan melakukan operasi I/ O.

Lanjutkan otomatis

Bagian ini menjelaskan cara menjalankan pekerjaan pelatihan dengan fungsionalitas SageMaker HyperPod auto-resume, yang menyediakan infrastruktur ketahanan tanpa sentuhan untuk secara otomatis memulihkan pekerjaan pelatihan dari pos pemeriksaan terakhir yang disimpan jika terjadi kegagalan perangkat keras untuk cluster dengan lebih dari 16 node.

Dengan fungsionalitas auto-resume, jika pekerjaan gagal karena kegagalan perangkat keras atau masalah sementara di antara pelatihan, SageMaker HyperPod auto-resume memulai alur kerja penggantian node dan memulai ulang pekerjaan setelah node yang salah diganti.

Menggunakan fungsi SageMaker HyperPod auto-resume dengan Slurm

Bila Anda menggunakan SageMaker HyperPod auto-resume dengan Slurm, Anda harus menjalankan pekerjaan di dalam alokasi eksklusif yang diperoleh baik dengan menggunakan atau. `salloc` `sbatch` Bagaimanapun, Anda perlu memodifikasi skrip entrypoint untuk memastikan bahwa semua langkah penyiapan berjalan dalam satu `srun` perintah saat melanjutkan pekerjaan. Melalui skrip entrypoint, penting untuk mengatur lingkungan pada node yang diganti agar konsisten dengan lingkungan tempat langkah pekerjaan dijalankan sebelum dihentikan. Preseden berikut menunjukkan cara menyiapkan skrip entrypoint untuk menjaga lingkungan tetap konsisten dan menjalankannya sebagai satu perintah. `srun`

> **Tip**
>
> Jika Anda menggunakan`sbatch`, Anda dapat menjaga skrip batch sederhana dengan membuat skrip terpisah untuk mengatur lingkungan dan menggunakan satu `srun` perintah.

1. Buat skrip menggunakan contoh kode berikut dan simpan sebagai`train_auto_resume.sh`. Skrip ini menyebarkan pengaturan lingkungan pelatihan dengan asumsi bahwa tidak ada konfigurasi manual yang sebelumnya dibuat untuk node yang diganti. Ini memastikan bahwa lingkungan adalah node-agnostik, sehingga ketika sebuah node diganti, lingkungan yang sama disediakan pada node sebelum melanjutkan pekerjaan.

   > **Note**
   >
   > Contoh kode berikut menunjukkan bagaimana menemukan daftar simpul Slurm yang terkait dengan pekerjaan. Jangan gunakan variabel `$SLURM_JOB_NODELIST` lingkungan yang disediakan oleh Slurm, karena nilainya mungkin sudah usang

setelah melanjutkan pekerjaan SageMaker HyperPod secara otomatis. Contoh kode berikut menunjukkan bagaimana mendefinisikan `NODE_LIST` variabel baru untuk menggantikan`SLURM_JOB_NODELIST`, dan kemudian mengatur `MASTER_NODE` dan `MASTER_ADDR` variabel off dari `NODE_LIST` variabel.

```
#!/bin/bash

# Filename: train_auto_resume.sh
# Sample containerized script to launch a training job with a single srun which can
 be auto-resumed.

# Place your training environment setup here.
# Example: Install conda, docker, activate virtual env, etc.

# Get the list of nodes for a given job
NODE_LIST=$(scontrol show jobid=$SLURM_JOBID | \ # Show details of the SLURM job
            awk -F= '/NodeList=/{print $2}' | \  # Extract NodeList field
            grep -v Exc)                         # Exclude nodes marked as excluded

# Determine the master node from the node list
MASTER_NODE=$(scontrol show hostname $NODE_LIST | \ # Convert node list to hostnames
              head -n 1)                              # Select the first hostname as
 master node

# Get the master node address
MASTER_ADDR=$(scontrol show node=$MASTER_NODE | \ # Show node information
              awk -F= '/NodeAddr=/{print $2}' | \ # Extract NodeAddr
              awk '{print $1}')                   # Print the first part of NodeAddr


# Torchrun command to launch the training job
torchrun_cmd="torchrun --nnodes=$SLURM_NNODES \
                       --nproc_per_node=1 \
                       --node_rank=$SLURM_NODE \
                       --master-addr=$MASTER_ADDR \
                       --master_port=1234 \
                       <your_training_script.py>"

# Execute the torchrun command in the 'pytorch' Conda environment,
# streaming output live
```

```
/opt/conda/bin/conda run --live-stream -n pytorch $torchrun_cmd
```

> **Tip**
>
> Anda dapat menggunakan skrip sebelumnya untuk menambahkan lebih banyak perintah untuk menginstal dependensi tambahan apa pun untuk pekerjaan Anda. Namun, kami menyarankan agar Anda menyimpan skrip penginstalan dependensi ke kumpulan skrip siklus hidup yang digunakan selama pembuatan klaster. Jika Anda menggunakan lingkungan virtual yang dihosting di direktori bersama, Anda juga dapat menggunakan skrip ini untuk mengaktifkan lingkungan virtual.

2. Luncurkan pekerjaan dengan SageMaker HyperPod resume otomatis diaktifkan dengan menambahkan tanda `--auto-resume=1` untuk menunjukkan bahwa `srun` perintah harus dicoba ulang secara otomatis jika terjadi kegagalan perangkat keras.

> **Note**
>
> Jika Anda telah menyiapkan alokasi sumber daya menggunakan `sbatch` atau`salloc`, Anda dapat menjalankan beberapa `srun` perintah dalam alokasi. Jika terjadi kegagalan, fungsi SageMaker HyperPod auto-resume hanya beroperasi pada langkah pekerjaan saat ini dari `srun` perintah dengan bendera`--auto-resume=1`. Dengan kata lain, mengaktifkan auto-resume dalam perintah tidak berlaku untuk `srun` `srun` perintah lain yang diluncurkan dalam sesi alokasi sumber daya.

Berikut ini adalah contoh `srun` perintah dengan `auto-resume` diaktifkan.

Menggunakan sbatch

Karena sebagian besar logika untuk mengatur lingkungan sudah ada`train_auto_resume.sh`, skrip batch harus sederhana dan mirip dengan contoh kode berikut. Asumsikan bahwa skrip batch berikut disimpan sebagai`batch.sh`.

```
#!/bin/bash
#SBATCH --nodes 2
#SBATCH --exclusive
srun --auto-resume=1 train_auto_resume.sh
```

Jalankan skrip batch sebelumnya menggunakan perintah berikut.

```
sbatch batch.sh
```

Menggunakan salloc

Mulailah dengan memperoleh alokasi eksklusif, dan jalankan `srun` perintah dengan `--auto-resume` flag dan skrip entrypoint.

```
salloc -N 2 --exclusive
srun --auto-resume=1 train_auto_resume.sh
```

Cara mengganti instance yang salah di luar resume otomatis SageMaker HyperPod

Penggantian node hanya berfungsi dengan penyediaan cluster Slurm oleh. SageMaker HyperPod Dalam hal ini, Anda mungkin perlu meminta penggantian node dengan menempatkan node yang tidak sehat di salah satu status berikut: `"Down"` `"Drained"`,, atau `"Fail"` dengan alasan`"Action:Replace"`. Pengguna slurm dengan hak administrator dapat melakukan ini dengan memanggil perintah berikut.

> **Warning**
>
> Lanjutkan dengan hati-hati saat Anda menjalankan perintah ini. Node menjadi tidak dapat digunakan sampai penggantian selesai.

```
scontrol update node=<ip-ipv4> state=fail reason="Action:Replace"
```

Dalam contoh perintah sebelumnya, `ip-ipv4` adalah nama simpul Slurm yang perlu Anda tentukan, yang sama dengan nama host instance.

Setelah menjalankan perintah ini, node Anda masuk ke status yang ditentukan (Anda dapat memantau status menggunakan`sinfo`) dan diganti menggunakan nama host yang sama. Proses ini mungkin memakan waktu beberapa menit tergantung pada instance yang tersedia di Availability Zone Anda dan waktu yang diperlukan untuk menjalankan skrip siklus hidup. Saat proses aktif, hindari mengubah status node secara manual atau memulai ulang pengontrol Slurm; melakukannya dapat

menyebabkan kegagalan. Jika node tidak kembali ke keadaan semula (idle) setelah waktu yang lama (lebih dari 1 jam), hubungi AWSSupport.

SageMaker HyperPod manajemen klaster

Topik berikut membahas pencatatan dan pengelolaan SageMaker HyperPod cluster.

SageMaker HyperPod Peristiwa pencatatan

Semua peristiwa dan log dari SageMaker HyperPod disimpan ke Amazon CloudWatch di bawah nama grup log`/aws/sagemaker/Clusters/[ClusterName]/[ClusterID]`. Setiap panggilan ke `CreateCluster` API membuat grup log baru. Daftar berikut berisi semua aliran log yang tersedia yang dikumpulkan di setiap grup log.

Nama Grup Log	Nama Aliran Log
`/aws/sagemaker/Clusters/[ClusterName]/[ClusterID]`	`LifecycleConfig/[instance-group-name]/[instance-id]`

Logging SageMaker HyperPod di tingkat instans

Anda dapat mengakses LifecycleScript log yang dipublikasikan CloudWatch selama konfigurasi instance cluster. Setiap instance dalam cluster yang dibuat menghasilkan aliran log terpisah, dapat dibedakan berdasarkan formatnya. `LifecycleConfig/[instance-group-name]/[instance-id]`

Semua log yang ditulis untuk `/var/log/provision/provisioning.log` diunggah ke aliran sebelumnya CloudWatch . Sampel LifecycleScripts saat `1.architectures/5.sagemaker_hyperpods/LifecycleScripts/base-config`mengarahkan mereka `stdout` dan `stderr` ke lokasi ini. Jika Anda menggunakan skrip kustom Anda, tulis log Anda ke `/var/log/provision/provisioning.log` lokasi agar CloudWatch tersedia.

Pemberian tag pada sumber daya

AWSSistem penandaan membantu mengelola, mengidentifikasi, mengatur, mencari, dan memfilter sumber daya. SageMaker HyperPod mendukung penandaan, sehingga Anda dapat mengelola

cluster sebagai AWS sumber daya. Selama pembuatan klaster atau pengeditan cluster yang ada, Anda dapat menambahkan atau mengedit tag untuk klaster. Untuk mempelajari selengkapnya tentang penandaan secara umum, lihat Menandai sumber daya Anda AWS.

Menggunakan UI SageMaker HyperPod konsol

Saat Anda membuat cluster baru dan mengedit cluster, Anda dapat menambahkan, menghapus, atau mengedit tag.

Menggunakan SageMaker HyperPod API

Saat Anda menulis file permintaan UpdateClusterAPI CreateClusteratau dalam format JSON, edit `Tags` bagian tersebut.

Menggunakan perintah AWS CLI penandaan untuk SageMaker

Untuk menandai sebuah cluster

Gunakan `aws sagemaker add-tags`sebagai berikut.

```
aws sagemaker add-tags --resource-arn cluster_ARN --tags Key=string,Value=string
```

Untuk menghapus tag sebuah cluster

Gunakan `aws sagemaker delete-tags`sebagai berikut.

```
aws sagemaker delete-tags --resource-arn cluster_ARN --tag-keys "tag_key"
```

Untuk membuat daftar tag untuk sumber daya

Gunakan `aws sagemaker list-tags`sebagai berikut.

```
aws sagemaker list-tags --resource-arn cluster_ARN
```

SageMaker HyperPod referensi

Gunakan halaman referensi ini untuk menemukan informasi dan referensi yang lebih rinci SageMaker HyperPod.

SageMaker HyperPod harga

Topik berikut memberikan informasi tentang SageMaker HyperPod harga. Untuk mengetahui detail lebih lanjut tentang harga per jam untuk menggunakan SageMaker HyperPod instans, lihat juga SageMaker Harga Amazon.

Permintaan kapasitas

Anda dapat mengalokasikan kapasitas komputasi sesuai permintaan atau cadangan SageMaker untuk digunakan pada. SageMaker HyperPod Pembuatan klaster sesuai permintaan mengalokasikan kapasitas yang tersedia dari kumpulan kapasitas SageMaker sesuai permintaan. Atau, Anda dapat meminta kapasitas yang dipesan untuk memastikan akses dengan mengirimkan tiket untuk peningkatan kuota. Permintaan kapasitas masuk diprioritaskan oleh SageMaker dan Anda menerima perkiraan waktu untuk alokasi kapasitas.

Layanan penagihan

Ketika Anda menyediakan kapasitas komputasi aktif SageMaker HyperPod, Anda ditagih selama durasi alokasi kapasitas. SageMaker HyperPod tagihan muncul di tagihan ulang tahun Anda dengan item baris untuk jenis alokasi kapasitas (sesuai permintaan, cadangan), jenis instans, dan waktu yang dihabiskan untuk menggunakan instans.

Untuk mengirimkan tiket untuk kenaikan kuota, lihatthe section called “SageMaker HyperPod kuota”.

SageMaker HyperPod API

Daftar berikut adalah set lengkap SageMaker HyperPod API untuk mengirimkan permintaan tindakan dalam format JSON ke SageMaker melalui atau. AWS CLI AWS SDK for Python (Boto3)

- CreateCluster
- DeleteCluster
- DescribeCluster
- DescribeClusterNode
- ListClusterNodes
- ListClusters
- UpdateCluster

SageMaker HyperPod DLAMI

SageMaker HyperPod menjalankan SageMaker HyperPod DLAMI, yang dibangun di atas AWSDeep Learning Base GPU AMI (Ubuntu 20.04).

SageMaker HyperPod DLAMI dibundel dengan paket tambahan berikut.

- Sistem File Jaringan (NFS)
- Munge: 0.5.15
- Buburan: 23.2.3
- `aws-neuronx-dkms`: 2. *
- `aws-neuronx-collectives`: 2. *
- `aws-neuronx-runtime-lib`: 2. *
- `aws-neuronx-tools`: 2. *
- SageMaker HyperPod paket perangkat lunak cluster untuk mendukung fitur seperti pemeriksaan kesehatan cluster dan auto-resume

Templat skrip konfigurasi siklus hidup

Contoh skrip konfigurasi siklus hidup disediakan di repositori Pelatihan Terdistribusi Awsome. GitHub

```
git clone https://github.com/aws-samples/awsome-distributed-training/
```

Templat skrip konfigurasi slurm

Templat skrip siklus hidup untuk pekerjaan Slurm tersedia di bawah. `1.architectures/5.sagemaker_hyperpods/LifecycleScripts/base-config`

- `add_users.sh`: Template yang dapat Anda gunakan untuk menambahkan pengguna Slurm. Ini membaca pengguna dari `shared_users_samples.txt` dan menciptakan pengguna Linux yang sesuai.
- `lifecycle_script.py`: Skrip Python untuk konfigurasi siklus hidup untuk Slurm.
- `mount_fsx.sh`: Skrip untuk memasang `fsx` volume. Ini juga menjalankan layanan untuk secara otomatis memeriksa mount, dan memasang kembali jika perlu.
- `on_create.sh`: Diperlukan. Ini adalah skrip titik masuk, templat yang dapat Anda gunakan untuk menyiapkan konfigurasi simpul umum. Inilah yang Anda sediakan untuk `CreateCluster` dan`UpdateCluster`.

- `provisioning_parameters.json`: File konfigurasi inti untuk memberi nama node cluster untuk mengatur Slurm.
- `setup_mariadb_accounting.sh`: Skrip pengaturan untuk MariaDB untuk penyediaan akuntansi Slurm.
- `shared_users_sample.txt`: Contoh file teks untuk menambahkan pengguna bersama Slurm.
- `start_slurm.sh`: Script ini memulai slurm controller tergantung pada jenis node Slurm. Ini dimulai `slurmctld` untuk node kepala, dan `slurmd` untuk node komputasi dan login.

SageMaker HyperPod Referensi izin API

Saat menyiapkan kontrol akses untuk memungkinkan menjalankan operasi SageMaker HyperPod API dan menulis kebijakan izin yang dapat dilampirkan ke pengguna IAM untuk administrator cloud, gunakan tabel berikut sebagai referensi.

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber Daya
CreateCluster	`sagemaker:CreateCluster`	`arn:aws:sagemaker:region:account-id:cluster/cluster-id`
DeleteCluster	`sagemaker:DeleteCluster`	`arn:aws:sagemaker:region:account-id:cluster/cluster-id`
DescribeCluster	`sagemaker:DescribeCluster`	`arn:aws:sagemaker:region:account-id:cluster/cluster-id`
DescribeClusterNode	`sagemaker:DescribeClusterNode`	`arn:aws:sagemaker:region:account-id:cluster/cluster-id`

ListClusterNodes	`sagemaker:ListClusterNodes`	`arn:aws:sagemaker:region:account-id:cluster/cluster-id`
ListClusters	`sagemaker:ListClusters`	`arn:aws:sagemaker:region:account-id:cluster/cluster-id`
UpdateCluster	`sagemaker:UpdateCluster`	`arn:aws:sagemaker:region:account-id:cluster/cluster-id`

Untuk daftar lengkap izin dan jenis sumber daya untuk SageMaker API, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon SageMaker di Referensi Otorisasi AWS Layanan.

SageMaker HyperPod perintah di AWS CLI

Berikut ini adalah AWS CLI perintah SageMaker HyperPod untuk menjalankan operasi HyperPod API inti.

- buat-cluster
- hapus-cluster
- deskripsikan-cluster
- describe-cluster-node
- list-cluster-nodes
- daftar-cluster
- perbaruan-kluster

SageMaker HyperPod Modul Python di AWS SDK for Python (Boto3)

Berikut ini adalah metode AWS SDK for Python (Boto3) klien SageMaker untuk menjalankan operasi HyperPod API inti.

- create_cluster
- delete_cluster
- deskripsi_cluster
- deskripsi_cluster_node
- list_cluster_nodes
- list_cluster
- update_cluster

Catatan SageMaker HyperPod rilis Amazon

Lihat catatan rilis berikut untuk melacak pembaruan terbaru untuk Amazon SageMaker HyperPod.

SageMaker HyperPod Catatan Rilis: 29 November 2023

- Meluncurkan Amazon SageMaker HyperPod di AWS re:Invent 2023.

Gunakan AI generatif di lingkungan SageMaker notebook

Jupyter AI adalah ekstensi sumber terbuka untuk JupyterLab mengintegrasikan kemampuan AI generatif ke dalam notebook Jupyter. Melalui antarmuka obrolan Jupyter AI dan perintah ajaib, pengguna bereksperimen dengan kode yang dihasilkan dari instruksi bahasa alami, menjelaskan kode yang ada, mengajukan pertanyaan tentang file lokal mereka, membuat seluruh buku catatan, dan banyak lagi. Ekstensi menghubungkan notebook Jupyter dengan model bahasa besar (LLM) yang dapat digunakan pengguna untuk menghasilkan teks, kode, atau gambar, dan untuk mengajukan pertanyaan tentang data mereka sendiri. Jupyter AI mendukung penyedia model generatif seperti AI21, Anthropic, (AWSdan SageMaker JumpStart Amazon Bedrock), Cohere, dan OpenAI.

Paket ekstensi disertakan dalam SageMaker Distribusi Amazon versi 1.2 dan seterusnya. Amazon SageMaker Distribution adalah lingkungan Docker untuk ilmu data dan komputasi ilmiah yang digunakan sebagai gambar default instance JupyterLab notebook. Pengguna lingkungan IPython yang berbeda dapat menginstal Jupyter AI secara manual.

Di bagian ini, kami memberikan ikhtisar kemampuan AI Jupyter dan mendemonstrasikan cara mengonfigurasi model yang disediakan oleh atau SageMaker JumpStart Amazon Bedrock dari

JupyterLabatau notebook Studio Classic. Untuk informasi lebih mendalam tentang proyek AI Jupyter, lihat dokumentasinya. Atau, Anda dapat merujuk ke posting blog Generative AI di Jupyter untuk ikhtisar dan contoh kemampuan AI Jupyter utama.

Sebelum menggunakan Jupyter AI dan berinteraksi dengan LLM Anda, pastikan Anda memenuhi prasyarat berikut:

- Untuk model yang dihosting olehAWS, Anda harus memiliki ARN dari SageMaker titik akhir Anda atau memiliki akses ke Amazon Bedrock. Untuk penyedia model lainnya, Anda harus memiliki kunci API yang digunakan untuk mengautentikasi dan mengotorisasi permintaan ke model Anda. Jupyter AI mendukung berbagai penyedia model dan model bahasa, lihat daftar model yang didukung untuk tetap diperbarui pada model terbaru yang tersedia. Untuk informasi tentang cara menerapkan model SageMaker JumpStart, lihat Menerapkan Model dalam dokumentasi. SageMaker JumpStart Anda perlu meminta akses ke Amazon Bedrock untuk menggunakannya sebagai penyedia model Anda.
- Pastikan pustaka Jupyter AI hadir di lingkungan Anda. Jika tidak, instal paket yang diperlukan dengan mengikuti instruksi diInstal Jupyter AI.
- Biasakan diri Anda dengan kemampuan Jupyter AI di. Fitur Jupyter AI
- Konfigurasikan model target yang ingin Anda gunakan dengan mengikuti petunjuk diKonfigurasikan penyedia model Anda.

Setelah menyelesaikan langkah-langkah prasyarat, Anda dapat melanjutkan ke. Gunakan Jupyter AI di JupyterLab atau Studio Classic

Topik

- Instal Jupyter AI
- Fitur Jupyter AI
- Konfigurasikan penyedia model Anda
- Gunakan Jupyter AI di JupyterLab atau Studio Classic

Instal Jupyter AI

Untuk pengguna Amazon SageMaker Distribution, sebaiknya pilih gambar SageMaker Distribusi versi 1.2 atau yang lebih baru. Tidak diperlukan instalasi lebih lanjut. Pengguna JupyterLab di Studio dapat memilih versi SageMaker Distribusi Amazon mereka saat membuat ruang.

Untuk pengguna lingkungan IPython lainnya, versi paket Jupyter AI yang direkomendasikan bergantung pada versi yang mereka gunakan. JupyterLab

Distribusi Jupyter AI terdiri dari dua paket.

- `jupyter_ai`: Paket ini menyediakan JupyterLab ekstensi dan antarmuka pengguna obrolan asli (UI). Ini bertindak sebagai asisten percakapan menggunakan model bahasa besar pilihan Anda.
- `jupyter_ai_magics`: Paket ini menyediakan IPython `%%ai` dan perintah `%ai` ajaib yang dengannya Anda dapat memanggil model bahasa besar (LLM) dari sel notebook Anda.

> **Note**
>
> Instalasi `jupyter_ai` juga menginstal`jupyter_ai_magics`. Namun, Anda dapat menginstal `jupyter_ai_magics` secara independen tanpa JupyterLab atau`jupyter_ai`. Perintah ajaib `%%ai` dan `%ai` bekerja di lingkungan kernel IPython apa pun. Jika Anda hanya menginstal`jupyter_ai_magics`, Anda tidak dapat menggunakan UI obrolan.

Untuk pengguna JupyterLab 3, khususnya pengguna Studio Classic, kami sarankan menginstal `jupyter-ai` versi 1.5.x atau versi 1.x yang lebih baru. Namun, kami sangat menyarankan menggunakan Jupyter AI dengan JupyterLab 4. `jupyter-ai`Versi yang kompatibel dengan JupyterLab 3 mungkin tidak mengizinkan pengguna untuk mengatur parameter model tambahan seperti suhu, pengambilan sampel top-k dan top-p, token maks atau panjang maks, atau perjanjian lisensi penerimaan pengguna.

Untuk pengguna JupyterLab 4 lingkungan yang tidak menggunakan SageMaker Distribusi, kami sarankan menginstal `jupyter-ai` versi 2.5.x atau versi 2.x yang lebih baru.

Lihat petunjuk penginstalan di bagian Instalasi dokumentasi Jupyter AI.

Fitur Jupyter AI

Anda dapat mengakses kemampuan AI Jupyter melalui dua metode berbeda: menggunakan UI obrolan atau menggunakan perintah ajaib di dalam notebook.

Dari antarmuka pengguna obrolan asisten AI

Antarmuka obrolan menghubungkan Anda dengan Jupyternaut, agen percakapan yang menggunakan model bahasa pilihan Anda.

Setelah meluncurkan JupyterLab aplikasi yang diinstal dengan Jupyter AI,
Anda dapat mengakses antarmuka obrolan dengan memilih ikon obrolan

di panel navigasi kiri. Pengguna pertama kali diminta untuk mengonfigurasi model mereka. Lihat Konfigurasikan penyedia model Anda di UI obrolan untuk petunjuk konfigurasi.

Menggunakan UI obrolan, Anda dapat:

- Menjawab pertanyaan: Misalnya, Anda dapat meminta Jupyternaut untuk membuat fungsi Python yang menambahkan file CSV ke bucket Amazon S3. Selanjutnya, Anda dapat menyempurnakan jawaban Anda dengan pertanyaan tindak lanjut, seperti menambahkan parameter ke fungsi untuk memilih jalur tempat file ditulis.
- Berinteraksi dengan file di JupyterLab: Anda dapat menyertakan sebagian buku catatan Anda di prompt Anda dengan memilihnya. Kemudian, Anda dapat menggantinya dengan jawaban yang disarankan model atau menyalin jawaban secara manual ke clipboard Anda.
- Hasilkan seluruh buku catatan dari petunjuk: Dengan memulai prompt Anda dengan`/generate`, Anda memicu proses pembuatan notebook di latar belakang tanpa mengganggu penggunaan Jupyternaut. Pesan yang berisi tautan ke file baru ditampilkan setelah proses selesai.
- Belajar dari dan mengajukan pertanyaan tentang file lokal: Dengan menggunakan `/learn` perintah, Anda dapat mengajarkan model penyematan pilihan Anda tentang file lokal dan kemudian mengajukan pertanyaan tentang file-file tersebut menggunakan `/ask` perintah. Jupyter AI menyimpan konten yang disematkan dalam database vektor FAISS lokal, kemudian menggunakan retrieval-augmented generation (RAG) untuk memberikan jawaban berdasarkan apa yang telah dipelajari. Untuk menghapus semua informasi yang dipelajari sebelumnya dari model penyematan Anda, gunakan. `/learn -d`

Untuk daftar lengkap fitur dan petunjuk terperinci tentang penggunaannya, lihat dokumentasi antarmuka obrolan Jupyter AI. Untuk mempelajari cara mengonfigurasi akses ke model di Jupyternaut, lihat. Konfigurasikan penyedia model Anda di UI obrolan

Dari sel notebook

Menggunakan `%%ai` dan perintah `%ai` ajaib, Anda dapat berinteraksi dengan model bahasa pilihan Anda dari sel notebook Anda atau antarmuka baris perintah IPython apa pun. `%%ai`Perintah menerapkan instruksi Anda ke seluruh sel, sedangkan `%ai` menerapkannya ke baris tertentu.

Contoh berikut mengilustrasikan perintah `%%ai` ajaib memanggil model Anthropic Claude untuk mengeluarkan file HTML yang berisi gambar kotak putih dengan batas hitam.

```
%%ai anthropic:claude-v1.2 -f html
Create a square using SVG with a black border and white fill.
```

Untuk mempelajari tentang sintaks dari setiap perintah, gunakan`%ai help`. Untuk membuat daftar penyedia dan model yang didukung oleh ekstensi, jalankan`%ai list`.

Untuk daftar lengkap fitur dan petunjuk terperinci tentang penggunaannya, lihat dokumentasi perintah ajaib Jupyter AI. Secara khusus, Anda dapat menyesuaikan format output model Anda menggunakan `--format` parameter `-f` or, mengizinkan interpolasi variabel dalam prompt, termasuk khusus `In` dan `Out` variabel, dan banyak lagi.

Untuk mempelajari cara mengonfigurasi akses ke model, lihatKonfigurasikan penyedia model Anda di buku catatan.

Konfigurasikan penyedia model Anda

> **Note**
>
> Di bagian ini, kami berasumsi bahwa bahasa dan model penyematan yang Anda rencanakan untuk digunakan sudah diterapkan. Untuk model yang disediakan olehAWS, Anda seharusnya sudah memiliki ARN SageMaker titik akhir Anda atau akses ke Amazon Bedrock. Untuk penyedia model lainnya, Anda harus memiliki kunci API yang digunakan untuk mengautentikasi dan mengotorisasi permintaan ke model Anda.
> Jupyter AI mendukung berbagai penyedia model dan model bahasa, lihat daftar model yang didukung untuk tetap diperbarui pada model terbaru yang tersedia. Untuk informasi tentang cara menerapkan model yang disediakan oleh SageMaker JumpStart, lihat Menerapkan Model dalam dokumentasi. SageMaker JumpStart Anda perlu meminta akses ke Amazon Bedrock untuk menggunakannya sebagai penyedia model Anda.

Konfigurasi Jupyter AI bervariasi tergantung pada apakah Anda menggunakan UI obrolan atau perintah ajaib.

Konfigurasikan penyedia model Anda di UI obrolan

> **Note**
>
> Anda dapat mengonfigurasi beberapa LLM dan menyematkan model mengikuti instruksi yang sama. Namun, Anda harus mengkonfigurasi setidaknya satu model Bahasa.

Untuk mengonfigurasi UI obrolan

1. Di JupyterLab, akses antarmuka obrolan dengan memilih ikon obrolan

   di panel navigasi kiri.
2. Pilih ikon konfigurasi

   di sudut kanan atas panel kiri. Ini membuka panel konfigurasi Jupyter AI.
3. Isi kolom yang terkait dengan penyedia layanan Anda.

   - Untuk model yang disediakan oleh SageMaker JumpStart atau Amazon Bedrock
     - Dalam daftar dropdown model bahasa, pilih model yang digunakan dengan SageMaker JumpStart atau `sagemaker-endpoint bedrock` untuk model yang dikelola oleh Amazon Bedrock.
     - Parameter berbeda berdasarkan apakah model Anda digunakan SageMaker atau Amazon Bedrock.
       - Untuk model yang digunakan dengan SageMaker JumpStart:
         - Masukkan nama titik akhir Anda di nama Endpoint, dan kemudian Wilayah AWS di mana model Anda digunakan dalam nama Region. Untuk mengambil ARN dari SageMaker titik akhir, navigasikan https://console.aws.amazon.com/sagemaker/ ke dan kemudian pilih Inferensi dan Titik Akhir di menu sebelah kiri.
         - Rekatkan JSON skema Permintaan yang disesuaikan dengan model Anda, dan jalur Response yang sesuai untuk mengurai output model.

> **Note**
>
> Anda dapat menemukan format permintaan dan respons dari berbagai model SageMaker JumpStart pondasi di notebook contoh berikut. Setiap notebook diberi nama sesuai model yang ditunjukkannya.

    - Untuk model yang dikelola oleh Amazon Bedrock: Tambahkan AWS profil yang menyimpan AWS kredensional Anda di sistem Anda (opsional), lalu Wilayah AWS di mana model Anda digunakan dalam nama Wilayah.
  - (Opsional) Pilih model penyematan yang dapat Anda akses. Model penyematan digunakan untuk menangkap informasi tambahan dari dokumen lokal, memungkinkan model pembuatan teks untuk menanggapi pertanyaan dalam konteks dokumen tersebut.
  - Pilih Simpan Perubahan dan arahkan ke ikon panah kiri

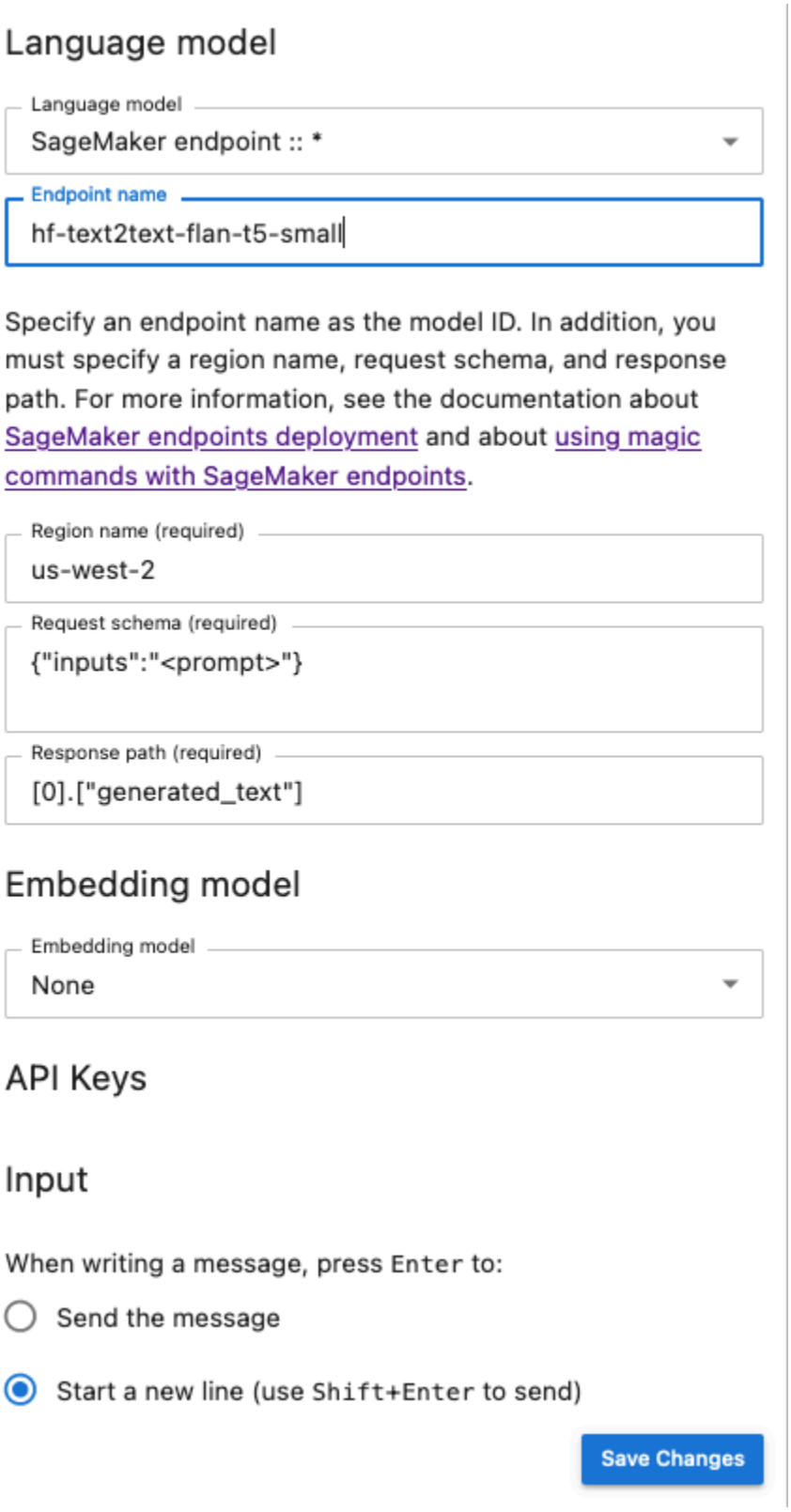
    

    di sudut kiri atas panel kiri. Ini membuka UI obrolan AI Jupyter. Anda dapat mulai berinteraksi dengan model Anda.
- Untuk model yang dihosting oleh penyedia pihak ketiga
  - Dalam daftar dropdown model bahasa, pilih ID penyedia Anda. Anda dapat menemukan detail masing-masing penyedia, termasuk ID mereka, di daftar penyedia model AI Jupyter.
  - (Opsional) Pilih model penyematan yang dapat Anda akses. Model penyematan digunakan untuk menangkap informasi tambahan dari dokumen lokal, memungkinkan model pembuatan teks untuk menanggapi pertanyaan dalam konteks dokumen tersebut.
  - Masukkan kunci API model Anda.
  - Pilih Simpan Perubahan dan arahkan ke ikon panah kiri

    di sudut kiri atas panel kiri. Ini membuka UI obrolan AI Jupyter. Anda dapat mulai berinteraksi dengan model Anda.

Snapshot berikut adalah ilustrasi panel konfigurasi UI obrolan yang disetel untuk memanggil model FLAN-T5-Small yang disediakan oleh dan diterapkan di. SageMaker JumpStart SageMaker


Language model
Language model
SageMaker endpoint :: *
Endpoint name
hf-text2text-flan-t5-small
Specify an endpoint name as the model ID. In addition, you must specify a region name, request schema, and response path. For more information, see the documentation about SageMaker endpoints deployment and about using magic commands with SageMaker endpoints.
Region name (required)
us-west-2
Request schema (required)
{"inputs":"<prompt>"}
Response path (required)
[0].["generated_text"]
Embedding model
Embedding model
None
API Keys
Input
When writing a message, press Enter to:
Send the message
Start a new line (use Shift+Enter to send)
Save Changes

Berikan parameter model tambahan dan parameter khusus ke permintaan Anda

Model Anda mungkin memerlukan parameter tambahan, seperti atribut yang disesuaikan untuk persetujuan perjanjian pengguna atau penyesuaian parameter model lain seperti suhu atau panjang respons. Sebaiknya konfigurasi pengaturan ini sebagai opsi start up JupyterLab aplikasi Anda menggunakan Konfigurasi Siklus Hidup. Untuk informasi tentang cara membuat Konfigurasi Siklus Hidup dan melampirkannya ke domain Anda, atau ke profil pengguna dari SageMaker konsol, lihat Membuat dan mengaitkan konfigurasi siklus hidup. Anda dapat memilih skrip LCC Anda saat membuat ruang untuk JupyterLab aplikasi Anda.

Gunakan skema JSON berikut untuk mengonfigurasi parameter tambahan Anda:

```
{
  "AiExtension": {
    "model_parameters": {
      "<provider_id>:<model_id>": { Dictionary of model parameters which is unpacked
 and passed as-is to the provider.}
    }
  }
}
```

Skrip berikut adalah contoh file konfigurasi JSON yang dapat Anda gunakan saat membuat JupyterLab aplikasi LCC untuk mengatur panjang maksimum model AI21 Labs Jurassic-2 yang digunakan di Amazon Bedrock. Meningkatkan panjang respons yang dihasilkan model dapat mencegah pemotongan sistematis respons model Anda.

```
#!/bin/bash
set -eux

mkdir -p /home/sagemaker-user/.jupyter

json='{"AiExtension": {"model_parameters": {"bedrock:ai21.j2-mid-v1": {"model_kwargs":
 {"maxTokens": 200}}}}}'
# equivalent to %%ai bedrock:ai21.j2-mid-v1 -m {"model_kwargs":{"maxTokens":200}}

# File path
file_path="/home/sagemaker-user/.jupyter/jupyter_jupyter_ai_config.json"

#jupyter --paths
```

```
# Write JSON to file
echo "$json" > "$file_path"

# Confirmation message
echo "JSON written to $file_path"

restart-jupyter-server

# Waiting for 30 seconds to make sure the Jupyter Server is up and running
sleep 30
```

Skrip berikut adalah contoh file konfigurasi JSON untuk membuat JupyterLab aplikasi LCC yang digunakan untuk mengatur parameter model tambahan untuk model Anthropic Claude yang digunakan di Amazon Bedrock.

```
#!/bin/bash
set -eux

mkdir -p /home/sagemaker-user/.jupyter

json='{"AiExtension": {"model_parameters": {"bedrock:anthropic.claude-v2":
{"model_kwargs":{"temperature":0.1,"top_p":0.5,"top_k":25
0,"max_tokens_to_sample":2}}}}}'
# equivalent to %%ai bedrock:anthropic.claude-v2 -m {"model_kwargs":
{"temperature":0.1,"top_p":0.5,"top_k":250,"max_tokens_to_sample":2000}}

# File path
file_path="/home/sagemaker-user/.jupyter/jupyter_jupyter_ai_config.json"

#jupyter --paths

# Write JSON to file
echo "$json" > "$file_path"

# Confirmation message
echo "JSON written to $file_path"

restart-jupyter-server

# Waiting for 30 seconds to make sure the Jupyter Server is up and running
sleep 30
```

Setelah Anda melampirkan LCC Anda ke domain Anda, atau profil pengguna, tambahkan LCC Anda ke ruang Anda saat meluncurkan aplikasi Anda JupyterLab . Untuk memastikan bahwa file konfigurasi Anda diperbarui oleh LCC, jalankan `more ~/.jupyter/jupyter_jupyter_ai_config.json` di terminal. Isi file harus sesuai dengan konten file JSON yang diteruskan ke LCC.

Konfigurasikan penyedia model Anda di buku catatan

Untuk memanggil model melalui Jupyter AI di dalam JupyterLab atau notebook Studio Classic menggunakan perintah dan ajaib **`%%ai%ai`**

1. Instal pustaka klien khusus untuk penyedia model Anda di lingkungan notebook Anda. Misalnya, saat menggunakan model OpenAI, Anda perlu menginstal pustaka `openai` klien. Anda dapat menemukan daftar pustaka klien yang diperlukan per penyedia di kolom paket Python dari daftar penyedia Model AI Jupyter.

   > **Note**
   >
   > Untuk model yang di-host olehAWS, `boto3` sudah diinstal pada gambar SageMaker Distribusi yang digunakan oleh JupyterLab, atau gambar Ilmu Data apa pun yang digunakan dengan Studio Classic.

2. - Untuk model yang diselenggarakan oleh AWS

     Pastikan peran eksekusi Anda memiliki izin untuk memanggil SageMaker titik akhir Anda untuk model yang disediakan oleh SageMaker JumpStart atau bahwa Anda memiliki akses ke Amazon Bedrock.

   - Untuk model yang dihosting oleh penyedia pihak ketiga

     Ekspor kunci API penyedia Anda di lingkungan notebook menggunakan variabel lingkungan. Anda dapat menggunakan perintah ajaib berikut. Ganti perintah `provider_API_key` dalam dengan variabel lingkungan yang ditemukan di kolom variabel Lingkungan dari daftar penyedia Model AI Jupyter untuk penyedia Anda.

     ```
     %env provider_API_key=your_API_key
     ```

Gunakan Jupyter AI di JupyterLab atau Studio Classic

Menggunakan model bahasa dari UI obrolan

Tulis pesan Anda di kotak teks UI obrolan untuk mulai berinteraksi dengan model Anda. Untuk menghapus riwayat pesan, gunakan `/clear` perintah.

> **Note**
>
> Menghapus riwayat pesan tidak menghapus konteks obrolan dengan penyedia model.

Gunakan model bahasa dari sel notebook

Sebelum menggunakan `%ai` perintah `%%ai` dan untuk memanggil model bahasa, muat ekstensi IPython dengan menjalankan perintah berikut di sel notebook JupyterLab atau Studio Classic.

```
%load_ext jupyter_ai_magics
```

- Untuk model yang diselenggarakan olehAWS:
  - Untuk memanggil model yang digunakan SageMaker, teruskan string `sagemaker-endpoint:`*`endpoint-name`* ke perintah `%%ai` ajaib dengan parameter yang diperlukan di bawah ini, lalu tambahkan prompt Anda di baris berikut.

    Tabel berikut mencantumkan parameter wajib dan opsional saat menjalankan model yang dihosting oleh SageMaker atau Amazon Bedrock.

Nama Parameter	Parameter	Versi Pendek	Deskripsi
Permintaan skema	`--request-schema`	`-q`	Wajib: Objek JSON yang diharapkan titik akhir, dengan prompt diganti menjadi nilai apa pun yang cocok dengan string literal. `<prompt>`

Nama Parameter	Parameter	Versi Pendek	Deskripsi
Nama wilayah	`--region-name`	`-n`	Wajib: Wilayah AWS Tempat model dikerahkan.
Jalur respons	`--response-path`	`-p`	Wajib: String JsonPath yang digunakan untuk mengekstrak output model bahasa dari respons JSON dari titik akhir.

Nama Parameter	Parameter	Versi Pendek	Deskripsi
Parameter model tambahan	`--model-parameters`	`-m`	Opsional: Nilai JSON yang menentukan parameter tambahan yang akan diteruskan ke model. Nilai yang diterima diuraikan ke dalam kamus, dibongkar, dan langsung diteruskan ke kelas penyedia. Ini berguna ketika titik akhir atau model memerlukan parameter khusus. Misalnya, dalam model Llama 2 saat menerima Perjanjian Lisensi Pengguna Akhir (EULA) diperlukan, Anda dapat meneruskan penerimaan EULA ke titik akhir menggunakan. `-m {"endpoint_kwargs":{"CustomAttributes":"accept_eula=true"}}` Atau, Anda dapat menggunakan `-m` parameter

Nama Parameter	Parameter	Versi Pendek	Deskripsi
			untuk meneruskan parameter model tambahan, seperti menyetel jumlah token maksimum untuk respons yang dihasilkan model. Misalnya, saat bekerja dengan model Jurassic AI21 Labs:. `-m {"model_kwargs":{"maxTokens":256}}`
Format keluaran	`--format`	`-f`	Opsional: Tampilan IPython digunakan untuk merender output. Ini bisa berupa salah satu dari nilai berikut`[code|html|image|json|markdown|math|md|text]` , asalkan model yang dipanggil mendukung format yang ditentukan.

Perintah berikut memanggil model LLAMA2-7b yang dihosting oleh. SageMaker

```
%%ai sagemaker-endpoint:jumpstart-dft-meta-textgeneration-llama-2-7b -q
 {"inputs":"<prompt>","parameters":
{"max_new_tokens":64,"top_p":0.9,"temperature":0.6,"return_full_text":false}}
```

```
 -n us-east-2 -p [0].generation -m {"endpoint_kwargs":
{"CustomAttributes":"accept_eula=true"}} -f text
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Contoh berikut memanggil model flan-T5-Small yang dihosting oleh. SageMaker

```
%%ai sagemaker-endpoint:hf-text2text-flan-t5-small --request-
schema={"inputs":"<prompt>","parameters":{"num_return_sequences":4}} --region-
name=us-west-2 --response-path=[0]["generated_text"] -f text
What is the atomic number of Hydrogen?
```

- Untuk memanggil model yang diterapkan di Amazon Bedrock, teruskan string `bedrock:`*`model-name`* ke perintah `%%ai` ajaib dengan parameter opsional apa pun yang ditentukan dalam daftar parameter untuk menjalankan model yang dihosting oleh atau SageMaker JumpStart Amazon Bedrock, lalu tambahkan prompt Anda di baris berikut.

  Contoh berikut memanggil model AI21 Labs Jurassic-2 yang diselenggarakan oleh Amazon Bedrock.

  ```
  %%ai bedrock:ai21.j2-mid-v1 -m {"model_kwargs":{"maxTokens":256}} -f code
  Write a function in python implementing a bubbble sort.
  ```

- Untuk model yang dihosting oleh penyedia pihak ketiga

  Untuk memanggil model yang dihosting oleh penyedia pihak ketiga, teruskan string *`provider-id`*`:`*`model-name`* ke perintah `%%ai` ajaib dengan opsional `Output format`, lalu tambahkan prompt Anda di baris berikut. Anda dapat menemukan detail masing-masing penyedia, termasuk ID mereka, di daftar penyedia model AI Jupyter.

  Perintah berikut meminta model Anthropic Claude untuk mengeluarkan file HTML yang berisi gambar kotak putih dengan batas hitam.

  ```
  %%ai anthropic:claude-v1.2 -f html
  Create a square using SVG with a black border and white fill.
  ```

Label data dengan human-in-the-loop

Untuk melatih model pembelajaran mesin, Anda memerlukan kumpulan data berlabel besar, berkualitas tinggi, dan berlabel. Anda dapat memberi label pada data Anda menggunakan Amazon SageMaker Ground Truth. Pilih dari salah satu jenis tugas bawaan Ground Truth atau buat alur kerja pelabelan khusus Anda sendiri. Untuk meningkatkan keakuratan label data Anda dan mengurangi total biaya pelabelan data Anda, gunakan fitur pelabelan data yang disempurnakan oleh Ground Truth seperti pelabelan data otomatis dan konsolidasi anotasi.

Topik

Gunakan Amazon SageMaker Ground Truth untuk Label Data

Untuk melatih model pembelajaran mesin, Anda memerlukan kumpulan data berlabel besar, berkualitas tinggi, dan berlabel. Ground Truth membantu Anda membangun dataset pelatihan berkualitas tinggi untuk model machine learning Anda. Dengan Ground Truth, Anda dapat menggunakan pekerja dari Amazon Mechanical Turk, perusahaan vendor yang Anda pilih, atau tenaga kerja pribadi internal bersama dengan pembelajaran mesin untuk memungkinkan Anda membuat kumpulan data berlabel. Anda dapat menggunakan output set data berlabel dari Ground Truth untuk melatih model Anda sendiri. Anda juga dapat menggunakan output sebagai kumpulan data pelatihan untuk Amazon SageMaker model.

Bergantung pada aplikasi ML-mu, kamu dapat memilih salah satu tipe tugas bawaan Ground Truth agar pekerja menghasilkan jenis label tertentu untuk data kamu. Anda juga dapat membuat alur kerja pelabelan khusus untuk menyediakan UI dan alat Anda sendiri kepada pekerja yang memberi label pada data Anda. Untuk mempelajari selengkapnya tentang Ground Truth yang dibangun pada jenis tugas, lihatJenis Tugas Bawaan. Untuk mempelajari cara membuat alur kerja pelabelan kustom, lihatMembuat Alur Kerja Pelabelan Kustom.

Untuk mengotomatiskan pelabelan set data pelatihan Anda, Anda dapat menggunakan secara opsionalpelabelan data otomatis, Proses Ground Truth yang menggunakan pembelajaran mesin untuk memutuskan data mana yang perlu diberi label oleh manusia. Pelabelan data otomatis dapat mengurangi waktu pelabelan dan upaya manual yang diperlukan. Untuk informasi selengkapnya, lihat Pelabelan Data. Untuk membuat alur kerja pelabelan kustom, lihatMembuat Alur Kerja Pelabelan Kustom.

Gunakan alat pra-bangun atau kustom untuk menetapkan tugas pelabelan untuk set data pelatihan Anda. SEBUAHpelabelan template UIadalah halaman web yang Ground Truth gunakan untuk menyajikan tugas dan instruksi kepada pekerja Anda. Klaster SageMaker konsol menyediakan built-in template untuk pelabelan data. Anda dapat menggunakan template ini untuk memulai, atau Anda dapat membangun tugas dan instruksi Anda sendiri dengan menggunakan komponen HTML 2.0 kami. Untuk informasi selengkapnya, lihat Membuat Alur Kerja Pelabelan Kustom.

Gunakan tenaga kerja pilihan Anda untuk memberi label pada kumpulan data Anda. Anda dapat memilih tenaga kerja Anda dari:

- Tenaga kerja Amazon Mechanical Turk lebih dari 500.000 kontraktor independen di seluruh dunia.
- Tenaga kerja pribadi yang Anda buat dari karyawan atau kontraktor Anda untuk menangani data dalam organisasi Anda.
- Perusahaan vendor yang dapat Anda temukan diAWS Marketplaceyang mengkhususkan diri dalam layanan pelabelan data.

Untuk informasi selengkapnya, lihat Membuat dan Mengelola Tenaga Kerja.

Anda menyimpan kumpulan data Anda di bucket Amazon S3. Ember berisi tiga hal: Data yang akan diberi label, file manifes masukan yang digunakan Ground Truth untuk membaca file data, dan file manifes keluaran. File output berisi hasil pekerjaan pelabelan. Untuk informasi selengkapnya, lihat Gunakan.

Acara dari pekerjaan pelabelan Anda muncul di Amazon CloudWatch di bawah`/aws/sagemaker/LabelingJobs`grup. CloudWatch menggunakan nama pekerjaan pelabelan sebagai nama untuk aliran log.

Apakah Anda Pengguna Ground Truth?

Jika Anda baru pertama kali menggunakan Ground Truth, kami menyarankan agar Anda melakukan hal berikut:

1. BacaMulai—Bagian ini memandu Anda melakukan penyiapan tugas pelabelan Ground Truth pertama Anda.
2. Jelajahi topik lainnya—Tergantung pada kebutuhan Anda, lakukan hal berikut:
   - Jelajahi tipe tugas bawaan- Gunakan tipe tugas bawaan untuk merampingkan proses pembuatan pekerjaan pelabelan. LihatJenis Tugas Bawaanuntuk mempelajari lebih lanjut tentang Ground Truth built-in jenis tugas.
   - Kelola tenaga kerja pelabelan Anda—Buat tim kerja baru dan kelola tenaga kerja Anda yang ada. Untuk informasi selengkapnya, lihat Membuat dan Mengelola Tenaga Kerja.
   - Pelajari tugas streaming- Buat pekerjaan pelabelan streaming dan kirim objek set data baru ke pekerja secara real time menggunakan pekerjaan pelabelan yang terus berjalan. Pekerja terus menerima objek data baru untuk diberi label selama pekerjaan pelabelan aktif dan objek baru dikirim ke sana. Untuk mempelajari selengkapnya, lihat Pekerjaan Pelabelan Streaming Ground Truth.
3. Lihat`Reference`—Bagian ini menjelaskan operasi untuk mengotomatisasi operasi Ground Truth.

Mulai

Video ini menunjukkan cara menyiapkan dan menggunakan Amazon SageMaker Ground Truth. (Panjang: 9:37)

Untuk memulai menggunakan Amazon SageMaker Ground Truth, ikuti petunjuk di bagian berikut. Bagian di sini menjelaskan cara menggunakan konsol untuk membuat pekerjaan pelabelan, menetapkan tenaga kerja publik atau swasta, dan mengirim pekerjaan pelabelan ke tenaga kerja Anda. Anda juga dapat mempelajari cara memantau kemajuan pekerjaan pelabelan.

Jika Anda ingin membuat alur kerja pelabelan kustom, lihatMembuat Alur Kerja Pelabelan Kustomuntuk instruksi.

Sebelum membuat pekerjaan pelabelan, Anda harus mengunggah set data Anda ke bucket Amazon S3. Untuk informasi selengkapnya, lihat Gunakan.

Topik

- Langkah 1: Sebelum Anda Memulai
- Langkah 2: Membuat Job pelabelan
- Langkah 3: Pilih Pekerja
- Langkah 4: Konfigurasikan Alat Kotak Bounding

- Langkah 5: Memiilih Job pelabelan Anda

Langkah 1: Sebelum Anda Memulai

Sebelum Anda mulai menggunakan SageMaker konsol untuk membuat pekerjaan pelabelan, Anda harus mengatur dataset untuk digunakan. Lakukan ini:

1. Simpan dua gambar di URL HTTP yang tersedia untuk umum. Gambar digunakan saat membuat instruksi untuk menyelesaikan tugas pelabelan. Gambar harus memiliki rasio aspek sekitar 2:1. Untuk latihan ini, isi gambar tidak penting.
2. Buat bucket Amazon S3 untuk menahan file input dan output. Bucket harus berada di Wilayah yang sama tempat Anda menjalankan Ground Truth. Catat nama bucket karena Anda menggunakannya selama langkah 2.

   Ground Truth mengharuskan semua bucket S3 yang berisi data gambar input pekerjaan pelabelan yang memiliki kebijakan CORS terlampir. Untuk lebih mempelajari tentang perubahan ini, lihatPersyaratan Izin.
3. Anda dapat membuat peran IAM atau membiarkan SageMaker membuat peran dengan AmazonSageMakerFullAccessKebijakan IAM. LihatMembuat Peran IAMdan menetapkan kebijakan izin berikut untuk pengguna yang membuat pekerjaan pelabelan:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "sagemakergroundtruth",
            "Effect": "Allow",
            "Action": [
                "cognito-idp:CreateGroup",
                "cognito-idp:CreateUserPool",
                "cognito-idp:CreateUserPoolDomain",
                "cognito-idp:AdminCreateUser",
                "cognito-idp:CreateUserPoolClient",
                "cognito-idp:AdminAddUserToGroup",
                "cognito-idp:DescribeUserPoolClient",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:UpdateUserPool"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Selanjutnya

[Langkah 2: Membuat Job pelabelan]

Langkah 2: Membuat Job pelabelan

Pada langkah ini Anda menggunakan konsol untuk membuat pekerjaan pelabelan. Anda memberi tahu Amazon SageMaker Ground Truth bucket Amazon S3 tempat file manifes disimpan dan mengonfigurasi parameter untuk pekerjaan. Untuk informasi selengkapnya tentang menyimpan data dalam bucket Amazon S3, lihatGunakan.

Untuk membuat tugas pelabelan

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Dari navigasi kiri, pilihTugas pelabelan.
3. PilihBuat tugas pelabelanuntuk memulai proses pembuatan tugas.
4. DiGambaran umum Jobbagian, berikan informasi berikut:
   - Nama tugas— Berikan pekerjaan pelabelan nama yang menggambarkan pekerjaan. Nama ini ditampilkan dalam daftar pekerjaan Anda. Nama harus unik di akun Anda dalamAWSWilayah.
   - Nama atribut label- Biarkan ini tidak dicentang karena nilai default adalah pilihan terbaik untuk pekerjaan pengantar ini.
   - Penyiapan data input— PilihPenyiapan data otomatis. Opsi ini memungkinkan Anda untuk secara otomatis terhubung ke data input Anda di S3.
   - Lokasi S3 untuk set data input- Masukkan lokasi S3 tempat Anda menambahkan gambar pada langkah 1.
   - Lokasi S3 untuk set data keluaran- Lokasi di mana data output Anda ditulis dalam S3.
   - Jenis data- Gunakan menu drop-down untuk memilihCitra. Ground Truth akan menggunakan semua gambar yang ditemukan di lokasi S3 untuk kumpulan data input sebagai masukan untuk pekerjaan pelabelan Anda.
   - Peran IAM- Buat atau pilih peran IAM dengan AmazonSageMakerFullAccess Kebijakan IAM terlampir.
5. DiJenis tugasbagian, untukKategori tugasbidang, pilihCitra.

6. DiPemilihan tugaspilihKotak pembatas.
7. PilihSelanjutnyauntuk beralih ke mengkonfigurasi pekerjaan pelabelan Anda.

Selanjutnya

Langkah 3: Pilih Pekerja

Langkah 3: Pilih Pekerja

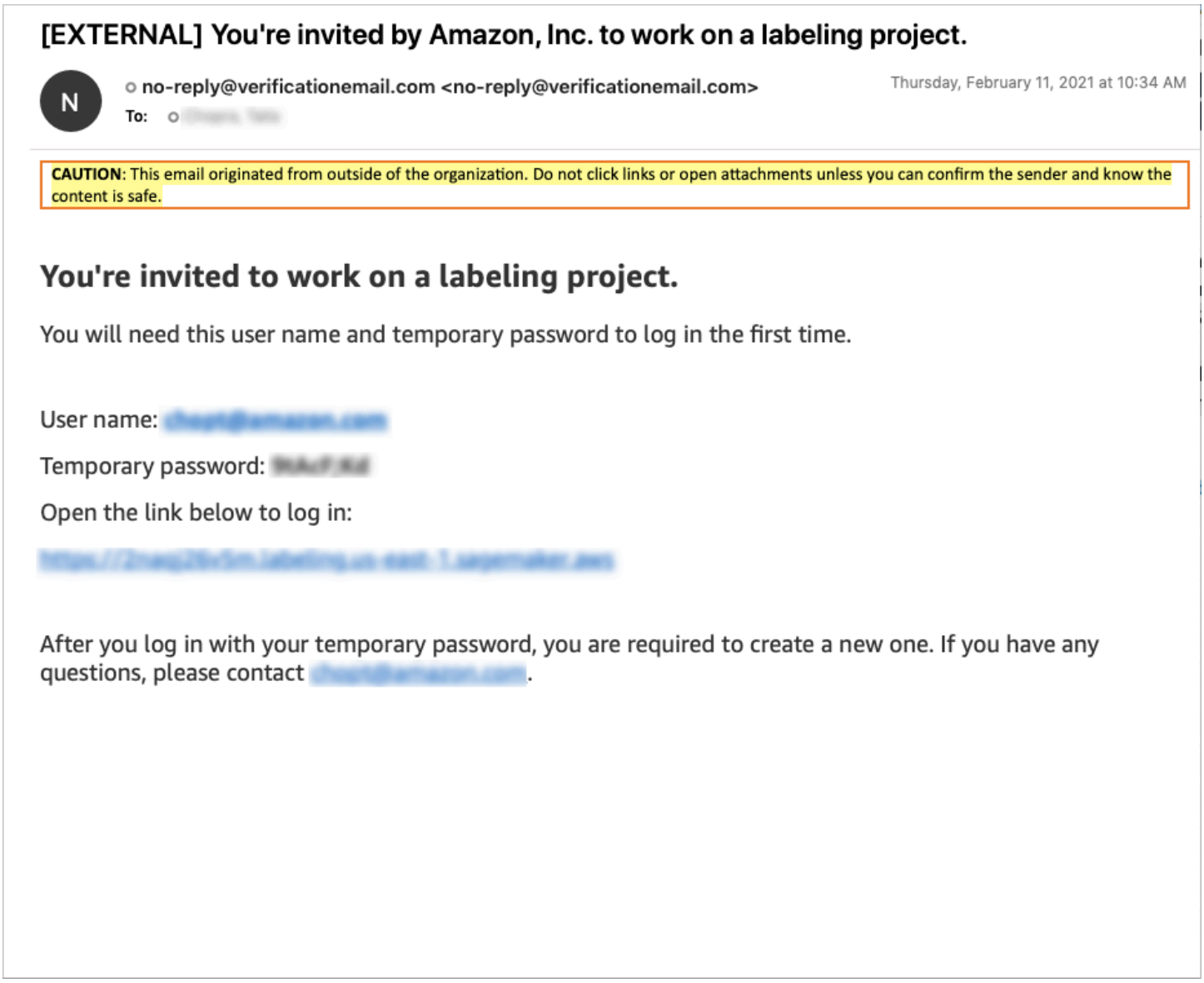
Pada langkah ini Anda memilih tenaga kerja untuk memberi label pada kumpulan data Anda. Dianjurkan agar Anda membuat tenaga kerja pribadi untuk menguji Amazon SageMaker Ground Truth. Gunakan alamat email untuk mengundang anggota tenaga kerja Anda. Jika Anda membuat tenaga kerja pribadi dalam langkah ini, Anda tidak akan dapat mengimpor kumpulan pengguna Amazon Cognito nanti. Jika Anda ingin membuat tenaga kerja pribadi menggunakan pangkalan pengguna Amazon Cognito, lihatMengelola Tenaga Kerja Pribadi (Amazon Cognito)dan menggunakan tenaga kerja Mechanical Turk sebagai gantinya dalam tutorial ini.

> **Tip**
>
> Untuk mempelajari tentang opsi tenaga kerja lain yang dapat Anda gunakan dengan Ground Truth, lihatMembuat dan Mengelola Tenaga Kerja.

Untuk membuat tenaga kerja privat:

1. DiPekerjabagian, pilihPrivat.
2. Jika ini pertama kalinya Anda menggunakan tenaga kerja pribadi, diAlamat emailbidang, masukkan hingga 100 alamat email. Alamat harus dipisahkan dengan koma. Anda harus menyertakan alamat email Anda sendiri sehingga Anda adalah bagian dari tenaga kerja dan dapat melihat tugas pelabelan objek data.
3. DiNama Organisasibidang, masukkan nama organisasi Anda. Informasi ini digunakan untuk menyesuaikan email yang dikirim untuk mengundang seseorang ke tenaga kerja pribadi Anda. Anda dapat mengubah nama organisasi setelah kolam pengguna dibuat melalui konsol.
4. DiEmail kontakbidang masukkan alamat email yang anggota tenaga kerja gunakan untuk melaporkan masalah dengan tugas.

Jika Anda menambahkan diri Anda ke tenaga kerja pribadi, Anda akan menerima email yang terlihat mirip seperti berikut ini. Amazon, Inc.digantikan oleh organisasi yang Anda masukkan pada langkah 3 dari prosedur sebelumnya. Pilih tautan dalam email untuk masuk menggunakan kata sandi sementara yang disediakan. Jika diminta, ubah kata sandi Anda. Ketika Anda berhasil masuk, Anda melihat portal pekerja tempat tugas pelabelan Anda muncul.



> **Tip**
>
> Anda dapat menemukan tautan ke portal pekerja tenaga kerja pribadi Anda diMelabelan tenaga kerjabagian dari area Ground Truth dari SageMakerkonsol. Untuk melihat tautannya, pilihPrivattab. Tautan berada di bawahURL masuk di portalheaderRingkasan tenaga kerja.

Jika Anda memilih untuk menggunakan tenaga kerja Amazon Mechanical Turk untuk memberi label pada set data, Anda dikenai biaya untuk tugas pelabelan yang diselesaikan pada set data.

Untuk menggunakan tenaga kerja Amazon Mechanical Turk:

1. DiPekerjabagian, pilihPublik.
2. setHarga per tugas.
3. Jika berlaku, pilihDataset tidak berisi konten dewasauntuk mengakui bahwa kumpulan data sampel tidak memiliki konten dewasa. Informasi ini memungkinkan Amazon SageMaker Ground Truth untuk memperingatkan pekerja eksternal di Mechanical Turk bahwa mereka mungkin menemukan konten yang berpotensi menyinggung dalam kumpulan data Anda.
4. Pilih kotak centang di samping pernyataan berikut untuk mengetahui bahwa kumpulan data sampel tidak berisi informasi identitas pribadi (PII). Ini adalah persyaratan untuk menggunakan Mechanical Turk dengan Ground Truth. Jika data masukan Anda memang mengandung PII, gunakan tenaga kerja pribadi untuk tutorial ini.

   Anda memahami dan menyetujui bahwa tenaga kerja Amazon Mechanical Turk terdiri dari kontraktor independen yang berlokasi di seluruh dunia dan bahwa Anda tidak boleh membagikan informasi rahasia, informasi pribadi, atau informasi kesehatan yang dilindungi dengan tenaga kerja ini.

Selanjutnya

Langkah 4: Konfigurasikan Alat Kotak Bounding

Langkah 4: Konfigurasikan Alat Kotak Bounding

Akhirnya Anda mengkonfigurasi alat kotak pembatas untuk memberikan instruksi kepada pekerja Anda. Anda dapat mengonfigurasi judul tugas yang menjelaskan tugas dan memberikan instruksi tingkat tinggi untuk pekerja. Anda dapat memberikan instruksi cepat dan instruksi lengkap. Instruksi cepat ditampilkan di sebelah gambar yang akan diberi label. Instruksi lengkap berisi instruksi terperinci untuk menyelesaikan tugas. Dalam contoh ini, Anda hanya memberikan petunjuk cepat. Anda dapat melihat contoh instruksi lengkap dengan memilihInstruksi lengkapdi bagian bawah.

Untuk mengkonfigurasi alat kotak pembatas

1. DiDeskripsi tugasjenis bidang dalam instruksi singkat untuk tugas. Misalnya:

   ```
   Draw a box around any objects in the image.
   ```

Ganti*objek*dengan nama objek yang muncul di gambar Anda.

2. DiLabelbidang, ketik nama kategori untuk objek yang pekerja harus menggambar kotak pembatas di sekitar. Misalnya, jika Anda meminta pekerja untuk menggambar kotak di sekitar pemain sepak bola, Anda dapat menggunakan “Pemain Sepak Bola” di bidang ini.
3. KlasterInstruksi singkatbagian memungkinkan Anda untuk membuat petunjuk yang ditampilkan pada halaman dengan gambar yang pekerja Anda label. Kami menyarankan agar Anda menyertakan contoh kotak pembatas yang ditarik dengan benar dan contoh kotak yang salah digambar. Untuk membuat instruksi Anda sendiri, gunakan langkah-langkah ini:

   a. Pilih teks antaraCONTOH YANG BAIKdan placeholder gambar. Menggantinya dengan teks berikut:

      ```
      Draw the box around the object with a small border.
      ```

   b. Pilih placeholder gambar pertama dan hapus.

   c. Pilih tombol gambar dan kemudian masukkan URL HTTPS dari salah satu gambar yang Anda buat di langkah 1. Dimungkinkan juga untuk menyematkan gambar secara langsung di bagian instruksi singkat, namun bagian ini memiliki kuota 100 kilobyte (termasuk teks). Jika gambar dan teks Anda melebihi 100 kilobyte, Anda akan menerima kesalahan.

   d. Pilih teks antaraCONTOH BURUKdan placeholder gambar. Menggantinya dengan teks berikut:

      ```
      Don't make the bounding box too large or cut into the object.
      ```

   e. Pilih placeholder gambar kedua dan hapus.

   f. Pilih tombol gambar dan kemudian masukkan URL HTTPS dari gambar lain yang Anda buat pada langkah 1.
4. PilihPratinjauuntuk melihat pratinjau UI pekerja. Pratinjau terbuka di tab baru, dan jadi jika browser Anda memblokir pop up, Anda mungkin perlu mengaktifkan tab secara manual untuk membuka. Saat Anda menambahkan satu atau lebih anotasi ke pratinjau, lalu pilihKIRIMKANAnda dapat melihat pratinjau data keluaran yang akan dibuat anotasi Anda.
5. Setelah Anda mengonfigurasi dan memverifikasi instruksi Anda, pilihBuatuntuk membuat pekerjaan pelabelan.

Jika Anda menggunakan tenaga kerja pribadi, Anda dapat menavigasi ke portal pekerja yang Anda masukiLangkah 3: Pilih Pekerjadari tutorial ini untuk melihat tugas pelabelan Anda. Tugas mungkin memerlukan waktu beberapa menit untuk muncul.

Selanjutnya

Langkah 5: Memiilih Job pelabelan Anda

Langkah 5: Memiilih Job pelabelan Anda

Setelah Anda membuat pekerjaan pelabelan Anda, Anda melihat daftar semua pekerjaan yang telah Anda buat. Anda dapat menggunakan daftar ini untuk memantau status pekerjaan pelabelan Anda. Daftar ini memiliki bidang berikut:

- Nama- Nama yang Anda tetapkan pada pekerjaan saat Anda membuatnya.
- Status— Status penyelesaian tugas. Status dapat berupa salah satu dari Selesai, Gagal, Sedang berlangsung, atau Berhenti.
- Objek/total berlabel- Menunjukkan jumlah total objek dalam pekerjaan pelabelan dan berapa banyak dari mereka telah diberi label.
- Waktu pembuatan— Tanggal dan waktu Anda menciptakan pekerjaan.

Anda juga dapat mengkloning, rantai, atau menghentikan pekerjaan. Pilih pekerjaan dan kemudian pilih salah satu dari berikut ini dariTindakanMenu:

- clone- Membuat pekerjaan pelabelan baru dengan konfigurasi yang disalin dari pekerjaan yang dipilih. Anda dapat mengkloning pekerjaan ketika Anda ingin mengubah pekerjaan dan menjalankannya lagi. Misalnya, Anda dapat mengkloning pekerjaan yang dikirim ke tenaga kerja pribadi sehingga Anda dapat mengirimkannya ke tenaga kerja Amazon Mechanical Turk. Atau Anda dapat mengkloning pekerjaan untuk menjalankan ulang terhadap dataset baru yang disimpan di lokasi yang sama dengan pekerjaan asli.
- Rantai- Menciptakan pekerjaan pelabelan baru yang dapat membangun data dan model (jika ada) dari pekerjaan yang berhenti, gagal, atau selesai. Untuk informasi selengkapnya tentang kasus penggunaan dan cara menggunakannya, lihatTugas Pelabelan Rantai.
- Berhenti— Menghentikan tugas berjalan. Anda tidak dapat memulai ulang tugas yang berhenti. Anda dapat mengkloning pekerjaan untuk memulai kembali atau rantai pekerjaan untuk melanjutkan dari mana ia tinggalkan. Label untuk objek yang sudah diberi label ditulis ke lokasi file keluaran. Untuk informasi selengkapnya, lihat Data Output.

Gambar Label

Gunakan Ground Truth untuk memberi label pada gambar. Pilih salah satu dari jenis tugas bawaan berikut untuk mempelajari lebih lanjut tentang jenis tugas tersebut. Setiap halaman berisi petunjuk untuk membantu Anda membuat pekerjaan pelabelan menggunakan jenis tugas tersebut.

> **Tip**
>
> Untuk mempelajari lebih lanjut tentang jenis file yang didukung dan kuota data input, lihatInput Data.

Topik

Kotak pembatas

Gambar yang digunakan untuk melatih model pembelajaran mesin seringkali mengandung lebih dari satu objek. Untuk mengklasifikasikan dan melokalisasi satu atau beberapa objek dalam gambar, gunakan jenis tugas tugas pelabelan kotak pembatas Amazon SageMaker Ground Truth. Dalam konteks ini, lokalisasi berarti pixel-lokasi kotak pembatas.

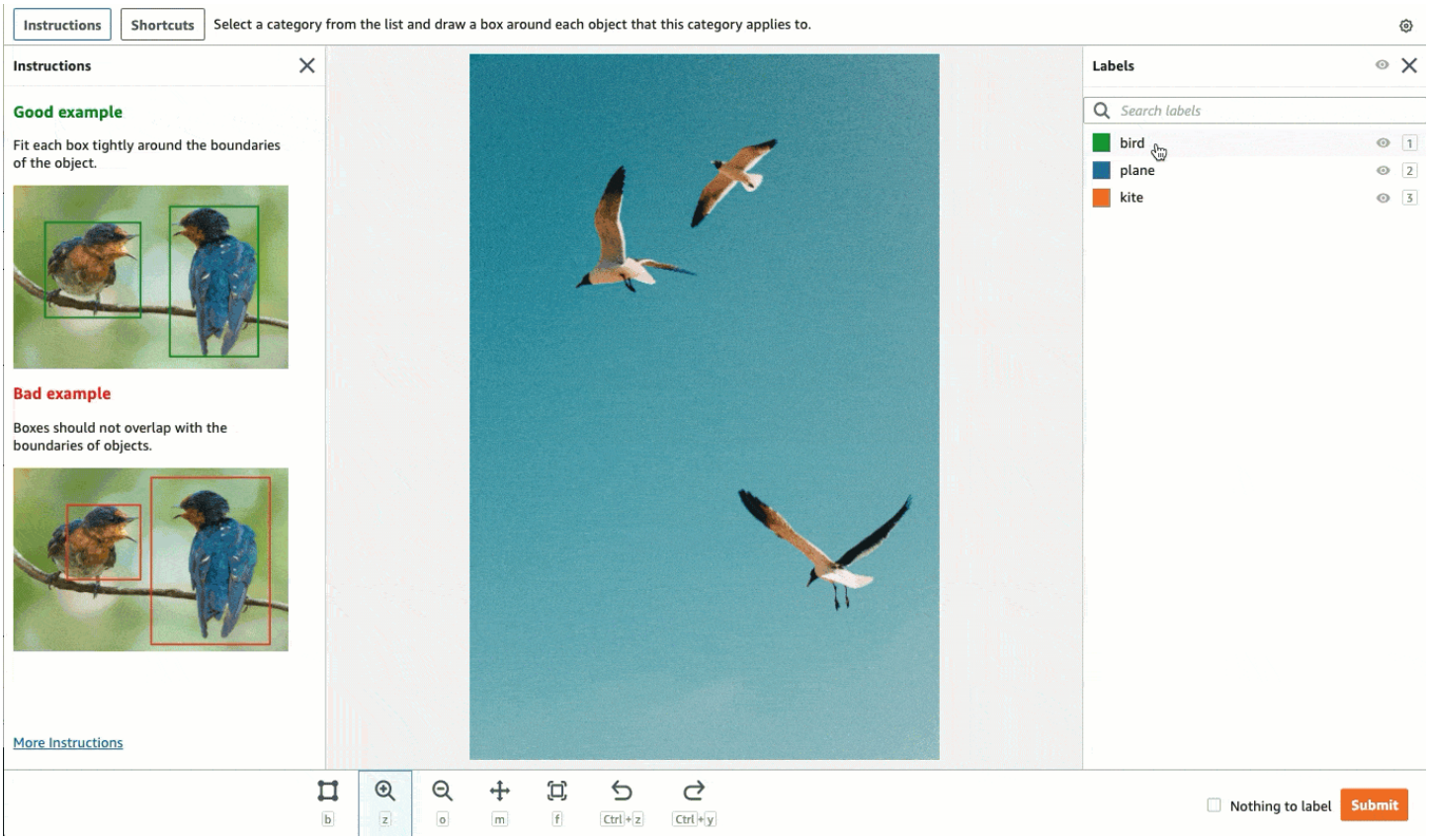
Anda membuat pekerjaan pelabelan kotak pembatas menggunakan bagian Ground Truth dari SageMaker konsol Amazon atau `CreateLabelingJob`operasi.

> **Important**
>
> Untuk jenis tugas ini, jika Anda membuat file manifes sendiri, gunakan`"source-ref"` untuk mengidentifikasi lokasi setiap file gambar di Amazon S3 yang ingin diberi label. Untuk informasi selengkapnya, lihat Input Data.

Membuat Job Pelabelan Kotak Bounding (Konsol)

Anda dapat mengikuti petunjukMembuat Job Pelabelan (Konsol) untuk mempelajari cara membuat pekerjaan pelabelan kotak pembatas di SageMaker konsol. Di Langkah 10, pilih Gambar dari menu drop down kategori Tugas, dan pilih Bounding kotak sebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan dengan konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan hingga 50 label yang dapat dipilih pekerja.



Buat Job Pelabelan Kotak Bounding (API)

Untuk membuat pekerjaan pelabelan kotak pembatas, gunakan operasi SageMaker API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWS SDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjau bagian Lihat Juga **`CreateLabelingJob`**.

Ikuti petunjukBuat Job Pelabelan (API) dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Pra-anotasi fungsi Lambda untuk jenis tugas ini diakhiri dengan`PRE-BoundingBox`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihat PreHumanTaskLambdaArn.
- Fungsi Lambda anotasi-konsolidasi untuk jenis tugas ini diakhiri dengan`ACS-BoundingBox`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihat AnnotationConsolidationLambdaArn.

Berikut ini adalah contoh permintaanAWS Python SDK (Boto3) untuk membuat pekerjaan pelabelan dalam Wilayah US East (N. Virginia). Semua parameter berwarna merah harus diganti dengan spesifikasi dan sumber daya Anda.

```
response = client.create_labeling_job(
    LabelingJobName='example-bounding-box-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/worker-task-template.html'
        },
```

```
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
BoundingBox',
        'TaskKeywords': [
            'Bounding Box',
        ],
        'TaskTitle': 'Bounding Box task',
        'TaskDescription': 'Draw bounding boxes around objects in an image',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-BoundingBox'
          }
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Menyediakan Template untuk Pekerjaan Pelabelan Kotak Bounding

Jika Anda membuat pekerjaan pelabelan menggunakan API, Anda harus menyediakan templat tugas pekerja`UiTemplateS3Uri`. Salin dan modifikasi template berikut. Hanya memodifikasi `short-instructions`, `full-instructions`, dan`header`. Unggah template ini ke S3, dan berikan URI S3 untuk file ini`UiTemplateS3Uri`.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-bounding-box
    name="boundingBox"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="please draw box"
    labels="{{ task.input.labels | to_json | escape }}"
  >

    <full-instructions header="Bounding box instructions">
      <ol><li><strong>Inspect</strong> the image</li><li><strong>Determine</strong>
      if the specified label is/are visible in the picture.</li>
```

```
      <li><strong>Outline</strong> each instance of the specified label in the image
 using the provided “Box” tool.</li></ol>
      <ul><li>Boxes should fit tight around each object</li>
      <li>Do not include parts of the object are overlapping or that cannot be seen,
 even though you think you can interpolate the whole shape.</li>
      <li>Avoid including shadows.</li>
      <li>If the target is off screen, draw the box up to the edge of the image.</li>

    </full-instructions>

    <short-instructions>
      <h3><span style="color: rgb(0, 138, 0);">Good example</span></h3>
      <p>Enter description of a correct bounding box label and add images</p>
      <h3><span style="color: rgb(230, 0, 0);">Bad example</span></h3>
      <p>Enter description of an incorrect bounding box label and add images</p>
    </short-instructions>

  </crowd-bounding-box>
</crowd-form>
```

Data Keluaran Kotak Pembatas

Setelah Anda membuat tugas pelabelan kotak pembatas, data keluaran Anda akan ditempatkan di bucket Amazon S3 yang ditentukan dalam`S3OutputPath` parameter saat menggunakan API atau di bidang lokasi set data Output dari bagian Ringkasan Job konsol.

Misalnya, file manifes keluaran dari tugas kotak pembatas kelas tunggal yang berhasil diselesaikan akan berisi hal-hal berikut:

```
[
  {
    "boundingBox": {
      "boundingBoxes": [
        {
          "height": 2832,
          "label": "bird",
          "left": 681,
          "top": 599,
          "width": 1364
        }
      ],
      "inputImageProperties": {
        "height": 3726,
```

```
        "width": 2662
      }
    }
  }
]
```

`boundingBoxes`Parameter mengidentifikasi lokasi kotak pembatas yang digambar di sekitar objek yang diidentifikasi sebagai “burung” relatif terhadap sudut kiri atas gambar yang diambil menjadi koordinat piksel (0,0). Pada contoh sebelumnya, **`left`**dan **`top`**identifikasi lokasi piksel di sudut kiri atas kotak pembatas relatif terhadap sudut kiri atas gambar. Dimensi kotak pembatas diidentifikasi dengan **`height`**dan **`width`**. `inputImageProperties`Parameter memberikan dimensi piksel dari gambar input asli.

Saat Anda menggunakan jenis tugas kotak pembatas, Anda dapat membuat pekerjaan pelabelan kotak pembatas tunggal dan multi-kelas. File manifes keluaran dari kotak pembatas multi-kelas yang berhasil diselesaikan akan berisi yang berikut ini:

```
[
  {
    "boundingBox": {
      "boundingBoxes": [
        {
          "height": 938,
          "label": "squirrel",
          "left": 316,
          "top": 218,
          "width": 785
        },
        {
          "height": 825,
          "label": "rabbit",
          "left": 1930,
          "top": 2265,
          "width": 540
        },
        {
          "height": 1174,
          "label": "bird",
          "left": 748,
          "top": 2113,
          "width": 927
        },
```

```
        {
          "height": 893,
          "label": "bird",
          "left": 1333,
          "top": 847,
          "width": 736
        }
      ],
      "inputImageProperties": {
        "height": 3726,
        "width": 2662
      }
    }
  }
]
```

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan dari pekerjaan pelabelan kotak pembatas, lihatOutput Job Kotak Bounding.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Segmentasi Semantic

Untuk mengidentifikasi konten gambar pada tingkat piksel, gunakan Amazon SageMaker Tugas pelabelan segmentasi semantik Ground Truth. Saat diberi pekerjaan pelabelan segmentasi semantik, pekerja mengklasifikasikan piksel dalam gambar ke dalam sekumpulan label atau kelas yang telah ditentukan. Ground Truth mendukung pekerjaan pelabelan segmentasi semantik tunggal dan multi-kelas.

Gambar yang berisi sejumlah besar objek yang perlu disegmentasi membutuhkan lebih banyak waktu. Untuk membantu pekerja (dari tenaga kerja swasta atau vendor) memberi label pada objek ini dalam waktu yang lebih singkat dan dengan akurasi yang lebih besar, Ground Truth menyediakan alat segmentasi otomatis yang dibantu AI. Untuk informasi, lihat Alat Segmentasi Otomatis.

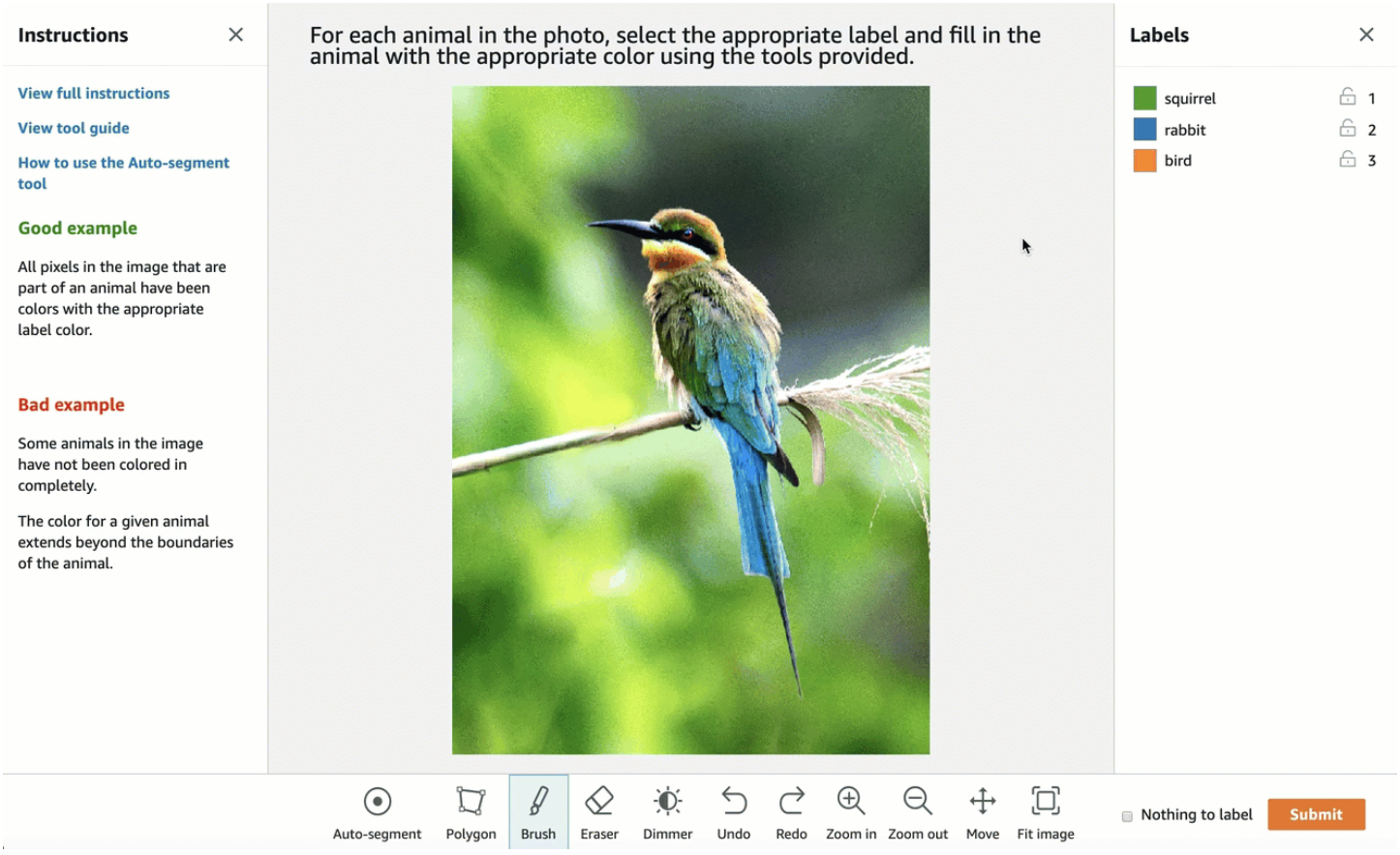
Anda membuat pekerjaan pelabelan segmentasi semantik menggunakan bagian Ground Truth di Amazon SageMakerkonsol atau`CreateLabelingJob`operasi.

> **Important**
>
> Untuk jenis tugas ini, jika Anda membuat file manifes Anda sendiri, gunakan`"source-ref"`untuk mengidentifikasi lokasi dari setiap file gambar di Amazon S3 yang ingin Anda beri label. Untuk informasi selengkapnya, lihat Input Data.

Membuat Job Pelabelan Segmentasi Semantik (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan segmentasi semantik di SageMaker konsol. Pada Langkah 10, pilihCitradariKategori tugasdrop down menu, dan memilihSegmentasi semantiksebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan dengan konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan label yang dapat dipilih pekerja.



Buat Job Pelabelan Segmentasi Semantik (API)

Untuk membuat pekerjaan pelabelan segmentasi semantik, gunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian dari`CreateLabelingJob`.

Ikuti petunjuk diBuat Job Pelabelan (API)dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Pra-anotasi fungsi Lambda untuk jenis tugas ini diakhiri dengan`PRE-SemanticSegmentation`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn.
- Fungsi Lambda anotasi-konsolidasi untuk jenis tugas ini diakhiri dengan`ACS-SemanticSegmentation`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihatAnnotationConsolidationLambdaArn.

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di US East (N. Virginia). Semua parameter berwarna merah harus diganti dengan spesifikasi dan sumber daya Anda.

```
response = client.create_labeling_job(
    LabelingJobName='example-semantic-segmentation-labeling-job,
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
```

```
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/worker-task-template.html'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
SemanticSegmentation,
        'TaskKeywords': [
            'Semantic Segmentation',
        ],
        'TaskTitle': 'Semantic segmentation task',
        'TaskDescription': 'For each category provided, segment out each relevant
 object using the color associated with that category',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-SemanticSegmentation'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Berikan Template untuk Pekerjaan Pelabelan Segmentasi Semantik

Jika Anda membuat pekerjaan pelabelan menggunakan API, Anda harus menyediakan template tugas pekerja di`UiTemplateS3Uri`. Salin dan modifikasi template berikut. Hanya mengubah`short-instructions`,`full-instructions`, dan`header`.

Unggah template ini ke S3, dan berikan URI S3 untuk file ini`UiTemplateS3Uri`.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
```

```
  <crowd-semantic-segmentation
    name="crowd-semantic-segmentation"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Please segment out all pedestrians."
    labels="{{ task.input.labels | to_json | escape }}"
  >
    <full-instructions header="Segmentation instructions">
      <ol><li><strong>Read</strong> the task carefully and inspect the image.</li>
      <li><strong>Read</strong> the options and review the examples provided to
 understand more about the labels.</li>
      <li><strong>Choose</strong> the appropriate label that best suits an object and
 paint that object using the tools provided.</li></ol>
    </full-instructions>
    <short-instructions>
      <h2><span style="color: rgb(0, 138, 0);">Good example</span></h2>
      <p>Enter description to explain a correctly done segmentation</p>
      <p><br></p><h2><span style="color: rgb(230, 0, 0);">Bad example</span></h2>
      <p>Enter description of an incorrectly done segmentation</p>
    </short-instructions>
  </crowd-semantic-segmentation>
</crowd-form>
```

Data Output Segmentasi

Setelah Anda membuat pekerjaan pelabelan segmentasi semantik, data keluaran Anda akan ditempatkan di bucket Amazon S3 yang ditentukan dalam`S3OutputPath`parameter saat menggunakan API atau di`Lokasi set data output`bidang`Gambaran umum Job`bagian dari konsol.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Untuk melihat contoh file manifes keluaran untuk pekerjaan pelabelan segmentasi semantik, lihatOutput Segmentasi Semantik Awan Titik 3D.

Alat Segmentasi Otomatis

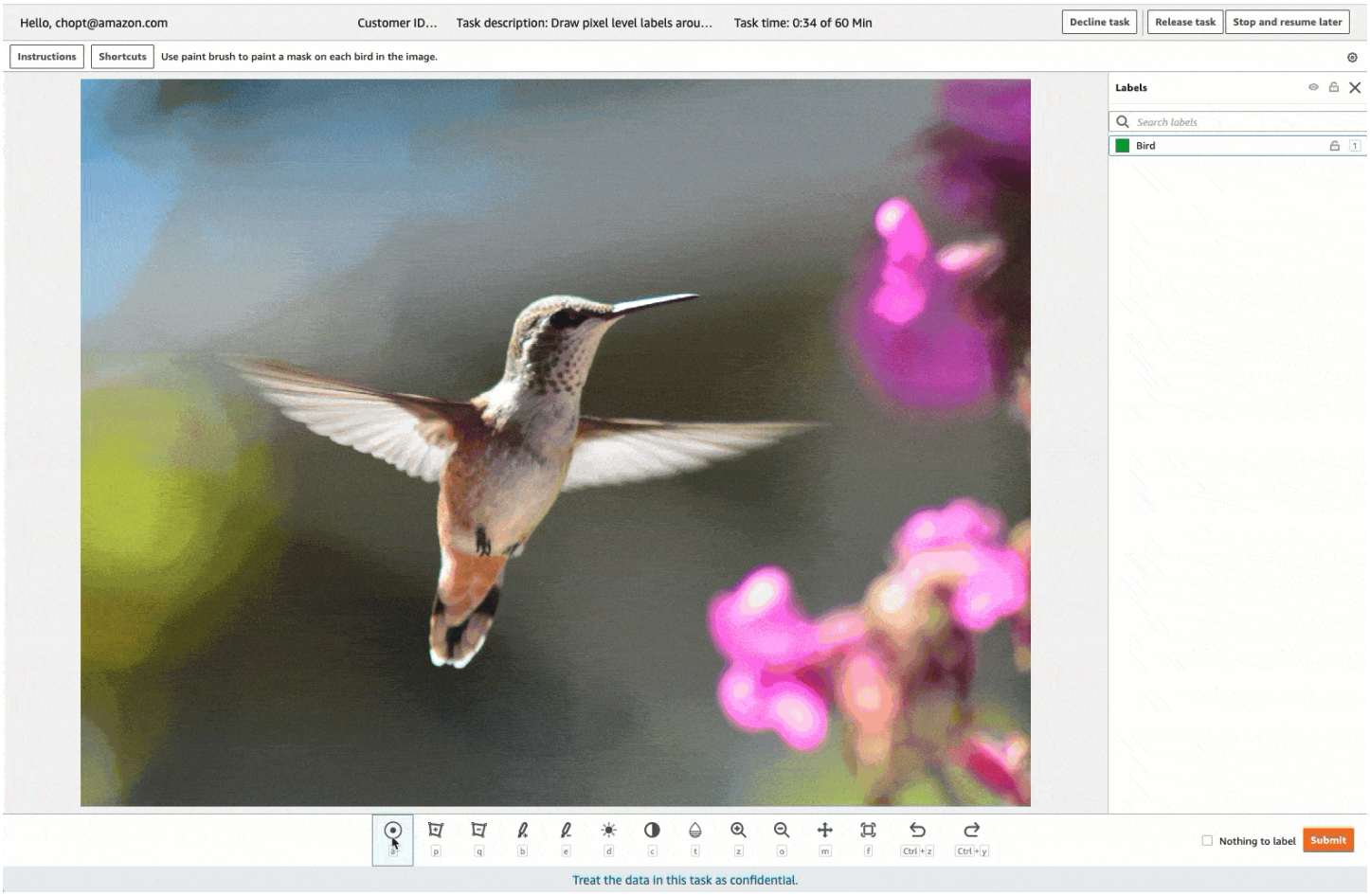
Segmentasi gambar adalah proses membagi gambar menjadi beberapa segmen, atau set piksel berlabel. Di Amazon SageMaker Ground Truth, proses mengidentifikasi semua piksel yang berada di bawah label tertentu melibatkan penerapan pengisi berwarna, atau “topeng”, di atas piksel tersebut. Beberapa tugas pekerjaan pelabelan berisi gambar dengan sejumlah besar objek yang perlu tersegmentasi. Untuk membantu pekerja memberi label pada objek ini dalam waktu yang lebih singkat dan dengan akurasi yang lebih besar, Ground Truth menyediakan alat segmentasi otomatis

untuk tugas segmentasi yang ditugaskan ke tenaga kerja pribadi dan vendor. Alat ini menggunakan model pembelajaran mesin untuk secara otomatis mengelompokkan objek individual dalam gambar dengan input pekerja minimal. Pekerja dapat menyempurnakan topeng yang dihasilkan oleh alat segmentasi otomatis menggunakan alat lain yang ditemukan di konsol pekerja. Ini membantu pekerja menyelesaikan tugas segmentasi gambar dengan lebih cepat dan lebih akurat, sehingga menghasilkan biaya yang lebih rendah dan kualitas label yang lebih tinggi.

> **Note**
>
> Alat segmentasi otomatis tersedia untuk tugas segmentasi yang dikirim ke tenaga kerja pribadi atau tenaga kerja vendor. Ini tidak tersedia untuk tugas yang dikirim ke tenaga kerja publik (Amazon Mechanical Turk).

Pratinjau Alat

Ketika pekerja diberi pekerjaan pelabelan yang menyediakan alat segmentasi otomatis, mereka diberikan petunjuk terperinci tentang cara menggunakan alat ini. Misalnya, pekerja mungkin melihat hal berikut di konsol pekerja:



Pekerja dapat menggunakanLihat instruksi lengkapuntuk mempelajari cara menggunakan alat ini. Pekerja perlu menempatkan titik pada empat titik ekstrem (paling atas, paling bawah, paling kiri, dan paling kanan) dari objek yang diminati, dan alat akan secara otomatis menghasilkan topeng untuk objek tersebut. Pekerja dapat menyempurnakan masker lebih lanjut menggunakan alat lain yang disediakan, atau dengan menggunakan alat segmen otomatis pada bagian yang lebih kecil dari objek yang terlewatkan.

Alat

Alat segmentasi otomatis muncul secara otomatis di konsol pekerja Anda jika Anda membuat pekerjaan pelabelan segmentasi semantik menggunakan Amazon SageMaker konsol. Sementara menciptakan pekerjaan segmentasi semantik di SageMaker konsol, Anda akan dapat melihat pratinjau alat saat membuat instruksi pekerja. Untuk mempelajari cara membuat pekerjaan pelabelan segmentasi semantik di SageMaker konsol, lihatMulai.

Jika Anda membuat pekerjaan pelabelan segmentasi instans kustom di SageMaker konsol atau membuat pekerjaan pelabelan segmentasi instans atau semantik menggunakan Ground Truth API,

Anda perlu membuat template tugas khusus untuk merancang konsol dan instruksi pekerja Anda. Untuk menyertakan alat segmentasi otomatis di konsol pekerja Anda, pastikan bahwa ketentuan berikut terpenuhi dalam template tugas kustom Anda:

- Untuk pekerjaan pelabelan segmentasi semantik yang dibuat menggunakan API,`<crowd-semantic-segmentation>`Ada dalam template tugas. Untuk pekerjaan pelabelan segmentasi instans kustom,`<crowd-instance-segmentation>`tag ini ada dalam template tugas.
- Tugas ditugaskan ke tenaga kerja pribadi atau tenaga kerja vendor.
- Gambar yang akan diberi label adalah objek Amazon Simple Storage Service Amazon S3) yang telah ditandatangani sebelumnya untuk Worker sehingga mereka dapat mengaksesnya. Hal ini berlaku jika template tugas menyertakan`grant_read_access`filter. Untuk informasi tentang`grant_read_access`filter, lihatMenambahkan otomatisasi dengan Liquid.

Berikut ini adalah contoh template tugas kustom untuk pekerjaan pelabelan segmentasi instans kustom, yang mencakup`<crowd-instance-segmentation/>`tag dan`grant_read_access`Filter cair.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-instance-segmentation
    name="crowd-instance-segmentation"
    src="{{ task.input.taskObject | grant_read_access }}"
    labels="['Car','Road']"
   <full-instructions header="Segmentation instructions">
      Segment each instance of each class of objects in the image.
    </full-instructions>

    <short-instructions>
      <p>Segment each instance of each class of objects in the image.</p>

      <h3 style="color: green">GOOD EXAMPLES</h3>
      <img src="path/to/image.jpg" style="width: 100%">
      <p>Good because A, B, C.</p>

      <h3 style="color: red">BAD EXAMPLES</h3>
      <img src="path/to/image.jpg" style="width: 100%">
      <p>Bad because X, Y, Z.</p>
    </short-instructions>
  </crowd-instance-segmentation>
```

```
</crowd-form>
```

Klasifikasi Gambar (Label Tunggal)

Gunakan Amazon SageMaker Tugas pelabelan klasifikasi gambar Ground Truth saat Anda membutuhkan pekerja untuk mengklasifikasikan gambar menggunakan label yang telah ditentukan sebelumnya yang Anda tentukan. Pekerja ditampilkan gambar dan diminta untuk memilih satu label untuk setiap gambar.

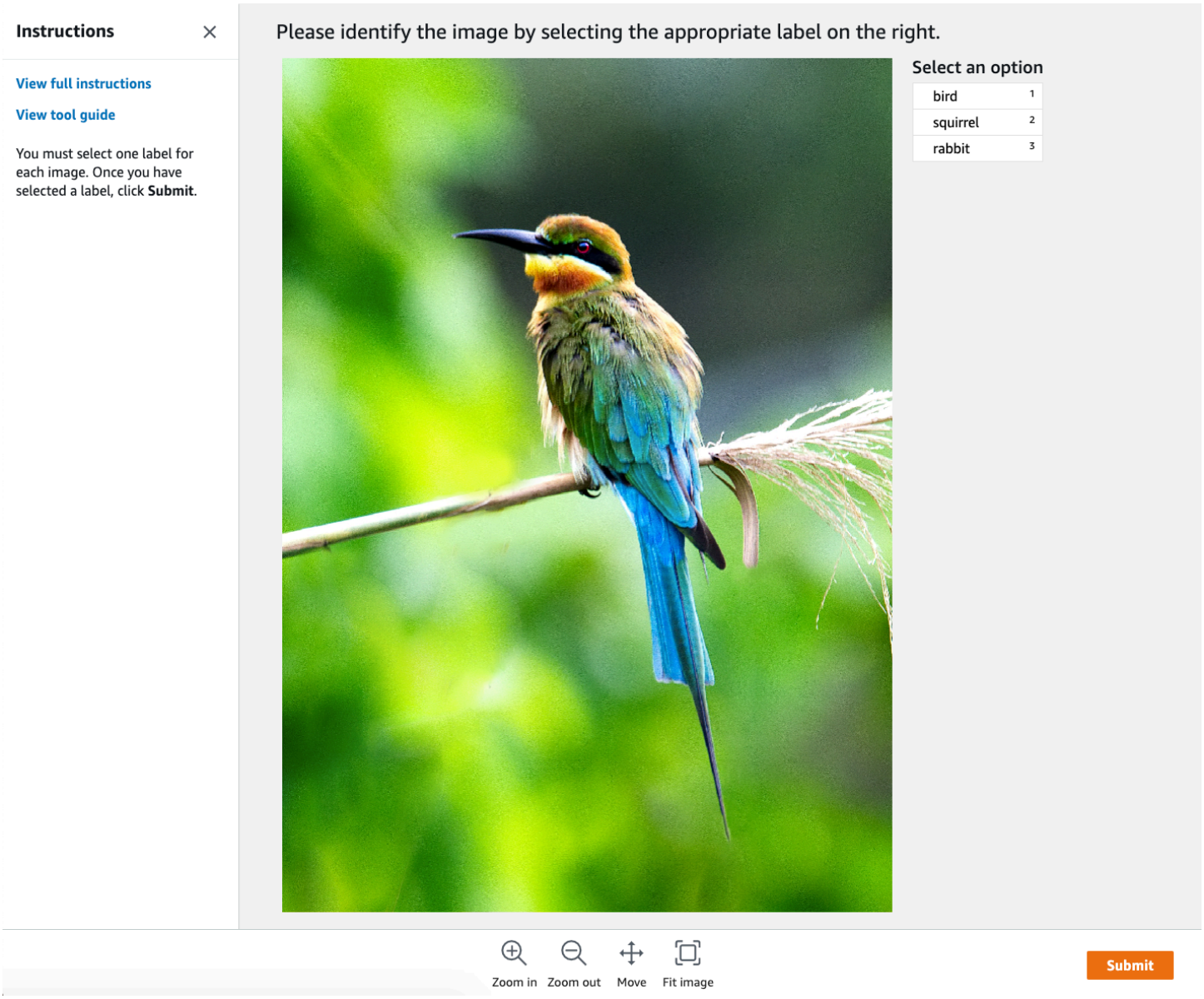
Anda dapat membuat pekerjaan pelabelan klasifikasi gambar menggunakan bagian Ground Truth di Amazon SageMakerkonsol atau`CreateLabelingJob`operasi.

> **Important**
>
> Untuk jenis tugas ini, jika Anda membuat file manifes Anda sendiri, gunakan`"source-ref"`untuk mengidentifikasi lokasi setiap file gambar di Amazon S3 yang ingin Anda beri label. Untuk informasi selengkapnya, lihat Input Data.

Buat Job Pelabelan Klasifikasi Gambar (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan klasifikasi gambar di SageMaker konsol. Pada Langkah 10, pilihCitradariKategori tugasdrop down menu, dan memilihKlasifikasi Gambar (Label Tunggal)sebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan dengan konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan label yang dapat dipilih pekerja.



Buat Job Pelabelan Klasifikasi Gambar (API)

Untuk membuat pekerjaan pelabelan klasifikasi gambar, gunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian dari`CreateLabelingJob`.

Ikuti petunjuk diBuat Job Pelabelan (API)dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Pra-anotasi fungsi Lambda untuk jenis tugas ini diakhiri dengan`PRE-ImageMultiClass`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn.

- Fungsi Lambda anotasi-konsolidasi untuk jenis tugas ini diakhiri dengan`ACS-ImageMultiClass`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihatAnnotationConsolidationLambdaArn.

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di US East (N. Virginia). Semua parameter berwarna merah harus diganti dengan spesifikasi dan sumber daya Anda.

```
response = client.create_labeling_job(
    LabelingJobName='example-image-classification-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/worker-task-template.html'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
ImageMultiClass,
        'TaskKeywords': [
            Image classification',
```

```
        ],
        'TaskTitle': 'Image classification task',
        'TaskDescription': 'Carefully inspect the image and classify it by selecting
 one label from the categories provided.',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-ImageMultiClass'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Menyediakan Template untuk Pekerjaan Pelabelan Klasifikasi Gambar

Jika Anda membuat pekerjaan pelabelan menggunakan API, Anda harus menyediakan template tugas pekerja di`UiTemplateS3Uri`. Salin dan modifikasi template berikut. Hanya mengubah`short-instructions`,`full-instructions`, dan`header`.

Unggah template ini ke S3, dan berikan URI S3 untuk file ini`UiTemplateS3Uri`.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-image-classifier
    name="crowd-image-classifier"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="please classify"
    categories="{{ task.input.labels | to_json | escape }}"
  >
    <full-instructions header="Image classification instructions">
      <ol><li><strong>Read</strong> the task carefully and inspect the image.</li>
      <li><strong>Read</strong> the options and review the examples provided to
 understand more about the labels.</li>
      <li><strong>Choose</strong> the appropriate label that best suits the image.</
li></ol>
    </full-instructions>
```

```
    <short-instructions>
      <h3><span style="color: rgb(0, 138, 0);">Good example</span></h3>
      <p>Enter description to explain the correct label to the workers</p>
      <h3><span style="color: rgb(230, 0, 0);">Bad example</span></h3><p>Enter
 description of an incorrect label</p>
    </short-instructions>
  </crowd-image-classifier>
</crowd-form>
```

Data keluaran klasifikasi Citra

Setelah Anda membuat pekerjaan pelabelan klasifikasi gambar, data keluaran Anda akan ditempatkan di bucket Amazon S3 yang ditentukan dalam`S3OutputPath`parameter saat menggunakan API atau di`Lokasi set data output`bidang`Gambaran umum Job`bagian konsol.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Untuk melihat contoh file manifes keluaran dari pekerjaan pelabelan klasifikasi gambar, lihatOutput.

Klasifikasi Gambar (Multi-label)

Gunakan Amazon SageMaker Tugas pelabelan klasifikasi gambar multi-label Ground Truth saat Anda membutuhkan pekerja untuk mengklasifikasikan beberapa objek dalam gambar. Misalnya, gambar berikut ini menampilkan seekor anjingnya dan kucing. Anda dapat menggunakan klasifikasi gambar multi-label untuk mengaitkan label “dog” dan “cat” dengan gambar ini.

Saat mengerjakan tugas klasifikasi gambar multi-label, pekerja harus memilih semua label yang berlaku, tetapi harus memilih setidaknya satu. Saat membuat pekerjaan menggunakan jenis tugas ini, Anda dapat menyediakan hingga 50 kategori label.

Saat membuat pekerjaan pelabelan di konsol, Ground Truth tidak menyediakan kategori “tidak ada” ketika tidak ada label yang berlaku untuk gambar. Untuk memberikan opsi ini kepada pekerja,

sertakan label yang mirip dengan “none” atau “other” saat Anda membuat pekerjaan klasifikasi gambar multi-label.

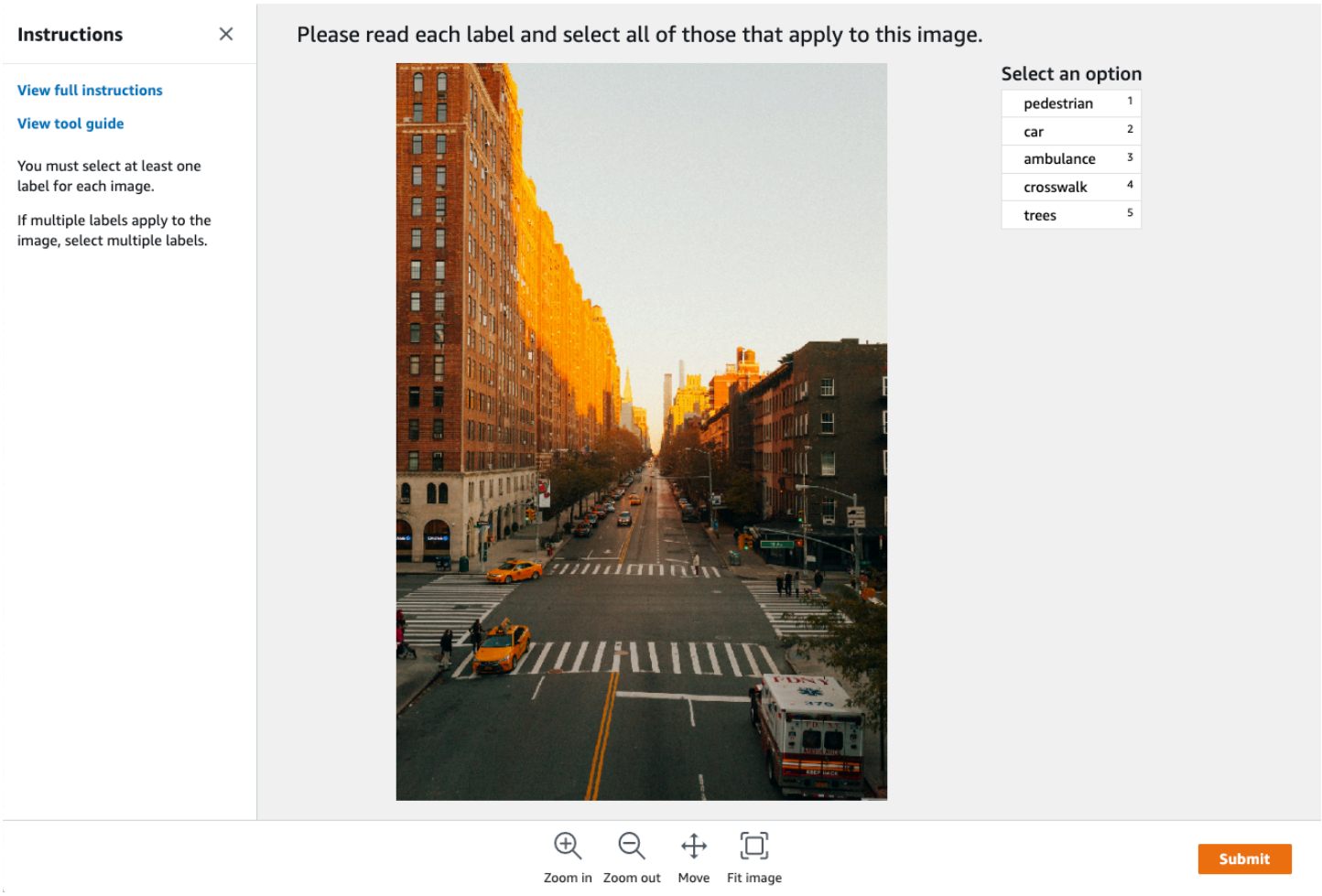
Untuk membatasi pekerja memilih label tunggal untuk setiap gambar, gunakanKlasifikasi Gambar (Label Tunggal)jenis tugas.

> **Important**
>
> Untuk jenis tugas ini, jika Anda membuat file manifes Anda sendiri, gunakan`"source-ref"`untuk mengidentifikasi lokasi dari setiap file gambar di Amazon S3 yang ingin Anda beri label. Untuk informasi selengkapnya, lihat Input Data.

Buat Job Pelabelan Klasifikasi Gambar Multi-Label (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan klasifikasi gambar multi-label di SageMaker konsol. Pada Langkah 10, pilihCitradariKategori tugasdrop down menu, dan memilihKlasifikasi Gambar (Multi-label)sebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan di konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan label yang dapat dipilih pekerja.



Membuat Job Pelabelan Klasifikasi Gambar Multi-Label (API)

Untuk membuat pekerjaan pelabelan klasifikasi gambar multi-label, gunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian dari`CreateLabelingJob`.

Ikuti petunjuknya diBuat Job Pelabelan (API)dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Pra-anotasi fungsi Lambda untuk jenis tugas ini diakhiri dengan`PRE-ImageMultiClassMultiLabel`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn.
- Fungsi Lambda anotasi-konsolidasi untuk jenis tugas ini diakhiri dengan`ACS-ImageMultiClassMultiLabel`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihatAnnotationConsolidationLambdaArn.

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di US East (N. Virginia). Semua parameter berwarna merah harus diganti dengan spesifikasi dan sumber daya Anda.

```
response = client.create_labeling_job(
    LabelingJobName='example-multi-label-image-classification-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/worker-task-template.html'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
ImageMultiClassMultiLabel',
        'TaskKeywords': [
            'Image Classification',
        ],
        'TaskTitle': 'Multi-label image classification task',
        'TaskDescription': 'Select all labels that apply to the images shown',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
```

```
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-ImageMultiClassMultiLabel'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Menyediakan Template untuk Klasifikasi Gambar Multi-label

Jika Anda membuat pekerjaan pelabelan menggunakan API, Anda harus menyediakan template tugas pekerja`UiTemplateS3Uri`. Salin dan modifikasi template berikut. Hanya mengubah`short-instructions`,`full-instructions`, dan`header`.

Unggah template ini ke S3, dan berikan URI S3 untuk file ini`UiTemplateS3Uri`.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-image-classifier-multi-select
    name="crowd-image-classifier-multi-select"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Please identify all classes in image"
    categories="{{ task.input.labels | to_json | escape }}"
  >
    <full-instructions header="Multi Label Image classification instructions">
      <ol><li><strong>Read</strong> the task carefully and inspect the image.</li>
      <li><strong>Read</strong> the options and review the examples provided to
 understand more about the labels.</li>
       <li><strong>Choose</strong> the appropriate labels that best suit the image.</
li></ol>
    </full-instructions>
    <short-instructions>
      <h3><span style="color: rgb(0, 138, 0);">Good example</span></h3>
      <p>Enter description to explain the correct label to the workers</p>
      <h3><span style="color: rgb(230, 0, 0);">Bad example</span></h3>
      <p>Enter description of an incorrect label</p>
   </short-instructions>
```

```
  </crowd-image-classifier-multi-select>
</crowd-form>
```

Data Keluaran Klasifikasi Gambar Multi-label

Setelah Anda membuat pekerjaan pelabelan klasifikasi gambar multi-label, data keluaran Anda akan ditempatkan di bucket Amazon S3 yang ditentukan dalam`S3OutputPath`parameter saat menggunakan API atau diLokasi set data outputbidangGambaran umum Jobbagian dari konsol.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Untuk melihat contoh file manifes keluaran untuk pekerjaan pelabelan klasifikasi gambar multi-label, lihatOutput Job Klasifikasi Multi-label.

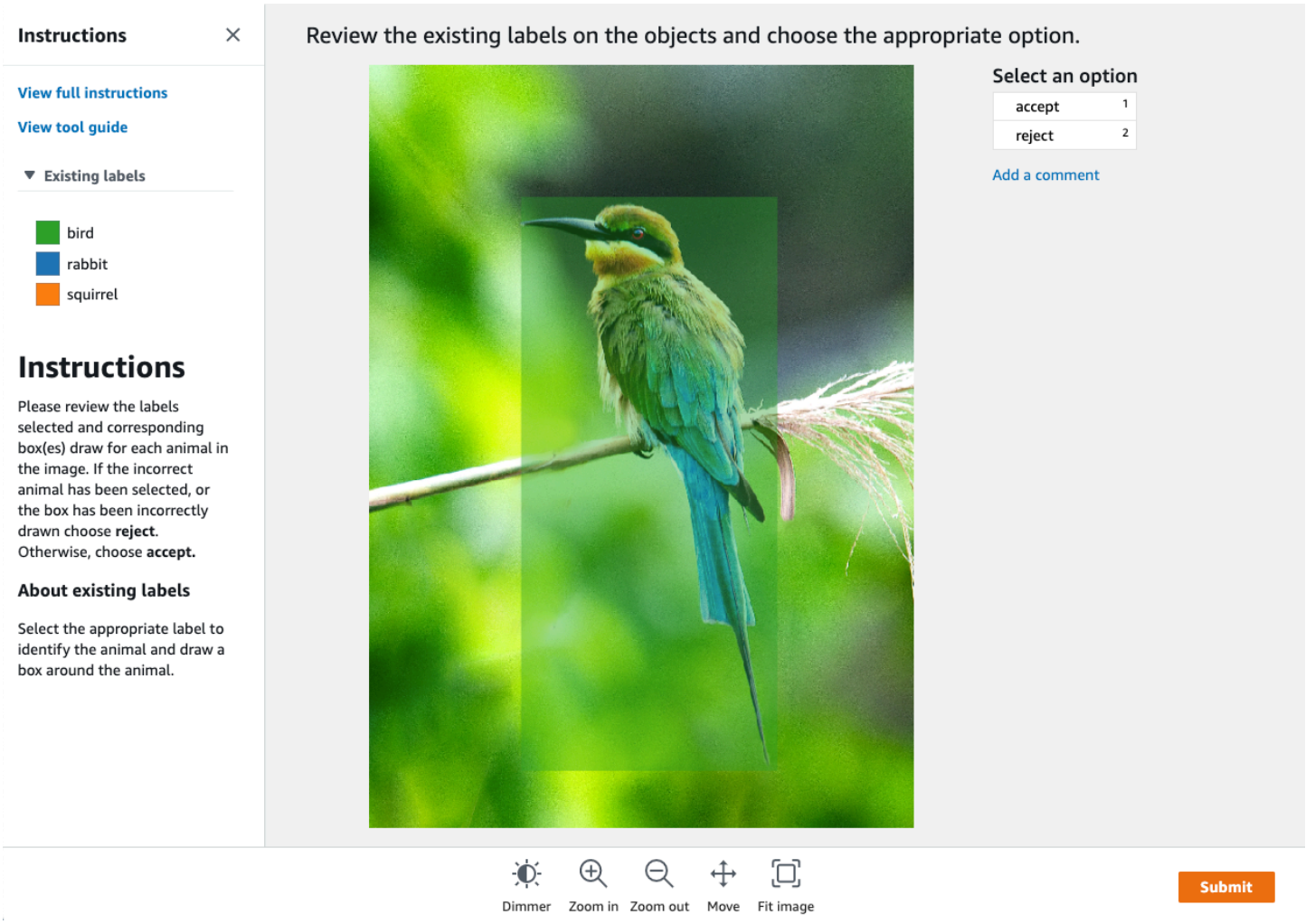
Label verifikasi gambar

Membangun kumpulan data pelatihan yang sangat akurat untuk algoritma machine learning (ML/machine learning) Anda adalah proses berulang. Biasanya, Anda meninjau dan terus menyesuaikan label Anda sampai Anda puas bahwa label tersebut secara akurat mewakili kebenaran dasar, atau apa yang dapat diamati secara langsung di dunia nyata.

Anda dapat menggunakan Amazon SageMaker Tugas verifikasi label gambar Ground Truth untuk mengarahkan pekerja untuk meninjau label kumpulan data dan meningkatkan akurasi label. Pekerja dapat menunjukkan apakah label yang ada benar atau menilai kualitas label. Mereka juga dapat menambahkan komentar untuk menjelaskan alasan mereka. Amazon SageMaker Ground Truth mendukung verifikasi label untukKotak pembatasdanSegmentasi Semanticlabel.

Anda membuat pekerjaan pelabelan verifikasi label gambar menggunakan bagian Ground Truth di Amazon SageMaker konsol atauCreateLabelingJoboperasi.

Ground Truth menyediakan konsol pekerja yang mirip dengan yang berikut untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan dengan konsol, Anda dapat memodifikasi gambar dan konten yang ditampilkan. Untuk mempelajari cara membuat pekerjaan pelabelan dengan menggunakan konsol Ground Truth, lihatMembuat Job Pelabelan (Konsol).



Anda dapat membuat pekerjaan pelabelan verifikasi label menggunakan SageMaker konsol atau API. Untuk mempelajari cara membuat pekerjaan pelabelan menggunakan operasi Ground Truth API`CreateLabelingJob`, LihatBuat Job Pelabelan (API).

Gunakan Ground Truth untuk Label Teks

Gunakan Ground Truth untuk memberi label teks. Pilih salah satu jenis tugas bawaan berikut untuk mempelajari lebih lanjut tentang jenis tugas tersebut. Setiap halaman menyertakan instruksi untuk membantu Anda membuat pekerjaan pelabelan menggunakan jenis tugas tersebut.

> **Tip**
>
> Untuk mempelajari lebih lanjut tentang jenis file yang didukung dan kuota data input, lihatInput Data.

Topik

Pengenalan Entitas bernama

Untuk mengekstrak informasi dari teks tidak terstruktur dan mengklasifikasikannya ke dalam kategori yang telah ditentukan, gunakan Amazon SageMaker Ground Truth bernama pengenalan entitas (NER) tugas pelabelan. Secara tradisional, NER melibatkan memilah-milah data teks untuk menemukan frasa kata benda, yang disebutentitas bernama, dan mengkategorikan masing-masing dengan label, seperti “orang,” “organisasi,” atau “merek.” Anda dapat memperluas tugas ini untuk memberi label rentang teks yang lebih panjang dan mengkategorikan urutan tersebut dengan label yang telah ditentukan sebelumnya yang Anda tentukan.

Saat ditugaskan dengan pekerjaan pelabelan pengenalan entitas bernama, pekerja menerapkan label Anda ke kata atau frasa tertentu dalam blok teks yang lebih besar. Mereka memilih label, lalu menerapkannya dengan menggunakan kursor untuk menyorot bagian teks tempat label berlaku. Alat pengenalan entitas bernama Ground Truth mendukung anotasi yang tumpang tindih, pemilihan label dalam konteks, dan pemilihan multi-label untuk satu sorotan. Selain itu, pekerja dapat menggunakan keyboard mereka untuk memilih label dengan cepat.

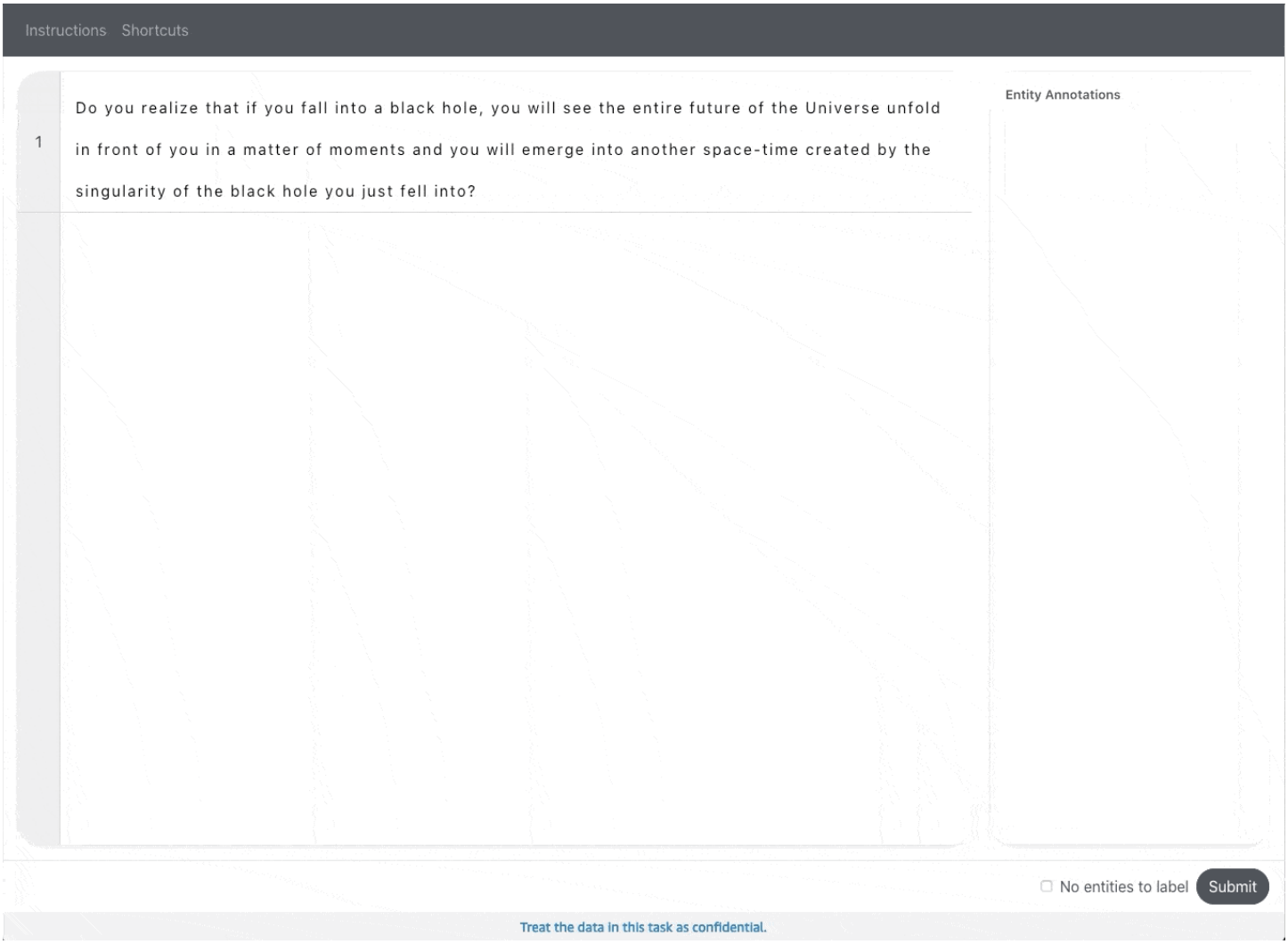
Anda dapat membuat pekerjaan pelabelan pengenalan entitas bernama menggunakan bagian Ground Truth di Amazon SageMakerkonsol atau`CreateLabelingJob`operasi.

> **Important**
>
> Jika Anda membuat file manifes masukan secara manual, gunakan`"source"`untuk mengidentifikasi teks yang Anda ingin diberi label. Untuk informasi selengkapnya, lihat Input Data.

Buat Job Pelabelan Pengenalan Entitas Bernama (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan pengenalan entitas bernama di SageMaker konsol Pada Langkah 10, pilihTextdariKategori tugasdrop down menu, dan memilihPengenalan entitas bernamasebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan dengan konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan label yang dapat dipilih pekerja.



Membuat Entity Recognition Labeling Job (API) Bernama

Untuk membuat pekerjaan pelabelan pengenalan entitas bernama, menggunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian dari`CreateLabelingJob`.

Ikuti instruksi diBuat Job Pelabelan (API)dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Pra-anotasi fungsi Lambda untuk jenis tugas ini diakhiri dengan`PRE-NamedEntityRecognition`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn.

- Fungsi Lambda anotasi-konsolidasi untuk jenis tugas ini diakhiri dengan`ACS-NamedEntityRecognition`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihat`AnnotationConsolidationLambdaArn`.
- Anda harus memberikan ARN berikut`HumanTaskUiArn`:

```
arn:aws:sagemaker:aws-region:394669845002:human-task-ui/NamedEntityRecognition
```

  Ganti*aws-region*dengan Wilayah AWS yang Anda gunakan untuk membuat pekerjaan pelabelan. Misalnya, gunakan`us-west-1`jika Anda membuat pekerjaan pelabelan di US West (N. California).

- Berikan instruksi pekerja dalam file konfigurasi kategori label menggunakan`instructions`parameter Anda dapat menggunakan string, atau bahasa markup HTML di`shortInstruction`dan`fullInstruction`Bidang bidang. Untuk rincian lebih lanjut, lihat Menyediakan Instruksi Pekerja dalam File Konfigurasi Kategori Label.

```
"instructions": {"shortInstruction":"<h1>Add header</h1><p>Add Instructions</p>",
 "fullInstruction":"<p>Add additional instructions.</p>"}
```

Berikut ini adalah contoh dari sebuahAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di US East (N. Virginia). Semua parameter berwarna merah harus diganti dengan spesifikasi dan sumber daya Anda.

```
response = client.create_labeling_job(
    LabelingJobName='example-ner-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
```

```
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'HumanTaskUiArn': 'arn:aws:sagemaker:us-east-1:394669845002:human-task-ui/
NamedEntityRecognition'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
NamedEntityRecognition',
        'TaskKeywords': [
            'Named entity Recognition',
        ],
        'TaskTitle': 'Named entity Recognition task',
        'TaskDescription': 'Apply the labels provided to specific words or phrases
 within the larger text block.',
        'NumberOfHumanWorkersPerDataObject': 1,
        'TaskTimeLimitInSeconds': 28800,
        'TaskAvailabilityLifetimeInSeconds': 864000,
        'MaxConcurrentTaskCount': 1000,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-NamedEntityRecognition'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Menyediakan Instruksi Pekerja dalam File Konfigurasi Kategori Label

Anda harus memberikan instruksi pekerja dalam file konfigurasi kategori label yang Anda identifikasi dengan`LabelCategoryConfigS3Uri`parameter dalam`CreateLabelingJob`. Anda dapat

menggunakan instruksi ini untuk memberikan detail tentang tugas yang Anda inginkan untuk dilakukan pekerja dan membantu mereka menggunakan alat ini secara efisien.

Anda memberikan instruksi singkat dan panjang menggunakan`shortInstruction`dan`fullInstruction`di dalam`instructions`parameter, masing-masing. Untuk mempelajari selengkapnya tentang tipe instruksi ini, lihatMembuat Halaman Instruksi.

Berikut ini adalah contoh file konfigurasi kategori label dengan instruksi yang dapat digunakan untuk pekerjaan pelabelan pengenalan entitas bernama.

```
{
  "document-version": "2018-11-28",
  "labels": [
    {
      "label": "label1",
      "shortDisplayName": "L1"
    },
    {
      "label": "label2",
      "shortDisplayName": "L2"
    },
    {
      "label": "label3",
      "shortDisplayName": "L3"
    },
    {
      "label": "label4",
      "shortDisplayName": "L4"
    },
    {
      "label": "label5",
      "shortDisplayName": "L5"
    }
  ],
  "instructions": {
    "shortInstruction": "<p>Enter description of the labels that workers have
                        to choose from</p><br><p>Add examples to help workers
 understand the label</p>",
    "fullInstruction": "<ol>
                        <li><strong>Read</strong> the text carefully.</li>
                        <li><strong>Highlight</strong> words, phrases, or sections of
 the text.</li>
```

```
                        <li><strong>Choose</strong> the label that best matches what
 you have highlighted.</li>
                        <li>To <strong>change</strong> a label, choose highlighted text
 and select a new label.</li>
                        <li>To <strong>remove</strong> a label from highlighted text,
 choose the X next to the
                        abbreviated label name on the highlighted text.</li>
                        <li>You can select all of a previously highlighted text, but
 not a portion of it.</li>
                        </ol>"
  }
}
```

Data Keluaran Pengenalan Entitas Bernama

Setelah Anda membuat pekerjaan pelabelan pengenalan entitas bernama, data keluaran Anda akan ditempatkan di bucket Amazon S3 yang ditentukan dalam`S3OutputPath`parameter saat menggunakan API atau di`Lokasi set data output`bidang`Gambaran umum Job`bagian dari konsol.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Klasifikasi Teks (Label Tunggal)

Untuk mengkategorikan artikel dan teks ke dalam kategori yang telah ditentukan, gunakan klasifikasi teks. Misalnya, Anda dapat menggunakan klasifikasi teks untuk mengidentifikasi sentimen yang disampaikan dalam ulasan atau emosi yang mendasari bagian teks. Gunakan Amazon SageMaker Klasifikasi teks Ground Truth agar pekerja mengurutkan teks ke dalam kategori yang Anda tentukan.

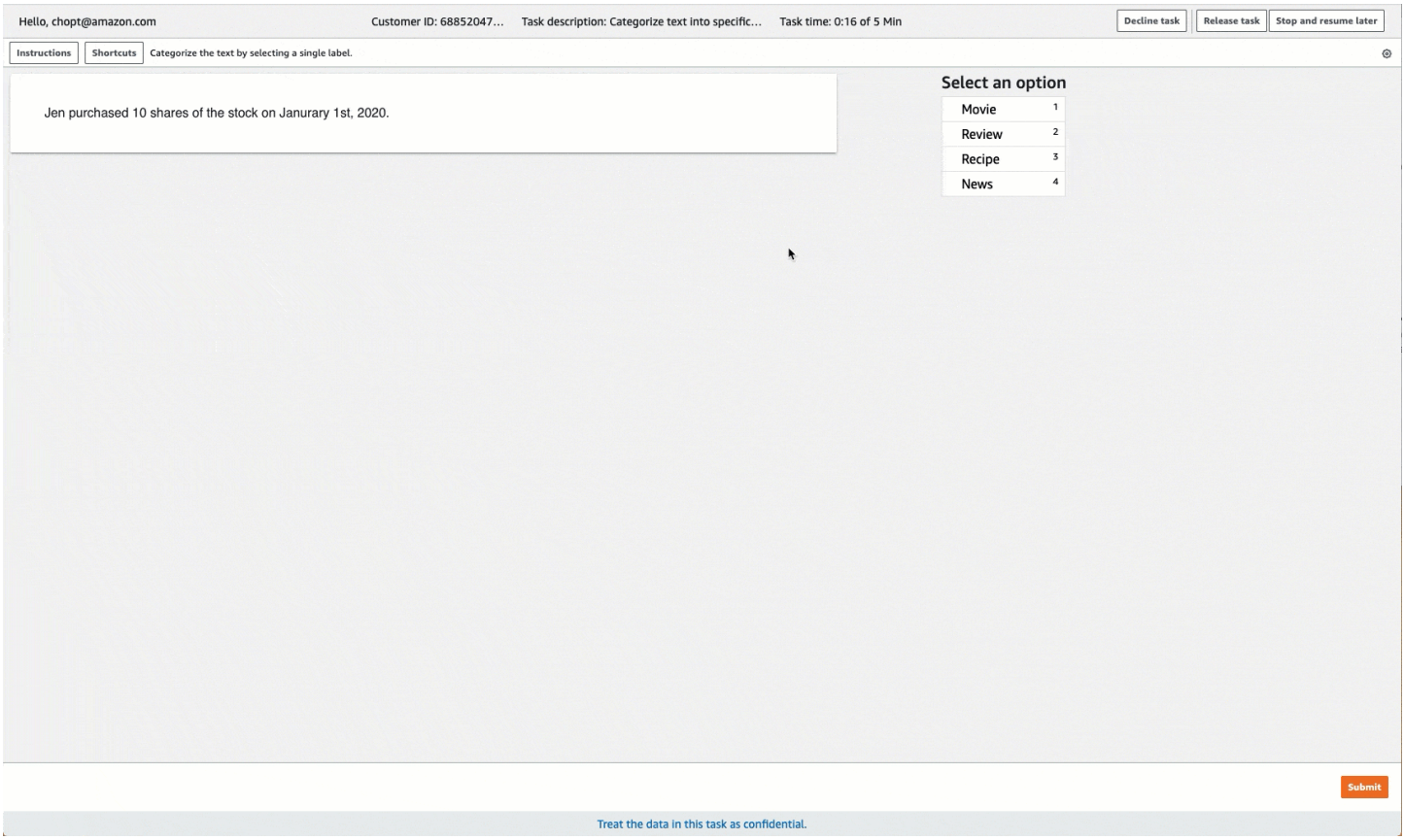
Anda membuat pekerjaan pelabelan klasifikasi teks menggunakan bagian Ground Truth di Amazon SageMaker konsol atau`CreateLabelingJob`operasi.

> **Important**
>
> Jika Anda membuat file manifes masukan secara manual, gunakan`"source"`untuk mengidentifikasi teks yang ingin diberi label. Untuk informasi selengkapnya, lihat Input Data.

Buat Job Pelabelan Klasifikasi Teks (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan klasifikasi teks di SageMaker konsol. Pada Langkah 10, pilihTextdariKategori tugasdrop down menu, dan memilihKlasifikasi Teks (Label Tunggal)sebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan dengan konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan label yang dapat dipilih pekerja.



Buat Job Pelabelan Klasifikasi Teks (API)

Untuk membuat pekerjaan pelabelan klasifikasi teks, gunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian dari`CreateLabelingJob`.

Ikuti petunjuk diBuat Job Pelabelan (API)dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Pra-anotasi fungsi Lambda untuk jenis tugas ini diakhiri dengan`PRE-TextMultiClass`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn.
- Fungsi Lambda anotasi-konsolidasi untuk jenis tugas ini diakhiri dengan`ACS-TextMultiClass`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihatAnnotationConsolidationLambdaArn.

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di Wilayah US East (N. Virginia). Semua parameter berwarna merah harus diganti dengan spesifikasi dan sumber daya Anda.

```
response = client.create_labeling_job(
    LabelingJobName='example-text-classification-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/worker-task-template.html'
        },
```

```
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
TextMultiClass,
        'TaskKeywords': [
            Text classification',
        ],
        'TaskTitle': Text classification task',
        'TaskDescription': 'Carefully read and classify this text using the categories
 provided.',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-TextMultiClass'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Berikan Template untuk Pekerjaan Pelabelan Klasifikasi Teks

Jika Anda membuat pekerjaan pelabelan menggunakan API, Anda harus menyediakan template tugas pekerja di`UiTemplateS3Uri`. Salin dan modifikasi template berikut. Hanya mengubah`short-instructions`,`full-instructions`, dan`header`.

Unggah template ini ke S3, dan berikan URI S3 untuk file ini`UiTemplateS3Uri`.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-classifier
    name="crowd-classifier"
    categories="{{ task.input.labels | to_json | escape }}"
    header="classify text"
  >
    <classification-target style="white-space: pre-wrap">
      {{ task.input.taskObject }}
    </classification-target>
    <full-instructions header="Classifier instructions">
```

```
      <ol><li><strong>Read</strong> the text carefully.</li>
      <li><strong>Read</strong> the examples to understand more about the options.</li>
      <li><strong>Choose</strong> the appropriate labels that best suit the text.</
li></ol>
    </full-instructions>
    <short-instructions>
      <p>Enter description of the labels that workers have to choose from</p>
      <p><br></p><p><br></p><p>Add examples to help workers understand the label</p>
      <p><br></p><p><br></p><p><br></p><p><br></p><p><br></p>
    </short-instructions>
  </crowd-classifier>
  </crowd-form>
```

Data Output Klasifikasi Teks

Setelah Anda membuat tugas pelabelan klasifikasi teks, data keluaran Anda akan ditempatkan di bucket Amazon S3 yang ditentukan dalam`S3OutputPath`parameter saat menggunakan API atau di`Lokasi set data output`bidang`Gambaran umum Job`bagian konsol.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Untuk melihat contoh file manifes keluaran dari tugas pelabelan klasifikasi teks, lihatOutput.

Klasifikasi Teks (Multi-label)

Untuk mengkategorikan artikel dan teks ke dalam beberapa kategori yang telah ditentukan, gunakan jenis tugas klasifikasi teks multi-label. Misalnya, Anda dapat menggunakan jenis tugas ini untuk mengidentifikasi lebih dari satu emosi yang disampaikan dalam teks.

Saat mengerjakan tugas klasifikasi teks multi-label, pekerja harus memilih semua label yang berlaku, tetapi harus memilih setidaknya satu. Saat membuat pekerjaan menggunakan jenis tugas ini, Anda dapat menyediakan hingga 50 kategori label.

Amazon SageMaker Ground Truth tidak menyediakan kategori “tidak ada” ketika tidak ada label yang berlaku. Untuk memberikan opsi ini kepada pekerja, sertakan label yang mirip dengan “none” atau “other” saat Anda membuat tugas klasifikasi teks multi-label.

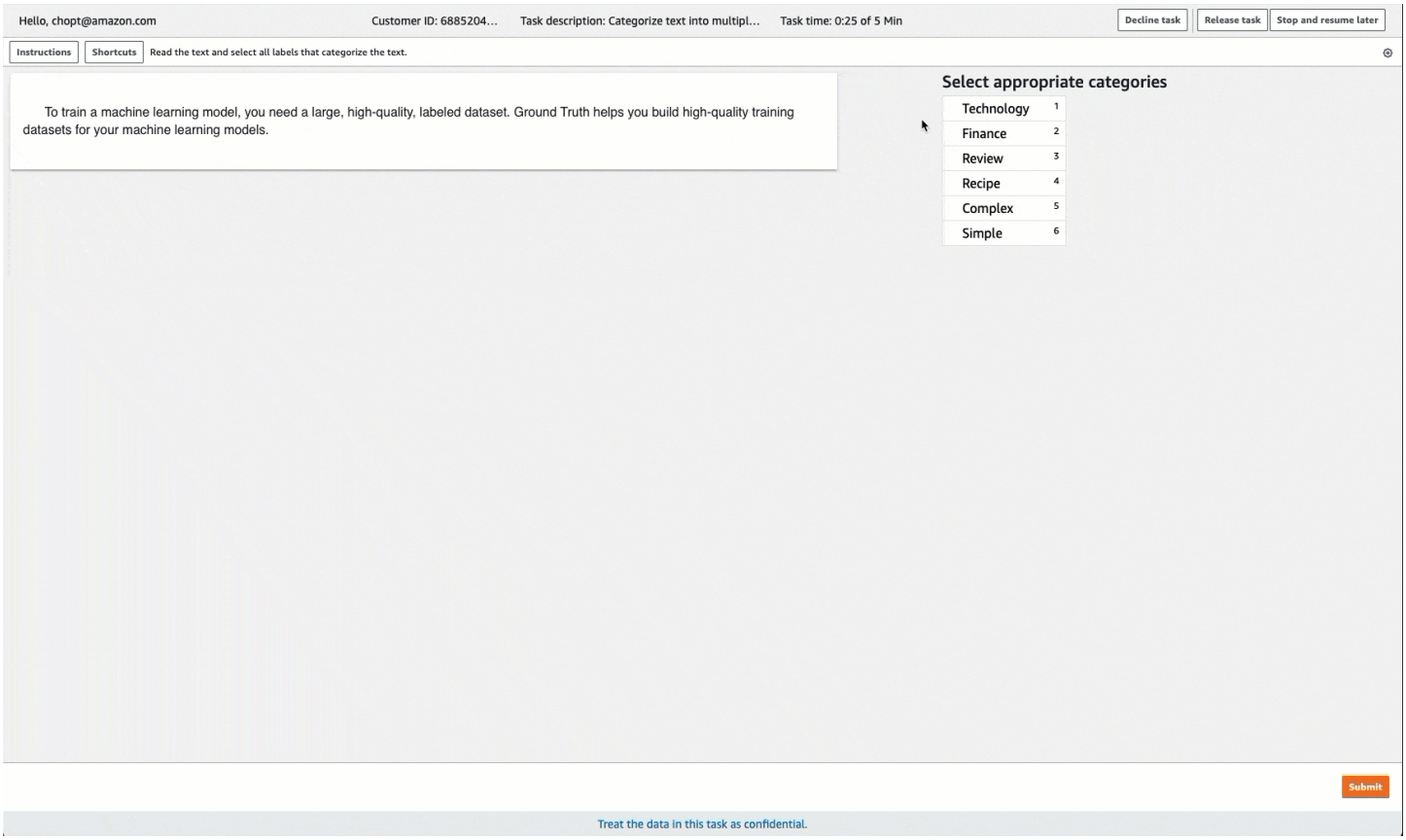
Untuk membatasi pekerja memilih label tunggal untuk setiap dokumen atau pemilihan teks, gunakanKlasifikasi Teks (Label Tunggal)jenis tugas.

> **Important**
>
> Jika Anda membuat file manifes masukan secara manual, gunakan`"source"`untuk mengidentifikasi teks yang ingin diberi label. Untuk informasi selengkapnya, lihat Input Data.

Membuat Job Pelabelan Klasifikasi Teks Multi-Label (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan klasifikasi teks multi-label di Amazon SageMaker konsol. Pada Langkah 10, pilihTextdariKategori tugasdrop down menu, dan memilihKlasifikasi Teks (Multi-label)sebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan dengan konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan label yang dapat dipilih pekerja.



Membuat Job Pelabelan Klasifikasi Teks Multi-Label (API)

Untuk membuat tugas pelabelan klasifikasi teks multi-label, gunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK.
Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian`CreateLabelingJob`.

Ikuti petunjuk diBuat Job Pelabelan (API)dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Pra-anotasi fungsi Lambda untuk jenis tugas ini diakhiri dengan`PRE-TextMultiClassMultiLabel`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn.
- Fungsi Lambda anotasi-konsolidasi untuk jenis tugas ini diakhiri dengan`ACS-TextMultiClassMultiLabel`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihatAnnotationConsolidationLambdaArn.

Berikut ini adalah contohAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di Wilayah US East (N. Virginia). Semua parameter berwarna merah harus diganti dengan spesifikasi dan sumber daya Anda.

```
response = client.create_labeling_job(
    LabelingJobName='example-multi-label-text-classification-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
```

```
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/custom-worker-task-template.html'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda::function:PRE-
TextMultiClassMultiLabel,
        'TaskKeywords': [
            'Text Classification',
        ],
        'TaskTitle': 'Multi-label text classification task',
        'TaskDescription': 'Select all labels that apply to the text shown',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-TextMultiClassMultiLabel'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Membuat Template untuk Klasifikasi Teks Multi-label

Jika Anda membuat pekerjaan pelabelan menggunakan API, Anda harus menyediakan template tugas pekerja di`UiTemplateS3Uri`. Salin dan modifikasi template berikut. Hanya mengubah`short-instructions`,`full-instructions`, dan`header`.

Unggah template ini ke S3, dan berikan URI S3 untuk file ini`UiTemplateS3Uri`.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
```

```
  <crowd-classifier-multi-select
    name="crowd-classifier-multi-select"
    categories="{{ task.input.labels | to_json | escape }}"
    header="Please identify all classes in the below text"
  >
    <classification-target style="white-space: pre-wrap">
      {{ task.input.taskObject }}
    </classification-target>
    <full-instructions header="Classifier instructions">
      <ol><li><strong>Read</strong> the text carefully.</li>
      <li><strong>Read</strong> the examples to understand more about the options.</li>
      <li><strong>Choose</strong> the appropriate labels that best suit the text.</
li></ol>
    </full-instructions>
    <short-instructions>
      <p>Enter description of the labels that workers have to choose from</p>
      <p><br></p>
      <p><br></p><p>Add examples to help workers understand the label</p>
      <p><br></p><p><br></p><p><br></p><p><br></p><p><br></p>
    </short-instructions>
  </crowd-classifier-multi-select>
  </crowd-form>
```

Untuk mempelajari cara membuat template kustom, lihatMembuat Alur Kerja Pelabelan Kustom.

Data Keluaran Klasifikasi Teks Multi-label

Setelah Anda membuat tugas pelabelan klasifikasi teks multi-label, data keluaran Anda akan ditempatkan di bucket Amazon S3 yang ditentukan dalam`S3OutputPath`parameter saat menggunakan API atau di`Lokasi set data output`bidang`Gambaran umum Job`bagian konsol.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Untuk melihat contoh file manifes keluaran untuk pekerjaan pelabelan klasifikasi teks multi-label, lihatOutput Job Klasifikasi Multi-label.

Label Video dan Bingkai Video

Anda dapat menggunakan Ground Truth untuk mengklasifikasikan video dan membuat anotasi bingkai video (gambar diam yang diekstrak dari video) menggunakan salah satu dari tiga jenis tugas video bawaan. Jenis tugas ini merampingkan proses pembuatan pekerjaan pelabelan bingkai video dan video menggunakan Amazon SageMaker konsol, API, dan SDK khusus bahasa.

- Klasifikasi klip video - Memungkinkan pekerja untuk mengklasifikasikan video ke dalam kategori yang Anda tentukan. Misalnya, Anda dapat menggunakan jenis tugas ini agar pekerja mengkategorikan video ke dalam topik seperti olahraga, komedi, musik, dan pendidikan. Untuk mempelajari selengkapnya, lihat Klasifikasi video.
- Pekerjaan pelabelan bingkai video - Memungkinkan pekerja untuk membuat anotasi bingkai video yang diekstrak dari video menggunakan kotak pembatas, polyline, poligon, atau alat anotasi keypoint. Ground Truth menawarkan dua tipe tugas bawaan untuk memberi label pada bingkai video:
  - Deteksi objek bingkai video: Memungkinkan pekerja untuk mengidentifikasi dan menemukan objek dalam bingkai video.
  - Pelacakan objek bingkai video: Memungkinkan pekerja untuk melacak pergerakan objek di seluruh frame video.
  - Pekerjaan penyesuaian bingkai video: Minta pekerja menyesuaikan label, atribut kategori label, dan atribut bingkai dari deteksi objek bingkai video sebelumnya atau pekerjaan pelabelan pelacakan objek.
  - Lowongan kerja verifikasi bingkai video: Minta pekerja memverifikasi label, atribut kategori label, dan atribut bingkai dari deteksi objek bingkai video sebelumnya atau pekerjaan pelabelan pelacakan objek.

  Jika Anda memiliki file video, Anda dapat menggunakan alat ekstraksi bingkai otomatis Ground Truth untuk mengekstrak bingkai video dari video Anda. Untuk mempelajari selengkapnya, lihat Data Input Bingkai Video.

> **Tip**
>
> Untuk mempelajari lebih lanjut tentang jenis file yang didukung dan kuota data input, lihatInput Data.

Topik

- Klasifikasi video
- Label Bingkai Video
- Instruksi Pekerja

Klasifikasi video

Gunakan Amazon SageMaker Tugas pelabelan klasifikasi video Ground Truth saat Anda membutuhkan pekerja untuk mengklasifikasikan video menggunakan label yang telah ditentukan sebelumnya yang Anda tentukan. Pekerja ditampilkan video dan diminta untuk memilih satu label untuk setiap video.

Anda membuat pekerjaan pelabelan klasifikasi video menggunakan bagian Ground Truth di Amazon SageMaker konsol atauCreateLabelingJoboperasi.

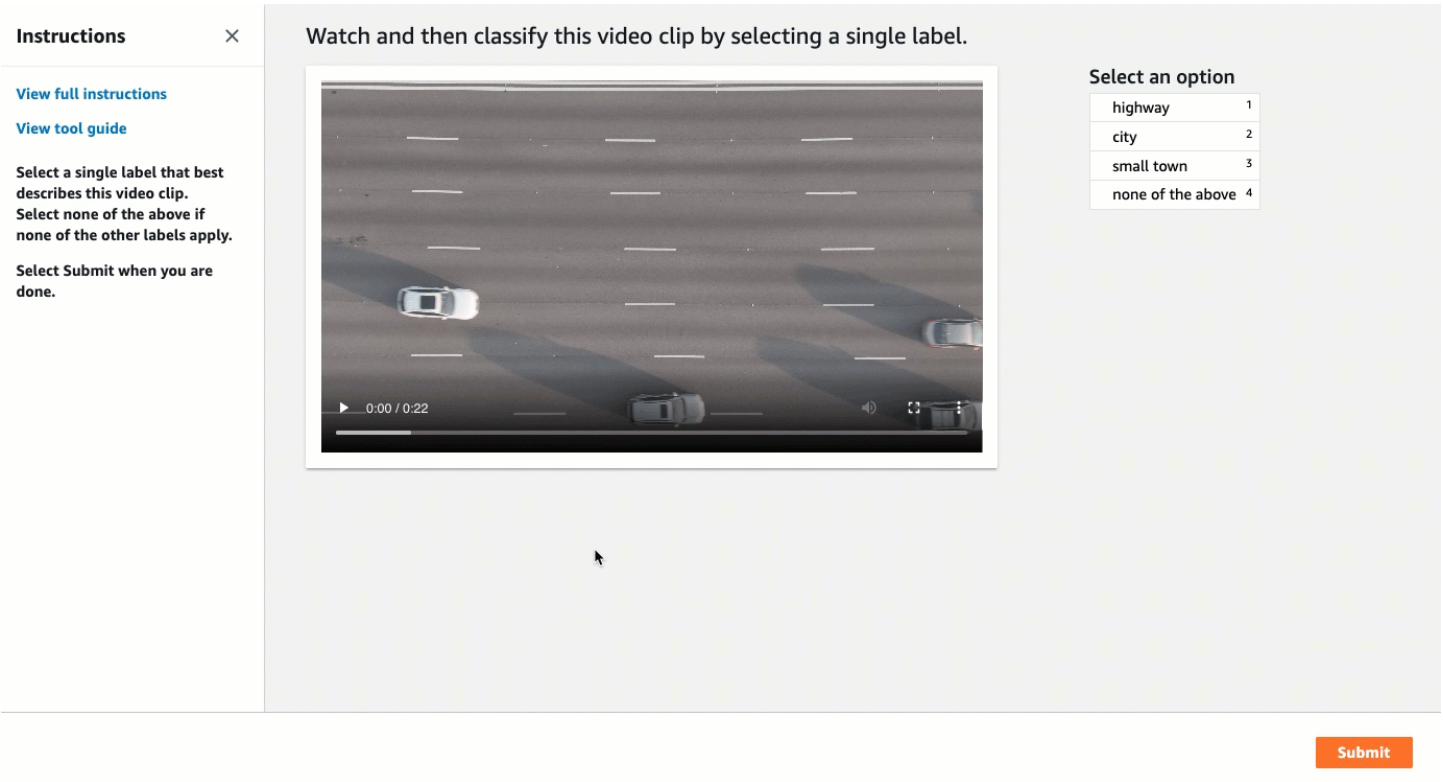
File video Anda harus dikodekan dalam format yang didukung oleh browser yang digunakan oleh tim kerja yang memberi label pada data Anda. Dianjurkan agar Anda memverifikasi bahwa semua format file video dalam file manifes masukan ditampilkan dengan benar menggunakan pratinjau UI pekerja. Anda dapat mengkomunikasikan browser yang didukung kepada pekerja Anda menggunakan instruksi pekerja. Untuk melihat format file yang didukung, lihatFormat Data yang Didukung.

> **Important**
>
> Untuk jenis tugas ini, jika Anda membuat file manifes Anda sendiri, gunakan`"source-ref"`untuk mengidentifikasi lokasi dari setiap file video di Amazon S3 yang ingin Anda beri label. Untuk informasi selengkapnya, lihat Input Data.

Membuat Job Pelabelan Klasifikasi Video (Konsol)

Anda bisa mengikuti petunjuk dalamMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan klasifikasi video di SageMaker konsol. Pada langkah 10, pilihVideodariKategori tugasdaftar dropdown, dan pilihKlasifikasi videosebagai jenis tugas.

Ground Truth menyediakan UI pekerja yang mirip dengan berikut ini untuk tugas pelabelan. Saat Anda membuat pekerjaan pelabelan di konsol, Anda menentukan instruksi untuk membantu pekerja menyelesaikan pekerjaan dan label yang dapat dipilih pekerja.



Membuat Job Pelabelan Klasifikasi Video (API)

Bagian ini mencakup detail yang perlu Anda ketahui saat membuat pekerjaan pelabelan menggunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian dari`CreateLabelingJob`.

Ikuti petunjuk diBuat Job Pelabelan (API)dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Gunakan fungsi pra-anotasi Lambda yang diakhiri dengan`PRE-VideoClassification`. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn.
- Gunakan anotasi-konsolidasi fungsi Lambda yang diakhiri dengan`ACS-VideoClassification`. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihatAnnotationConsolidationLambdaArn.

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di Wilayah US East (N. Virginia).

```
response = client.create_labeling_job(
```

```
    LabelingJobName='example-video-classification-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:region:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/worker-task-template.html'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
VideoClassification',
        'TaskKeywords': [
            'Video Classification',
        ],
        'TaskTitle': 'Video classification task',
        'TaskDescription': 'Select a label to classify this video',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-VideoClassification'
        },
```

```
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Menyediakan Template untuk Klasifikasi Video

Jika Anda membuat pekerjaan pelabelan menggunakan API, Anda harus menyediakan template tugas pekerja`UiTemplateS3Uri`. Salin dan modifikasi template berikut dengan memodifikasi`short-instructions`,`full-instructions`, dan`header`. Unggah template ini ke Amazon S3, dan berikan URI Amazon S3 ke file ini`UiTemplateS3Uri`.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

              <crowd-form>
                  <crowd-classifier
                    name="crowd-classifier"
                    categories="{{ task.input.labels | to_json | escape }}"
                    header="Please classify video"
                  >
                    <classification-target>
                       <video width="100%" controls/>
                        <source src="{{ task.input.taskObject | grant_read_access }}"
 type="video/mp4"/>
                        <source src="{{ task.input.taskObject | grant_read_access }}"
 type="video/webm"/>
                        <source src="{{ task.input.taskObject | grant_read_access }}"
 type="video/ogg"/>
                      Your browser does not support the video tag.
                      </video>
                    </classification-target>
                    <full-instructions header="Video classification instructions">
                      <ol><li><strong>Read</strong> the task carefully and inspect the
 video.</li>
                        <li><strong>Read</strong> the options and review the examples
 provided to understand more about the labels.</li>
                        <li><strong>Choose</strong> the appropriate label that best
 suits the video.</li></ol>
                    </full-instructions>
                    <short-instructions>
```

```
                    <h3><span style="color: rgb(0, 138, 0);">Good example</span></h3>
                      <p>Enter description to explain the correct label to the
 workers</p>
                      <p><img src="https://d7evko5405gb7.cloudfront.net/
fe4fed9b-660c-4477-9294-2c66a15d6bbe/src/images/quick-instructions-example-
placeholder.png" style="max-width:100%"></p>
                      <h3><span style="color: rgb(230, 0, 0);">Bad example</span></
h3>
                      <p>Enter description of an incorrect label</p>
                      <p><img src="https://d7evko5405gb7.cloudfront.net/
fe4fed9b-660c-4477-9294-2c66a15d6bbe/src/images/quick-instructions-example-
placeholder.png" style="max-width:100%"></p>
                    </short-instructions>
                  </crowd-classifier>
              </crowd-form>
```

Data Output Klasifikasi Video

Setelah Anda membuat tugas pelabelan klasifikasi video, data keluaran Anda terletak di bucket Amazon S3 yang ditentukan dalam`S3OutputPath`parameter saat menggunakan API atau diLokasi set data outputbidangGambaran umum Jobbagian dari konsol.

Untuk mempelajari lebih lanjut tentang file manifes keluaran yang dihasilkan oleh Ground Truth dan struktur file yang digunakan Ground Truth untuk menyimpan data keluaran Anda, lihatData Output.

Untuk melihat contoh file manifes keluaran untuk tugas pelabelan klasifikasi video, lihatOutput.

Label Bingkai Video

Anda dapat menggunakan jenis tugas bingkai video bawaan Ground Truth agar pekerja membuat anotasi bingkai video menggunakan kotak pembatas, polyline, poligon, atau titik kunci. SEBUAHbingkai videoadalah urutan gambar yang telah diekstrak dari video.

Jika Anda tidak memiliki bingkai video, Anda dapat menyediakan file video (file MP4) dan menggunakan alat ekstraksi bingkai otomatis Ground Truth untuk mengekstrak bingkai video. Untuk mempelajari selengkapnya, lihat Menyediakan File Video.

Anda dapat menggunakan jenis tugas video bawaan berikut untuk membuat pekerjaan pelabelan bingkai video menggunakan Amazon SageMaker konsol, API, dan SDK khusus bahasa.

- Deteksi objek bingkai video- Gunakan jenis tugas ini saat Anda ingin pekerja mengidentifikasi dan menemukan objek dalam urutan bingkai video. Anda menyediakan daftar kategori, dan

pekerja dapat memilih satu kategori pada satu waktu dan membubuhi anotasi objek yang kategori berlaku di semua frame. Misalnya, Anda dapat menggunakan tugas ini untuk meminta pekerja mengidentifikasi dan melokalisasi berbagai objek dalam sebuah adegan, seperti mobil, sepeda, dan pejalan kaki.

- Pelacakan objek bingkai video- Gunakan jenis tugas ini ketika Anda ingin pekerja untuk melacak pergerakan contoh objek di seluruh urutan frame video. Saat pekerja menambahkan anotasi ke satu frame, anotasi tersebut dikaitkan dengan ID instance unik. Pekerja menambahkan anotasi yang terkait dengan ID yang sama di semua frame lain untuk mengidentifikasi objek atau orang yang sama. Misalnya, pekerja dapat melacak pergerakan kendaraan melintasi urutan bingkai video dengan menggambar kotak pembatas yang terkait dengan ID yang sama di sekitar kendaraan di setiap bingkai yang muncul.

Gunakan topik berikut untuk mempelajari lebih lanjut tentang jenis tugas bawaan ini dan cara membuat pekerjaan pelabelan menggunakan setiap jenis tugas. LihatJenisuntuk mempelajari lebih lanjut tentang alat anotasi (kotak pembatas, polyline, poligon, dan keypoint) yang tersedia untuk jenis tugas ini.

Sebelum membuat tugas pelabelan, kami merekomendasikan agar Anda meninjauIkhtisar Job Pelabelan Bingkai Video.

Topik

- Deteksi Objek Bingkai Video
- Pelacakan Objek Bingkai Video
- Ikhtisar Job Pelabelan Bingkai Video

Deteksi Objek Bingkai Video

Anda dapat menggunakan jenis tugas deteksi objek bingkai video agar pekerja mengidentifikasi dan menemukan objek dalam urutan bingkai video (gambar yang diekstrak dari video) menggunakan kotak pembatas, polyline, poligon, atau keypointalat penjelasan. Alat yang Anda pilih menentukan jenis tugas bingkai video yang Anda buat. Misalnya, Anda dapat menggunakan pekerja tipe tugas deteksi objek bingkai video kotak pembatas untuk mengidentifikasi dan melokalisasi berbagai objek dalam serangkaian bingkai video, seperti mobil, sepeda, dan pejalan kaki.

Anda dapat membuat pekerjaan pelabelan deteksi objek bingkai video menggunakan Amazon SageMaker Ground Truth konsol, yang SageMaker APIAWSSDK. Untuk mempelajari lebih lanjut, lihatBuat Job Pelabelan Deteksi Objek Bingkai Videodan pilih metode pilihan Anda. LihatJenisuntuk

mempelajari lebih lanjut tentang alat anotasi yang dapat Anda pilih saat membuat pekerjaan pelabelan.

Ground Truth menyediakan UI pekerja dan alat untuk menyelesaikan tugas pekerjaan pelabelan Anda:Pratinjau UI Pekerja.

Anda dapat membuat pekerjaan untuk menyesuaikan anotasi yang dibuat dalam pekerjaan pelabelan deteksi objek video menggunakan jenis tugas penyesuaian deteksi objek video. Untuk mempelajari selengkapnya, lihat Buat Penyesuaian Deteksi Objek Bingkai Video atau Job Pelabelan Verifikasi.

Pratinjau UI Pekerja

Ground Truth menyediakan pekerja dengan antarmuka pengguna web (UI) untuk menyelesaikan tugas penjelasan deteksi objek bingkai video Anda. Anda dapat melihat pratinjau dan berinteraksi dengan UI pekerja saat membuat pekerjaan pelabelan di konsol. Jika Anda adalah pengguna baru, sebaiknya buat tugas pelabelan melalui konsol menggunakan kumpulan data input kecil untuk melihat pratinjau UI pekerja dan memastikan bingkai video, label, dan atribut label Anda muncul seperti yang diharapkan.

UI memberi pekerja alat pelabelan bantu berikut untuk menyelesaikan tugas deteksi objek Anda:

- Untuk semua tugas, pekerja dapat menggunakanSalin ke selanjutnyadanSalin ke semuafitur untuk menyalin anotasi ke frame berikutnya atau ke semua frame berikutnya masing-masing.
- Untuk tugas-tugas yang mencakup alat kotak pembatas, pekerja dapat menggunakanPrediksi selanjutnyafitur untuk menggambar kotak pembatas dalam satu bingkai, dan kemudian memiliki Ground Truth memprediksi lokasi kotak dengan label yang sama di semua frame lainnya. Pekerja kemudian dapat melakukan penyesuaian untuk memperbaiki lokasi kotak yang diprediksi.

Buat Job Pelabelan Deteksi Objek Bingkai Video

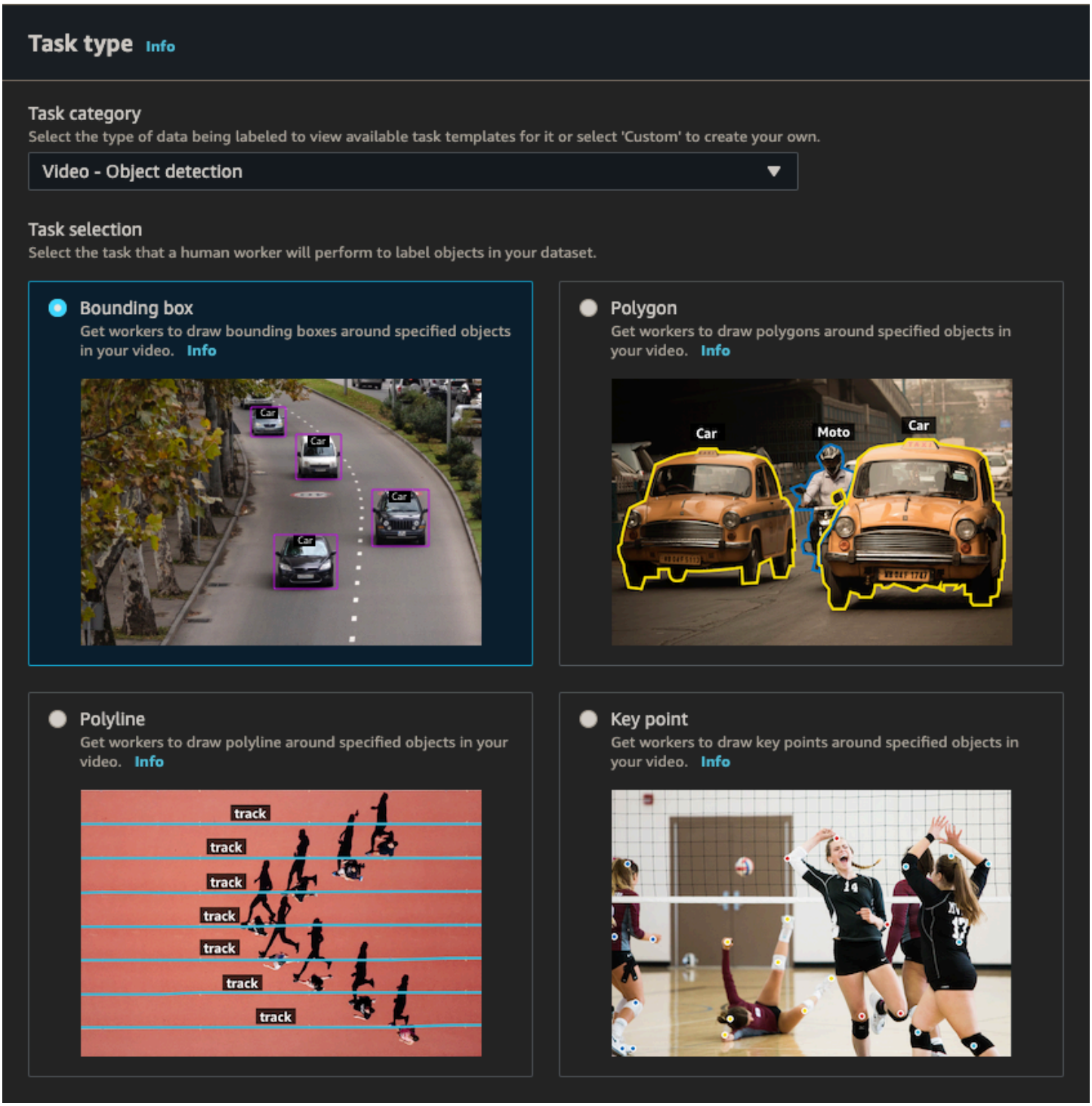
Anda dapat membuat pekerjaan pelabelan deteksi objek bingkai video menggunakan SageMaker konsol`CreateLabelingJob`Operasi API

Bagian ini mengasumsikan Anda sudah meninjauIkhtisar Job Pelabelan Bingkai Videodan telah memilih jenis data input dan koneksi dataset input yang Anda gunakan.

Buat Job Pelabelan (Konsol)

Anda bisa mengikuti petunjuk dalamMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelacakan objek bingkai video di SageMaker konsol. Pada langkah 10,

pilihDeteksi objekdariKategori tugasdaftar dropdown. Pilih jenis tugas yang Anda inginkan dengan memilih salah satu kartu diPemilihan tugas.



Buat Job Pelabelan (API)

Anda membuat pekerjaan pelabelan deteksi objek menggunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK.

Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat JugabagianCreateLabelingJob.

Buat Job Pelabelan (API)memberikan ikhtisar tentangCreateLabelingJoboperasi. Ikuti petunjuk ini dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Anda harus memasukkan ARN untukHumanTaskUiArn. Gunakan `arn:aws:sagemaker:<region>:394669845002:human-task-ui/VideoObjectDetection`. Ganti*<region>*denganAWSWilayah tempat Anda membuat tugas pelabelan.

  Jangan sertakan entri untukUiTemplateS3Uriparameter.
- KlasterLabelAttributeNameharus diakhiri`-ref`. Sebagai contoh, *`video-od-labels`*`-ref`.
- File manifes masukan Anda harus berupa file manifes urutan bingkai video. Anda dapat membuat file manifes ini menggunakan SageMaker konsol, atau buat secara manual dan meng-upload ke Amazon S3. Untuk informasi selengkapnya, lihat Pengaturan data.
- Anda hanya dapat menggunakan tim kerja pribadi atau vendor untuk membuat pekerjaan pelabelan deteksi objek bingkai video.
- Anda menentukan label, kategori label dan atribut bingkai, jenis tugas, dan instruksi pekerja dalam file konfigurasi kategori label. Tentukan jenis tugas (kotak pembatas, polylines, poligon atau keypoint) menggunakanannotationTypedalam file konfigurasi kategori label Anda. Untuk informasi selengkapnya, lihatBuat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkaiuntuk mempelajari cara membuat file ini.
- Anda perlu menyediakan ARN yang telah ditentukan sebelumnya untuk fungsi Lambda pra-anotasi dan pasca-anotasi (ACS). ARN ini khusus untukAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda.
  - Untuk menemukan pra-anotasi Lambda ARN, lihatPreHumanTaskLambdaArn. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri denganPRE-VideoObjectDetection.
  - Untuk menemukan pasca-anotasi Lambda ARN, lihatAnnotationConsolidationLambdaArn. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri denganACS-VideoObjectDetection.
- Jumlah pekerja yang ditentukan dalamNumberOfHumanWorkersPerDataObjectharus1.
- Pelabelan data otomatis tidak didukung untuk pekerjaan pelabelan bingkai video. Jangan tentukan nilai untuk parameter diLabelingJobAlgorithmsConfig.

- Pekerjaan pelabelan pelacakan objek bingkai video dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini di`TaskTimeLimitInSeconds`(hingga 7 hari, atau 604.800 detik).

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di US East (N. Virginia).

```
response = client.create_labeling_job(
    LabelingJobName='example-video-od-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://DOC-EXAMPLE-BUCKET/path/video-frame-sequence-
input-manifest.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://DOC-EXAMPLE-BUCKET/prefix/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/prefix/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:us-east-1:*:workteam/private-crowd/*',
        'UiConfig': {
            'HumanTaskUiArn: 'arn:aws:sagemaker:us-east-1:394669845002:human-task-ui/
VideoObjectDetection'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
VideoObjectDetection',
        'TaskKeywords': [
```

```
            'Video Frame Object Detection',
        ],
        'TaskTitle': 'Video frame object detection task',
        'TaskDescription': 'Classify and identify the location of objects and people in
 video frames',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-VideoObjectDetection'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Buat Penyesuaian Deteksi Objek Bingkai Video atau Job Pelabelan Verifikasi

Anda dapat membuat pekerjaan pelabelan penyesuaian dan verifikasi menggunakan konsol Ground Truth atau`CreateLabelingJob`API Untuk mempelajari lebih lanjut tentang penyesuaian dan verifikasi pekerjaan pelabelan, dan untuk mempelajari cara membuatnya, lihatVerifikasi dan Sesuaikan Label.

Format Data Output

Saat Anda membuat pekerjaan pelabelan deteksi objek bingkai video, tugas dikirim ke pekerja. Ketika pekerja ini menyelesaikan tugas mereka, label ditulis ke lokasi keluaran Amazon S3 yang Anda tentukan saat Anda membuat pekerjaan pelabelan. Untuk mempelajari tentang format data output deteksi objek bingkai video, lihatOutput Deteksi Objek Bingkai Video. Jika Anda adalah pengguna baru Ground Truth, lihatData Outputuntuk mempelajari lebih lanjut tentang format data keluaran Ground Truth.

Pelacakan Objek Bingkai Video

Anda dapat menggunakan jenis tugas pelacakan objek bingkai video agar pekerja melacak pergerakan objek dalam urutan bingkai video (gambar yang diekstrak dari video) menggunakan kotak pembatas, polyline, poligon, atau keypointalat penjelasan. Alat yang Anda pilih menentukan jenis

tugas bingkai video yang Anda buat. Misalnya, Anda dapat menggunakan jenis tugas pelacakan objek bingkai video kotak pembatas untuk meminta pekerja melacak pergerakan objek, seperti mobil, sepeda, dan pejalan kaki dengan menggambar kotak di sekitarnya.

Anda menyediakan daftar kategori, dan setiap anotasi yang ditambahkan pekerja ke bingkai video diidentifikasi sebagaicontohdari kategori itu menggunakan ID instance. Misalnya, jika Anda memberikan mobil kategori label, mobil pertama yang dianotasi pekerja akan memiliki contoh ID mobil: 1. Mobil kedua yang dianotasi pekerja akan memiliki contoh ID car:2. Untuk melacak pergerakan objek, pekerja menambahkan anotasi yang terkait dengan ID instance yang sama di sekitar objek di semua frame.

Anda dapat membuat pekerjaan pelabelan pelacakan objek bingkai video menggunakan Amazon SageMaker Ground Truth konsol, yang SageMaker APIAWSSDK. Untuk mempelajari lebih lanjut, lihatBuat Job Pelabelan Deteksi Objek Bingkai Videodan pilih metode pilihan Anda. LihatJenisuntuk mempelajari lebih lanjut tentang alat anotasi yang dapat Anda pilih saat membuat pekerjaan pelabelan.

Ground Truth menyediakan UI pekerja dan alat untuk menyelesaikan tugas pekerjaan pelabelan Anda:Pratinjau UI Pekerja.

Anda dapat membuat pekerjaan untuk menyesuaikan anotasi yang dibuat dalam pekerjaan pelabelan deteksi objek video menggunakan jenis tugas penyesuaian deteksi objek video. Untuk mempelajari selengkapnya, lihat Buat Penyesuaian Deteksi Objek Bingkai Video atau Job Pelabelan Verifikasi.

Pratinjau UI Pekerja

Ground Truth menyediakan pekerja dengan antarmuka pengguna web (UI) untuk menyelesaikan tugas penjelasan pelacakan objek bingkai video Anda. Anda dapat melihat pratinjau dan berinteraksi dengan UI pekerja saat membuat pekerjaan pelabelan di konsol. Jika Anda adalah pengguna baru, sebaiknya buat tugas pelabelan melalui konsol menggunakan kumpulan data input kecil untuk melihat pratinjau UI pekerja dan memastikan bingkai video, label, dan atribut label Anda muncul seperti yang diharapkan.

UI memberi pekerja alat pelabelan bantu berikut untuk menyelesaikan tugas pelacakan objek Anda:

- Untuk semua tugas, pekerja dapat menggunakanSalin ke selanjutnyadanSalin ke semuafitur untuk menyalin anotasi dengan ID unik yang sama ke frame berikutnya atau ke semua frame berikutnya masing-masing.
- Untuk tugas-tugas yang mencakup alat kotak pembatas, pekerja dapat menggunakanPrediksi selanjutnyafitur untuk menggambar kotak pembatas dalam satu bingkai, dan kemudian memiliki

Ground Truth memprediksi lokasi kotak dengan ID unik yang sama di semua frame lainnya. Pekerja kemudian dapat melakukan penyesuaian untuk memperbaiki lokasi kotak yang diprediksi.

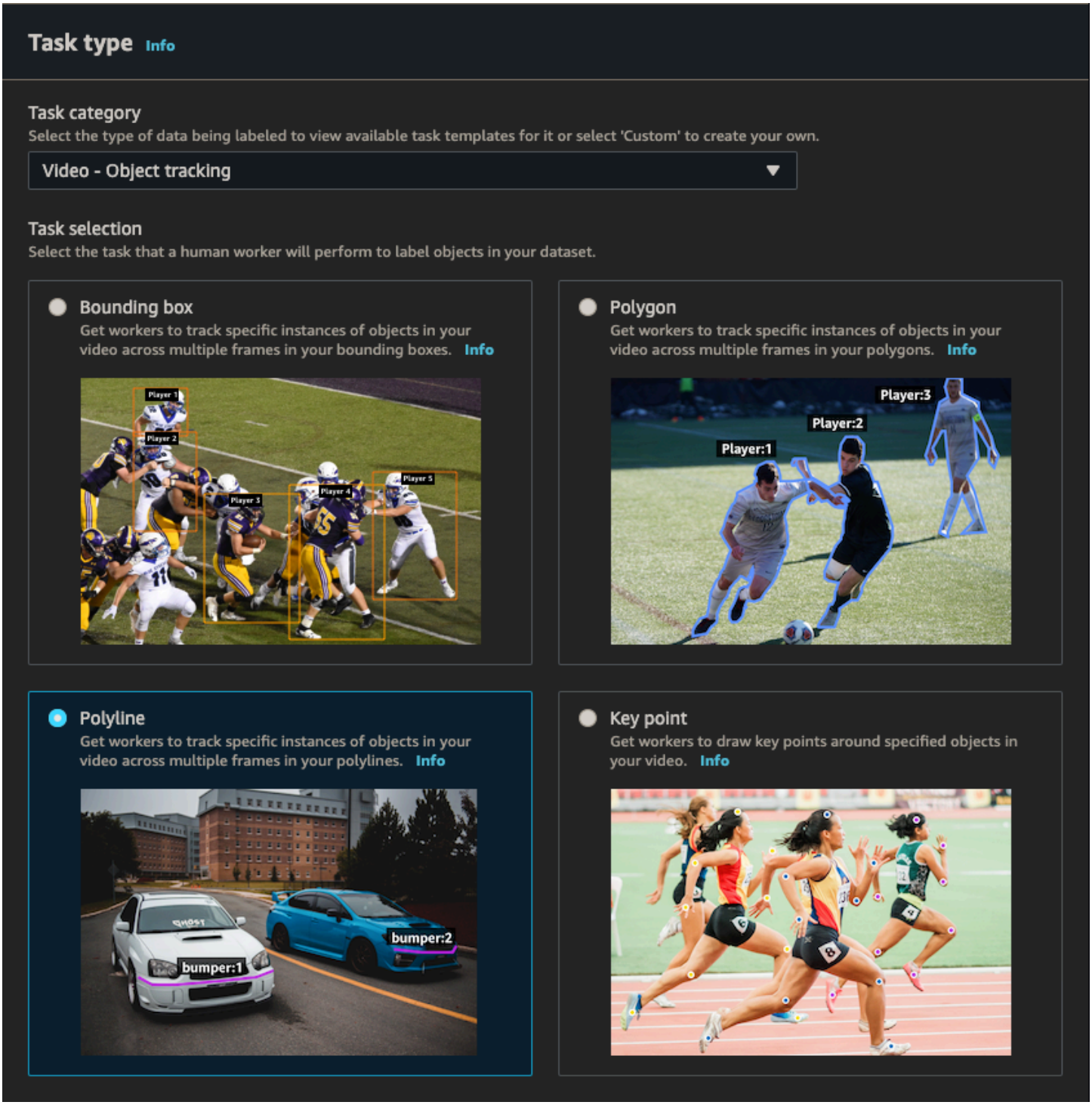
Buat Job Pelabelan Pelacakan Objek Bingkai Video

Anda dapat membuat pekerjaan pelabelan pelacakan objek bingkai video menggunakan SageMaker konsol`CreateLabelingJob`Operasi API

Bagian ini mengasumsikan Anda sudah meninjauIkhtisar Job Pelabelan Bingkai Videodan telah memilih jenis data input dan koneksi dataset input yang Anda gunakan.

Buat Job Pelabelan (Konsol)

Anda bisa mengikuti petunjuk dalamMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelacakan objek bingkai video di SageMaker konsol. Pada langkah 10, pilihVideo - Pelacakan objekdariKategori tugasdaftar dropdown. Pilih jenis tugas yang Anda inginkan dengan memilih salah satu kartu diPemilihan tugas.



Buat Job Pelabelan (API)

Anda membuat pekerjaan pelabelan pelacakan objek menggunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian`CreateLabelingJob`.

Buat Job Pelabelan (API)memberikan ikhtisar tentangCreateLabelingJoboperasi. Ikuti petunjuk ini dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Anda harus memasukkan ARN untukHumanTaskUiArn. Gunakan `arn:aws:sagemaker:<region>:394669845002:human-task-ui/VideoObjectTracking`. Ganti*<region>*denganAWSWilayah tempat Anda membuat tugas pelabelan.

  Jangan sertakan entri untukUiTemplateS3Uriparameter.
- KlasterLabelAttributeNameharus diakhiri`-ref`. Sebagai contoh, *`ot-labels`*`-ref`.
- File manifes masukan Anda harus berupa file manifes urutan bingkai video. Anda dapat membuat file manifes ini menggunakan SageMaker konsol, atau buat secara manual dan meng-upload ke Amazon S3. Untuk informasi selengkapnya, lihat Pengaturan data. Jika Anda membuat tugas pelabelan streaming, file manifes masukan bersifat opsional.
- Anda hanya dapat menggunakan tim kerja pribadi atau vendor untuk membuat pekerjaan pelabelan deteksi objek bingkai video.
- Anda menentukan label, kategori label dan atribut bingkai, jenis tugas, dan instruksi pekerja dalam file konfigurasi kategori label. Tentukan jenis tugas (kotak pembatas, polylines, poligon atau keypoint) menggunakanannotationTypedalam file konfigurasi kategori label Anda. Untuk informasi selengkapnya, lihatBuat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkaiuntuk mempelajari cara membuat file ini.
- Anda perlu menyediakan ARN yang telah ditentukan sebelumnya untuk fungsi Lambda pra-anotasi dan pasca-anotasi (ACS). ARN ini khusus untukAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda.
  - Untuk menemukan pra-anotasi Lambda ARN, lihatPreHumanTaskLambdaArn. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri dengan`PRE-VideoObjectTracking`.
  - Untuk menemukan pasca-anotasi Lambda ARN, lihatAnnotationConsolidationLambdaArn. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri dengan`ACS-VideoObjectTracking`.
- Jumlah pekerja yang ditentukan dalam`NumberOfHumanWorkersPerDataObject`harus`1`.
- Pelabelan data otomatis tidak didukung untuk pekerjaan pelabelan bingkai video. Jangan tentukan nilai untuk parameter diLabelingJobAlgorithmsConfig.

- Pekerjaan pelabelan pelacakan objek bingkai video dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini di`TaskTimeLimitInSeconds`(hingga 7 hari, atau 604.800 detik).

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan di US East (N. Virginia).

```
response = client.create_labeling_job(
    LabelingJobName='example-video-ot-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://DOC-EXAMPLE-BUCKET/path/video-frame-sequence-
input-manifest.json'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://DOC-EXAMPLE-BUCKET/prefix/file-to-store-output-data',
        'KmsKeyId': 'string'
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/prefix/label-categories.json',
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:us-east-1:*:workteam/private-crowd/*',
        'UiConfig': {
            'HumanTaskUiArn: 'arn:aws:sagemaker:us-east-1:394669845002:human-task-ui/
VideoObjectTracking'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-east-1:432418664414:function:PRE-
VideoObjectTracking',
        'TaskKeywords': [
```

```
            'Video Frame Object Tracking',
        ],
        'TaskTitle': 'Video frame object tracking task',
        'TaskDescription': Tracking the location of objects and people across video
 frames',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-VideoObjectTracking'
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Buat Penyesuaian Pelacakan Objek Bingkai Video atau Job Pelabelan Verifikasi

Anda dapat membuat pekerjaan pelabelan penyesuaian dan verifikasi menggunakan konsol Ground Truth atau`CreateLabelingJob`API Untuk mempelajari lebih lanjut tentang penyesuaian dan verifikasi pekerjaan pelabelan, dan untuk mempelajari cara membuatnya, lihatVerifikasi dan Sesuaikan Label.

Format Data Output

Saat Anda membuat pekerjaan pelabelan pelacakan objek bingkai video, tugas dikirim ke pekerja. Ketika pekerja ini menyelesaikan tugas mereka, label ditulis ke lokasi keluaran Amazon S3 yang Anda tentukan saat Anda membuat pekerjaan pelabelan. Untuk mempelajari tentang format data keluaran pelacakan objek bingkai video, lihatOutput Pelacakan Objek Bingkai Video. Jika Anda adalah pengguna baru Ground Truth, lihatData Outputuntuk mempelajari lebih lanjut tentang format data keluaran Ground Truth.

Ikhtisar Job Pelabelan Bingkai Video

Gunakan halaman ini untuk mempelajari tentang deteksi objek dan pelacakan objek pekerjaan pelabelan bingkai video. Informasi di halaman ini berlaku untuk kedua jenis tugas bawaan ini.

Pekerjaan pelabelan bingkai video unik karena hal-hal berikut:

- Anda dapat menyediakan objek data yang siap untuk dianotasi (bingkai video), atau Anda dapat menyediakan file video dan memiliki Ground Truth secara otomatis mengekstrak bingkai video.
- Pekerja memiliki kemampuan untuk menyelamatkan pekerjaan saat mereka pergi.
- Anda tidak dapat menggunakanAmazon Mechanical Turktenaga kerja untuk menyelesaikan tugas pelabelan Anda.
- Ground Truth menyediakan UI pekerja, serta alat bantu dan pelabelan dasar, untuk membantu pekerja menyelesaikan tugas Anda. Anda tidak perlu menyediakan templat tugas pekerja.

Gunakan topik berikut untuk mempelajari lebih banyak.

Topik

- Data Input
- Job Penyelesaian
- Jenis
- Tenaga kerja
- Antarmuka Pengguna Pekerja (UI)
- Persyaratan Izin Job Bingkai Video

Data Input

Pekerjaan pelabelan bingkai video menggunakanurutanframe video. Urutan tunggal adalah serangkaian gambar yang telah diekstraksi dari satu video. Anda dapat memberikan urutan bingkai video Anda sendiri, atau memiliki Ground Truth secara otomatis mengekstrak urutan bingkai video dari file video Anda. Untuk mempelajari selengkapnya, lihat Menyediakan File Video.

Ground Truth menggunakan file urutan untuk mengidentifikasi semua gambar dalam satu urutan. Semua urutan yang ingin Anda sertakan dalam tugas pelabelan tunggal diidentifikasi dalam file manifes masukan. Setiap urutan digunakan untuk membuat tugas pekerja tunggal. Anda dapat secara otomatis membuat file urutan dan file manifes masukan menggunakan pengaturan data otomatis Ground Truth. Untuk mempelajari selengkapnya, lihat Pengaturan Data Input Bingkai Video Otomatis.

Untuk mempelajari cara membuat file urutan dan file manifes masukan secara manual, lihatMembuat File Manifes Masukan Bingkai Video.

Job Penyelesaian

Pekerjaan pelabelan bingkai video dan video dapat membutuhkan waktu berjam-jam pekerja untuk menyelesaikannya. Anda dapat mengatur jumlah total waktu yang pekerja dapat bekerja pada setiap tugas ketika Anda membuat pekerjaan pelabelan. Waktu maksimum yang dapat Anda tetapkan bagi pekerja untuk mengerjakan tugas adalah 7 hari. Nilai default-nya adalah 3 hari.

Kami sangat menyarankan Anda membuat tugas yang dapat diselesaikan pekerja dalam waktu 12 jam. Pekerja harus menjaga UI pekerja tetap terbuka saat mengerjakan tugas. Mereka dapat menghemat pekerjaan saat mereka pergi dan Ground Truth menyimpan pekerjaan mereka setiap 15 menit.

Saat menggunakan SageMaker `CreateLabelingJob`Operasi API, mengatur total waktu tugas tersedia untuk pekerja di`TaskTimeLimitInSeconds`parameter`HumanTaskConfig`.

Saat Anda membuat pekerjaan pelabelan di konsol, Anda dapat menentukan batas waktu ini saat memilih jenis tenaga kerja dan tim kerja Anda.

Jenis

Saat Anda membuat pelacakan objek video atau pekerjaan pelabelan deteksi objek video, Anda menentukan jenis anotasi yang ingin dibuat pekerja saat mengerjakan tugas pelabelan Anda. Jenis anotasi menentukan jenis data keluaran Ground Truth kembali dan mendefinisikan`Jenis`untuk pekerjaan pelabelan Anda.

Jika Anda membuat pekerjaan pelabelan menggunakan operasi API`CreateLabelingJob`, Anda menentukan jenis tugas menggunakan parameter file konfigurasi kategori label`annotationType`. Untuk mempelajari selengkapnya, lihat Buat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai.

Jenis tugas berikut tersedia untuk pelacakan objek video atau pekerjaan pelabelan deteksi objek video:

- Kotak pembatas- Pekerja dilengkapi dengan alat untuk membuat anotasi kotak pembatas. Kotak pembatas adalah kotak yang diambil pekerja di sekitar objek untuk mengidentifikasi lokasi piksel dan label objek itu dalam bingkai.
- Polyline- Pekerja dilengkapi dengan alat untuk membuat anotasi polyline. Sebuah polyline didefinisikan oleh serangkaian memerintahkan x, y koordinat. Setiap titik yang ditambahkan ke polyline terhubung ke titik sebelumnya dengan garis. Polyline tidak harus ditutup (titik awal dan titik akhir tidak harus sama) dan tidak ada batasan pada sudut yang terbentuk di antara garis.

- Polygon- Pekerja dilengkapi dengan alat untuk membuat anotasi poligon. Sebuah poligon adalah bentuk tertutup didefinisikan oleh serangkaian memerintahkan x, y koordinat. Setiap titik yang ditambahkan ke poligon terhubung ke titik sebelumnya dengan garis dan tidak ada batasan pada sudut yang terbentuk di antara garis. Dua garis (sisi) poligon tidak bisa menyeberang. Titik awal dan akhir poligon harus sama.
- Keypoint- Pekerja dilengkapi dengan alat untuk membuat anotasi keypoint. Keypoint adalah titik tunggal yang terkait dengan koordinat x, y dalam bingkai video.

Tenaga kerja

Saat Anda membuat pekerjaan pelabelan bingkai video, Anda perlu menentukan tim kerja untuk menyelesaikan tugas anotasi Anda. Anda dapat memilih tim kerja dari tenaga kerja pribadi pekerja Anda sendiri, atau dari tenaga kerja vendor yang Anda pilih diAWS Marketplace. Anda tidak dapat menggunakan tenaga kerja Amazon Mechanical Turk untuk pekerjaan pelabelan bingkai video.

Untuk mempelajari lebih lanjut tentang tenaga kerja vendor, lihatMengelola Tenaga Kerja Vendor.

Untuk mempelajari cara membuat dan mengelola tenaga kerja pribadi, lihatMenggunakan Tenaga Kerja Pribadi.

Antarmuka Pengguna Pekerja (UI)

Ground Truth menyediakan antarmuka pengguna pekerja (UI), alat, dan fitur pelabelan bantu untuk membantu pekerja menyelesaikan tugas pelabelan video Anda. Anda dapat melihat pratinjau UI pekerja saat membuat pekerjaan pelabelan di konsol.

Saat Anda membuat pekerjaan pelabelan menggunakan operasi API`CreateLabelingJob`, Anda harus memberikan ARN yang disediakan oleh Ground Truth dalam parameter`HumanTaskUiArn`untuk menentukan UI pekerja untuk jenis tugas Anda. Anda dapat menggunakan`HumanTaskUiArn`dengan SageMaker `RenderUiTemplate`Operasi API untuk melihat pratinjau UI pekerja.

Anda memberikan instruksi pekerja, label, dan secara opsional, atribut yang dapat digunakan pekerja untuk memberikan informasi lebih lanjut tentang label dan bingkai video. Atribut ini disebut sebagai atribut kategori label dan atribut bingkai masing-masing. Semuanya ditampilkan di UI pekerja.

Kategori Label dan Atribut Bingkai

Saat Anda membuat pelacakan objek video atau pekerjaan pelabelan deteksi objek video, Anda dapat menambahkan satu atau lebihatributdanatribut:

- Kategori- Daftar opsi (string), kotak teks formulir bebas, atau bidang numerik yang terkait dengan satu atau lebih label. Hal ini digunakan oleh pekerja untuk menyediakan metadata tentang label.
- Atribut- Daftar opsi (string), kotak teks formulir bebas, atau bidang numerik yang muncul pada setiap bingkai video yang dikirim pekerja untuk membuat anotasi. Hal ini digunakan oleh pekerja untuk menyediakan metadata tentang frame video.

Selain itu, Anda dapat menggunakan atribut label dan bingkai agar pekerja memverifikasi label dalam tugas verifikasi label bingkai video.

Gunakan bagian berikut untuk mempelajari lebih lanjut tentang atribut ini. Untuk mempelajari cara menambahkan kategori label dan atribut bingkai ke pekerjaan pelabelan, gunakanBuat Job pelabelanpadaHalaman jenispilihan Anda.

Kategori Atribut

Tambahkan atribut kategori label ke label untuk memberi pekerja kemampuan untuk memberikan informasi lebih lanjut tentang anotasi yang mereka buat. Atribut kategori label ditambahkan ke label individual, atau ke semua label. Ketika atribut kategori label diterapkan ke semua label itu disebut sebagaiatribut kategori label global.

Misalnya, jika Anda menambahkan kategori labelmobil, Anda mungkin juga ingin menangkap data tambahan tentang mobil berlabel Anda, seperti jika mereka tersumbat atau ukuran mobil. Anda dapat menangkap metadata ini menggunakan atribut kategori label. Dalam contoh ini, jika Anda menambahkan atributtersumbatke kategori label mobil, Anda dapat menetapkanparsial,seluruhnya,tidakkepadatersumbatatribut dan memungkinkan pekerja untuk memilih salah satu pilihan ini.

Saat membuat tugas verifikasi label, Anda menambahkan atribut kategori label ke setiap label yang ingin diverifikasi oleh pekerja.

Atribut tingkat

Tambahkan atribut bingkai untuk memberi pekerja kemampuan untuk memberikan informasi lebih lanjut tentang bingkai video individual. Setiap atribut frame yang Anda tambahkan muncul di semua frame.

Misalnya, Anda dapat menambahkan atribut number-frame agar pekerja mengidentifikasi jumlah objek yang mereka lihat dalam bingkai tertentu.

Dalam contoh lain, Anda mungkin ingin memberikan kotak teks bentuk bebas untuk memberi pekerja kemampuan untuk memberikan jawaban atas pertanyaan.

Saat membuat tugas verifikasi label, Anda dapat menambahkan satu atau beberapa atribut bingkai untuk meminta pekerja memberikan umpan balik pada semua label dalam bingkai video.

Instruksi Pekerja

Anda dapat memberikan instruksi pekerja untuk membantu pekerja Anda menyelesaikan tugas pelabelan bingkai video Anda. Anda mungkin ingin membahas topik berikut saat menulis instruksi Anda:

- Praktik terbaik dan hal-hal yang harus dihindari saat membuat anotasi objek.
- Kategori label atribut yang disediakan (untuk deteksi objek dan tugas pelacakan objek) dan bagaimana menggunakannya.
- Cara menghemat waktu saat memberi label dengan menggunakan pintasan keyboard.

Anda dapat menambahkan instruksi pekerja Anda menggunakan SageMaker konsol sambil membuat pekerjaan pelabelan. Jika Anda membuat pekerjaan pelabelan menggunakan operasi API`CreateLabelingJob`, Anda menentukan instruksi pekerja dalam file konfigurasi kategori label Anda.

Selain instruksi Anda, Ground Truth menyediakan tautan untuk membantu pekerja menavigasi dan menggunakan portal pekerja. Lihat petunjuk ini dengan memilih jenis tugas padaInstruksi Pekerja.

Menolak

Pekerja dapat menolak tugas.

Pekerja menolak tugas jika instruksi tidak jelas, data input tidak ditampilkan dengan benar, atau jika mereka mengalami masalah lain dengan tugas tersebut. Jika jumlah pekerja per objek dataset (`NumberOfHumanWorkersPerDataObject`) Menolak tugas, objek data ditandai sebagai kadaluarsa dan tidak akan dikirim ke pekerja tambahan.

Persyaratan Izin Job Bingkai Video

Saat Anda membuat pekerjaan pelabelan bingkai video, selain persyaratan izin yang ditemukan diTetapkan Izin IAM untuk Menggunakan Ground Truth, Anda harus menambahkan kebijakan CORS ke bucket S3 Anda yang berisi file manifes masukan Anda.

Menambahkan Kebijakan Izin CORS ke S3 Bucket

Saat Anda membuat pekerjaan pelabelan bingkai video, Anda menentukan bucket di S3 tempat data input dan file manifes Anda berada dan di mana data keluaran Anda akan disimpan. Ember ini mungkin sama. Anda harus melampirkan kebijakan Cross-origin resource sharing (CORS) kebijakan ke bucket input dan output Anda. Jika Anda menggunakan konsol Amazon S3 untuk menambahkan kebijakan ke bucket Anda, Anda harus menggunakan format JSON.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "HEAD",
            "PUT"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [
            "Access-Control-Allow-Origin"
        ],
        "MaxAgeSeconds": 3000
    }
]
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>HEAD</AllowedMethod>
    <AllowedMethod>PUT</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>Access-Control-Allow-Origin</ExposeHeader>
    <AllowedHeader>*</AllowedHeader>
```

```
</CORSRule>
</CORSConfiguration>
```

Untuk mempelajari cara menambahkan kebijakan CORS ke bucket S3, lihatBagaimana cara menambahkan pembagian sumber daya lintas domain dengan CORS?dalam Panduan Pengguna Amazon Simple Storage Service.

Instruksi Pekerja

Topik ini memberikan gambaran umum tentang portal pekerja Ground Truth dan alat yang tersedia untuk menyelesaikan tugas pelabelan bingkai video Anda. Pertama, pilih jenis tugas yang sedang Anda kerjakanTopik.

> **Important**
>
> Dianjurkan agar Anda menyelesaikan tugas Anda menggunakan browser web Google Chrome atau Firefox.

Untuk pekerjaan penyesuaian, pilih jenis tugas tugas pelabelan asli yang menghasilkan label yang Anda sesuaikan. Tinjau dan sesuaikan label dalam tugas Anda sesuai kebutuhan.

Topik

- Bekerja pada Tugas Pelacakan Objek Bingkai Video
- Bekerja pada Tugas Deteksi Objek Bingkai Video

Bekerja pada Tugas Pelacakan Objek Bingkai Video

Tugas pelacakan objek bingkai video mengharuskan Anda untuk melacak pergerakan objek di seluruh bingkai video. Bingkai video adalah gambar diam dari adegan video.

Anda dapat menggunakan UI pekerja untuk menavigasi antara bingkai video dan menggunakan alat yang disediakan untuk mengidentifikasi objek unik dan melacak pergerakannya dari satu dari yang berikutnya. Gunakan halaman ini untuk mempelajari cara menavigasi UI pekerja, menggunakan alat yang disediakan, dan menyelesaikan tugas Anda.

Dianjurkan agar Anda menyelesaikan tugas Anda menggunakan browser web Google Chrome atau Firefox.

> **Important**
>
> Jika Anda melihat anotasi telah ditambahkan ke satu atau beberapa bingkai video saat Anda membuka tugas, sesuaikan anotasi tersebut dan tambahkan anotasi tambahan sesuai kebutuhan.

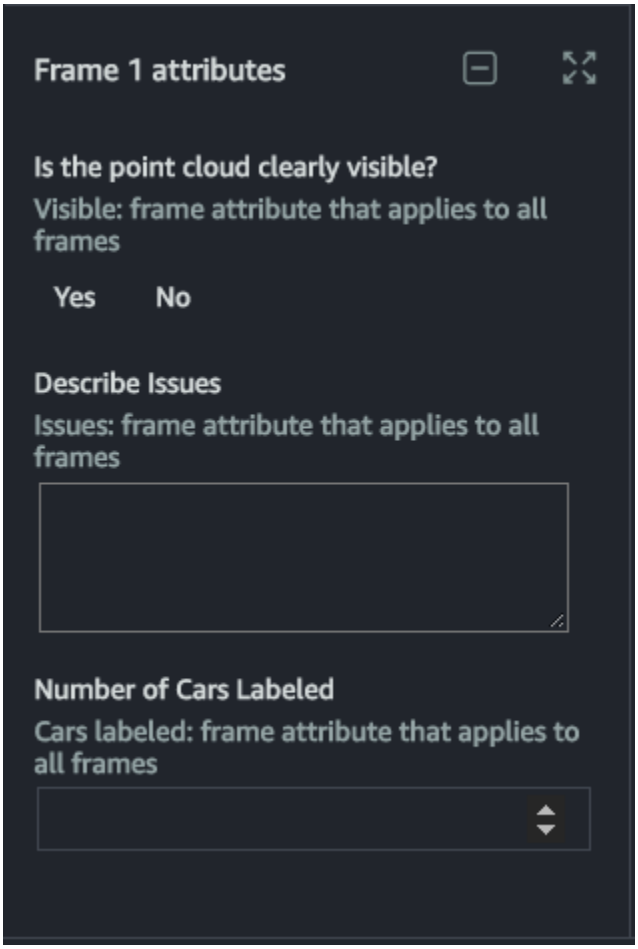
Topik

Tugas

Saat Anda mengerjakan tugas pelacakan objek bingkai video, Anda harus memilih kategori dariKategori labelmenu di sisi kanan portal pekerja Anda untuk mulai membuat anotasi. Setelah Anda memilih kategori, gunakan alat yang disediakan untuk membuat anotasi objek yang diterapkan kategori. Anotasi ini akan dikaitkan dengan ID label unik yang seharusnya hanya digunakan untuk objek tersebut. Gunakan ID label yang sama ini untuk membuat anotasi tambahan untuk objek yang sama di semua bingkai video yang muncul. LihatPanduan Alatuntuk mempelajari lebih lanjut tentang alat yang disediakan.

Setelah menambahkan label, Anda mungkin melihat panah mengarah ke bawah di samping label diLabelMenu Pilih panah ini dan kemudian pilih satu opsi untuk setiap atribut label yang Anda lihat untuk memberikan informasi lebih lanjut tentang label tersebut.

Anda mungkin melihat atribut frame di bawahLabelMenu Atribut ini akan muncul di setiap frame dalam tugas Anda. Gunakan petunjuk atribut ini untuk memasukkan informasi tambahan tentang setiap frame.



Setelah menambahkan label, Anda dapat dengan cepat menambahkan dan mengedit nilai atribut kategori label dengan menggunakan panah penunjuk ke bawah di samping label diLabelMenu Jika Anda memilih ikon pensil di sebelah label diLabelMenuEditmenu akan muncul. Anda dapat mengedit ID label, kategori label, dan atribut kategori label menggunakan menu ini.

Untuk mengedit anotasi, pilih label anotasi yang ingin Anda edit diLabelmenu atau pilih anotasi dalam bingkai. Saat Anda mengedit atau menghapus anotasi, tindakan hanya akan mengubah anotasi dalam satu bingkai.

Jika Anda mengerjakan tugas yang menyertakan alat kotak pembatas, gunakan ikon prediksi berikutnya untuk memprediksi lokasi semua kotak pembatas yang telah Anda gambar dalam bingkai di bingkai berikutnya. Jika Anda memilih satu kotak dan kemudian memilih ikon prediksi berikutnya, hanya kotak yang akan diprediksi dalam bingkai berikutnya. Jika Anda belum menambahkan kotak apa pun ke bingkai saat ini, Anda akan menerima galat. Anda harus menambahkan setidaknya satu kotak ke bingkai sebelum menggunakan fitur ini.

Setelah Anda menggunakan ikon prediksi berikutnya, tinjau lokasi setiap kotak di bingkai berikutnya dan buat penyesuaian ke lokasi dan ukuran kotak jika perlu.

Untuk semua alat lainnya, Anda dapat menggunakanSalin ke selanjutnyadanSalin ke semuaalat untuk menyalin anotasi Anda ke frame berikutnya atau semua masing-masing.

Menavigasi UI

Anda dapat menavigasi antara bingkai video menggunakan bilah navigasi di sudut kiri bawah UI Anda.

Gunakan tombol putar untuk secara otomatis bergerak melalui seluruh urutan bingkai.

Gunakan frame berikutnya dan tombol frame sebelumnya untuk bergerak maju atau mundur satu frame pada satu waktu. Anda juga dapat memasukkan nomor bingkai untuk menavigasi ke frame itu.

Anda dapat memperbesar dan memperkecil semua bingkai video. Setelah Anda memperbesar ke dalam bingkai video, Anda dapat bergerak dalam bingkai itu menggunakan ikon bergerak. Saat Anda menetapkan tampilan baru dalam satu bingkai video dengan memperbesar dan memindahkan dalam bingkai itu, semua bingkai video diatur ke tampilan yang sama. Anda dapat mengatur ulang semua bingkai video ke tampilan aslinya menggunakan ikon layar yang pas. Untuk opsi tampilan tambahan, lihatPanduan ikon.

Ketika Anda berada di UI, Anda akan melihat menu berikut:

- Petunjuk- Tinjau instruksi ini sebelum memulai tugas Anda. Selain itu, pilihInstruksi lainnyadan tinjau instruksi ini.
- Pintasan- Gunakan menu ini untuk melihat pintasan keyboard yang dapat Anda gunakan untuk menavigasi bingkai video dan menggunakan alat yang disediakan.
- bantuan- Gunakan opsi ini untuk merujuk kembali ke dokumentasi ini.

Label Edit Massal dan Atribut Bingkai

Anda dapat mengedit atribut label secara massal dan atribut bingkai (atribut).

Ketika Anda mengedit atribut secara massal, Anda menentukan satu atau beberapa rentang frame yang ingin Anda terapkan edit. Atribut yang Anda pilih diedit di semua frame dalam kisaran tersebut, termasuk bingkai awal dan akhir yang Anda tentukan. Saat Anda mengedit atribut label secara massal, rentang yang Anda tentukanharusmengandung label yang dilampirkan atribut label. Jika Anda menentukan frame yang tidak mengandung label ini, Anda akan menerima galat.

Untuk mengedit atribut secara massal Andaharustentukan nilai yang diinginkan untuk atribut terlebih dahulu. Misalnya, jika Anda ingin mengubah atribut dariyakepadaTidak, Anda harus memilihTidak, dan kemudian melakukan pengeditan massal.

Anda juga dapat menentukan nilai baru untuk atribut yang belum diisi dan kemudian menggunakan fitur edit massal untuk mengisi nilai tersebut dalam beberapa frame. Untuk melakukannya, pilih nilai yang diinginkan untuk atribut dan selesaikan prosedur berikut.

Untuk mengedit label atau atribut secara massal:

1. Gunakan mouse Anda untuk mengklik kanan atribut yang ingin Anda edit secara massal.
2. Tentukan rentang bingkai yang ingin Anda terapkan pengeditan massal menggunakan tanda hubung (`-`) di kotak teks. Misalnya, jika Anda ingin menerapkan pengeditan ke bingkai satu hingga sepuluh, masukkan`1-10`. Jika Anda ingin menerapkan edit ke frame dua hingga lima, delapan sampai sepuluh dan dua puluh masukkan`2-5,8-10,20`.
3. PilihTerkonfirmasi.

Jika Anda mendapatkan pesan kesalahan, verifikasi bahwa Anda memasukkan rentang yang valid dan bahwa label yang terkait dengan atribut label yang Anda edit (jika berlaku) ada di semua frame yang ditentukan.

Anda dapat dengan cepat menambahkan label ke semua frame sebelumnya atau berikutnya menggunakanDuplikat ke frame sebelumnyadanDuplikat ke frame berikutnyapilihanLabelmenu di bagian atas layar.

Panduan Alat

Tugas Anda akan mencakup satu atau lebih alat. Alat yang disediakan menentukan jenis anotasi yang akan Anda buat untuk mengidentifikasi dan melacak objek. Gunakan tabel berikut untuk mempelajari selengkapnya tentang setiap alat yang disediakan.

Alat	Ikon	Action	Deskripsi
Kotak pembatas		Tambahkan anotasi kotak pembatas.	Pilih ikon ini untuk menambahkan kotak pembatas. Setiap kotak pembatas yang Anda tambahkan

Alat	Ikon	Action	Deskripsi
			dikaitkan dengan kategori yang Anda pilih dari menu drop down kategori Label. Pilih kotak pembatas atau label terkait untuk menyesuai kannya.
Kotak pembatas		Memprediksi kotak pembatas di frame berikutnya.	Pilih kotak pembatas, lalu pilih ikon ini untuk memprediksi lokasi kotak itu di bingkai berikutnya. Anda dapat memilih ikon beberapa kali berturut-turut untuk secara otomatis mendeteksi lokasi kotak dalam beberapa bingkai. Misalnya, pilih ikon ini 5 kali untuk memprediksi lokasi kotak pembatas di 5 frame berikutnya.

Alat	Ikon	Action	Deskripsi
Keypoint		Tambahkan anotasi keypoint.	Pilih ikon ini untuk menambahkan keypoint. Klik pada objek gambar untuk menempatkan keypoint di lokasi itu. Setiap keypoint yang Anda tambahkan dikaitkan dengan kategori yang Anda pilih dari menu drop down kategori Label. Pilih keypoint atau label terkait untuk menyesuaikannya.

Alat	Ikon	Action	Deskripsi
Polyline		Tambahkan anotasi polyline.	Pilih ikon ini untuk menambahkan polyline. Untuk menambahkan polyline, terus klik di sekitar objek yang menarik untuk menambahkan poin baru. Untuk berhenti menggamba r polyline, pilih titik terakhir yang Anda tempatkan untuk kedua kalinya (titik ini akan berwarna hijau), atau tekanENpada keyboard Anda. Setiap titik yang ditambahkan ke polyline terhubung ke titik sebelumnya dengan garis. Polyline tidak harus ditutup (titik awal dan titik akhir tidak harus sama) dan tidak ada batasan pada sudut yang terbentuk di antara garis. Setiap polyline yang Anda tambahkan dikaitkan dengan kategori yang Anda

Alat	Ikon	Action	Deskripsi
			pilih dari menu drop down kategori Label. Pilih polyline atau label terkait untuk menyesuaikannya.

Alat	Ikon	Action	Deskripsi
Polygon		Tambahkan anotasi poligon.	Pilih ikon ini untuk menambahkan poligon. Untuk menambahkan poligon, terus klik di sekitar objek yang menarik untuk menambahkan poin baru. Untuk berhenti menggambar poligon, pilih titik awal (titik ini akan berwarna hijau). Poligon adalah bentuk tertutup yang didefinis ikan oleh serangkai an titik yang Anda tempatkan. Setiap titik yang ditambahkan ke poligon terhubung ke titik sebelumny a dengan garis dan tidak ada batasan pada sudut yang terbentuk di antara garis. Titik awal dan akhir harus sama. Setiap poligon yang Anda tambahkan dikaitkan dengan kategori yang Anda pilih dari menu drop-down kategori Label. Pilih poligon atau

Alat	Ikon	Action	Deskripsi
			label terkait untuk menyesuaikannya.
Salin ke Berikutnya		Salin anotasi ke frame berikutnya.	Jika satu atau lebih anotasi dipilih dalam bingkai saat ini, anotasi tersebut akan disalin ke frame berikutnya. Jika tidak ada anotasi yang dipilih, semua anotasi dalam bingkai saat ini akan disalin ke frame berikutnya.
Salin ke Semua		Salin anotasi ke semua frame berikutnya.	Jika satu atau lebih anotasi dipilih dalam bingkai saat ini, anotasi tersebut akan disalin ke semua frame berikutnya. Jika tidak ada anotasi yang dipilih, semua anotasi dalam bingkai saat ini akan disalin ke semua frame berikutnya.

Panduan ikon

Gunakan tabel ini untuk mempelajari ikon yang Anda lihat di UI. Anda dapat secara otomatis memilih beberapa ikon ini menggunakan pintasan keyboard yang ditemukan diPintasanMenu

Ikon	Action	Deskripsi
	keterangan	Pilih ikon ini untuk menyesuaikan kecerahan semua bingkai video.
	kontras	Pilih ikon ini untuk menyesuaikan kontras semua bingkai video.
	Perbesar	Pilih ikon ini untuk memperbesar semua bingkai video.
	Perbesar	Pilih ikon ini untuk memperkecil semua bingkai video.
	pindahkan layar	Setelah Anda memperbesar ke dalam bingkai video, pilih ikon ini untuk bergerak dalam bingkai video tersebut. Anda dapat bergerak di sekitar bingkai video menggunak an mouse Anda dengan mengklik dan menyeret bingkai ke arah yang Anda inginkan untuk bergerak. Ini akan mengubah tampilan di semua bingkai tampilan.
	Layar	Setel ulang semua bingkai video ke posisi semula.
	membatalkan	Membatalkan tindakan. Anda dapat menggunakan ikon ini untuk menghapus kotak pembatas yang baru saja Anda tambahkan, atau untuk membatalkan penyesuaian yang Anda buat ke kotak pembatas.
	pengulangan	Ulangi tindakan yang dibatalkan menggunakan ikon undo.
	hapus label	Hapus label. Ini akan menghapus kotak pembatas yang terkait dengan label dalam satu bingkai.

Ikon	Action	Deskripsi
	tampilkan atau sembunyikan label	Pilih ikon ini untuk menampilkan label yang telah disembunyikan. Jika ikon ini memiliki garis miring, pilih untuk menyembunyikan label.
	Edit label	Pilih ikon ini untuk membukaEditMenu Gunakan menu ini untuk mengedit kategori label, ID, dan untuk menambah atau mengedit atribut label.

Pintasan

Pintasan keyboard yang tercantum dalamPintasanmenu dapat membantu Anda dengan cepat memilih ikon, membatalkan dan mengulang anotasi, dan menggunakan alat untuk menambah dan mengedit anotasi. Misalnya, setelah Anda menambahkan kotak pembatas, Anda dapat menggunakanPuntuk dengan cepat memprediksi lokasi kotak itu dalam bingkai berikutnya.

Sebelum Anda memulai tugas Anda, disarankan agar Anda meninjauPintasanmenu dan berkenalan dengan perintah-perintah ini.

Rilis, Hentikan, dan Lanjutkan, dan Tolak Tugas

Saat Anda membuka tugas pelabelan, tiga tombol di kanan atas memungkinkan Anda menolak tugas (Menolak), lepaskan (Tugas), dan lanjutkan di lain waktu (Berhenti dan lanjutkan). Daftar berikut menjelaskan apa yang terjadi ketika Anda memilih salah satu opsi ini:

- Menolak tugas: Anda hanya boleh menolak tugas jika ada yang salah dengan tugas, seperti gambar bingkai video yang tidak jelas atau masalah dengan UI. Jika Anda menolak tugas, Anda tidak akan dapat kembali ke tugas.
- Tugas: Gunakan opsi ini untuk melepaskan tugas dan memungkinkan orang lain untuk mengerjakannya. Ketika Anda melepaskan tugas, Anda kehilangan semua pekerjaan yang dilakukan pada tugas itu dan pekerja lain di tim Anda dapat mengambilnya. Jika cukup pekerja yang mengambil tugas, Anda mungkin tidak akan dapat kembali ke sana. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja. Jika tugas masih tersedia, statusnya akanTersedia. Jika pekerja lain mengambilnya, itu akan hilang dari portal Anda.
- Berhenti dan lanjutkan: Anda dapat menggunakanBerhenti dan lanjutkantombol untuk berhenti bekerja dan kembali ke tugas di lain waktu. Anda harus menggunakanSimpantombol untuk menyimpan pekerjaan Anda sebelum Anda memilihBerhenti dan lanjutkan. Bila Anda memilih

tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja, dan status tugasnya adalahDihentikan. Anda dapat memilih tugas yang sama untuk melanjutkan pekerjaan di atasnya.

Ketahuilah bahwa orang yang membuat tugas pelabelan Anda menentukan batas waktu di mana semua tugas banyak diselesaikan. Jika Anda tidak kembali ke dan menyelesaikan tugas ini dalam batas waktu itu, itu akan kedaluwarsa dan pekerjaan Anda tidak akan diserahkan. Hubungi administrator Anda untuk informasi lebih lanjut.

Menyimpan Pekerjaan Anda dan Mengirimkan

Anda harus menyimpan pekerjaan Anda secara berkala menggunakanSimpanTombol. Ground Truth akan secara otomatis menyimpan pekerjaan Anda pernah 15 menit.

Ketika Anda membuka tugas, Anda harus menyelesaikan pekerjaan Anda di atasnya sebelum menekanKIRIMKAN.

Bekerja pada Tugas Deteksi Objek Bingkai Video

Tugas deteksi objek bingkai video mengharuskan Anda untuk mengklasifikasikan dan mengidentifikasi lokasi objek dalam bingkai video menggunakan anotasi. Bingkai video adalah gambar diam dari adegan video.

Anda dapat menggunakan UI pekerja untuk menavigasi antara bingkai video dan membuat anotasi untuk mengidentifikasi objek yang menarik. Gunakan bagian di halaman ini untuk mempelajari cara menavigasi UI pekerja, menggunakan alat yang disediakan, dan menyelesaikan tugas Anda.

Dianjurkan agar Anda menyelesaikan tugas Anda menggunakan browser web Google Chrome.

> **Important**
>
> Jika Anda melihat anotasi telah ditambahkan ke satu atau beberapa bingkai video saat Anda membuka tugas, sesuaikan anotasi tersebut dan tambahkan anotasi tambahan sesuai kebutuhan.

Topik

- Tugas

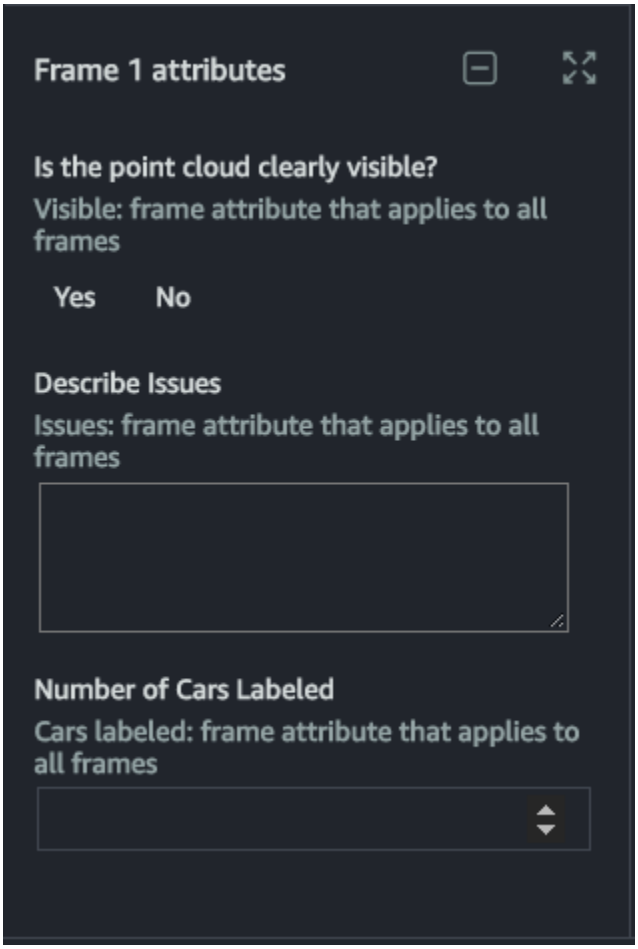
- Menavigasi UI
- Label Edit Massal dan Atribut Bingkai
- Panduan Alat
- Panduan UI
- Pintasan
- Rilis, Hentikan, dan Lanjutkan, dan Tolak Tugas
- Menyimpan Pekerjaan Anda dan Mengirimkan

Tugas

Saat Anda mengerjakan tugas deteksi objek bingkai video, Anda harus memilih kategori dariKategori labelmenu di sisi kanan portal pekerja Anda untuk mulai membuat anotasi. Setelah Anda memilih kategori, gambar anotasi di sekitar objek yang berlaku untuk kategori ini. Untuk mempelajari lebih lanjut tentang alat yang Anda lihat di UI pekerja, lihatPanduan Alat.

Setelah menambahkan label, Anda mungkin melihat panah mengarah ke bawah di samping label diLabelMenu Pilih panah ini dan kemudian pilih satu opsi untuk setiap atribut label yang Anda lihat untuk memberikan informasi lebih lanjut tentang label tersebut.

Anda mungkin melihat atribut frame di bawahLabelMenu Atribut ini akan muncul di setiap frame dalam tugas Anda. Gunakan petunjuk atribut ini untuk memasukkan informasi tambahan tentang setiap frame.



Untuk mengedit anotasi, pilih label anotasi yang ingin Anda edit diLabelmenu atau pilih anotasi dalam bingkai. Saat Anda mengedit atau menghapus anotasi, tindakan hanya akan mengubah anotasi dalam satu bingkai.

Jika Anda mengerjakan tugas yang menyertakan alat kotak pembatas, gunakan ikon prediksi berikutnya untuk memprediksi lokasi semua kotak pembatas yang telah Anda gambar dalam bingkai di bingkai berikutnya. Jika Anda memilih satu kotak dan kemudian memilih ikon prediksi berikutnya, hanya kotak yang akan diprediksi dalam bingkai berikutnya. Jika Anda belum menambahkan kotak apa pun ke bingkai saat ini, Anda akan menerima galat. Anda harus menambahkan setidaknya satu kotak ke bingkai sebelum menggunakan fitur ini.

> **Note**
>
> Fitur prediksi berikutnya tidak akan menimpa anotasi yang dibuat secara manual. Ini hanya akan menambahkan anotasi. Jika Anda menggunakan prediksi berikutnya dan sebagai

hasilnya memiliki lebih dari satu kotak pembatas di sekitar satu objek, hapus semua kecuali satu kotak. Setiap objek hanya boleh diidentifikasi dengan satu kotak.

Setelah Anda menggunakan ikon prediksi berikutnya, tinjau lokasi setiap kotak di bingkai berikutnya dan buat penyesuaian ke lokasi dan ukuran kotak jika perlu.

Untuk semua alat lainnya, Anda dapat menggunakanSalin ke selanjutnyadanSalin ke semuaalat untuk menyalin anotasi Anda ke frame berikutnya atau semua masing-masing.

Menavigasi UI

Anda dapat menavigasi antara bingkai video menggunakan bilah navigasi di sudut kiri bawah UI Anda.

Gunakan tombol putar untuk secara otomatis bermain melalui beberapa frame.

Gunakan frame berikutnya dan tombol frame sebelumnya untuk bergerak maju atau mundur satu frame pada satu waktu. Anda juga dapat memasukkan nomor bingkai untuk menavigasi ke frame itu.

Anda dapat memperbesar dan memperkecil semua bingkai video. Setelah Anda memperbesar ke dalam bingkai video, Anda dapat bergerak dalam bingkai itu menggunakan ikon bergerak. Saat Anda menavigasi ke tampilan baru dalam satu bingkai video dengan memperbesar dan memindahkan dalam bingkai itu, semua bingkai video diatur ke tampilan yang sama. Anda dapat mengatur ulang semua bingkai video ke tampilan aslinya menggunakan ikon layar yang pas. Untuk mempelajari selengkapnya, lihat Panduan UI.

Ketika Anda berada di UI, Anda akan melihat menu berikut:

- Petunjuk- Tinjau instruksi ini sebelum memulai tugas Anda. Selain itu, pilihInstruksi lainnyadan tinjau instruksi ini.
- Pintasan- Gunakan menu ini untuk melihat pintasan keyboard yang dapat Anda gunakan untuk menavigasi bingkai video dan menggunakan alat anotasi yang disediakan.
- bantuan- Gunakan opsi ini untuk merujuk kembali ke dokumentasi ini.

Jika Anda

Label Edit Massal dan Atribut Bingkai

Anda dapat mengedit atribut label secara massal dan atribut bingkai (atribut).

Ketika Anda mengedit atribut secara massal, Anda menentukan satu atau beberapa rentang frame yang ingin Anda terapkan edit. Atribut yang Anda pilih diedit di semua frame dalam kisaran tersebut, termasuk bingkai awal dan akhir yang Anda tentukan. Saat Anda mengedit atribut label secara massal, rentang yang Anda tentukanharusmengandung label yang dilampirkan atribut label. Jika Anda menentukan frame yang tidak mengandung label ini, Anda akan menerima galat.

Untuk mengedit atribut secara massal Andaharustentukan nilai yang diinginkan untuk atribut terlebih dahulu. Misalnya, jika Anda ingin mengubah atribut dariyakepadaTidak, Anda harus memilihTidak, dan kemudian melakukan pengeditan massal.

Anda juga dapat menentukan nilai baru untuk atribut yang belum diisi dan kemudian menggunakan fitur edit massal untuk mengisi nilai tersebut dalam beberapa frame. Untuk melakukannya, pilih nilai yang diinginkan untuk atribut dan selesaikan prosedur berikut.

Untuk mengedit label atau atribut secara massal:

1. Gunakan mouse Anda untuk mengklik kanan atribut yang ingin Anda edit secara massal.
2. Tentukan rentang bingkai yang ingin Anda terapkan pengeditan massal menggunakan tanda hubung (`-`) di kotak teks. Misalnya, jika Anda ingin menerapkan pengeditan ke bingkai satu hingga sepuluh, masukkan`1-10`. Jika Anda ingin menerapkan edit ke frame dua hingga lima, delapan sampai sepuluh dan dua puluh masukkan`2-5,8-10,20`.
3. PilihTerkonfirmasi.

Jika Anda mendapatkan pesan kesalahan, verifikasi bahwa Anda memasukkan rentang yang valid dan bahwa label yang terkait dengan atribut label yang Anda edit (jika berlaku) ada di semua frame yang ditentukan.

Anda dapat dengan cepat menambahkan label ke semua frame sebelumnya atau berikutnya menggunakanDuplikat ke frame sebelumnyadanDuplikat ke frame berikutnyapilihanLabelmenu di bagian atas layar.

Panduan Alat

Tugas Anda akan mencakup satu atau lebih alat. Alat yang disediakan menentukan jenis anotasi yang akan Anda buat untuk mengidentifikasi dan memberi label objek. Gunakan tabel berikut untuk mempelajari lebih lanjut tentang alat atau alat yang mungkin Anda lihat di UI pekerja.

Alat	Ikon	Action	Deskripsi
Kotak pembatas		Tambahkan anotasi kotak pembatas.	Pilih ikon ini untuk menambahkan kotak pembatas. Setiap kotak pembatas yang Anda tambahkan dikaitkan dengan kategori yang Anda pilih dari menu drop down kategori Label. Pilih kotak pembatas atau label terkait untuk menyesuai kannya.
Prediksi selanjutnya		Memprediksi kotak pembatas di frame berikutnya.	Pilih kotak pembatas, lalu pilih ikon ini untuk mempredik si lokasi kotak itu di bingkai berikutnya. Anda dapat memilih ikon beberapa kali berturut-turut untuk secara otomatis mendeteksi lokasi kotak dalam beberapa bingkai. Misalnya, pilih ikon ini 5 kali untuk memprediksi lokasi kotak pembatas di 5 frame berikutnya.
Keypoint		Tambahkan anotasi keypoint.	Pilih ikon ini untuk menambahkan keypoint. Klik pada objek gambar untuk

Alat	Ikon	Action	Deskripsi
			menempatkan keypoint di lokasi itu. Setiap keypoint yang Anda tambahkan dikaitkan dengan kategori yang Anda pilih dari menu drop down kategori Label. Pilih keypoint atau label terkait untuk menyesuaikannya.

Alat	Ikon	Action	Deskripsi
Polyline		Tambahkan anotasi polyline.	Pilih ikon ini untuk menambahkan polyline. Untuk menambahkan polyline, terus klik di sekitar objek yang menarik untuk menambahkan poin baru. Untuk berhenti menggamba r polyline, pilih titik terakhir yang Anda tempatkan untuk kedua kalinya (titik ini akan berwarna hijau), atau tekanENpada keyboard Anda. Setiap titik yang ditambahkan ke polyline terhubung ke titik sebelumnya dengan garis. Polyline tidak harus ditutup (titik awal dan titik akhir tidak harus sama) dan tidak ada batasan pada sudut yang terbentuk di antara garis. Setiap polyline yang Anda tambahkan dikaitkan dengan kategori yang Anda

Alat	Ikon	Action	Deskripsi
			pilih dari menu drop down kategori Label. Pilih polyline atau label terkait untuk menyesuaikannya.

Alat	Ikon	Action	Deskripsi
Polygon		Tambahkan anotasi poligon.	Pilih ikon ini untuk menambahkan poligon. Untuk menambahkan poligon, terus klik di sekitar objek yang menarik untuk menambahkan poin baru. Untuk berhenti menggambar poligon, pilih titik awal (titik ini akan berwarna hijau). Poligon adalah bentuk tertutup yang didefinis ikan oleh serangkai an titik yang Anda tempatkan. Setiap titik yang ditambahkan ke poligon terhubung ke titik sebelumny a dengan garis dan tidak ada batasan pada sudut yang terbentuk di antara garis. Dua garis (sisi) poligon tidak bisa menyeberang. Garis akan menjadi merah jika melanggar kondisi ini. Titik awal dan akhir harus sama. Setiap poligon yang Anda tambahkan

Alat	Ikon	Action	Deskripsi
			dikaitkan dengan kategori yang Anda pilih dari menu drop-down kategori Label. Pilih poligon atau label terkait untuk menyesuaikannya.
Salin ke Berikutnya		Salin anotasi ke frame berikutnya.	Jika satu atau lebih anotasi dipilih dalam bingkai saat ini, anotasi tersebut akan disalin ke frame berikutnya. Jika tidak ada anotasi yang dipilih, semua anotasi dalam bingkai saat ini akan disalin ke frame berikutnya.
Salin ke Semua		Salin anotasi ke semua frame berikutnya.	Jika satu atau lebih anotasi dipilih dalam bingkai saat ini, anotasi tersebut akan disalin ke semua frame berikutnya. Jika tidak ada anotasi yang dipilih, semua anotasi dalam bingkai saat ini akan disalin ke semua frame berikutnya.

Panduan UI

Gunakan tabel ini untuk mempelajari tentang ikon yang Anda lihat di portal tugas pekerja Anda. Anda dapat secara otomatis memilih ikon ini menggunakan pintasan keyboard yang ditemukan diPintasanMenu

Ikon		Deskripsi
	keterangan	Pilih ikon ini untuk menyesuaikan kecerahan semua bingkai video.
	kontras	Pilih ikon ini untuk menyesuaikan kontras semua bingkai video.
	Perbesar	Pilih ikon ini untuk memperbesar semua bingkai video.
	Perbesar	Pilih ikon ini untuk memperkecil semua bingkai video.
	pindahkan layar	Setelah Anda memperbesar ke dalam bingkai video, pilih ikon ini untuk bergerak dalam bingkai video tersebut. Anda dapat bergerak dalam bingkai video menggunak an mouse Anda dengan mengklik dan menyeret bingkai ke arah yang Anda inginkan untuk bergerak. Ini akan mengubah tampilan di semua bingkai tampilan.
	Layar	Setel ulang semua bingkai video ke posisi semula.
	membatalkan	Membatalkan tindakan. Anda dapat menggunakan ikon ini untuk menghapus kotak pembatas yang baru saja Anda tambahkan, atau untuk membatalkan penyesuaian yang Anda buat ke kotak pembatas.

Ikon		Deskripsi
	pengulangan	Ulangi tindakan yang dibatalkan menggunakan ikon undo.
	hapus label	Hapus label. Ini akan menghapus kotak pembatas yang terkait dengan label dalam satu bingkai.
	tampilkan atau sembunyikan label	Pilih ikon ini untuk menampilkan label yang telah disembunyikan. Jika ikon ini memiliki garis miring, pilih untuk menyembunyikan label.

Pintasan

Pintasan keyboard yang tercantum dalamPintasanmenu dapat membantu Anda dengan cepat memilih ikon, membatalkan dan mengulang anotasi, dan menggunakan alat untuk menambah dan mengedit anotasi. Misalnya, setelah Anda menambahkan kotak pembatas, Anda dapat menggunakanPuntuk dengan cepat memprediksi lokasi kotak itu dalam bingkai berikutnya.

Sebelum Anda memulai tugas Anda, disarankan agar Anda meninjauPintasanmenu dan berkenalan dengan perintah-perintah ini.

Rilis, Hentikan, dan Lanjutkan, dan Tolak Tugas

Saat Anda membuka tugas pelabelan, tiga tombol di kanan atas memungkinkan Anda menolak tugas (Menolak tugas), lepaskan (Tugas), dan lanjutkan di lain waktu (Berhenti dan lanjutkan). Daftar berikut menjelaskan apa yang terjadi ketika Anda memilih salah satu opsi ini:

- Menolak tugas: Anda hanya boleh menolak tugas jika ada yang salah dengan tugas, seperti gambar bingkai video yang tidak jelas atau masalah dengan UI. Jika Anda menolak tugas, Anda tidak akan dapat kembali ke tugas.
- Tugas: Gunakan opsi ini untuk melepaskan tugas dan memungkinkan orang lain untuk mengerjakannya. Ketika Anda melepaskan tugas, Anda kehilangan semua pekerjaan yang dilakukan pada tugas itu dan pekerja lain di tim Anda dapat mengambilnya. Jika cukup pekerja yang mengambil tugas, Anda mungkin tidak akan dapat kembali ke sana. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja. Jika tugas masih tersedia, statusnya akanTersedia. Jika pekerja lain mengambilnya, itu akan hilang dari portal Anda.

- Berhenti dan lanjutkan: Anda dapat menggunakanBerhenti dan lanjutkantombol untuk berhenti bekerja dan kembali ke tugas di lain waktu. Anda harus menggunakanSimpantombol untuk menyimpan pekerjaan Anda sebelum Anda memilihBerhenti dan lanjutkan. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja, dan status tugasnya adalahDihentikan. Anda dapat memilih tugas yang sama untuk melanjutkan pekerjaan di atasnya.

  Ketahuilah bahwa orang yang membuat tugas pelabelan Anda menentukan batas waktu di mana semua tugas banyak diselesaikan. Jika Anda tidak kembali ke dan menyelesaikan tugas ini dalam batas waktu itu, itu akan kedaluwarsa dan pekerjaan Anda tidak akan diserahkan. Hubungi administrator Anda untuk informasi lebih lanjut.

Menyimpan Pekerjaan Anda dan Mengirimkan

Anda harus menyimpan pekerjaan Anda secara berkala. Ground Truth secara otomatis menghemat pekerjaan Anda setiap 15 menit.

Ketika Anda membuka tugas, Anda harus menyelesaikan pekerjaan Anda sebelum menekanKIRIMKAN.

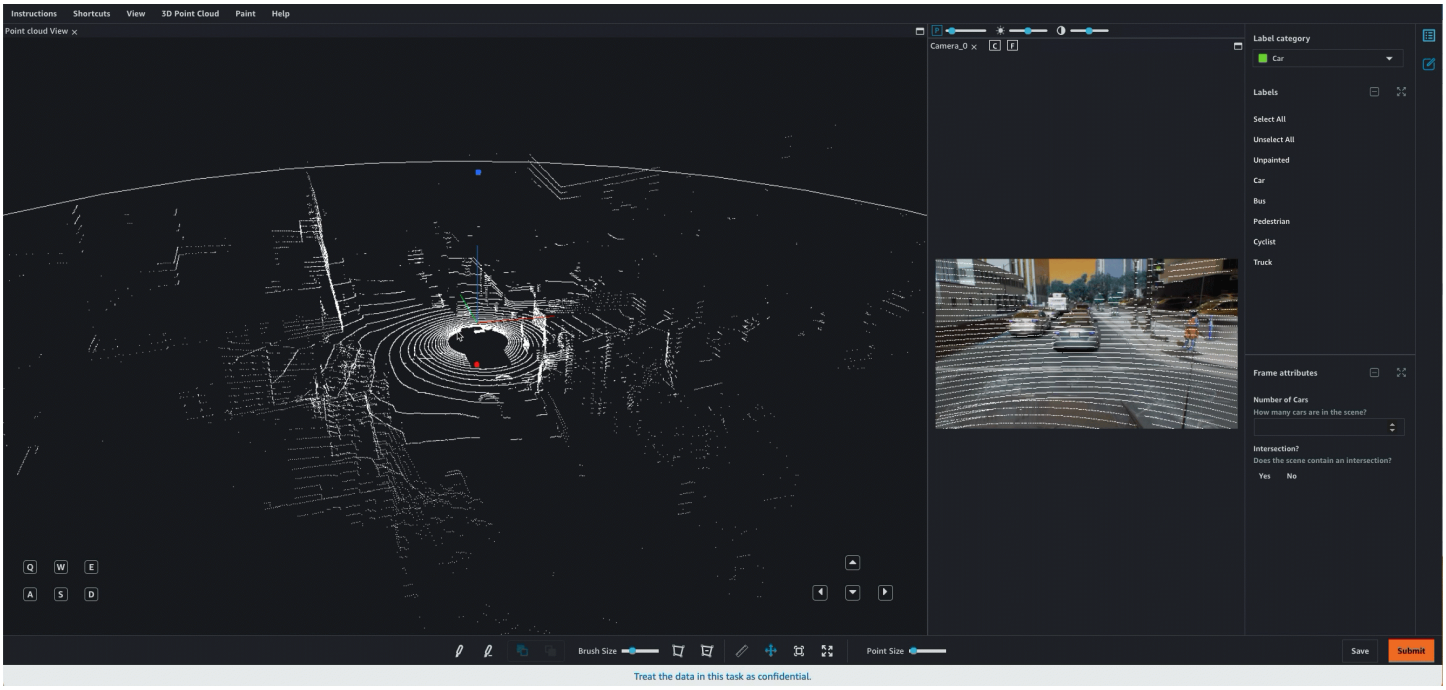
Gunakan Ground Truth untuk Label 3D Point Clouds

Buat pekerjaan pelabelan awan titik 3D agar pekerja memberi label objek di awan titik 3D yang dihasilkan dari sensor 3D seperti sensor Deteksi Cahaya dan Rentang (LiDAR) dan kamera kedalaman, atau dihasilkan dari rekonstruksi 3D dengan menjahit gambar yang diambil oleh agen seperti drone.

3D Titik Awan

Awan titik terdiri dari data visual tiga dimensi (3D) yang terdiri dari titik-titik. Setiap titik dijelaskan menggunakan tiga koordinat, biasanya`x`,`y`, dan`z`. Untuk menambahkan warna atau variasi intensitas titik ke titik awan, poin dapat dijelaskan dengan atribut tambahan, seperti`i`untuk intensitas atau nilai untuk merah (`r`), hijau (g), dan biru (b) Saluran warna 8-bit. Saat Anda membuat pekerjaan pelabelan cloud titik Ground Truth 3D, Anda dapat menyediakan cloud titik dan, secara opsional, data fusi sensor.

Gambar berikut menunjukkan adegan awan titik 3D tunggal yang dirender oleh Ground Truth dan ditampilkan dalam UI pekerja segmentasi semantik.



LiDAR

Sensor Light Detection and Ranging (LiDAR) adalah jenis sensor umum yang digunakan untuk mengumpulkan pengukuran yang digunakan untuk menghasilkan data cloud titik. LiDAR adalah metode penginderaan jauh yang menggunakan cahaya dalam bentuk laser berdenyut untuk mengukur jarak objek dari sensor. Anda dapat menyediakan data cloud titik 3D yang dihasilkan dari sensor LiDAR untuk pekerjaan pelabelan cloud titik Ground Truth 3D menggunakan format data mentah yang dijelaskan diFormat Data 3D Mentah yang Diterima.

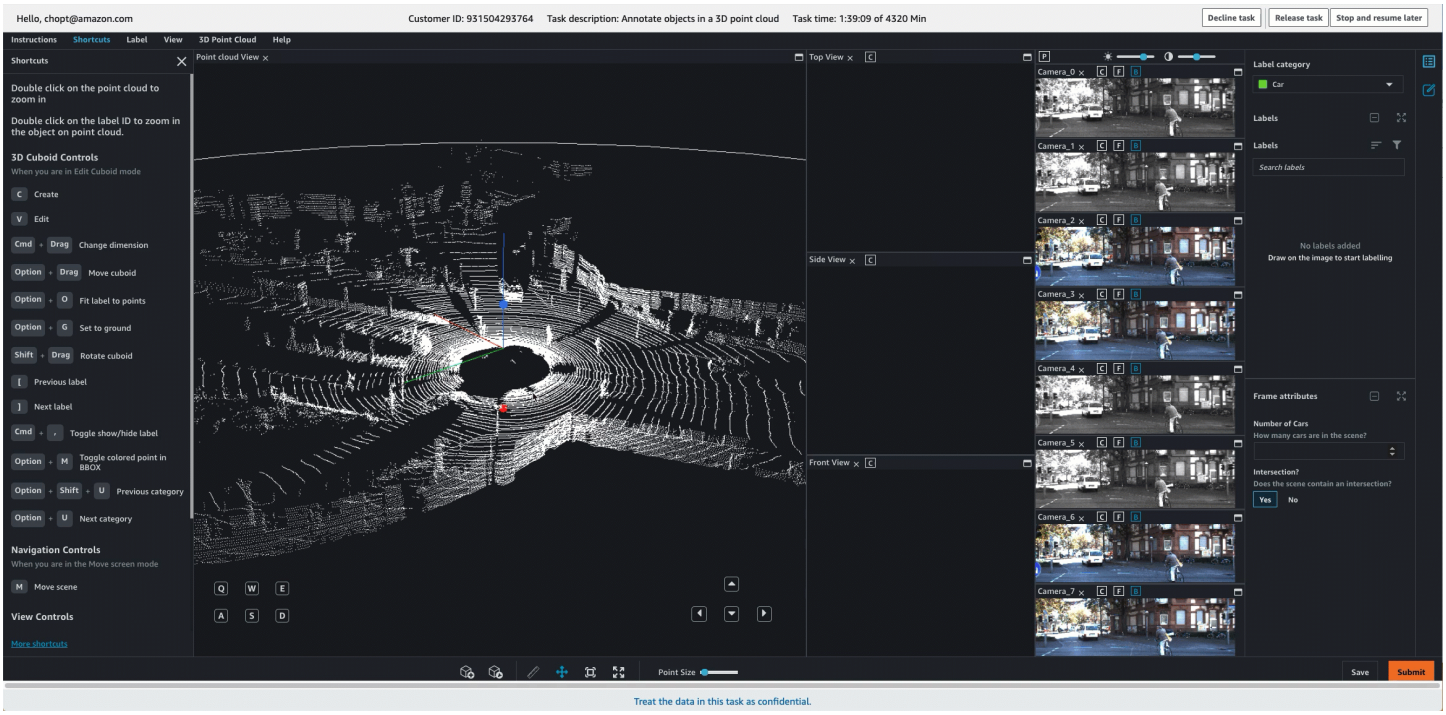
Sensor Fusion

Pekerjaan pelabelan cloud titik 3D Ground Truth mencakup fitur fusi sensor yang mendukung fusi sensor kamera video untuk semua jenis tugas. Beberapa sensor dilengkapi dengan beberapa perangkat LiDAR dan kamera video yang menangkap gambar dan mengaitkannya dengan bingkai LiDAR. Untuk membantu anotator menyelesaikan tugas Anda secara visual dengan keyakinan tinggi, Anda dapat menggunakan fitur fusi sensor Ground Truth untuk memproyeksikan anotasi (label) dari cloud titik 3D hingga gambar kamera 2D dan sebaliknya menggunakan matriks ekstrinsik pemindai 3D (seperti LiDAR) dan matriks ekstrinsik dan intrinsik kamera. Untuk mempelajari selengkapnya, lihat Sensor Fusion.

Label 3D Titik Awan

Ground Truth menyediakan antarmuka pengguna (UI) dan alat yang digunakan pekerja untuk memberi label ataumembuat keteranganAwan titik 3D. Saat Anda menggunakan jenis tugas deteksi objek atau segmentasi semantik, pekerja dapat membuat anotasi bingkai cloud titik tunggal. Bila Anda menggunakan pelacakan objek, pekerja membubuhi keterangan urutan frame. Anda dapat menggunakan pelacakan objek untuk melacak pergerakan objek di semua frame secara berurutan.

Berikut ini menunjukkan bagaimana pekerja akan menggunakan portal pekerja Ground Truth dan alat untuk membuat anotasi cloud titik 3D untuk tugas deteksi objek. Untuk contoh visual serupa dari jenis tugas lainnya, lihatJenis Tugas 3D Point Cloud.



Alat Pelabelan Bantu untuk Anotasi Point Cloud

Ground Truth menawarkan alat pelabelan bantu untuk membantu pekerja menyelesaikan tugas anotasi point cloud Anda lebih cepat dan dengan lebih akurat. Untuk detail tentang alat pelabelan bantu yang disertakan dalam UI pekerja untuk setiap jenis tugas,pilih jenis tugasdan merujuk padaMelihat Antarmuka Tugas Pekerjabagian dari halaman itu.

Langkah Selanjutnya

Anda dapat membuat enam jenis tugas saat Anda menggunakan pekerjaan pelabelan awan titik 3D Ground Truth. Gunakan topik diJenis Tugas 3D Point Cloudtentang hal inijenis tugasdan untuk mempelajari cara membuat pekerjaan pelabelan menggunakan jenis tugas pilihan Anda.

Pekerjaan pelabelan awan titik 3D berbeda dari modalitas pelabelan Ground Truth lainnya. Sebelum membuat pekerjaan pelabelan, kami rekomendasikan agar Anda membacaIkhtisar Pekerjaan Pelabelan Titik Cloud 3D. Selain itu, tinjau kuota data input diKuota Job Pelabelan 3D Point Cloud dan Bingkai Video.

Untuk end-to-end demo menggunakan SageMaker API danAWSPython SDK (boto 3) untuk membuat 3D titik awan pelabelan pekerjaan, lihatBuat-3D-pointcloud-labeling-job.ipynbdi dalamSageMaker Contoh tab notebook.

> **Important**
>
> Jika Anda menggunakan instance notebook yang dibuat sebelum 5 Juni 2020 untuk menjalankan notebook ini, Anda harus menghentikan dan memulai ulang instance notebook agar notebook berfungsi.

Topik

- Jenis Tugas 3D Point Cloud
- Ikhtisar Pekerjaan Pelabelan Titik Cloud 3D
- Instruksi Pekerja

Jenis Tugas 3D Point Cloud

Anda dapat menggunakan modalitas pelabelan awan titik 3D Ground Truth untuk berbagai kasus penggunaan. Daftar berikut menjelaskan secara singkat setiap jenis tugas awan titik 3D. Untuk detail tambahan dan petunjuk tentang cara membuat pekerjaan pelabelan menggunakan jenis tugas tertentu, pilih nama jenis tugas untuk melihat halaman jenis tugasnya.

- Deteksi objek awan titik 3D - Gunakan jenis tugas ini saat Anda ingin pekerja menemukan dan mengklasifikasikan objek dalam awan titik 3D dengan menambahkan dan memasang kubus 3D di sekitar objek.
- Pelacakan objek awan titik 3D - Gunakan jenis tugas ini saat Anda ingin pekerja menambahkan dan menyesuaikan kubus 3D di sekitar objek untuk melacak pergerakan mereka melintasi urutan bingkai awan titik 3D. Misalnya, Anda dapat menggunakan jenis tugas ini untuk meminta pekerja melacak pergerakan kendaraan melintasi beberapa titik cloud frame.

- Segmentasi semantik awan titik 3D - Gunakan jenis tugas ini ketika Anda ingin pekerja membuat topeng segmentasi semantik tingkat titik dengan mengecat objek di awan titik 3D menggunakan warna berbeda di mana setiap warna ditetapkan ke salah satu kelas yang Anda tentukan.
- Jenis tugas penyesuaian awan titik 3D - Masing-masing jenis tugas di atas memiliki jenis tugas penyesuaian terkait yang dapat Anda gunakan untuk mengaudit dan menyesuaikan anotasi yang dihasilkan dari pekerjaan pelabelan cloud titik 3D. Lihat halaman jenis tugas dari jenis terkait untuk mempelajari cara membuat pekerjaan pelabelan penyesuaian untuk tugas itu.

Deteksi Objek Awan Titik 3D

Gunakan jenis tugas ini ketika Anda ingin pekerja untuk mengklasifikasikan objek dalam awan titik 3D dengan menggambar kubus 3D di sekitar objek. Misalnya, Anda dapat menggunakan jenis tugas ini untuk meminta pekerja mengidentifikasi berbagai jenis objek di titik awan, seperti mobil, sepeda, dan pejalan kaki.

Untuk jenis tugas ini,Obyek databahwa label pekerja adalah bingkai cloud titik tunggal. Ground Truth menjadikan cloud titik 3D menggunakan data cloud titik yang Anda berikan. Anda juga dapat menyediakan data kamera untuk memberi pekerja lebih banyak informasi visual tentang pemandangan dalam bingkai, dan untuk membantu pekerja menggambar kubus 3D di sekitar objek.

Ground Truth penyedia pekerja dengan alat untuk membubuhi keterangan objek dengan 9 derajat kebebasan (x, y, z, rx, ry, rz, l, w, h) dalam tiga dimensi di kedua adegan 3D dan tampilan sisi yang diproyeksikan (atas, samping, dan belakang). Jika Anda memberikan informasi fusi sensor (seperti data kamera), saat pekerja menambahkan cuboid untuk mengidentifikasi objek di awan titik 3D, cuboid muncul dan dapat dimodifikasi dalam gambar 2D. Setelah cuboid ditambahkan, semua pengeditan dilakukan pada cuboid itu dalam adegan 2D atau 3D diproyeksikan ke tampilan lain.

Anda dapat membuat pekerjaan untuk menyesuaikan anotasi yang dibuat dalam pekerjaan pelabelan deteksi objek awan titik 3D menggunakan jenis tugas penyesuaian objek titik awan titik 3D.

Jika Anda adalah pengguna baru modalitas pelabelan cloud titik 3D Ground Truth, kami sarankan Anda meninjauIkhtisar Pekerjaan Pelabelan Titik Cloud 3D. Modalitas pelabelan ini berbeda dari jenis tugas Ground Truth lainnya, dan halaman ini memberikan ikhtisar detail penting yang harus Anda ketahui saat membuat pekerjaan pelabelan cloud titik 3D.

Topik

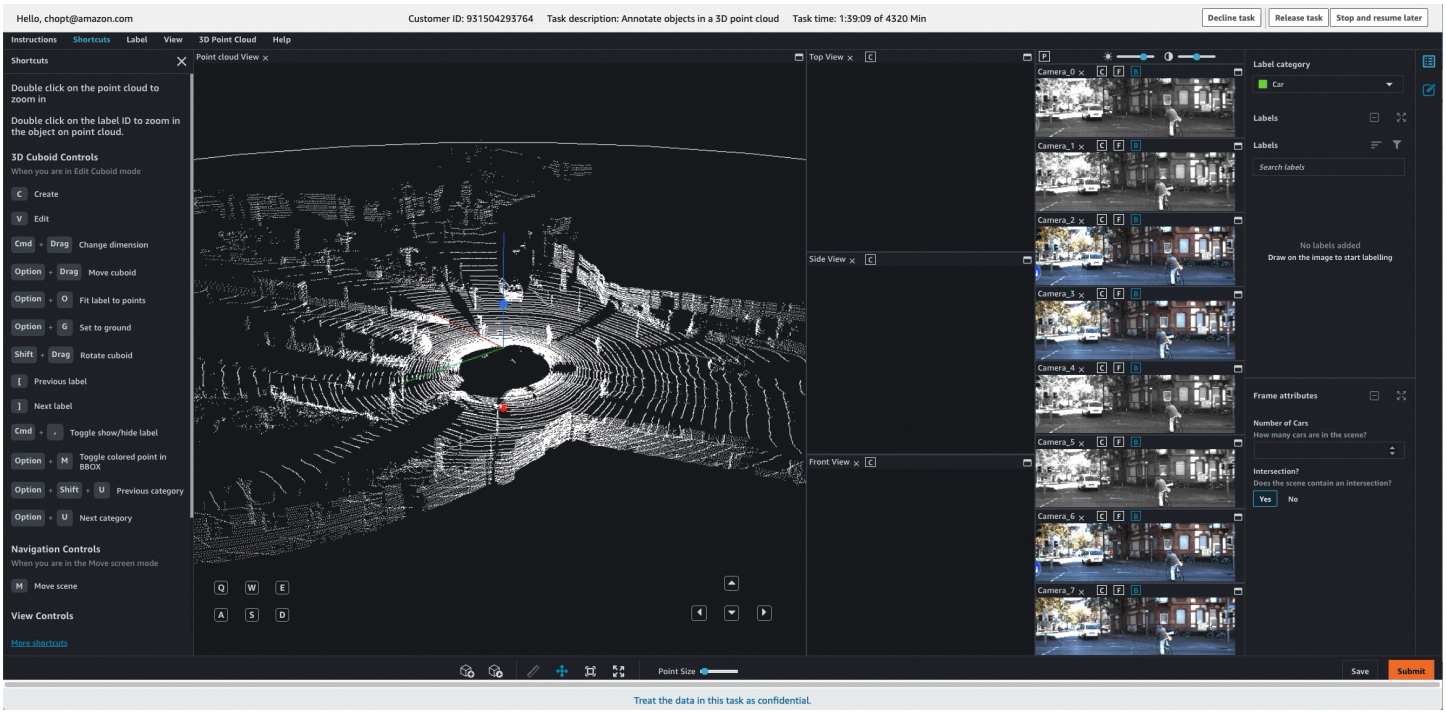
- Melihat Antarmuka Tugas Pekerja
- Buat Job Pelabelan Deteksi Objek 3D Point Cloud

- Membuat 3D Point Cloud Object Detection Adjustment atau Verifikasi Labeling
- Format Data Output

Melihat Antarmuka Tugas Pekerja

Ground Truth menyediakan pekerja dengan portal web dan alat untuk menyelesaikan tugas penjelasan deteksi objek titik awan 3D Anda. Saat membuat tugas pelabelan, Anda menyediakan Amazon Resource Name (ARN) untuk Ground Truth worker UI yang telah dibuat sebelumnya di`HumanTaskUiArn`parameter. Saat Anda membuat pekerjaan pelabelan menggunakan jenis tugas ini di konsol, UI pekerja ini akan digunakan secara otomatis. Anda dapat melihat pratinjau dan berinteraksi dengan UI pekerja saat membuat pekerjaan pelabelan di konsol. Jika Anda adalah pengguna baru, disarankan agar Anda membuat pekerjaan pelabelan menggunakan konsol untuk memastikan atribut label Anda, bingkai titik awan, dan jika berlaku, gambar, muncul seperti yang diharapkan.

Berikut ini adalah GIF dari antarmuka tugas pekerja deteksi objek titik awan 3D. Jika Anda menyediakan data kamera untuk fusi sensor dalam sistem koordinat dunia, gambar dicocokkan dengan pemandangan dalam bingkai titik awan. Gambar-gambar ini muncul di portal pekerja seperti yang ditunjukkan pada GIF berikut.

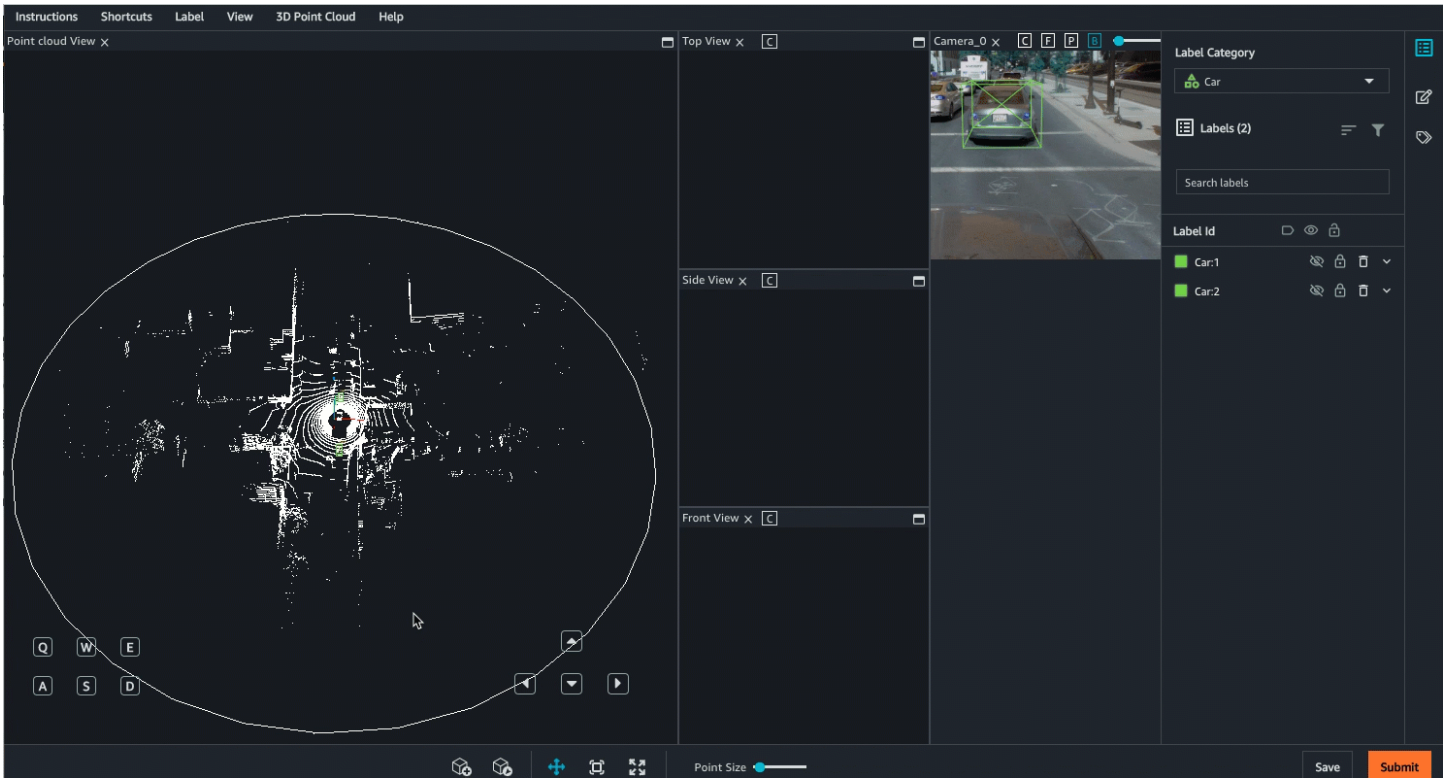


Pekerja dapat menavigasi dalam adegan 3D menggunakan keyboard dan mouse mereka. Mereka dapat:

- Klik dua kali pada objek tertentu di titik awan untuk memperbesarnya.
- Gunakan mouse-scroller atau trackpad untuk memperbesar dan memperkecil titik awan.
- Gunakan kedua tombol panah keyboard dan tombol Q, E, A, dan D untuk bergerak Naik, Bawah, Kiri, Kanan. Gunakan tombol keyboard W dan S untuk memperbesar dan memperkecil.

Setelah pekerja menempatkan cuboid dalam adegan 3D, tampilan samping akan muncul dengan tiga tampilan samping yang diproyeksikan: atas, samping, dan belakang. Pandangan samping ini menunjukkan titik-titik di dalam dan di sekitar kubus yang ditempatkan dan membantu pekerja memperbaiki batas berbentuk kubus di daerah itu. Pekerja dapat memperbesar dan memperkecil masing-masing tampilan samping menggunakan mouse mereka.

Video berikut menunjukkan pergerakan di sekitar awan titik 3D dan di tampilan samping.



Opsi dan fitur tampilan tambahan tersedia diLihatmenu di UI pekerja. Lihatlaman instruksi pekerjauntuk ikhtisar komprehensif UI Pekerja.

Alat Pelabelan Bantu

Ground Truth membantu pekerja membuat anotasi awan titik 3D lebih cepat dan lebih akurat menggunakan pembelajaran mesin dan alat pelabelan bantu bertenaga visi komputer untuk tugas pelacakan objek cloud titik 3D. Alat pelabelan bantu berikut tersedia untuk jenis tugas ini:

- Gertakan- Pekerja dapat menambahkan berbentuk kubus di sekitar objek dan menggunakan pintasan keyboard atau opsi menu untuk memiliki alat autofit Ground Truth menjentikkan kubus dengan erat di sekitar objek.
- Atur ke ground- Setelah seorang pekerja menambahkan berbentuk kubus ke adegan 3D, pekerja dapat secara otomatis menjentikkan kubus ke tanah. Misalnya, pekerja dapat menggunakan fitur ini untuk mengambil cuboid ke jalan atau trotoar di tempat kejadian.
- Pelabelan multi-tampilan- Setelah seorang pekerja menambahkan kubus 3D ke adegan 3D, panel samping menampilkan perspektif depan, samping, dan atas untuk membantu pekerja menyesuaikan berbentuk kubus dengan erat di sekitar objek. Dalam semua pandangan ini, berbentuk kubus menyertakan panah yang menunjukkan orientasi, atau judul objek. Ketika pekerja menyesuaikan berbentuk kubus, penyesuaian akan muncul secara real time pada semua tampilan (yaitu, 3D, atas, samping, dan depan).
- Sensor fusi- Jika Anda menyediakan data untuk fusi sensor, pekerja dapat menyesuaikan anotasi dalam adegan 3D dan dalam gambar 2D, dan anotasi akan diproyeksikan ke tampilan lain secara real time. Selain itu, pekerja akan memiliki pilihan untuk melihat arah kamera menghadap dan kamera frustum.
- Lihat opsi- Memungkinkan pekerja untuk dengan mudah menyembunyikan atau melihat kubus, teks label, jaring tanah, dan atribut titik tambahan seperti warna atau intensitas. Pekerja juga dapat memilih antara proyeksi perspektif dan ortogonal.

Buat Job Pelabelan Deteksi Objek 3D Point Cloud

Anda dapat membuat pekerjaan pelabelan cloud titik 3D menggunakan SageMaker konsol atau operasi API,`CreateLabelingJob`. Untuk membuat pekerjaan pelabelan untuk jenis tugas ini, Anda memerlukan yang berikut:

- Sebuah file manifes masukan single-frame. Untuk mempelajari cara membuat file manifes jenis ini, lihatMembuat File Manifes Input Point Cloud Frame. Jika Anda adalah pengguna baru modalitas pelabelan cloud titik Ground Truth 3D, Anda mungkin juga ingin meninjauFormat Data 3D Mentah yang Diterima.
- Sebuah tim kerja dari tenaga kerja swasta atau vendor. Anda tidak dapat menggunakan Amazon Mechanical Turk untuk pekerjaan pelabelan bingkai video. Untuk mempelajari cara membuat tenaga kerja dan tim kerja, lihatMembuat dan Mengelola Tenaga Kerja.

Selain itu, pastikan bahwa Anda telah meninjau dan memuaskanTetapkan Izin IAM untuk Menggunakan Ground Truth.

Gunakan salah satu bagian berikut untuk mempelajari cara membuat pekerjaan pelabelan menggunakan konsol atau API.

Buat Job Pelabelan (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan deteksi objek titik awan 3D di SageMaker konsol. Saat membuat pekerjaan pelabelan Anda, perhatikan hal-hal berikut:

- File manifes masukan Anda harus berupa file manifes satu bingkai. Untuk informasi selengkapnya, lihat Membuat File Manifes Input Point Cloud Frame.
- Secara opsional, Anda dapat memberikan kategori label dan atribut bingkai. Pekerja dapat menetapkan satu atau beberapa atribut ini ke anotasi untuk memberikan informasi lebih lanjut tentang objek tersebut. Misalnya, Anda mungkin ingin menggunakan atributtersumbatagar pekerja mengidentifikasi kapan suatu objek terhalang sebagian.
- Pelabelan data otomatis dan konsolidasi anotasi tidak didukung untuk tugas pelabelan cloud titik 3D.
- Pekerjaan pelabelan deteksi objek awan titik 3D dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini ketika Anda memilih tim kerja Anda (hingga 7 hari, atau 604800 detik).

Buat Job Pelabelan (API)

Bagian ini mencakup detail yang perlu Anda ketahui saat membuat pekerjaan pelabelan menggunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian dari`CreateLabelingJob`.

Buat Job Pelabelan (API), memberikan ikhtisar tentang`CreateLabelingJob`operasi. Ikuti petunjuk ini dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Anda harus memasukkan ARN untuk`HumanTaskUiArn`. Gunakan `arn:aws:sagemaker:<region>:394669845002:human-task-ui/PointCloudObjectDetection`. Ganti*<region>*denganAWSWilayah tempat Anda membuat pekerjaan pelabelan.

  Seharusnya tidak ada entri untuk`UiTemplateS3Uri`parameter.

- File manifes masukan Anda harus berupa file manifes satu bingkai. Untuk informasi selengkapnya, lihat Membuat File Manifes Input Point Cloud Frame.
- Anda menentukan label, kategori label, dan atribut bingkai, serta instruksi pekerja dalam file konfigurasi kategori label. Untuk mempelajari cara membuat file ini, lihatBuat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai.
- Anda perlu menyediakan ARN yang telah ditentukan sebelumnya untuk fungsi Lambda pra-anotasi dan pasca-anotasi (ACS). ARN ini khusus untukAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda.
  - Untuk menemukan pra-anotasi Lambda ARN, lihat`PreHumanTaskLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar. Misalnya, jika Anda membuat pekerjaan pelabelan Anda di us-timur-1, ARN akan menjadi`arn:aws:lambda:us-east-1:432418664414:function:PRE-3DPointCloudObjectDetection`.
  - Untuk menemukan pasca-anotasi Lambda ARN, lihat`AnnotationConsolidationLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar. Misalnya, jika Anda membuat pekerjaan pelabelan Anda di us-timur-1, ARN akan menjadi`arn:aws:lambda:us-east-1:432418664414:function:ACS-3DPointCloudObjectDetection`.
- Jumlah pekerja yang ditentukan dalam`NumberOfHumanWorkersPerDataObject`harus`1`.
- Pelabelan data otomatis tidak didukung untuk pekerjaan pelabelan cloud titik 3D. Anda tidak boleh menentukan nilai untuk parameter di`LabelingJobAlgorithmsConfig`.
- Pekerjaan pelabelan deteksi objek awan titik 3D dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini`TaskTimeLimitInSeconds`(hingga 7 hari, atau 604.800 detik).

Membuat 3D Point Cloud Object Detection Adjustment atau Verifikasi Labeling

Anda dapat membuat pekerjaan pelabelan penyesuaian atau verifikasi menggunakan konsol Ground Truth atau`CreateLabelingJob`API. Untuk mempelajari lebih lanjut tentang penyesuaian dan verifikasi pekerjaan pelabelan, dan untuk mempelajari cara membuatnya, lihatVerifikasi dan Sesuaikan Label.

Saat Anda membuat pekerjaan pelabelan penyesuaian, data input Anda ke pekerjaan pelabelan dapat mencakup label, dan pengukuran yaw, pitch, dan roll dari pekerjaan pelabelan sebelumnya atau sumber eksternal. Dalam pekerjaan penyesuaian, pitch, dan roll akan divisualisasikan dalam UI pekerja, tetapi tidak dapat dimodifikasi. Yaw dapat disesuaikan.

Ground Truth menggunakan sudut Tait-Bryan dengan rotasi intrinsik berikut untuk memvisualisasikan yaw, pitch and roll di UI pekerja. Pertama, rotasi diterapkan pada kendaraan sesuai dengan sumbu z (yaw). Selanjutnya, kendaraan yang diputar diputar sesuai dengan sumbu Anda (pitch) intrinsik. Akhirnya, kendaraan diputar sesuai dengan intrinsik x" -axis (roll).

Format Data Output

Saat Anda membuat pekerjaan pelabelan deteksi objek awan titik 3D, tugas dikirim ke pekerja. Ketika pekerja ini menyelesaikan tugas mereka, label ditulis ke bucket Amazon S3 yang Anda tentukan saat Anda membuat pekerjaan pelabelan. Format data keluaran menentukan apa yang Anda lihat di bucket Amazon S3 saat status pekerjaan pelabelan Anda (LabelingJobStatus) adalah`Completed`.

Jika Anda adalah pengguna baru Ground Truth, lihatData Outputuntuk mempelajari lebih lanjut tentang format data keluaran Ground Truth. Untuk mempelajari tentang format data keluaran deteksi objek awan titik 3D, lihatOutput Deteksi Objek Awan Titik 3D.

Pelacakan Objek Awan Titik 3D

Gunakan jenis tugas ini ketika Anda ingin pekerja menambahkan dan menyesuaikan kubus 3D di sekitar objek untuk melacak gerakan mereka di bingkai awan titik 3D. Misalnya, Anda dapat menggunakan jenis tugas ini untuk meminta pekerja melacak pergerakan kendaraan melintasi beberapa titik cloud frame.

Untuk jenis tugas ini, objek data yang label pekerja adalah urutan frame titik awan. Urutan didefinisikan sebagai serangkaian temporal frame titik awan. Ground Truth membuat serangkaian visualisasi awan titik 3D menggunakan urutan yang Anda berikan dan pekerja dapat beralih di antara frame cloud titik 3D ini di antarmuka tugas pekerja.

Pekerja penyedia Ground Truth dengan alat untuk membubuhi keterangan objek dengan 9 derajat kebebasan: (x, y, z, rx, ry, rz, l, w, h) dalam tiga dimensi di kedua adegan 3D dan tampilan sisi yang diproyeksikan (atas, samping, dan belakang). Ketika seorang pekerja menggambar berbentuk kubus di sekitar objek, berbentuk kubus itu diberi ID unik, misalnya`Car:1` untuk satu mobil dalam urutan dan`Car:2` untuk yang lain. Pekerja menggunakan ID tersebut untuk memberi label pada objek yang sama dalam beberapa frame.

Anda juga dapat menyediakan data kamera untuk memberi pekerja lebih banyak informasi visual tentang pemandangan dalam bingkai, dan untuk membantu pekerja menggambar kubus 3D di sekitar objek. Ketika pekerja menambahkan kubus 3D untuk mengidentifikasi objek baik dalam gambar 2D atau awan titik 3D, dan berbentuk kubus muncul di tampilan lain.

Anda dapat menyesuaikan anotasi yang dibuat dalam pekerjaan pelabelan deteksi objek awan titik 3D menggunakan jenis tugas penyesuaian pelacakan objek titik awan 3D.

Jika Anda adalah pengguna baru modalitas pelabelan cloud titik Ground Truth 3D, kami sarankan Anda meninjauIkhtisar Pekerjaan Pelabelan Titik Cloud 3D. Modalitas pelabelan ini berbeda dari jenis tugas Ground Truth lainnya, dan halaman ini memberikan ikhtisar detail penting yang harus Anda ketahui saat membuat pekerjaan pelabelan cloud titik 3D.

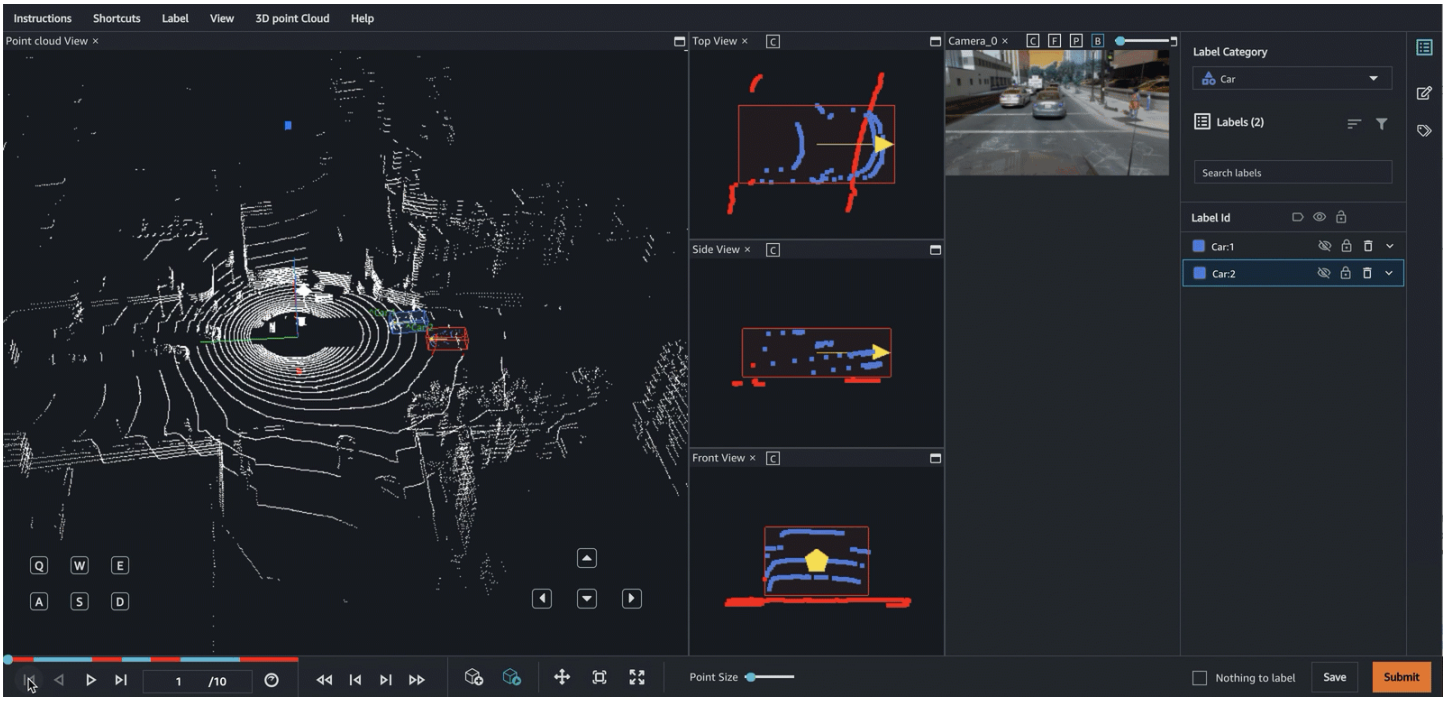
Topik

- Melihat Antarmuka Tugas Pekerja
- Buat Job Pelabelan Pelacakan Objek Titik Cloud 3D
- Membuat 3D Point Cloud Object Tracking Adjustment atau Verifikasi Pelabelan Job
- Format Data Output

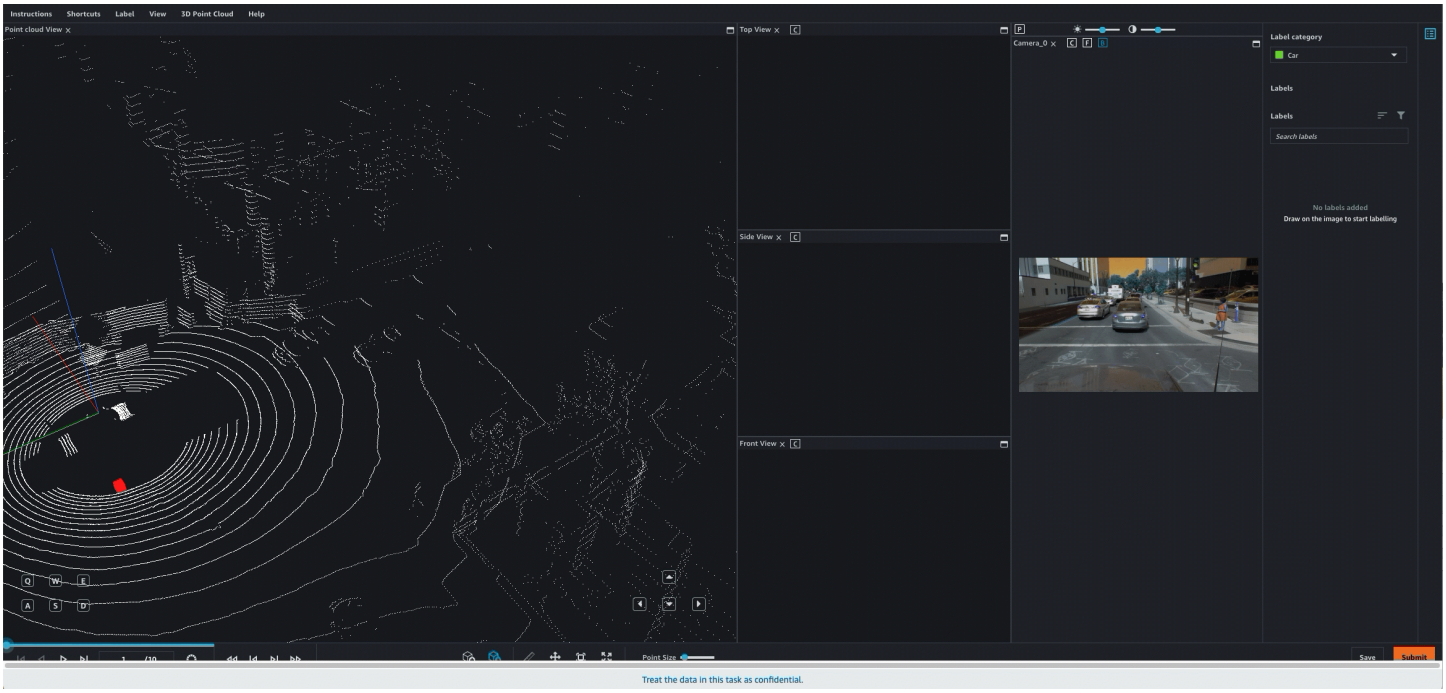
Melihat Antarmuka Tugas Pekerja

Ground Truth menyediakan pekerja dengan portal web dan alat untuk menyelesaikan tugas penjelasan pelacakan objek titik awan 3D Anda. Ketika Anda membuat pekerjaan pelabelan, Anda menyediakan Amazon Resource Name (ARN) untuk UI Ground Truth yang dibuat sebelumnya dalam`HumanTaskUiArn` parameter. Ketika Anda membuat pekerjaan pelabelan menggunakan jenis tugas ini di konsol, UI ini secara otomatis digunakan. Anda dapat melihat pratinjau dan berinteraksi dengan UI pekerja saat membuat pekerjaan pelabelan di konsol. Jika Anda adalah penggunaan baru, disarankan agar Anda membuat pekerjaan pelabelan menggunakan konsol untuk memastikan atribut label Anda, bingkai titik awan, dan jika berlaku, gambar, muncul seperti yang diharapkan.

Berikut ini adalah GIF dari antarmuka tugas pekerja pelacakan objek awan titik 3D dan menunjukkan bagaimana pekerja dapat menavigasi frame titik awan dalam urutan. Alat anotasi adalah bagian dari antarmuka tugas pekerja. Mereka tidak tersedia untuk antarmuka pratinjau.



Setelah pekerja menambahkan kubus tunggal, berbentuk kubus yang direplikasi dalam semua frame dari urutan dengan ID yang sama. Setelah pekerja menyesuaikan berbentuk kubus di bingkai lain, Ground Truth akan menginterpolasi pergerakan objek itu dan menyesuaikan semua kubus di antara bingkai yang disesuaikan secara manual. GIF berikut menunjukkan fitur interpolasi ini. Di bilah navigasi di kiri bawah, area merah menunjukkan bingkai yang disesuaikan secara manual.



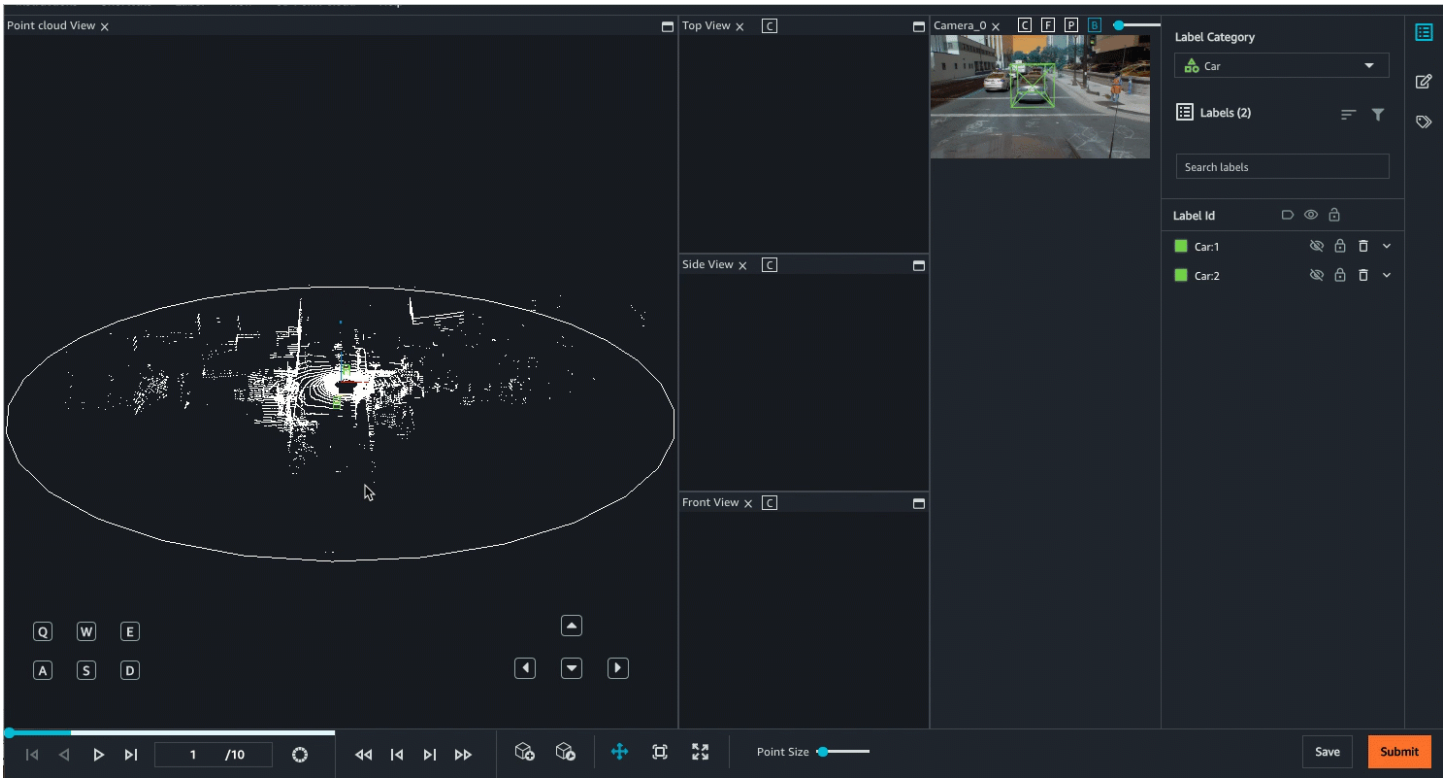
Jika Anda menyediakan data kamera untuk fusi sensor, gambar dicocokkan dengan pemandangan dalam bingkai cloud titik. Gambar-gambar ini muncul di portal pekerja seperti yang ditunjukkan pada GIF berikut.

Pekerja dapat menavigasi dalam adegan 3D menggunakan keyboard dan mouse mereka. Mereka dapat:

- Klik dua kali pada objek tertentu di titik awan untuk memperbesarnya.
- Gunakan mouse-scroller atau trackpad untuk memperbesar dan memperkecil titik awan.
- Gunakan kedua tombol panah keyboard dan tombol Q, E, A, dan D untuk bergerak Naik, Bawah, Kiri, Kanan. Gunakan tombol keyboard W dan S untuk memperbesar dan memperkecil.

Setelah pekerja menempatkan cuboids dalam adegan 3D, tampilan samping akan muncul dengan tiga tampilan samping yang diproyeksikan: atas, samping, dan belakang. Pandangan samping ini menunjukkan titik-titik di dalam dan di sekitar kubus yang ditempatkan dan membantu pekerja memperbaiki batas berbentuk kubus di daerah itu. Pekerja dapat memperbesar dan memperkecil masing-masing tampilan samping menggunakan mouse mereka.

Video berikut menunjukkan pergerakan di sekitar awan titik 3D dan di tampilan samping.



Opsi dan fitur tampilan tambahan tersedia. Lihat halaman instruksi pekerja untuk ikhtisar komprehensif UI Pekerja.

Alat Pekerja

Pekerja dapat menavigasi melalui cloud titik 3D dengan memperbesar dan memperkecil, dan bergerak ke segala arah di sekitar awan menggunakan pintasan mouse dan keyboard. Jika pekerja mengklik titik di titik awan, UI akan secara otomatis memperbesar area tersebut. Pekerja dapat menggunakan berbagai alat untuk menggambar berbentuk kubus 3D di sekitar objek. Untuk informasi selengkapnya, lihat Alat Pelabelan Assistive.

Setelah pekerja menempatkan kubus 3D di titik awan, mereka dapat menyesuaikan kubus ini agar pas di sekitar mobil menggunakan berbagai tampilan: langsung di kubus 3D, dalam tampilan samping yang menampilkan tiga perspektif zoom in dari titik awan di sekitar kotak, dan jika Anda menyertakan gambar untuk fusi sensor, langsung di gambar 2D.

Lihat opsi yang memungkinkan pekerja menyembunyikan atau melihat teks label, ground mesh, dan atribut titik tambahan dengan mudah. Pekerja juga dapat memilih antara proyeksi perspektif dan ortogonal.

Alat Pelabelan Bantu

Ground Truth membantu pekerja membuat anotasi awan titik 3D lebih cepat dan lebih akurat menggunakan UX, pembelajaran mesin, dan alat pelabelan bantu bertenaga penglihatan komputer untuk tugas pelacakan objek cloud titik 3D. Alat pelabelan bantu berikut tersedia untuk jenis tugas ini:

- Isi otomatis label - Saat pekerja menambahkan berbentuk kubus ke bingkai, cuboid dengan dimensi dan orientasi yang sama secara otomatis ditambahkan ke semua frame dalam urutan.
- Interpolasi label - Setelah seorang pekerja memberi label satu objek dalam dua bingkai, Ground Truth menggunakan anotasi tersebut untuk menginterpolasi pergerakan objek itu di antara dua bingkai tersebut. Interpolasi label dapat dinyalakan dan dimatikan.
- Pengelolaan label dan atribut massal - Pekerja dapat menambahkan, menghapus, dan mengganti nama anotasi, atribut kategori label, dan atribut bingkai secara massal.
  - Pekerja dapat secara manual menghapus anotasi untuk objek tertentu sebelum atau sesudah bingkai. Misalnya, pekerja dapat menghapus semua label untuk objek setelah frame 10 jika objek tersebut tidak lagi terletak di adegan setelah frame tersebut.
  - Jika pekerja secara tidak sengaja menghapus semua anotasi untuk objek secara massal, mereka dapat menambahkannya kembali. Misalnya, jika pekerja menghapus semua anotasi

untuk objek sebelum frame 100, mereka dapat menambahkannya secara massal ke frame tersebut.

  - Pekerja dapat mengganti nama label dalam satu bingkai dan semua kuboid 3D yang ditetapkan bahwa label diperbarui dengan nama baru di semua frame.
  - Pekerja dapat menggunakan pengeditan massal untuk menambahkan atau mengedit atribut kategori label dan atribut bingkai dalam beberapa bingkai.
- Snapping - Pekerja dapat menambahkan berbentuk kubus di sekitar objek dan menggunakan pintasan keyboard atau opsi menu agar alat autofit Ground Truth menjentikkan kubus dengan erat di sekitar batas objek.
- Fit to ground - Setelah seorang pekerja menambahkan berbentuk kubus ke adegan 3D, pekerja dapat secara otomatis menjepret berbentuk kubus ke tanah. Misalnya, pekerja dapat menggunakan fitur ini untuk mengambil cuboid ke jalan atau trotoar di tempat kejadian.
- Pelabelan multi-tampilan - Setelah seorang pekerja menambahkan berbentuk kubus 3D ke adegan 3D, panel samping menampilkan perspektif depan dan dua sisi untuk membantu pekerja menyesuaikan berbentuk kubus dengan erat di sekitar objek. Pekerja dapat membuat anotasi cloud titik 3D, panel samping dan penyesuaian muncul di tampilan lain secara real time.
- Sensor fusi — Jika Anda menyediakan data untuk fusi sensor, pekerja dapat menyesuaikan anotasi dalam adegan 3D dan dalam gambar 2D, dan anotasi akan diproyeksikan ke tampilan lain secara real time.
- Gabungan otomatis kuboid - Pekerja dapat secara otomatis menggabungkan dua kuboid di semua frame jika mereka menentukan bahwa kuboid dengan label berbeda sebenarnya mewakili satu objek.
- Opsi tampilan - Memungkinkan pekerja untuk dengan mudah menyembunyikan atau melihat teks label, jaring tanah, dan atribut titik tambahan seperti warna atau intensitas. Pekerja juga dapat memilih antara proyeksi perspektif dan ortogonal.

Buat Job Pelabelan Pelacakan Objek Titik Cloud 3D

Anda dapat membuat pekerjaan pelabelan cloud titik 3D menggunakan SageMaker konsol atau operasi API, `CreateLabelingJob`. Untuk membuat pekerjaan pelabelan untuk jenis tugas ini, Anda memerlukan yang berikut:

- Sebuah file manifes masukan urutan. Untuk mempelajari cara membuat file manifes jenis ini, lihatMembuat Manifes Input Urutan Point Cloud. Jika Anda adalah pengguna baru modalitas

pelabelan cloud titik Ground Truth 3D, kami sarankan Anda meninjauFormat Data 3D Mentah yang Diterima.

- Sebuah tim kerja dari tenaga kerja swasta atau vendor. Anda tidak dapat menggunakan Amazon Mechanical Turk untuk pekerjaan pelabelan cloud titik 3D. Untuk mempelajari cara membuat tenaga kerja dan tim kerja, lihatMembuat dan Mengelola Tenaga Kerja.

Selain itu, pastikan bahwa Anda telah meninjau dan memuaskanTetapkan Izin IAM untuk Menggunakan Ground Truth.

Untuk mempelajari cara membuat pekerjaan pelabelan menggunakan konsol atau API, lihat bagian berikut.

Buat Job Pelabelan (API)

Bagian ini mencakup detail yang perlu Anda ketahui saat membuat pekerjaan pelabelan menggunakan operasi SageMaker API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWS SDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjau bagian Lihat Juga **`CreateLabelingJob`**.

Buat Job Pelabelan (API)memberikan gambaran umum tentang`CreateLabelingJob` operasi. Ikuti petunjuk ini dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Anda harus memasukkan ARN untuk`HumanTaskUiArn`. Gunakan `arn:aws:sagemaker:<region>:394669845002:human-task-ui/PointCloudObjectTracking`. Ganti*`<region>`* denganAWS Wilayah tempat Anda membuat pekerjaan pelabelan.

  Seharusnya tidak ada entri untuk`UiTemplateS3Uri` parameter.
- Anda `LabelAttributeName`harus berakhir`-ref`. Sebagai contoh, *`ot-labels`*`-ref`.
- File manifes masukan Anda harus berupa file manifes urutan bingkai titik awan. Untuk informasi selengkapnya, lihat Membuat Manifes Input Urutan Point Cloud.
- Anda menentukan label, kategori label, dan atribut bingkai, serta instruksi pekerja dalam file konfigurasi kategori label. Untuk informasi selengkapnya, lihatBuat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai untuk mempelajari cara membuat file ini.
- Anda perlu menyediakan ARN yang telah ditentukan sebelumnya untuk fungsi Lambda pra-anotasi dan pasca-anotasi (ACS). ARN ini khusus untukAWS Wilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda.

  - Untuk menemukan pra-anotasi Lambda ARN, lihat `PreHumanTaskLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri dengan`PRE-3DPointCloudObjectTracking`.
  - Untuk menemukan pasca-anotasi Lambda ARN, lihat `AnnotationConsolidationLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri dengan`ACS-3DPointCloudObjectTracking`.
- Jumlah pekerja yang ditentukan`NumberOfHumanWorkersPerDataObject` harus`1`.
- Pelabelan data otomatis tidak didukung untuk pekerjaan pelabelan cloud titik 3D. Anda tidak harus menentukan nilai untuk parameter di`LabelingJobAlgorithmsConfig`.
- Pekerjaan pelabelan pelacakan objek cloud titik 3D dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini`TaskTimeLimitInSeconds` (hingga 7 hari, atau 604.800 detik).

Buat Job Pelabelan (Konsol)

Anda dapat mengikuti petunjukMembuat Job Pelabelan (Konsol) untuk mempelajari cara membuat pekerjaan pelabelan pelacakan objek awan titik 3D di SageMaker konsol. Ketika Anda membuat pekerjaan pelabelan Anda, perhatikan hal-hal berikut:

- File manifes masukan Anda harus berupa file manifes urutan. Untuk informasi selengkapnya, lihat Membuat Manifes Input Urutan Point Cloud.
- Secara opsional, Anda dapat memberikan atribut kategori label. Pekerja dapat menetapkan satu atau beberapa atribut ini ke anotasi untuk memberikan informasi lebih lanjut tentang objek tersebut. Misalnya, Anda mungkin ingin menggunakan atribut yang tersumbat agar pekerja mengidentifikasi ketika suatu objek terhalang sebagian.
- Pelabelan data otomatis dan konsolidasi anotasi tidak didukung untuk tugas pelabelan cloud titik 3D.
- Pekerjaan pelabelan pelacakan objek cloud titik 3D dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini ketika Anda memilih tim kerja Anda (hingga 7 hari, atau 604800 detik).

Membuat 3D Point Cloud Object Tracking Adjustment atau Verifikasi Pelabelan Job

Anda dapat membuat pekerjaan pelabelan penyesuaian dan verifikasi menggunakan konsol Ground Truth atau`CreateLabelingJob` API. Untuk mempelajari lebih lanjut tentang penyesuaian dan verifikasi pekerjaan pelabelan, dan untuk mempelajari cara membuatnya, lihatVerifikasi dan Sesuaikan Label.

Saat Anda membuat pekerjaan pelabelan penyesuaian, data input Anda ke pekerjaan pelabelan dapat mencakup label, dan pengukuran yaw, pitch, dan roll dari pekerjaan pelabelan sebelumnya atau sumber eksternal. Dalam pekerjaan penyesuaian, pitch, dan roll akan divisualisasikan dalam UI pekerja, tetapi tidak dapat dimodifikasi. Yaw dapat disesuaikan.

Ground Truth menggunakan sudut Tait-Bryan dengan rotasi intrinsik berikut untuk memvisualisasikan yaw, pitch and roll di UI pekerja. Pertama, rotasi diterapkan pada kendaraan sesuai dengan sumbu z (yaw). Selanjutnya, kendaraan yang diputar diputar sesuai dengan sumbu Anda (pitch) intrinsik. Akhirnya, kendaraan diputar sesuai dengan intrinsik x” -axis (roll).

Format Data Output

Saat Anda membuat pekerjaan pelabelan pelacakan objek awan titik 3D, tugas dikirim ke pekerja. Ketika pekerja ini menyelesaikan tugas mereka, anotasi mereka ditulis ke bucket Amazon S3 yang Anda tentukan saat Anda membuat pekerjaan pelabelan. Format data keluaran menentukan apa yang Anda lihat di bucket Amazon S3 saat status pekerjaan pelabelan (LabelingJobStatus) Anda berada`Completed`.

Jika Anda adalah pengguna baru Ground Truth, lihatData Output untuk mempelajari lebih lanjut tentang format data keluaran Ground Truth. Untuk mempelajari tentang format data keluaran pelacakan objek awan titik 3D, lihatOutput Pelacakan Objek Awan Titik 3D.

Segmentasi Semantik Awan Titik 3D

Segmentasi semantik melibatkan mengklasifikasikan titik-titik individu dari awan titik 3D ke dalam kategori yang telah ditentukan sebelumnya. Gunakan jenis tugas ini ketika Anda ingin pekerja membuat topeng segmentasi semantik tingkat titik untuk awan titik 3D. Misalnya, jika Anda menentukan kelas`car`,`pedestrian`, dan`bike`, pekerja memilih satu kelas pada satu waktu, dan warna semua poin yang kelas ini berlaku untuk warna yang sama di titik awan.

Untuk jenis tugas ini, objek data yang label pekerja adalah bingkai cloud titik tunggal. Ground Truth menghasilkan visualisasi awan titik 3D menggunakan data cloud titik yang Anda berikan. Anda

juga dapat menyediakan data kamera untuk memberi pekerja lebih banyak informasi visual tentang pemandangan dalam bingkai, dan untuk membantu pekerja melukis objek. Ketika seorang pekerja melukis objek baik dalam gambar 2D atau awan titik 3D, cat muncul di tampilan lain.

Anda dapat menyesuaikan anotasi yang dibuat dalam pekerjaan pelabelan deteksi objek awan titik 3D menggunakan jenis tugas penyesuaian segmentasi semantik titik 3D.

Jika Anda adalah pengguna baru modalitas pelabelan cloud titik 3D Ground Truth, kami sarankan Anda meninjauIkhtisar Pekerjaan Pelabelan Titik Cloud 3D. Modalitas pelabelan ini berbeda dari jenis tugas Ground Truth lainnya, dan topik ini memberikan ikhtisar detail penting yang harus Anda perhatikan saat membuat pekerjaan pelabelan cloud titik 3D.

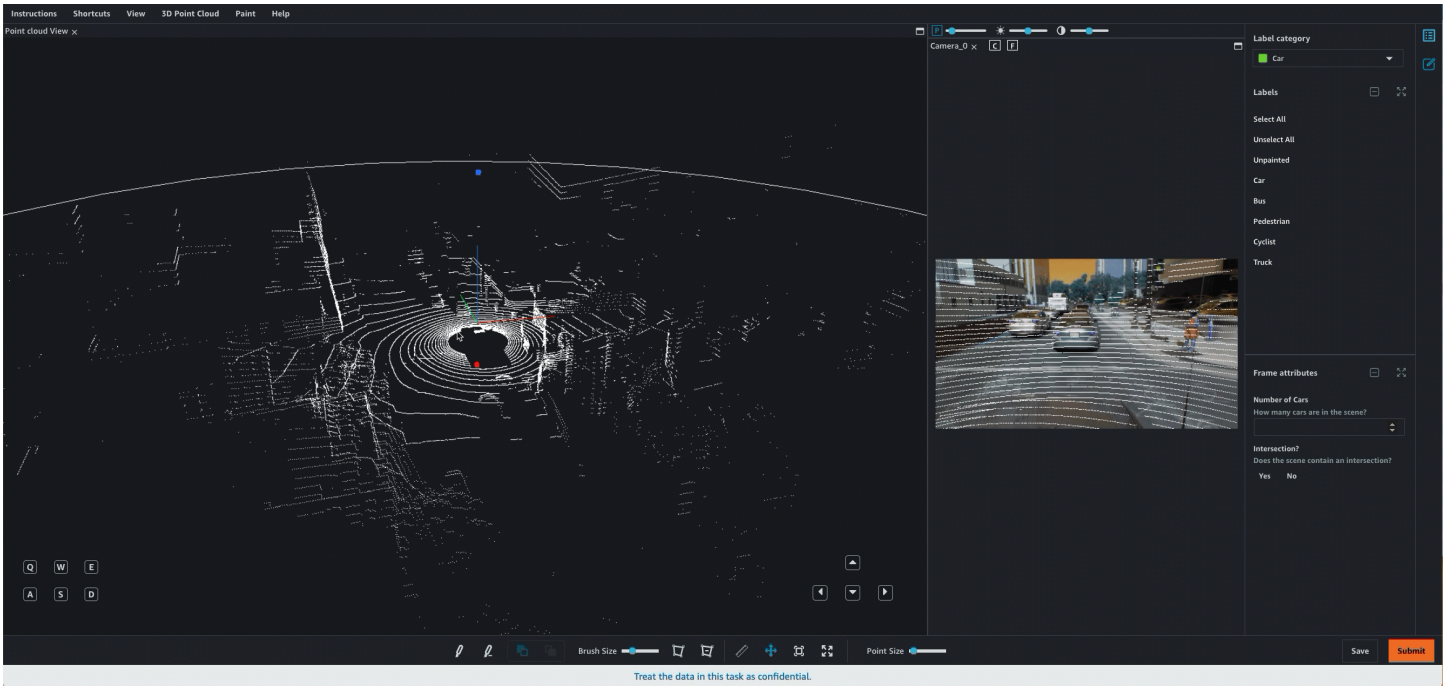
Topik

- Melihat Antarmuka Tugas Pekerja
- Buat Job Pelabelan Segmentasi Semantik Titik Cloud 3D
- Buat Penyesuaian Segmentasi Semantik Titik Cloud 3D atau Job Pelabelan Verifikasi
- Format Data Output

Melihat Antarmuka Tugas Pekerja

Ground Truth memberi pekerja portal web dan alat untuk menyelesaikan tugas anotasi segmentasi semantik titik 3D Anda. Saat membuat tugas pelabelan, berikan Amazon Resource Name (ARN) untuk Ground Truth UI yang dibuat sebelumnya di`HumanTaskUiArn`parameter. Ketika Anda membuat pekerjaan pelabelan menggunakan jenis tugas ini di konsol, UI ini secara otomatis digunakan. Anda dapat melihat pratinjau dan berinteraksi dengan UI pekerja saat membuat pekerjaan pelabelan di konsol. Jika Anda adalah penggunaan baru, disarankan agar Anda membuat pekerjaan pelabelan menggunakan konsol untuk memastikan atribut label Anda, bingkai titik awan, dan jika berlaku, gambar, muncul seperti yang diharapkan.

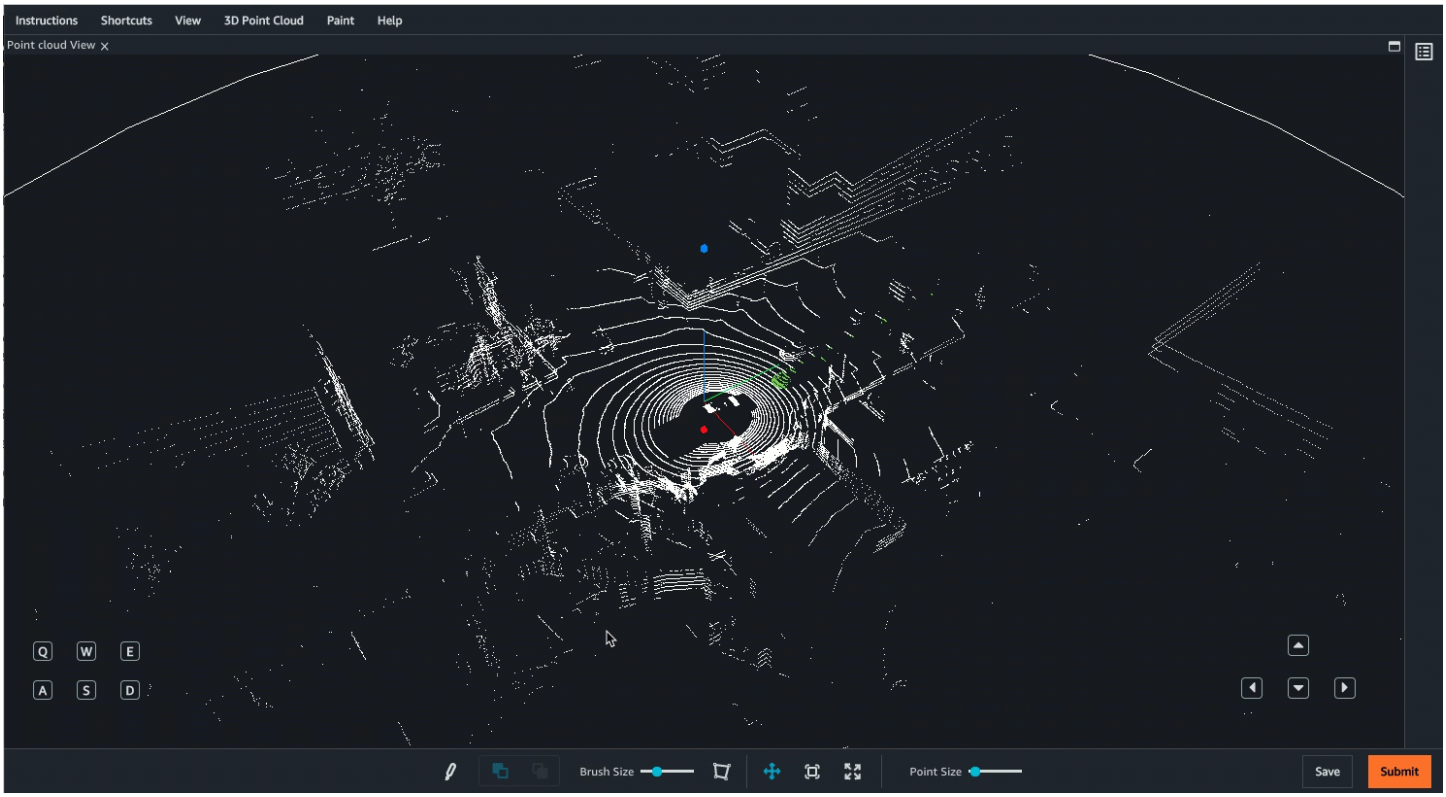
Berikut ini adalah GIF dari antarmuka tugas pekerja segmentasi semantik titik awan 3D. Jika Anda memberikan data kamera untuk fusi sensor, gambar dicocokkan dengan pemandangan dalam bingkai titik awan. Pekerja dapat melukis objek baik di awan titik 3D atau gambar 2D, dan cat muncul di lokasi yang sesuai di media lainnya. Gambar-gambar ini muncul di portal pekerja seperti yang ditunjukkan pada GIF berikut.



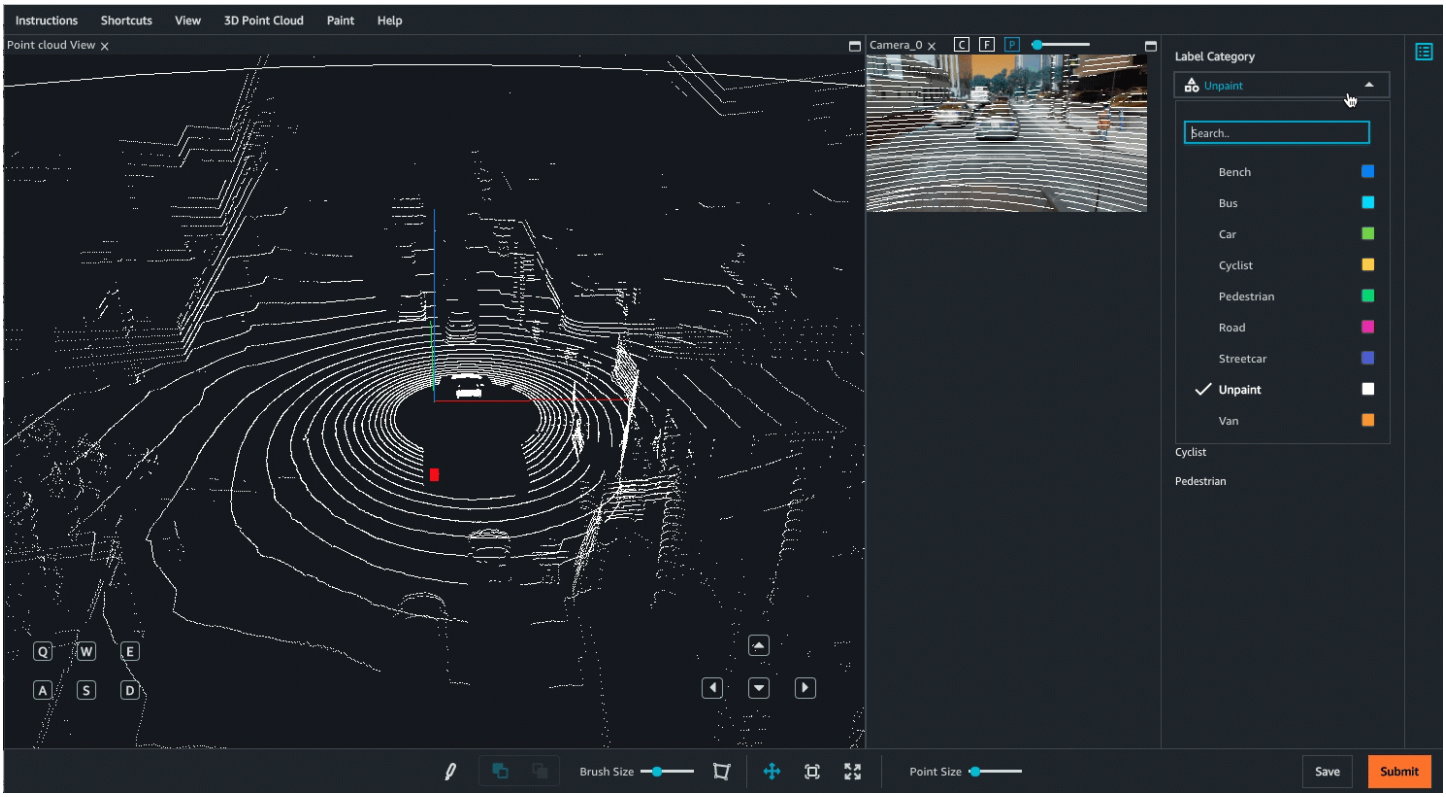
Pekerja dapat menavigasi dalam adegan 3D menggunakan keyboard dan mouse mereka. Mereka dapat:

- Klik dua kali pada objek tertentu di titik awan untuk memperbesarnya.
- Gunakan mouse-scroller atau trackpad untuk memperbesar dan memperkecil titik awan.
- Gunakan kedua tombol panah keyboard dan tombol Q, E, A, dan D untuk bergerak Naik, Bawah, Kiri, Kanan. Gunakan tombol keyboard W dan S untuk memperbesar dan memperkecil.

Video berikut menunjukkan pergerakan di sekitar titik awan 3D. Pekerja dapat menyembunyikan dan memperluas kembali semua tampilan samping dan menu. Dalam GIF ini, tampilan samping dan menu telah runtuh.



GIF berikut menunjukkan bagaimana pekerja dapat memberi label pada beberapa objek dengan cepat, menyempurnakan objek yang dicat menggunakan opsi Unpaint dan kemudian hanya melihat titik yang telah dicat.



Opsi dan fitur tampilan tambahan tersedia. Lihathalaman instruksi pekerjauntuk ikhtisar komprehensif UI Pekerja.

Alat pekerja

Pekerja dapat menavigasi melalui cloud titik 3D dengan memperbesar dan memperkecil, dan bergerak ke segala arah di sekitar awan menggunakan pintasan mouse dan keyboard. Saat Anda membuat pekerjaan segmentasi semantik, pekerja memiliki alat berikut yang tersedia untuk mereka:

- Kuas cat untuk melukis dan mengecat benda. Pekerja melukis objek dengan memilih kategori label dan kemudian melukis di awan titik 3D. Pekerja unpaint objek dengan memilih opsi Unpaint dari menu kategori label dan menggunakan kuas cat untuk menghapus cat.
- Alat poligon yang dapat digunakan pekerja untuk memilih dan melukis area di titik awan.
- Alat cat latar belakang, yang memungkinkan pekerja melukis di belakang objek yang telah mereka anotasi tanpa mengubah anotasi aslinya. Misalnya, pekerja mungkin menggunakan alat ini untuk mengecat jalan setelah mengecat semua mobil di jalan.
- Lihat opsi yang memungkinkan pekerja untuk dengan mudah menyembunyikan atau melihat teks label, mesh tanah, dan atribut titik tambahan seperti warna atau intensitas. Pekerja juga dapat memilih antara proyeksi perspektif dan ortogonal.

Buat Job Pelabelan Segmentasi Semantik Titik Cloud 3D

Anda dapat membuat pekerjaan pelabelan cloud titik 3D menggunakan SageMaker konsol atau operasi API,`CreateLabelingJob`. Untuk membuat pekerjaan pelabelan untuk jenis tugas ini, Anda memerlukan yang berikut:

- Sebuah file manifes masukan single-frame. Untuk mempelajari cara membuat file manifes ini, lihatMembuat File Manifes Input Point Cloud Frame. Jika Anda adalah pengguna baru modalitas pelabelan cloud titik Ground Truth 3D, kami sarankan Anda meninjauFormat Data 3D Mentah yang Diterima.
- Sebuah tim kerja dari tenaga kerja swasta atau vendor. Anda tidak dapat menggunakan pekerja Amazon Mechanical Turk untuk pekerjaan pelabelan cloud titik 3D. Untuk mempelajari cara membuat tim kerja dan tim kerja, lihatMembuat dan Mengelola Tenaga Kerja.
- File konfigurasi kategori label. Untuk informasi selengkapnya, lihat Buat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai.

Selain itu, pastikan bahwa Anda telah meninjau dan memuaskanTetapkan Izin IAM untuk Menggunakan Ground Truth.

Gunakan salah satu bagian berikut untuk mempelajari cara membuat pekerjaan pelabelan menggunakan konsol atau API.

Buat Job Pelabelan (Konsol)

Anda dapat mengikuti instruksiMembuat Job Pelabelan (Konsol)untuk mempelajari cara membuat pekerjaan pelabelan segmentasi semantik titik 3D di SageMaker konsol. Saat membuat tugas pelabelan Anda, perhatikan hal-hal berikut:

- File manifes masukan Anda harus berupa file manifes satu bingkai. Untuk informasi selengkapnya, lihat Membuat File Manifes Input Point Cloud Frame.
- Pelabelan data otomatis dan konsolidasi anotasi tidak didukung untuk tugas pelabelan cloud titik 3D.
- Pekerjaan pelabelan segmentasi semantik cloud titik 3D dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini ketika Anda memilih tim kerja Anda (hingga 7 hari, atau 604800 detik).

Buat Job Pelabelan (API)

Bagian ini mencakup detail yang perlu Anda ketahui saat membuat pekerjaan pelabelan menggunakan SageMaker Operasi API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWSSDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjauLihat Jugabagian`CreateLabelingJob`.

HalamanBuat Job Pelabelan (API), memberikan ikhtisar tentang`CreateLabelingJob`operasi. Ikuti petunjuk ini dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Anda harus memasukkan ARN untuk`HumanTaskUiArn`. Gunakan `arn:aws:sagemaker:<region>:394669845002:human-task-ui/PointCloudSemanticSegmentation`. Ganti*`<region>`*denganAWSWilayah tempat Anda membuat pekerjaan pelabelan.

  Seharusnya tidak ada entri untuk`UiTemplateS3Uri`parameter.
- Klaster`LabelAttributeName`harus diakhiri`-ref`. Sebagai contoh, *`ss-labels`*`-ref`.
- File manifes masukan Anda harus berupa file manifes satu bingkai. Untuk informasi selengkapnya, lihat Membuat File Manifes Input Point Cloud Frame.
- Anda menentukan label dan instruksi pekerja dalam file konfigurasi kategori label. LihatBuat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkaiuntuk mempelajari cara membuat file ini.
- Anda perlu menyediakan ARN yang telah ditentukan sebelumnya untuk fungsi Lambda pra-anotasi dan pasca-anotasi (ACS). ARN ini khusus untukAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda.
  - Untuk menemukan pra-anotasi Lambda ARN, lihat`PreHumanTaskLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar. Misalnya, jika Anda membuat pekerjaan pelabelan Anda di us-timur-1, ARN akan`arn:aws:lambda:us-east-1:432418664414:function:PRE-3DPointCloudSemanticSegmentation`.
  - Untuk menemukan pasca-anotasi Lambda ARN, lihat`AnnotationConsolidationLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar. Misalnya, jika Anda membuat pekerjaan pelabelan Anda di us-timur-1, ARN akan`arn:aws:lambda:us-east-1:432418664414:function:ACS-3DPointCloudSemanticSegmentation`.
- Jumlah pekerja yang ditentukan`NumberOfHumanWorkersPerDataObject`seharusnya`1`.

- Pelabelan data otomatis tidak didukung untuk pekerjaan pelabelan cloud titik 3D. Anda tidak boleh menentukan nilai untuk parameter di`LabelingJobAlgorithmsConfig`.
- Pekerjaan pelabelan segmentasi semantik cloud titik 3D dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini di`TaskTimeLimitInSeconds`(hingga 7 hari, atau 604800 detik).

Buat Penyesuaian Segmentasi Semantik Titik Cloud 3D atau Job Pelabelan Verifikasi

Anda dapat membuat pekerjaan pelabelan penyesuaian dan verifikasi menggunakan konsol Ground Truth atau`CreateLabelingJob`API. Untuk mempelajari lebih lanjut tentang penyesuaian dan verifikasi pekerjaan pelabelan, dan untuk mempelajari cara membuatnya, lihatVerifikasi dan Sesuaikan Label.

Format Data Output

Saat Anda membuat pekerjaan pelabelan segmentasi semantik cloud titik 3D, tugas dikirim ke pekerja. Ketika pekerja ini menyelesaikan tugas mereka, anotasi mereka ditulis ke bucket Amazon S3 yang Anda tentukan saat Anda membuat pekerjaan pelabelan. Format data keluaran menentukan apa yang Anda lihat di bucket Amazon S3 saat status pekerjaan pelabelan Anda (LabelingJobStatus) adalah`Completed`.

Jika Anda adalah pengguna baru Ground Truth, lihatData Outputuntuk mempelajari lebih lanjut tentang format data keluaran Ground Truth. Untuk mempelajari tentang format data keluaran deteksi objek awan titik 3D, lihatOutput Segmentasi Semantik Awan Titik 3D.

Pelacakan Objek Awan Titik 3D-2D

Gunakan jenis tugas ini saat Anda ingin pekerja menautkan anotasi cloud titik 3D dengan anotasi gambar 2D dan juga menautkan anotasi gambar 2D di antara berbagai kamera. Saat ini, Ground Truth mendukung kuboid untuk anotasi di cloud titik 3D dan kotak pembatas untuk anotasi dalam video 2D. Misalnya, Anda dapat menggunakan jenis tugas ini untuk meminta pekerja menautkan pergerakan kendaraan di cloud titik 3D dengan video 2D-nya. Dengan menggunakan tautan 3D-2D, Anda dapat dengan mudah mengkorelasikan data cloud titik (seperti jarak berbentuk kubus) ke data video (kotak pembatas) hingga 8 kamera.

Ground Truth memberi pekerja alat untuk membubuhi anotasi kubus di cloud titik 3D dan kotak pembatas hingga 8 kamera menggunakan UI anotasi yang sama. Pekerja juga dapat menghubungkan berbagai kotak pembatas untuk objek yang sama di kamera yang berbeda.

Misalnya, kotak pembatas di camera1 dapat ditautkan ke kotak pembatas di camera2. Ini memungkinkan Anda untuk mengkorelasikan objek di beberapa kamera menggunakan ID unik.

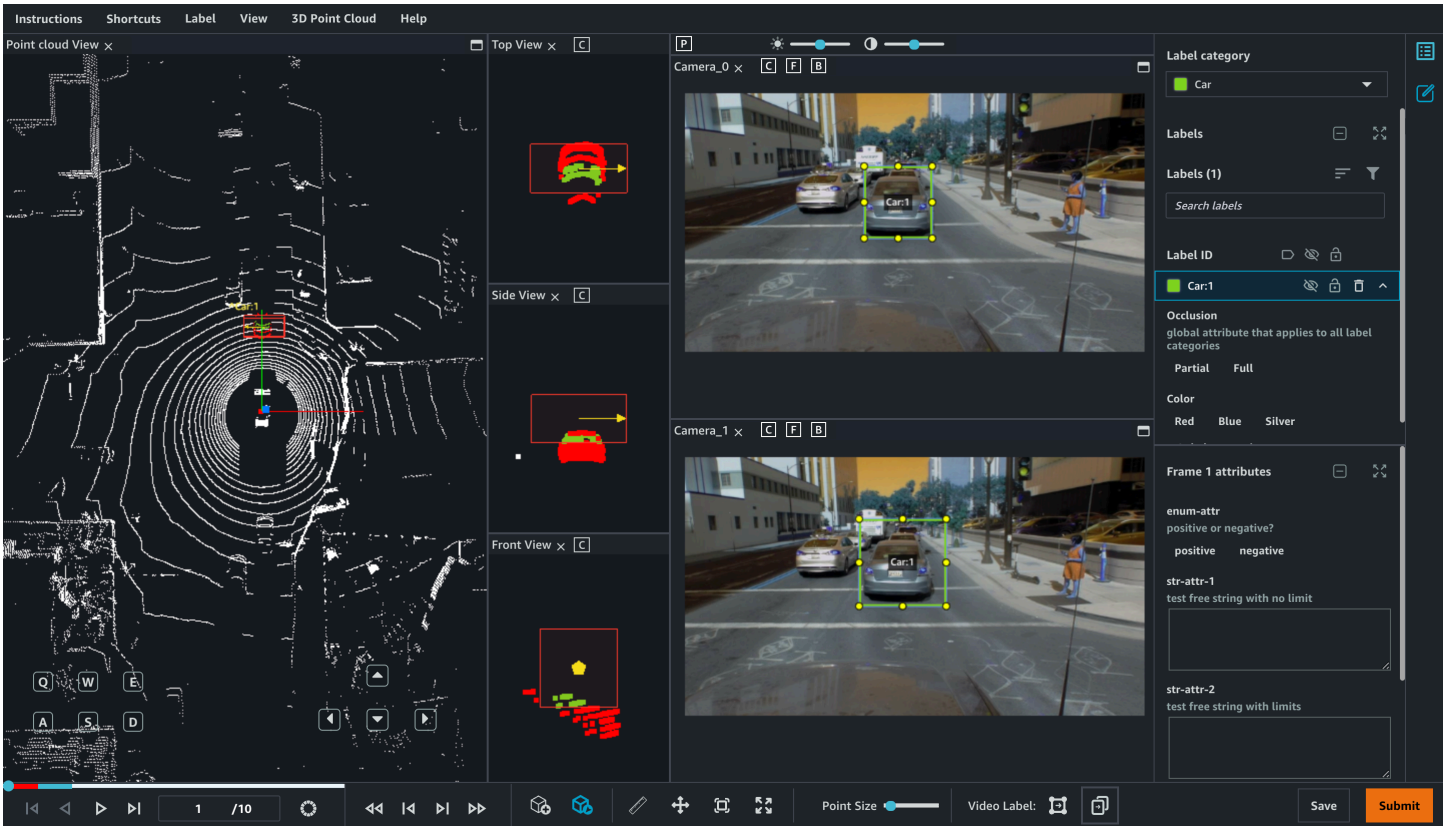
> **Note**
>
> Saat ini, SageMaker tidak mendukung pembuatan pekerjaan menghubungkan 3D-2D menggunakan konsol. Untuk membuat pekerjaan menghubungkan 3D-2D menggunakan SageMaker API, lihatBuat Job Pelabelan (API).

Topik

- Melihat Antarmuka Tugas Pekerja
- Format Data
- Buat Job Pelabelan Pelacakan Objek Titik Cloud 3D-2D
- Data

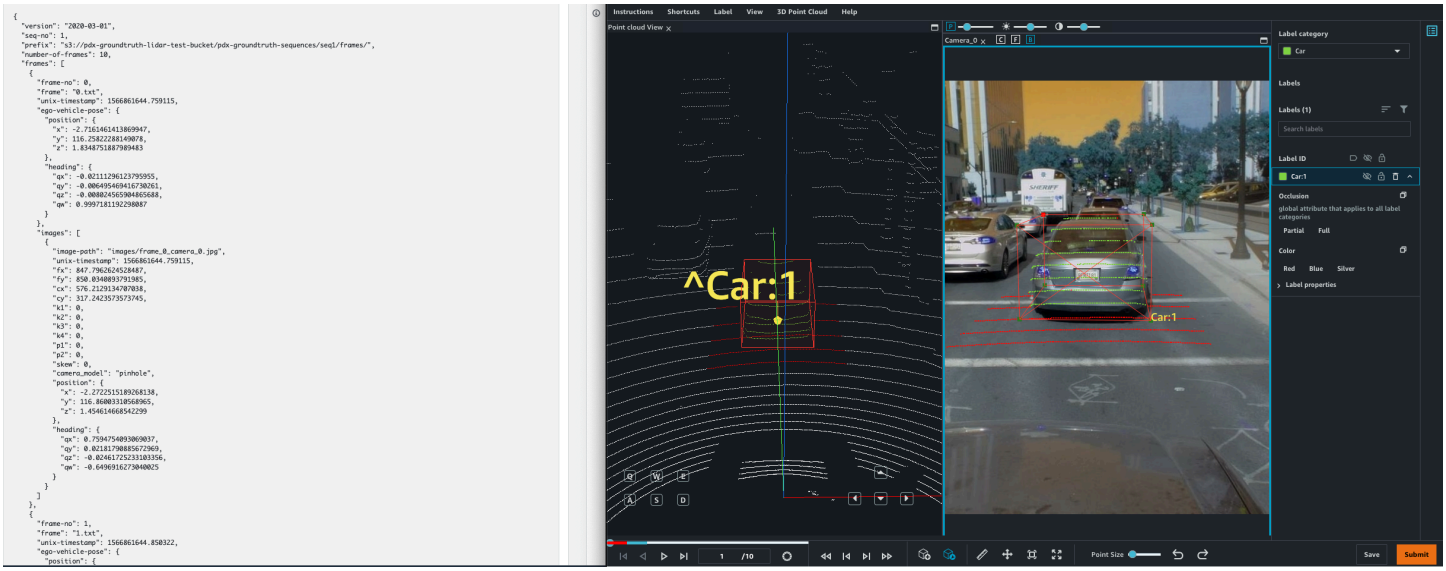
Melihat Antarmuka Tugas Pekerja

Ground Truth memberi pekerja portal web dan alat untuk menyelesaikan tugas anotasi pelacakan objek 3D-2D Anda. Ketika Anda membuat tugas pelabelan, Anda menyediakan Amazon Resource Name (ARN) untuk Amazon Resource Name (ARN) untuk UI Ground Truth yang dibuat sebelumnya dalam`HumanTaskUiArn` parameter. Untuk menggunakan UI saat Anda membuat pekerjaan pelabelan untuk jenis tugas ini menggunakan API, Anda harus menyediakan`HumanTaskUiArn`. Anda dapat melihat pratinjau dan berinteraksi dengan UI pekerja saat membuat pekerjaan pelabelan melalui API. Alat anotasi adalah bagian dari antarmuka tugas pekerja. Mereka tidak tersedia untuk antarmuka pratinjau. Gambar berikut menunjukkan antarmuka tugas pekerja yang digunakan untuk tugas anotasi pelacakan objek cloud titik 3D-2D.



Ketika interpolasi tidak tidak diaktifkan secara default. Setelah pekerja menambahkan kubus tunggal, berbentuk kubus itu direplikasi di semua frame urutan dengan ID yang sama. Jika pekerja menyesuaikan berbentuk kubus dalam bingkai lain, Ground Truth menginterpolasi pergerakan objek itu dan menyesuaikan semua kuboid di antara bingkai yang disesuaikan secara manual. Selain itu, dengan menggunakan bagian tampilan kamera, berbentuk kubus dapat ditampilkan dengan proyeksi (menggunakan tombol B untuk “label beralih” dalam tampilan kamera) yang memberi pekerja referensi dari gambar kamera. Keakuratan proyeksi cuboid terhadap gambar didasarkan pada akurasi kalibrasi yang ditangkap dalam data ekstrinsik dan intrinsink.

Jika Anda menyediakan data kamera untuk fusi sensor, gambar dicocokkan dengan pemandangan dalam bingkai cloud titik. Perhatikan bahwa data kamera harus disinkronkan waktu dengan data cloud titik untuk memastikan penggambaran akurat titik awan ke citra pada setiap frame dalam urutan seperti yang ditunjukkan pada gambar berikut.



File manifes menyimpan data ekstrinsik dan intrinsik dan pose untuk memungkinkan proyeksi berbentuk kubus pada gambar kamera ditampilkan dengan menggunakan tombol P.

Pekerja dapat menavigasi dalam adegan 3D menggunakan keyboard dan mouse mereka. Mereka dapat:

- Klik dua kali pada objek tertentu di titik awan untuk memperbesarnya.
- Gunakan mouse-scroller atau trackpad untuk memperbesar dan memperkecil titik awan.
- Gunakan kedua tombol panah keyboard dan tombol Q, E, A, dan D untuk bergerak Naik, Bawah, Kiri, Kanan. Gunakan tombol keyboard W dan S untuk memperbesar dan memperkecil.

Setelah pekerja menempatkan cuboids dalam adegan 3D, tampilan samping muncul dengan tiga tampilan samping yang diproyeksikan: atas, samping, dan depan. Pandangan samping ini menunjukkan titik-titik di dalam dan di sekitar kubus yang ditempatkan dan membantu pekerja memperbaiki batas berbentuk kubus di daerah itu. Pekerja dapat memperbesar dan memperkecil masing-masing tampilan samping menggunakan mouse mereka.

Pekerja harus terlebih dahulu memilih cuboid untuk menggambar kotak pembatas yang sesuai pada salah satu tampilan kamera. Ini menghubungkan cuboid dan kotak pembatas dengan nama umum dan ID unik.

Pekerja juga dapat menggambar kotak pembatas terlebih dahulu, pilih dan gambar berbentuk kubus yang sesuai untuk menghubungkannya.

Opsi dan fitur tampilan tambahan tersedia. Lihat halaman instruksi pekerja untuk ikhtisar komprehensif UI Pekerja.

Alat Pekerja

Pekerja dapat menavigasi melalui cloud titik 3D dengan memperbesar dan memperkecil, dan bergerak ke segala arah di sekitar awan menggunakan pintasan mouse dan keyboard. Jika pekerja mengklik titik di titik awan, UI secara otomatis memperbesar ke area tersebut. Pekerja dapat menggunakan berbagai alat untuk menggambar berbentuk kubus 3D di sekitar objek. Untuk informasi selengkapnya, lihat Alat Pelabelan Bantu dalam diskusi berikut.

Setelah pekerja menempatkan kubus 3D di titik awan, mereka dapat menyesuaikan kubus ini agar pas di sekitar mobil menggunakan berbagai tampilan: langsung di awan titik 3D, dalam tampilan samping yang menampilkan tiga perspektif zoom in dari titik awan di sekitar kotak, dan jika Anda menyertakan gambar untuk fusi sensor, langsung dalam gambar 2D.

Opsi tampilan tambahan memungkinkan pekerja untuk dengan mudah menyembunyikan atau melihat teks label, jaring tanah, dan atribut titik tambahan. Pekerja juga dapat memilih antara proyeksi perspektif dan ortogonal.

Alat Pelabelan Bantu

Ground Truth membantu pekerja membuat anotasi awan titik 3D lebih cepat dan lebih akurat menggunakan UX, pembelajaran mesin, dan alat pelabelan bantu bertenaga penglihatan komputer untuk tugas pelacakan objek cloud titik 3D. Alat pelabelan bantu berikut tersedia untuk jenis tugas ini:

- Isi otomatis label - Saat pekerja menambahkan berbentuk kubus ke bingkai, cuboid dengan dimensi, orientasi, dan posisi xyz yang sama secara otomatis ditambahkan ke semua frame dalam urutan.
- Interpolasi label - Setelah seorang pekerja memberi label satu objek dalam dua bingkai, Ground Truth menggunakan anotasi tersebut untuk menginterpolasi pergerakan objek itu di antara semua bingkai. Interpolasi label dapat dinyalakan dan dimatikan. Ia tidak menyala secara default. Misalnya, jika seorang pekerja yang bekerja dengan 5 frame menambahkan berbentuk kubus dalam bingkai 2, maka akan disalin ke semua 5 frame. Jika pekerja kemudian membuat penyesuaian dalam bingkai 4, bingkai 2 dan 4 sekarang bertindak sebagai dua titik, di mana garis cocok. Kubus kemudian diinterpolasi dalam bingkai 1,3 dan 5.
- Pengelolaan label dan atribut massal - Pekerja dapat menambahkan, menghapus, dan mengganti nama anotasi, atribut kategori label, dan atribut bingkai secara massal.

  - Pekerja dapat secara manual menghapus anotasi untuk objek tertentu sebelum dan sesudah bingkai, atau di semua frame. Misalnya, pekerja dapat menghapus semua label untuk objek setelah frame 10 jika objek tersebut tidak lagi terletak di adegan setelah frame tersebut.
  - Jika pekerja secara tidak sengaja menghapus semua anotasi untuk objek secara massal, mereka dapat menambahkannya kembali. Misalnya, jika pekerja menghapus semua anotasi untuk objek sebelum frame 100, mereka dapat menambahkannya secara massal ke frame tersebut.
  - Pekerja dapat mengganti nama label dalam satu bingkai dan semua kuboid 3D yang ditetapkan bahwa label diperbarui dengan nama baru di semua frame.
  - Pekerja dapat menggunakan pengeditan massal untuk menambahkan atau mengedit atribut kategori label dan atribut bingkai dalam beberapa bingkai.
- Snapping - Pekerja dapat menambahkan berbentuk kubus di sekitar objek dan menggunakan pintasan keyboard atau opsi menu agar alat autofit Ground Truth menjentikkan kubus dengan erat di sekitar batas objek.
- Fit to ground - Setelah seorang pekerja menambahkan berbentuk kubus ke adegan 3D, pekerja dapat secara otomatis menjepret berbentuk kubus ke tanah. Misalnya, pekerja dapat menggunakan fitur ini untuk menjepret berbentuk kubus ke jalan atau trotoar di tempat kejadian.
- Pelabelan multi-tampilan - Setelah seorang pekerja menambahkan kubus 3D ke adegan 3D, panel samping menampilkan perspektif depan dan dua sisi untuk membantu pekerja menyesuaikan kubus dengan erat di sekitar objek. Pekerja dapat membuat anotasi cloud titik 3D, panel samping dan penyesuaian muncul di tampilan lain secara real time.
- Sensor fusi — Jika Anda menyediakan data untuk fusi sensor, pekerja dapat menyesuaikan anotasi dalam adegan 3D dan dalam gambar 2D, dan anotasi diproyeksikan ke tampilan lain secara real time. Untuk mempelajari lebih lanjut tentang data untuk fusi sensor, lihat Memahami Sistem Koordinat dan Fusion Sensor.
- Gabungan otomatis kuboid - Pekerja dapat secara otomatis menggabungkan dua kuboid di semua frame jika mereka menentukan bahwa kuboid dengan label berbeda sebenarnya mewakili satu objek.
- Opsi tampilan - Memungkinkan pekerja untuk dengan mudah menyembunyikan atau melihat teks label, jaring tanah, dan atribut titik tambahan seperti warna atau intensitas. Pekerja juga dapat memilih antara proyeksi perspektif dan ortogonal.

Format Data

Anda dapat membuat pekerjaan pelacakan objek 3D-2D menggunakan operasi SageMaker API, `CreateLabelingJob`. Untuk membuat pekerjaan pelabelan untuk jenis tugas ini, Anda memerlukan yang berikut ini:

- Sebuah file manifes masukan urutan. Untuk mempelajari cara membuat file manifes jenis ini, lihatMembuat Manifes Input Urutan Point Cloud. Jika Anda adalah pengguna baru modalitas pelabelan cloud titik Ground Truth 3D, kami sarankan Anda meninjauFormat Data 3D Mentah yang Diterima.
- Anda menentukan label, kategori label, dan atribut bingkai, serta instruksi pekerja dalam file konfigurasi kategori label. Untuk informasi selengkapnya, lihat Membuat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai untuk mempelajari cara membuat file ini. Berikut ini adalah contoh yang menunjukkan file konfigurasi kategori label untuk membuat pekerjaan pelacakan objek 3D-2D.

```
{
    "document-version": "2020-03-01",
    "categoryGlobalAttributes": [
        {
            "name": "Occlusion",
            "description": "global attribute that applies to all label categories",
            "type": "string",
            "enum":[
                "Partial",
                "Full"
            ]
        }
    ],
    "labels":[
        {
            "label": "Car",
            "attributes": [
                {
                    "name": "Type",
                    "type": "string",
                    "enum": [
                        "SUV",
                        "Sedan"
                    ]
                }
```

```
            ]
        },
        {
            "label": "Bus",
            "attributes": [
                {
                    "name": "Size",
                    "type": "string",
                    "enum": [
                        "Large",
                        "Medium",
                        "Small"
                    ]
                }
            ]
        }
    ],
    "instructions": {
        "shortIntroduction": "Draw a tight cuboid around objects after you select a
 category.",
        "fullIntroduction": "<p>Use this area to add more detailed worker
 instructions.</p>"
    },
    "annotationType": [
        {
            "type": "BoundingBox"
        },
        {
            "type": "Cuboid"
        }
    ]
}
```

> **Note**
>
> Anda perlu menyediakan`BoundingBox` dan`Cuboid` sebagai annotationType dalam file konfigurasi kategori label untuk membuat pekerjaan pelacakan objek 3D-2D.

Buat Job Pelabelan Pelacakan Objek Titik Cloud 3D-2D

Anda dapat membuat pekerjaan pelabelan cloud titik 3D-2D menggunakan operasi SageMaker API, `CreateLabelingJob`. Untuk membuat pekerjaan pelabelan untuk jenis tugas ini, Anda memerlukan yang berikut ini:

- Sebuah tim kerja dari tenaga kerja swasta atau vendor. Anda tidak dapat menggunakan Amazon Mechanical Turk untuk pekerjaan pelabelan cloud titik 3D. Untuk mempelajari cara membuat tim kerja dan tim kerja, lihatMembuat dan Mengelola Tenaga Kerja.
- Tambahkan kebijakan CORS ke bucket S3 yang berisi data input di konsol Amazon S3. Untuk mengatur header CORS yang diperlukan pada bucket S3 yang berisi gambar masukan Anda di konsol S3, ikuti petunjuk yang terperinci dalam Persyaratan Izin CORS.
- Selain itu, pastikan bahwa Anda telah meninjau dan puas denganTetapkan Izin IAM untuk Menggunakan Ground Truth.

Untuk mempelajari cara membuat tugas pelabelan menggunakan API, lihat bagian berikut.

Buat Job Pelabelan (API)

Bagian ini mencakup detail yang perlu Anda ketahui saat membuat pekerjaan pelabelan pelacakan objek 3D-2D menggunakan operasi SageMaker API`CreateLabelingJob`. API ini mendefinisikan operasi ini untuk semuaAWS SDK. Untuk melihat daftar SDK khusus bahasa yang didukung untuk operasi ini, tinjau bagian Lihat Juga **`CreateLabelingJob`**.

Buat Job Pelabelan (API)memberikan gambaran umum tentang`CreateLabelingJob` operasi. Ikuti petunjuk ini dan lakukan hal berikut saat Anda mengonfigurasi permintaan Anda:

- Anda harus memasukkan ARN untuk`HumanTaskUiArn`. Gunakan `arn:aws:sagemaker:<region>:394669845002:human-task-ui/PointCloudObjectTracking`. Ganti*`<region>`* denganAWS Wilayah tempat Anda membuat pekerjaan pelabelan.

  Seharusnya tidak ada entri untuk`UiTemplateS3Uri` parameter.
- Anda `LabelAttributeName`harus berakhir`-ref`. Sebagai contoh, *`ot-labels`*`-ref`.
- File manifes masukan Anda harus berupa file manifes urutan bingkai titik awan. Untuk informasi selengkapnya, lihat Membuat Manifes Input Urutan Point Cloud. Anda juga perlu menyediakan file konfigurasi kategori label seperti yang disebutkan di atas.

- Anda perlu menyediakan ARN yang telah ditentukan sebelumnya untuk fungsi Lambda pra-anotasi dan pasca-anotasi (ACS). ARN ini khusus untukAWS Wilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda.
  - Untuk menemukan pra-anotasi Lambda ARN, lihat `PreHumanTaskLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri dengan`PRE-3DPointCloudObjectTracking`.
  - Untuk menemukan pasca-anotasi Lambda ARN, lihat `AnnotationConsolidationLambdaArn`. Gunakan Wilayah tempat Anda membuat pekerjaan pelabelan Anda untuk menemukan ARN yang benar yang diakhiri dengan`ACS-3DPointCloudObjectTracking`.
- Jumlah pekerja yang ditentukan`NumberOfHumanWorkersPerDataObject` harus`1`.
- Pelabelan data otomatis tidak didukung untuk pekerjaan pelabelan cloud titik 3D. Anda tidak harus menentukan nilai untuk parameter di`LabelingJobAlgorithmsConfig`.
- Pekerjaan pelabelan pelacakan objek 3D-2D dapat memakan waktu beberapa jam untuk menyelesaikannya. Anda dapat menentukan batas waktu yang lebih lama untuk pekerjaan pelabelan ini`TaskTimeLimitInSeconds` (hingga 7 hari, atau 604.800 detik).

> **Note**
>
> Setelah Anda berhasil membuat pekerjaan pelacakan objek 3D-2D, itu muncul di konsol di bawah pekerjaan pelabelan. Jenis tugas untuk pekerjaan ditampilkan sebagai Point Cloud Object Tracking.

Data

Saat Anda membuat tugas pelabelan pelacakan objek 3D-2D, tugas dikirim ke pekerja. Ketika pekerja ini menyelesaikan tugas mereka, anotasi mereka ditulis ke bucket Amazon S3 yang Anda tentukan saat Anda membuat pekerjaan pelabelan. Format data keluaran menentukan apa yang Anda lihat di bucket Amazon S3 saat status pekerjaan pelabelan (LabelingJobStatus) Anda berada`Completed`.

Jika Anda adalah pengguna baru Ground Truth, lihatData Output untuk mempelajari lebih lanjut tentang format data keluaran Ground Truth. Untuk mempelajari tentang format data keluaran pelacakan objek awan titik 3D-2D, lihatTitik Pelacakan Objek 3D-2D Output Pelacakan Objek Cloud.

Ikhtisar Pekerjaan Pelabelan Titik Cloud 3D

Topik ini memberikan gambaran umum dari tugas pelabelan awan titik 3D Ground Truth 3D. Anda dapat menggunakan pekerjaan pelabelan awan titik 3D agar pekerja memberi label objek di awan titik 3D yang dihasilkan dari sensor 3D seperti LiDAR dan kamera kedalaman atau dihasilkan dari rekonstruksi 3D dengan menjahit gambar yang diambil oleh agen seperti drone.

Waktu Pra-pemrosesan Job

Saat Anda membuat tugas pelabelan titik 3D, Anda perlu memberikanfile manifes masukan. File manifes masukan dapat berupa:

- SEBUAHfile manifes masukan bingkaiyang memiliki bingkai cloud titik tunggal pada setiap baris.
- SEBUAHfile manifes masukan urutanyang memiliki urutan tunggal pada setiap baris. Urutan didefinisikan sebagai serangkaian temporal frame titik awan.

Untuk kedua jenis file manifes,waktu pra-pemrosesan pekerjaan(yaitu, waktu sebelum Ground Truth mulai mengirim tugas ke pekerja Anda) tergantung pada jumlah total dan ukuran frame cloud titik yang Anda berikan dalam file manifes masukan Anda. Untuk file manifes input frame, ini adalah jumlah baris dalam file manifes Anda. Untuk file manifes urutan, ini adalah jumlah frame di setiap urutan dikalikan dengan jumlah total urutan, atau baris, dalam file manifes Anda.

Selain itu, jumlah titik per titik awan dan jumlah objek data sensor gabungan (seperti gambar) menjadi faktor waktu pra-pemrosesan pekerjaan. Rata-rata, Ground Truth dapat melakukan pra-proses frame cloud 200 titik dalam waktu sekitar 5 menit. Jika Anda membuat pekerjaan pelabelan cloud titik 3D dengan sejumlah besar frame cloud titik, Anda mungkin mengalami waktu pra-pemrosesan pekerjaan yang lebih lama. Misalnya, jika Anda membuat file manifes masukan urutan dengan urutan awan 4 titik, dan setiap urutan berisi 200 titik awan, Ground Truth memproses 800 titik awan sehingga waktu pra-pemrosesan pekerjaan Anda mungkin sekitar 20 menit. Selama waktu ini, status pekerjaan pelabelan Anda adalah`InProgress`.

Sementara pekerjaan pelabelan cloud titik 3D Anda adalah pra-pemrosesan, Anda menerima CloudWatch pesan yang memberi tahu Anda tentang status tugas Anda. Untuk mengidentifikasi pesan-pesan ini, cari`3D_POINT_CLOUD_PROCESSING_STATUS`di log Tugas pelabelan Anda.

Untukfile manifes masukan bingkai, CloudWatch log akan memiliki pesan yang mirip dengan berikut ini:

```
{
```

```
    "labeling-job-name": "example-point-cloud-labeling-job",
    "event-name": "3D_POINT_CLOUD_PROCESSING_STATUS",
    "event-log-message": "datasetObjectId from: 0 to 10, status: IN_PROGRESS"
}
```

Pesan log peristiwa,`datasetObjectId from: 0 to 10, status: IN_PROGRESS`mengidentifikasi jumlah frame dari manifes masukan Anda yang telah diproses. Anda menerima pesan baru setiap kali bingkai diproses. Misalnya, setelah satu frame diproses, Anda menerima pesan lain yang mengatakan`datasetObjectId from: 1 to 10, status: IN_PROGRESS`.

Untukfile manifes masukan urutan, CloudWatch log akan memiliki pesan yang mirip dengan berikut ini:

```
{
    "labeling-job-name": "example-point-cloud-labeling-job",
    "event-name": "3D_POINT_CLOUD_PROCESSING_STATUS",
    "event-log-message": "datasetObjectId: 0, status: IN_PROGRESS"
}
```

Pesan log peristiwa,`datasetObjectId from: 0, status: IN_PROGRESS`mengidentifikasi jumlah urutan dari manifes masukan Anda yang telah diproses. Anda menerima pesan baru setiap kali urutan diproses. Misalnya, setelah satu urutan diproses, Anda menerima pesan yang bertuliskan`datasetObjectId from: 1, status: IN_PROGRESS`sebagai urutan berikutnya mulai diproses.

Waktu Penyelesaian Job

Pekerjaan pelabelan cloud titik 3D dapat membutuhkan waktu berjam-jam pekerja untuk menyelesaikannya. Anda dapat mengatur jumlah total waktu yang pekerja dapat bekerja pada setiap tugas ketika Anda membuat pekerjaan pelabelan. Waktu maksimum yang dapat Anda tetapkan bagi pekerja untuk mengerjakan tugas adalah 7 hari. Nilai default-nya adalah 3 hari.

Sangat disarankan agar Anda membuat tugas yang dapat diselesaikan pekerja dalam waktu 12 jam. Pekerja harus menjaga UI pekerja tetap terbuka saat mengerjakan tugas. Mereka dapat menghemat pekerjaan saat mereka pergi dan Ground Truth akan menyelamatkan pekerjaan mereka setiap 15 menit.

Saat menggunakan SageMaker `CreateLabelingJob`Operasi API, mengatur total waktu tugas tersedia untuk pekerja di`TaskTimeLimitInSeconds`parameter dari`HumanTaskConfig`.

Saat Anda membuat pekerjaan pelabelan di konsol, Anda dapat menentukan batas waktu ini saat memilih jenis tenaga kerja dan tim kerja Anda.

Tenaga kerja

Saat Anda membuat pekerjaan pelabelan cloud titik 3D, Anda perlu menentukan tim kerja yang akan menyelesaikan tugas anotasi titik awan Anda. Anda dapat memilih tim kerja dari tenaga kerja pribadi pekerja Anda sendiri, atau dari tenaga kerja vendor yang Anda pilih diAWS Marketplace. Anda tidak dapat menggunakan tenaga kerja Amazon Mechanical Turk untuk pekerjaan pelabelan cloud titik 3D.

Untuk mempelajari tentang tenaga kerja vendor, lihatMengelola Tenaga Kerja Vendor.

Untuk mempelajari cara membuat dan mengelola tenaga kerja pribadi, lihatMenggunakan Tenaga Kerja Pribadi.

Antarmuka Pengguna Pekerja (UI)

Ground Truth menyediakan antarmuka pengguna pekerja (UI), alat, dan fitur pelabelan bantu untuk membantu pekerja menyelesaikan tugas pelabelan cloud titik 3D Anda.

Anda dapat melihat pratinjau UI pekerja saat membuat pekerjaan pelabelan di konsol.

Saat Anda membuat pekerjaan pelabelan menggunakan operasi API`CreateLabelingJob`, Anda harus memberikan ARN yang disediakan oleh Ground Truth dalam parameter`HumanTaskUiArn`untuk menentukan UI pekerja untuk jenis tugas Anda. Anda dapat menggunakan`HumanTaskUiArn`dengan SageMaker `RenderUiTemplate`Operasi API untuk melihat pratinjau UI pekerja.

Anda memberikan instruksi pekerja, label, dan secara opsional, atribut kategori label yang ditampilkan di UI pekerja.

Atribut Kategori Label

Saat Anda membuat pelacakan objek awan titik 3D atau pekerjaan pelabelan deteksi objek, Anda dapat menambahkan satu atau lebihatribut kategori label. Anda dapat menambahkanatribut bingkaike semua jenis tugas cloud titik 3D:

- Atribut kategori label- Daftar opsi (string), kotak teks formulir bebas, atau bidang numerik yang terkait dengan satu atau lebih label. Hal ini digunakan oleh pekerja untuk memberikan metadata tentang label.

- Atribut bingkai- Daftar opsi (string), kotak teks formulir bebas, atau bidang numerik yang muncul pada setiap titik bingkai awan pekerja dikirim untuk membuat anotasi. Hal ini digunakan oleh pekerja untuk menyediakan metadata tentang frame.

Selain itu, Anda dapat menggunakan atribut label dan bingkai agar pekerja memverifikasi label dalam tugas verifikasi label awan titik 3D.

Gunakan bagian berikut untuk mempelajari selengkapnya tentang atribut ini. Untuk mempelajari cara menambahkan kategori label dan atribut bingkai ke pekerjaan pelabelan, gunakanBuat Job pelabelanbagian padaHalaman jenis tugaspilihan Anda.

Atribut Kategori Label

Tambahkan atribut kategori label ke label untuk memberi pekerja kemampuan untuk memberikan informasi lebih lanjut tentang anotasi yang mereka buat. Atribut kategori label ditambahkan ke label individual, atau ke semua label. Ketika atribut kategori label diterapkan ke semua label itu disebut sebagaiatribut kategori label global.

Misalnya, jika Anda menambahkan kategori labelmobil, Anda mungkin juga ingin menangkap data tambahan tentang mobil berlabel Anda, seperti jika mereka tersumbat atau ukuran mobil. Anda dapat menangkap metadata ini menggunakan atribut kategori label. Dalam contoh ini, jika Anda menambahkan atributtersumbatke kategori label mobil, Anda dapat menetapkanparsial,seluruhnya,tidakkepadatersumbatatribut dan memungkinkan pekerja untuk memilih salah satu pilihan ini.

Saat membuat tugas verifikasi label, Anda menambahkan atribut kategori label ke setiap label yang ingin diverifikasi oleh pekerja.

Atribut Bingkai

Tambahkan atribut frame untuk memberi pekerja kemampuan untuk memberikan informasi lebih lanjut tentang frame cloud titik individual. Anda dapat menentukan hingga 10 atribut frame, dan atribut ini akan muncul di semua frame.

Misalnya, Anda dapat menambahkan atribut bingkai yang memungkinkan pekerja memasukkan angka. Anda mungkin ingin menggunakan atribut ini agar pekerja mengidentifikasi jumlah objek yang mereka lihat dalam bingkai tertentu.

Dalam contoh lain, Anda mungkin ingin memberikan kotak teks bentuk bebas untuk memberi pekerja kemampuan untuk memberikan jawaban formulir gratis untuk pertanyaan.

Saat membuat tugas verifikasi label, Anda dapat menambahkan satu atau beberapa atribut frame untuk meminta pekerja memberikan umpan balik pada semua label dalam bingkai titik awan.

Instruksi pekerja

Anda dapat memberikan instruksi pekerja untuk membantu pekerja Anda menyelesaikan tugas pelabelan titik cloud Anda. Anda mungkin ingin menggunakan petunjuk ini untuk melakukan hal berikut:

- Praktik terbaik dan hal-hal yang harus dihindari saat membuat anotasi objek.
- Penjelasan atribut kategori label yang disediakan (untuk deteksi objek dan tugas pelacakan objek), dan cara menggunakannya.
- Saran tentang cara menghemat waktu saat memberi label dengan menggunakan pintasan keyboard.

Anda dapat menambahkan instruksi pekerja Anda menggunakan SageMaker konsol sambil membuat pekerjaan pelabelan. Jika Anda membuat pekerjaan pelabelan menggunakan operasi API`CreateLabelingJob`, Anda menentukan instruksi pekerja dalam file konfigurasi kategori label Anda.

Selain instruksi Anda, Ground Truth menyediakan tautan untuk membantu pekerja menavigasi dan menggunakan portal pekerja. Lihat petunjuk ini dengan memilih jenis tugasInstruksi Pekerja.

Tugas Menolak

Pekerja dapat menolak tugas.

Pekerja menolak tugas jika instruksi tidak jelas, data input tidak ditampilkan dengan benar, atau jika mereka mengalami masalah lain dengan tugas tersebut. Jika jumlah pekerja per objek dataset (`NumberOfHumanWorkersPerDataObject`) Menolak tugas, objek data ditandai sebagai kadaluarsa dan tidak akan dikirim ke pekerja tambahan.

Persyaratan Izin Job Pelabelan 3D Point Cloud

Saat Anda membuat pekerjaan pelabelan cloud titik 3D, selain persyaratan izin yang ditemukan diTetapkan Izin IAM untuk Menggunakan Ground Truth, Anda harus menambahkan kebijakan CORS ke bucket S3 Anda yang berisi file manifes masukan Anda.

Menambahkan Kebijakan Izin CORS ke S3 Bucket

Saat Anda membuat tugas pelabelan cloud titik 3D, Anda menentukan bucket di S3 tempat data input dan file manifes Anda berada dan di mana data keluaran Anda akan disimpan. Ember ini mungkin sama. Anda harus melampirkan kebijakan Cross-origin resource sharing (CORS) kebijakan ke bucket input dan output Anda. Jika Anda menggunakan konsol Amazon S3 untuk menambahkan kebijakan ke bucket Anda, Anda harus menggunakan format JSON.

JSON

```
[
        {
            "AllowedHeaders": [
                "*"
            ],
            "AllowedMethods": [
                "GET",
                "HEAD",
                "PUT"
            ],
            "AllowedOrigins": [
                "*"
            ],
            "ExposeHeaders": [
                "Access-Control-Allow-Origin"
            ],
            "MaxAgeSeconds": 3000
        }
    ]
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
    <CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <CORSRule>
        <AllowedOrigin>*</AllowedOrigin>
        <AllowedMethod>GET</AllowedMethod>
        <AllowedMethod>HEAD</AllowedMethod>
        <AllowedMethod>PUT</AllowedMethod>
        <MaxAgeSeconds>3000</MaxAgeSeconds>
        <ExposeHeader>Access-Control-Allow-Origin</ExposeHeader>
        <AllowedHeader>*</AllowedHeader>
```

```
    </CORSRule>
    </CORSConfiguration>
```

Untuk mempelajari cara menambahkan kebijakan CORS ke bucket S3, lihatBagaimana cara menambahkan pembagian sumber daya lintas domain dengan CORS?dalam Panduan Pengguna Amazon Simple Storage Service.

Instruksi Pekerja

Topik ini memberikan gambaran umum tentang portal pekerja Ground Truth dan alat yang tersedia untuk menyelesaikan tugas pelabelan 3D Point Cloud Anda. Pertama, pilih jenis tugas yang sedang Anda kerjakanTopik.

Untuk pekerjaan penyesuaian, pilih jenis tugas tugas pelabelan asli yang menghasilkan label yang Anda sesuaikan. Tinjau dan sesuaikan label dalam tugas Anda sesuai kebutuhan.

> **Important**
>
> Dianjurkan agar Anda menyelesaikan tugas Anda menggunakan browser web Google Chrome atau Firefox.

Topik

- Segmentasi Semantik Awan Titik 3D
- Deteksi Objek Awan Titik 3D
- Pelacakan Objek Awan Titik 3D

Segmentasi Semantik Awan Titik 3D

Gunakan halaman ini untuk membiasakan diri dengan antarmuka pengguna dan alat yang tersedia untuk menyelesaikan tugas segmentasi semantik cloud titik 3D Anda.

Topik

- Tugas Anda
- Menavigasi UI
- Panduan ikon
- Pintasan jalan si

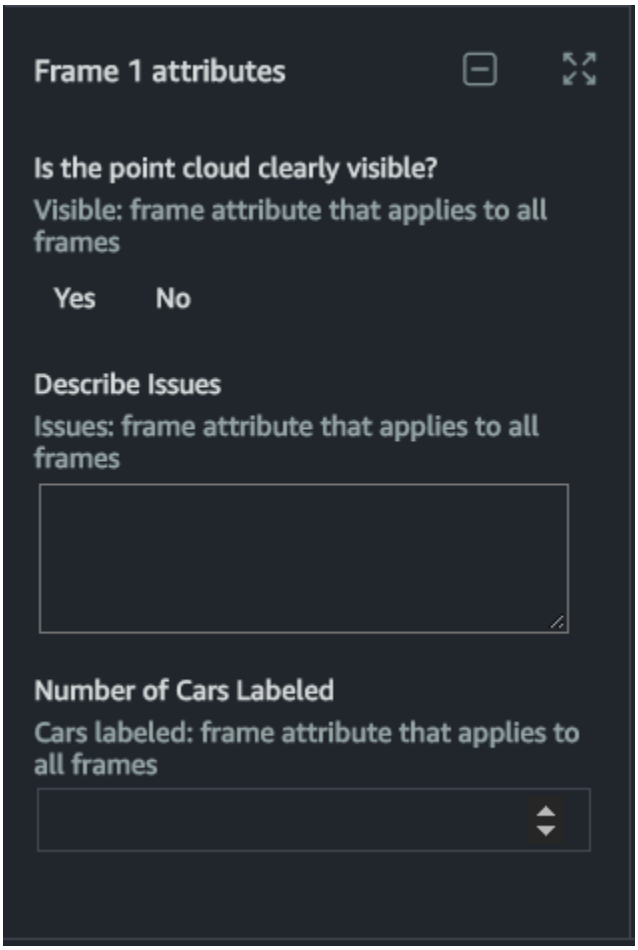
- Rilis, Hentikan, dan Lanjutkan, dan Tolak Tugas
- Menyimpan Pekerjaan Anda dan Mengirimkan

Tugas Anda

Saat Anda mengerjakan tugas segmentasi semantik cloud titik 3D, Anda harus memilih kategori dariAnotasimenu di sisi kanan portal pekerja Anda menggunakan menu drop-downKategori Label. Setelah Anda memilih kategori, gunakan kuas cat dan alat poligon untuk melukis setiap objek di awan titik 3D yang berlaku untuk kategori ini. Misalnya, jika Anda memilih kategoriCar, Anda akan menggunakan alat ini untuk melukis semua mobil di titik awan. Video berikut menunjukkan bagaimana menggunakan alat kuas cat untuk melukis objek.

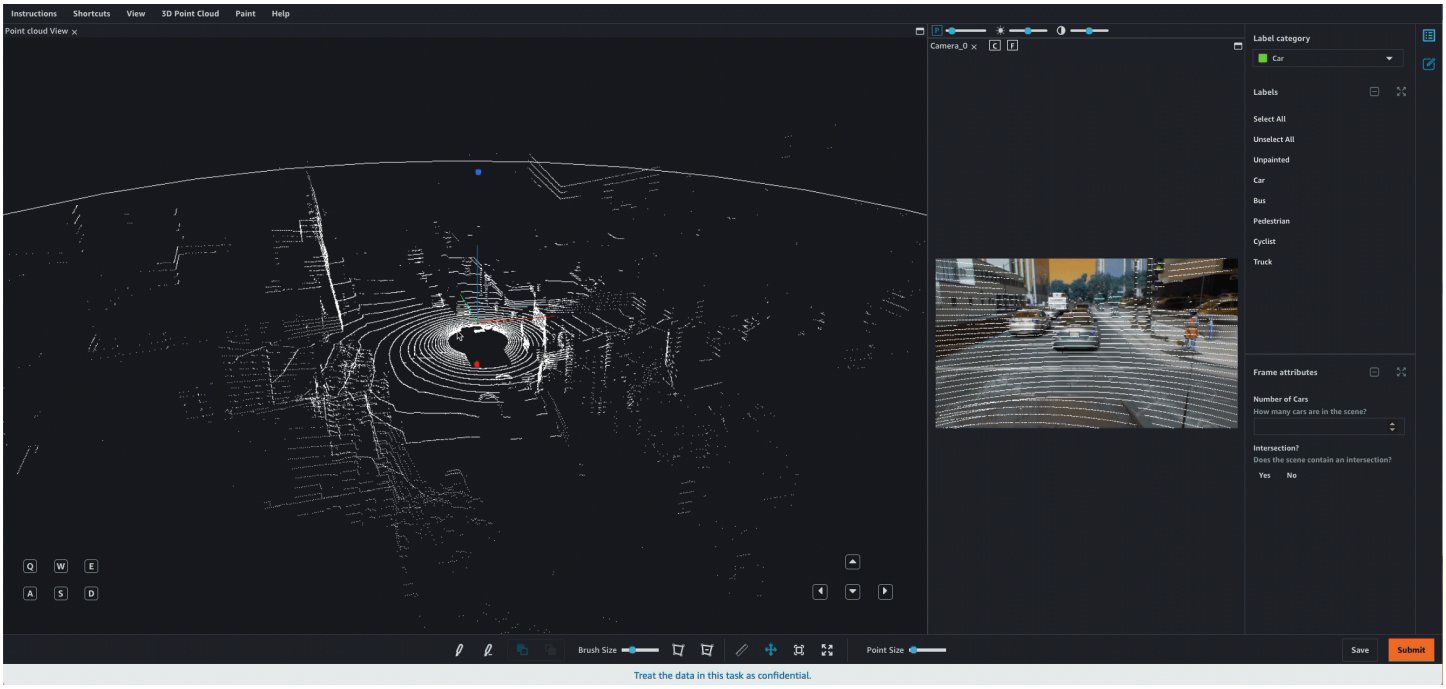
Jika Anda melihat satu atau lebih gambar di portal pekerja Anda, Anda dapat melukis di gambar atau melukis di awan titik 3D dan cat akan muncul di media lain.

Anda mungkin melihat atribut frame di bawahLabelMenu. Gunakan petunjuk atribut ini untuk memasukkan informasi tambahan tentang titik awan.

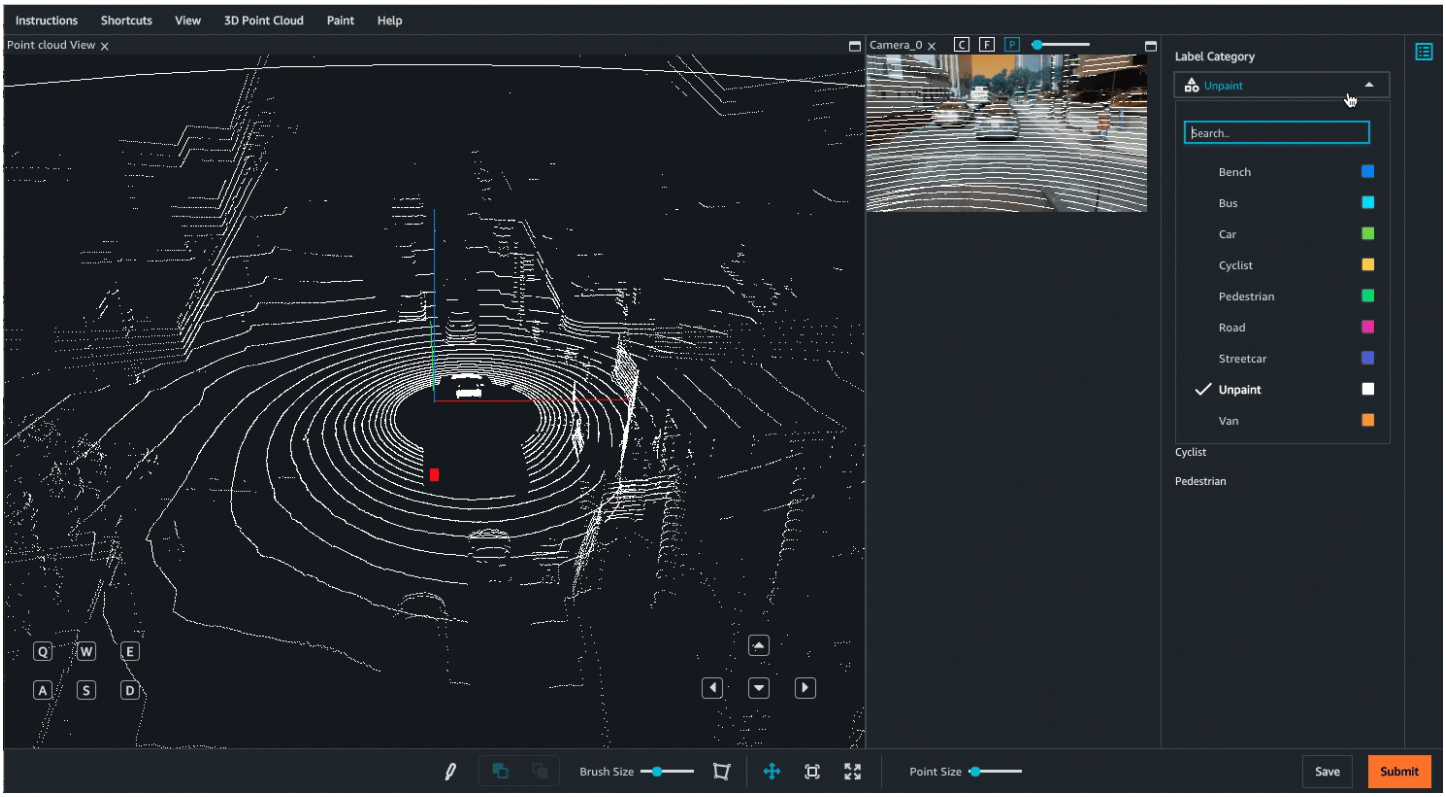


> **Important**
>
> Jika Anda melihat bahwa objek telah dicat saat Anda membuka tugas, sesuaikan anotasi tersebut.

Video berikut menyertakan gambar yang dapat dianotasi. Anda mungkin tidak melihat gambar dalam tugas Anda.



Setelah Anda melukis satu atau beberapa objek menggunakan kategori label, Anda dapat memilih kategori itu dari menu Kategori Label di sebelah kanan untuk hanya melihat titik yang dicat untuk kategori itu.

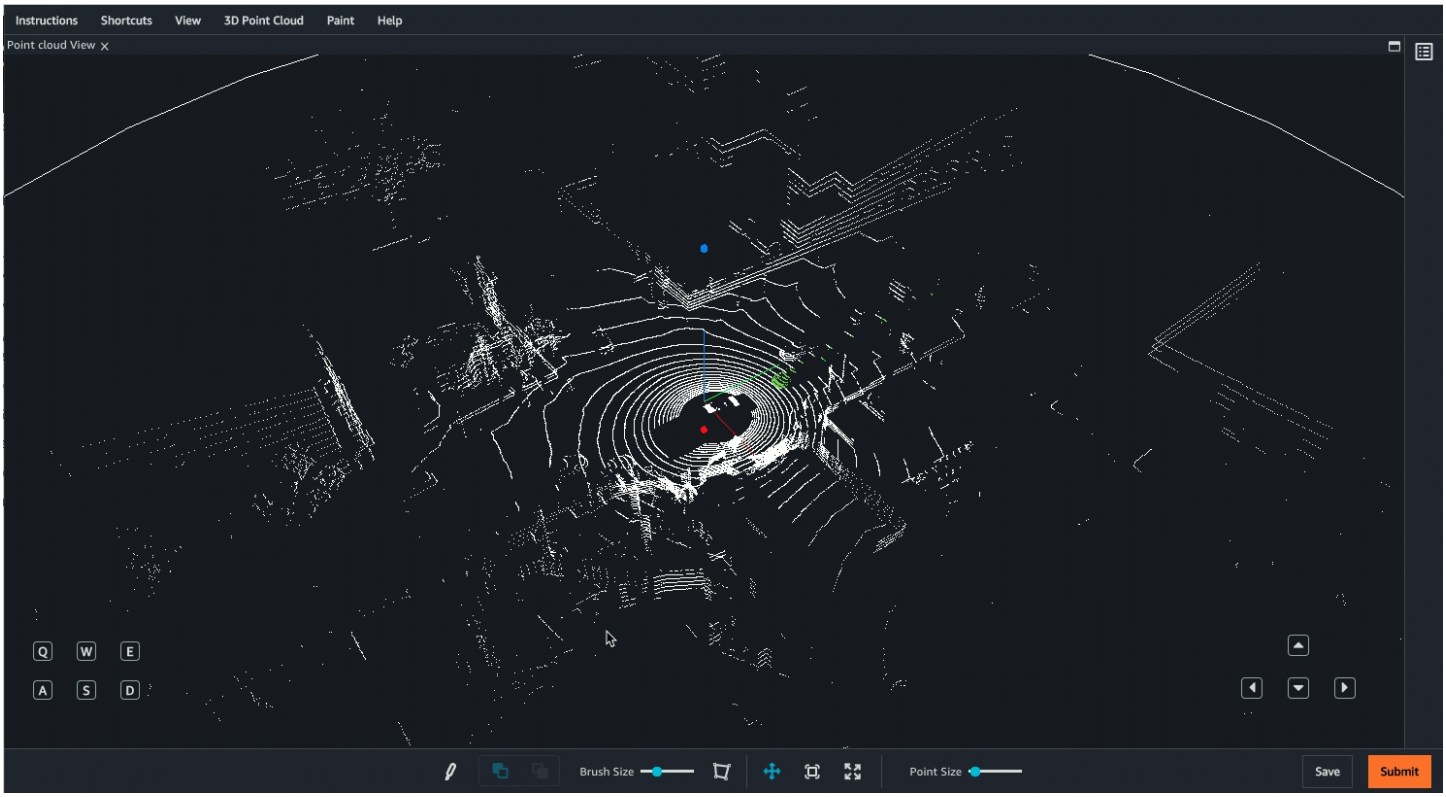


Menavigasi UI

Anda dapat menavigasi dalam adegan 3D menggunakan keyboard dan mouse mereka. Anda dapat:

- Klik dua kali pada objek tertentu di titik awan untuk memperbesarnya.
- Gunakan mouse-scroller atau trackpad untuk memperbesar dan memperkecil titik awan.
- Gunakan kedua tombol panah keyboard dan tombol Q, E, A, dan D untuk bergerak Naik, Bawah, Kiri, Kanan. Gunakan tombol keyboard W dan S untuk memperbesar dan memperkecil.

Video berikut menunjukkan pergerakan di sekitar awan titik 3D dan di tampilan samping. Anda dapat menyembunyikan dan memperluas kembali semua tampilan samping menggunakan ikon layar penuh. Dalam GIF ini, tampilan samping dan menu telah runtuh.



Ketika Anda berada di UI pekerja, Anda akan melihat menu berikut:

- Petunjuk- Tinjau instruksi ini sebelum memulai tugas Anda.
- Pintasan jalan si- Gunakan menu ini untuk melihat pintasan keyboard yang dapat Anda gunakan untuk menavigasi titik awan dan menggunakan alat anotasi yang disediakan.
- Lihat- Gunakan menu ini untuk mengaktifkan dan menonaktifkan opsi tampilan yang berbeda. Misalnya, Anda dapat menggunakan menu ini untuk menambahkan ground mesh ke titik awan, dan untuk memilih proyeksi titik awan.
- Awan Titik 3D- Gunakan menu ini untuk menambahkan atribut tambahan ke titik-titik di titik awan, seperti warna, dan intensitas piksel. Perhatikan bahwa beberapa atau semua opsi ini mungkin tidak tersedia.
- Cat- Gunakan menu ini untuk memodifikasi fungsionalitas kuas cat.

Saat Anda membuka tugas, ikon adegan bergerak aktif, dan Anda dapat bergerak di sekitar titik awan menggunakan mouse dan tombol navigasi di area titik awan layar. Untuk kembali ke tampilan asli yang Anda lihat saat pertama kali membuka tugas, pilih ikon reset scene.

Setelah Anda memilih ikon cat, Anda dapat menambahkan cat ke titik awan dan gambar (jika disertakan). Anda harus memilih ikon adegan pindah lagi untuk pindah ke area lain di awan titik 3D atau gambar.

Untuk menutup semua panel di sebelah kanan dan membuat layar penuh cloud titik 3D, pilih ikon layar penuh.

Untuk gambar kamera dan panel samping, Anda memiliki opsi tampilan berikut:

- C- Lihat sudut kamera pada tampilan titik awan.
- F- Lihat frustum, atau bidang pandang, kamera yang digunakan untuk menangkap gambar itu pada tampilan cloud titik.
- P- Lihat titik awan yang dilapis pada gambar.

Panduan ikon

Gunakan tabel ini untuk mempelajari tentang ikon yang tersedia di portal tugas pekerja Anda.

Ikon	Nama	Penjelasan
	menyikat	Pilih ikon ini untuk mengaktifkan brush tool. Untuk digunakan dengan alat ini, pilih dan pindahkan objek yang ingin Anda lukis dengan mouse Anda. Setelah Anda memilihnya, semua yang Anda cat dikaitkan dengan kategori yang Anda pilih.
	poligon	Pilih ikon ini untuk menggunakan alat cat poligon. Gunakan alat ini untuk menggambar poligon di sekitar objek yang ingin Anda cat. Setelah Anda memilihny a, semua yang Anda gambar poligon di sekitar akan dikaitkan dengan kategori yang telah Anda pilih.
	Atur ulang adegan	Pilih ikon ini untuk mengatur ulang tampilan titik awan, panel samping, dan jika berlaku, semua gambar ke posisi semula saat tugas pertama kali dibuka.

Ikon	Nama	Penjelasan
	pindahkan adegan	Pilih ikon ini untuk memindahkan adegan. Secara default, ikon ini akan dipilih saat Anda pertama kali memulai tugas.
	Layar penuh	Pilih ikon ini untuk membuat visualisasi awan titik 3D layar penuh, dan untuk menutup semua panel samping.

Ikon	Nama	Penjelasan
	penggaris	Gunakan ikon ini untuk mengukur jarak, dalam meter, di titik awan. Anda mungkin ingin menggunakan alat ini jika instruksi Anda meminta Anda untuk membuat anotasi semua objek dalam jarak tertentu dari pusat berbentuk kubus atau objek yang digunakan untuk menangkap data. Saat Anda memilih ikon ini, Anda dapat menempatk an titik awal (penanda pertama) di mana saja di titik awan dengan memilihnya dengan mouse Anda. Alat ini akan secara otomatis menggunakan interpolasi untuk menempatkan penanda pada titik terdekat dalam jarak ambang ke lokasi yang Anda pilih, jika tidak penanda akan ditempatkan di tanah. Jika Anda menempatkan titik awal karena kesalahan, Anda dapat menggunak an tombol Escape untuk mengembalikan penempatan penanda. Setelah Anda menempatkan penanda pertama, Anda melihat garis putus-putus dan label dinamis yang menunjukkan jarak yang telah Anda pindah dari penanda pertama. Klik di tempat lain di titik awan untuk menempatkan penanda kedua. Ketika Anda menempatk an penanda kedua, garis putus-putus menjadi padat, dan jarak diatur. Setelah Anda mengatur jarak, Anda dapat mengeditn ya dengan memilih salah satu penanda. Anda dapat menghapus penggaris dengan memilih di mana saja pada penggaris dan menggunakan tombol Hapus pada keyboard Anda.

Pintasan jalan si

Pintasan yang tercantum dalamPintasan jalan simenu dapat membantu Anda menavigasi awan titik 3D dan menggunakan alat cat.

Sebelum Anda memulai tugas Anda, disarankan agar Anda meninjauPintasan jalan simenu dan berkenalan dengan perintah-perintah ini.

Rilis, Hentikan, dan Lanjutkan, dan Tolak Tugas

Saat Anda membuka tugas pelabelan, tiga tombol di kanan atas memungkinkan Anda menolak tugas (tugas tolak balik), lepaskan (tugas rilis), dan berhenti dan melanjutkan di lain waktu (Berhenti dan melanjutkan nanti). Daftar berikut menjelaskan apa yang terjadi ketika Anda memilih salah satu opsi ini:

- tugas tolak balik: Anda hanya boleh menolak tugas jika ada yang salah dengan tugas, seperti masalah dengan cloud titik 3D, gambar, atau UI. Jika Anda menolak tugas, Anda tidak akan dapat kembali ke tugas.
- tugas rilis: Jika Anda melepaskan tugas, Anda kehilangan semua pekerjaan yang dilakukan pada tugas itu. Ketika tugas dilepaskan, pekerja lain di tim Anda dapat mengambilnya. Jika cukup pekerja mengambil tugas, Anda mungkin tidak akan dapat kembali ke sana. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja. Jika tugas masih tersedia, statusnya akanTersedia. Jika pekerja lain mengambilnya, itu akan hilang dari portal Anda.
- Berhenti dan melanjutkan nanti: Anda dapat menggunakanBerhenti dan melanjutkan nantitombol untuk berhenti bekerja dan kembali ke tugas di lain waktu. Anda harus menggunakanSimpantombol untuk menyimpan pekerjaan Anda sebelum Anda memilihBerhenti dan melanjutkan nanti. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja, dan status tugasnya adalahDihentikan. Anda dapat memilih tugas yang sama untuk melanjutkan pekerjaan di atasnya.

  Ketahuilah bahwa orang yang membuat tugas pelabelan Anda menentukan batas waktu di mana semua tugas banyak diselesaikan. Jika Anda tidak kembali ke dan menyelesaikan tugas ini dalam batas waktu itu, itu akan kedaluwarsa dan pekerjaan Anda tidak akan diserahkan. Hubungi administrator untuk informasi lebih lanjut.

Menyimpan Pekerjaan Anda dan Mengirimkan

Anda harus menyimpan pekerjaan Anda secara berkala. Ground Truth akan secara otomatis menyimpan pekerjaan Anda pernah 15 menit.

Ketika Anda membuka tugas, Anda harus menyelesaikan pekerjaan Anda di atasnya sebelum menekanKIRIMKAN.

Deteksi Objek Awan Titik 3D

Gunakan halaman ini untuk membiasakan diri dengan antarmuka pengguna dan alat yang tersedia untuk menyelesaikan tugas deteksi objek titik awan 3D Anda.

Topik

Tugas Anda

Saat Anda mengerjakan tugas deteksi objek awan titik 3D, Anda harus memilih kategori dariAnotasimenu di sisi kanan portal pekerja Anda menggunakanKategori LabelMenu. Setelah Anda memilih kategori, gunakan add cuboid dan fit cuboid tools agar sesuai dengan berbentuk kubus di sekitar objek di awan titik 3D yang berlaku untuk kategori ini. Setelah Anda menempatkan berbentuk kubus, Anda dapat memodifikasi dimensi, lokasi, dan orientasinya langsung di titik awan, dan tiga panel ditampilkan di sebelah kanan.

Jika Anda melihat satu atau lebih gambar di portal pekerja Anda, Anda juga dapat memodifikasi kubus dalam gambar atau di awan titik 3D dan pengeditan akan muncul di media lain.

Jika Anda melihat kubus telah ditambahkan ke awan titik 3D saat Anda membuka tugas, sesuaikan kubus tersebut dan tambahkan kubus tambahan sesuai kebutuhan.

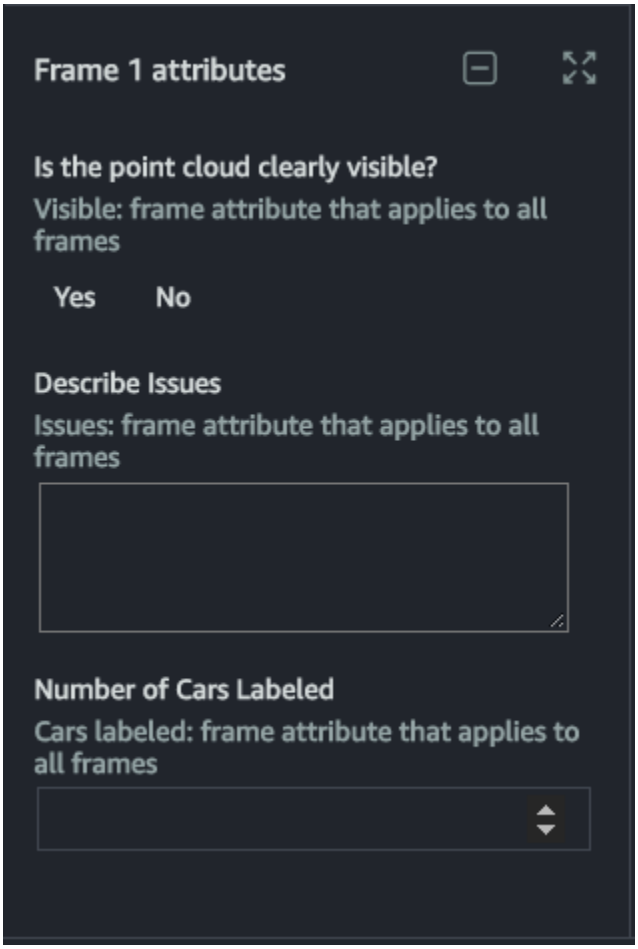
Untuk mengedit berbentuk kubus, termasuk memindahkan, mengarahkan ulang, dan mengubah dimensi berbentuk kubus, Anda harus menggunakan tombol pintas. Anda dapat melihat daftar lengkap tombol pintas diPintasan jalan simenu di UI Anda. Berikut ini adalah kombinasi kunci penting yang harus Anda ketahui sebelum memulai tugas pelabelan Anda.

Perintah Mac	Perintah Windows	Action
Cmd+Seret	Ctrl + Seret	Memodifikasi dimensi berbentuk kubus.

Perintah Mac	Perintah Windows	Action
Pilihan+Seret	Alt+Seret	Pindahkan berbentuk kubus.
Shift+Seret	Shift+Seret	Putar kubus.
Pilihan+O	Alt+O	Paskan kubus dengan erat di sekitar titik-titik yang telah digambar. Sebelum menggunakan opsi, pastikan berbentuk kubus sepenuhnya mengelilingi objek yang diminati.
Pilihan+G	Alt+G	Atur berbentuk kubus ke tanah.

Label individu mungkin memiliki satu atau lebih atribut label. Jika label memiliki atribut label yang terkait dengannya, label akan muncul saat Anda memilih panah penunjuk ke bawah di samping label dariID labelMenu. Isi nilai yang diperlukan untuk semua atribut label.

Anda mungkin melihat atribut frame di bawahLabelMenu. Gunakan petunjuk atribut ini untuk memasukkan informasi tambahan tentang setiap frame.



Menavigasi UI

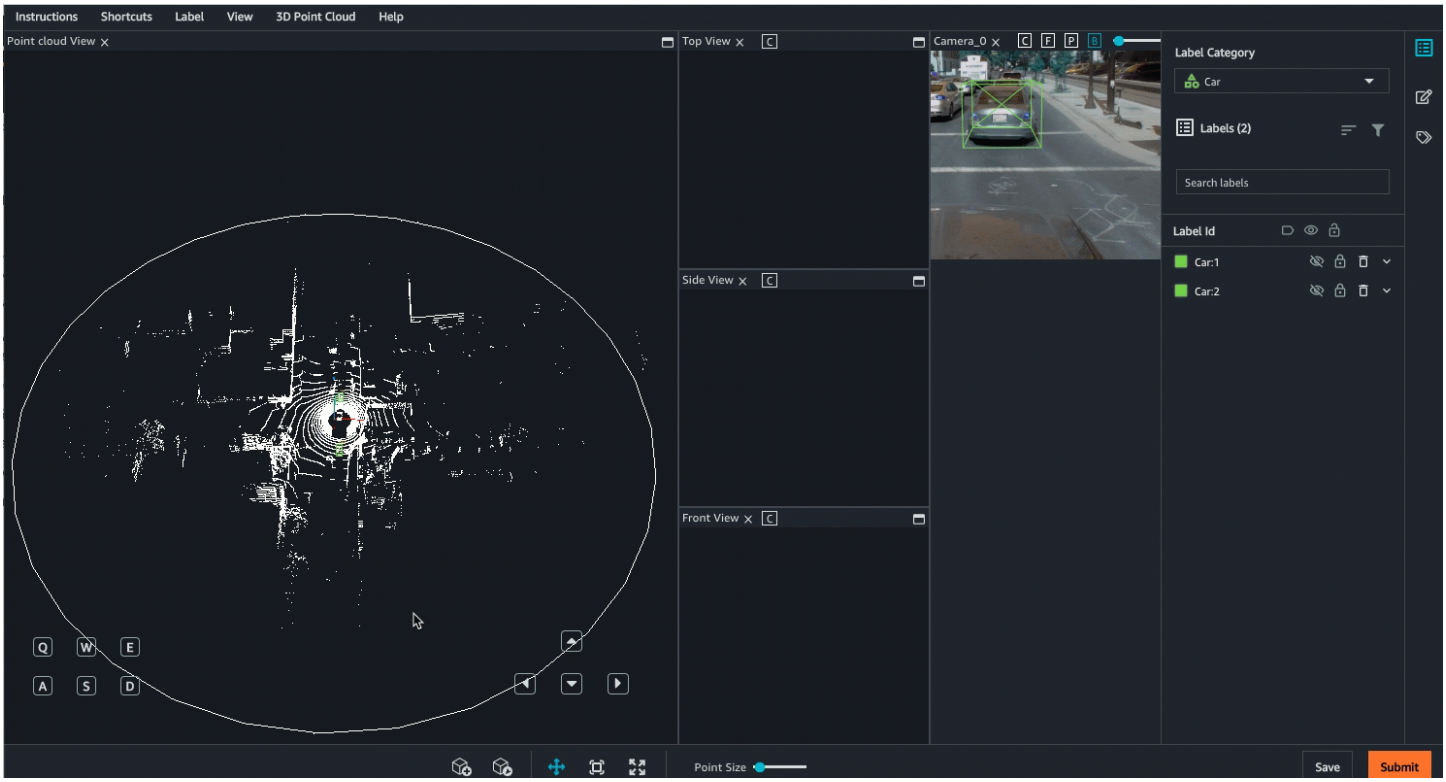
Anda dapat menavigasi dalam adegan 3D menggunakan keyboard dan mouse Anda. Anda dapat:

- Klik dua kali pada objek tertentu di titik awan untuk memperbesarnya.
- Anda dapat menggunakan tombol [dan] pada keyboard Anda untuk memperbesar dan berpindah dari satu label ke label berikutnya. Jika tidak ada label yang dipilih, saat Anda memilih [atau], UI akan memperbesar label pertama diID labelDaftar.
- Gunakan mouse-scroller atau trackpad untuk memperbesar dan memperkecil titik awan.
- Gunakan kedua tombol panah keyboard dan tombol Q, E, A, dan D untuk bergerak Naik, Bawah, Kiri, Kanan. Gunakan tombol keyboard W dan S untuk memperbesar dan memperkecil.

Setelah Anda menempatkan cuboids dalam adegan 3D, tampilan samping akan muncul dengan tiga tampilan yang diproyeksikan: atas, samping, dan belakang. Pandangan samping ini menunjukkan titik-titik di dalam dan di sekitar kubus yang ditempatkan dan membantu pekerja memperbaiki batas

berbentuk kubus di daerah itu. Pekerja dapat memperbesar dan memperkecil masing-masing tampilan samping menggunakan mouse mereka.

Video berikut menunjukkan pergerakan di sekitar awan titik 3D dan di tampilan samping.

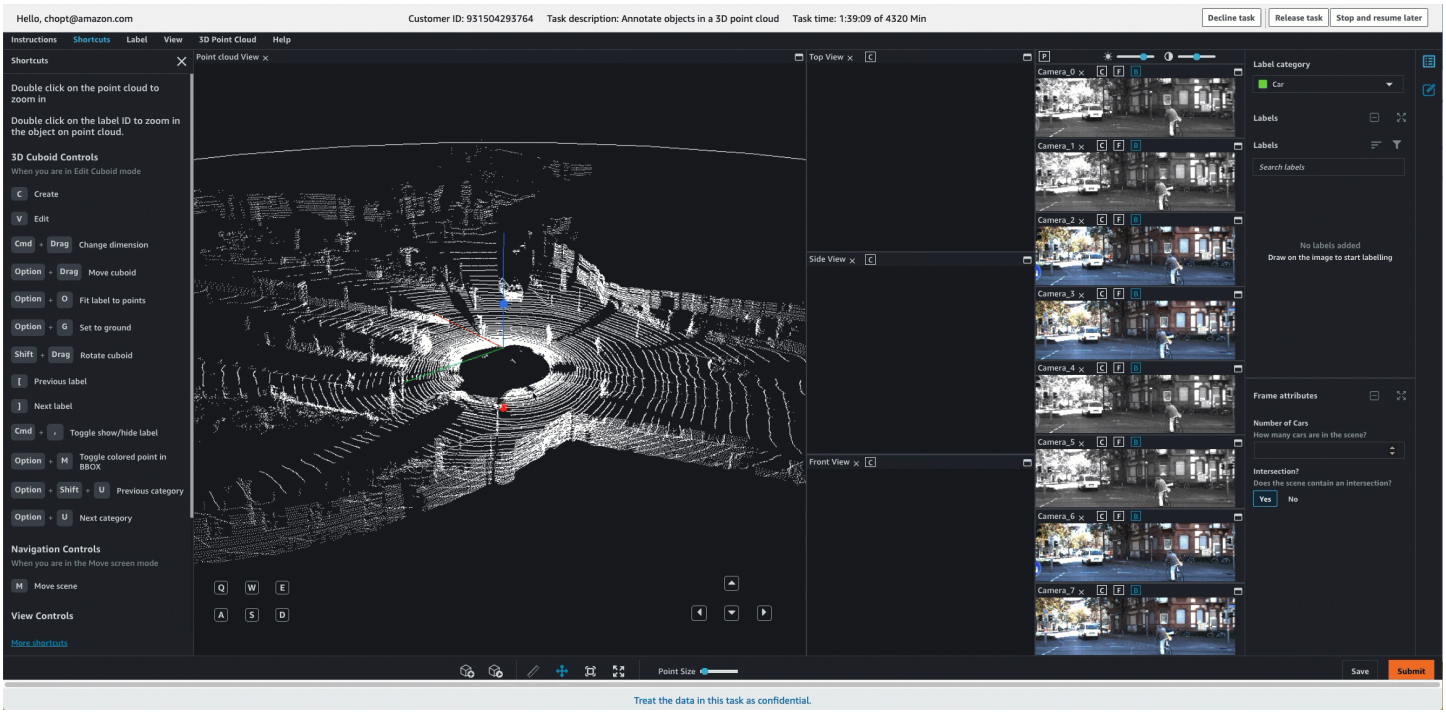


Ketika Anda berada di UI pekerja, Anda akan melihat menu berikut:

- Petunjuk- Tinjau instruksi ini sebelum memulai tugas Anda.
- Pintasan jalan si- Gunakan menu ini untuk melihat pintasan keyboard yang dapat Anda gunakan untuk menavigasi titik awan dan menggunakan alat anotasi yang disediakan.
- Label- Gunakan menu ini untuk memodifikasi berbentuk kubus. Pertama, pilih berbentuk kubus, lalu pilih opsi dari menu ini. Menu ini mencakup alat pelabelan bantu seperti mengatur berbentuk kubus ke tanah dan secara otomatis memasang kubus ke batas objek.
- Lihat- Gunakan menu ini untuk mengaktifkan dan menonaktifkan opsi tampilan yang berbeda. Misalnya, Anda dapat menggunakan menu ini untuk menambahkan ground mesh ke titik awan, dan untuk memilih proyeksi titik awan.
- Awan Titik 3D- Gunakan menu ini untuk menambahkan atribut tambahan ke titik-titik di titik awan, seperti warna, dan intensitas piksel. Perhatikan bahwa opsi ini mungkin tidak tersedia.

Saat Anda membuka tugas, ikon adegan bergerak aktif, dan Anda dapat bergerak di sekitar titik awan menggunakan mouse dan tombol navigasi di area titik awan layar. Untuk kembali ke tampilan asli yang Anda lihat saat pertama kali membuka tugas, pilih ikon reset scene. Mengatur ulang tampilan tidak akan mengubah anotasi Anda.

Setelah Anda memilih ikon add cuboid, Anda dapat menambahkan cuboids ke visualisasi awan titik 3D. Setelah Anda menambahkan berbentuk kubus, Anda dapat menyesuaikannya dalam tiga tampilan (atas, samping, dan depan) dan dalam gambar (jika disertakan).



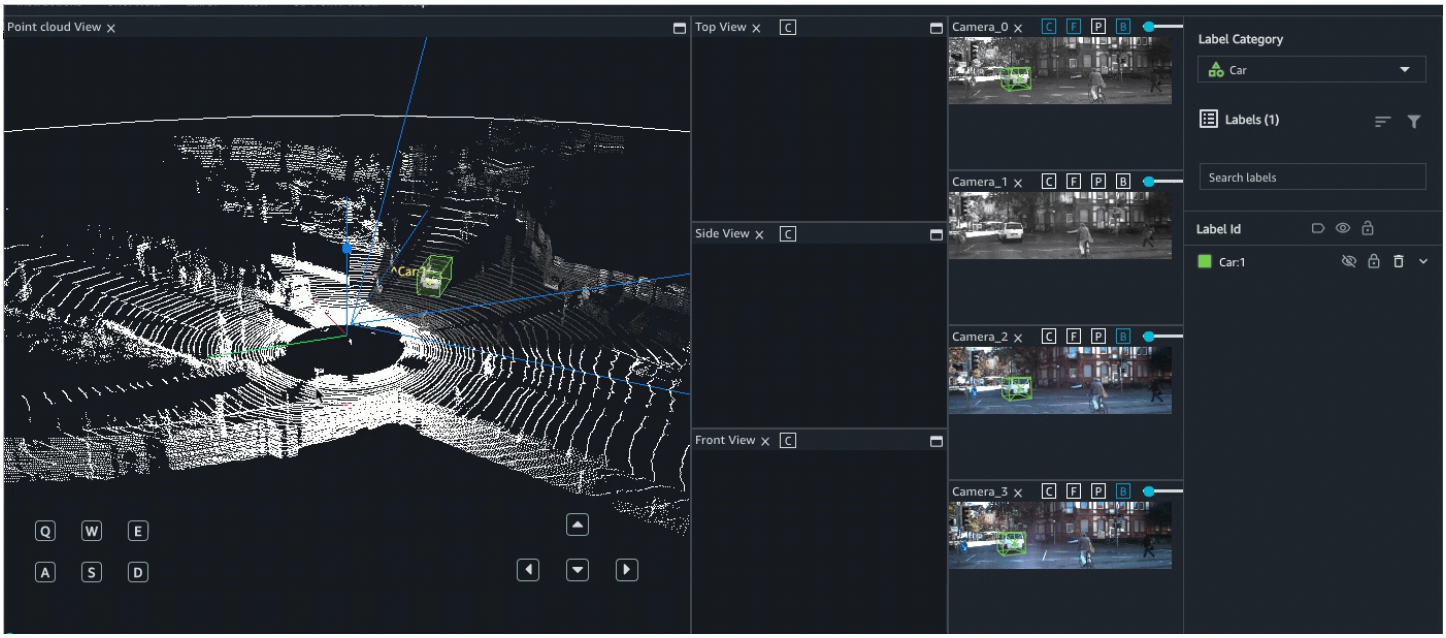
Anda harus memilih ikon adegan pindah lagi untuk pindah ke area lain di awan titik 3D atau gambar.

Untuk menutup semua panel di sebelah kanan dan membuat awan titik 3D layar penuh, pilih ikon layar penuh.

Jika gambar kamera disertakan, Anda mungkin memiliki opsi tampilan berikut:

- C- Lihat sudut kamera pada tampilan titik awan.
- F- Lihat frustum, atau bidang pandang, kamera yang digunakan untuk menangkap gambar itu pada tampilan cloud titik.
- P- Lihat titik awan yang dilapis pada gambar.
- B— Lihat kubus dalam gambar.

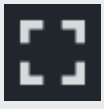
Video berikut menunjukkan cara menggunakan opsi penayangan ini. KlasterFdigunakan untuk melihat bidang pandang kamera (area abu-abu),CPilihan menunjukkan arah kamera menghadap dan sudut kamera (garis biru), danBdigunakan untuk melihat berbentuk kubus.



Panduan ikon

Gunakan tabel ini untuk mempelajari tentang ikon yang Anda lihat di portal tugas pekerja Anda.

Ikon		Deskripsi
	tambahkan kubus	Pilih ikon ini untuk menambahkan berbentuk kubus. Setiap berbentuk kubus yang Anda tambahkan dikaitkan dengan kategori yang Anda pilih.
	edit kubus	Pilih ikon ini untuk mengedit berbentuk kubus. Setelah Anda menambahkan berbentuk kubus, Anda dapat mengedit dimensi, lokasi, dan orientasinya. Setelah cuboid ditambahkan, secara otomatis beralih ke mode edit berbentuk kubus.
	penggaris	Gunakan ikon ini untuk mengukur jarak, dalam meter, di titik awan. Anda mungkin ingin menggunakan alat ini jika instruksi Anda meminta Anda untuk membuat anotasi semua objek dalam jarak tertentu dari pusat berbentuk kubus atau objek yang digunakan untuk menangkap data.

Ikon		Deskripsi
		Saat Anda memilih ikon ini, Anda dapat menempatk an titik awal (penanda pertama) di mana saja di titik awan dengan memilihnya dengan mouse Anda. Alat ini akan secara otomatis menggunakan interpolasi untuk menempatkan penanda pada titik terdekat dalam jarak ambang ke lokasi yang Anda pilih, jika tidak penanda akan ditempatkan di tanah. Jika Anda menempatkan titik awal karena kesalahan, Anda dapat menggunak an tombol Escape untuk mengembalikan penempatan penanda. Setelah Anda menempatkan penanda pertama, Anda melihat garis putus-putus dan label dinamis yang menunjukkan jarak yang telah Anda pindah dari penanda pertama. Klik di tempat lain di titik awan untuk menempatkan penanda kedua. Ketika Anda menempatk an penanda kedua, garis putus-putus menjadi padat, dan jarak diatur. Setelah Anda mengatur jarak, Anda dapat mengeditn ya dengan memilih salah satu penanda. Anda dapat menghapus penggaris dengan memilih di mana saja pada penggaris dan menggunakan tombol Hapus pada keyboard Anda.
	Atur ulang adegan	Pilih ikon ini untuk mengatur ulang tampilan titik awan, panel samping, dan jika berlaku, semua gambar ke posisi semula saat tugas pertama kali dibuka.
	pindahkan adegan	Pilih ikon ini untuk memindahkan adegan. Secara default, ikon ini dipilih saat Anda pertama kali memulai tugas.
	Layar penuh	Pilih ikon ini untuk membuat visualisasi awan titik 3D layar penuh, dan untuk menutup semua panel samping.

Ikon		Deskripsi
	Tampilkan label	Tampilkan label dalam visualisasi awan titik 3D, dan jika berlaku, dalam gambar.
	label	Sembunyikan label dalam visualisasi awan titik 3D, dan jika berlaku, dalam gambar.
	Hapus label	Menghapus label.

Pintasan jalan si

Pintasan yang tercantum dalamPintasan jalan simenu dapat membantu Anda menavigasi titik awan 3D dan menggunakan alat untuk menambah dan mengedit kubus.

Sebelum Anda memulai tugas Anda, disarankan agar Anda meninjauPintasan jalan simenu dan berkenalan dengan perintah-perintah ini. Anda perlu menggunakan beberapa kontrol berbentuk kubus 3D untuk mengedit berbentuk kubus Anda.

Rilis, Hentikan, dan Lanjutkan, dan Tolak Tugas

Saat Anda membuka tugas pelabelan, tiga tombol di kanan atas memungkinkan Anda menolak tugas (tugas tolak balik), lepaskan (tugas rilis), dan berhenti dan melanjutkan di lain waktu (Berhenti dan melanjutkan nanti). Daftar berikut menjelaskan apa yang terjadi ketika Anda memilih salah satu opsi ini:

- tugas tolak balik: Anda hanya boleh menolak tugas jika ada yang salah dengan tugas, seperti masalah dengan cloud titik 3D, gambar, atau UI. Jika Anda menolak tugas, Anda tidak akan dapat kembali ke tugas.
- tugas rilis: Jika Anda melepaskan tugas, Anda kehilangan semua pekerjaan yang dilakukan pada tugas itu. Ketika tugas dilepaskan, pekerja lain di tim Anda dapat mengambilnya. Jika cukup pekerja mengambil tugas, Anda mungkin tidak akan dapat kembali ke sana. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja. Jika tugas masih tersedia, statusnya akanTersedia. Jika pekerja lain mengambilnya, itu akan hilang dari portal Anda.
- Berhenti dan melanjutkan nanti: Anda dapat menggunakanBerhenti dan melanjutkan nantitombol untuk berhenti bekerja dan kembali ke tugas di lain waktu. Sebagai Anda harus

menggunakanSimpantombol untuk menyimpan pekerjaan Anda sebelum Anda memilihBerhenti dan melanjutkan nanti. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja, dan status tugasnya adalahDihentikan. Anda dapat memilih tugas yang sama untuk melanjutkan pekerjaan di atasnya.

Ketahuilah bahwa orang yang membuat tugas pelabelan Anda menentukan batas waktu di mana semua tugas banyak diselesaikan. Jika Anda tidak kembali ke dan menyelesaikan tugas ini dalam batas waktu itu, itu akan kedaluwarsa dan pekerjaan Anda tidak akan diserahkan. Hubungi administrator untuk informasi lebih lanjut.

Menyimpan Pekerjaan Anda dan Mengirimkan

Anda harus menyimpan pekerjaan Anda secara berkala. Ground Truth akan secara otomatis menyimpan pekerjaan Anda pernah 15 menit.

Ketika Anda membuka tugas, Anda harus menyelesaikan pekerjaan Anda di atasnya sebelum menekanKIRIMKAN.

Pelacakan Objek Awan Titik 3D

Gunakan halaman ini untuk membiasakan diri dengan antarmuka pengguna dan alat yang tersedia untuk menyelesaikan tugas deteksi objek titik awan 3D Anda.

Topik

Tugas Anda

Saat Anda mengerjakan tugas pelacakan objek awan titik 3D, Anda harus memilih kategori dariAnotasimenu di sisi kanan portal pekerja Anda menggunakanKategori LabelMenu. Setelah Anda memilih kategori, gunakan add cuboid dan fit cuboid tools agar sesuai dengan cuboid di sekitar

objek di awan titik 3D yang diterapkan kategori ini. Setelah Anda menempatkan berbentuk kubus, Anda dapat memodifikasi lokasi, dimensi, dan orientasinya langsung di titik awan, dan tiga panel ditampilkan di sebelah kanan. Jika Anda melihat satu atau lebih gambar di portal pekerja Anda, Anda juga dapat memodifikasi kubus dalam gambar atau di awan titik 3D dan pengeditan akan muncul di media lain.

> **Important**
>
> Jika Anda melihat kubus telah ditambahkan ke bingkai awan titik 3D saat Anda membuka tugas, sesuaikan kubus tersebut dan tambahkan kubus tambahan sesuai kebutuhan.

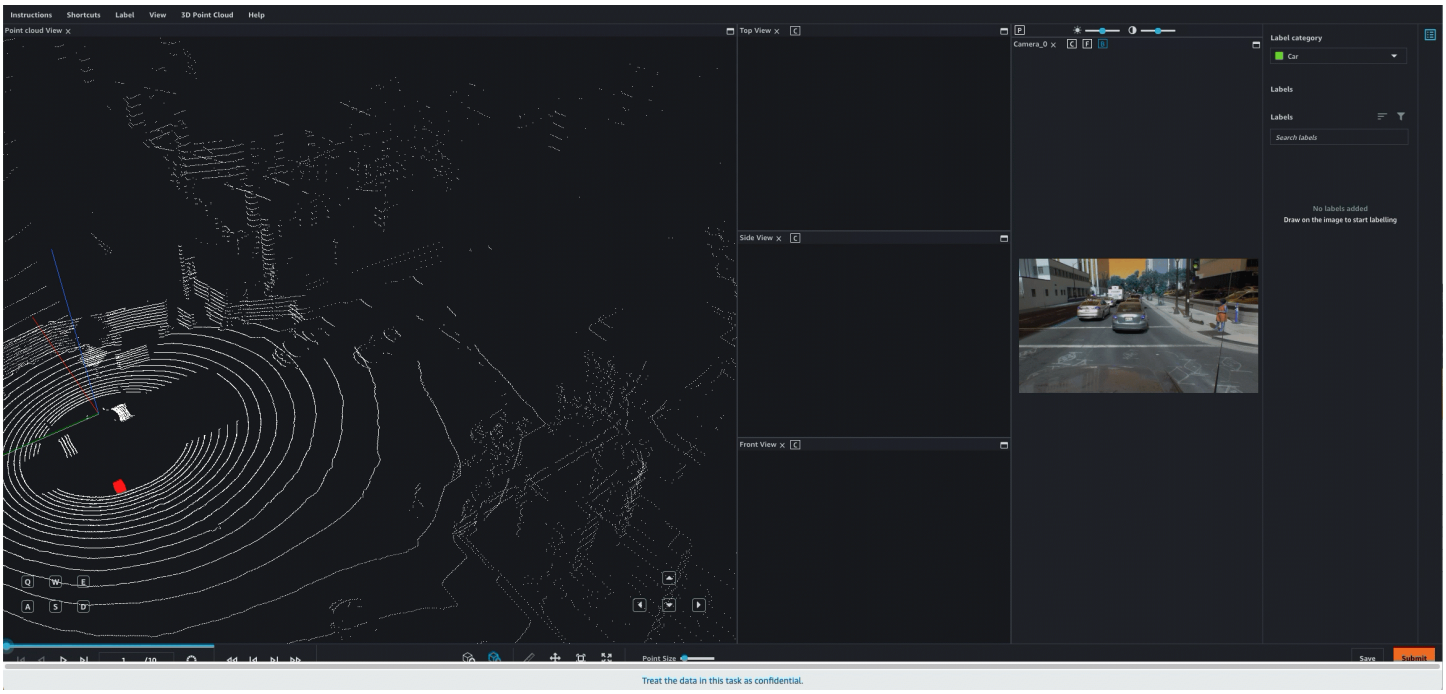
Untuk mengedit berbentuk kubus, termasuk memindahkan, mengarahkan ulang, dan mengubah dimensi berbentuk kubus, Anda harus menggunakan tombol pintas. Anda dapat melihat daftar lengkap tombol pintas diPintasan jalan simenu di UI Anda. Berikut ini adalah kombinasi kunci penting yang harus Anda ketahui sebelum memulai tugas pelabelan Anda.

Perintah Mac	Perintah Windows	Action
Cmd+Seret	Ctrl + Seret	Memodifikasi dimensi berbentuk kubus.
Pilihan+Seret	Alt+Seret	Pindahkan berbentuk kubus.
Shift+Seret	Shift+Seret	Putar kubus.
Pilihan+O	Alt+O	Paskan kubus dengan erat di sekitar titik-titik yang telah digambar. Sebelum menggunakan opsi, pastikan berbentuk kubus sepenuhny a mengelilingi objek yang diminati.
Pilihan+G	Alt+G	Atur berbentuk kubus ke tanah.

Ketika Anda membuka tugas Anda, dua frame akan dimuat. Jika tugas Anda mencakup lebih dari dua frame, Anda perlu menggunakan bilah navigasi di sudut kiri bawah, atau ikon bingkai beban untuk memuat bingkai tambahan. Anda harus membuat anotasi dan menyesuaikan label di semua frame sebelum mengirimkan.

Setelah Anda memasukkan kubus dengan erat di sekitar batas objek, navigasikan ke bingkai lain menggunakan bilah navigasi di sudut kiri bawah UI. Jika objek yang sama telah pindah ke lokasi baru, tambahkan kubus lain dan pasangkan dengan erat di sekitar batas-batas objek. Setiap kali Anda menambahkan kubus secara manual, Anda melihat bilah urutan bingkai di sudut kiri bawah layar berubah menjadi merah di mana bingkai itu terletak secara temporal dalam urutan.

UI Anda secara otomatis menyimpulkan lokasi objek itu di semua frame lain setelah Anda menempatkan berbentuk kubus. Ini disebutinterpolasi. Anda dapat melihat pergerakan objek itu, dan kubus yang disimpulkan dan dibuat secara manual menggunakan panah. Sesuaikan kubus yang disimpulkan sesuai kebutuhan. Video berikut menunjukkan cara menavigasi antar bingkai. Video berikut menunjukkan bagaimana, jika Anda menambahkan berbentuk kubus dalam satu bingkai, dan kemudian menyesuaikannya di yang lain, UI Anda akan secara otomatis menyimpulkan lokasi berbentuk kubus di semua frame di antaranya.

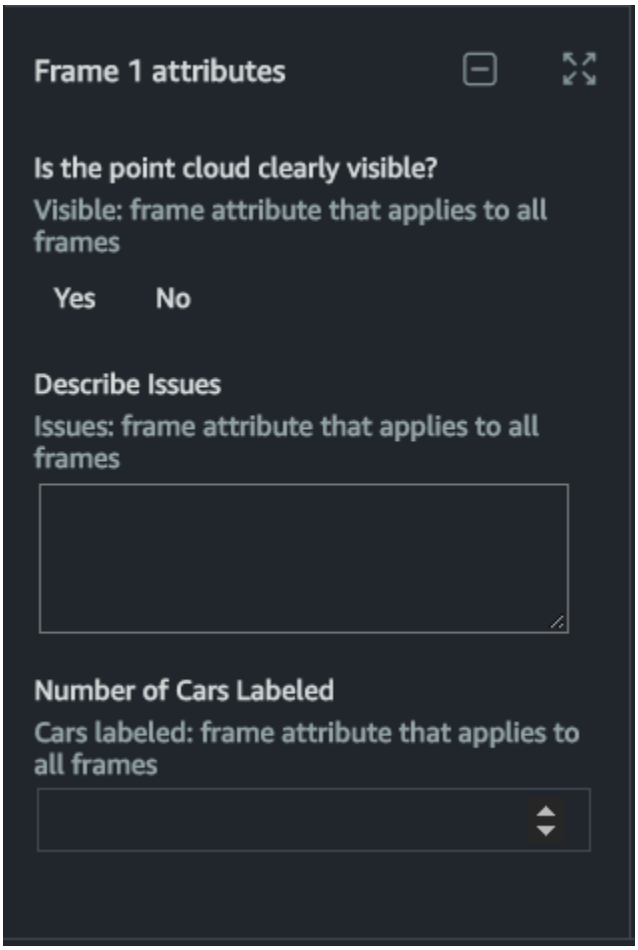


> **Tip**
>
> Anda dapat mematikan interpolasi berbentuk kubus otomatis di seluruh frame menggunakan item menu 3D Point Cloud. PilihAwan Titik 3Ddari menu atas, lalu pilihInterpolasi Kuboid Di

Frame. Ini akan menghapus centang opsi ini dan menghentikan interpolasi berbentuk kubus. Anda dapat memilih ulang item ini untuk mengaktifkan kembali interpolasi berbentuk kubus. Mematikan interpolasi berbentuk kubus tidak akan berdampak pada kubus yang telah diinterpolasi melintasi bingkai.

Label individu mungkin memiliki satu atau lebih atribut label. Jika label memiliki atribut label yang terkait dengannya, label akan muncul saat Anda memilih panah penunjuk ke bawah di samping label dariID labelMenu. Isi nilai yang diperlukan untuk semua atribut label.

Anda mungkin melihat atribut frame di bawahID labelMenu. Atribut ini akan muncul di setiap frame dalam tugas Anda. Gunakan petunjuk atribut ini untuk memasukkan informasi tambahan tentang setiap frame.



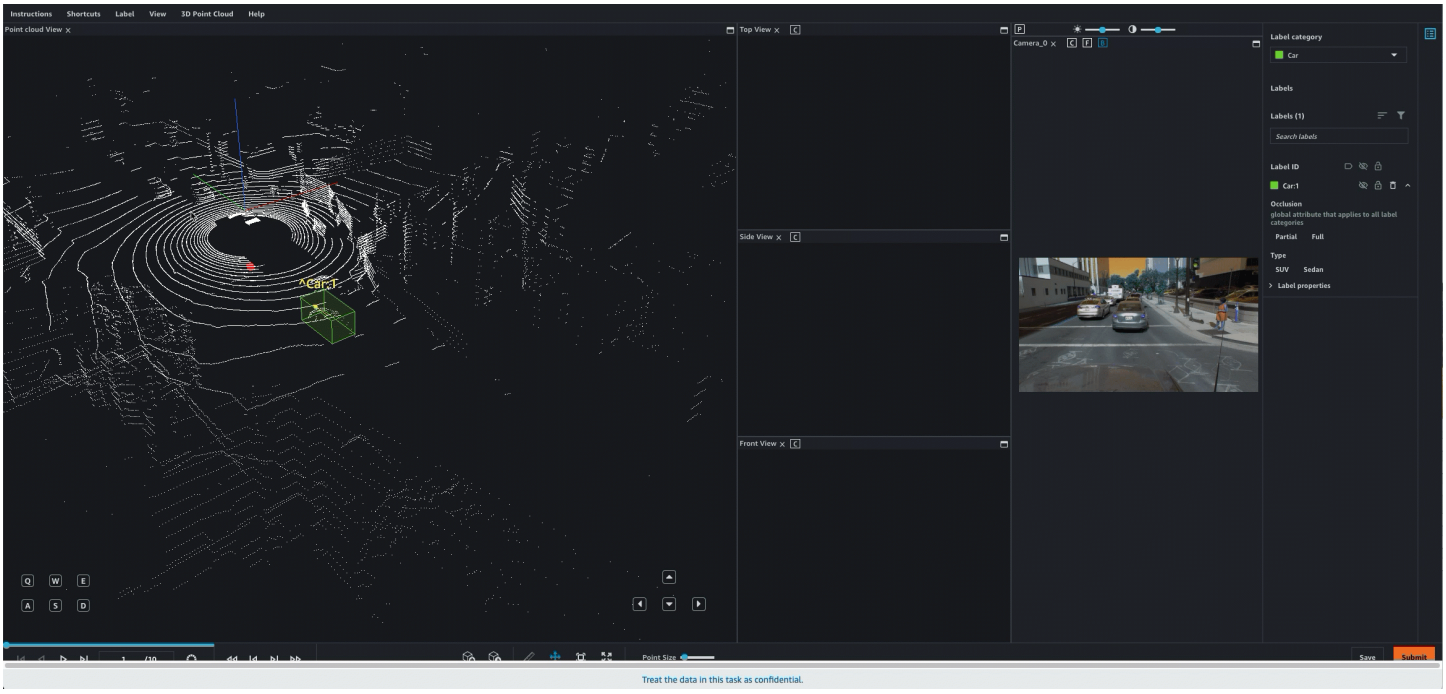
Menavigasi UI

Anda dapat menavigasi dalam adegan 3D menggunakan keyboard dan mouse Anda. Anda dapat:

- Klik dua kali pada objek tertentu di titik awan untuk memperbesarnya.
- Anda dapat menggunakan tombol [dan] pada keyboard Anda untuk memperbesar dan berpindah dari satu label ke label berikutnya. Jika tidak ada label yang dipilih, saat Anda memilih [atau], UI akan memperbesar label pertama diID labelDaftar.
- Gunakan mouse-scroller atau trackpad untuk memperbesar dan memperkecil titik awan.
- Gunakan kedua tombol panah keyboard dan tombol Q, E, A, dan D untuk bergerak Naik, Bawah, Kiri, Kanan. Gunakan tombol keyboard W dan S untuk memperbesar dan memperkecil.

Setelah Anda menempatkan cuboids dalam adegan 3D, tampilan samping akan muncul dengan tiga tampilan yang diproyeksikan: atas, samping, dan belakang. Pandangan samping ini menunjukkan titik-titik di dalam dan di sekitar kubus yang ditempatkan dan membantu pekerja memperbaiki batas berbentuk kubus di daerah itu. Pekerja dapat memperbesar dan memperkecil masing-masing tampilan samping menggunakan mouse mereka.

Video berikut menunjukkan pergerakan di sekitar awan titik 3D dan di tampilan samping.



Ketika Anda berada di UI pekerja, Anda akan melihat menu berikut:

- Petunjuk- Tinjau instruksi ini sebelum memulai tugas Anda.
- Pintasan jalan si- Gunakan menu ini untuk melihat pintasan keyboard yang dapat Anda gunakan untuk menavigasi titik awan dan menggunakan alat anotasi yang disediakan.

- Label- Gunakan menu ini untuk memodifikasi berbentuk kubus. Pertama, pilih berbentuk kubus, lalu pilih opsi dari menu ini. Menu ini mencakup alat pelabelan bantu seperti mengatur berbentuk kubus ke tanah dan secara otomatis memasang kubus ke batas objek.
- Lihat- Gunakan menu ini untuk mengaktifkan dan menonaktifkan opsi tampilan yang berbeda. Misalnya, Anda dapat menggunakan menu ini untuk menambahkan ground mesh ke titik awan, dan untuk memilih proyeksi titik awan.
- Awan Titik 3D- Gunakan menu ini untuk menambahkan atribut tambahan ke titik-titik di titik awan, seperti warna, dan intensitas piksel. Perhatikan bahwa opsi ini mungkin tidak tersedia.

Saat Anda membuka tugas, ikon adegan bergerak aktif, dan Anda dapat bergerak di sekitar titik awan menggunakan mouse dan tombol navigasi di area titik awan layar. Untuk kembali ke tampilan asli yang Anda lihat saat pertama kali membuka tugas, pilih ikon reset scene.

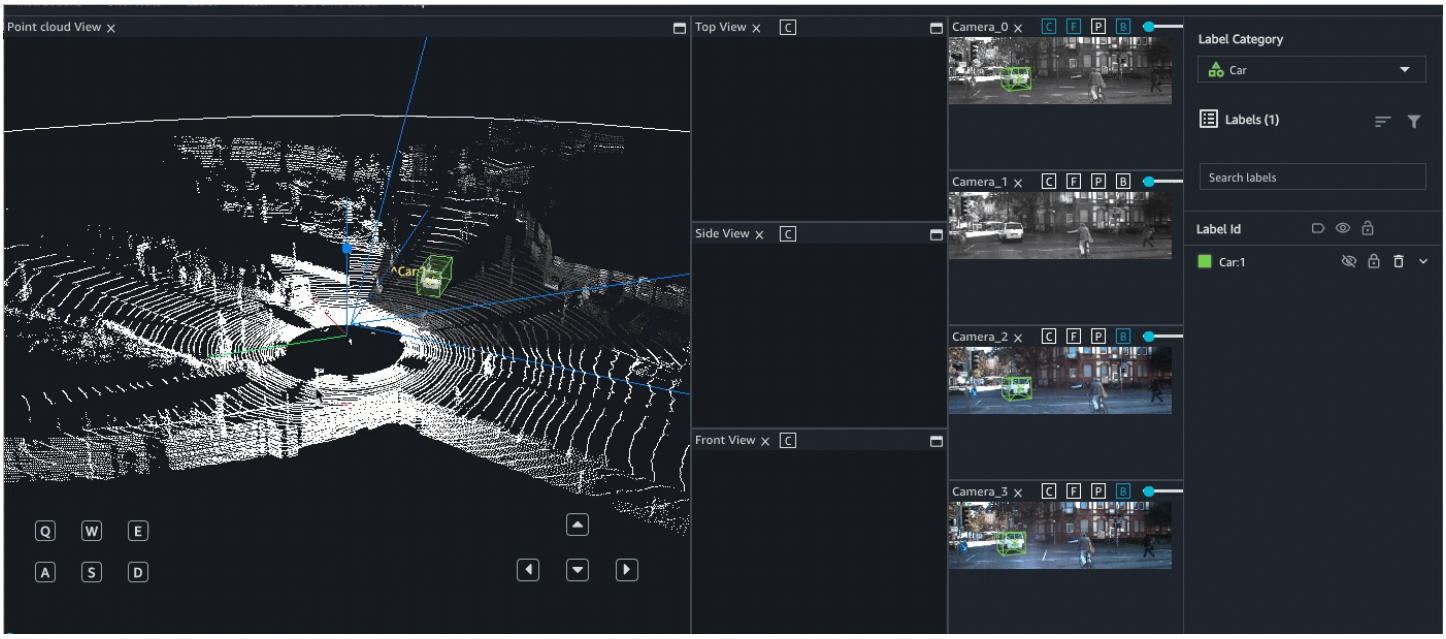
Setelah Anda memilih ikon add cuboid, Anda dapat menambahkan kubus ke titik awan dan gambar (jika disertakan). Anda harus memilih ikon adegan pindah lagi untuk pindah ke area lain di awan titik 3D atau gambar.

Untuk menutup semua panel di sebelah kanan dan membuat awan titik 3D layar penuh, pilih ikon layar penuh.

Jika gambar kamera disertakan, Anda mungkin memiliki opsi tampilan berikut:

- C- Lihat sudut kamera pada tampilan titik awan.
- F- Lihat frustum, atau bidang pandang, kamera yang digunakan untuk menangkap gambar itu pada tampilan cloud titik.
- P- Lihat titik awan yang dilapis pada gambar.
- B— Lihat kubus dalam gambar.

Video berikut menunjukkan cara menggunakan opsi penayangan ini. KlasterFdigunakan untuk melihat bidang pandang kamera (area abu-abu),CPilihan menunjukkan arah kamera menghadap dan sudut kamera (garis biru), danBdigunakan untuk melihat berbentuk kubus.



Hapus kuboid

Anda dapat memilih ID berbentuk kubus atau label dan:

- Hapus kubus individu dalam bingkai saat ini yang Anda lihat.
- Hapus semua kuboid dengan ID label sebelum atau sesudah bingkai yang Anda lihat.
- Hapus semua kuboid dengan ID label itu di semua bingkai.

Kasus penggunaan umum untuk penghapusan berbentuk kubus adalah jika objek meninggalkan adegan.

Anda dapat menggunakan satu atau beberapa opsi ini untuk menghapus kuboid yang ditempatkan secara manual dan diinterpolasi dengan ID label yang sama.

- Untuk menghapus semua cuboid sebelum atau setelah frame Anda saat ini, pilih cuboid, pilihLabelitem menu di bagian atas UI dan kemudian pilih salah satuHapus di frame sebelumnyaatauHapus di frame berikutnya. Gunakan menu Pintasan untuk melihat tombol pintas yang dapat Anda gunakan untuk opsi ini.
- Untuk menghapus label di semua frame, pilihHapus di semua framedariLabelmenu, atau gunakan pintasanShift + Hapuspada keyboard Anda.

- Untuk menghapus cuboid individu dari satu frame, pilih cuboid dan pilih ikon trashcan () di samping ID label itu diID labelsidebar di sebelah kanan atau gunakan tombol Delete pada keyboard Anda untuk menghapus berbentuk kubus itu.

Jika Anda telah menempatkan lebih dari satu kubus secara manual dengan label yang sama dalam bingkai yang berbeda, saat Anda menghapus salah satu kubus yang ditempatkan secara manual, semua kubus yang diinterpolasi menyesuaikan. Penyesuaian ini terjadi karena UI menggunakan kubus yang ditempatkan secara manual sebagai titik jangkar saat menghitung lokasi kubus yang diinterpolasi. Saat Anda menghapus salah satu titik jangkar ini, UI harus menghitung ulang posisi kubus yang diinterpolasi.

Jika Anda menghapus berbentuk kubus dari bingkai, tetapi kemudian memutuskan bahwa Anda ingin mendapatkannya kembali, Anda dapat menggunakanDuplikat ke frame sebelumnyaatauDuplikat ke frame berikutnyapilihan diLabelmenu untuk menyalin berbentuk kubus ke semua sebelumnya atau semua frame berikut, masing-masing.

Kategori Label Edit Massal dan Atribut Bingkai

Anda dapat mengedit atribut label dan atribut bingkai secara massal.

Ketika Anda mengedit atribut secara massal, Anda menentukan satu atau beberapa rentang frame yang ingin Anda terapkan edit. Atribut yang Anda pilih diedit di semua frame dalam kisaran tersebut, termasuk bingkai awal dan akhir yang Anda tentukan. Saat Anda mengedit atribut label secara massal, rentang yang Anda tentukanharusmengandung label yang dilampirkan atribut label. Jika Anda menentukan bingkai yang tidak mengandung label ini, Anda akan menerima galat.

Untuk mengedit atribut secara massal Andaharustentukan nilai yang diinginkan untuk atribut terlebih dahulu. Misalnya, jika Anda ingin mengubah atribut dariyakepadaTidak, Anda harus memilihTidak, dan kemudian melakukan pengeditan massal.

Anda juga dapat menentukan nilai baru untuk atribut yang belum diisi dan kemudian menggunakan fitur edit massal untuk mengisi nilai tersebut dalam beberapa frame. Untuk melakukannya, pilih nilai yang diinginkan untuk atribut dan selesaikan prosedur berikut.

Untuk mengedit label atau atribut secara massal:

1. Gunakan mouse Anda untuk mengklik kanan atribut yang ingin Anda edit secara massal.

2. Tentukan rentang bingkai yang ingin Anda terapkan pengeditan massal menggunakan tanda hubung (-) di kotak teks. Misalnya, jika Anda ingin menerapkan pengeditan ke bingkai satu hingga sepuluh, masukkan`1-10`. Jika Anda ingin menerapkan edit ke frame dua hingga lima, delapan sampai sepuluh dan dua puluh masukkan`2-5,8-10,20`.
3. PilihTerkonfirmasi.

Jika Anda mendapatkan pesan kesalahan, verifikasi bahwa Anda memasukkan rentang yang valid dan bahwa label yang terkait dengan atribut label yang Anda edit (jika berlaku) ada di semua frame yang ditentukan.

Anda dapat dengan cepat menambahkan label ke semua frame sebelumnya atau berikutnya menggunakanDuplikat ke frame sebelumnyadanDuplikat ke frame berikutnyapilihan diLabelmenu di bagian atas layar.

Panduan ikon

Gunakan tabel ini untuk mempelajari tentang ikon yang Anda lihat di portal tugas pekerja Anda.

Ikon		Deskripsi
	tambahkan kubus	Pilih ikon ini untuk menambahkan berbentuk kubus. Setiap berbentuk kubus yang Anda tambahkan dikaitkan dengan kategori yang Anda pilih.
	edit kubus	Pilih ikon ini untuk mengedit berbentuk kubus. Setelah Anda menambahkan berbentuk kubus, Anda dapat mengedit dimensi, lokasi, dan orientasinya. Setelah cuboid ditambahkan, secara otomatis beralih ke mode edit berbentuk kubus.
	penggaris	Gunakan ikon ini untuk mengukur jarak, dalam meter, di titik awan. Anda mungkin ingin menggunakan alat ini jika instruksi Anda meminta Anda untuk membuat anotasi semua objek dalam jarak tertentu dari pusat berbentuk kubus atau objek yang digunakan untuk menangkap data. Saat Anda memilih ikon ini, Anda dapat menempatk an titik awal (penanda pertama) di mana saja di titik

Ikon		Deskripsi
		awan dengan memilihnya dengan mouse Anda. Alat ini akan secara otomatis menggunakan interpolasi untuk menempatkan penanda pada titik terdekat dalam jarak ambang ke lokasi yang Anda pilih, jika tidak penanda akan ditempatkan di tanah. Jika Anda menempatkan titik awal karena kesalahan, Anda dapat menggunak an tombol Escape untuk mengembalikan penempatan penanda. Setelah Anda menempatkan penanda pertama, Anda melihat garis putus-putus dan label dinamis yang menunjukkan jarak yang telah Anda pindah dari penanda pertama. Klik di tempat lain di titik awan untuk menempatkan penanda kedua. Ketika Anda menempatk an penanda kedua, garis putus-putus menjadi padat, dan jarak diatur. Setelah Anda mengatur jarak, Anda dapat mengeditn ya dengan memilih salah satu penanda. Anda dapat menghapus penggaris dengan memilih di mana saja pada penggaris dan menggunakan tombol Hapus pada keyboard Anda.
	Atur ulang adegan	Pilih ikon ini untuk mengatur ulang tampilan titik awan, panel samping, dan jika berlaku, semua gambar ke posisi semula saat tugas pertama kali dibuka.
	pindahkan adegan	Pilih ikon ini untuk memindahkan adegan. Secara default, ikon ini dipilih saat Anda pertama kali memulai tugas.
	Layar penuh	Pilih ikon ini untuk membuat visualisasi awan titik 3D layar penuh dan runtuh semua panel samping.
	frame	Pilih ikon ini untuk memuat frame tambahan.

Ikon		Deskripsi
	label	Sembunyikan label dalam visualisasi awan titik 3D, dan jika berlaku, dalam gambar.
	Tampilkan label	Tampilkan label dalam visualisasi awan titik 3D, dan jika berlaku, dalam gambar.
	Hapus label	Menghapus label. Opsi ini hanya dapat digunakan untuk menghapus label yang telah Anda buat atau disesuaikan secara manual.

Pintasan jalan si

Pintasan yang tercantum dalamPintasan jalan simenu dapat membantu Anda menavigasi titik awan 3D dan menggunakan alat untuk menambah dan mengedit kubus.

Sebelum Anda memulai tugas Anda, disarankan agar Anda meninjauPintasan jalan simenu dan berkenalan dengan perintah-perintah ini. Anda perlu menggunakan beberapa kontrol berbentuk kubus 3D untuk mengedit berbentuk kubus Anda.

Rilis, Hentikan, dan Lanjutkan, dan Tolak Tugas

Saat Anda membuka tugas pelabelan, tiga tombol di kanan atas memungkinkan Anda menolak tugas (tugas tolak balik), lepaskan (tugas rilis), dan berhenti dan melanjutkan di lain waktu (Berhenti dan melanjutkan nanti). Daftar berikut menjelaskan apa yang terjadi ketika Anda memilih salah satu opsi ini:

- tugas tolak balik: Anda hanya harus menolak tugas jika ada sesuatu yang salah dengan tugas, seperti masalah dengan awan titik 3D, gambar atau UI. Jika Anda menolak tugas, Anda tidak akan dapat kembali ke tugas.
- tugas rilis: Gunakan opsi ini untuk melepaskan tugas dan memungkinkan orang lain untuk mengerjakannya. Ketika Anda melepaskan tugas, Anda kehilangan semua pekerjaan yang dilakukan pada tugas itu dan pekerja lain di tim Anda dapat mengambilnya. Jika cukup pekerja mengambil tugas, Anda mungkin tidak akan dapat kembali ke sana. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja. Jika tugas masih tersedia, statusnya akanTersedia. Jika pekerja lain mengambilnya, itu akan hilang dari portal Anda.

- Berhenti dan melanjutkan nanti: Anda dapat menggunakanBerhenti dan melanjutkan nantitombol untuk berhenti bekerja dan kembali ke tugas di lain waktu. Anda harus menggunakanSimpantombol untuk menyimpan pekerjaan Anda sebelum Anda memilihBerhenti dan melanjutkan nanti. Bila Anda memilih tombol ini dan kemudian pilihTerkonfirmasi, Anda dikembalikan ke portal pekerja, dan status tugasnya adalahDihentikan. Anda dapat memilih tugas yang sama untuk melanjutkan pekerjaan di atasnya.

  Ketahuilah bahwa orang yang membuat tugas pelabelan Anda menentukan batas waktu di mana semua tugas banyak diselesaikan. Jika Anda tidak kembali ke dan menyelesaikan tugas ini dalam batas waktu itu, itu akan kedaluwarsa dan pekerjaan Anda tidak akan diserahkan. Hubungi administrator untuk informasi lebih lanjut.

Menyimpan Pekerjaan Anda dan Mengirimkan

Anda harus menyimpan pekerjaan Anda secara berkala. Ground Truth akan secara otomatis menyimpan pekerjaan Anda pernah 15 menit.

Ketika Anda membuka tugas, Anda harus menyelesaikan pekerjaan Anda di atasnya sebelum menekanKIRIMKAN.

Verifikasi dan Sesuaikan Label

Ketika label pada dataset perlu divalidasi, Amazon SageMaker Ground Truth menyediakan fungsionalitas agar pekerja memverifikasi bahwa label sudah benar atau untuk menyesuaikan label sebelumnya.

Jenis pekerjaan ini terbagi dalam dua kategori berbeda:

- Verifikasi- Pekerja menunjukkan apakah label yang ada benar, atau menilai kualitasnya, dan dapat menambahkan komentar untuk menjelaskan alasan mereka. Pekerja tidak akan dapat memodifikasi atau menyesuaikan label.

  Jika Anda membuat awan titik 3D atau penyesuaian label bingkai video atau pekerjaan verifikasi, Anda dapat memilih untuk membuat atribut kategori label (tidak didukung untuk segmentasi semantik titik awan 3D) dan atribut bingkai dapat diedit oleh pekerja.
- Penyesuaian label- Pekerja menyesuaikan anotasi sebelumnya dan, jika berlaku, label kategori dan atribut bingkai untuk memperbaikinya.

Ground Truth berikuttipe tugas bawaandukungan penyesuaian dan verifikasi pelabelan pekerjaan:

- Kotak pembatas
- Segmentasi semantik
- Deteksi objek awan titik 3D, pelacakan objek cloud titik 3D, dan segmentasi semantik awan titik 3D
- Semua deteksi objek bingkai video dan jenis tugas pelacakan objek bingkai video - kotak pembatas, polyline, poligon, dan keypoint

> **Tip**
>
> Untuk pekerjaan verifikasi pelabelan cloud titik 3D dan bingkai video, disarankan agar Anda menambahkan atribut kategori label baru atau atribut bingkai ke pekerjaan pelabelan. Pekerja dapat menggunakan atribut ini untuk memverifikasi label individu atau seluruh frame. Untuk mempelajari lebih lanjut tentang kategori label dan atribut bingkai, lihatAntarmuka Pengguna Pekerja (UI)untuk cloud titik 3D danAntarmuka Pengguna Pekerja (UI)untuk bingkai video.

Anda dapat memulai verifikasi label dan pekerjaan penyesuaian menggunakan SageMaker konsol atau API.

Topik

- Persyaratan untuk Membuat Pekerjaan Verifikasi dan Penyesuaian Pelabelan
- Membuat Job Verifikasi Label (Konsol)
- Membuat Label Adjustment Job (Console)
- Memulai Verifikasi Label atau Job Penyesuaian (API)
- Label Verifikasi dan Penyesuaian Data dalam Manifes Output
- Perhatian dan Pertimbangan

Persyaratan untuk Membuat Pekerjaan Verifikasi dan Penyesuaian Pelabelan

Untuk membuat verifikasi label atau pekerjaan penyesuaian, kriteria berikut harus dipenuhi.

- Untuk tugas pelabelan non streaming: File manifes masukan yang Anda gunakan harus berisi nama atribut label (`LabelAttributeName`) dari label yang ingin Anda sesuaikan. Saat Anda menyusun tugas pelabelan yang berhasil diselesaikan, file manifes keluaran digunakan sebagai file manifes masukan untuk tugas baru yang dirantai. Untuk mempelajari lebih lanjut tentang format file manifes keluaran yang dihasilkan Ground Truth untuk setiap jenis tugas, lihatData Output.

Untuk tugas pelabelan streaming: Pesan Amazon SNS yang Anda kirim ke topik masukan Amazon SNS dari pekerjaan penyesuaian atau pelabelan verifikasi harus berisi nama atribut label label yang ingin disesuaikan atau diverifikasi. Untuk melihat contoh bagaimana Anda dapat membuat pekerjaan pelabelan penyesuaian atau verifikasi dengan pekerjaan pelabelan streaming, lihat iniNotebook Jupyterdi GitHub.

- Jenis tugas verifikasi atau penyesuaian pelabelan pekerjaan harus sama dengan jenis tugas dari pekerjaan asli kecuali Anda menggunakanLabel verifikasi gambarjenis tugas untuk memverifikasi kotak pembatas atau label gambar segmentasi semantik. Lihat titik bullet berikutnya untuk detail lebih lanjut tentang persyaratan jenis tugas bingkai video.
- Untuk verifikasi anotasi bingkai video dan pekerjaan penyesuaian, Anda harus menggunakan jenis tugas anotasi yang sama yang digunakan untuk membuat anotasi dari tugas pelabelan sebelumnya. Misalnya, jika Anda membuat pekerjaan deteksi objek bingkai video agar pekerja menggambar kotak pembatas di sekitar objek, dan kemudian Anda membuat pekerjaan penyesuaian deteksi objek video, Anda harus menentukankotak pembatassebagai jenis tugas anotasi. Untuk mempelajari lebih lanjut jenis tugas anotasi bingkai video, lihatJenis.
- Jenis tugas yang Anda pilih untuk penyesuaian atau verifikasi pelabelan pekerjaan harus mendukung alur kerja audit. Ground Truth berikuttipe tugas bawaanmendukung pekerjaan penyesuaian dan pelabelan verifikasi: kotak pembatas, segmentasi semantik, deteksi objek awan titik 3D, pelacakan objek awan titik 3D, dan segmentasi semantik awan titik 3D, dan semua deteksi objek bingkai video dan jenis tugas pelacakan objek bingkai video - kotak pembatas, polyline, poligon dan titik kunci.

Membuat Job Verifikasi Label (Konsol)

Pekerjaan pelabelan kotak pembatas dan segmentasi semantik dibuat dengan memilihVerifikasijenis tugas di konsol. Untuk membuat pekerjaan verifikasi untuk jenis tugas cloud titik 3D dan bingkai video, Anda harus memilih jenis tugas yang sama dengan pekerjaan pelabelan asli dan memilih untuk menampilkan label yang ada. Gunakan salah satu bagian berikut untuk membuat tugas verifikasi label untuk jenis tugas Anda.

Topik

- Membuat Lowongan Verifikasi Label Gambar (Konsol)
- Membuat Job Verifikasi Label Bingkai Video atau Point Cloud (Konsol)

Membuat Lowongan Verifikasi Label Gambar (Konsol)

Gunakan prosedur berikut untuk membuat kotak pembatas atau tugas verifikasi segmentasi semantik menggunakan konsol. Prosedur ini mengasumsikan bahwa Anda telah membuat kotak pembatas atau tugas pelabelan segmentasi semantik dan statusnya adalah Complete. Ini pekerjaan pelabelan yang menghasilkan label yang ingin Anda verifikasi.

Untuk membuat pekerjaan verifikasi label gambar:

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/dan memiilihLowongan.
2. Mulai tugas pelabelan baru denganrantaipekerjaan sebelumnya atau mulai dari awal, menentukan manifes masukan yang berisi objek data berlabel.
3. DiJenis tugaspanel, pilihVerifikasi.
4. Pilih Selanjutnya.
5. DiPekerjabagian, pilih jenis tenaga kerja yang ingin Anda gunakan. Untuk detail lebih lanjut tentang opsi tenaga kerja Anda, lihatMembuat dan Mengelola Tenaga Kerja.
6. (Opsional) Setelah Anda memilih tenaga kerja Anda, tentukanBatas waktu habisdanWaktu kedaluwarsa.
7. DiOpsi tampilan label yang adapanel, sistem menampilkan nama atribut label yang tersedia dalam manifes Anda. Pilih nama atribut label yang mengidentifikasi label yang ingin diverifikasi oleh pekerja. Ground Truth mencoba mendeteksi dan mengisi nilai-nilai ini dengan menganalisis manifes, tetapi Anda mungkin perlu menetapkan nilai yang benar.
8. Gunakan area instruksi dari perancang alat untuk memberikan konteks tentang apa yang diminta pelabel sebelumnya dan apa yang perlu diperiksa oleh pemverifikasi saat ini.

   Anda dapat menambahkan label baru yang dipilih pekerja untuk memverifikasi label. Misalnya, Anda dapat meminta pekerja untuk memverifikasi kualitas gambar, dan memberikan labelHapusdanKabur. Pekerja juga akan memiliki opsi untuk menambahkan komentar untuk menjelaskan pilihan mereka.
9. PilihLihat pratinjauuntuk memeriksa apakah alat ini menampilkan label sebelumnya dengan benar dan menyajikan tugas verifikasi label dengan jelas.
10. Pilih Buat. Ini akan membuat dan memulai pekerjaan pelabelan Anda.

Membuat Job Verifikasi Label Bingkai Video atau Point Cloud (Konsol)

Gunakan prosedur berikut untuk membuat tugas verifikasi cloud titik 3D atau bingkai video menggunakan konsol. Prosedur ini mengasumsikan bahwa Anda telah membuat pekerjaan pelabelan menggunakan jenis tugas yang menghasilkan jenis label yang ingin Anda verifikasi dan statusnya Lengkap.

Untuk membuat pekerjaan verifikasi label gambar:

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/dan memiilihLowongan.
2. Mulai tugas pelabelan baru denganrantaipekerjaan sebelumnya atau mulai dari awal, menentukan manifes masukan yang berisi objek data berlabel.
3. DiJenis tugaspanel, pilih jenis tugas yang sama sebagai pekerjaan pelabelan yang Anda dirantai. Misalnya, jika pekerjaan pelabelan asli adalah pekerjaan pelabelan titik kunci deteksi objek bingkai video, pilih jenis tugas itu.
4. Pilih Selanjutnya.
5. DiPekerjabagian, pilih jenis tenaga kerja yang ingin Anda gunakan. Untuk detail lebih lanjut tentang opsi tenaga kerja Anda, lihatMembuat dan Mengelola Tenaga Kerja.
6. (Opsional) Setelah Anda memilih tenaga kerja Anda, tentukanBatas waktu habisdanWaktu kedaluwarsa.
7. Alihkan sakelar di sampingLabel tampilan yang ada.
8. PilihVerifikasi.
9. UntukNama atribut, pilih nama dari manifes Anda yang sesuai dengan label yang ingin ditampilkan untuk verifikasi. Anda hanya akan melihat nama atribut label untuk label yang cocok dengan jenis tugas yang Anda pilih di layar sebelumnya. Ground Truth mencoba mendeteksi dan mengisi nilai-nilai ini dengan menganalisis manifes, tetapi Anda mungkin perlu menetapkan nilai yang benar.
10. Gunakan area instruksi dari perancang alat untuk memberikan konteks tentang apa yang diminta pelabel sebelumnya dan apa yang perlu diperiksa oleh pemverifikasi saat ini.

    Anda tidak dapat memodifikasi atau menambahkan label baru. Anda dapat menghapus, memodifikasi, dan menambahkan atribut kategori label baru atau atribut bingkai. Dianjurkan agar Anda menambahkan atribut kategori label baru atau atribut bingkai ke pekerjaan pelabelan. Pekerja dapat menggunakan atribut ini untuk memverifikasi label individu atau seluruh frame.

Secara default, atribut kategori label yang sudah ada sebelumnya dan atribut bingkai tidak akan dapat diedit oleh pekerja. Jika Anda ingin membuat kategori label atau atribut frame dapat diedit, pilihIzinkan pekerja untuk mengedit atribut inicentang kotak untuk atribut itu.

Untuk mempelajari lebih lanjut tentang kategori label dan atribut bingkai, lihatAntarmuka Pengguna Pekerja (UI)untuk cloud titik 3D danAntarmuka Pengguna Pekerja (UI)untuk bingkai video.

11. PilihLihat pratinjauuntuk memeriksa apakah alat ini menampilkan label sebelumnya dengan benar dan menyajikan tugas verifikasi label dengan jelas.
12. Pilih Buat. Ini akan membuat dan memulai pekerjaan pelabelan Anda.

Membuat Label Adjustment Job (Console)

Gunakan salah satu bagian berikut untuk membuat tugas verifikasi label untuk jenis tugas Anda.

Topik

- Membuat Image Label Adjustment Job (Console)
- Membuat Point Cloud atau Video Frame Label Adjustment Job (Konsol)

Membuat Image Label Adjustment Job (Console)

Gunakan prosedur berikut untuk membuat kotak pembatas atau tugas pelabelan penyesuaian segmentasi semantik menggunakan konsol. Prosedur ini mengasumsikan bahwa Anda telah membuat kotak pembatas atau tugas pelabelan segmentasi semantik dan statusnya adalah Complete. Ini pekerjaan pelabelan yang menghasilkan label yang ingin Anda sesuaikan.

Untuk membuat pekerjaan penyesuaian label gambar (konsol)

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/dan memiilihLowongan.
2. Mulai tugas pelabelan baru denganrantaipekerjaan sebelumnya atau mulai dari awal, menentukan manifes masukan yang berisi objek data berlabel.
3. Pilih jenis tugas yang sama dengan pekerjaan pelabelan asli.
4. Pilih Selanjutnya.
5. DiPekerjabagian, pilih jenis tenaga kerja yang ingin Anda gunakan. Untuk detail lebih lanjut tentang opsi tenaga kerja Anda, lihatMembuat dan Mengelola Tenaga Kerja.

6. (Opsional) Setelah Anda memilih tenaga kerja Anda, tentukanBatas waktu habisdanWaktu kedaluwarsa.
7. PerluasOpsi tampilan label yang adadengan memilih panah di sebelah judul.
8. Centang kotak di sebelahSaya ingin menampilkan label yang ada dari dataset untuk pekerjaan ini.
9. UntukNama atribut, pilih nama dari manifes Anda yang sesuai dengan label yang ingin Anda tampilkan untuk penyesuaian. Anda hanya akan melihat nama atribut label untuk label yang cocok dengan jenis tugas yang Anda pilih di layar sebelumnya. Ground Truth mencoba mendeteksi dan mengisi nilai-nilai ini dengan menganalisis manifes, tetapi Anda mungkin perlu menetapkan nilai yang benar.
10. Gunakan area instruksi dari perancang alat untuk memberikan konteks tentang apa yang dilabel sebelumnya ditugaskan untuk dilakukan dan apa yang perlu diperiksa dan disesuaikan oleh pemverifikasi saat ini.
11. PilihLihat pratinjauuntuk memeriksa bahwa alat menunjukkan label sebelumnya dengan benar dan menyajikan tugas dengan jelas.
12. Pilih Buat. Ini akan membuat dan memulai pekerjaan pelabelan Anda.

Membuat Point Cloud atau Video Frame Label Adjustment Job (Konsol)

Gunakan prosedur berikut untuk membuat tugas penyesuaian 3D point cloud atau frame video menggunakan konsol. Prosedur ini mengasumsikan bahwa Anda telah membuat pekerjaan pelabelan menggunakan jenis tugas yang menghasilkan jenis label yang ingin Anda verifikasi dan statusnya Lengkap.

Untuk membuat awan titik 3D atau pekerjaan penyesuaian label bingkai video (konsol)

1. Buka SageMaker konsol:https://console.aws.amazon.com/sagemaker/dan memiilihLowongan.
2. Mulai tugas pelabelan baru denganrantaipekerjaan sebelumnya atau mulai dari awal, menentukan manifes masukan yang berisi objek data berlabel.
3. Pilih jenis tugas yang sama dengan pekerjaan pelabelan asli.
4. Alihkan sakelar di sampingLabel tampilan yang ada.
5. PilihPenyesuaian.
6. UntukNama atribut, pilih nama dari manifes Anda yang sesuai dengan label yang ingin Anda tampilkan untuk penyesuaian. Anda hanya akan melihat nama atribut label untuk label yang cocok dengan jenis tugas yang Anda pilih di layar sebelumnya. Ground Truth mencoba

mendeteksi dan mengisi nilai-nilai ini dengan menganalisis manifes, tetapi Anda mungkin perlu menetapkan nilai yang benar.

7. Gunakan area instruksi dari perancang alat untuk memberikan konteks tentang apa yang diminta pelabel sebelumnya dan apa yang perlu diperiksa oleh pengatur saat ini.

   Anda tidak dapat menghapus atau memodifikasi label yang ada tetapi Anda dapat menambahkan label baru. Anda dapat menghapus, memodifikasi, dan menambahkan atribut kategori label baru atau atribut bingkai.

   Secara default, atribut kategori label yang sudah ada sebelumnya dan atribut bingkai akan dapat diedit oleh pekerja. Jika Anda ingin membuat kategori label atau atribut frame tidak dapat diedit, batalkanIzinkan pekerja untuk mengedit atribut inicentang kotak untuk atribut itu.

   Untuk mempelajari lebih lanjut tentang kategori label dan atribut bingkai, lihatAntarmuka Pengguna Pekerja (UI)untuk cloud titik 3D danAntarmuka Pengguna Pekerja (UI)untuk bingkai video.
8. PilihLihat pratinjauuntuk memeriksa bahwa alat menunjukkan label sebelumnya dengan benar dan menyajikan tugas dengan jelas.
9. Pilih Buat. Ini akan membuat dan memulai pekerjaan pelabelan Anda.

Memulai Verifikasi Label atau Job Penyesuaian (API)

Mulai verifikasi label atau pekerjaan penyesuaian dengan merantai pekerjaan yang berhasil diselesaikan atau memulai pekerjaan baru dari awal menggunakan`CreateLabelingJob`operasi. Prosedurnya hampir sama dengan menyiapkan pekerjaan pelabelan baru`CreateLabelingJob`, dengan beberapa modifikasi. Gunakan bagian berikut untuk mempelajari modifikasi apa yang diperlukan untuk merantai pekerjaan pelabelan untuk membuat pekerjaan penyesuaian atau verifikasi pelabelan.

Saat Anda membuat pekerjaan pelabelan penyesuaian atau verifikasi menggunakan Ground Truth API, Andaharusmenggunakan yang berbeda`LabelAttributeName`dari pekerjaan pelabelan asli. Pekerjaan pelabelan asli adalah pekerjaan yang digunakan untuk membuat label yang ingin Anda sesuaikan atau diverifikasi.

> **Important**
>
> Berkas konfigurasi kategori label yang Anda identifikasi untuk pekerjaan penyesuaian atau verifikasi`LabelCategoryConfigS3Uri`dari`CreateLabelingJob`harus berisi label yang

sama yang digunakan dalam pekerjaan pelabelan asli. Anda dapat menambahkan label baru. Untuk pekerjaan cloud titik 3D dan bingkai video, Anda dapat menambahkan kategori label dan atribut bingkai baru ke file konfigurasi kategori label.

Kotak Bounding dan Segmentasi Semantik

Untuk membuat kotak pembatas atau verifikasi label segmentasi semantik atau pekerjaan penyesuaian, gunakan panduan berikut untuk menentukan atribut API untuk`CreateLabelingJob`operasi.

- Gunakan`LabelAttributeName`parameter untuk menentukan nama label keluaran yang ingin Anda gunakan untuk label yang diverifikasi atau disesuaikan. Anda harus menggunakan yang berbeda`LabelAttributeName`daripada yang digunakan untuk pekerjaan pelabelan asli.
- Jika Anda merantai pekerjaan, label dari pekerjaan pelabelan sebelumnya untuk disesuaikan atau diverifikasi akan ditentukan dalam template UI kustom. Untuk mempelajari cara membuat template kustom, lihatBuat Templat Tugas Pekerja Kustom.

  Identifikasi lokasi template UI di`UiTemplateS3Uri`parameter. SageMaker menyediakan widget yang dapat Anda gunakan dalam template kustom Anda untuk menampilkan label lama. Gunakan`initial-value`atribut di salah satu elemen kerumunan berikut untuk mengekstrak label yang memerlukan verifikasi atau penyesuaian dan memasukkannya ke dalam template tugas Anda:

  - kerumunan-semantik-segmentasi—Gunakan elemen kerumunan ini di template tugas UI kustom Anda untuk menentukan label segmentasi semantik yang perlu diverifikasi atau disesuaikan.
  - kerumunan-batas-kotak—Gunakan elemen kerumunan ini di template tugas UI kustom Anda untuk menentukan label kotak pembatas yang perlu diverifikasi atau disesuaikan.
- Klaster`LabelCategoryConfigS3Uri`parameter harus berisi kategori label yang sama dengan pekerjaan pelabelan sebelumnya.
- Gunakan kotak pembatas atau penyesuaian segmentasi semantik atau verifikasi lambda ARN untuk`PreHumanTaskLambdaArn`dan`AnnotationConsolidationLambdaArn`:
  - Untuk kotak pembatas, penyesuaian pelabelan pekerjaan lambda fungsi ARN diakhiri dengan`AdjustmentBoundingBox`dan fungsi lambda verifikasi ARN diakhiri dengan`VerificationBoundingBox`.

- Untuk segmentasi semantik, penyesuaian pelabelan pekerjaan lambda fungsi ARN diakhiri dengan`AdjustmentSemanticSegmentation`dan fungsi lambda verifikasi ARN diakhiri dengan`VerificationSemanticSegmentation`.

Awan Titik 3D dan Bingkai Video

- Gunakan`LabelAttributeName`parameter untuk menentukan nama label keluaran yang ingin Anda gunakan untuk label yang diverifikasi atau disesuaikan. Anda harus menggunakan yang berbeda`LabelAttributeName`daripada yang digunakan untuk pekerjaan pelabelan asli.
- Anda harus menggunakan UI tugas manusia Amazon Resource Name (ARN) (`HumanTaskUiArn`) digunakan untuk pekerjaan pelabelan asli. Untuk melihat ARN yang didukung, lihat`HumanTaskUiArn`.
- Dalam file konfigurasi kategori label, Anda harus menentukan nama atribut label (`LabelAttributeName`) dari pekerjaan pelabelan sebelumnya yang Anda gunakan untuk membuat pekerjaan pelabelan penyesuaian atau verifikasi di`auditLabelAttributeName`parameter.
- Anda menentukan apakah pekerjaan pelabelan Andaverifikasiataupengaturanpelabelan pekerjaan menggunakan`editsAllowed`parameter dalam file konfigurasi kategori label Anda yang diidentifikasi oleh`LabelCategoryConfigS3Uri`parameter.
  - Untukverifikasipelabelan pekerjaan, Anda harus menggunakan`editsAllowed`parameter untuk menentukan bahwa semua label tidak dapat diubah.`editsAllowed`harus diatur ke`"none"`di setiap entri di`labels`. Opsional, Anda dapat menentukan apakah atau tidak label kategori atribut dan frame atribut dapat disesuaikan oleh pekerja.
  - Secara opsional, untukpengaturanpelabelan pekerjaan, Anda dapat menggunakan`editsAllowed`parameter untuk menentukan label, atribut kategori label, dan atribut bingkai yang dapat atau tidak dapat dimodifikasi oleh pekerja. Jika Anda tidak menggunakan parameter ini, semua label, atribut kategori label, dan atribut bingkai akan disesuaikan.

  Untuk mempelajari lebih lanjut tentang`editsAllowed`parameter dan mengkonfigurasi file konfigurasi kategori label Anda, lihatSkema File Konfigurasi Kategori Label.
- Gunakan cloud titik 3D atau penyesuaian bingkai video lambda ARN untuk`PreHumanTaskLambdaArn`dan`AnnotationConsolidationLambdaArn`untuk penyesuaian dan verifikasi pelabelan pekerjaan:

- Untuk awan titik 3D, penyesuaian dan verifikasi pelabelan pekerjaan lambda fungsi ARN diakhiri dengan`Adjustment3DPointCloudSemanticSegmentation`,`Adjustment3DPointCloudObjectT` dan`Adjustment3DPointCloudObjectDetection`untuk segmentasi semantik awan titik 3D, deteksi objek, dan pelacakan objek masing-masing.
- Untuk bingkai video, penyesuaian dan verifikasi pelabelan pekerjaan lambda fungsi ARN diakhiri dengan`AdjustmentVideoObjectDetection`dan`AdjustmentVideoObjectTracking`untuk deteksi objek bingkai video dan pelacakan objek masing-masing.

Ground Truth menyimpan data keluaran dari verifikasi label atau pekerjaan penyesuaian di bucket S3 yang Anda tentukan di`S3OutputPath`parameter`CreateLabelingJob`operasi. Untuk informasi selengkapnya tentang data keluaran dari verifikasi label atau pekerjaan pelabelan penyesuaian, lihatLabel Verifikasi dan Penyesuaian Data dalam Manifes Output.

Label Verifikasi dan Penyesuaian Data dalam Manifes Output

Amazon SageMaker Ground Truth menulis data verifikasi label ke manifes keluaran dalam metadata untuk label. Ia menambahkan dua properti ke metadata:

- SEBUAH`type`properti, dengan nilai”`groundtruth/label-verification`.
- SEBUAH`worker-feedback`properti, dengan array`comment`nilai-nilai. Properti ini ditambahkan ketika pekerja memasukkan komentar. Jika tidak ada komentar, kolom tidak muncul.

Contoh manifes keluaran berikut menunjukkan bagaimana data verifikasi label muncul:

```
{
  "source-ref":"S3 bucket location",
  "verify-bounding-box":"1",
  "verify-bounding-box-metadata":
  {
    "class-name": "bad",
    "confidence": 0.93,
    "type": "groundtruth/label-verification",
    "job-name": "verify-bounding-boxes",
    "human-annotated": "yes",
    "creation-date": "2018-10-18T22:18:13.527256",
    "worker-feedback": [
      {"comment": "The bounding box on the bird is too wide on the right side."},
      {"comment": "The bird on the upper right is not labeled."}
    ]
```

```
  }
}
```

Output pekerja dari tugas penyesuaian menyerupai output pekerja dari tugas asli, kecuali bahwa itu berisi nilai-nilai yang disesuaikan dan`adjustment-status`properti dengan nilai`adjusted`atau`unadjusted`untuk menunjukkan apakah penyesuaian dilakukan.

Untuk contoh lebih lanjut dari output tugas yang berbeda, lihatData Output.

Perhatian dan Pertimbangan

Untuk mendapatkan perilaku yang diharapkan saat membuat verifikasi label atau pekerjaan penyesuaian, verifikasi data input Anda dengan cermat.

- Jika Anda menggunakan data gambar, verifikasi bahwa file manifes berisi informasi warna RGB heksadesimal.
- Untuk menghemat biaya pemrosesan, filter data Anda untuk memastikan Anda tidak memasukkan objek yang tidak diinginkan dalam manifes masukan pekerjaan pelabelan Anda.
- Tambahkan izin Amazon S3 yang diperlukan untuk memastikan data input Anda diproses dengan benar.

Saat Anda membuat pekerjaan pelabelan penyesuaian atau verifikasi menggunakan Ground Truth API, Andaharusmenggunakan yang berbeda`LabelAttributeName`dari pekerjaan pelabelan asli.

Persyaratan Informasi Warna untuk Pekerjaan Segmentasi Semantik

Untuk mereproduksi informasi warna dengan benar dalam tugas verifikasi atau penyesuaian, alat ini memerlukan informasi warna RGB heksadesimal dalam manifes (misalnya, #FFFFFF untuk putih). Saat Anda menyiapkan verifikasi Segmentasi Semantik atau pekerjaan penyesuaian, alat ini memeriksa manifes untuk menentukan apakah informasi ini ada. Jika tidak dapat menemukannya, Amazon SageMaker Ground Truth menampilkan pesan kesalahan dan setup pekerjaan berakhir.

Dalam iterasi sebelumnya dari alat Segmentasi Semantik, informasi warna kategori tidak dikeluarkan dalam format RGB heksadesimal ke manifes keluaran. Fitur itu diperkenalkan ke manifes keluaran pada saat yang sama alur kerja verifikasi dan penyesuaian diperkenalkan. Oleh karena itu, manifes keluaran yang lebih lama tidak kompatibel dengan alur kerja baru ini.

Filter Data Anda Sebelum Memulai Job

Amazon SageMaker Ground Truth memproses semua objek dalam manifes masukan Anda. Jika Anda memiliki kumpulan data berlabel sebagian, Anda mungkin ingin membuat manifes kustom menggunakanKueri Amazon S3 Selectpada manifes masukan Anda. Objek yang tidak berlabel gagal secara individual, tetapi tidak menyebabkan pekerjaan gagal, dan mungkin dikenakan biaya pemrosesan. Memfilter objek yang tidak ingin Anda verifikasi mengurangi biaya Anda.

Jika Anda membuat pekerjaan verifikasi menggunakan konsol, Anda dapat menggunakan alat penyaringan yang disediakan di sana. Jika Anda membuat pekerjaan menggunakan API, buat penyaringan data Anda sebagai bagian dari alur kerja Anda jika diperlukan.

Membuat Alur Kerja Pelabelan Kustom

Dokumen ini memandu Anda melalui proses pengaturan alur kerja dengan templat pelabelan khusus. Untuk mempelajari lebih lanjut tentang memulai pekerjaan pelabelan, lihatMulai. Di bagian itu, saat Anda memilih Jenis tugas, pilih Tugas pelabelan khusus, lalu ikuti petunjuk bagian ini untuk mengonfigurasinya.

Topik

- Langkah 1: Menyiapkan tenaga kerja Anda
- Langkah 2: Membuat template tugas pekerja kustom Anda
- Langkah 3: Memproses dengan AWS Lambda
- Demo Template: Anotasi Gambar dengan crowd-bounding-box
- Template Demo: Maksud Pelabelan dengan crowd-classifier
- Alur Kerja Kustom melalui API

Untuk informasi selengkapnya tentang membuat alur kerja pelabelan kustom, lihat Membuat alur kerja pelabelan data kustom dengan Amazon Ground SageMaker Truth.

Langkah 1: Menyiapkan tenaga kerja Anda

Pada langkah ini Anda menggunakan konsol untuk menetapkan jenis pekerja mana yang akan digunakan dan membuat sub-pilihan yang diperlukan untuk tipe pekerja. Ini mengasumsikan Anda telah menyelesaikan langkah-langkah hingga titik ini di Mulai bagian dan telah memilih tugas pelabelan kustom sebagai jenis Tugas.

Untuk mengkonfigurasi tenaga kerja Anda.

1. Pertama pilih opsi dari jenis Worker. Ada tiga jenis yang tersedia saat ini:

   - Publik menggunakan tenaga kerja sesuai permintaan kontraktor independen, yang didukung oleh Amazon Mechanical Turk. Mereka dibayar berdasarkan per tugas.
   - Private menggunakan karyawan atau kontraktor Anda untuk menangani data yang perlu tetap berada dalam organisasi Anda.
   - Vendor menggunakan vendor pihak ketiga yang mengkhususkan diri dalam menyediakan layanan pelabelan data, yang tersedia melalui Marketplace. AWS
2. Jika Anda memilih opsi Publik, Anda diminta untuk mengatur jumlah pekerja per objek kumpulan data. Memiliki lebih dari satu pekerja melakukan tugas yang sama pada objek yang sama dapat membantu meningkatkan akurasi hasil Anda. Defaultnya adalah tiga. Anda dapat menaikkan atau menurunkan itu tergantung pada akurasi yang Anda butuhkan.

   Anda juga diminta untuk menetapkan harga per tugas dengan menggunakan menu drop-down. Menu merekomendasikan poin harga berdasarkan berapa lama waktu yang dibutuhkan untuk menyelesaikan tugas.

   Metode yang disarankan untuk menentukan ini adalah dengan terlebih dahulu menjalankan tes singkat tugas Anda dengan tenaga kerja pribadi. Tes ini memberikan perkiraan realistis tentang berapa lama tugas yang dibutuhkan untuk diselesaikan. Anda kemudian dapat memilih rentang perkiraan yang termasuk dalam menu Harga per tugas. Jika waktu rata-rata Anda lebih dari 5 menit, pertimbangkan untuk memecah tugas Anda menjadi unit yang lebih kecil.

Selanjutnya

Langkah 2: Membuat template tugas pekerja kustom Anda

Langkah 2: Membuat template tugas pekerja kustom Anda

Template tugas pekerja adalah file yang digunakan oleh Ground Truth untuk menyesuaikan antarmuka pengguna pekerja (UI), atau UI tugas manusia. Anda dapat membuat template tugas pekerja menggunakan HTML, CSS,, bahasa template Liquid JavaScript, dan Crowd HTML Elements. Liquid digunakan untuk mengotomatiskan template, dan Crowd HTML Elements dapat digunakan untuk menyertakan alat anotasi umum dan memberikan logika untuk dikirimkan ke Ground Truth.

Gunakan topik berikut untuk mempelajari cara membuat templat tugas pekerja. Anda dapat melihat repositori contoh template tugas pekerja Ground Truth. GitHub

Topik

Dimulai dengan template dasar

Anda dapat menggunakan editor template di konsol Ground Truth untuk mulai membuat template. Editor ini mencakup sejumlah templat dasar yang telah dirancang sebelumnya dan fitur pengisian otomatis HTML dan Crowd HTML Element.

Untuk mengakses editor template kustom Ground Truth:

1. Mengikuti petunjuk di Membuat Job Pelabelan (Konsol) dan pilih Kustom untuk pekerjaan pelabelan Jenis tugas.
2. Saat Anda memilih Berikutnya, Anda akan dapat mengakses editor templat dan templat dasar di bagian Pengaturan tugas pelabelan khusus.
3. (Opsional) Pilih template dasar dari menu drop-down di bawah Template. Jika Anda lebih suka membuat template dari awal, pilih Custom dari menu drop-down untuk kerangka template minimal.

Mengembangkan template secara lokal

Meskipun Anda perlu berada di konsol untuk menguji bagaimana template Anda akan memproses data yang masuk, Anda dapat menguji tampilan dan nuansa HTML template Anda dan elemen kustom di browser Anda dengan menambahkan kode ini ke bagian atas file HTML Anda.

Example

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
```

Ini memuat kode yang diperlukan untuk membuat elemen HTML kustom. Gunakan ini jika Anda ingin mengembangkan tampilan dan nuansa template Anda di editor pilihan Anda daripada di konsol.

Ingat, ini tidak akan mengurai variabel Anda. Anda mungkin ingin menggantinya dengan konten sampel saat mengembangkan secara lokal.

Menggunakan Aset Eksternal

Templat kustom Amazon SageMaker Ground Truth memungkinkan skrip eksternal dan style sheet disematkan. Misalnya, blok kode berikut menunjukkan bagaimana Anda akan menambahkan style sheet yang terletak di `https://www.example.com/my-enhancement-styles.css` template Anda.

Example

```
<script src="https://www.example.com/my-enhancment-script.js"></script>
<link rel="stylesheet" type="text/css" href="https://www.example.com/my-enhancement-
styles.css">
```

Jika Anda menemukan kesalahan, pastikan bahwa server asal Anda mengirimkan jenis MIME yang benar dan encoding header dengan aset.

Misalnya, tipe MIME dan encoding untuk skrip jarak jauh adalah:. `application/javascript;CHARSET=UTF-8`

Jenis MIME dan encoding untuk stylesheet jarak jauh adalah:. `text/css;CHARSET=UTF-8`

Lacak variabel Anda

Dalam proses membangun sampel di bawah ini, akan ada langkah yang menambahkan variabel untuk mewakili potongan-potongan data yang dapat berubah dari tugas ke tugas, pekerja ke pekerja. Jika Anda memulai dengan salah satu contoh templat, Anda harus memastikan bahwa Anda mengetahui variabel yang sudah digunakannya. Saat Anda membuat skrip AWS Lambda pra-anotasi, outputnya harus berisi nilai untuk salah satu variabel yang Anda pilih untuk disimpan.

Nilai yang Anda gunakan untuk variabel dapat berasal dari file manifes Anda. Semua pasangan kunci-nilai dalam objek data Anda disediakan untuk Lambda pra-anotasi Anda. Jika ini adalah skrip pass-through sederhana, mencocokkan kunci untuk nilai dalam objek data Anda ke nama variabel dalam template Anda adalah cara termudah untuk meneruskan nilai-nilai tersebut ke formulir tugas yang dilihat pekerja Anda.

Sampel sederhana

Semua tugas dimulai dan diakhiri dengan `<crowd-form> </crowd-form>` elemen. Seperti `<form>` elemen HTML standar, semua kode formulir Anda harus berada di antara mereka.

Untuk tugas analisis tweet sederhana, gunakan elemen. `<crowd-classifier>` Hal ini membutuhkan atribut berikut:

- nama - nama variabel yang akan digunakan untuk hasil dalam bentuk output.
- kategori - array JSON diformat dari jawaban yang mungkin.
- header - judul untuk alat anotasi

Sebagai anak-anak `<crowd-classifier>` elemen, Anda harus memiliki tiga wilayah.

- **<**classification-target**>**- teks pekerja akan mengklasifikasikan berdasarkan opsi yang ditentukan dalam `categories` atribut di atas.
- **<**full-instructions**>**- instruksi yang tersedia dari tautan “Lihat instruksi lengkap” di alat. Ini dapat dibiarkan kosong, tetapi disarankan agar Anda memberikan instruksi yang baik untuk mendapatkan hasil yang lebih baik.
- **<**short-instructions**>**- deskripsi yang lebih singkat tentang tugas yang muncul di sidebar alat. Ini dapat dibiarkan kosong, tetapi disarankan agar Anda memberikan instruksi yang baik untuk mendapatkan hasil yang lebih baik.

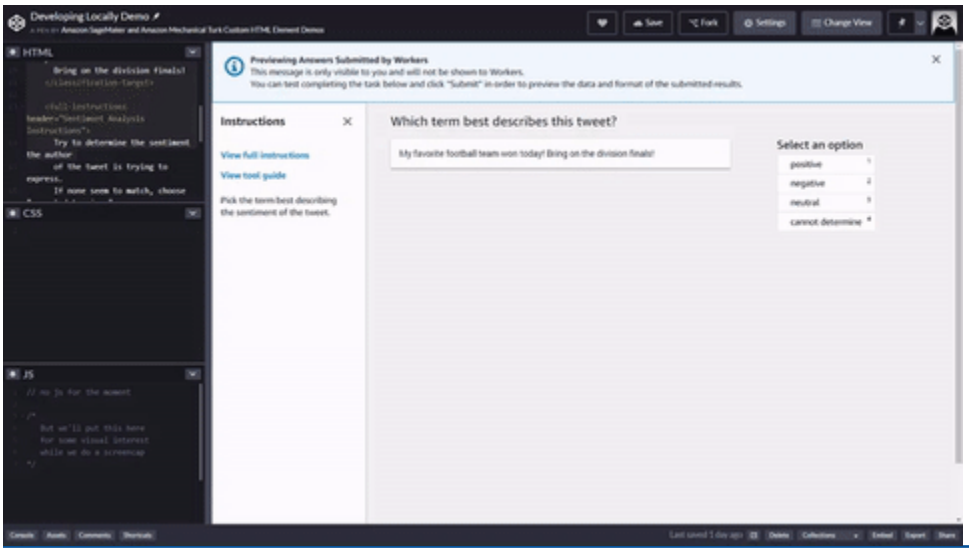
Versi sederhana dari alat ini akan terlihat seperti ini.

Example menggunakan **`crowd-classifier`**

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-classifier
    name="tweetFeeling"
    categories="['positive','negative','neutral', 'unclear']"
    header="Which term best describes this tweet?"
  >
     <classification-target>
      My favorite football team won today!
      Bring on the division finals!
    </classification-target>
```

```
    <full-instructions header="Sentiment Analysis Instructions">
      Try to determine the sentiment the author
      of the tweet is trying to express.
      If none seem to match, choose "cannot determine."
    </full-instructions>

    <short-instructions>
      Pick the term best describing the sentiment
      of the tweet.
    </short-instructions>

  </crowd-classifier>
</crowd-form>
```

Anda dapat menyalin dan menempelkan kode ke editor di alur kerja pembuatan pekerjaan pelabelan Ground Truth untuk melihat pratinjau alat, atau mencoba demo kode ini. CodePen



Menambahkan otomatisasi dengan Liquid

Sistem template kustom kami menggunakan Liquid untuk otomatisasi. Ini adalah bahasa markup inline open source. Dalam Liquid, teks antara kurung kurawal tunggal dan simbol persen adalah instruksi atau tag yang melakukan operasi seperti aliran kontrol atau iterasi. Teks antara kurung kurawal ganda adalah variabel atau objek yang mengeluarkan nilainya.

Penggunaan Liquid yang paling umum adalah mengurai data yang berasal dari Lambda pra-anotasi Anda dan mengeluarkan variabel yang relevan untuk membuat tugas. `taskInput`Objek yang dikembalikan oleh Anda Lambda pra-anotasi akan tersedia sebagai `task.input` objek di template Anda.

Properti dalam objek data manifes Anda diteruskan ke Anda Lambda pra-anotasi sebagai`event.dataObject`. Skrip pass-through sederhana hanya mengembalikan objek itu sebagai objek. `taskInput` Anda akan mewakili nilai dari manifes Anda sebagai variabel sebagai berikut.

Example Manifest objek data

```
{
  "source": "This is a sample text for classification",
  "labels": [ "angry" , "sad" , "happy" , "inconclusive" ],
  "header": "What emotion is the speaker feeling?"
}
```

Example Contoh HTML menggunakan variabel

```
<crowd-classifier
  name='tweetFeeling'
  categories='{{ task.input.labels | to_json }}'
  header='{{ task.input.header }}' >
<classification-target>
  {{ task.input.source }}
</classification-target>
```

Perhatikan penambahan " `| to_json`" ke `labels` properti di atas. Itu adalah filter untuk mengubah array menjadi representasi JSON dari array. Filter variabel dijelaskan di bagian selanjutnya.

Daftar berikut mencakup dua jenis tag Liquid yang mungkin berguna bagi Anda untuk mengotomatiskan pemrosesan data input template. Jika Anda memilih salah satu jenis tag berikut, Anda akan diarahkan ke dokumentasi Liquid.

- Aliran kontrol: Termasuk operator logika pemrograman seperti `if/elseunless`,, dan`case/when`.
- Iterasi: Memungkinkan Anda menjalankan blok kode berulang kali menggunakan pernyataan seperti untuk loop.

  Untuk contoh template HTML yang menggunakan elemen Liquid untuk membuat loop for, lihat translation-review-and-correction.liquid.html di. GitHub

Untuk informasi dan dokumentasi lebih lanjut, kunjungi beranda Liquid.

Filter variabel

Selain filter dan tindakan Liquid standar, Ground Truth menawarkan beberapa filter tambahan. Filter diterapkan dengan menempatkan karakter pipe (|) setelah nama variabel, kemudian menentukan nama filter. Filter dapat dirantai dalam bentuk:

Example

```
{{ <content> | <filter> | <filter> }}
```

Autoescape dan pelarian eksplisit

Secara default, input akan berupa HTML yang lolos untuk mencegah kebingungan antara teks variabel dan HTML Anda. Anda dapat secara eksplisit menambahkan `escape` filter untuk membuatnya lebih jelas bagi seseorang yang membaca sumber template Anda bahwa pelolosan sedang dilakukan.

escape_once

`escape_once`memastikan bahwa jika Anda telah lolos dari kode Anda, itu tidak akan lolos kembali di atas itu. Misalnya, agar & tidak menjadi &;.

skip_autoescape

`skip_autoescape`berguna ketika konten Anda dimaksudkan untuk digunakan sebagai HTML. Misalnya, Anda mungkin memiliki beberapa paragraf teks dan beberapa gambar dalam instruksi lengkap untuk kotak pembatas.

> Gunakan dengan **`skip_autoescape`** hemat
>
> Praktik terbaik dalam template adalah menghindari meneruskan kode fungsional atau markup `skip_autoescape` kecuali Anda benar-benar yakin Anda memiliki kontrol ketat atas apa yang sedang diteruskan. Jika Anda meneruskan masukan pengguna, Anda bisa membuka pekerja Anda hingga serangan Cross Site Scripting.

to_json

`to_json`akan menyandikan apa yang Anda berikan ke JSON (JavaScript Object Notation). Jika Anda memberinya makan sebuah objek, itu akan membuat serial itu.

grant_read_access

`grant_read_access`mengambil URI S3 dan mengkodekannya ke URL HTTPS dengan token akses berumur pendek untuk sumber daya itu. Hal ini memungkinkan untuk menampilkan objek foto, audio, atau video pekerja yang disimpan dalam ember S3 yang tidak dapat diakses publik.

Example dari filter

Input

```
auto-escape: {{ "Have you read 'James & the Giant Peach'?" }}
explicit escape: {{ "Have you read 'James & the Giant Peach'?" | escape }}
explicit escape_once: {{ "Have you read 'James &amp; the Giant Peach'?" |
 escape_once }}
skip_autoescape: {{ "Have you read 'James & the Giant Peach'?" | skip_autoescape }}
to_json: {{ jsObject | to_json }}
grant_read_access: {{ "s3://mybucket/myphoto.png" | grant_read_access }}
```

Example

Output

```
auto-escape: Have you read &#39;James &amp; the Giant Peach&#39;?
explicit escape: Have you read &#39;James &amp; the Giant Peach&#39;?
explicit escape_once: Have you read &#39;James &amp; the Giant Peach&#39;?
skip_autoescape: Have you read 'James & the Giant Peach'?
to_json: { "point_number": 8, "coords": [ 59, 76 ] }
grant_read_access: https://s3.amazonaws.com/mybucket/myphoto.png?<access token and
 other params>
```

Example dari template klasifikasi otomatis.

Untuk mengotomatiskan contoh klasifikasi teks sederhana, ganti teks tweet dengan variabel.

Template klasifikasi teks di bawah ini dengan otomatisasi ditambahkan. Perubahan/penambahan disorot dengan huruf tebal.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-classifier
    name="tweetFeeling"
    categories="['positive', 'negative', 'neutral', 'cannot determine']"
    header="Which term best describes this tweet?"
```

```
  >
    <classification-target>
       {{ task.input.source }}
    </classification-target>

    <full-instructions header="Analyzing a sentiment">
      Try to determine the feeling the author
      of the tweet is trying to express.
      If none seem to match, choose "other."
    </full-instructions>

    <short-instructions>
      Pick the term best describing the sentiment
      of the tweet.
    </short-instructions>

  </crowd-classifier>
</crowd-form>
```

Teks tweet yang ada di sampel sebelumnya sekarang diganti dengan objek. `entry.taskInput`Objek menggunakan `source` (atau nama lain yang Anda tentukan dalam Lambda pra-anotasi Anda) sebagai nama properti untuk teks dan dimasukkan langsung ke dalam HTML berdasarkan berada di antara tanda kurung kurawal ganda.

nd-to-end Demo E

Anda dapat melihat end-to-end demo berikut yang mencakup contoh fungsi Lambda:

- Demo Template: Anotasi Gambar dengan `crowd-bounding-box`
- Template Demo: Maksud Pelabelan dengan `crowd-classifier`

Langkah 3: Memproses dengan AWS Lambda

Pada langkah ini, Anda mempelajari cara membuat dan menentukan dua jenis fungsi AWSLambda yang diperlukan untuk membuat alur kerja pelabelan kustom:

- Lambda Pra-anotasi: Fungsi ini memulai dan melakukan pra-proses setiap objek data yang dikirim ke pekerjaan pelabelan Anda sebelum mengirimkannya ke pekerja.
- Lambda pasca-anotasi: Fungsi ini memproses hasil setelah pekerja mengirimkan tugas. Jika Anda menentukan beberapa pekerja per objek data, fungsi ini mungkin menyertakan logika untuk mengkonsolidasikan anotasi.

Jika Anda adalah pengguna baru Lambda dan Ground Truth, kami sarankan Anda menggunakan halaman di bagian ini sebagai berikut:

1. Pertama, tinjauPersyaratan Fungsi Lambda Pra-anotasi dan Pasca-anotasi.
2. Kemudian, gunakan halaman Izin yang Diperlukan Untuk Digunakan AWS Lambda Dengan Ground Truth untuk mempelajari tentang persyaratan keamanan dan izin untuk menggunakan fungsi Lambda pra-anotasi dan pasca-anotasi Anda dalam pekerjaan pelabelan kustom Ground Truth.
3. Selanjutnya, Anda perlu mengunjungi konsol Lambda atau menggunakan API Lambda untuk membuat fungsi Anda. Gunakan bagian ini Buat Fungsi Lambda untuk Alur Kerja Pelabelan Kustom untuk mempelajari cara membuat fungsi Lambda.
4. Untuk mempelajari cara menguji fungsi Lambda Anda, lihat. Uji Fungsi Lambda Pra-Anotasi dan Pasca-Anotasi
5. Setelah Anda membuat fungsi Lambda pra-pemrosesan dan pasca-pemrosesan, pilih fungsi Lambda dari bagian fungsi Lambda yang muncul setelah editor kode untuk HTML kustom Anda di konsol Ground Truth. Untuk mempelajari cara menggunakan fungsi ini dalam permintaan `CreateLabelingJob` API, lihatBuat Job Pelabelan (API).

Untuk tutorial alur kerja pelabelan khusus yang mencakup contoh fungsi Lambda pra-anotasi dan pasca-anotasi, dalam dokumen "". Demo Template: Anotasi Gambar dengan `crowd-bounding-box`

Topik

- Persyaratan Fungsi Lambda Pra-anotasi dan Pasca-anotasi
- Izin yang Diperlukan Untuk Digunakan AWS Lambda Dengan Ground Truth
- Buat Fungsi Lambda untuk Alur Kerja Pelabelan Kustom
- Uji Fungsi Lambda Pra-Anotasi dan Pasca-Anotasi

Persyaratan Fungsi Lambda Pra-anotasi dan Pasca-anotasi

Gunakan bagian ini untuk mempelajari sintaks permintaan yang dikirim ke fungsi Lambda pra-anotasi dan pasca-anotasi, dan sintaks respons yang dibutuhkan Ground Truth untuk menjalankan alur kerja pelabelan khusus.

Topik

- Lambda pra-anotasi

- Lambda pasca-anotasi

Lambda pra-anotasi

Sebelum tugas pelabelan dikirim ke pekerja, fungsi Lambda pra-anotasi Anda dipanggil.

Ground Truth mengirimkan fungsi Lambda Anda permintaan berformat JSON untuk memberikan detail tentang pekerjaan pelabelan dan objek data. Tabel berikut berisi skema permintaan pra-anotasi. Setiap parameter dijelaskan di bawah ini.

Data object identified with "source-ref"

```
{
    "version": "2018-10-16",
    "labelingJobArn": <labelingJobArn>
    "dataObject" : {
        "source-ref": <s3Uri>
    }
}
```

Data object identified with "source"

```
{
    "version": "2018-10-16",
    "labelingJobArn": <labelingJobArn>
    "dataObject" : {
        "source": <string>
    }
}
```

- `version`(string): Ini adalah nomor versi yang digunakan secara internal oleh Ground Truth.
- `labelingJobArn`(string): Ini adalah Nama Sumber Daya Amazon, atau ARN, dari pekerjaan pelabelan Anda. ARN ini dapat digunakan untuk mereferensikan pekerjaan pelabelan saat menggunakan operasi Ground Truth API seperti. `DescribeLabelingJob`
- The `dataObject` (objek JSON): Kunci berisi satu baris JSON, baik dari file manifes masukan Anda atau dikirim dari Amazon SNS. Objek garis JSON dalam manifes Anda dapat berukuran hingga 100 kilobyte dan berisi berbagai data. Untuk pekerjaan anotasi gambar yang sangat mendasar, `dataObject` JSON mungkin hanya berisi `source-ref` kunci, mengidentifikasi gambar yang akan dianotasi. Jika objek data (misalnya, baris teks) disertakan langsung dalam file manifes

masukan, objek data diidentifikasi dengan`source`. Jika Anda membuat pekerjaan verifikasi atau penyesuaian, baris ini mungkin berisi data label dan metadata dari pekerjaan pelabelan sebelumnya.

Tabel berikut mencakup contoh blok kode permintaan pra-anotasi. Setiap parameter dalam contoh permintaan ini dijelaskan di bawah tabel tab.

Data object identified with "source-ref"

```
{
    "version": "2018-10-16",
    "labelingJobArn": "arn:aws:sagemaker:<aws_region>:<aws_account_number>:labeling-job/<labeling_job_name>"
    "dataObject" : {
        "source-ref": "s3://<input-data-bucket>/<data-object-file-name>"
    }
}
```

Data object identified with "source"

```
{
    "version": "2018-10-16",
    "labelingJobArn": "arn:aws:sagemaker:<aws_region>:<aws_account_number>:labeling-job/<labeling_job_name>"
    "dataObject" : {
        "source": "Sue purchased 10 shares of the stock on April 10th, 2020"
    }
}
```

Sebagai imbalannya, Ground Truth membutuhkan respons yang diformat seperti berikut:

Example dari data pengembalian yang diharapkan

```
{
    "taskInput": <json object>,
    "isHumanAnnotationRequired": <boolean> # Optional
}
```

Pada contoh sebelumnya, `<json object>` kebutuhan untuk memuat semua data yang dibutuhkan template tugas pekerja kustom Anda. Jika Anda melakukan tugas kotak pembatas di mana instruksi

tetap sama sepanjang waktu, itu mungkin hanya sumber daya HTTP (S) atau Amazon S3 untuk file gambar Anda. Jika itu adalah tugas analisis sentimen dan objek yang berbeda mungkin memiliki pilihan yang berbeda, itu adalah referensi objek sebagai string dan pilihan sebagai array string.

> **Implikasi dari `isHumanAnnotationRequired`**
>
> Nilai ini opsional karena defaultnya. `true` Kasus penggunaan utama untuk secara eksplisit menyetelnya adalah ketika Anda ingin mengecualikan objek data ini agar tidak diberi label oleh pekerja manusia.

Jika Anda memiliki campuran objek dalam manifes Anda, dengan beberapa memerlukan anotasi manusia dan beberapa tidak membutuhkannya, Anda dapat menyertakan `isHumanAnnotationRequired` nilai di setiap objek data. Anda dapat menambahkan logika ke Lambda pra-anotasi Anda untuk menentukan secara dinamis apakah suatu objek memerlukan anotasi, dan menetapkan nilai boolean ini sesuai dengan itu.

Contoh Fungsi Lambda Pra-anotasi

Berikut ini, fungsi Lambda pra-anotasi dasar mengakses objek JSON `dataObject` dari permintaan awal, dan mengembalikannya dalam parameter. `taskInput`

```
import json

def lambda_handler(event, context):
    return {
        "taskInput":  event['dataObject']
    }
```

Dengan asumsi file manifes masukan digunakan `"source-ref"` untuk mengidentifikasi objek data, template tugas pekerja yang digunakan dalam pekerjaan pelabelan yang sama dengan Lambda pra-anotasi ini harus menyertakan elemen Liquid seperti berikut untuk dicerna: `dataObject`

```
{{ task.input.source-ref | grant_read_access }}
```

Jika file manifes masukan digunakan `source` untuk mengidentifikasi objek data, template tugas kerja dapat menelan `dataObject` dengan yang berikut:

```
{{ task.input.source }}
```

Contoh Lambda pra-anotasi berikut mencakup logika untuk mengidentifikasi kunci yang digunakan dalam`dataObject`, dan untuk menunjuk ke objek data yang menggunakan dalam pernyataan pengembalian `taskObject` Lambda.

```
import json

def lambda_handler(event, context):

    # Event received
    print("Received event: " + json.dumps(event, indent=2))

    # Get source if specified
    source = event['dataObject']['source'] if "source" in event['dataObject'] else None

    # Get source-ref if specified
    source_ref = event['dataObject']['source-ref'] if "source-ref" in
 event['dataObject'] else None

    # if source field present, take that otherwise take source-ref
    task_object = source if source is not None else source_ref

    # Build response object
    output = {
        "taskInput": {
            "taskObject": task_object
        },
        "humanAnnotationRequired": "true"
    }

    print(output)
    # If neither source nor source-ref specified, mark the annotation failed
    if task_object is None:
        print(" Failed to pre-process {} !".format(event["labelingJobArn"]))
        output["humanAnnotationRequired"] = "false"

    return output
```

Lambda pasca-anotasi

Ketika semua pekerja telah membuat anotasi objek data atau kapan [TaskAvailabilityLifetimeInSeconds](#)telah tercapai, mana yang lebih dulu, Ground Truth

mengirimkan anotasi tersebut ke Lambda pasca-anotasi Anda. Lambda ini umumnya digunakan untuk. Anotasi Terkonsolidasi

> **Tip**
>
> Untuk melihat contoh fungsi Lambda pasca-konsolidasi, lihat annotation_consolidation_lambda.py di repositori -recipe. aws-sagemaker-ground-truth GitHub

Blok kode berikut berisi skema permintaan pasca-anotasi. Setiap parameter dijelaskan dalam daftar berpoin berikut.

```
{
    "version": "2018-10-16",
    "labelingJobArn": <string>,
    "labelCategories": [<string>],
    "labelAttributeName": <string>,
    "roleArn" : <string>,
    "payload": {
        "s3Uri": <string>
    }
}
```

- `version`(string): Nomor versi yang digunakan secara internal oleh Ground Truth.
- `labelingJobArn`(string): Nama Sumber Daya Amazon, atau ARN, dari pekerjaan pelabelan Anda. ARN ini dapat digunakan untuk mereferensikan pekerjaan pelabelan saat menggunakan operasi Ground Truth API seperti. `DescribeLabelingJob`
- `labelCategories`(daftar string): Termasuk kategori label dan atribut lain yang Anda tentukan di konsol, atau yang Anda sertakan dalam file konfigurasi kategori label.
- `labelAttributeName`(string): Entah nama pekerjaan pelabelan Anda, atau nama atribut label yang Anda tentukan saat Anda membuat pekerjaan pelabelan.
- `roleArn`(string): Nama Sumber Daya Amazon (ARN) dari peran eksekusi IAM yang Anda tentukan saat Anda membuat pekerjaan pelabelan.
- `payload`(Objek JSON): JSON yang menyertakan `s3Uri` kunci, yang mengidentifikasi lokasi data anotasi untuk objek data tersebut di Amazon S3. Blok kode kedua di bawah ini menunjukkan contoh file anotasi ini.

Blok kode berikut berisi contoh permintaan pasca-anotasi. Setiap parameter dalam permintaan contoh ini dijelaskan di bawah blok kode.

Example dari permintaan Lambda pasca-anotasi

```
{
    "version": "2018-10-16",
    "labelingJobArn": "arn:aws:sagemaker:us-west-2:111122223333:labeling-job/labeling-job-name",
    "labelCategories": ["Ex Category1","Ex Category2", "Ex Category3"],
    "labelAttributeName": "labeling-job-attribute-name",
    "roleArn" : "arn:aws:iam::111122223333:role/role-name",
    "payload": {
        "s3Uri": "s3://DOC-EXAMPLE-BUCKET/annotations.json"
    }
 }
```

> **Note**
>
> Jika tidak ada pekerja yang bekerja pada objek data dan `TaskAvailabilityLifetimeInSeconds` telah tercapai, objek data ditandai sebagai gagal dan tidak disertakan sebagai bagian dari pemanggilan Lambda pasca-anotasi.

Blok kode berikut berisi skema payload. Ini adalah file yang ditunjukkan oleh `s3Uri` parameter di objek JSON permintaan Lambda pasca-anotasi. `payload` Misalnya, jika blok kode sebelumnya adalah permintaan Lambda pasca-anotasi, file anotasi berikut berada di. `s3://DOC-EXAMPLE-BUCKET/annotations.json`

Setiap parameter dijelaskan dalam daftar berpoin berikut.

Example dari file anotasi

```
[
    {
        "datasetObjectId": <string>,
        "dataObject": {
            "s3Uri": <string>,
            "content": <string>
        },
        "annotations": [{
```

```
            "workerId": <string>,
            "annotationData": {
                "content": <string>,
                "s3Uri": <string>
            }
        }]
    }
]
```

- `datasetObjectId`(string): Mengidentifikasi ID unik yang Ground Truth tetapkan ke setiap objek data yang Anda kirim ke pekerjaan pelabelan.
- `dataObject`(Objek JSON): Objek data yang diberi label. Jika objek data disertakan dalam file manifes masukan dan diidentifikasi menggunakan `source` kunci (misalnya, string), `dataObject` termasuk `content` kunci, yang mengidentifikasi objek data. Jika tidak, lokasi objek data (misalnya, tautan atau URI S3) diidentifikasi dengan`s3Uri`.
- `annotations`(daftar objek JSON): Daftar ini berisi objek JSON tunggal untuk setiap anotasi yang dikirimkan oleh pekerja untuk itu. `dataObject` Sebuah objek JSON tunggal berisi unik `workerId` yang dapat digunakan untuk mengidentifikasi pekerja yang mengirimkan anotasi itu. `annotationData`Kuncinya berisi salah satu dari yang berikut:
  - `content`(string): Berisi data anotasi.
  - `s3Uri`(string): Berisi URI S3 yang mengidentifikasi lokasi data anotasi.

Tabel berikut berisi contoh konten yang mungkin Anda temukan di payload untuk berbagai jenis anotasi.

Named Entity Recognition Payload

```
[
    {
      "datasetObjectId": "1",
      "dataObject": {
        "content": "Sift 3 cups of flour into the bowl."
      },
      "annotations": [
        {
          "workerId": "private.us-west-2.ef7294f850a3d9d1",
          "annotationData": {
            "content": "{\"crowd-entity-annotation\":{\"entities\":[{\"endOffset
\":4,\"label\":\"verb\",\"startOffset\":0},{\"endOffset\":6,\"label\":\"number
```

```
\",\"startOffset\":5},{\"endOffset\":20,\"label\":\"object\",\"startOffset\":15},
{\"endOffset\":34,\"label\":\"object\",\"startOffset\":30}]}}"
          }
        }
      ]
    }
]
```

Semantic Segmentation Payload

```
[
    {
      "datasetObjectId": "2",
      "dataObject": {
        "s3Uri": "s3://DOC-EXAMPLE-BUCKET/gt-input-data/images/bird3.jpg"
      },
      "annotations": [
        {
          "workerId": "private.us-west-2.ab1234c5678a919d0",
          "annotationData": {
            "content": "{\"crowd-semantic-segmentation\":{\"inputImageProperties\":
{\"height\":2000,\"width\":3020},\"labelMappings\":{\"Bird\":{\"color\":\"#2ca02c
\"}},\"labeledImage\":{\"pngImageData\":\"iVBOR...\"}}}"
          }
        }
      ]
    }
  ]
```

Bounding Box Payload

```
[
    {
      "datasetObjectId": "0",
      "dataObject": {
        "s3Uri": "s3://DOC-EXAMPLE-BUCKET/gt-input-data/images/bird1.jpg"
      },
      "annotations": [
        {
          "workerId": "private.us-west-2.ab1234c5678a919d0",
          "annotationData": {
```

```
            "content": "{\"boundingBox\":{\"boundingBoxes\":[{\"height\":2052,
\"label\":\"Bird\",\"left\":583,\"top\":302,\"width\":1375}],\"inputImageProperties
\":{\"height\":2497,\"width\":3745}}}"
          }
        }
      ]
    }
 ]
```

Fungsi Lambda pasca-anotasi Anda mungkin berisi logika yang mirip dengan yang berikut untuk diulang dan mengakses semua anotasi yang terdapat dalam permintaan. Untuk contoh lengkap, lihat annotation_consolidation_lambda.py di GitHub repositori aws-sagemaker-ground-truth-recipe. Dalam GitHub contoh ini, Anda harus menambahkan logika konsolidasi anotasi Anda sendiri.

```
for i in range(len(annotations)):
    worker_id = annotations[i]["workerId"]
    annotation_content = annotations[i]['annotationData'].get('content')
    annotation_s3_uri = annotations[i]['annotationData'].get('s3uri')
    annotation = annotation_content if annotation_s3_uri is None else
 s3_client.get_object_from_s3(
        annotation_s3_uri)
    annotation_from_single_worker = json.loads(annotation)

    print("{} Received Annotations from worker [{}] is [{}]"
            .format(log_prefix, worker_id, annotation_from_single_worker))
```

> **Tip**
>
> Ketika Anda menjalankan algoritma konsolidasi pada data, Anda dapat menggunakan layanan AWS database untuk menyimpan hasil, atau Anda dapat meneruskan hasil yang diproses kembali ke Ground Truth. Data yang Anda kembalikan ke Ground Truth disimpan dalam manifes anotasi terkonsolidasi dalam bucket S3 yang ditentukan untuk keluaran selama konfigurasi tugas pelabelan.

Sebagai imbalannya, Ground Truth membutuhkan respons yang diformat seperti berikut:

Example dari data pengembalian yang diharapkan

```
[
```

```
   {
        "datasetObjectId": <string>,
        "consolidatedAnnotation": {
            "content": {
                "<labelattributename>": {
                    # ... label content
                }
            }
        }
    },
   {
        "datasetObjectId": <string>,
        "consolidatedAnnotation": {
            "content": {
                "<labelattributename>": {
                    # ... label content
                }
            }
        }
    }
    .
    .
    .
]
```

Pada titik ini, semua data yang Anda kirim ke bucket S3 Anda, selain`datasetObjectId`, ada di `content` objek.

Saat Anda mengembalikan anotasi`content`, ini akan menghasilkan entri dalam manifes keluaran pekerjaan Anda seperti berikut:

Example dari format label dalam manifes keluaran

```
{  "source-ref"/"source" : "<s3uri or content>",
   "<labelAttributeName>": {
        # ... label content from you
    },
   "<labelAttributeName>-metadata": { # This will be added by Ground Truth
        "job_name": <labelingJobName>,
        "type": "groundTruth/custom",
        "human-annotated": "yes",
        "creation_date": <date> # Timestamp of when received from Post-labeling Lambda
    }
```

```
}
```

Karena sifat template kustom yang berpotensi kompleks dan data yang dikumpulkannya, Ground Truth tidak menawarkan pemrosesan data lebih lanjut.

Izin yang Diperlukan Untuk Digunakan AWS Lambda Dengan Ground Truth

Anda mungkin perlu mengonfigurasi beberapa atau semua hal berikut untuk membuat dan menggunakan AWS Lambda Ground Truth.

- Anda perlu memberikan peran IAM atau izin pengguna (secara kolektif, entitas IAM) untuk membuat fungsi Lambda pra-anotasi dan pasca-anotasi AWS Lambda menggunakan, dan memilihnya saat membuat pekerjaan pelabelan.
- Peran eksekusi IAM yang ditentukan saat pekerjaan pelabelan dikonfigurasi memerlukan izin untuk menjalankan fungsi Lambda pra-anotasi dan pasca-anotasi.
- Fungsi Lambda pasca-anotasi mungkin memerlukan izin untuk mengakses Amazon S3.

Gunakan bagian berikut untuk mempelajari cara membuat entitas IAM dan memberikan izin yang dijelaskan di atas.

Topik

- Berikan Izin untuk Membuat dan Memilih AWS Lambda Fungsi
- Berikan Izin Peran Eksekusi IAM untuk Memanggil Fungsi AWS Lambda
- Berikan Izin Lambda Pasca-Anotasi untuk Mengakses Anotasi

Berikan Izin untuk Membuat dan Memilih AWS Lambda Fungsi

Jika Anda tidak memerlukan izin terperinci untuk mengembangkan fungsi Lambda pra-anotasi dan pasca-anotasi, Anda dapat melampirkan kebijakan terkelola ke pengguna atau peran. AWS `AWSLambda_FullAccess` Kebijakan ini memberikan izin luas untuk menggunakan semua fitur Lambda, serta izin untuk melakukan tindakan di layanan lain yang berinteraksi AWS dengan Lambda.

Untuk membuat kebijakan yang lebih terperinci untuk kasus penggunaan yang sensitif terhadap keamanan, lihat dokumentasi Kebijakan IAM berbasis identitas untuk Lambda di Panduan Pengembang untuk mempelajari cara AWS Lambda membuat kebijakan IAM yang sesuai dengan kasus penggunaan Anda.

Kebijakan untuk Menggunakan Konsol Lambda

Jika Anda ingin memberikan izin entitas IAM untuk menggunakan konsol Lambda, lihat Menggunakan konsol Lambda di Panduan Pengembang. AWS Lambda

Selain itu, jika Anda ingin pengguna dapat mengakses dan menerapkan fungsi pra-anotasi dan pasca-anotasi starter Ground Truth menggunakan di AWS Serverless Application Repository konsol Lambda, Anda harus menentukan *<aws-region>*di mana Anda ingin menerapkan fungsi (ini harus AWS Wilayah yang sama yang digunakan untuk membuat pekerjaan pelabelan), dan menambahkan kebijakan berikut ke peran IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:ListApplicationVersions",
                "serverlessrepo:GetApplication",
                "serverlessrepo:CreateCloudFormationTemplate"
            ],
            "Resource": "arn:aws:serverlessrepo:<aws-region>:838997950401:applications/
aws-sagemaker-ground-truth-recipe"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "serverlessrepo:SearchApplications",
            "Resource": "*"
        }
    ]
}
```

Kebijakan untuk Melihat Fungsi Lambda di Ground Truth Console

Untuk memberikan izin entitas IAM untuk melihat fungsi Lambda di konsol Ground Truth saat pengguna membuat pekerjaan pelabelan khusus, entitas harus memiliki izin yang dijelaskan Berikan Izin IAM untuk Menggunakan Amazon SageMaker Ground Truth Console di dalamnya, termasuk izin yang dijelaskan di bagian. Izin Alur Kerja Pelabelan Kustom

Berikan Izin Peran Eksekusi IAM untuk Memanggil Fungsi AWS Lambda

Jika Anda menambahkan kebijakan terkelola IAM AmazonSageMakerGroundTruthExecutionke peran eksekusi IAM yang digunakan untuk membuat pekerjaan pelabelan, peran ini memiliki izin untuk mencantumkan dan memanggil fungsi Lambda dengan salah satu string berikut dalam nama fungsi:,,,, atau. `GtRecipe` `SageMaker` `Sagemaker` `sagemaker` `LabelingFunction`

Jika nama fungsi Lambda pra-anotasi atau pasca-anotasi tidak menyertakan salah satu istilah dalam paragraf sebelumnya, atau jika Anda memerlukan izin yang lebih terperinci daripada yang ada dalam kebijakan terkelola, Anda dapat menambahkan kebijakan yang serupa dengan berikut `AmazonSageMakerGroundTruthExecution` ini untuk memberikan izin peran eksekusi untuk menjalankan fungsi pra-anotasi dan pasca-anotasi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action":
                "lambda:InvokeFunction",
            "Resource": [
                "arn:aws:lambda:<region>:<account-id>:function:<pre-annotation-lambda-
name>",
                "arn:aws:lambda:<region>:<account-id>:function:<post-annotation-lambda-
name>"
            ]
        }
    ]
}
```

Berikan Izin Lambda Pasca-Anotasi untuk Mengakses Anotasi

Seperti dijelaskan dalamLambda pasca-anotasi, permintaan Lambda pasca-anotasi menyertakan lokasi data anotasi di Amazon S3. Lokasi ini diidentifikasi oleh `s3Uri` string dalam `payload` objek. Untuk memproses anotasi saat masuk, bahkan untuk fungsi pass through sederhana, Anda perlu menetapkan izin yang diperlukan untuk peran eksekusi Lambda pasca-anotasi untuk membaca file dari Amazon S3.

Ada banyak cara Anda dapat mengonfigurasi Lambda Anda untuk mengakses data anotasi di Amazon S3. Dua cara umum adalah:

- Izinkan peran eksekusi Lambda untuk mengambil peran eksekusi yang diidentifikasi `roleArn` dalam SageMaker permintaan Lambda pasca-anotasi. Peran SageMaker eksekusi ini adalah yang digunakan untuk membuat pekerjaan pelabelan, dan memiliki akses ke bucket keluaran Amazon S3 tempat data anotasi disimpan.
- Berikan izin peran eksekusi Lambda untuk mengakses bucket keluaran Amazon S3 secara langsung.

Gunakan bagian berikut untuk mempelajari cara mengonfigurasi opsi ini.

Berikan Izin Lambda untuk Mengambil Peran Eksekusi SageMaker

Untuk mengizinkan fungsi Lambda mengambil peran SageMaker eksekusi, Anda harus melampirkan kebijakan ke peran eksekusi fungsi Lambda, dan memodifikasi hubungan kepercayaan peran SageMaker eksekusi untuk memungkinkan Lambda mengasumsikan peran tersebut.

1. Lampirkan kebijakan IAM berikut ke peran eksekusi fungsi Lambda Anda untuk mengambil peran eksekusi SageMaker yang diidentifikasi. `Resource` Ganti *222222222222* dengan ID AWS akun. Ganti *sm-execution-role* dengan nama peran yang diasumsikan.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "arn:aws:iam::222222222222:role/sm-execution-role"
    }
}
```

2. Ubah kebijakan kepercayaan dari peran SageMaker eksekusi untuk memasukkan yang berikut ini`Statement`. Ganti *222222222222* dengan ID AWS akun. Ganti *my-lambda-execution-role* dengan nama peran yang diasumsikan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::222222222222:role/my-lambda-execution-role"
            },
```

```
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Berikan Izin Peran Eksekusi Lambda untuk Mengakses S3

Anda dapat menambahkan kebijakan yang mirip dengan berikut ini ke peran eksekusi fungsi Lambda pasca-anotasi untuk memberikan izin baca S3. Ganti *DOC-EXAMPLE-BUCKET* dengan nama bucket keluaran yang Anda tentukan saat membuat pekerjaan pelabelan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

Untuk menambahkan izin baca S3 ke peran eksekusi Lambda di konsol Lambda, gunakan prosedur berikut.

Tambahkan izin baca S3 ke Lambda pasca-anotasi:

1. Buka halaman Fungsi di konsol Lambda.
2. Pilih nama fungsi pasca-anotasi.
3. Pilih Konfigurasi, lalu pilih Izin.
4. Pilih nama Peran dan halaman ringkasan untuk peran itu terbuka di konsol IAM di tab baru.
5. Pilih Lampirkan kebijakan.
6. Lakukan salah satu dari cara berikut:

    - Cari dan pilih **AmazonS3ReadOnlyAccess**untuk memberikan izin fungsi untuk membaca semua ember dan objek di akun.

- Jika Anda memerlukan izin yang lebih terperinci, pilih Buat kebijakan dan gunakan contoh kebijakan di bagian sebelumnya untuk membuat kebijakan. Perhatikan bahwa Anda harus menavigasi kembali ke halaman ringkasan peran eksekusi setelah membuat kebijakan.

7. Jika Anda menggunakan kebijakan `AmazonS3ReadOnlyAccess` terkelola, pilih Lampirkan kebijakan.

   Jika Anda membuat kebijakan baru, navigasikan kembali ke halaman ringkasan peran eksekusi Lambda dan lampirkan kebijakan yang baru saja Anda buat.

Buat Fungsi Lambda untuk Alur Kerja Pelabelan Kustom

Anda dapat membuat fungsi Lambda menggunakan konsol Lambda, SDKAWS CLI, atau AWS SDK dalam bahasa pemrograman yang didukung pilihan Anda. Gunakan Panduan AWS Lambda Pengembang untuk mempelajari lebih lanjut tentang masing-masing opsi ini:

- Untuk mempelajari cara membuat fungsi Lambda menggunakan konsol, lihat Membuat fungsi Lambda dengan konsol.
- Untuk mempelajari cara membuat fungsi Lambda menggunakanAWS CLI, lihat Menggunakan AWSLambda dengan Antarmuka Baris AWS Perintah.
- Pilih bagian yang relevan dalam daftar isi untuk mempelajari lebih lanjut tentang bekerja dengan Lambda dalam bahasa pilihan Anda. Misalnya, pilih Bekerja dengan Python untuk mempelajari lebih lanjut tentang menggunakan Lambda dengan. AWS SDK for Python (Boto3)

Ground Truth menyediakan templat pra-anotasi dan pasca-anotasi melalui resep AWS Serverless Application Repository (SAR). Gunakan prosedur berikut untuk memilih resep Ground Truth di konsol Lambda.

Gunakan resep Ground Truth SAR untuk membuat fungsi Lambda pra-anotasi dan pasca-anotasi:

1. Buka halaman Fungsi di konsol Lambda.
2. Pilih Buat fungsi.
3. Pilih Jelajahi repositori aplikasi tanpa server.
4. Di kotak teks pencarian, masukkan aws-sagemaker-ground-truth-recipe dan pilih aplikasi itu.
5. Pilih Terapkan. Aplikasi ini mungkin membutuhkan waktu beberapa menit untuk digunakan.

Setelah aplikasi di-deploy, dua fungsi muncul di bagian Fungsi konsol `serverlessrepo-aws-sagema-GtRecipePreHumanTaskFunc-<id>` Lambda: dan. `serverlessrepo-aws-sagema-GtRecipeAnnotationConsol-<id>`

6. Pilih salah satu fungsi ini dan tambahkan logika kustom Anda di bagian Kode.
7. Setelah Anda selesai membuat perubahan, pilih Deploy untuk menerapkannya.

Uji Fungsi Lambda Pra-Anotasi dan Pasca-Anotasi

Anda dapat menguji pra-anotasi dan memposting anotasi fungsi Lambda di konsol Lambda. Jika Anda adalah pengguna baru Lambda, Anda dapat mempelajari cara menguji, atau memanggil, fungsi Lambda Anda di konsol menggunakan tutorial fungsi Buat Lambda dengan konsol di Panduan Pengembang. AWS Lambda

Anda dapat menggunakan bagian di halaman ini untuk mempelajari cara menguji templat pra-anotasi dan pasca-anotasi Ground Truth yang disediakan melalui (SARAWS Serverless Application Repository).

Topik

- Prasyarat
- Uji Fungsi Lambda Pra-anotasi
- Uji Fungsi Lambda Pasca-Anotasi

Prasyarat

Anda harus melakukan hal berikut untuk menggunakan tes yang dijelaskan di halaman ini.

- Anda memerlukan akses ke konsol Lambda, dan Anda memerlukan izin untuk membuat dan menjalankan fungsi Lambda. Untuk mempelajari cara mengatur izin ini, lihatBerikan Izin untuk Membuat dan Memilih AWS Lambda Fungsi.
- Jika Anda belum menerapkan resep Ground Truth SAR, gunakan prosedur Buat Fungsi Lambda untuk Alur Kerja Pelabelan Kustom untuk melakukannya.
- Untuk menguji fungsi Lambda pasca-anotasi, Anda harus memiliki file data di Amazon S3 dengan data anotasi sampel. Untuk tes sederhana, Anda dapat menyalin dan menempelkan kode berikut ke dalam file dan menyimpannya sebagai `sample-annotations.json` dan mengunggah file ini ke Amazon S3. Perhatikan URI S3 dari file ini—Anda memerlukan informasi ini untuk mengonfigurasi pengujian Lambda pasca-anotasi.

```
[{"datasetObjectId":"0","dataObject":{"content":"To train a machine learning model,
 you need a large, high-quality, labeled dataset. Ground Truth helps you build
 high-quality training datasets for your machine learning models."},"annotations":
[{"workerId":"private.us-west-2.0123456789","annotationData":{"content":"{\"crowd-
entity-annotation\":{\"entities\":[{\"endOffset\":8,\"label\":\"verb\",\"startOffset
\":3},{\"endOffset\":27,\"label\":\"adjective\",\"startOffset\":11},{\"endOffset
\":33,\"label\":\"object\",\"startOffset\":28},{\"endOffset\":51,\"label\":
\"adjective\",\"startOffset\":46},{\"endOffset\":65,\"label\":\"adjective\",
\"startOffset\":53},{\"endOffset\":74,\"label\":\"adjective\",\"startOffset\":67},
{\"endOffset\":82,\"label\":\"adjective\",\"startOffset\":75},{\"endOffset\":102,
\"label\":\"verb\",\"startOffset\":97},{\"endOffset\":112,\"label\":\"verb\",
\"startOffset\":107},{\"endOffset\":125,\"label\":\"adjective\",\"startOffset
\":113},{\"endOffset\":134,\"label\":\"adjective\",\"startOffset\":126},{\"endOffset
\":143,\"label\":\"object\",\"startOffset\":135},{\"endOffset\":169,\"label
\":\"adjective\",\"startOffset\":153},{\"endOffset\":176,\"label\":\"object\",
\"startOffset\":170}]}}"}}]},{"datasetObjectId":"1","dataObject":{"content":"Sift
 3 cups of flour into the bowl."},"annotations":[{"workerId":"private.us-
west-2.0123456789","annotationData":{"content":"{\"crowd-entity-annotation\":
{\"entities\":[{\"endOffset\":4,\"label\":\"verb\",\"startOffset\":0},{\"endOffset
\":6,\"label\":\"number\",\"startOffset\":5},{\"endOffset\":20,\"label\":\"object
\",\"startOffset\":15},{\"endOffset\":34,\"label\":\"object\",\"startOffset
\":30}]}}"}}]},{"datasetObjectId":"2","dataObject":{"content":"Jen purchased 10
 shares of the stock on Janurary 1st, 2020."},"annotations":[{"workerId":"private.us-
west-2.0123456789","annotationData":{"content":"{\"crowd-entity-annotation
\":{\"entities\":[{\"endOffset\":3,\"label\":\"person\",\"startOffset\":0},
{\"endOffset\":13,\"label\":\"verb\",\"startOffset\":4},{\"endOffset\":16,\"label
\":\"number\",\"startOffset\":14},{\"endOffset\":58,\"label\":\"date\",\"startOffset
\":40}]}}"}}]},{"datasetObjectId":"3","dataObject":{"content":"The narrative
 was interesting, however the character development was weak."},"annotations":
[{"workerId":"private.us-west-2.0123456789","annotationData":{"content":"{\"crowd-
entity-annotation\":{\"entities\":[{\"endOffset\":29,\"label\":\"adjective\",
\"startOffset\":18},{\"endOffset\":73,\"label\":\"adjective\",\"startOffset
\":69}]}}"}}]}]
```


- Anda harus menggunakan petunjuk Berikan Izin Lambda Pasca-Anotasi untuk Mengakses Anotasi untuk memberikan izin peran eksekusi fungsi Lambda pasca-anotasi Anda untuk mengambil peran eksekusi yang SageMaker Anda gunakan untuk membuat pekerjaan pelabelan. Fungsi Lambda pasca-anotasi menggunakan peran SageMaker eksekusi untuk mengakses file data anotasi,, di S3. `sample-annotations.json`

Uji Fungsi Lambda Pra-anotasi

Gunakan prosedur berikut untuk menguji fungsi Lambda pra-anotasi yang dibuat saat Anda menerapkan resep Ground AWS Serverless Application Repository Truth (SAR).

Uji resep Ground Truth SAR pra-anotasi fungsi Lambda

1. Buka halaman Fungsi di konsol Lambda.
2. Pilih fungsi pra-anotasi yang digunakan dari resep Ground Truth SAR. Nama fungsi ini mirip dengan`serverlessrepo-aws-sagema-GtRecipePreHumanTaskFunc-<id>`.
3. Di bagian Sumber kode, pilih panah di sebelah Uji.
4. Pilih Konfigurasikan acara pengujian.
5. Tetap pilih opsi Create new test event.
6. Di bawah template Event, pilih SageMakerGround Truth PreHumanTask.
7. Berikan tes Anda nama Acara.
8. Pilih Buat.
9. Pilih panah di sebelah Uji lagi dan Anda akan melihat bahwa tes yang Anda buat dipilih, yang ditunjukkan dengan titik dengan nama acara. Jika tidak dipilih, pilih.
10. Pilih Test untuk menjalankan tes.

Setelah Anda menjalankan tes, Anda dapat melihat hasil Eksekusi. Dalam log Fungsi, Anda akan melihat respons yang mirip dengan yang berikut ini:

```
START RequestId: cd117d38-8365-4e1a-bffb-0dcd631a878f Version: $LATEST
Received event: {
  "version": "2018-10-16",
  "labelingJobArn": "arn:aws:sagemaker:us-east-2:123456789012:labeling-job/example-
job",
  "dataObject": {
    "source-ref": "s3://sagemakerexample/object_to_annotate.jpg"
  }
}
{'taskInput': {'taskObject': 's3://sagemakerexample/object_to_annotate.jpg'},
 'isHumanAnnotationRequired': 'true'}
END RequestId: cd117d38-8365-4e1a-bffb-0dcd631a878f
REPORT RequestId: cd117d38-8365-4e1a-bffb-0dcd631a878f Duration: 0.42 ms Billed
 Duration: 1 ms Memory Size: 128 MB Max Memory Used: 43 MB
```

Dalam respons ini, kita dapat melihat output fungsi Lambda cocok dengan sintaks respons pra-anotasi yang diperlukan:

```
{'taskInput': {'taskObject': 's3://sagemakerexample/object_to_annotate.jpg'},
 'isHumanAnnotationRequired': 'true'}
```

Uji Fungsi Lambda Pasca-Anotasi

Gunakan prosedur berikut untuk menguji fungsi Lambda pasca-anotasi yang dibuat saat Anda menerapkan resep Ground AWS Serverless Application Repository Truth (SAR).

Uji resep Ground Truth SAR pasca-anotasi Lambda

1. Buka halaman Fungsi di konsol Lambda.
2. Pilih fungsi pasca-anotasi yang digunakan dari resep Ground Truth SAR. Nama fungsi ini mirip dengan`serverlessrepo-aws-sagema-GtRecipeAnnotationConsol-<id>`.
3. Di bagian Sumber kode, pilih panah di sebelah Uji.
4. Pilih Konfigurasikan acara pengujian.
5. Tetap pilih opsi Create new test event.
6. Di bawah template Event, pilih SageMakerGround Truth AnnotationConsolidation.
7. Berikan tes Anda nama Acara.
8. Ubah kode template yang disediakan sebagai berikut:

   - Ganti Nama Sumber Daya Amazon (ARN) `roleArn` dengan ARN dari peran SageMaker eksekusi yang Anda gunakan untuk membuat pekerjaan pelabelan.
   - Ganti URI S3 `s3Uri` dengan URI `sample-annotations.json` file yang Anda tambahkan ke Amazon S3.

   Setelah Anda melakukan modifikasi ini, tes Anda akan terlihat mirip dengan yang berikut:

   ```
   {
     "version": "2018-10-16",
     "labelingJobArn": "arn:aws:sagemaker:us-east-2:123456789012:labeling-job/example-job",
     "labelAttributeName": "example-attribute",
     "roleArn": "arn:aws:iam::222222222222:role/sm-execution-role",
     "payload": {
   ```

```
    "s3Uri": "s3://your-bucket/sample-annotations.json"
  }
}
```

9. Pilih Buat.
10. Pilih panah di sebelah Uji lagi dan Anda akan melihat bahwa tes yang Anda buat dipilih, yang ditunjukkan dengan titik dengan nama acara. Jika tidak dipilih, pilih.
11. Pilih Test untuk menjalankan tes.

Setelah Anda menjalankan pengujian, Anda akan melihat `-- Consolidated Output --` bagian di Function Logs, yang berisi daftar semua anotasi yang disertakan di `sample-annotations.json` dalamnya.

Demo Template: Anotasi Gambar dengan **`crowd-bounding-box`**

Ketika Anda memilih untuk menggunakan template kustom sebagai jenis tugas Anda di konsol Amazon SageMaker Ground Truth, Anda mencapai panel tugas pelabelan kustom. Di sana Anda dapat memilih dari beberapa templat dasar. Template mewakili beberapa tugas yang paling umum dan memberikan contoh untuk bekerja saat Anda membuat template tugas pelabelan khusus Anda. Jika Anda tidak menggunakan konsol, atau sebagai bantuan tambahan, lihat Amazon SageMaker Ground Truth Sample Task UI untuk repositori template demo untuk berbagai jenis tugas pekerjaan pelabelan.

Demonstrasi ini bekerja dengan BoundingBoxtemplate. Demonstrasi juga berfungsi dengan AWS Lambda fungsi yang diperlukan untuk memproses data Anda sebelum dan sesudah tugas. Di repositori Github di atas, untuk menemukan template yang berfungsi dengan AWS Lambda fungsi, cari `{{ task.input.<property name> }}` di template.

Topik

- Templat kustom Starter Bounding Box
- Template kustom Bounding Box Anda sendiri
- File manifes Anda
- Fungsi Lambda pra-anotasi Anda
- Fungsi Lambda pasca-anotasi Anda
- Output dari pekerjaan pelabelan Anda

Templat kustom Starter Bounding Box

Ini adalah template kotak pembatas starter yang disediakan.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-bounding-box
    name="boundingBox"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="{{ task.input.header }}"
    labels="{{ task.input.labels | to_json | escape }}"
  >

    <!-- The <full-instructions> tag is where you will define the full instructions of
 your task. -->
    <full-instructions header="Bounding Box Instructions" >
      <p>Use the bounding box tool to draw boxes around the requested target of
 interest:</p>
      <ol>
        <li>Draw a rectangle using your mouse over each instance of the target.</li>
        <li>Make sure the box does not cut into the target, leave a 2 - 3 pixel
 margin</li>
        <li>
          When targets are overlapping, draw a box around each object,
          include all contiguous parts of the target in the box.
          Do not include parts that are completely overlapped by another object.
        </li>
        <li>
          Do not include parts of the target that cannot be seen,
          even though you think you can interpolate the whole shape of the target.
        </li>
        <li>Avoid shadows, they're not considered as a part of the target.</li>
        <li>If the target goes off the screen, label up to the edge of the image.</li>
      </ol>
    </full-instructions>

    <!-- The <short-instructions> tag allows you to specify instructions that are
 displayed in the left hand side of the task interface.
    It is a best practice to provide good and bad examples in this section for quick
 reference. -->
    <short-instructions>
      Use the bounding box tool to draw boxes around the requested target of interest.
    </short-instructions>
```

```
  </crowd-bounding-box>
</crowd-form>
```

Template kustom menggunakan bahasa template Liquid, dan setiap item di antara kurung kurawal ganda adalah variabel. AWS LambdaFungsi pra-anotasi harus menyediakan objek bernama `taskInput` dan properti objek tersebut dapat diakses seperti `{{ task.input.<property name> }}` pada template Anda.

Template kustom Bounding Box Anda sendiri

Sebagai contoh, asumsikan Anda memiliki banyak koleksi foto hewan di mana Anda mengetahui jenis hewan dalam gambar dari pekerjaan klasifikasi gambar sebelumnya. Sekarang Anda ingin memiliki kotak pembatas yang digambar di sekitarnya.

Dalam sampel starter, ada tiga variabel:`taskObject`,`header`, dan`labels`.

Masing-masing akan diwakili di berbagai bagian kotak pembatas.

- `taskObject`adalah URL HTTP (S) atau URI S3 untuk foto yang akan dianotasi. Yang ditambahkan `| grant_read_access` adalah filter yang akan mengonversi URI S3 ke URL HTTPS dengan akses singkat ke sumber daya itu. Jika Anda menggunakan URL HTTP (S), itu tidak diperlukan.
- `header`adalah teks di atas foto yang akan diberi label, seperti “Gambar kotak di sekitar burung di foto.”
- `labels`adalah array, direpresentasikan sebagai`['item1', 'item2', ...]`. Ini adalah label yang dapat ditetapkan oleh pekerja ke kotak berbeda yang mereka gambar. Anda dapat memiliki satu atau banyak.

Setiap nama variabel berasal dari objek JSON dalam respons dari Lambda pra-anotasi Anda, Nama-nama di atas hanya disarankan, Gunakan nama variabel apa pun yang masuk akal bagi Anda dan akan mempromosikan keterbacaan kode di antara tim Anda.

> **Gunakan variabel hanya bila diperlukan**
>
> Jika bidang tidak akan berubah, Anda dapat menghapus variabel itu dari template dan menggantinya dengan teks itu, jika tidak, Anda harus mengulangi teks itu sebagai nilai di setiap objek dalam manifes Anda atau mengkodekannya ke dalam fungsi Lambda pra-anotasi Anda.

Example : Templat Kotak Pembatas Akhir yang Disesuaikan

Untuk menjaga hal-hal sederhana, template ini akan memiliki satu variabel, satu label, dan instruksi yang sangat dasar. Dengan asumsi manifes Anda memiliki properti “hewan” di setiap objek data, nilai itu dapat digunakan kembali di dua bagian templat.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-bounding-box
    name="boundingBox"
    labels="[ '{{ task.input.animal }}' ]"
    src="{{ task.input.source-ref | grant_read_access }}"
    header="Draw a box around the {{ task.input.animal }}."
  >
    <full-instructions header="Bounding Box Instructions" >
      <p>Draw a bounding box around the {{ task.input.animal }} in the image. If
      there is more than one {{ task.input.animal }} per image, draw a bounding
      box around the largest one.</p>
      <p>The box should be tight around the {{ task.input.animal }} with
      no more than a couple of pixels of buffer around the
      edges.</p>
      <p>If the image does not contain a {{ task.input.animal }}, check the <strong>
      Nothing to label</strong> box.
    </full-instructions>
    <short-instructions>
      <p>Draw a bounding box around the {{ task.input.animal }} in each image. If
      there is more than one {{ task.input.animal }} per image, draw a bounding
      box around the largest one.</p>
    </short-instructions>
  </crowd-bounding-box>
</crowd-form>
```

Perhatikan penggunaan kembali `{{ task.input.animal }}` seluruh template. Jika manifes Anda memiliki semua nama hewan yang dimulai dengan huruf kapital, Anda dapat menggunakan`{{ task.input.animal | downcase }}`, memasukkan salah satu filter bawaan Liquid dalam kalimat yang perlu disajikan huruf kecil.

File manifes Anda

File manifes Anda harus memberikan nilai variabel yang Anda gunakan dalam template Anda. Anda dapat melakukan beberapa transformasi data manifes Anda di Lambda pra-anotasi Anda, tetapi jika

Anda tidak perlu, Anda mempertahankan risiko kesalahan yang lebih rendah dan Lambda Anda akan berjalan lebih cepat. Berikut adalah contoh file manifes untuk template.

```
{"source-ref": "<S3 image URI>", "animal": "horse"}
{"source-ref": "<S3 image URI>", "animal" : "bird"}
{"source-ref": "<S3 image URI>", "animal" : "dog"}
{"source-ref": "<S3 image URI>", "animal" : "cat"}
```

Fungsi Lambda pra-anotasi Anda

Sebagai bagian dari pengaturan pekerjaan, berikan ARN fungsi AWS Lambda yang dapat dipanggil untuk memproses entri manifes Anda dan meneruskannya ke mesin template.

> **Menamai fungsi Lambda Anda**
>
> Praktik terbaik dalam penamaan fungsi Anda adalah dengan menggunakan salah satu dari empat string berikut sebagai bagian dari nama fungsi:`SageMaker`,, `Sagemakersagemaker`, atau`LabelingFunction`. Ini berlaku untuk fungsi pra-anotasi dan pasca-anotasi Anda.

Saat Anda menggunakan konsol, jika Anda memiliki fungsi AWS Lambda yang dimiliki oleh akun Anda, daftar drop-down fungsi yang memenuhi persyaratan penamaan akan disediakan untuk memilih salah satu.

Dalam contoh yang sangat mendasar ini, Anda hanya melewati informasi dari manifes tanpa melakukan pemrosesan tambahan apa pun di atasnya. Contoh fungsi pra-anotasi ini ditulis untuk Python 3.7.

```
import json

def lambda_handler(event, context):
    return {
        "taskInput": event['dataObject']
    }
```

Objek JSON dari manifes Anda akan disediakan sebagai anak dari `event` objek. Properti di dalam `taskInput` objek akan tersedia sebagai variabel untuk template Anda, jadi cukup mengatur nilai `taskInput` to `event['dataObject']` akan meneruskan semua nilai dari objek manifes Anda ke template Anda tanpa harus menyalinnya satu per satu. Jika Anda ingin mengirim lebih banyak nilai ke template, Anda dapat menambahkannya ke `taskInput` objek.

Fungsi Lambda pasca-anotasi Anda

Sebagai bagian dari pengaturan pekerjaan, berikan ARN fungsi AWS Lambda yang dapat dipanggil untuk memproses data formulir ketika seorang pekerja menyelesaikan tugas. Ini bisa sesederhana atau serumit yang Anda inginkan. Jika Anda ingin melakukan konsolidasi jawaban dan penilaian saat masuk, Anda dapat menerapkan algoritma penilaian dan/atau konsolidasi pilihan Anda. Jika Anda ingin menyimpan data mentah untuk pemrosesan offline, itu adalah opsi.

> **Berikan izin ke Lambda pasca-anotasi Anda**
>
> Data anotasi akan berada dalam file yang ditunjuk oleh `s3Uri` string di `payload` objek. Untuk memproses anotasi saat masuk, bahkan untuk fungsi pass through sederhana, Anda perlu menetapkan `S3ReadOnly` akses ke Lambda Anda sehingga dapat membaca file anotasi.
>
> Di halaman Konsol untuk membuat Lambda Anda, gulir ke panel peran Eksekusi. Pilih Buat peran baru dari satu atau beberapa templat. Beri nama peran itu. Dari drop-down Templat kebijakan, pilih izin hanya-baca objek Amazon S3. Simpan Lambda dan peran akan disimpan dan dipilih.

Contoh berikut adalah dalam Python 2.7.

```
import json
import boto3
from urlparse import urlparse

def lambda_handler(event, context):
    consolidated_labels = []

    parsed_url = urlparse(event['payload']['s3Uri']);
    s3 = boto3.client('s3')
    textFile = s3.get_object(Bucket = parsed_url.netloc, Key = parsed_url.path[1:])
    filecont = textFile['Body'].read()
    annotations = json.loads(filecont);

    for dataset in annotations:
        for annotation in dataset['annotations']:
            new_annotation = json.loads(annotation['annotationData']['content'])
            label = {
                'datasetObjectId': dataset['datasetObjectId'],
                'consolidatedAnnotation' : {
```

```
                'content': {
                    event['labelAttributeName']: {
                        'workerId': annotation['workerId'],
                        'boxesInfo': new_annotation,
                        'imageSource': dataset['dataObject']
                        }
                    }
                }
            }
            consolidated_labels.append(label)

    return consolidated_labels
```

Lambda pasca-anotasi akan sering menerima kumpulan hasil tugas di objek acara. Batch itu akan menjadi `payload` objek yang harus diulang oleh Lambda. Apa yang Anda kirim kembali akan menjadi objek yang memenuhi kontrak API.

Output dari pekerjaan pelabelan Anda

Anda akan menemukan output pekerjaan di folder yang dinamai setelah pekerjaan pelabelan Anda di bucket S3 target yang Anda tentukan. Ini akan berada di subfolder bernama`manifests`.

Untuk tugas kotak pembatas, output yang Anda temukan di manifes keluaran akan terlihat sedikit seperti demo di bawah ini. Contoh telah dibersihkan untuk dicetak. Output aktual akan menjadi satu baris per catatan.

Example : JSON dalam manifes keluaran Anda

```
{
  "source-ref":"<URL>",
  "<label attribute name>":
    {
       "workerId":"<URL>",
       "imageSource":"<image URL>",
       "boxesInfo":"{\"boundingBox\":{\"boundingBoxes\":[{\"height\":878, \"label\":
\"bird\", \"left\":208, \"top\":6, \"width\":809}], \"inputImageProperties\":{\"height
\":924, \"width\":1280}}}"},
  "<label attribute name>-metadata":
    {
      "type":"groundTruth/custom",
      "job_name":"<Labeling job name>",
      "human-annotated":"yes"
    },
```

```
  "animal" : "bird"
}
```

Perhatikan bagaimana `animal` atribut tambahan dari manifes asli Anda diteruskan ke manifes keluaran pada tingkat yang sama dengan data `source-ref` dan pelabelan. Properti apa pun dari manifes masukan Anda, apakah itu digunakan dalam template Anda atau tidak, akan diteruskan ke manifes keluaran.

Template Demo: Maksud Pelabelan dengan `crowd-classifier`

Jika Anda memilih templat khusus, Anda akan mencapai panel tugas pelabelan khusus. Di sana Anda dapat memilih dari beberapa templat pemula yang mewakili beberapa tugas yang lebih umum. Template menyediakan titik awal untuk bekerja dalam membangun template tugas pelabelan khusus Anda.

Dalam demonstrasi ini, Anda bekerja dengan template Deteksi Maksud, yang menggunakan `klasifikasi kerumunan` elemen, dan AWS Lambda fungsi yang diperlukan untuk memproses data Anda sebelum dan sesudah tugas.

Topik

- Templat kustom Deteksi Maksud Pemula
- Templat kustom Deteksi Maksud Anda
- Fungsi Lambda pra-anotasi Anda
- Fungsi Lambda pasca-anotasi Anda
- Output pekerjaan pelabelan Anda

Templat kustom Deteksi Maksud Pemula

Ini adalah template deteksi maksud yang disediakan sebagai titik awal.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-classifier
    name="intent"
    categories="{{ task.input.labels | to_json | escape }}"
    header="Pick the most relevant intention expressed by the below text"
  >
    <classification-target>
```

```
      {{ task.input.utterance }}
    </classification-target>

    <full-instructions header="Intent Detection Instructions">
        <p>Select the most relevant intention expressed by the text.</p>
        <div>
           <p><strong>Example: </strong>I would like to return a pair of shoes</p>
           <p><strong>Intent: </strong>Return</p>
        </div>
    </full-instructions>

    <short-instructions>
      Pick the most relevant intention expressed by the text
    </short-instructions>
  </crowd-classifier>
</crowd-form>
```

Template kustom menggunakan bahasa template Liquid, dan setiap item di antara kurung kurawal ganda adalah variabel. Fungsi AWS Lambda pra-anotasi harus menyediakan objek bernama `taskInput` dan properti objek tersebut dapat diakses seperti `{{ task.input.<property name> }}` pada templat Anda.

Templat kustom Deteksi Maksud Anda

Dalam template starter, ada dua variabel: `task.input.labels` properti di tag pembuka `crowd-classifier` elemen dan konten `task.input.utterance` di `classification-target` wilayah.

Kecuali Anda perlu menawarkan set label yang berbeda dengan ucapan yang berbeda, menghindari variabel dan hanya menggunakan teks akan menghemat waktu pemrosesan dan menciptakan lebih sedikit kemungkinan kesalahan. Template yang digunakan dalam demonstrasi ini akan menghapus variabel itu, tetapi variabel dan filter seperti dijelaskan `to_json` secara lebih rinci dalam artikel `crowd-bounding-box`demonstrasi.

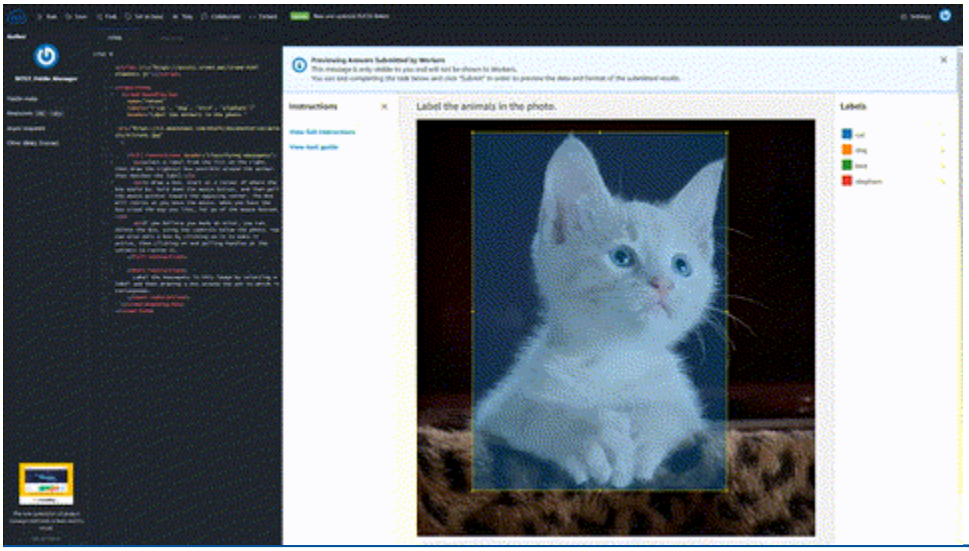
Styling Elemen Anda

Dua bagian dari elemen kustom yang terkadang diabaikan adalah `<short-instructions>` wilayah `<full-instructions>` dan. Instruksi yang baik menghasilkan hasil yang baik.

Dalam elemen yang menyertakan wilayah ini, `<short-instructions>` muncul secara otomatis di panel “Instruksi” di sebelah kiri layar pekerja. `<full-instructions>`Ditautkan dari tautan “Lihat instruksi lengkap” di dekat bagian atas panel itu. Mengklik tautan akan membuka panel modal dengan instruksi yang lebih rinci.

Anda tidak hanya dapat menggunakan HTML, CSS, dan JavaScript di bagian ini, Anda dianjurkan untuk jika Anda yakin Anda dapat memberikan serangkaian instruksi dan contoh yang kuat yang akan membantu pekerja menyelesaikan tugas Anda dengan kecepatan dan akurasi yang lebih baik.

Example Cobalah sampel dengan JSFiddle



Cobalah contoh `<crowd-classifier>` tugas. Contoh diberikan oleh JSFiddle, oleh karena itu semua variabel template diganti dengan nilai hard-code. Klik tautan “Lihat instruksi lengkap” untuk melihat serangkaian contoh dengan gaya CSS yang diperluas. Anda dapat melakukan fork proyek untuk bereksperimen dengan perubahan Anda sendiri pada CSS, menambahkan gambar sampel, atau menambahkan JavaScript fungsionalitas yang diperluas.

Example : Templat Deteksi Maksud Akhir yang Disesuaikan

Ini menggunakan `<crowd-classifier>`tugas contoh, tetapi dengan variabel untuk`<classification-target>`. Jika Anda mencoba untuk menjaga desain CSS yang konsisten di antara serangkaian pekerjaan pelabelan yang berbeda, Anda dapat menyertakan stylesheet eksternal menggunakan `<link rel...>` elemen dengan cara yang sama seperti yang Anda lakukan dalam dokumen HTML lainnya.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-classifier
    name="intent"
    categories="['buy', 'eat', 'watch', 'browse', 'leave']"
    header="Pick the most relevant intent expressed by the text below"
```

```
  >
    <classification-target>
      {{ task.input.source }}
    </classification-target>

    <full-instructions header="Emotion Classification Instructions">
      <p>In the statements and questions provided in this exercise, what category of
 action is the speaker interested in doing?</p>
          <table>
            <tr>
              <th>Example Utterance</th>
              <th>Good Choice</th>
            </tr>
            <tr>
              <td>When is the Seahawks game on?</td>
              <td>
                eat<br>
                <greenbg>watch</greenbg>
                <botchoice>browse</botchoice>
              </td>
            </tr>
            <tr>
              <th>Example Utterance</th>
              <th>Bad Choice</th>
            </tr>
            <tr>
              <td>When is the Seahawks game on?</td>
              <td>
                buy<br>
                <greenbg>eat</greenbg>
                <botchoice>watch</botchoice>
              </td>
            </tr>
          </table>
    </full-instructions>

    <short-instructions>
      What is the speaker expressing they would like to do next?
    </short-instructions>
  </crowd-classifier>
</crowd-form>
<style>
  greenbg {
    background: #feee23;
```

```
  display: block;
}

table {
  *border-collapse: collapse; /* IE7 and lower */
  border-spacing: 0;
}

th, tfoot, .fakehead {
  background-color: #8888ee;
  color: #f3f3f3;
  font-weight: 700;
}

th, td, tfoot {
    border: 1px solid blue;
}

th:first-child {
  border-radius: 6px 0 0 0;
}

th:last-child {
  border-radius: 0 6px 0 0;
}

th:only-child{
  border-radius: 6px 6px 0 0;
}

tfoot:first-child {
  border-radius: 0 0 6px 0;
}

tfoot:last-child {
  border-radius: 0 0 0 6px;
}

tfoot:only-child{
  border-radius: 6px 6px;
}

td {
  padding-left: 15px ;
```

```
    padding-right: 15px ;
  }

  botchoice {
    display: block;
    height: 17px;
    width: 490px;
    overflow: hidden;
    position: relative;
    background: #fff;
    padding-bottom: 20px;
  }

  botchoice:after {
    position: absolute;
    bottom: 0;
    left: 0;
    height: 100%;
    width: 100%;
    content: "";
    background: linear-gradient(to top,
       rgba(255,255,255, 1) 55%,
       rgba(255,255,255, 0) 100%
    );
    pointer-events: none; /* so the text is still selectable */
  }
</style>
```

Example : File manifes Anda

Jika Anda menyiapkan file manifes secara manual untuk tugas klasifikasi teks seperti ini, minta data Anda diformat dengan cara berikut.

```
{"source": "Roses are red"}
{"source": "Violets are Blue"}
{"source": "Ground Truth is the best"}
{"source": "And so are you"}
```

Ini berbeda dari file manifes yang digunakan untuk demonstrasi Demo Template: Anotasi Gambar dengan `crowd-bounding-box` "" `source-ref` yang digunakan sebagai nama properti, bukan. `source` Penggunaan `source-ref` menunjuk S3 URI untuk gambar atau file lain yang harus dikonversi ke HTTP. Jika tidak, `source` harus digunakan seperti itu dengan string teks di atas.

Fungsi Lambda pra-anotasi Anda

Sebagai bagian dari pengaturan pekerjaan, berikan ARN dari AWS Lambda sebuah yang dapat dipanggil untuk memproses entri manifes Anda dan meneruskannya ke mesin template.

Fungsi Lambda ini diperlukan untuk memiliki salah satu dari empat string berikut sebagai bagian dari nama fungsi:`SageMaker`,,`Sagemaker`, `sagemaker` atau. `LabelingFunction`

Ini berlaku untuk Lambdas pra-anotasi dan pasca-anotasi Anda.

Saat Anda menggunakan konsol, jika Anda memiliki Lambda yang dimiliki oleh akun Anda, daftar drop-down fungsi yang memenuhi persyaratan penamaan akan disediakan untuk memilih salah satu.

Dalam contoh yang sangat mendasar ini, di mana Anda hanya memiliki satu variabel, ini terutama fungsi pass-through. Berikut adalah contoh pra-pelabelan Lambda menggunakan Python 3.7.

```
import json

def lambda_handler(event, context):
    return {
        "taskInput":  event['dataObject']
    }
```

`dataObject`Properti `event` berisi properti dari objek data dalam manifes Anda.

Dalam demonstrasi ini, yang merupakan lintasan sederhana, Anda hanya meneruskannya langsung sebagai `taskInput` nilainya. Jika Anda menambahkan properti dengan nilai-nilai tersebut ke `event['dataObject']` objek, mereka akan tersedia untuk template HTML Anda sebagai variabel Liquid dengan format`{{ task.input.<property name> }}`.

Fungsi Lambda pasca-anotasi Anda

Sebagai bagian dari pengaturan pekerjaan, berikan ARN dari fungsi Lambda yang dapat dipanggil untuk memproses data formulir ketika seorang pekerja menyelesaikan tugas. Ini bisa sesederhana atau serumit yang Anda inginkan. Jika Anda ingin melakukan konsolidasi jawaban dan penilaian saat data masuk, Anda dapat menerapkan algoritma penilaian atau konsolidasi pilihan Anda. Jika Anda ingin menyimpan data mentah untuk pemrosesan offline, itu adalah opsi.

> **Tetapkan izin untuk fungsi Lambda pasca-anotasi Anda**
>
> Data anotasi akan berada dalam file yang ditunjuk oleh `s3Uri` string di `payload` objek. Untuk memproses anotasi saat masuk, bahkan untuk fungsi pass through sederhana, Anda

perlu menetapkan `S3ReadOnly` akses ke Lambda Anda sehingga dapat membaca file anotasi.

Di halaman Konsol untuk membuat Lambda Anda, gulir ke panel peran Eksekusi. Pilih Buat peran baru dari satu atau beberapa templat. Beri nama peran itu. Dari drop-down Templat kebijakan, pilih izin hanya-baca objek Amazon S3. Simpan Lambda dan peran akan disimpan dan dipilih.

Contoh berikut adalah untuk Python 3.7.

```
import json
import boto3
from urllib.parse import urlparse

def lambda_handler(event, context):
    consolidated_labels = []

    parsed_url = urlparse(event['payload']['s3Uri']);
    s3 = boto3.client('s3')
    textFile = s3.get_object(Bucket = parsed_url.netloc, Key = parsed_url.path[1:])
    filecont = textFile['Body'].read()
    annotations = json.loads(filecont);

    for dataset in annotations:
        for annotation in dataset['annotations']:
            new_annotation = json.loads(annotation['annotationData']['content'])
            label = {
                'datasetObjectId': dataset['datasetObjectId'],
                'consolidatedAnnotation' : {
                'content': {
                    event['labelAttributeName']: {
                        'workerId': annotation['workerId'],
                        'result': new_annotation,
                        'labeledContent': dataset['dataObject']
                        }
                    }
                }
            }
            consolidated_labels.append(label)

    return consolidated_labels
```

Output pekerjaan pelabelan Anda

Lambda pasca-anotasi akan sering menerima kumpulan hasil tugas di objek acara. Batch itu akan menjadi `payload` objek yang harus diulang oleh Lambda.

Anda akan menemukan output pekerjaan di folder yang dinamai setelah pekerjaan pelabelan Anda di bucket S3 target yang Anda tentukan. Ini akan berada di subfolder bernama`manifests`.

Untuk tugas deteksi maksud, output dalam manifes keluaran akan terlihat sedikit seperti demo di bawah ini. Contohnya telah dibersihkan dan diberi jarak agar lebih mudah dibaca manusia. Output aktual akan lebih terkompresi untuk pembacaan mesin.

Example : JSON dalam manifes keluaran Anda

```
[
  {
    "datasetObjectId":"<Number representing item's place in the manifest>",
     "consolidatedAnnotation":
     {
       "content":
       {
         "<name of labeling job>":
         {
           "workerId":"private.us-east-1.XXXXXXXXXXXXXXXXXXXXX",
           "result":
           {
             "intent":
             {
                 "label":"<label chosen by worker>"
             }
           },
           "labeledContent":
           {
             "content":"<text content that was labeled>"
           }
         }
       }
     }
   },
  "datasetObjectId":"<Number representing item's place in the manifest>",
     "consolidatedAnnotation":
     {
       "content":
```

```
        {
          "<name of labeling job>":
          {
            "workerId":"private.us-east-1.6UDLPKQZHYWJQSCA4MBJBB7FWE",
            "result":
            {
              "intent":
              {
                  "label": "<label chosen by worker>"
              }
            },
            "labeledContent":
            {
              "content": "<text content that was labeled>"
            }
          }
        }
      },
        ...
        ...
        ...
]
```

Ini akan membantu Anda membuat dan menggunakan template kustom Anda sendiri.

Alur Kerja Kustom melalui API

Ketika Anda telah membuat template UI kustom Anda (Langkah 2) dan memproses fungsi Lambda (Langkah 3), Anda harus menempatkan template di bucket Amazon S3 dengan format nama file:. `<FileName>.liquid.html`

Gunakan `CreateLabelingJob`tindakan untuk mengonfigurasi tugas Anda. Anda akan menggunakan lokasi template kustom (Langkah 2: Membuat template tugas pekerja kustom Anda) yang disimpan dalam *`<filename>`*`.liquid.html` file di S3 sebagai nilai untuk `UiTemplateS3Uri` bidang dalam `UiConfig`objek di dalam `HumanTaskConfig`objek.

Untuk tugas AWS Lambda yang dijelaskan dalamLangkah 3: Memproses dengan AWS Lambda, ARN tugas pasca-anotasi akan digunakan sebagai nilai untuk `AnnotationConsolidationLambdaArn` bidang, dan tugas pra-anotasi akan digunakan sebagai nilai untuk `PreHumanTaskLambdaArn`.

Buat Job Pelabelan

Anda dapat membuat pekerjaan pelabelan di SageMaker konsol Amazon dan dengan menggunakan AWS SDK dalam bahasa pilihan Anda untuk dijalankan. `CreateLabelingJob` Setelah pekerjaan pelabelan dibuat, Anda dapat melacak metrik pekerja (untuk tenaga kerja pribadi) dan status pekerjaan pelabelan Anda menggunakan. CloudWatch

Sebelum Anda membuat pekerjaan pelabelan, disarankan agar Anda meninjau halaman berikut, sebagaimana berlaku:

- Anda dapat menentukan data input menggunakan penyiapan data otomatis di konsol, atau file manifes masukan di konsol atau saat menggunakan `CreateLabelingJob` API. Untuk penyiapan data otomatis, lihatPengaturan Data Otomatis. Untuk mempelajari cara membuat file manifes masukan, lihatMenggunakan File Manifes Masukan.
- Tinjau kuota data input pekerjaan pelabelan:. Kuota

Setelah Anda memilih jenis tugas Anda, gunakan topik di halaman ini untuk mempelajari cara membuat pekerjaan pelabelan.

Jika Anda adalah pengguna Ground Truth baru, kami sarankan Anda mulai dengan menelusuri demo diMulai.

> **Important**
>
> Ground Truth mengharuskan semua bucket S3 yang berisi data gambar input pekerjaan pelabelan agar kebijakan CORS dilampirkan. Untuk mempelajari informasi lebih lanjut, lihat Persyaratan Izin.

Topik

- Jenis Tugas Bawaan
- Membuat Halaman Instruksi
- Membuat Job Pelabelan (Konsol)
- Buat Job Pelabelan (API)
- Buat Pekerjaan Pelabelan Streaming
- Buat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai

Jenis Tugas Bawaan

Amazon SageMaker Ground Truth memiliki beberapa tipe tugas bawaan. Ground Truth menyediakan template tugas pekerja untuk tipe tugas bawaan. Selain itu, beberapa tipe tugas bawaan mendukungPelabelan Data. Topik berikut menjelaskan setiap jenis tugas bawaan dan demo template tugas pekerja yang disediakan oleh Ground Truth di konsol. Untuk mempelajari cara membuat pekerjaan pelabelan di konsol menggunakan salah satu jenis tugas ini, pilih halaman jenis tugas.

Gambar Label	Teks Label	Label Video dan Bingkai Video	Label Awan Titik 3D
• Kotak pembatas • Klasifikasi Gambar (Label Tunggal) • Klasifikasi Gambar (Multi-label) • Segmentasi Semantic • Verifikasi dan Sesuaikan Label	• Pengenalan Entitas bernama • Klasifikasi Teks (Label Tunggal) • Klasifikasi Teks (Multi-label)	• Klasifikasi video • Deteksi Objek Bingkai Video • Pelacakan Objek Bingkai Video	• Deteksi Objek Awan Titik 3D • Pelacakan Objek Awan Titik 3D • Segmentasi Semantik Awan Titik 3D

> **Note**
>
> Setiap frame video dan tipe tugas cloud titik 3D memiliki jenis tugas penyesuaian yang Anda gunakan untuk memverifikasi dan menyesuaikan label dari pekerjaan pelabelan sebelumnya. Pilih bingkai video atau halaman jenis tugas cloud titik 3D di atas untuk mempelajari cara menyesuaikan label yang dibuat menggunakan jenis tugas tersebut.

Membuat Halaman Instruksi

Buat instruksi khusus untuk pelabelan pekerjaan untuk meningkatkan akurasi pekerja Anda dalam menyelesaikan tugas mereka. Anda dapat mengubah instruksi default yang disediakan di konsol atau Anda dapat membuat milik Anda sendiri. Instruksi ditunjukkan kepada pekerja di halaman tempat mereka menyelesaikan tugas pelabelan mereka.

Ada dua jenis instruksi:

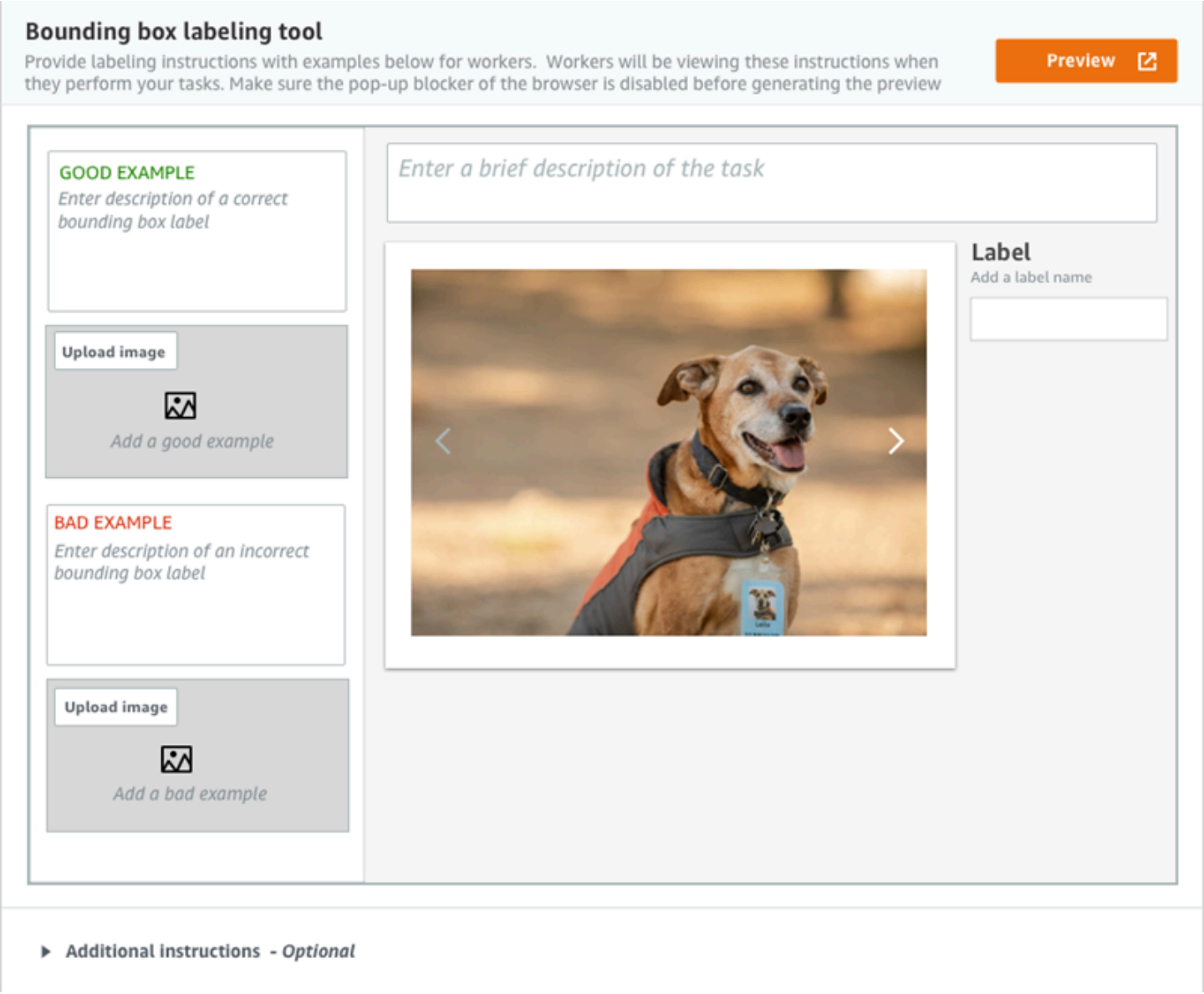
- Instruksi singkat—instructions yang ditampilkan pada halaman web yang sama di mana pekerja menyelesaikan tugas mereka. Instruksi ini harus memberikan referensi yang mudah untuk menunjukkan kepada pekerja cara yang benar untuk memberi label pada suatu objek.
- Instruksi lengkap—instructions yang ditampilkan pada kotak dialog yang melapisi halaman tempat pekerja menyelesaikan tugas mereka. Kami menyarankan Anda memberikan instruksi terperinci untuk menyelesaikan tugas dengan beberapa contoh yang menunjukkan kasus tepi dan situasi sulit lainnya untuk memberi label objek.

Buat instruksi di konsol saat Anda membuat pekerjaan pelabelan Anda. Mulailah dengan instruksi yang ada untuk tugas dan gunakan editor untuk memodifikasinya agar sesuai dengan pekerjaan pelabelan Anda.

> **Note**
>
> Setelah Anda membuat pekerjaan pelabelan Anda, itu akan secara otomatis dimulai dan Anda tidak akan dapat memodifikasi instruksi pekerja Anda. Jika Anda perlu mengubah instruksi pekerja, hentikan pekerjaan pelabelan yang Anda buat, kloning, dan ubah instruksi pekerja Anda sebelum membuat pekerjaan baru.
> Anda dapat mengkloning pekerjaan pelabelan di konsol dengan memilih pekerjaan pelabelan dan kemudian memilihclonedi dalamTindakanMenu menu.
> Untuk mengkloning pekerjaan pelabelan menggunakan Amazon SageMaker API atau Amazon pilihan Anda SageMaker SDK, buat permintaan baru ke`CreateLabelingJob`operasi dengan spesifikasi yang sama dengan pekerjaan awal Anda setelah memodifikasi instruksi pekerja Anda.

Instruksi singkat

Instruksi singkat muncul di halaman web yang sama yang digunakan pekerja untuk memberi label pada objek data Anda. Misalnya, berikut ini adalah halaman pengeditan untuk tugas kotak pembatas. Panel instruksi singkat ada di sebelah kiri.



Perlu diingat bahwa seorang pekerja hanya akan menghabiskan beberapa detik untuk melihat instruksi singkatnya. Pekerja harus dapat memindai dan memahami informasi Anda dengan cepat. Dalam semua kasus, perlu sedikit waktu untuk memahami instruksi daripada yang diperlukan untuk menyelesaikan tugas. Ingatlah hal-hal ini:

- Instruksi Anda harus jelas dan sederhana.
- Gambar lebih baik daripada kata-kata. Buat ilustrasi sederhana tentang tugas Anda yang dapat segera dipahami oleh pekerja Anda.
- Jika Anda harus menggunakan kata-kata, gunakan contoh singkat dan ringkas.
- Instruksi singkat Anda lebih penting daripada instruksi lengkap Anda.

Amazon SageMaker Ground Truth konsol menyediakan editor sehingga Anda dapat membuat petunjuk singkat Anda. Ganti teks placeholder dan gambar dengan instruksi untuk tugas Anda. Pratinjau halaman tugas pekerja dengan memilihPratinjau. Pratinjau akan terbuka di jendela baru, pastikan untuk mematikan pemblokiran pop-up sehingga jendela akan ditampilkan.

Instruksi lengkap

Anda dapat memberikan instruksi tambahan untuk pekerja Anda di kotak dialog yang melapisi halaman tempat pekerja memberi label pada objek data Anda. Gunakan instruksi lengkap untuk menjelaskan tugas yang lebih kompleks dan untuk menunjukkan kepada pekerja cara yang tepat untuk memberi label pada kasus tepi atau benda sulit lainnya.

Anda dapat membuat instruksi lengkap menggunakan editor di konsol Ground Truth. Seperti instruksi cepat, ingatlah hal-hal berikut:

- Pekerja akan ingin instruksi rinci beberapa kali pertama bahwa menyelesaikan tugas Anda. Informasi apa pun yang merekaharusharus dalam petunjuk cepat.
- Gambar lebih penting daripada kata-kata.
- Teks harus ringkas.
- Instruksi lengkap harus melengkapi instruksi singkat. Jangan ulangi informasi yang muncul dalam instruksi singkat.

Konsol Ground Truth menyediakan editor sehingga Anda dapat membuat instruksi lengkap. Ganti teks placeholder dan gambar dengan instruksi untuk tugas Anda. Pratinjau halaman instruksi lengkap dengan memilihPratinjau. Pratinjau akan terbuka di jendela baru, pastikan untuk mematikan pemblokiran pop-up sehingga jendela akan ditampilkan.

Tambahkan contoh gambar ke instruksi Anda

Gambar memberikan contoh yang berguna bagi pekerja Anda. Untuk menambahkan gambar yang dapat diakses publik ke instruksi Anda:

- Tempatkan kursor di mana gambar harus pergi di editor instruksi.
- Klik ikon gambar di bilah alat editor.
- Masukkan URL gambar Anda.

Jika gambar instruksi Anda di Amazon S3 tidak dapat diakses publik:

- Sebagai URL gambar, masukkan:`{{ 'https://s3.amazonaws.com/your-bucket-name/image-file-name' | grant_read_access }}`.
- Ini membuat URL gambar dengan kode akses satu kali berumur pendek yang ditambahkan sehingga browser pekerja dapat menampilkannya. Ikon gambar yang rusak ditampilkan di editor instruksi, tetapi melihat pratinjau alat menampilkan gambar di pratinjau yang diberikan.

Membuat Job Pelabelan (Konsol)

Anda dapat menggunakan Amazon SageMaker konsol untuk membuat pekerjaan pelabelan untuk semua jenis tugas bawaan Ground Truth dan alur kerja pelabelan khusus. Untuk tipe tugas bawaan, kami menyarankan Anda menggunakan halaman inihalaman untuk jenis tugas Anda. Setiap halaman jenis tugas menyertakan detail spesifik tentang membuat pekerjaan pelabelan menggunakan jenis tugas tersebut.

Anda perlu memberikan yang berikut ini untuk membuat pekerjaan pelabelan di SageMaker konsol:

- File manifes masukan di Amazon S3. Anda dapat menempatkan kumpulan data input di Amazon S3 dan secara otomatis membuat file manifes menggunakan konsol Ground Truth (tidak didukung untuk pekerjaan pelabelan cloud titik 3D).

  Atau, Anda dapat secara manual membuat file manifes masukan. Untuk mempelajari caranya, lihat Input Data.
- Bucket Amazon S3 untuk menyimpan data keluaran Anda.
- Peran IAM dengan izin untuk mengakses sumber daya Anda di Amazon S3 dan dengan SageMakerkebijakan eksekusi terlampir. Untuk solusi umum, Anda dapat melampirkan kebijakan terkelola, AmazonSageMakerFullAccess, untuk peran IAM dan termasuk`sagemaker`di nama ember Anda.

  Untuk kebijakan yang lebih terperinci, lihatthe section called “Izin IAM”.

  Jenis tugas cloud titik 3D memiliki pertimbangan keamanan tambahan. Pelajari selengkapnya.
- Sebuah tim kerja. Anda membuat tim kerja dari tenaga kerja yang terdiri dari pekerja Amazon Mechanical Turk, vendor, atau pekerja pribadi Anda sendiri.Untuk lebih bersandar, lihatMembuat dan Mengelola Tenaga Kerja.

  Anda tidak dapat menggunakan tenaga kerja Mechanical Turk untuk pekerjaan pelabelan cloud titik 3D atau bingkai video.

- Jika Anda menggunakan alur kerja pelabelan kustom, Anda harus menyimpan template tugas pekerja di Amazon S3 dan menyediakan URI Amazon S3 untuk template tersebut. Untuk informasi selengkapnya, lihat Langkah 2: Membuat template tugas pekerja kustom Anda.
- (Opsional) SebuahAWS KMSkunci ARN jika Anda mau SageMaker untuk mengenkripsi output dari pekerjaan pelabelan Anda menggunakan milik Anda sendiriAWS KMSkunci enkripsi alih-alih kunci layanan Amazon S3 default.
- (Opsional) Label yang ada untuk kumpulan data yang Anda gunakan untuk pekerjaan pelabelan Anda. Gunakan opsi ini jika Anda ingin pekerja menyesuaikan, atau menyetujui dan menolak label.
- Jika Anda ingin membuat pekerjaan pelabelan penyesuaian atau verifikasi, Anda harus memiliki file manifes keluaran di Amazon S3 yang berisi label yang ingin disesuaikan atau diverifikasi. Opsi ini hanya didukung untuk pekerjaan pelabelan gambar segmentasi kotak dan segmentasi semantik serta pekerjaan pelabelan cloud titik 3D dan bingkai video. Disarankan agar Anda menggunakan instruksiVerifikasi dan Sesuaikan Labeluntuk membuat pekerjaan pelabelan verifikasi atau penyesuaian.

> **Important**
>
> Tim kerja Anda, file manifes masukan, bucket keluaran, dan sumber daya lainnya di Amazon S3 harus samaAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda.

Saat Anda membuat pekerjaan pelabelan menggunakan SageMaker konsol, Anda menambahkan instruksi dan label pekerja ke UI pekerja yang disediakan Ground Truth. Anda dapat melihat pratinjau dan berinteraksi dengan UI pekerja sambil membuat pekerjaan pelabelan di konsol. Anda juga dapat melihat pratinjau UI pekerja dihalaman tipe tugas.

Untuk membuat pekerjaan pelabelan (konsol)

1. Sign in SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilihPekerjaan pelabelan.
3. PadaPekerjaan pelabelanhalaman, pilihBuat pekerjaan pelabelan.
4. UntukNama Job, masukkan nama untuk pekerjaan pelabelan Anda.
5. (Opsional) Jika Anda ingin mengidentifikasi label Anda dengan kunci, pilihSaya ingin menentukan nama atribut label yang berbeda dari nama pekerjaan pelabelan. Jika Anda tidak memilih opsi ini, nama pekerjaan pelabelan yang Anda tentukan pada langkah sebelumnya akan digunakan untuk mengidentifikasi label Anda di file manifes keluaran Anda.

6. Pilih pengaturan data untuk membuat koneksi antara dataset input Anda dan Ground Truth.

   - UntukPengaturan data:
     - Ikuti instruksi diPengaturan Data Otomatisuntuk pekerjaan pelabelan gambar, teks, dan klip video.
     - Ikuti instruksi diPengaturan Data Input Bingkai Video Otomatisuntuk pekerjaan pelabelan bingkai video.
   - UntukPengaturan data:
     - UntukMasukan lokasi dataset, berikan lokasi di Amazon S3 tempat file manifes masukan Anda berada. Misalnya, jika file manifes masukan Anda, manifest.json, terletak dicontoh-ember, masukkans3://example-bucket/manifest.json.
     - UntukLokasi dataset, berikan lokasi di Amazon S3 tempat Anda ingin Ground Truth menyimpan data output dari pekerjaan pelabelan Anda.
7. UntukPeran IAM, pilih peran IAM yang ada atau buat peran IAM dengan izin untuk mengakses sumber daya Anda di Amazon S3, untuk menulis ke bucket Amazon S3 keluaran yang ditentukan di atas, dan dengan SageMaker kebijakan eksekusi terlampir.
8. (Opsional) UntukKonfigurasi tambahan, Anda dapat menentukan berapa banyak kumpulan data Anda yang ingin diberi label oleh pekerja, dan jika Anda mau SageMaker untuk mengenkripsi data keluaran untuk pekerjaan pelabelan Anda menggunakanAWS KMSkunci enkripsi. Untuk mengenkripsi data Anda, Anda harus memiliki dataAWS KMSizin yang dilampirkan pada peran IAM yang Anda berikan di langkah sebelumnya. Untuk rincian lebih lanjut, lihat the section called “Izin IAM”.
9. DiJenis tugasbagian, di bawahKategori tugas, gunakan daftar dropdown untuk memilih kategori tugas Anda.
10. DiPemilihan tugas, pilih jenis tugas Anda.
11. (Opsional) Berikan tag untuk pekerjaan pelabelan Anda agar lebih mudah ditemukan di konsol nanti.
12. Pilih Selanjutnya.
13. DiPekerjabagian, pilih jenis tenaga kerja yang ingin Anda gunakan. Untuk detail selengkapnya tentang opsi tenaga kerja Anda, lihatMembuat dan Mengelola Tenaga Kerja.
14. (Opsional) Setelah Anda memilih tenaga kerja Anda, tentukanBatas waktu tugas. Ini adalah jumlah waktu maksimum yang dimiliki seorang pekerja untuk mengerjakan suatu tugas.

Untuk tugas anotasi cloud titik 3D, batas waktu tugas default adalah 3 hari. Batas waktu default untuk klasifikasi teks dan gambar serta pekerjaan pelabelan verifikasi label adalah 5 menit. Batas waktu default untuk semua pekerjaan pelabelan lainnya adalah 60 menit.

15. (Opsional) Untuk kotak pembatas, segmentasi semantik, bingkai video, dan jenis tugas cloud titik 3D, Anda dapat memilihTampilkan label yang adajika Anda ingin menampilkan label untuk kumpulan data masukan agar pekerja dapat memverifikasi atau menyesuaikan.

    Untuk pekerjaan pelabelan segmentasi kotak pembatas dan semantik, ini akan menciptakan pekerjaan pelabelan penyesuaian.

    Untuk pekerjaan pelabelan cloud titik 3D dan bingkai video:

    - PilihPenyesuaianuntuk membuat pekerjaan pelabelan penyesuaian. Ketika Anda memilih opsi ini, Anda dapat menambahkan label baru tetapi Anda tidak dapat menghapus atau mengedit label yang ada dari pekerjaan sebelumnya. Secara opsional, Anda dapat memilih atribut kategori label dan atribut bingkai yang ingin diedit oleh pekerja. Untuk membuat atribut dapat diedit, pilih kotak centangIzinkan pekerja untuk mengedit atribut iniuntuk atribut itu.

      Secara opsional, Anda dapat menambahkan kategori label baru dan atribut bingkai.
    - PilihVerifikasiuntuk membuat pekerjaan pelabelan penyesuaian. Ketika Anda memilih opsi ini, Anda tidak dapat menambahkan, memodifikasi, atau menghapus label yang ada dari pekerjaan sebelumnya. Secara opsional, Anda dapat memilih atribut kategori label dan atribut bingkai yang ingin diedit oleh pekerja. Untuk membuat atribut dapat diedit, pilih kotak centangIzinkan pekerja untuk mengedit atribut iniuntuk atribut itu.

      Sebaiknya Anda dapat menambahkan atribut kategori label baru ke label yang ingin diverifikasi oleh pekerja, atau menambahkan satu atau beberapa atribut bingkai agar pekerja memberikan informasi tentang seluruh bingkai.

    Untuk informasi selengkapnya, lihat Verifikasi dan Sesuaikan Label.

16. Konfigurasikan UI pekerja Anda:

    - Jika Anda menggunakantipe tugas bawaan, tentukan instruksi dan label pekerja.
      - Untuk klasifikasi gambar dan klasifikasi teks (single dan multi-label) Anda harus menentukan setidaknya dua kategori label. Untuk semua jenis tugas bawaan lainnya, Anda harus menentukan setidaknya satu kategori label.

- (Opsional) Jika Anda membuat pekerjaan pelabelan cloud titik 3D atau bingkai video, Anda dapat menentukan atribut kategori label (tidak didukung untuk segmentasi semantik cloud titik 3D) dan atribut bingkai. Atribut kategori label dapat ditetapkan untuk satu atau beberapa label. Atribut bingkai akan muncul di setiap titik cloud atau label pekerja bingkai video. Untuk mempelajari lebih lanjut, lihatAntarmuka Pengguna Pekerja (UI)untuk awan titik 3D danAntarmuka Pengguna Pekerja (UI)untuk bingkai video.
- (Opsional) TambahkanInstruksi tambahanuntuk membantu pekerja Anda menyelesaikan tugas Anda.

- Jika Anda membuat alur kerja pelabelan khusus, Anda harus:
  - Masukkantemplate kustomdi kotak kode. Template khusus dapat dibuat menggunakan kombinasi HTML, bahasa template Liquid, dan komponen web bawaan kami. Secara opsional, Anda dapat memilih template dasar dari menu tarik-turun untuk memulai.
  - Tentukan fungsi lambda pra-anotasi dan pasca-anotasi. Untuk mempelajari cara membuat fungsi-fungsi ini, lihatLangkah 3: Memproses dengan AWS Lambda.

17. (Opsional) Anda dapat memilihLihat pratinjauuntuk melihat pratinjau instruksi, label, dan interaksi pekerja Anda dengan UI pekerja. Pastikan pemblokir pop-up browser dinonaktifkan sebelum membuat pratinjau.
18. Pilih Create (Buat).

Setelah Anda berhasil membuat pekerjaan pelabelan Anda, Anda diarahkan kePekerjaan pelabelanhalaman. Status pekerjaan pelabelan yang baru saja Anda buat adalahSedang berlangsung. Status ini semakin diperbarui saat pekerja menyelesaikan tugas Anda. Ketika semua tugas berhasil diselesaikan, status berubah menjadiSelesai.

Jika masalah terjadi saat membuat pekerjaan pelabelan, statusnya berubah menjadiGagal.

Untuk melihat detail lebih lanjut tentang pekerjaan, pilih nama pekerjaan pelabelan.

Langkah Selanjutnya

Setelah status pekerjaan pelabelan Anda berubah menjadiSelesai, Anda dapat melihat data keluaran di bucket Amazon S3 yang Anda tentukan saat membuat pekerjaan pelabelan. Untuk detail tentang format data keluaran Anda, lihatData Output.

Buat Job Pelabelan (API)

Untuk membuat pekerjaan pelabelan menggunakan Amazon SageMaker API, Anda menggunakan`CreateLabelingJob`operasi. Untuk petunjuk spesifik tentang membuat pekerjaan pelabelan untuk jenis tugas bawaan, lihat ituhalaman data data. Untuk mempelajari cara membuat pekerjaan pelabelan streaming, yang merupakan pekerjaan pelabelan yang berjalan terus-menerus, lihatBuat Pekerjaan Pelabelan Streaming.

Untuk menggunakan`CreateLabelingJob`operasi, Anda perlu melakukan hal berikut ini:

- Template tugas pekerja (`UiTemplateS3Uri`) atau tugas manusia UI ARN (`HumanTaskUiArn`) di Amazon S3.
  - Untuk pekerjaan cloud titik 3D, pekerjaan deteksi dan pelacakan objek video, dan pekerjaan NER, gunakan ARN yang terdaftar di`HumanTaskUiArn`untuk jenis data Anda.
  - Jika Anda menggunakan tipe tugas bawaan selain tugas cloud titik 3D, Anda dapat menambahkan instruksi pekerja ke salah satu templat yang sudah dibuat sebelumnya dan menyimpan template (menggunakan ekstensi.html atau .liquid) di bucket S3 Anda. Temukan templat pra-bangun dihalaman data data.
  - Jika Anda menggunakan alur kerja pelabelan kustom, Anda dapat membuat template kustom dan menyimpan template di bucket S3 Anda. Untuk mempelajari cara membuat template pekerja kustom, lihatLangkah 2: Membuat template tugas pekerja kustom Anda. Untuk elemen HTML kustom yang dapat Anda gunakan untuk menyesuaikan template Anda, lihatCrowd HTML Elemen Referensi. Untuk repositori template demo untuk berbagai tugas pelabelan, lihatAmazon SageMaker Ground Truth Contoh Tugas UI.
- File manifes input yang menentukan data input Anda di Amazon S3. Tentukan lokasi file manifes masukan Anda`ManifestS3Uri`. Untuk informasi tentang membuat manifes input, lihatInput Data. Jika Anda membuat pekerjaan pelabelan streaming, ini opsional. Untuk mempelajari cara membuat tugas pelabelan streaming, lihatBuat Pekerjaan Pelabelan Streaming.
- Sebuah bucket Amazon S3 untuk menyimpan data output Anda. Anda menentukan bucket ini, dan opsional, awalan di`S3OutputPath`.
- File konfigurasi kategori label. Setiap nama kategori label harus unik. Tentukan lokasi file ini di Amazon S3 menggunakan`LabelCategoryConfigS3Uri`parameter. Kategori format dan label untuk file ini bergantung pada jenis tugas yang Anda gunakan:
  - Untuk klasifikasi gambar dan klasifikasi teks (single dan multi-label) Anda harus menentukan setidaknya dua kategori label. Untuk semua jenis tugas lainnya, jumlah minimum kategori label yang diperlukan adalah satu.

- Untuk tugas pengenalan entitas bernama, Anda harus memberikan instruksi pekerja dalam file ini. LihatMenyediakan Instruksi Pekerja dalam File Konfigurasi Kategori Labeluntuk detail dan contoh.
- Untuk jenis tugas cloud titik 3D dan bingkai video, gunakan format diBuat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai.
- Untuk semua jenis tugas bawaan dan tugas khusus lainnya, file konfigurasi kategori label Anda harus berupa file JSON dalam format berikut. Identifikasi label yang ingin Anda gunakan dengan menggantilabel_1,label_2,...,label_ndengan kategori label Anda.

```
{
    "document-version": "2018-11-28"
    "labels": [
        {"label": "label_1"},
        {"label": "label_2"},
        ...
        {"label": "label_n"}
    ]
}
```

- SesiAWS Identity and Access Management(IAM) peran denganAmazonSageMakerGroundTruthExecutionkebijakan IAM terkelola terlampir dan dengan izin untuk mengakses bucket S3 Anda. Tentukan peran ini diRoleArn. Untuk mempelajari tentang kebijakan ini, lihatMenggunakan Kebijakan Terkelola IAM dengan Ground Truth. Jika Anda memerlukan izin yang lebih terperinci, lihatthe section called “Izin IAM”.

  Jika nama bucket input atau output Anda tidak mengandungsagemaker, Anda dapat melampirkan kebijakan yang serupa dengan berikut ini ke peran yang diteruskan keCreateLabelingJoboperasi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::my_input_bucket/*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::my_output_bucket/*"
            ]
        }
    ]
}
```

- Pra-anotasi dan pasca-anotasi (atau anotasi-konsolidasi)AWS LambdaAmazon Resource Name (ARN) untuk memproses data input dan output Anda.
  - Fungsi Lambda telah ditentukan sebelumnya di masing-masingAWSWilayah untuk jenis tugas bawaan. Untuk menemukan pra-anotasi Lambda ARN untuk Wilayah Anda, lihatPreHumanTaskLambdaArn. Untuk menemukan anotasi-konsolidasi Lambda ARN untuk Wilayah Anda, lihatAnnotationConsolidationLambdaArn.
  - Untuk alur kerja pelabelan khusus, Anda harus memberikan pra dan pasca-anotasi kustom Lambda ARN. Untuk mempelajari cara membuat fungsi Lambda ini, lihatLangkah 3: Memproses dengan AWS Lambda.
- Sebuah tim kerja ARN yang Anda tentukanWorkteamArn. Anda menerima tim kerja ARN saat Anda berlangganan tenaga kerja vendor atau membuat tim kerja pribadi. Jika Anda membuat pekerjaan pelabelan untuk bingkai video atau jenis tugas titik awan, Anda tidak dapat menggunakanAmazon Mechanical Turktenaga kerja. Untuk semua jenis tugas lainnya, untuk menggunakan tenaga kerja Mechanical Turk, gunakan ARN berikut. Ganti*region*denganAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan.

  ```
  arn:aws:sagemaker:region:394669845002:workteam/public-crowd/default
  ```

  Jika Anda menggunakanAmazon Mechanical Turktenaga kerja, gunakanContentClassifiersparameter dalamDataAttributesdariInputConfiguntuk menyatakan bahwa konten Anda terbebas dari informasi pribadi dan konten dewasa.

  Ground Truthmemerlukanbahwa data input Anda bebas dari informasi identitas pribadi (PII) jika Anda menggunakan tenaga kerja Mechanical Turk. Jika Anda menggunakan Mechanical Turk dan tidak menentukan bahwa data input Anda bebas dari PII menggunakanFreeOfPersonallyIdentifiableInformationbendera, pekerjaan pelabelan

Anda akan gagal. Gunakan`FreeOfAdultContent`tandai untuk menyatakan bahwa data masukan Anda bebas dari konten dewasa.SageMaker dapat membatasi pekerja Amazon Mechanical Turk yang dapat melihat tugas Anda jika berisi konten dewasa.

Untuk mempelajari tentang tim kerja dan tenaga kerja, lihatMembuat dan Mengelola Tenaga Kerja.

- Jika Anda menggunakan tenaga kerja Mechanical Turk, Anda harus menentukan harga yang akan Anda bayarkan kepada pekerja untuk melakukan satu tugas**`PublicWorkforceTaskPrice`**.
- Untuk mengkonfigurasi tugas, Anda harus memberikan deskripsi tugas dan judul menggunakan`TaskDescription`dan**`TaskTitle`**masing-masing. Secara opsional, Anda dapat memberikan batas waktu yang mengontrol berapa lama pekerja harus mengerjakan tugas individu (**`TaskTimeLimitInSeconds`**) dan berapa lama tugas tetap di portal pekerja, tersedia untuk pekerja (`TaskAvailabilityLifetimeInSeconds`).
- (Opsional) Untukbeberapa jenis data, Anda dapat memiliki beberapa pekerja memberi label pada objek data tunggal dengan memasukkan angka yang lebih besar dari satu untuk`NumberOfHumanWorkersPerDataObject`parameter. Untuk informasi selengkapnya tentang konsolidasi anotasi, lihatAnotasi Terkonsolidasi.
- (Opsional) Untuk membuat pekerjaan pelabelan data otomatis, tentukan salah satu ARN yang tercantum dalam`LabelingJobAlgorithmSpecificationArn`di`LabelingJobAlgorithmsConfig`. ARN ini mengidentifikasi algoritma yang digunakan dalam pekerjaan pelabelan data otomatis. Jenis tugas yang terkait dengan ARN ini harus sesuai dengan jenis tugas`PreHumanTaskLambdaArn`dan`AnnotationConsolidationLambdaArn`Anda tentukan. Pelabelan data otomatis didukung untuk jenis tugas berikut: klasifikasi gambar, kotak pembatas, segmentasi semantik, dan klasifikasi teks. Jumlah minimum objek yang diizinkan untuk pelabelan data otomatis adalah 1.250, dan kami sangat menyarankan untuk menyediakan minimal 5.000 objek. Untuk mempelajari data data otomatis, lihatPelabelan Data.
- (Opsional) Anda dapat menyediakan`StoppingConditions`yang menyebabkan pekerjaan pelabelan berhenti jika salah satu syarat terpenuhi. Anda dapat menggunakan kondisi berhenti untuk mengontrol biaya pekerjaan pelabelan.

Contoh

Contoh kode berikut menunjukkan cara membuat pekerjaan pelabelan menggunakan`CreateLabelingJob`. Untuk contoh tambahan, kami merekomendasikan Anda menggunakan salah satuPekerjaan Pelabelan Ground TruthNotebook Jupyter di SageMakerContoh bagian dari SageMaker instans notebook. Untuk mempelajari cara menggunakan contoh notebook

dari SageMaker Contoh, lihatContoh Notebook. Anda juga dapat melihat contoh notebook ini GitHub di dalamSageMaker Contoh repositori.

AWS SDK for Python (Boto3)

Berikut ini adalah contoh dariAWSPermintaan Python SDK (Boto3)untuk membuat pekerjaan pelabelan untuk tipe data bawaan di Wilayah US East (N. Virginia) menggunakan tenaga kerja pribadi. Ganti Semua*teks merah-italized*dengan sumber daya dan spesifikasi pekerjaan pelabelan Anda.

```
response = client.create_labeling_job(
    LabelingJobName="example-labeling-job",
    LabelAttributeName="label",
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': "s3://bucket/path/manifest-with-input-data.json"
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                "FreeOfPersonallyIdentifiableInformation"|"FreeOfAdultContent",
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': "s3://bucket/path/file-to-store-output-data",
        'KmsKeyId': "string"
    },
    RoleArn="arn:aws:iam::*:role/*",
    LabelCategoryConfigS3Uri="s3://bucket/path/label-categories.json",
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
    HumanTaskConfig={
        'WorkteamArn': "arn:aws:sagemaker:region:*:workteam/private-crowd/*",
        'UiConfig': {
            'UiTemplateS3Uri': "s3://bucket/path/custom-worker-task-template.html"
        },
        'PreHumanTaskLambdaArn': "arn:aws:lambda:us-
east-1:432418664414:function:PRE-tasktype",
        'TaskKeywords': [
```

```
            "Images",
            "Classification",
            "Multi-label"
        ],
        'TaskTitle': "Multi-label image classification task",
        'TaskDescription': "Select all labels that apply to the images shown",
        'NumberOfHumanWorkersPerDataObject': 1,
        'TaskTimeLimitInSeconds': 3600,
        'TaskAvailabilityLifetimeInSeconds': 21600,
        'MaxConcurrentTaskCount': 1000,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': "arn:aws:lambda:us-
east-1:432418664414:function:ACS-"
        },
    Tags=[
        {
            'Key': "string",
            'Value': "string"
        },
    ]
)
```

AWS CLI

Berikut ini adalah contoh dariAWSPermintaan data bawaan di Wilayah US East (N. Virginia) menggunakanTenaga kerja Amazon Mechanical Turk. Untuk informasi selengkapnya, lihatstart-human-loopdi dalamAWS CLIReferensi Perintah. Ganti Semua*teks merah-italized*dengan sumber daya dan spesifikasi pekerjaan pelabelan Anda.

```
$ aws --region us-east-1 sagemaker create-labeling-job \
--labeling-job-name "example-labeling-job" \
--label-attribute-name "label" \
--role-arn "arn:aws:iam::account-id:role/role-name" \
--input-config '{
        "DataAttributes": {
            "ContentClassifiers": [
                "FreeOfPersonallyIdentifiableInformation",
                "FreeOfAdultContent"
            ]
        },
        "DataSource": {
            "S3DataSource": {
                "ManifestS3Uri": "s3://bucket/path/manifest-with-input-data.json"
```

```
            }
        }
    }' \
--output-config '{
        "KmsKeyId": "",
        "S3OutputPath": "s3://bucket/path/file-to-store-output-data"
    }' \
--human-task-config '{
        "AnnotationConsolidationConfig": {
            "AnnotationConsolidationLambdaArn": "arn:aws:lambda:us-
east-1:432418664414:function:ACS-"
        },
        "TaskAvailabilityLifetimeInSeconds": 21600,
        "TaskTimeLimitInSeconds": 3600,
        "NumberOfHumanWorkersPerDataObject": 1,
        "PreHumanTaskLambdaArn":  "arn:aws:lambda:us-
east-1:432418664414:function:PRE-tasktype",
        "WorkteamArn": "arn:aws:sagemaker:us-east-1:394669845002:workteam/public-
crowd/default",
        "PublicWorkforceTaskPrice": {
            "AmountInUsd": {
                "Dollars": 0,
                "TenthFractionsOfACent": 6,
                "Cents": 3
            }
        },
        "TaskDescription": "Select all labels that apply to the images shown",
        "MaxConcurrentTaskCount": 1000,
        "TaskTitle": "Multi-label image classification task",,
        "TaskKeywords": [
            "Images",
            "Classification",
            "Multi-label"
        ],
        "UiConfig": {
            "UiTemplateS3Uri": "s3://bucket/path/custom-worker-task-template.html"
        }
    }'
```

Untuk informasi selengkapnya tentang operasi ini, lihatCreateLabelingJob. Untuk informasi tentang cara menggunakan SDK khusus bahasa lainnya, lihatLihat Jugadi dalamCreateLabelingJobsTopik.

Buat Pekerjaan Pelabelan Streaming

Pekerjaan pelabelan streaming memungkinkan Anda mengirim objek data individual secara real time ke pekerjaan pelabelan streaming yang terus berjalan. Untuk membuat pekerjaan pelabelan streaming, Anda harus membuat topik masukan Amazon SNS dan menentukan topik ini dalam `CreateLabelingJob`parameter`InputConfig`. `SnsDataSource` Secara opsional, Anda juga dapat membuat topik keluaran Amazon SNS dan menentukannya `OutputConfig` jika Anda ingin menerima data label secara real time.

> **Important**
>
> Jika Anda adalah pengguna baru pekerjaan pelabelan streaming Ground Truth, disarankan agar Anda meninjau Pekerjaan Pelabelan Streaming Ground Truth sebelum membuat pekerjaan pelabelan streaming.

Gunakan bagian berikut untuk membuat sumber daya yang Anda butuhkan dan dapat digunakan untuk membuat pekerjaan pelabelan streaming:

- Pelajari cara membuat topik SNS dengan izin yang diperlukan untuk pekerjaan pelabelan streaming Ground Truth dengan mengikuti langkah-langkah di dalamnya. Membuat Topik Input dan Output Amazon SNS Topik SNS Anda harus dibuat di AWS Wilayah yang sama dengan pekerjaan pelabelan Anda.
- Lihat Berlangganan Endpoint ke Topik Output Amazon SNS Anda untuk mempelajari cara menyiapkan titik akhir untuk menerima data keluaran tugas pelabelan pada titik akhir yang ditentukan setiap kali tugas pelabelan selesai.
- Untuk mempelajari cara mengonfigurasi bucket Amazon S3 Anda untuk mengirim notifikasi ke topik masukan Amazon SNS Anda, lihat. Menyiapkan Pemberitahuan Peristiwa Amazon S3 Bucket
- Secara opsional, tambahkan objek data yang ingin diberi label segera setelah pekerjaan pelabelan dimulai ke manifes masukan Anda. Untuk informasi selengkapnya, lihat Buat File Manifest (Opsional).
- Ada sumber daya lain yang diperlukan untuk membuat pekerjaan pelabelan, seperti peran IAM, bucket Amazon S3, templat tugas pekerja, dan kategori label. Ini dijelaskan dalam dokumentasi Ground Truth tentang pembuatan pekerjaan pelabelan. Untuk informasi selengkapnya, lihat Buat Job Pelabelan.

> **Important**
>
> Saat Anda membuat pekerjaan pelabelan, Anda harus memberikan peran eksekusi IAM. Lampirkan kebijakan AWS terkelola AmazonSageMakerGroundTruthExecutionke peran ini untuk memastikan bahwa kebijakan tersebut memerlukan izin untuk menjalankan tugas pelabelan Anda.

Saat Anda mengirimkan permintaan untuk membuat pekerjaan pelabelan streaming, status pekerjaan pelabelan Anda adalah`Initializing`. Setelah pekerjaan pelabelan aktif, negara berubah menjadi`InProgress`. Jangan mengirim objek data baru ke pekerjaan pelabelan Anda atau mencoba menghentikan pekerjaan pelabelan Anda saat berada di `Initializing` negara bagian. Setelah status berubah`InProgress`, Anda dapat mulai mengirim objek data baru menggunakan Amazon SNS dan konfigurasi Amazon S3.

Topik

- Membuat Topik Input dan Output Amazon SNS
- Menyiapkan Pemberitahuan Peristiwa Amazon S3 Bucket
- Buat File Manifest (Opsional)
- Contoh: Gunakan SageMaker API Untuk Membuat Pekerjaan Pelabelan Streaming
- Hentikan Pekerjaan Pelabelan Streaming

Membuat Topik Input dan Output Amazon SNS

Anda perlu membuat input Amazon SNS untuk membuat pekerjaan pelabelan streaming. Secara opsional, Anda dapat memberikan topik keluaran Amazon SNS.

Saat Anda membuat topik Amazon SNS untuk digunakan dalam pekerjaan pelabelan streaming Anda, catat topik Amazon Resource Name (ARN). ARN akan menjadi nilai input untuk parameter `SnsTopicArn` di `InputConfig` dan `OutputConfig` ketika Anda membuat pekerjaan pelabelan.

Membuat Topik Masukan

Topik masukan Anda digunakan untuk mengirim objek data baru ke Ground Truth. Untuk membuat topik masukan, ikuti petunjuk dalam Membuat topik Amazon SNS di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Catat ARN topik masukan Anda dan gunakan sebagai masukan untuk `CreateLabelingJob` parameter `SnsTopicArn` di`InputConfig`.

Membuat Topik Output

Jika Anda memberikan topik keluaran, itu digunakan untuk mengirim pemberitahuan ketika objek data diberi label. Saat Anda membuat topik, Anda memiliki opsi untuk menambahkan kunci enkripsi. Gunakan opsi ini untuk menambahkan kunci yang dikelola AWS Key Management Service pelanggan ke topik Anda untuk mengenkripsi data keluaran pekerjaan pelabelan Anda sebelum dipublikasikan ke topik keluaran Anda.

Untuk membuat topik keluaran, ikuti petunjuk dalam Membuat topik Amazon SNS di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Jika Anda menambahkan enkripsi, Anda harus melampirkan izin tambahan ke topik. Tambahkan Enkripsi ke Topik Output Anda (Opsional)Lihat. untuk informasi lebih lanjut.

> **Important**
>
> Untuk menambahkan kunci yang dikelola pelanggan ke topik keluaran Anda saat membuat topik di konsol, jangan gunakan opsi alias/aws/sns (Default). Pilih kunci yang dikelola pelanggan yang Anda buat.

Catat topik masukan ARN Anda dan gunakan dalam `CreateLabelingJob` permintaan Anda dalam parameter `SnsTopicArn` di`OutputConfig`.

Tambahkan Enkripsi ke Topik Output Anda (Opsional)

Untuk mengenkripsi pesan yang dipublikasikan ke topik keluaran Anda, Anda perlu memberikan kunci yang dikelola AWS KMS pelanggan untuk topik Anda. Ubah kebijakan berikut dan tambahkan ke kunci yang dikelola pelanggan Anda untuk memberikan izin Ground Truth untuk mengenkripsi data keluaran sebelum mempublikasikannya ke topik keluaran Anda.

Ganti *<account_id>*dengan ID akun yang Anda gunakan untuk membuat topik Anda. Untuk mempelajari cara menemukan ID AWS akun Anda, lihat Menemukan ID AWS Akun Anda.

```
{
    "Id": "key-console-policy",
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<account_id>:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<account_id>:role/Admin"
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        }
    ]
}
```

Selain itu, Anda harus mengubah dan menambahkan kebijakan berikut ke peran eksekusi yang Anda gunakan untuk membuat tugas pelabelan Anda (nilai masukan untuk`RoleArn`).

Ganti *<account_id>*dengan ID akun yang Anda gunakan untuk membuat topik Anda. Ganti *<region>*dengan AWS Wilayah yang Anda gunakan untuk membuat pekerjaan pelabelan Anda. Ganti *<key_id>* dengan ID kunci yang dikelola pelanggan Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "sid1",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:<region>:<account_id>:key/<key_id>"
    }
  ]
}
```

Untuk informasi selengkapnya tentang membuat dan mengamankan kunci, lihat Membuat Kunci dan Menggunakan Kebijakan Utama dalam Panduan AWS Key Management Service Pengembang.

Berlangganan Endpoint ke Topik Output Amazon SNS Anda

Saat pekerja menyelesaikan tugas pelabelan dari pekerjaan pelabelan streaming Ground Truth, Ground Truth menggunakan topik keluaran Anda untuk mempublikasikan data keluaran ke satu atau lebih titik akhir yang Anda tentukan. Untuk menerima notifikasi saat pekerja menyelesaikan tugas pelabelan, Anda harus berlangganan titik akhir ke topik keluaran Amazon SNS Anda.

Untuk mempelajari cara menambahkan titik akhir ke topik keluaran Anda, lihat Berlangganan topik Amazon SNS di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Untuk mempelajari lebih lanjut tentang format data keluaran yang dipublikasikan ke titik akhir ini, lihatData Output.

> **Important**
>
> Jika Anda tidak berlangganan titik akhir ke topik keluaran Amazon SNS, Anda tidak akan menerima pemberitahuan saat objek data baru diberi label.

Menyiapkan Pemberitahuan Peristiwa Amazon S3 Bucket

Anda dapat menambahkan notifikasi peristiwa ke bucket Amazon S3 Anda menggunakan konsol Amazon S3, API, dan AWS SDK khusus bahasa, atau. AWS Command Line Interface Siapkan acara

ini untuk mengirim notifikasi ke topik masukan Amazon SNS yang sama dengan `SnsTopicArn` yang Anda tentukan `InputConfig` saat Anda membuat pekerjaan pelabelan. Jangan menyiapkan notifikasi peristiwa menggunakan lokasi Amazon S3 yang sama dengan yang Anda tentukan `OutputConfig` — melakukannya dapat mengakibatkan objek data yang tidak diinginkan diproses oleh Ground Truth untuk pelabelan. `S3OutputPath`

Anda memutuskan jenis acara yang ingin Anda kirim ke topik Amazon SNS Anda. Tanah Kebenaran menciptakan pekerjaan pelabelan ketika Anda mengirim peristiwa penciptaan objek.

Struktur peristiwa yang dikirim ke topik masukan Amazon SNS Anda harus berupa pesan JSON yang diformat menggunakan struktur yang sama yang ditemukan dalam struktur pesan peristiwa.

Untuk melihat contoh cara menyiapkan notifikasi peristiwa untuk bucket Amazon S3 Anda menggunakan konsol Amazon S3, SDK untuk .NET, dan AWS AWS SDK untuk Java, ikuti panduan ini, Walkthrough: Konfigurasikan bucket untuk notifikasi (topik SNS atau antrean SQS) di Panduan Pengguna Amazon Simple Storage Service.

Buat File Manifest (Opsional)

Saat Anda membuat pekerjaan pelabelan streaming, Anda memiliki opsi satu kali untuk menambahkan objek (seperti gambar atau teks) ke file manifes masukan yang Anda tentukan`CreateLabelingJob`. `ManifestS3Uri` Ketika pekerjaan pelabelan streaming dimulai, objek ini dikirim ke pekerja atau ditambahkan ke antrean Amazon SQS jika jumlah total objek melebihi. `MaxConcurrentTaskCount` Hasilnya ditambahkan ke jalur Amazon S3 yang Anda tentukan saat membuat pekerjaan pelabelan secara berkala saat pekerja menyelesaikan tugas pelabelan. Data keluaran dikirim ke titik akhir apa pun yang Anda berlangganan topik keluaran Anda.

Jika Anda ingin menyediakan objek awal untuk diberi label, buat file manifes yang mengidentifikasi objek ini dan letakkan di Amazon S3. Tentukan URI S3 dari file manifes ini di `ManifestS3Uri` dalam`InputConfig`.

Untuk mempelajari cara memformat file manifes Anda, lihatInput Data. Untuk menggunakan SageMaker konsol agar secara otomatis menghasilkan file manifes (tidak didukung untuk jenis tugas cloud titik 3D), lihatPengaturan Data Otomatis.

Contoh: Gunakan SageMaker API Untuk Membuat Pekerjaan Pelabelan Streaming

Berikut ini adalah contoh permintaan AWS Python SDK (Boto3) yang dapat Anda gunakan untuk memulai pekerjaan pelabelan streaming untuk jenis tugas bawaan di Wilayah AS Timur (Virginia Utara). Untuk rincian lebih lanjut tentang setiap parameter di bawah ini lihat `CreateLabelingJob`.

Untuk mempelajari cara membuat pekerjaan pelabelan menggunakan API ini dan SDK spesifik bahasa terkait, lihat Membuat Pekerjaan Pelabelan (API).

Dalam contoh ini, perhatikan parameter berikut:

- `SnsDataSource`— Parameter ini muncul di `InputConfig` dan `OutputConfig` dan digunakan untuk mengidentifikasi masukan dan output Anda topik Amazon SNS masing-masing. Untuk membuat pekerjaan pelabelan streaming, Anda diharuskan untuk memberikan topik masukan Amazon SNS. Secara opsional, Anda juga dapat memberikan topik keluaran Amazon SNS.
- `S3DataSource`- Parameter ini opsional. Gunakan parameter ini jika Anda ingin menyertakan file manifes masukan objek data yang ingin diberi label segera setelah pekerjaan pelabelan dimulai.
- `StoppingConditions`- Parameter ini diabaikan saat Anda membuat pekerjaan pelabelan streaming. Untuk mempelajari lebih lanjut tentang menghentikan pekerjaan pelabelan streaming, lihatHentikan Pekerjaan Pelabelan Streaming.
- Pekerjaan pelabelan streaming tidak mendukung pelabelan data otomatis. Jangan sertakan `LabelingJobAlgorithmsConfig` parameternya.

```
response = client.create_labeling_job(
    LabelingJobName= 'example-labeling-job',
    LabelAttributeName='label',
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': 's3://bucket/path/manifest-with-input-data.json'
            },
            'SnsDataSource': {
                'SnsTopicArn': 'arn:aws:sns:us-east-1:123456789012:your-sns-input-
topic'
            }
        },
        'DataAttributes': {
            'ContentClassifiers': [
                'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
            ]
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/file-to-store-output-data',
        'KmsKeyId': 'string',
        'SnsTopicArn': 'arn:aws:sns:us-east-1:123456789012:your-sns-output-topic'
```

```
    },
    RoleArn='arn:aws:iam::*:role/*',
    LabelCategoryConfigS3Uri='s3://bucket/path/label-categories.json',
    HumanTaskConfig={
        'WorkteamArn': 'arn:aws:sagemaker:us-east-1:*:workteam/private-crowd/*',
        'UiConfig': {
            'UiTemplateS3Uri': 's3://bucket/path/custom-worker-task-template.html'
        },
        'PreHumanTaskLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:PRE-tasktype',
        'TaskKeywords': [
            'Example key word',
        ],
        'TaskTitle': 'Multi-label image classification task',
        'TaskDescription': 'Select all labels that apply to the images shown',
        'NumberOfHumanWorkersPerDataObject': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'MaxConcurrentTaskCount': 123,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': 'arn:aws:lambda:us-
east-1:432418664414:function:ACS-tasktype'
            }
        },
    Tags=[
        {
            'Key': 'string',
            'Value': 'string'
        },
    ]
)
```

Hentikan Pekerjaan Pelabelan Streaming

Anda dapat menghentikan pekerjaan pelabelan streaming secara manual menggunakan operasi StopLabelingJob.

Jika pekerjaan pelabelan Anda tetap menganggur selama lebih dari 10 hari, itu secara otomatis dihentikan oleh Ground Truth. Dalam konteks ini, pekerjaan pelabelan dianggap tidak aktif jika tidak ada objek yang dikirim ke topik input Amazon SNS dan tidak ada objek yang tersisa dalam antrean Amazon SQS Anda, menunggu untuk diberi label. Misalnya, jika tidak ada objek data yang diumpankan ke topik input Amazon SNS dan semua objek yang diumpankan ke pekerjaan pelabelan

sudah diberi label, Ground Truth memulai pengatur waktu. Setelah timer dimulai, jika tidak ada item yang diterima dalam periode 10 hari, pekerjaan pelabelan dihentikan.

Ketika pekerjaan pelabelan dihentikan, statusnya adalah `STOPPING` sementara Ground Truth membersihkan pelabelan sumber daya pekerjaan dan berhenti berlangganan topik Amazon SNS Anda dari antrean Amazon SQS Anda. Amazon SQS tidak dihapus oleh Ground Truth karena antrean ini mungkin berisi objek data yang belum diproses. Anda harus menghapus antrean secara manual jika Anda ingin menghindari biaya tambahan dari Amazon SQS. Untuk mempelajari selengkapnya, lihat harga Amazon SQS.

Buat File Konfigurasi Kategori Pelabelan dengan Kategori Label dan Atribut Bingkai

Saat Anda membuat cloud titik 3D atau pekerjaan pelabelan bingkai video menggunakan Amazon SageMaker Operasi API`CreateLabelingJob`, Anda menggunakan file konfigurasi kategori label untuk menentukan label dan instruksi pekerja Anda. Secara opsional, Anda juga dapat memberikan yang berikut dalam file atribut kategori label Anda:

- Anda dapat menyediakanatribut kategori labeluntuk bingkai video dan pelacakan objek awan titik 3D dan jenis tugas deteksi objek. Pekerja dapat menggunakan satu atau lebih atribut untuk memberikan informasi lebih lanjut tentang suatu objek. Misalnya, Anda mungkin ingin menggunakan atributtersumbatagar pekerja mengidentifikasi kapan suatu objek terhalang sebagian. Anda dapat menentukan atribut kategori label untuk satu label menggunakan`categoryAttributes`parameter, atau untuk semua label menggunakan`categoryGlobalAttributes`parameter.
- Anda dapat menyediakanatribut bingkaiuntuk bingkai video dan pelacakan objek awan titik 3D dan jenis tugas deteksi objek menggunakan`frameAttributes`. Saat Anda membuat atribut frame, atribut tersebut akan muncul di setiap frame atau titik awan dalam tugas pekerja. Dalam pekerjaan pelabelan bingkai video, ini adalah atribut yang ditetapkan pekerja ke seluruh bingkai video. Untuk pekerjaan pelabelan cloud titik 3D, atribut ini diterapkan ke cloud titik tunggal. Gunakan atribut frame agar pekerja memberikan informasi lebih lanjut tentang adegan dalam bingkai atau titik cloud tertentu.
- Untuk pekerjaan pelabelan bingkai video, Anda menggunakan file konfigurasi kategori label untuk menentukan jenis tugas (kotak pembatas, polyline, poligon, atau keypoint) yang dikirim ke pekerja.

Untuk pekerja, menentukan nilai untuk atribut kategori label dan atribut bingkai akan bersifat opsional.

> **Important**
>
> Anda hanya harus memberikan nama atribut label di`auditLabelAttributeName`jika Anda menjalankan pekerjaan audit untuk memverifikasi atau menyesuaikan label. Gunakan parameter ini untuk memasukkanLabelAttributeNamedigunakan dalam pekerjaan pelabelan yang menghasilkan anotasi yang Anda ingin pekerja Anda sesuaikan. Saat Anda membuat pekerjaan pelabelan di konsol, jika Anda tidak menentukan nama atribut label,Namapekerjaan Anda digunakan sebagai LabelAttributeName.

Topik

- Skema File Konfigurasi Kategori Label
- Contoh: File Konfigurasi Kategori Label untuk Pekerjaan Pelabelan Cloud Titik 3D
- Contoh: File Konfigurasi Kategori Label untuk Pekerjaan Pelabelan Bingkai Video
- Membuat Instruksi Pekerja

Skema File Konfigurasi Kategori Label

Tabel berikut mencantumkan elemen yang dapat dan harus Anda sertakan dalam file konfigurasi kategori label Anda.

> **Note**
>
> Parameter`annotationType`hanya didukung untuk pekerjaan pelabelan bingkai video.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`frameAttributes`	Tidak	Daftar objek JSON. Parameter yang Diperlukan di setiap Objek JSON: `name`, `type`, `description`	Gunakan parameter ini untuk membuat atribut frame yang diterapkan ke semua frame atau awan titik 3D dalam pekerjaan pelabelan Anda.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi

		`minimum`dan`maximum`diperlukan jika`type`adalah`"number"`
		
		Parameter opsional di setiap Objek JSON:
		
		`enum`, `editsAllowed` , `isRequired`
			Lihat tabel ketiga di bagian ini untuk informasi selengkapnya.

`categoryGlobalAttributes`	Tidak	Daftar objek JSON.
		
		Parameter yang Diperlukan di setiap Objek JSON:
		
		`name`, `type`
		
		`minimum`dan`maximum`diperlukan jika`type`adalah`"number"`
		
		Parameter opsional di setiap Objek JSON:
		
		`description` , `enum`, `editsAllowed` , `isRequired`
			Gunakan parameter ini untuk membuat atribut kategori label yang diterapkan ke semua label yang Anda tentukan`labels`. Lihat tabel ketiga di bagian ini untuk informasi selengkapnya.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi

`labels`	Ya	Daftar hingga 30 objek JSON
		
		Parameter yang Diperlukan di setiap Objek JSON:
		
		`label`
		
		Parameter opsional di setiap Objek JSON:
		
		`categoryAttributes` , `editsAllowed`
			Gunakan parameter ini untuk menentukan label, atau kelas Anda. Tambahkan satu`label`untuk setiap kelas.
			
			Untuk menambahkan atribut kategori label ke label, tambahkan `categoryAttributes` ke label itu.
			
			Gunakan`editsAllowed` untuk menentukan apakah label dapat diedit dalam pekerjaan pelabelan penyesuaian. set`editsAllowed` kepada`"none"`untuk verifikasi pelabelan pekerjaan.
			
			Lihat tabel berikut untuk informasi selengkapnya.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi

`annotationType` (hanya didukung untuk pekerjaan pelabelan bingkai video)	Tidak	String
		
		Parameter yang Diterima:
		
		`BoundingBox` , `Polyline`, `Polygon`, `Keypoint`
		
		Bawaan:
		
		`BoundingBox`
			Gunakan ini untuk menentukan jenis tugas untuk pekerjaan pelabelan bingkai video Anda. Misalnya, untuk tugas deteksi objek bingkai video poligon, pilih`Polygon`.
			
			Jika Anda tidak menentuka `nannotationType` saat Anda membuat pekerjaan pelabelan bingkai video, Ground Truth akan digunakan `BoundingBox` secara default.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`instructions`	Tidak	Objek JSON Parameter yang Diperlukan di setiap Objek JSON: `"shortInstruction"` , `"fullInstruction"`	Gunakan parameter ini untuk menambahkan instruksi pekerja untuk membantu pekerja Anda menyelesaikan tugas mereka. Untuk informasi selengkapnya tentang instruksi pekerja, lihatInstruksi pekerja. Instruksi singkat harus di bawah 255 karakter dan instruksi panjang harus di bawah 2.048 karakter. Untuk informasi selengkapnya, lihat Membuat Instruksi Pekerja.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`auditLabelAttributeName`	Diperlukan untuk jenis tugas penyesuaian dan verifikasi	String	MasukkanLabelAttributeNamedigunakan dalam pekerjaan pelabelan yang ingin Anda sesuaikan anotasi. Hanya gunakan parameter ini jika Anda membuat pekerjaan penyesuaian untuk bingkai video dan deteksi objek awan titik 3D, pelacakan objek, atau segmentasi semantik awan titik 3D.

Tabel berikut menjelaskan parameter yang dapat dan harus Anda gunakan untuk membuat daftar`Labels`. Setiap parameter harus disertakan dalam objek JSON.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`label`	Ya	String	Nama kategori label yang ditampilkan kepada pekerja. Setiap nama kategori label harus unik.
`categoryAttributes`	Tidak	Daftar objek JSON. Parameter yang Diperlukan di setiap Objek JSON:	Gunakan parameter ini untuk menambahkan atribut kategori label ke label

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi

		`name`, `type`
		
		`minimum`dan`maximum`d
		n
		jika`type`adalah"`numbe`
		
		Parameter opsional di setiap Objek JSON:
		
		`description` , `enum`, `editsAllo wed` , `isRequired`
			tertentu yang Anda tentukan`labels`.
			
			Untuk menambahkan satu atau beberapa atribut kategori label ke label, sertakan`categoryA ttributes` Objek JSON dalam hal yang sama`labels`Objek JSON`label`.
			Lihat tabel berikut untuk informasi selengkapnya.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`editsAllowed`	Tidak	String Nilai yang Didukung: `"none"`: tidak ada modifikasi yang tidak diperbolehkan. or `"any"`(Default): semua modifikasi diperbolehkan.	Menentukan apakah atau tidak label dapat diedit oleh pekerja. Untuk bingkai video atau awan titik 3Dpengatura npelabelan pekerjaan, tambahkan parameter ini ke satu atau lebih objek JSON di`labels`daftar untuk menentukan apakah pekerja dapat mengedit label atau tidak. Untuk cloud titik 3D dan bingkai videoverifikas ipelabelan pekerjaan , tambahkan parameter ini dengan nilai`"none"`untuk setiap objek JSON di`labels`Daftar. Ini akan membuat semua label tidak dapat diedit.

Tabel berikut menjelaskan parameter yang dapat dan harus Anda gunakan untuk membuat atribut bingkai menggunakan`frameAttributes`dan atribut kategori label menggunakan`categoryGlobalAttributes`dan`categoryAttributes`parameter.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`name`	Ya	String	Gunakan parameter ini untuk menetapkan nama ke kategori label atau atribut frame Anda. Ini adalah nama atribut yang dilihat pekerja. Setiap nama atribut kategori label dalam file konfigurasi kategori label Anda harus unik. Atribut kategori label global dan label atribut kategori label tertentu tidak dapat memiliki nama yang sama.
`type`	Ya	String Nilai yang Diperlukan: `"string"` atau `"number"`	Gunakan parameter ini untuk menentukan kategori label atau tipe atribut frame. Jika Anda menentukan `"string"` untuk `type` dan berikan `enum` nilai untuk atribut ini, pekerja akan dapat memilih dari salah satu pilihan yang Anda berikan. Jika Anda menentukan `"string"` untuk `type` dan

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
			janganlah memberika nenumnilai, pekerja dapat memasukkan teks formulir bebas. Jika Anda menentuka nnumberuntuktype, pekerja dapat memasukkan nomor antaraminimumdanmaximumno yang Anda tentukan.
enum	Tidak	Daftar string	Gunakan parameter ini untuk menentukan opsi yang dapat dipilih pekerja untuk kategori label atau atribut bingkai ini. Pekerja dapat memilih satu nilai yang ditentukan dalamenum. Misalnya, jika Anda menentuka n["foo", "buzz", "bar"] untukenum, pekerja dapat memilih salah satufoo,buzz, ataubar. Anda harus menentuka n"string"untuktypeuntuk menggunak anenumDaftar.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`description`	`frameAttributes` : Ya `categoryAttributes` atau `categoryGlobalAttributes` : Tidak	String	Gunakan parameter ini untuk menambahkan deskripsi kategori label atau atribut frame. Anda dapat menggunakan bidang ini untuk memberi pekerja informasi lebih lanjut tentang atribut. Bidang ini hanya diperlukan untuk atribut frame.
`minimum` dan `maximum`	Atribut yang diperlukan `type` adalah `"number"`	Bilangan bulat	Gunakan parameter ini untuk menentukan nilai minimum dan maksimum (inklusif) pekerja dapat masuk untuk kategori label numerik atau atribut bingkai. Anda harus menentukan `"number"` untuk `type` untuk menggunakan `minimum` dan `maximum`.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`editsAllowed`	Tidak	String Nilai yang Diperlukan: `"none"`: tidak ada modifikasi yang tidak diperbolehkan. or `"any"`(Default): semua modifikasi diperbolehkan.	Menentukan apakah atau tidak kategori label atau atribut frame dapat diedit oleh pekerja. Untuk bingkai video atau awan titik 3Dpengaturandanverifikasipelabelan pekerjaan, tambahkan parameter ini untuk kategori label dan atribut bingkai objek JSON untuk menentukan apakah pekerja dapat mengedit atribut atau tidak.
`isRequired`	Tidak	Boolean	Menentukan apakah pekerja diperlukan untuk membubuhi keterangan atribut. Pekerja tidak dapat mengirimkan pekerjaan sampai semua atribut yang diperlukan dianotasi.

Kuota atribut kategori label dan label

Anda dapat menentukan hingga 10 atribut kategori label per kelas. Kuota 10 atribut ini mencakup atribut kategori label global. Misalnya, jika Anda membuat empat atribut kategori label global, dan kemudian menetapkan tiga atribut kategori label untuk labelX, label yang akan memiliki 4+3 = 7

kategori label atribut secara total. Untuk semua kategori label dan batas atribut kategori label, lihat tabel berikut.

Tipe	MIN	Maks
Label (`Labels`)	1	30
Kuota karakter nama label	1	16
Kategori label atribut per label (jumlah`categoryAttributes` dan`categoryGlobalAttributes`)	0	10
Gratis bentuk teks entri label kategori atribut per label (jumlah`categoryAttributes` dan`categoryGlobalAttributes`).	0	5
Atribut bingkai	0	10
Atribut entri teks formulir gratis di`frameAttributes`.	0	5
Kuota karakter nama atribut (`name`)	1	16
Deskripsi atribut kuota karakter (`description`)	0	128
Jenis atribut karakter kuota (`type`)	1	16
Nilai yang diizinkan dienumDaftar untuk`string`tambahan	1	10

Tipe	MIN	Maks
Kuota karakter untuk nilai dienumdaftar	1	16
Karakter maksimum dalam respons teks bentuk bebas untuk teks formulir gratisframeAttributes	0	1000
Karakter maksimum dalam respons teks bentuk bebas untuk teks formulir gratiscategoryAttributes dancategoryGlobalAttributes	0	80

Contoh: File Konfigurasi Kategori Label untuk Pekerjaan Pelabelan Cloud Titik 3D

Pilih tab di tabel berikut untuk melihat contoh file konfigurasi kategori label awan titik 3D untuk deteksi objek, pelacakan objek, segmentasi semantik, penyesuaian, dan pekerjaan pelabelan verifikasi.

3D Point Cloud Object Tracking and Object Detection

Berikut ini adalah contoh file konfigurasi kategori label yang menyertakan atribut kategori label untuk deteksi objek awan titik 3D atau pekerjaan pelabelan pelacakan objek. Contoh ini mencakup atribut dua bingkai, yang akan ditambahkan ke semua awan titik yang dikirimkan ke pekerjaan pelabelan. KlasterCarlabel akan mencakup empat kategori label atribut—X,Y,Z, dan atribut global,W.

```
{
    "documentVersion": "2020-03-01",
    "frameAttributes": [
        {
            "name":"count players",
            "description":"How many players to you see in the scene?",
            "type":"number"
        },
        {
```

```
            "name":"select one",
            "description":"describe the scene",
            "type":"string",
            "enum":["clear","blurry"],
            "isRequired":true
        },
    ],
    "categoryGlobalAttributes": [
        {
            "name":"W",
            "description":"label-attributes-for-all-labels",
            "type":"string",
            "enum": ["foo", "buzz", "biz"]
        }
    ],
    "labels": [
        {
            "label": "Car",
            "categoryAttributes": [
                {
                    "name":"X",
                    "description":"enter a number",
                    "type":"number",
                },
                {
                    "name":"Y",
                    "description":"select an option",
                    "type":"string",
                    "enum":["y1", "y2"]
                },
                {
                    "name":"Z",
                    "description":"submit a free-form response",
                    "type":"string",
                }
            ]
        },
        {
            "label": "Pedestrian",
            "categoryAttributes": [...]
        }
    ],
    "instructions": {"shortInstruction":"Draw a tight Cuboid",
"fullInstruction":"<html markup>"}
```

```
}
```

3D Point Cloud Semantic Segmentation

Berikut ini adalah contoh file konfigurasi kategori label untuk pekerjaan pelabelan segmentasi semantik titik awan 3D.

Atribut kategori label tidak didukung untuk jenis tugas segmentasi semantik awan titik 3D. Atribut bingkai didukung. Jika Anda memberikan atribut kategori label untuk pekerjaan pelabelan segmentasi semantik, mereka akan diabaikan.

```
{
    "documentVersion": "2020-03-01",
    "frameAttributes": [
        {
            "name":"count players",
            "description":"How many players to you see in the scene?",
            "type":"number"
        },
        {
            "name":"select one",
            "description":"describe the scene",
            "type":"string",
            "enum":["clear","blurry"]
        },
    ],
    "labels": [
        {
            "label": "Car",
        },
        {
            "label": "Pedestrian",
        },
        {
            "label": "Cyclist",
        }
    ],
    "instructions": {"shortInstruction":"Select the appropriate label and
 paint all objects in the point cloud that it applies to the same color",
 "fullInstruction":"<html markup>"}
}
```

Pilih tab di tabel berikut untuk melihat contoh file konfigurasi kategori label untuk verifikasi awan titik 3D atau pekerjaan pelabelan penyesuaian.

3D Point Cloud Adjustment

Berikut ini adalah contoh file konfigurasi kategori label untuk deteksi objek awan titik 3D atau pekerjaan pelabelan penyesuaian pelacakan objek. Untuk pekerjaan pelabelan penyesuaian segmentasi semantik cloud titik 3D,`categoryGlobalAttributes`dan`categoryAttributes`tidak didukung.

Anda harus menyertakan`auditLabelAttributeName`untuk menentukan nama atribut label dari pekerjaan pelabelan sebelumnya yang Anda gunakan untuk membuat pekerjaan pelabelan penyesuaian. Secara opsional, Anda dapat menggunakan`editsAllowed`parameter untuk menentukan apakah label atau frame atribut dapat diedit atau belum.

```
{
    "documentVersion": "2020-03-01",
    "frameAttributes": [
        {
            "name":"count players",
            "description":"How many players to you see in the scene?",
            "type":"number"
        },
        {
            "name":"select one",
            "editsAllowed":"none",
            "description":"describe the scene",
            "type":"string",
            "enum":["clear","blurry"]
        },
    ],
    "categoryGlobalAttributes": [
        {
            "name":"W",
            "editsAllowed":"any",
            "description":"label-attributes-for-all-labels",
            "type":"string",
            "enum": ["foo", "buzz", "biz"]
        }
    ],
    "labels": [
        {
```

```
            "label": "Car",
            "editsAllowed":"any",
            "categoryAttributes": [
                {
                    "name":"X",
                    "description":"enter a number",
                    "type":"number"
                },
                {
                    "name":"Y",
                    "description":"select an option",
                    "type":"string",
                    "enum":["y1", "y2"],
                    "editsAllowed":"any"
                },
                {
                    "name":"Z",
                    "description":"submit a free-form response",
                    "type":"string",
                    "editsAllowed":"none"
                }
            ]
        },
        {
            "label": "Pedestrian",
            "categoryAttributes": [...]
        }
    ],
    "instructions": {"shortInstruction":"Draw a tight Cuboid",
 "fullInstruction":"<html markup>"},
    // include auditLabelAttributeName for label adjustment jobs
    "auditLabelAttributeName": "myPrevJobLabelAttributeName"
}
```

3D Point Cloud Verification

Berikut ini adalah contoh file konfigurasi kategori label yang dapat Anda gunakan untuk deteksi objek awan titik 3D atau pekerjaan pelabelan verifikasi pelacakan objek. Untuk pekerjaan pelabelan verifikasi segmentasi semantik titik 3D,`categoryGlobalAttributes`dan`categoryAttributes`tidak didukung.

Anda harus menyertakan`auditLabelAttributeName`untuk menentukan nama atribut label dari pekerjaan pelabelan sebelumnya yang Anda gunakan untuk membuat pekerjaan pelabelan

verifikasi. Selain itu, Anda harus menggunakan`editsAllowed`parameter untuk menentukan bahwa tidak ada label dapat diedit.

```
{
    "documentVersion": "2020-03-01",
    "frameAttributes": [
        {
            "name":"count players",
            "editsAllowed":"any",
            "description":"How many players to you see in the scene?",
            "type":"number"
        },
        {
            "name":"select one",
            "editsAllowed":"any",
            "description":"describe the scene",
            "type":"string",
            "enum":["clear","blurry"]
        },
    ],
    "categoryGlobalAttributes": [
        {
            "name":"W",
            "editsAllowed":"none",
            "description":"label-attributes-for-all-labels",
            "type":"string",
            "enum": ["foo", "buzz", "biz"]
        }
    ],
    "labels": [
        {
            "label": "Car",
            "editsAllowed":"none",
            "categoryAttributes": [
                {
                    "name":"X",
                    "description":"enter a number",
                    "type":"number",
                    "editsAllowed":"none"
                },
                {
                    "name":"Y",
                    "description":"select an option",
```

```
                    "type":"string",
                    "enum":["y1", "y2"],
                    "editsAllowed":"any"
                },
                {
                    "name":"Z",
                    "description":"submit a free-form response",
                    "type":"string",
                    "editsAllowed":"none"
                }
            ]
        },
        {
            "label": "Pedestrian",
            "editsAllowed":"none",
            "categoryAttributes": [...]
        }
    ],
    "instructions": {"shortInstruction":"Draw a tight Cuboid",
 "fullInstruction":"<html markup>"},
    // include auditLabelAttributeName for label verification jobs
    "auditLabelAttributeName": "myPrevJobLabelAttributeName"
}
```

Contoh: File Konfigurasi Kategori Label untuk Pekerjaan Pelabelan Bingkai Video

Alat anotasi yang tersedia untuk pekerja dan jenis tugas yang digunakan bergantung pada nilai yang Anda tentukan`annotationType`. Misalnya, jika Anda ingin pekerja menggunakan poin-poin penting untuk melacak perubahan dalam pose objek tertentu di beberapa frame, Anda akan menentukan`Keypoint`untuk`annotationType`. Jika Anda tidak menentukan tipe anotasi,`BoundingBox`akan digunakan secara default.

Berikut ini adalah contoh file konfigurasi kategori label keypoint bingkai video dengan atribut kategori label label. Contoh ini mencakup dua atribut bingkai, yang akan ditambahkan ke semua frame yang dikirimkan ke pekerjaan pelabelan. Klaster`Car`label akan mencakup empat kategori label atribut —X,Y,Z, dan atribut global,`W`.

```
{
    "documentVersion": "2020-03-01",
    "frameAttributes": [
        {
```

```
            "name":"count players",
            "description":"How many players to you see in the scene?",
            "type":"number"
        },
        {
            "name":"select one",
            "description":"describe the scene",
            "type":"string",
            "enum":["clear","blurry"]
        },
    ],
    "categoryGlobalAttributes": [
        {
            "name":"W",
            "description":"label-attributes-for-all-labels",
            "type":"string",
            "enum": ["foo", "buz", "buz2"]
        }
    ],
    "labels": [
        {
            "label": "Car",
            "categoryAttributes": [
                {
                    "name":"X",
                    "description":"enter a number",
                    "type":"number",
                },
                {
                    "name":"Y",
                    "description":"select an option",
                    "type":"string",
                    "enum": ["y1", "y2"]
                },
                {
                    "name":"Z",
                    "description":"submit a free-form response",
                    "type":"string",
                }
            ]
        },
        {
            "label": "Pedestrian",
            "categoryAttributes": [...]
```

```
        }
    ],
    "annotationType":"Keypoint",
    "instructions": {"shortInstruction":"add example short instructions here",
 "fullInstruction":"<html markup>"}
}
```

Pilih tab dari tabel berikut untuk melihat contoh file konfigurasi kategori label untuk penyesuaian bingkai video dan pekerjaan pelabelan verifikasi.

Video Frame Adjustment

Berikut ini adalah contoh file konfigurasi kategori label yang dapat Anda gunakan untuk pekerjaan pelabelan penyesuaian bingkai video.

Anda harus menyertakan`auditLabelAttributeName`untuk menentukan nama atribut label dari pekerjaan pelabelan sebelumnya yang Anda gunakan untuk membuat pekerjaan pelabelan verifikasi. Secara opsional, Anda dapat menggunakan`editsAllowed`parameter untuk menentukan apakah atau tidak label, kategori label atribut, atau atribut bingkai dapat diedit.

```
{
    "documentVersion": "2020-03-01",
    "frameAttributes": [
        {
            "name":"count players",
            "editsAllowed":"none",
            "description":"How many players to you see in the scene?",
            "type":"number"
        },
        {
            "name":"select one",
            "description":"describe the scene",
            "type":"string",
            "enum":["clear","blurry"]
        },
    ],
    "categoryGlobalAttributes": [
        {
            "name":"W",
            "editsAllowed":"any",
            "description":"label-attributes-for-all-labels",
            "type":"string",
            "enum": ["foo", "buz", "buz2"]
```

```
        }
    ],
    "labels": [
        {
            "label": "Car",
            "editsAllowed":"any",
            "categoryAttributes": [
                {
                    "name":"X",
                    "description":"enter a number",
                    "type":"number",
                    "editsAllowed":"any"
                },
                {
                    "name":"Y",
                    "description":"select an option",
                    "type":"string",
                    "enum": ["y1", "y2"],
                    "editsAllowed":"any"
                },
                {
                    "name":"Z",
                    "description":"submit a free-form response",
                    "type":"string",
                    "editsAllowed":"none"
                }
            ]
        },
        {
            "label": "Pedestrian",
            "editsAllowed":"none",
            "categoryAttributes": [...]
        }
    ],
    "annotationType":"Keypoint",
    "instructions": {"shortInstruction":"add example short instructions here",
 "fullInstruction":"<html markup>"},
    // include auditLabelAttributeName for label adjustment jobs
    "auditLabelAttributeName": "myPrevJobLabelAttributeName"
}
```

Video Frame Verification

Berikut ini adalah contoh file konfigurasi kategori label untuk pekerjaan pelabelan bingkai video.

Anda harus menyertakan`auditLabelAttributeName`untuk menentukan nama atribut label dari pekerjaan pelabelan sebelumnya yang Anda gunakan untuk membuat pekerjaan pelabelan verifikasi. Selain itu, Anda harus menggunakan`editsAllowed`parameter untuk menentukan bahwa tidak ada label dapat diedit.

```
{
    "documentVersion": "2020-03-01",
    "frameAttributes": [
        {
            "name":"count players",
            "editsAllowed":"none",
            "description":"How many players to you see in the scene?",
            "type":"number"
        },
        {
            "name":"select one",
            "editsAllowed":"any",
            "description":"describe the scene",
            "type":"string",
            "enum":["clear","blurry"]
        },
    ],
    "categoryGlobalAttributes": [
        {
            "name":"W",
            "editsAllowed":"none",
            "description":"label-attributes-for-all-labels",
            "type":"string",
            "enum": ["foo", "buz", "buz2"]
        }
    ],
    "labels": [
        {
            "label": "Car",
            "editsAllowed":"none",
            "categoryAttributes": [
                {
                    "name":"X",
                    "description":"enter a number",
                    "type":"number",
                    "editsAllowed":"any"
                },
                {
```

```
                    "name":"Y",
                    "description":"select an option",
                    "type":"string",
                    "enum": ["y1", "y2"],
                    "editsAllowed":"any"
                },
                {
                    "name":"Z",
                    "description":"submit a free-form response",
                    "type":"string",
                    "editsAllowed":"none"
                }
            ]
        },
        {
            "label": "Pedestrian",
            "editsAllowed":"none",
            "categoryAttributes": [...]
        }
    ],
    "annotationType":"Keypoint",
    "instructions": {"shortInstruction":"add example short instructions here",
 "fullInstruction":"<html markup>"},
    // include auditLabelAttributeName for label adjustment jobs
    "auditLabelAttributeName": "myPrevJobLabelAttributeName"
}
```

Membuat Instruksi Pekerja

Buat instruksi khusus untuk pelabelan pekerjaan untuk meningkatkan akurasi pekerja Anda dalam menyelesaikan tugas mereka. Instruksi Anda dapat diakses saat pekerja memilihPetunjukpilihan menu di UI pekerja. Instruksi singkat harus di bawah 255 karakter dan instruksi panjang harus di bawah 2.048 karakter.

Ada dua jenis instruksi:

- Petunjuk singkat- Instruksi ini ditampilkan untuk bekerja ketika mereka memilihPetunjukdi menu UI pekerja. Mereka harus memberikan referensi yang mudah untuk menunjukkan kepada pekerja cara yang benar untuk memberi label pada suatu objek.
- Petunjuk lengkap- Instruksi ini ditampilkan saat pekerja memilihPetunjuk lebihdalam instruksi jendela pop-up. Kami menyarankan Anda memberikan instruksi terperinci untuk menyelesaikan

tugas dengan beberapa contoh yang menunjukkan kasus tepi dan situasi sulit lainnya untuk memberi label objek.

Untuk pekerjaan pelabelan cloud titik 3D dan bingkai video, Anda dapat menambahkan instruksi pekerja ke file konfigurasi kategori label Anda. Anda dapat menggunakan string tunggal untuk membuat instruksi atau Anda dapat menambahkan HTML mark up untuk menyesuaikan tampilan instruksi Anda dan menambahkan gambar. Pastikan bahwa setiap gambar yang Anda sertakan dalam instruksi Anda tersedia untuk umum, atau jika instruksi Anda ada di Amazon S3, pekerja Anda memiliki akses baca sehingga mereka dapat melihatnya.

Gunakan

Data input yang Anda berikan ke Amazon SageMaker Ground Truth dikirim ke pekerja Anda untuk diberi label. Anda memilih data yang akan dikirim ke pekerja Anda dengan membuat satu file manifes yang mendefinisikan semua data yang memerlukan pelabelan atau dengan mengirimkan objek data input ke pekerjaan pelabelan streaming yang sedang berlangsung untuk diberi label secara real time.

Data keluaran adalah hasil dari pekerjaan pelabelan Anda. File data keluaran, atau file manifes tambahan, berisi data label untuk setiap objek yang Anda kirim ke tugas pelabelan dan metadata tentang label yang ditetapkan ke objek data.

Bila Anda menggunakan klasifikasi gambar (single dan multi-label), klasifikasi teks (single dan multi-label), deteksi objek, dan segmentasi semantik yang dibangun dalam jenis tugas untuk membuat pekerjaan pelabelan, Anda dapat menggunakan file manifes augmented yang dihasilkan untuk meluncurkan pekerjaan SageMaker pelatihan. Untuk demonstrasi tentang cara menggunakan manifes tambahan untuk melatih model pembelajaran mesin deteksi objek dengan Amazon SageMaker, lihat object_detection_augmented_manifest_training.ipynb. Untuk informasi selengkapnya, lihat Menyediakan Dataset Metadata untuk Training Jobs dengan Augmented Manifest File.

Topik

- Input Data
- Data Input Awan Titik 3D
- Data Input Bingkai Video
- Data Output

Input Data

Data input adalah objek data yang Anda kirim ke tenaga kerja Anda untuk diberi label. Ada dua cara untuk mengirim objek data ke Ground Truth untuk diberi label:

- Kirim daftar objek data yang memerlukan pelabelan menggunakan file manifes masukan.
- Kirim objek data individual secara real time ke pekerjaan pelabelan streaming yang terus berjalan.

Jika Anda memiliki kumpulan data yang perlu diberi label satu kali, dan Anda tidak memerlukan pekerjaan pelabelan yang sedang berlangsung, buat tugas pelabelan standar menggunakan file manifes masukan.

Jika Anda ingin secara teratur mengirim objek data baru ke pekerjaan pelabelan Anda setelah dimulai, buat pekerjaan pelabelan streaming. Saat membuat tugas pelabelan streaming, Anda dapat menggunakan file manifes masukan untuk menentukan grup data yang ingin diberi label segera saat pekerjaan dimulai. Anda dapat terus mengirim objek data baru ke pekerjaan pelabelan streaming selama aktif.

> **Note**
>
> Pekerjaan pelabelan streaming hanya didukung melalui SageMaker API. Anda tidak dapat membuat pekerjaan pelabelan streaming menggunakan SageMaker konsol.

Jenis tugas berikut memiliki persyaratan dan opsi data input khusus:

- Untuk persyaratan data input pekerjaan pelabelan cloud titik 3D, lihatData Input Awan Titik 3D.
- Untuk persyaratan data input pekerjaan pelabelan bingkai video, lihatData Input Bingkai Video.

Topik

- Filter dan Pilih Data untuk Pelabelan

Menggunakan File Manifes Masukan

Setiap baris dalam file manifes masukan adalah entri yang berisi objek, atau referensi ke objek, untuk diberi label. Entri juga dapat berisi label dari pekerjaan sebelumnya dan untuk beberapa jenis tugas, informasi tambahan.

Input data dan file manifes harus disimpan di Amazon Simple Storage Service (Amazon S3). Masing-masing memiliki persyaratan penyimpanan dan akses khusus, sebagai berikut:

- Bucket Amazon S3 yang berisi data input harus berada dalamAWS Wilayah yang sama dengan tempat Anda menjalankan Amazon SageMaker Ground Truth. Anda harus memberi Amazon SageMaker akses ke data yang disimpan di bucket Amazon S3 sehingga dapat membacanya. Untuk informasi lebih lanjut tentang bucket Amazon S3, lihat Bekerja dengan bucket Amazon S3.
- File manifes harus berada diAWS Wilayah yang sama dengan file data, tetapi tidak perlu berada di lokasi yang sama dengan file data. Ini dapat disimpan di bucket Amazon S3 apa pun yang dapat diakses oleh peranAWS Identity and Access Management (IAM) yang Anda tetapkan ke Ground Truth saat Anda membuat tugas pelabelan.

> **Note**
>
> Jenis tugas cloud titik 3D dan bingkai video memiliki persyaratan dan atribut manifes masukan yang berbeda.
> Untuk jenis tugas cloud titik 3D, lihatMembuat File Manifes Input untuk Job Pelabelan Cloud Titik 3D.
> Untuk jenis tugas bingkai video, lihatMembuat File Manifes Masukan Bingkai Video.

Manifes adalah file yang dikodekan UTF-8 di mana setiap baris adalah objek JSON yang lengkap dan valid. Setiap baris dibatasi oleh jeda garis standar,\ n atau\ r\ n. Karena setiap baris harus berupa objek JSON yang valid, Anda tidak dapat memiliki karakter line break yang tidak lolos. Untuk informasi selengkapnya tentang format data, lihat Garis JSON.

Setiap objek JSON dalam file manifes tidak boleh lebih dari 100.000 karakter. Tidak ada atribut tunggal dalam suatu objek dapat lebih besar dari 20.000 karakter. Nama atribut tidak dapat dimulai dengan$ (tanda dolar).

Setiap objek JSON dalam file manifes harus berisi salah satu kunci berikut:`source-ref` atau`source`. Nilai kunci ditafsirkan sebagai berikut:

- `source-ref`— Sumber objek adalah objek Amazon S3 yang ditentukan dalam nilai. Gunakan nilai ini ketika objek adalah objek biner, seperti gambar.
- `source`- Sumber objek adalah nilainya. Gunakan nilai ini ketika objek adalah nilai teks.

Berikut ini adalah contoh file manifes untuk file yang disimpan dalam bucket Amazon S3:

```
{"source-ref": "S3 bucket location 1"}
{"source-ref": "S3 bucket location 2"}
   ...
{"source-ref": "S3 bucket location n"}
```

Gunakan`source-ref` kunci untuk file gambar untuk kotak pembatas, klasifikasi gambar (tunggal dan multi-label), segmentasi semantik, dan klip video untuk pekerjaan pelabelan klasifikasi video. Pekerjaan pelabelan cloud titik 3D dan bingkai video juga menggunakan`source-ref` kunci tetapi pekerjaan pelabelan ini memerlukan informasi tambahan dalam file manifes masukan. Untuk informasi lebih lanjut, lihat Data Input Awan Titik 3D dan Data Input Bingkai Video.

Berikut ini adalah contoh file manifest dengan data input yang disimpan dalam manifes:

```
{"source": "Lorem ipsum dolor sit amet"}
{"source": "consectetur adipiscing elit"}
   ...
{"source": "mollit anim id est laborum"}
```

Gunakan`source` kunci untuk klasifikasi teks tunggal dan multi-label dan pekerjaan pelabelan pengenalan entitas bernama.

Anda dapat menyertakan pasangan kunci lainnya dalam file manifes. Pasangan ini dilewatkan ke file output tidak berubah. Ini berguna ketika Anda ingin menyampaikan informasi di antara aplikasi Anda. Untuk informasi selengkapnya, lihat Data Output.

Pengaturan Data Otomatis

Anda dapat menggunakan pengaturan data otomatis untuk membuat file manifes untuk pekerjaan pelabelan Anda di konsol Ground Truth menggunakan gambar, video, bingkai video, file teks

(.txt), dan file nilai dipisahkan koma (.csv) yang disimpan di Amazon S3. Saat Anda menggunakan pengaturan data otomatis, Anda menentukan lokasi Amazon S3 tempat data input Anda disimpan dan jenis data input, dan Ground Truth mencari file yang cocok dengan jenis tersebut di lokasi yang Anda tentukan.

> **Note**
>
> Ground Truth tidak menggunakanAWS KMS kunci untuk mengakses data input Anda atau menulis file manifes masukan di lokasi Amazon S3 yang Anda tentukan. Pengguna atau peran yang membuat tugas pelabelan harus memiliki izin untuk mengakses objek data input Anda di Amazon S3.

Sebelum menggunakan prosedur berikut, pastikan bahwa gambar atau file input Anda diformat dengan benar:

- File gambar - File gambar harus sesuai dengan batas ukuran dan resolusi yang tercantum dalam tabel yang ditemukan diKuota.
- File teks - Data teks dapat disimpan dalam satu atau lebih file.txt. Setiap item yang ingin diberi label harus dipisahkan dengan jeda baris standar.
- File CSV - Data teks dapat disimpan dalam satu atau lebih file.csv. Setiap item yang ingin diberi label harus berada dalam baris terpisah.
- Video - File video dapat berupa salah satu format berikut: .mp4, .ogg, dan.webm. Jika Anda ingin mengekstrak bingkai video dari file video Anda untuk deteksi objek atau pelacakan objek, lihatMenyediakan File Video.
- Bingkai video - Bingkai video adalah gambar yang diekstrak dari video. Semua gambar yang diekstrak dari satu video disebut sebagai urutan bingkai video. Setiap urutan bingkai video harus memiliki kunci awalan unik di Amazon S3. Lihat Menyediakan Bingkai video. Untuk tipe data ini, lihatPengaturan Data Input Bingkai Video Otomatis

> **Important**
>
> Untuk deteksi objek bingkai video dan pekerjaan pelabelan pelacakan objek bingkai video, lihatPengaturan Data Input Bingkai Video Otomatis untuk mempelajari cara menggunakan pengaturan data otomatis.

Gunakan petunjuk ini untuk secara otomatis mengatur koneksi set data input Anda dengan Ground Truth.

Secara otomatis menghubungkan data Anda di Amazon S3 dengan Ground Truth

1. Arahkan ke halaman pekerjaan Buat pelabelan di SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.

   Tautan ini menempatkan Anda diAWS Wilayah Virginia Utara (us-east-1). Jika data input Anda berada di bucket Amazon S3 di Wilayah lain, beralihlah ke Wilayah tersebut. Untuk mengubahAWS Wilayah Anda, pada bilah navigasi, pilih nama Wilayah yang ditampilkan saat ini.
2. Pilih Buat pekerjaan pelabelan.
3. Masukkan nama Job.
4. Di bagian Pengaturan data input, pilih Pengaturan data otomatis.
5. Masukkan URI Amazon S3 untuk lokasi S3 untuk set data input.
6. Tentukan lokasi S3 Anda untuk kumpulan data keluaran. Di sinilah data output Anda disimpan.
7. Pilih tipe Data Anda menggunakan daftar dropdown.
8. Gunakan menu tarik-turun di bawah Peran IAM untuk memilih peran eksekusi. Jika Anda memilih Buat peran baru, tentukan bucket Amazon S3 yang ingin Anda berikan izin peran ini untuk diakses. Peran ini harus memiliki izin untuk mengakses bucket S3 yang Anda tentukan dalam Langkah 5 dan 6.
9. Pilih Selesaikan pengaturan data.

GIF berikut menunjukkan cara menggunakan pengaturan data otomatis untuk data gambar. Contoh ini akan membuat file,`dataset-`*`YYMMDDTHHMMSS`*`.manifest` di bucket`example-groundtruth-images` Amazon S3 yang*`YYMMDDTHHmmSS`* menunjukkan year (YY), month (), day (MMDD), dan waktu dalam hours (HH), minutes (mm) dan seconds (ss), bahwa file manifes masukan dibuat.

Format Data yang Didukung

Saat Anda membuat file manifes masukan untuk jenis tugas bawaan secara manual, data input Anda harus dalam salah satu format file dukungan berikut untuk jenis data input masing-masing. Untuk mempelajari pengaturan data otomatis, lihatPengaturan Data Otomatis.

> **Tip**
>
> Saat Anda menggunakan penyiapan data otomatis, format data tambahan dapat digunakan untuk menghasilkan file manifes masukan untuk bingkai video dan jenis tugas berbasis teks.

Jenis Tugas	Jenis	Format Support	Contoh Masukan Manifest Baris
Kotak Pembatas, Segmentasi Semantik, Klasifikasi Gambar (Label Tunggal dan Multi-label), Verifikasi dan Sesuaikan Label	Citra	.jpg, .jpeg, .png	`{"source-ref": "s3://DOC-EXAMPLE-BUCKET1 /example-image.png "}`
Dinamakan Entity Recognition, Klasifikasi Teks (Single dan Multi-Label)	Teks	Teks mentah	`{"source": "Lorem ipsum dolor sit amet"}`
Klasifikasi Video	Klip video	.mp4, .ogg, dan.webm	`{"source-ref": "s3:///example-video.mp4 "}`
Deteksi Objek Bingkai Video, Pelacakan Objek Bingkai Video (kotak pembatas, polyline, poligon atau keypoint)	Bingkai video dan file urutan bingkai video (untuk Pelacakan Objek)	Bingkai video: .jpg, .jpeg, .png File urutan: .json	Lihat Membuat File Manifes Masukan Bingkai Video.
Segmentasi Semantik Awan Titik 3D, Deteksi Objek Awan	Titik awan dan file urutan titik awan (untuk Pelacakan Objek)	Titik awan: Format paket biner dan ASCII. Untuk informasi selengkap	Lihat Membuat File Manifes Input untuk Job Pelabelan Cloud Titik 3D.

Jenis Tugas	Jenis	Format Support	Contoh Masukan Manifest Baris
Titik 3D, Pelacakan Objek Awan Titik 3D		nya, lihat Format Data 3D Mentah yang Diterima. File urutan: .json	

Pekerjaan Pelabelan Streaming Ground Truth

Jika Anda ingin terus-menerus mengirim objek data baru ke Amazon SageMaker Ground Truth untuk diberi label, gunakan pekerjaan pelabelan streaming. Pekerjaan pelabelan streaming memungkinkan Anda untuk:

- Kirim objek dataset baru ke pekerja secara real time menggunakan pekerjaan pelabelan yang terus berjalan. Pekerja terus menerima objek data baru untuk diberi label selama pekerjaan pelabelan aktif dan objek baru dikirim ke sana.
- Dapatkan visibilitas ke dalam jumlah objek yang telah antri dan sedang menunggu untuk diberi label. Gunakan informasi ini untuk mengontrol aliran objek data yang dikirim ke pekerjaan pelabelan Anda.
- Menerima data label untuk objek data individu secara real time sebagai pekerja selesai label mereka.

Pekerjaan pelabelan streaming Ground Truth tetap aktif sampai dihentikan secara manual atau telah menganggur selama lebih dari 10 hari. Anda dapat sebentar-sebentar mengirim objek data baru ke pekerja saat pekerjaan pelabelan aktif.

Jika Anda adalah pengguna baru pekerjaan pelabelan streaming Ground Truth, disarankan agar Anda meninjauCara Kerjanya.

GunakanBuat Pekerjaan Pelabelan Streaming untuk mempelajari cara membuat pekerjaan pelabelan streaming.

> **Note**
>
> Pekerjaan pelabelan streaming Ground Truth hanya didukung melalui SageMaker API.

Topik

Cara Kerjanya

Saat Anda membuat pekerjaan pelabelan streaming Ground Truth, pekerjaan tetap aktif hingga dihentikan secara manual, tetap diam selama lebih dari 10 hari, atau tidak dapat mengakses sumber data input. Anda dapat sebentar-sebentar mengirim objek data baru ke pekerja saat aktif. Seorang pekerja dapat terus menerima objek data baru secara real time selama jumlah total tugas yang saat ini tersedia untuk pekerja kurang dari nilai di `MaxConcurrentTaskCount`. Jika tidak, objek data dikirim ke antrean yang dibuat Ground Truth atas nama Anda di Amazon Simple Queue Service (Amazon SQS) untuk diproses nanti. Tugas-tugas ini dikirim ke pekerja segera setelah jumlah total tugas yang saat ini tersedia untuk pekerja jatuh di bawah`MaxConcurrentTaskCount`. Jika objek data tidak dikirim ke pekerja setelah 14 hari, itu berakhir. Anda dapat melihat jumlah tugas yang tertunda dalam antrian dan menyesuaikan jumlah objek yang Anda kirim ke pekerjaan pelabelan. Misalnya, Anda dapat mengurangi kecepatan di mana Anda mengirim objek ke pekerjaan pelabelan jika backlog objek yang tertunda bergerak di atas ambang batas.

Kirim Data ke Job Pelabelan Streaming

Anda dapat secara opsional mengirimkan data masukan ke pekerjaan pelabelan streaming satu kali saat membuat tugas pelabelan menggunakan file manifes masukan. Setelah pekerjaan pelabelan dimulai dan statusnya`InProgress`, Anda dapat mengirimkan objek data baru ke pekerjaan pelabelan Anda secara real time menggunakan topik input Amazon SNS Anda dan pemberitahuan acara Amazon S3.

Kirim Objek Data Saat Anda Memulai Job Pelabelan (Satu Kali):

- Gunakan File Manifes Input — Anda dapat secara opsional menentukan file manifes masukan URI Amazon S3`ManifestS3Uri` saat Anda membuat tugas pelabelan streaming. Ground Truth mengirimkan setiap objek data dalam file manifes kepada pekerja untuk diberi label segera setelah pekerjaan pelabelan dimulai. Untuk mempelajari selengkapnya, lihat Buat File Manifest (Opsional).

Setelah Anda mengirimkan permintaan untuk membuat pekerjaan pelabelan streaming, statusnya akan menjadi`Initializing`. Setelah pekerjaan pelabelan aktif, status berubah`InProgress` dan Anda dapat mulai menggunakan opsi waktu nyata untuk mengirimkan objek data tambahan untuk pelabelan.

Kirim Objek Data secara Real Time:

- Kirim objek data menggunakan pesan Amazon SNS — Anda dapat mengirim objek data baru Ground Truth ke label dengan mengirimkan pesan Amazon SNS. Anda akan mengirim pesan ini ke topik masukan Amazon SNS yang Anda buat dan tentukan saat Anda membuat pekerjaan pelabelan streaming Anda. Untuk informasi selengkapnya, lihat Mengirim Objek Data Menggunakan Amazon SNS.
- Kirim objek data dengan menempatkannya di bucket Amazon S3 — Setiap kali Anda menambahkan objek data baru ke bucket Amazon S3, Anda dapat meminta Ground Truth untuk memproses objek tersebut untuk diberi label. Untuk melakukan ini, Anda menambahkan pemberitahuan peristiwa ke bucket sehingga memberi tahu topik masukan Amazon SNS Anda setiap kali objek baru ditambahkan ke (atau dibuat dalam) bucket tersebut. Untuk informasi selengkapnya, lihat Mengirim Objek Data menggunakan Amazon S3. Opsi ini tidak tersedia untuk pekerjaan pelabelan berbasis teks seperti klasifikasi teks dan pengenalan entitas bernama.

> **Important**
>
> Jika Anda menggunakan konfigurasi Amazon S3, jangan gunakan lokasi Amazon S3 yang sama untuk konfigurasi data input dan data keluaran Anda. Anda menentukan awalan S3 untuk data keluaran Anda saat membuat pekerjaan pelabelan.

Mengirim Objek Data Menggunakan Amazon SNS

Anda dapat mengirim objek data ke tugas pelabelan streaming menggunakan Amazon Simple Notification Service (Amazon SNS). Amazon SNS adalah layanan web yang mengoordinasikan dan mengelola penyampaian pesan ke titik akhir (misalnya, alamat atauAWS Lambda fungsi email). Topik Amazon SNS bertindak sebagai saluran komunikasi antara dua titik akhir atau lebih. Anda menggunakan Amazon SNS untuk mengirim, atau mempublikasikan, objek data baru ke topik yang ditentukan dalam `CreateLabelingJob`parameter`SnsTopicArn` di`InputConfig`. Format pesan ini sama dengan satu baris dari file manifes masukan.

Misalnya, Anda dapat mengirim sepotong teks ke pekerjaan pelabelan klasifikasi teks aktif dengan menerbitkannya ke topik masukan Anda. Pesan yang Anda publikasikan mungkin serupa dengan yang berikut ini:

```
{"source": "Lorem ipsum dolor sit amet"}
```

Untuk mengirim objek gambar baru ke pekerjaan pelabelan klasifikasi gambar, pesan Anda mungkin terlihat mirip dengan yang berikut:

```
{"source-ref": "s3://awsexamplebucket/example-image.jpg"}
```

> **Note**
>
> Anda juga dapat menyertakan ID deduplikasi kustom dan kunci deduplikasi dalam pesan Amazon SNS Anda. Untuk mempelajari selengkapnya, lihat Penanganan Pesan Duplikat.

Ketika Ground Truth membuat pekerjaan pelabelan streaming Anda, itu berlangganan topik masukan Amazon SNS Anda.

Mengirim Objek Data menggunakan Amazon S3

Anda dapat mengirim satu atau beberapa objek data baru ke pekerjaan pelabelan streaming dengan menempatkannya di bucket Amazon S3 yang dikonfigurasi dengan pemberitahuan peristiwa Amazon SNS. Anda dapat menyiapkan acara untuk memberi tahu topik masukan Amazon SNS Anda kapan saja objek baru dibuat di bucket Anda. Anda harus menentukan topik masukan Amazon SNS yang sama ini dalam `CreateLabelingJob`parameter`SnsTopicArn` di`InputConfig`.

Kapan pun Anda mengonfigurasi bucket Amazon S3 untuk mengirim notifikasi ke Amazon SNS, Ground Truth akan menerbitkan peristiwa pengujian`"s3:TestEvent"`,, untuk memastikan bahwa topik tersebut ada dan bahwa pemilik bucket Amazon S3 yang ditentukan memiliki izin untuk mempublikasikan ke topik yang ditentukan. Dianjurkan agar Anda mengatur koneksi Amazon S3 Anda dengan Amazon SNS sebelum memulai pekerjaan pelabelan streaming. Jika tidak, peristiwa pengujian ini dapat mendaftar sebagai objek data dan dikirim ke Ground Truth untuk diberi label.

> **Important**
>
> Jika Anda menggunakan konfigurasi Amazon S3, jangan gunakan lokasi Amazon S3 yang sama untuk konfigurasi data input dan data keluaran Anda. Anda menentukan awalan S3 untuk data keluaran Anda saat membuat pekerjaan pelabelan.
> Untuk pekerjaan pelabelan berbasis gambar, Ground Truth mengharuskan semua bucket S3 untuk memiliki kebijakan CORS yang terpasang. Untuk mempelajari selengkapnya, lihat Persyaratan Izin.

Setelah Anda mengonfigurasi bucket Amazon S3 dan membuat pekerjaan pelabelan, Anda dapat menambahkan objek ke bucket dan Ground Truth mengirimkan objek tersebut ke pekerja atau menempatkannya di antrean Amazon SQS Anda.

Untuk mempelajari selengkapnya, lihat Menyiapkan Pemberitahuan Peristiwa Amazon S3 Bucket.

> **Important**
>
> Opsi ini tidak tersedia untuk pekerjaan pelabelan berbasis teks seperti klasifikasi teks dan pengenalan entitas bernama.

Mengelola Permintaan Pelabelan dengan Antrean Amazon SQS

Ketika Ground Truth menciptakan pekerjaan pelabelan streaming Anda, itu menciptakan antrian Amazon SQS diAWS akun yang digunakan untuk membuat pekerjaan pelabelan. Nama antrian*labeling_job_name* adalah`GroundTruth-`*labeling_job_name* di mana nama pekerjaan pelabelan Anda, dalam huruf kecil. Saat Anda mengirim objek data ke pekerjaan pelabelan Anda, Ground Truth mengirimkan objek data langsung ke pekerja atau menempatkan tugas dalam antrian Anda untuk diproses di lain waktu. Jika objek data tidak dikirim ke pekerja setelah 14 hari, objek tersebut akan kedaluwarsa dan dihapus dari antrian. Anda dapat menyiapkan alarm di Amazon SQS untuk mendeteksi kapan objek kedaluwarsa dan menggunakan mekanisme ini untuk mengontrol volume objek yang Anda kirim ke tugas pelabelan Anda.

> **Important**
>
> Memodifikasi, menghapus, atau mengirim objek langsung ke antrean Amazon SQS yang terkait dengan pekerjaan pelabelan streaming Anda dapat menyebabkan kegagalan pekerjaan.

Menerima Data Keluaran dari Job Pelabelan Streaming

Bucket keluaran Amazon S3 Anda diperbarui secara berkala dengan data keluaran baru dari pekerjaan pelabelan streaming Anda.

Secara opsional, Anda dapat menentukan topik keluaran Amazon SNS. Setiap kali pekerja mengirimkan objek berlabel, pemberitahuan dengan data keluaran dikirim ke topik itu. Anda dapat berlangganan titik akhir ke topik keluaran SNS Anda untuk menerima pemberitahuan atau memicu peristiwa saat Anda menerima data keluaran dari tugas pelabelan. Gunakan topik keluaran Amazon SNS jika Anda ingin melakukan real time chaining ke pekerjaan streaming lain dan menerima notifikasi Amazon SNS setiap kali objek data dikirimkan oleh pekerja.

Untuk mempelajari selengkapnya, lihat Berlangganan Endpoint ke Topik Output Amazon SNS Anda.

Penanganan Pesan Duplikat

Untuk objek data yang dikirim secara real time, Ground Truth menjamin idempotency dengan memastikan setiap objek unik hanya dikirim untuk diberi label sekali, bahkan jika pesan input yang merujuk ke objek itu diterima beberapa kali (pesan duplikat). Untuk melakukan ini, setiap objek data yang dikirim ke pekerjaan pelabelan streaming diberi ID deduplikasi, yang diidentifikasi dengan kunci deduplikasi.

Jika Anda mengirim permintaan untuk memberi label objek data secara langsung melalui topik input Amazon SNS Anda menggunakan pesan Amazon SNS, Anda dapat memilih kunci deduplikasi kustom dan ID deduplikasi untuk objek Anda. Untuk informasi selengkapnya, lihat Menentukan Kunci Deduplikasi dan ID dalam Pesan Amazon SNS.

Jika Anda tidak memberikan kunci deduplikasi Anda sendiri, atau jika Anda menggunakan konfigurasi Amazon S3 untuk mengirim objek data ke tugas pelabelan Anda, Ground Truth menggunakan salah satu dari berikut ini untuk ID deduplikasi:

- Untuk pesan yang dikirim langsung ke topik masukan Amazon SNS Anda, Ground Truth menggunakan ID pesan SNS.

- Untuk pesan yang berasal dari konfigurasi Amazon S3, Ground Truth membuat ID deduplikasi dengan menggabungkan URI Amazon S3 objek dengan token sequencer dalam pesan.

Menentukan Kunci Deduplikasi dan ID dalam Pesan Amazon SNS

Saat Anda mengirim objek data ke pekerjaan pelabelan streaming menggunakan pesan Amazon SNS, Anda memiliki opsi untuk menentukan kunci deduplikasi dan ID deduplikasi dengan salah satu cara berikut. Dalam semua skenario ini, identifikasi kunci deduplikasi Anda dengan`dataset-objectid-attribute-name`.

Bawa Kunci dan ID Deduplikasi Anda Sendiri

Buat kunci deduplikasi dan ID deduplikasi Anda sendiri dengan mengonfigurasi pesan Amazon SNS Anda sebagai berikut. Ganti*`byo-key`* dengan kunci Anda dan*`UniqueId`* dengan ID deduplikasi untuk objek data tersebut.

```
{
    "source-ref":"s3://bucket/prefix/object1",
    "dataset-objectid-attribute-name":"byo-key",
    "byo-key":"UniqueId"
}
```

Kunci deduplikasi Anda dapat terdiri dari hingga 140 karakter. Pola yang didukung meliputi:`"^[$a-zA-Z0-9](-*[a-zA-Z0-9])*"`.

ID deduplikasi Anda dapat terdiri dari hingga 1.024 karakter. Pola yang didukung meliputi:`^(https|s3)://([^/]+)/?(.*)$`.

Gunakan Kunci yang Ada untuk Kunci Deduplikasi Anda

Anda dapat menggunakan kunci yang ada dalam pesan Anda sebagai kunci deduplikasi. Ketika Anda melakukan ini, nilai yang terkait dengan kunci yang digunakan untuk ID deduplikasi.

Misalnya, Anda dapat menentukan gunakan`source-ref` kunci sebagai kunci deduplikasi Anda dengan memformat pesan Anda sebagai berikut:

```
{
    "source-ref":"s3://bucket/prefix/object1",
    "dataset-objectid-attribute-name":"source-ref"
}
```

Dalam contoh ini, Ground Truth menggunakan`"s3://bucket/prefix/object1"` untuk id deduplikasi.

Temukan Kunci Deduplikasi dan ID di Data Output Anda

Anda dapat melihat kunci deduplikasi dan ID dalam data output Anda. Kunci deduplikasi diidentifikasi oleh`dataset-objectid-attribute-name`.

Bila Anda menggunakan kunci deduplikasi kustom Anda sendiri, output Anda berisi sesuatu yang mirip dengan berikut:

```
"dataset-objectid-attribute-name": "byo-key",
"byo-key": "UniqueId",
```

Bila Anda tidak menentukan kunci, Anda dapat menemukan ID deduplikasi yang Ground Truth ditugaskan ke objek data Anda sebagai berikut. `$`*`label-attribute-name`*`-object-id`Parameter mengidentifikasi ID deduplikasi Anda.

```
{
    "source-ref":"s3://bucket/prefix/object1",
    "dataset-objectid-attribute-name":"$label-attribute-name-object-id"
    "label-attribute-name" :0,
    "label-attribute-name-metadata": {...},
    "$label-attribute-name-object-id":"<service-generated-key>"
}
```

Untuk*`<service-generated-key>`*, jika objek data datang melalui konfigurasi Amazon S3, Ground Truth menambahkan nilai unik yang digunakan oleh layanan dan memancarkan bidang baru yang dikunci`$`*`sequencer`* yang menunjukkan sequencer Amazon S3 yang digunakan. Jika objek diumpankan ke SNS langsung, Ground Truth menggunakan ID pesan SNS.

> **Note**
>
> Jangan gunakan`$` karakter dalam nama atribut label Anda.

Kuota

Set data input yang digunakan dalam pekerjaan pelabelan segmentasi semantik memiliki kuota 20.000 item. Untuk semua jenis pekerjaan pelabelan lainnya, kuota ukuran set data adalah 100.000

item. Untuk meminta peningkatan kuota untuk pekerjaan pelabelan selain pekerjaan segmentasi semantik, tinjau prosedur dalam AWSService Quotas untuk meminta peningkatan kuota.

Masukan data gambar untuk pekerjaan pelabelan pembelajaran aktif dan non-aktif tidak boleh melebihi kuota ukuran dan resolusi. Pembelajaran aktif mengacu pada pekerjaan pelabelan yang menggunakan pelabelan data otomatis. Pembelajaran non-aktif mengacu pada pekerjaan pelabelan yang tidak menggunakan pelabelan data otomatis.

Kuota tambahan berlaku untuk kategori label untuk semua jenis tugas, dan untuk data input dan atribut kategori pelabelan untuk jenis tugas 3D point cloud dan frame video.

Kuota

File input tidak dapat melebihi kuota ukuran berikut untuk pekerjaan pelabelan pembelajaran aktif dan non-aktif. Tidak ada kuota ukuran file input untuk video yang digunakan dalam pekerjaan pelabelan klasifikasi video.

Jenis Job pelabelan	Kuota
Klasifikasi gambar	40 MB
Kotak pembatas (deteksi objek)	40 MB
Segmentasi semantik	40 MB
Penyesuaian label kotak pembatas (deteksi objek)	40 MB
Penyesuaian label segmentasi semantik	40 MB
Verifikasi label kotak pembatas (deteksi objek)	40 MB
Verifikasi label segmentasi semantik	40 MB

Input Kuota Resolusi Gambar

Resolusi file gambar mengacu pada jumlah piksel dalam gambar, dan menentukan jumlah detail yang dipegang gambar. Kuota resolusi gambar berbeda tergantung pada jenis pekerjaan pelabelan dan algoritma SageMaker bawaan yang digunakan. Tabel berikut mencantumkan kuota resolusi untuk gambar yang digunakan dalam pekerjaan pelabelan pembelajaran aktif dan non-aktif.

Jenis Job pelabelan	Resolusi Kuota - Pembelajaran Non Aktif	Resolusi Kuota - Pembelajaran Aktif
Klasifikasi gambar	100 juta piksel	3840 x 2160 piksel (4 K)
Kotak pembatas (deteksi objek)	100 juta piksel	3840 x 2160 piksel (4 K)
Segmentasi semantik	100 juta piksel	1920 x 1080 piksel (1080 p)
Penyesuaian label deteksi objek	100 juta piksel	3840 x 2160 piksel (4 K)
Penyesuaian label segmentasi semantik	100 juta piksel	1920 x 1080 piksel (1080 p)
Verifikasi label deteksi objek	100 juta piksel	Tidak tersedia
Verifikasi label segmentasi semantik	100 juta piksel	Tidak tersedia

Kuota Kategori Label

Setiap jenis tugas tugas pelabelan memiliki kuota untuk jumlah kategori label yang dapat Anda tentukan. Pekerja memilih kategori label untuk membuat anotasi. Misalnya, Anda dapat menentukan kategori label mobil, pejalan kaki, dan biker saat membuat pekerjaan pelabelan kotak pembatas dan pekerja akan memilih kategori mobil sebelum menggambar kotak pembatas di sekitar mobil.

> **Important**
>
> Nama kategori label tidak boleh melebihi 256 karakter.
> Semua kategori label harus unik. Anda tidak dapat menentukan kategori label duplikat.

Batas kategori label berikut berlaku untuk pekerjaan pelabelan. Kuota untuk kategori label bergantung pada apakah Anda menggunakan operasi SageMaker API`CreateLabelingJob` atau konsol untuk membuat pekerjaan pelabelan.

Jenis Job pelabelan	Label Kategori Kuota - API	Label Kategori Kuota - Console
Klasifikasi gambar (Multi-label)	50	50
Klasifikasi gambar (Label tunggal)	Tidak terbatas.	30
Kotak pembatas (deteksi objek)	50	50
Verifikasi label	Tidak terbatas.	30
Segmentasi semantik (dengan pembelajaran aktif)	20	10
Segmentasi semantik (tanpa pembelajaran aktif)	Tidak terbatas.	10
Pengenalan entitas bernama	Tidak terbatas.	30
Klasifikasi teks (Multi-label)	50	50
Klasifikasi teks (Label tunggal)	Tidak terbatas.	30
Klasifikasi Video	30	30
Deteksi objek bingkai video	30	30
Pelacakan objek bingkai video	30	30
Deteksi objek awan titik 3D	30	30
Pelacakan objek awan titik 3D	30	30
Segmentasi semantik awan titik 3D	30	30

Kuota Job Pelabelan 3D Point Cloud dan Bingkai Video

Kuota berikut berlaku untuk data input pekerjaan pelabelan cloud titik 3D dan bingkai video.

Jenis Job pelabelan	Kuota
Deteksi objek bingkai video	2.000 frame video (gambar) per urutan
Deteksi objek bingkai video	10 urutan bingkai video per file manifes
Pelacakan objek bingkai video	2.000 frame video (gambar) per urutan
Pelacakan objek bingkai video	10 urutan bingkai video per file manifes
Deteksi objek awan titik 3D	100.000 titik cloud frame per pekerjaan pelabelan
Pelacakan objek awan titik 3D	Urutan bingkai awan 100.000 titik per pekerjaan pelabelan
Pelacakan objek awan titik 3D	500 titik cloud frame di setiap file urutan

Saat membuat bingkai video atau pekerjaan pelabelan awan titik 3D, Anda dapat menambahkan satu atau beberapa atribut kategori label ke setiap kategori label yang Anda tentukan agar pekerja memberikan informasi lebih lanjut tentang anotasi.

Setiap atribut kategori label memiliki atribut kategori label tunggal`name`, dan daftar satu atau lebih opsi (nilai) untuk dipilih. Untuk mempelajari lebih lanjut, lihatAntarmuka Pengguna Pekerja (UI) pekerjaan pelabelan cloud titik 3D danAntarmuka Pengguna Pekerja (UI) untuk pekerjaan pelabelan bingkai video.

Kuota berikut berlaku untuk jumlah kategori label atribut nama dan nilai yang dapat Anda tentukan untuk pekerjaan pelabelan.

Jenis Job pelabelan	Label Kategori Atribut (nama) Kuota	Label Kategori Atribut Nilai Kuota
Deteksi objek bingkai video	10	10

Jenis Job pelabelan	Label Kategori Atribut (nama) Kuota	Label Kategori Atribut Nilai Kuota
Pelacakan objek bingkai video	10	10
Deteksi objek awan titik 3D	10	10
Pelacakan objek awan titik 3D	10	10
Segmentasi semantik awan titik 3D	10	10

Filter dan Pilih Data untuk Pelabelan

Anda dapat menggunakan SageMaker konsol Amazon untuk memilih sebagian dari kumpulan data Anda untuk diberi label. Data harus disimpan dalam bucket Amazon S3. Anda memiliki tiga pilihan:

- Gunakan set data lengkap.
- Pilih sampel set data yang dipilih secara acak.
- Tentukan subset dari kumpulan data menggunakan kueri.

Opsi berikut tersedia di bagian Pelabelan pekerjaanSageMaker konsol setelah memilih Buat pekerjaan pelabelan. Untuk mempelajari cara membuat pekerjaan pelabelan di konsol, lihatMulai. Untuk mengonfigurasi kumpulan data yang Anda gunakan untuk pelabelan, di bagian Ringkasan Job, pilih Konfigurasi tambahan.

Gunakan Set Data Lengkap

Saat Anda memilih untuk menggunakan set data Penuh, Anda harus menyediakan file manifes untuk objek data Anda. Anda dapat menyediakan jalur bucket Amazon S3 yang berisi file manifes atau menggunakan SageMaker konsol untuk membuat file. Untuk mempelajari cara membuat file manifes menggunakan konsol, lihatPengaturan Data Otomatis.

Pilih Sampel Acak

Bila Anda ingin memberi label subset acak dari data Anda, pilih Sampel acak. Set data disimpan dalam bucket Amazon S3 yang ditentukan di bidang lokasi set data Input.

Setelah Anda menentukan persentase objek data yang ingin Anda sertakan dalam sampel, pilih Buat subset. SageMaker secara acak memilih objek data untuk pekerjaan pelabelan Anda. Setelah objek dipilih, pilih Gunakan subset ini.

SageMaker membuat file manifes untuk objek data yang dipilih. Ini juga memodifikasi nilai di bidang lokasi set data Input untuk menunjuk ke file manifes baru.

Tentukan Subset

Anda dapat menentukan subset objek data Anda menggunakan`SELECT` kueri Amazon S3 pada nama file objek.

`SELECT`Pernyataan query SQL didefinisikan untuk Anda. Anda memberikan`WHERE` klausa untuk menentukan objek data yang harus dikembalikan.

Untuk informasi selengkapnya tentang`SELECT` pernyataan Amazon S3, lihat Memilih Konten dari Objek.

Memilih Buat subset untuk memulai pemilihan, lalu pilih Gunakan subset ini untuk menggunakan data yang dipilih.

SageMaker membuat file manifes untuk objek data yang dipilih. Ini juga memperbarui nilai di bidang lokasi set data Input untuk menunjuk ke file manifes baru.

Data Input Awan Titik 3D

Untuk membuat tugas pelabelan cloud titik 3D, Anda harus membuat file manifes masukan. Gunakan topik ini untuk mempelajari persyaratan pemformatan file manifes masukan untuk setiap jenis tugas. Untuk mempelajari tentang format data input mentah yang diterima Ground Truth untuk pekerjaan pelabelan cloud titik 3D, lihat bagianFormat Data 3D Mentah yang Diterima.

Gunakanpelabelan jenis tugas pekerjaanuntuk memilih topik padaMembuat File Manifes Input untuk Job Pelabelan Cloud Titik 3Duntuk mempelajari tentang persyaratan pemformatan untuk setiap baris file manifes masukan Anda.

Topik

- Format Data 3D Mentah yang Diterima
- Membuat File Manifes Input untuk Job Pelabelan Cloud Titik 3D
- Memahami Sistem Koordinat dan Sensor Fusion

Format Data 3D Mentah yang Diterima

Ground Truth menggunakan data cloud titik 3D Anda untuk membuat adegan 3D yang dianotasi pekerja. Bagian ini menjelaskan format data mentah yang diterima untuk data cloud titik dan data fusi sensor untuk bingkai cloud titik. Untuk mempelajari cara membuat file manifes masukan untuk menghubungkan file data input mentah Anda dengan Ground Truth, lihatMembuat File Manifes Input untuk Job Pelabelan Cloud Titik 3D.

Untuk setiap frame, Ground Truth mendukung file Compact Binary Pack Format (.bin) dan ASCII (.txt). File-file ini berisi informasi tentang lokasi (`x`,`y`, dan`z`koordinat) dari semua titik yang membentuk bingkai itu, dan, secara opsional, informasi tentang warna piksel setiap titik untuk awan titik berwarna. Saat membuat file manifes masukan pekerjaan pelabelan cloud titik 3D, Anda dapat menentukan format data mentah di`format`parameter.

Tabel berikut mencantumkan elemen yang Ground Truth mendukung dalam file frame titik awan untuk menggambarkan poin individu.

Simbol	Nilai
`x`	Koordinat x titik.
`y`	Koordinat y dari titik.
`z`	Koordinat z titik.
`i`	Intensitas intinya.
`r`	Komponen saluran warna merah. Nilai 8-bit (0-255).
`g`	Komponen saluran warna hijau. Nilai 8-bit (0-255)
`b`	Komponen saluran warna biru. Nilai 8-bit (0-255)

Ground Truth mengasumsikan hal berikut tentang data masukan Anda:

- Semua koordinat posisi (x, y, z) berada dalam meter.

- Semua judul pose (qx, qy, qz, qw) diukur dalam SpasialKuaternion.

Format Paket Biner Ringkas

Compact Binary Pack Format mewakili titik awan sebagai set memerintahkan aliran poin. Setiap titik dalam aliran adalah paket biner memerintahkan nilai float 4-byte dalam beberapa varian dari bentuk`xyzirgb`. Klaster`x`,`y`, dan`z`elemen diperlukan dan informasi tambahan tentang pixel yang dapat dimasukkan dalam berbagai cara menggunakan`i`,`r`,`g`, dan`b`.

Untuk menggunakan file biner untuk memasukkan data frame cloud titik ke pekerjaan pelabelan cloud titik 3D Ground Truth, masukkan`binary/`di dalam`format`parameter untuk file manifes masukan Anda dan gantidengan urutan elemen di setiap paket biner. Misalnya, Anda dapat memasukkan salah satu langkah berikut untuk`format`parameter.

- `binary/xyzi`- Bila Anda menggunakan format ini, aliran elemen titik Anda akan berada dalam urutan berikut:`x1y1z1i1x2y2z2i2...`
- `binary/xyzrgb`- Bila Anda menggunakan format ini, aliran elemen titik Anda akan berada dalam urutan berikut:`x1y1z1r1g1b1x2y2z2r2g2b2...`
- `binary/xyzirgb`- Bila Anda menggunakan format ini, aliran elemen titik Anda akan berada dalam urutan berikut:`x1y1z1i1r1g1b1x2y2z2i2r2g2b2...`

Ketika Anda menggunakan file biner untuk data frame titik awan Anda, jika Anda tidak memasukkan nilai untuk`format`, format paket default`binary/xyzi`digunakan.

Format ASCII

Format ASCII menggunakan file teks untuk mewakili titik awan, di mana setiap baris dalam file cloud titik ASCII mewakili satu titik. Setiap titik adalah baris file teks dan berisi nilai dipisahkan ruang putih, yang masing-masing adalah 4-byte mengambang nilai ASCII. Klaster`x`,`y`, dan`z`elemen diperlukan untuk setiap titik dan informasi tambahan tentang titik itu dapat dimasukkan dalam berbagai cara menggunakan`i`,`r`,`g`, dan`b`.

Untuk menggunakan file teks untuk memasukkan data bingkai awan titik ke pekerjaan pelabelan awan titik 3D Ground Truth, masukkan`text/`di dalam`format`parameter untuk file manifes masukan Anda dan gantidengan urutan elemen titik pada setiap baris.

Misalnya, jika Anda masuk`text/xyzi`untuk`format`, file teks Anda untuk setiap frame cloud titik akan terlihat seperti berikut ini:

```
x1 y1 z1 i1
x2 y2 z2 i2
...
...
```

Jika Anda masuk`text/xyzrgb`, file teks Anda akan terlihat seperti berikut ini:

```
x1 y1 z1 r1 g1 b1
x2 y2 z2 r2 g2 b1
...
...
```

Ketika Anda menggunakan file teks untuk data frame titik awan Anda, jika Anda tidak memasukkan nilai untuk`format`, format default`text/xyzi`akan digunakan.

Batas Resolusi Point Cloud

Ground Truth tidak memiliki batas resolusi untuk frame cloud titik 3D. Namun, kami menyarankan Anda membatasi setiap titik cloud frame hingga 500K poin untuk kinerja optimal. Ketika Ground Truth merender visualisasi awan titik 3D, itu harus dapat dilihat di komputer pekerja Anda, yang tergantung pada perangkat keras komputer pekerja. Bingkai cloud titik yang lebih besar dari 1 juta poin mungkin tidak dirender pada mesin standar, atau mungkin membutuhkan waktu terlalu lama untuk dimuat.

Membuat File Manifes Input untuk Job Pelabelan Cloud Titik 3D

Saat membuat tugas pelabelan, Anda menyediakan file manifes masukan di mana setiap baris manifes menjelaskan unit tugas yang akan diselesaikan oleh anotator. Format file manifes masukan Anda bergantung pada jenis tugas Anda.

- Jika Anda membuat awan titik 3DDeteksi objekatausegmentasi semantikpekerjaan pelabelan, setiap baris dalam file manifes masukan Anda berisi informasi tentang satu bingkai awan titik 3D. Ini disebuttitik manifes masukan bingkai awan. Untuk mempelajari selengkapnya, lihat Membuat File Manifes Input Point Cloud Frame.
- Jika Anda membuat awan titik 3Dpelacakan objekpekerjaan pelabelan, setiap baris file manifes masukan Anda berisi urutan frame cloud titik 3D dan data terkait. Ini disebutmanifes masukan urutan awan titik. Untuk mempelajari selengkapnya, lihat Membuat Manifes Input Urutan Point Cloud.

Membuat File Manifes Input Point Cloud Frame

Manifes adalah file yang dikodekan UTF-8 di mana setiap baris adalah objek JSON yang lengkap dan valid. Setiap baris dibatasi oleh jeda garis standar,\ n atau\ r\ n. Karena setiap baris harus berupa objek JSON yang valid, Anda tidak dapat memiliki karakter line break yang tidak lolos. Dalam file manifes input single-frame, setiap baris dalam manifes berisi data untuk frame cloud titik tunggal. Data titik cloud frame dapat disimpan dalam format biner atau ASCII (lihatFormat Data 3D Mentah yang Diterima). Ini adalah format file manifes yang diperlukan untuk deteksi objek awan titik 3D dan segmentasi semantik. Secara opsional, Anda juga dapat menyediakan data fusi sensor kamera untuk setiap bingkai cloud titik.

Ground Truth mendukung titik awan dan kamera video sensor fusi disistem koordinat duniauntuk semua modalitas. Jika Anda dapat memperoleh ekstrinsik sensor 3D Anda (seperti ekstrinsik LiDAR), kami sarankan Anda mengubah bingkai cloud titik 3D ke dalam sistem koordinat dunia menggunakan ekstrinsik. Untuk informasi selengkapnya, lihat Sensor Fusion.

Namun, jika Anda tidak dapat memperoleh titik awan dalam sistem koordinat dunia, Anda dapat memberikan koordinat dalam sistem koordinat asli tempat data ditangkap. Jika Anda menyediakan data kamera untuk fusi sensor, disarankan agar Anda memberikan sensor LiDAR dan pose kamera di sistem koordinat dunia.

Untuk membuat file manifes input single-frame, Anda akan mengidentifikasi lokasi setiap frame cloud titik yang Anda inginkan untuk diberi label oleh pekerja menggunakan`source-ref`kunci. Selain itu, Anda harus menggunakan`source-ref-metadata`kunci untuk mengidentifikasi format dataset Anda, stempel waktu untuk frame itu, dan, secara opsional, data fusi sensor dan gambar kamera video.

Contoh berikut menunjukkan sintaks yang digunakan untuk file manifes masukan untuk tugas pelabelan cloud titik bingkai tunggal. Contohnya mencakup dua titik cloud frame. Untuk detail tentang tiap parameter, lihat tabel berikut contoh ini.

> **Important**
>
> Setiap baris dalam file manifes masukan Anda harus masukGaris JSONformat. Blok kode berikut menunjukkan file manifes masukan dengan dua objek JSON. Setiap objek JSON digunakan untuk menunjuk ke dan memberikan rincian tentang frame cloud titik tunggal. Objek JSON telah diperluas agar mudah dibaca, tetapi Anda harus meminimalkan setiap objek JSON agar sesuai dengan satu baris saat membuat file manifes masukan. Contoh disediakan di bawah blok kode ini.

```
{
    "source-ref": "s3://awsexamplebucket/examplefolder/frame1.bin",
    "source-ref-metadata":{
        "format": "binary/xyzi",
        "unix-timestamp": 1566861644.759115,
        "ego-vehicle-pose":{
            "position": {
                "x": -2.7161461413869947,
                "y": 116.25822288149078,
                "z": 1.8348751887989483
            },
            "heading": {
                "qx": -0.02111296123795955,
                "qy": -0.006495469416730261,
                "qz": -0.008024565904865688,
                "qw": 0.9997181192298087
            }
        },
        "prefix": "s3://awsexamplebucket/lidar_singleframe_dataset/someprefix/",
        "images": [
        {
            "image-path": "images/frame300.bin_camera0.jpg",
            "unix-timestamp": 1566861644.759115,
            "fx": 847.7962624528487,
            "fy": 850.0340893791985,
            "cx": 576.2129134707038,
            "cy": 317.2423573573745,
            "k1": 0,
            "k2": 0,
            "k3": 0,
            "k4": 0,
            "p1": 0,
            "p2": 0,
            "skew": 0,
            "position": {
                "x": -2.2722515189268138,
                "y": 116.86003310568965,
                "z": 1.454614668542299
            },
            "heading": {
                "qx": 0.7594754093069037,
                "qy": 0.02181790885672969,
                "qz": -0.02461725233103356,
```

```
                "qw": -0.6496916273040025
            },
            "camera-model": "pinhole"
        }]
    }
}
{
    "source-ref": "s3://awsexamplebucket/examplefolder/frame2.bin",
    "source-ref-metadata":{
        "format": "binary/xyzi",
        "unix-timestamp": 1566861632.759133,
        "ego-vehicle-pose":{
            "position": {
                "x": -2.7161461413869947,
                "y": 116.25822288149078,
                "z": 1.8348751887989483
            },
            "heading": {
                "qx": -0.02111296123795955,
                "qy": -0.006495469416730261,
                "qz": -0.008024565904865688,
                "qw": 0.9997181192298087
            }
        },
        "prefix": "s3://awsexamplebucket/lidar_singleframe_dataset/someprefix/",
        "images": [
        {
            "image-path": "images/frame300.bin_camera0.jpg",
            "unix-timestamp": 1566861644.759115,
            "fx": 847.7962624528487,
            "fy": 850.0340893791985,
            "cx": 576.2129134707038,
            "cy": 317.2423573573745,
            "k1": 0,
            "k2": 0,
            "k3": 0,
            "k4": 0,
            "p1": 0,
            "p2": 0,
            "skew": 0,
            "position": {
                "x": -2.2722515189268138,
                "y": 116.86003310568965,
                "z": 1.454614668542299
```

```
            },
            "heading": {
                "qx": 0.7594754093069037,
                "qy": 0.02181790885672969,
                "qz": -0.02461725233103356,
                "qw": -0.6496916273040025
            },
            "camera-model": "pinhole"
        }]
    }
}
```

Saat membuat file manifes masukan, Anda harus menciutkan objek JSON agar sesuai dengan satu baris. Misalnya, blok kode di atas akan muncul sebagai berikut dalam file manifes masukan:

```
{"source-ref":"s3://awsexamplebucket/examplefolder/frame1.bin","source-ref-metadata":
{"format":"binary/xyzi","unix-timestamp":1566861644.759115,"ego-vehicle-pose":
{"position":
{"x":-2.7161461413869947,"y":116.25822288149078,"z":1.8348751887989483},"heading":
{"qx":-0.02111296123795955,"qy":-0.006495469416730261,"qz":-0.008024565904865688,"qw":0.9997181
awsexamplebucket/lidar_singleframe_dataset/someprefix/","images":
[{"image-path":"images/frame300.bin_camera0.jpg","unix-
timestamp":1566861644.759115,"fx":847.7962624528487,"fy":850.0340893791985,"cx":576.21291347070
{"x":-2.2722515189268138,"y":116.86003310568965,"z":1.454614668542299},"heading":
{"qx":0.7594754093069037,"qy":0.02181790885672969,"qz":-0.02461725233103356,"qw":-0.64969162730
model":"pinhole"}]}}
{"source-ref":"s3://awsexamplebucket/examplefolder/frame2.bin","source-ref-metadata":
{"format":"binary/xyzi","unix-timestamp":1566861632.759133,"ego-vehicle-pose":
{"position":
{"x":-2.7161461413869947,"y":116.25822288149078,"z":1.8348751887989483},"heading":
{"qx":-0.02111296123795955,"qy":-0.006495469416730261,"qz":-0.008024565904865688,"qw":0.9997181
awsexamplebucket/lidar_singleframe_dataset/someprefix/","images":
[{"image-path":"images/frame300.bin_camera0.jpg","unix-
timestamp":1566861644.759115,"fx":847.7962624528487,"fy":850.0340893791985,"cx":576.21291347070
{"x":-2.2722515189268138,"y":116.86003310568965,"z":1.454614668542299},"heading":
{"qx":0.7594754093069037,"qy":0.02181790885672969,"qz":-0.02461725233103356,"qw":-0.64969162730
model":"pinhole"}]}}
```

Tabel berikut menunjukkan parameter yang dapat Anda sertakan dalam file manifes masukan Anda:

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`source-ref`	Ya	String Nilai string yang diterima: `s3://<bucket-name> /<folder-name> /point-cloud-frame-file`	Lokasi Amazon S3 dari bingkai cloud titik tunggal.
`source-ref-metadata`	Ya	Objek JSON Parameter yang diterima: `format`, `unix-timestamp` , `ego-vehicle-pose` , `position`, `prefix`, `images`	Gunakan parameter ini untuk menyertakan informasi tambahan tentang titik awan `disource-ref` , dan untuk menyediakan data kamera untuk fusi sensor.
`format`	Tidak	String Nilai string yang diterima:`"binary/xyz" ,"binary/xyzi" ,"binary/xyzrgb" ,"binary/xyzirgb" ,"text/xyz" ,"text/xyzi" ,"text/xyzrgb" ,"text/xyzirgb"` Nilai default:	Gunakan parameter ini untuk menentukan format data cloud titik Anda. Untuk informasi selengkapnya, lihat Format Data 3D Mentah yang Diterima.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
		Ketika file diidentifikasi disource-ref memiliki ekstensi. bin,binary/xyzi Ketika file diidentifikasi disource-ref memiliki ekstensi. txt,text/xyzi	
unix-timestamp	Ya	Jumlah Sebuah timestamp unix.	Stempel waktu unix adalah jumlah detik sejak 1 Januari 1970 hingga waktu UTC data dikumpulkan oleh sensor.
ego-vehicle-pose	Tidak	Objek JSON	Pose perangkat yang digunakan untuk mengumpulkan data titik awan. Untuk informasi selengkapnya tentang parameter ini, lihatSertakan Informasi Pose Kendaraan di Manifes Masukan Anda.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`prefix`	Tidak	String Nilai string yang diterima: `s3://<bucket-name> /<folder-name>/`	Lokasi di Amazon S3 di mana metadata Anda, seperti gambar kamera, tersimpan untuk bingkai ini. Prefiks harus diakhiri dengan garis miring ke depan:`/`.
`images`	Tidak	Daftar	Daftar parameter yang menggambarkan gambar kamera berwarna yang digunakan untuk fusi sensor. Anda dapat menyertakan hingga 8 gambar dalam daftar ini. Untuk informasi selengkapnya tentang parameter yang diperlukan untuk setiap gambar, lihatSertakan Data Kamera di Manifes Input Anda.

Sertakan Informasi Pose Kendaraan di Manifes Masukan Anda

Gunakan lokasi ego-kendaraan untuk memberikan informasi tentang lokasi kendaraan yang digunakan untuk menangkap data cloud titik. Ground Truth menggunakan informasi ini untuk menghitung LiDAR matriks ekstrinsik.

Ground Truth menggunakan matriks ekstrinsik untuk memproyeksikan label ke dan dari adegan 3D dan gambar 2D. Untuk informasi selengkapnya, lihat Sensor Fusion.

Tabel berikut memberikan informasi selengkapnya tentang`position`dan orientasi (`heading`) parameter yang diperlukan ketika Anda memberikan informasi ego-kendaraan.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`position`	Ya	Objek JSON Parameter yang dibutuhkan: `x`, `y`, dan `z`. Masukkan angka untuk parameter ini.	Vektor terjemahan kendaraan ego dalam sistem koordinat dunia.
`heading`	Ya	Objek JSON Parameter yang dibutuhkan: `qx`, `qy`, `qz`, dan `qw`. Masukkan angka untuk parameter ini.	Orientasi kerangka referensi perangkat atau sensor yang dipasang pada kendaraan yang merasakan sekitarnya, diukurkuaternion, (`qx`,`qy`,`qz`,`qw`) dalam sistem koordinat.

Sertakan Data Kamera di Manifes Input Anda

Jika Anda ingin menyertakan data kamera video dengan bingkai, gunakan parameter berikut untuk memberikan informasi tentang setiap gambar. KlasterWajibkolom di bawah ini berlaku ketika`images`parameter disertakan dalam berkas manifes masukan di bawah`source-ref-metadata`. Anda tidak diharuskan untuk menyertakan gambar dalam file manifes masukan Anda.

Jika Anda menyertakan gambar kamera, Anda harus menyertakan informasi tentang kamera`position`dan`heading`digunakan menangkap gambar dalam sistem koordinat dunia.

Jika gambar Anda terdistorsi, Ground Truth dapat secara otomatis mengubah mereka menggunakan informasi yang Anda berikan tentang gambar dalam file manifes masukan Anda, termasuk koefisien distorsi (`k1`,`k2`,`k3`,`k4`,`p1`,`p1`), model kamera dan matriks intrinsik kamera. Matriks intrinsik terdiri dari panjang fokus (`fx`,`fy`), dan titik utama (`cx`,`cy`). LihatMatriks intrinsik untuk mempelajari bagaimana Ground Truth menggunakan kamera intrinsik. Jika koefisien distorsi tidak disertakan, Ground Truth tidak akan merusak gambar.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`image-path`	Ya	String Contoh format: *<folder-name> /<imagefile.png>*	Lokasi relatif, di Amazon S3 file gambar Anda. Jalur relatif ini akan ditambahkan ke jalur yang Anda tentukan`prefix`.
`unix-timestamp`	Ya	Jumlah	Stempel waktu unix adalah jumlah detik sejak 1 Januari 1970 hingga waktu UTC data dikumpulkan oleh kamera.

`camera-model`	Tidak	String:
		
		Nilai yang Dapat diterima:
		
		`"pinhole"` , `"fisheye"`
		
		Default:
		
		`"pinhole"`
			Model kamera yang digunakan untuk menangkap gambar. Informasi ini digunakan untuk undistorsi gambar kamera.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`fx, fy`	Ya	Nomor	Panjang fokus kamera, di x (`fx`) dan y (`fy`) arah.
`cx, cy`	Ya	Nomor	x (`cx`) dan y (`cy`) koordinat titik utama.
`k1, k2, k3, k4`	Tidak	Jumlah	Koefisien distorsi radial. Didukung untuk keduanyafisheyedanlubang jarummodel kamera.
`p1, p2`	Tidak	Jumlah	Koefisien distorsi tangensial. Didukung untuklubang jarummodel kamera.
`skew`	Tidak	Jumlah	Parameter untuk mengukur kemiringan gambar.
`position`	Ya	Objek JSON Parameter yang dibutuhkan: `x`, `y`, dan `z`. Masukkan angka untuk parameter ini.	Lokasi atau asal kerangka referensi kamera yang dipasang pada kendaraan menangkap gambar.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`heading`	Ya	Objek JSON Parameter yang dibutuhkan: `qx`, `qy`, `qz`, dan `qw`. Masukkan angka untuk parameter ini.	Orientasi kerangka referensi kamera yang dipasang pada kendaraan menangkap gambar, diukur menggunak ankuaternion, (`qx`,`qy`,`qz`,`qw`), dalam sistem koordinat dunia.

Batas Bingkai Titik Cloud

Anda dapat menyertakan hingga 100.000 titik cloud frame dalam file manifes masukan Anda. Pekerjaan pelabelan awan titik 3D memiliki waktu pra-pemrosesan yang lebih lama daripada jenis tugas Ground Truth lainnya. Untuk informasi selengkapnya, lihat Waktu Pra-pemrosesan Job.

Membuat Manifes Input Urutan Point Cloud

Manifes adalah file yang dikodekan UTF-8 di mana setiap baris adalah objek JSON yang lengkap dan valid. Setiap baris dibatasi oleh jeda garis standar,\ n atau\ r\ n. Karena setiap baris harus berupa objek JSON yang valid, Anda tidak dapat memiliki karakter line break yang tidak lolos. Dalam file manifes input urutan awan titik, setiap baris dalam manifes berisi urutan frame titik awan. Data titik awan untuk setiap frame dalam urutan dapat disimpan dalam format biner atau ASCII. Untuk informasi selengkapnya, lihat Format Data 3D Mentah yang Diterima. Ini adalah format file manifes yang diperlukan untuk pelacakan objek cloud titik 3D. Secara opsional, Anda juga dapat memberikan atribut titik dan data fusi sensor kamera untuk setiap bingkai cloud titik. Saat membuat file manifes masukan urutan, Anda harus menyediakan data fusi sensor kamera LiDAR dan video disistem koordinat dunia.

Contoh berikut menunjukkan sintaks yang digunakan untuk file manifes masukan ketika setiap baris dalam manifes adalah file urutan. Setiap baris dalam file manifes masukan Anda harus masukGaris JSONformat.

```
{"source-ref": "s3://awsexamplebucket/example-folder/seq1.json"}
```

```
{"source-ref": "s3://awsexamplebucket/example-folder/seq2.json"}
```

Data untuk setiap urutan frame titik awan perlu disimpan dalam objek data JSON. Berikut ini adalah contoh format yang Anda gunakan untuk file urutan. Informasi tentang setiap frame disertakan sebagai objek JSON dan tercantum dalam`frames`daftar. Ini adalah contoh file urutan dengan dua titik file bingkai awan,`frame300.bin`dan`frame303.bin`. Klaster`...`digunakan untuk menunjukkan di mana Anda harus menyertakan informasi untuk frame tambahan. Tambahkan objek JSON untuk setiap frame dalam urutan.

Blok kode berikut termasuk objek JSON untuk file urutan tunggal. Objek JSON telah diperluas untuk dibaca.

```
{
  "seq-no": 1,
  "prefix": "s3://awsexamplebucket/example_lidar_sequence_dataset/seq1/",
  "number-of-frames": 100,
  "frames":[
    {
        "frame-no": 300,
        "unix-timestamp": 1566861644.759115,
        "frame": "example_lidar_frames/frame300.bin",
        "format": "binary/xyzi",
        "ego-vehicle-pose":{
            "position": {
                "x": -2.7161461413869947,
                "y": 116.25822288149078,
                "z": 1.8348751887989483
            },
            "heading": {
                "qx": -0.02111296123795955,
                "qy": -0.006495469416730261,
                "qz": -0.008024565904865688,
                "qw": 0.9997181192298087
            }
        },
        "images": [
        {
            "image-path": "example_images/frame300.bin_camera0.jpg",
            "unix-timestamp": 1566861644.759115,
            "fx": 847.7962624528487,
            "fy": 850.0340893791985,
            "cx": 576.2129134707038,
```

```
            "cy": 317.2423573573745,
            "k1": 0,
            "k2": 0,
            "k3": 0,
            "k4": 0,
            "p1": 0,
            "p2": 0,
            "skew": 0,
            "position": {
                "x": -2.2722515189268138,
                "y": 116.86003310568965,
                "z": 1.454614668542299
            },
            "heading": {
                "qx": 0.7594754093069037,
                "qy": 0.02181790885672969,
                "qz": -0.02461725233103356,
                "qw": -0.6496916273040025
            },
            "camera-model": "pinhole"
        }]
    },
    {
        "frame-no": 303,
        "unix-timestamp": 1566861644.759115,
        "frame": "example_lidar_frames/frame303.bin",
        "format": "text/xyzi",
        "ego-vehicle-pose":{...},
        "images":[{...}]
    },
     ...
  ]
}
```

Tabel berikut memberikan detail tentang parameter tingkat atas dari file urutan. Untuk informasi rinci tentang parameter yang diperlukan untuk frame individu dalam file urutan, lihatParameter untuk Frame Cloud Titik Individu.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`seq-no`	Ya	Bulat	Nomor urutan yang dipesan.
`prefix`	Ya	String Nilai yang Dapat diterima: `s3://<bucket-name> /<prefix>/`	Lokasi Amazon S3 di mana file urutan berada. Prefiks harus diakhiri dengan garis miring ke depan:`/`.
`number-of-frames`	Ya	Bulat	Jumlah total frame termasuk dalam file urutan. Jumlah ini harus sesuai dengan jumlah frame yang tercantum dalam`frames`parameter di baris berikutnya.
`frames`	Ya	Daftar objek JSON	Daftar data bingkai. Panjang daftar harus sama`number-of-frames`. Dalam UI pekerja, frame dalam urutan akan sama dengan urutan frame dalam array ini. Untuk detail tentang format tiap frame, lihatParameter untuk Frame Cloud Titik Individu.

Parameter untuk Frame Cloud Titik Individu

Tabel berikut menunjukkan parameter yang dapat Anda sertakan dalam file manifes input Anda.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`frame-no`	Tidak	Bulat	Nomor bingkai. Ini adalah identifie r opsional yang ditentukan oleh pelanggan untuk mengidentifikasi frame dalam urutan. ATRIBUT ini tidak digunakan oleh Ground Truth.
`unix-timestamp`	Ya	Jumlah	Stempel waktu unix adalah jumlah detik sejak 1 Januari 1970 hingga waktu UTC data dikumpulkan oleh sensor. Stempel waktu untuk setiap frame harus berbeda dan stempel waktu harus berurutan karena digunakan untuk interpolasi berbentuk kubus. Idealnya, ini harus menjadi stempel waktu nyata ketika data dikumpulk an. Jika ini tidak tersedia, Anda harus

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
			menggunakan urutan bertahap stempel waktu, di mana frame pertama dalam file urutan Anda sesuai dengan stempel waktu pertama dalam urutan.
`frame`	Ya	String Contoh format *<folder-name> /<sequence-file.json>*	Lokasi relatif, di Amazon S3 file urutan Anda. Jalur relatif ini akan ditambahkan ke jalur yang Anda tentukan`prefix`.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`format`	Tidak	String Nilai string yang diterima:`"binary/xyz" ,"binary/xyzi" ,"binary/xyzrgb" ,"binary/xyzirgb" ,"text/xyz" ,"text/xyzi" ,"text/xyzrgb" ,"text/xyzirgb"` Nilai default: Ketika file diidentifikasi di`source-ref` memiliki ekstensi. bin,`binary/xyzi` Ketika file diidentifikasi di`source-ref` memiliki ekstensi. txt,`text/xyzi`	Gunakan parameter ini untuk menentukan format data cloud titik Anda. Untuk informasi selengkapnya, lihat Format Data 3D Mentah yang Diterima.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`ego-vehicle-pose`	Tidak	Objek JSON	Pose perangkat yang digunakan untuk mengumpulkan data titik awan. Untuk informasi selengkapnya tentang parameter ini, lihatSertakan Informasi Pose Kendaraan di Manifes Masukan Anda.
`prefix`	Tidak	String Nilai string yang diterima: `s3://<bucket-name> /<folder-name>/`	Lokasi di Amazon S3 di mana metadata Anda, seperti gambar kamera, tersimpan untuk bingkai ini. Prefiks harus diakhiri dengan garis miring ke depan:`/`.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`images`	Tidak	Daftar	Parameter daftar yang menggambarkan gambar kamera warna yang digunakan untuk fusi sensor. Anda dapat menyertakan hingga 8 gambar dalam daftar ini. Untuk informasi selengkapnya tentang parameter yang diperlukan untuk setiap gambar, lihatSertakan Data Kamera di Manifes Input Anda.

Sertakan Informasi Pose Kendaraan di Manifes Masukan Anda

Gunakan lokasi ego-kendaraan untuk memberikan informasi tentang pose kendaraan yang digunakan untuk menangkap data cloud titik. Ground Truth menggunakan informasi ini untuk menghitung matriks ekstrinsik LiDAR.

Ground Truth menggunakan matriks ekstrinsik untuk memproyeksikan label ke dan dari adegan 3D dan gambar 2D. Untuk informasi selengkapnya, lihat Sensor Fusion.

Tabel berikut memberikan informasi selengkapnya tentang`position`dan orientasi (`heading`) parameter yang diperlukan ketika Anda memberikan informasi ego-kendaraan.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`position`	Ya	Objek JSON	Vektor terjemahan kendaraan ego dalam

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
		Parameter yang dibutuhkan: `x`, `y`, dan `z`. Masukkan angka untuk parameter ini.	sistem koordinat dunia.
`heading`	Ya	Objek JSON Parameter yang dibutuhkan: `qx`, `qy`, `qz`, dan `qw`. Masukkan angka untuk parameter ini.	Orientasi kerangka referensi perangkat atau sensor yang dipasang pada kendaraan yang merasakan sekitarnya, diukurkuaternion, (`qx`,`qy`,`qz`,`qw`) dalam sistem koordinat.

Sertakan Data Kamera di Manifes Input Anda

Jika Anda ingin menyertakan data kamera berwarna dengan bingkai, gunakan parameter berikut untuk memberikan informasi tentang setiap gambar. KlasterWajibkolom dalam tabel berikut berlaku ketika`images`parameter disertakan dalam file manifes masukan. Anda tidak diharuskan untuk menyertakan gambar dalam file manifes masukan Anda.

Jika Anda menyertakan gambar kamera, Anda harus menyertakan informasi tentang`position`dan orientasi (`heading`) dari kamera yang digunakan menangkap gambar.

Jika gambar Anda terdistorsi, Ground Truth dapat secara otomatis mengubah mereka menggunakan informasi yang Anda berikan tentang gambar dalam file manifes masukan Anda, termasuk koefisien distorsi (`k1`,`k2`,`k3`,`k4`,`p1`,`p1`), model kamera dan panjang fokus (`fx`,`fy`), dan titik utama (`cx`,`cy`). Untuk mempelajari lebih lanjut tentang koefisien ini dan gambar yang tidak terdistorsi, lihatKalibrasi Kamera Dengan OpenCV. Jika koefisien distorsi tidak disertakan, Ground Truth tidak akan merusak gambar.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`image-path`	Ya	String Contoh format: *<folder-name> /<imagefile.png>*	Lokasi relatif, di Amazon S3 file gambar Anda. Jalur relatif ini akan ditambahkan ke jalur yang Anda tentukan`prefix`.
`unix-timestamp`	Ya	Jumlah	Timestamp dari gambar.

`camera-model`	Tidak	String:
		
		Nilai yang Dapat diterima:
		
		`"pinhole"` , `"fisheye"`
		
		Default:
		
		`"pinhole"`
			Model kamera yang digunakan untuk menangkap gambar. Informasi ini digunakan untuk undistorsi gambar kamera.

`fx, fy`	Ya	Nomor	Panjang fokus kamera, di x (`fx`) dan y (`fy`) arah.
`cx, cy`	Ya	Nomor	x (`cx`) dan y (`cy`) koordinat titik utama.
`k1, k2, k3, k4`	Tidak	Jumlah	Koefisien distorsi radial. Didukung untuk keduanya`fisheye`dan`lubang jarum`model kamera.

Parameter	Diperlukan	Nilai yang Dapat diterima	Deskripsi
`p1, p2`	Tidak	Jumlah	Koefisien distorsi tangensial. Didukung untuklubang jarummodel kamera.
`skew`	Tidak	Jumlah	Parameter untuk mengukur kemiringan yang diketahui pada gambar.
`position`	Ya	Objek JSON Parameter yang dibutuhkan: `x`, `y`, dan `z`. Masukkan angka untuk parameter ini.	Lokasi atau asal kerangka referensi kamera yang dipasang pada kendaraan menangkap gambar.
`heading`	Ya	Objek JSON Parameter yang dibutuhkan: `qx`, `qy`, `qz`, dan `qw`. Masukkan angka untuk parameter ini.	Orientasi kerangka referensi kamera yang dipasang pada kendaraan menangkap gambar, diukur menggunakan ankuaternion, (`qx`,`qy`,`qz`,`qw`).

File Urutan dan Batas Bingkai Titik Cloud

Anda dapat menyertakan hingga 100.000 titik urutan bingkai awan dalam file manifes masukan Anda. Anda dapat menyertakan hingga 500 titik cloud frame di setiap file urutan.

Ingatlah bahwa pekerjaan pelabelan awan titik 3D memiliki waktu pra-pemrosesan yang lebih lama daripada jenis tugas Ground Truth lainnya. Untuk informasi selengkapnya, lihat Waktu Pra-pemrosesan Job.

Memahami Sistem Koordinat dan Sensor Fusion

Data titik awan selalu terletak di sistem koordinat. Sistem koordinat ini mungkin bersifat lokal untuk kendaraan atau perangkat yang merasakan lingkungan sekitar, atau mungkin sistem koordinat dunia. Saat Anda menggunakan pekerjaan pelabelan cloud titik Ground Truth 3D, semua anotasi dihasilkan menggunakan sistem koordinat data input Anda. Untuk beberapa pelabelan jenis tugas pekerjaan dan fitur, Anda harus menyediakan data dalam sistem koordinat dunia.

Dalam topik ini, Anda akan mempelajari hal-hal berikut:

- Saat Andadiperlukanmenyediakan data input dalam sistem koordinat dunia atau kerangka referensi global.
- Apa itu koordinat dunia dan bagaimana Anda dapat mengonversi data cloud titik ke sistem koordinat dunia.
- Bagaimana Anda dapat menggunakan sensor dan matriks ekstrinsik kamera untuk menyediakan data pose saat menggunakan fusi sensor.

Persyaratan Sistem Koordinat untuk Pekerjaan Pelabelan

Jika data cloud titik Anda dikumpulkan dalam sistem koordinat lokal, Anda dapat menggunakan matriks ekstrinsik dari sensor yang digunakan untuk mengumpulkan data untuk mengubahnya menjadi sistem koordinat dunia atau kerangka referensi global. Jika Anda tidak dapat memperoleh ekstrinsik untuk data cloud titik Anda dan, sebagai hasilnya, tidak dapat memperoleh awan titik dalam sistem koordinat dunia, Anda dapat memberikan data cloud titik dalam sistem koordinat lokal untuk deteksi objek awan titik 3D dan jenis tugas segmentasi semantik.

Untuk pelacakan objek, Anda harus menyediakan data titik awan dalam sistem koordinat dunia. Hal ini karena ketika Anda melacak objek di beberapa frame, kendaraan ego itu sendiri bergerak di dunia dan sehingga semua frame membutuhkan titik referensi.

Jika Anda menyertakan data kamera untuk fusi sensor, disarankan agar Anda memberikan pose kamera dalam sistem koordinat dunia yang sama dengan sensor 3D (seperti sensor LiDAR).

Menggunakan Data Point Cloud dalam Sistem Koordinat Dunia

Bagian ini menjelaskan apa sistem koordinat dunia (WCS), juga disebut sebagaikerangka referensi global, adalah dan menjelaskan bagaimana Anda dapat menyediakan data titik awan dalam sistem koordinat dunia.

Apa yang dimaksud dengan sistem koordinat dunia?

WCS atau kerangka referensi global adalah sistem koordinat universal tetap di mana sistem koordinat kendaraan dan sensor ditempatkan. Misalnya, jika beberapa titik awan frame terletak di sistem koordinat yang berbeda karena mereka dikumpulkan dari dua sensor, WCS dapat digunakan untuk menerjemahkan semua koordinat dalam frame titik awan ini ke dalam sistem koordinat tunggal, di mana semua frame memiliki asal yang sama, (0,0,0). Transformasi ini dilakukan dengan menerjemahkan asal setiap frame ke asal WCS menggunakan vektor terjemahan, dan memutar tiga sumbu (biasanya x, y, dan z) ke orientasi yang tepat menggunakan matriks rotasi. Transformasi tubuh yang kaku ini disebut atransformasi homogen.

Sistem koordinat dunia penting dalam perencanaan jalur global, lokalisasi, pemetaan, dan simulasi skenario mengemudi. Ground Truth menggunakan sistem koordinat dunia Cartesian tangan kanan seperti yang didefinisikan dalamISO 8855, di mana sumbu x maju menuju gerakan mobil, sumbu y dibiarkan, dan sumbu z menunjuk ke atas dari tanah.

Kerangka referensi global tergantung pada data. Beberapa dataset menggunakan posisi LiDAR di frame pertama sebagai asal. Dalam skenario ini, semua frame menggunakan frame pertama sebagai referensi dan judul perangkat dan posisi akan berada di dekat asal di frame pertama. Misalnya, kumpulan data KITTI memiliki frame pertama sebagai referensi untuk koordinat dunia. Set data lain menggunakan posisi perangkat yang berbeda dari asalnya.

Perhatikan bahwa ini bukan sistem koordinat GPS/IMU, yang biasanya diputar 90 derajat sepanjang sumbu z. Jika data cloud titik Anda berada dalam sistem koordinat GPS/IMU (seperti OxTS di kumpulan data AV KITTI open source), maka Anda perlu mengubah asal menjadi sistem koordinat dunia (biasanya sistem koordinat referensi kendaraan). Anda menerapkan transformasi ini dengan mengalikan data Anda dengan metrik transformasi (matriks rotasi dan vektor terjemahan). Ini akan mengubah data dari sistem koordinat aslinya menjadi sistem koordinat referensi global. Pelajari lebih lanjut tentang transformasi ini di bagian berikutnya.

Konversi Data Cloud Titik 3D ke WCS

Ground Truth mengasumsikan bahwa data cloud titik Anda telah diubah menjadi sistem koordinat referensi pilihan Anda. Misalnya, Anda dapat memilih sistem koordinat referensi sensor (seperti

LiDAR) sebagai sistem koordinat referensi global Anda. Anda juga dapat mengambil titik awan dari berbagai sensor dan mengubahnya dari pandangan sensor ke tampilan sistem koordinat referensi kendaraan. Anda menggunakan matriks ekstrinsik sensor, yang terdiri dari matriks rotasi dan vektor terjemahan, untuk mengubah data cloud titik Anda menjadi WCS atau kerangka referensi global.

Secara kolektif, vektor terjemahan dan matriks rotasi dapat digunakan untuk membuatmatriks ekstrinsik, yang dapat digunakan untuk mengkonversi data dari sistem koordinat lokal ke WCS. Misalnya, matriks ekstrinsik LiDAR Anda dapat disusun sebagai berikut, di manaRadalah matriks rotasi danTadalah vektor terjemahan:

```
LiDAR_extrinsic = [R T;0 0 0 1]
```

Misalnya, set data KITTI mengemudi otonom mencakup matriks rotasi dan vektor terjemahan untuk matriks transformasi ekstrinsik LiDAR untuk setiap frame. Klasterpykittimodul python dapat digunakan untuk memuat data KITTI, dan dalam dataset`dataset.oxts[i].T_w_imu`memberikan transformasi ekstrinsik LiDAR untuk`i`thframe dengan dapat dikalikan dengan poin dalam bingkai itu untuk mengubahnya menjadi kerangka dunia -`np.matmul(lidar_transform_matrix, points)`. Mengalikan titik dalam bingkai LiDAR dengan matriks ekstrinsik LiDAR mengubahnya menjadi koordinat dunia. Mengalikan titik dalam bingkai dunia dengan matriks ekstrinsik kamera memberikan koordinat titik dalam kerangka referensi kamera.

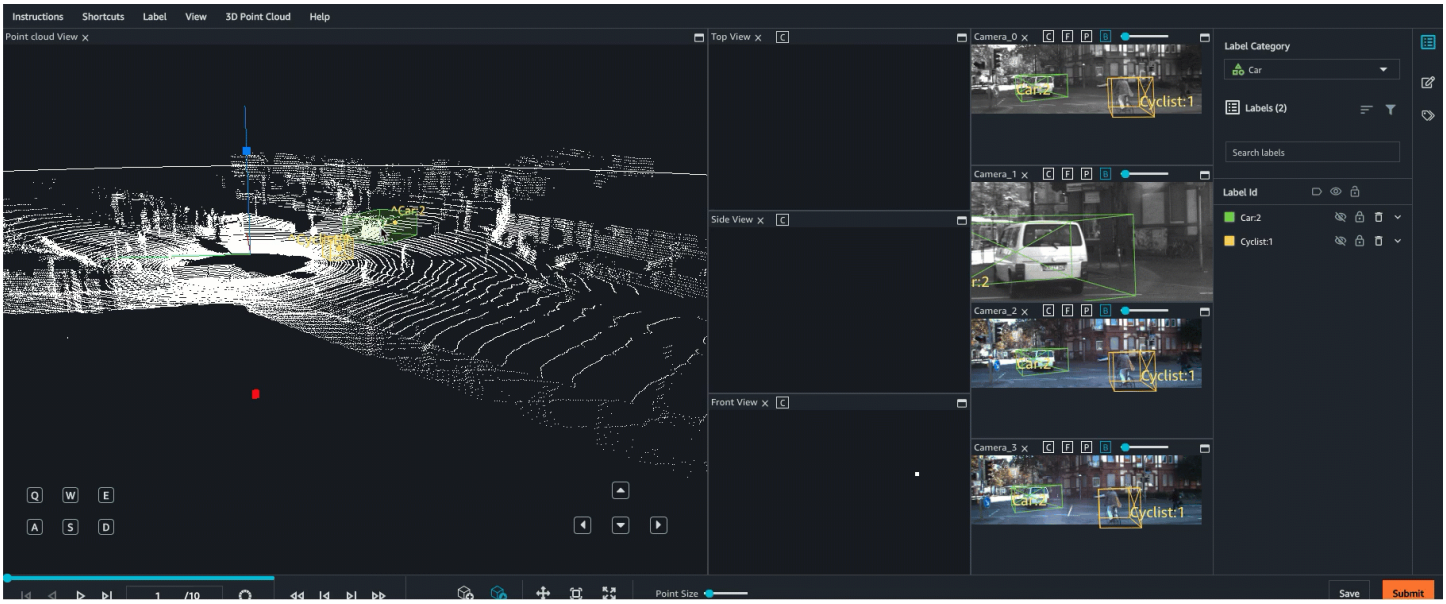
Contoh kode berikut menunjukkan bagaimana Anda dapat mengkonversi frame titik awan dari dataset KITTI menjadi WCS.

```
import pykitti
import numpy as np

basedir = '/Users/nameofuser/kitti-data'
date = '2011_09_26'
drive = '0079'

# The 'frames' argument is optional - default: None, which loads the whole dataset.
# Calibration, timestamps, and IMU data are read automatically.
# Camera and velodyne data are available via properties that create generators
# when accessed, or through getter methods that provide random access.
data = pykitti.raw(basedir, date, drive, frames=range(0, 50, 5))

# i is frame number
i = 0
```

```
# lidar extrinsic for the ith frame
lidar_extrinsic_matrix = data.oxts[i].T_w_imu

# velodyne raw point cloud in lidar scanners own coordinate system
points = data.get_velo(i)

# transform points from lidar to global frame using lidar_extrinsic_matrix
def generate_transformed_pcd_from_point_cloud(points, lidar_extrinsic_matrix):
    tps = []
    for point in points:
        transformed_points = np.matmul(lidar_extrinsic_matrix, np.array([point[0],
 point[1], point[2], 1], dtype=np.float32).reshape(4,1)).tolist()
       if len(point) > 3 and point[3] is not None:
           tps.append([transformed_points[0][0], transformed_points[1][0],
 transformed_points[2][0], point[3]])

    return tps

# customer transforms points from lidar to global frame using lidar_extrinsic_matrix
transformed_pcl = generate_transformed_pcd_from_point_cloud(points,
 lidar_extrinsic_matrix)
```

Sensor Fusion

Ground Truth mendukung fusi sensor data cloud titik dengan hingga 8 input kamera video. Fitur ini memungkinkan labeller manusia untuk melihat bingkai cloud titik 3D side-by-side dengan bingkai video yang disinkronkan. Selain menyediakan lebih banyak konteks visual untuk pelabelan, fusi sensor memungkinkan pekerja untuk menyesuaikan anotasi dalam adegan 3D dan dalam gambar 2D dan penyesuaian diproyeksikan ke tampilan lain. Video berikut menunjukkan pekerjaan pelabelan cloud titik 3D dengan LiDAR dan fusi sensor kamera.



Untuk hasil terbaik, saat menggunakan fusi sensor, cloud titik Anda harus berada di WCS. Ground Truth menggunakan sensor Anda (seperti LiDAR), kamera, dan informasi pose kendaraan ego untuk menghitung matriks ekstrinsik dan intrinsik untuk fusi sensor.

Matriks ekstrinsik

Ground Truth menggunakan sensor (seperti LiDAR) ekstrinsik dan matriks ekstrinsik dan intrinsik kamera untuk memproyeksikan objek ke dan dari kerangka referensi data titik awan ke kerangka referensi kamera.

Misalnya, untuk memproyeksikan label dari awan titik 3D ke bidang gambar kamera, Ground Truth mengubah titik 3D dari sistem koordinat LiDAR sendiri ke sistem koordinat kamera. Ini biasanya dilakukan dengan terlebih dahulu mengubah titik 3D dari sistem koordinat LiDAR sendiri ke sistem koordinat dunia (atau kerangka referensi global) menggunakan matriks ekstrinsik LiDAR. Ground Truth kemudian menggunakan kamera terbalik ekstrinsik (yang mengubah titik dari kerangka global referensi ke kerangka referensi kamera) untuk mengubah titik 3D dari sistem koordinat dunia yang diperoleh pada langkah sebelumnya ke dalam bidang gambar kamera. Matriks ekstrinsik LiDAR juga dapat digunakan untuk mengubah data 3D menjadi sistem koordinat dunia. Jika data 3D Anda sudah diubah menjadi sistem koordinat dunia maka transformasi pertama tidak berdampak pada terjemahan label, dan terjemahan label hanya bergantung pada ekstrinsik terbalik kamera. Matriks tampilan digunakan untuk memvisualisasikan label yang diproyeksikan. Untuk mempelajari lebih lanjut tentang transformasi ini dan matriks tampilan, lihatTransformasi Fusion Sensor Ground Truth.

Ground Truth menghitung matriks ekstrinsik ini dengan menggunakan LiDAR dan kameradata poseyang Anda berikan:`heading`(dalam kuaternion:`qx`,`qy`,`qz`, dan`qw`) dan`position`(`x`,`y`,`z`). Untuk

kendaraan, biasanya pos dan posisi dijelaskan dalam kerangka referensi kendaraan dalam sistem koordinat dunia dan disebut apose kendaraan ego. Untuk setiap kamera ekstrinsik, Anda dapat menambahkan informasi pose untuk kamera itu. Untuk informasi selengkapnya, lihat Pose.

Matriks intrinsik

Ground Truth menggunakan matriks ekstrinsik dan intrinsik kamera untuk menghitung metrik tampilan untuk mengubah label ke dan dari adegan 3D ke gambar kamera. Ground Truth menghitung matriks intrinsik kamera menggunakan panjang fokus kamera (`fx`,`fy`) dan koordinat pusat optik (`cx`,`cy`) yang Anda berikan. Untuk informasi selengkapnya, lihat Intrinsik dan Distorsi.

Distorsi Citra

Distorsi citra dapat terjadi karena berbagai alasan. Misalnya, gambar mungkin terdistorsi karena efek laras atau mata ikan. Ground Truth menggunakan parameter intrinsik bersama dengan distorsi co-efisien untuk undistorsi gambar yang Anda berikan saat membuat 3D titik awan pelabelan pekerjaan. Jika gambar kamera sudah tidak terdistorsi, semua koefisien distorsi harus diatur ke 0.

Untuk informasi selengkapnya tentang transformasi yang dilakukan Ground Truth untuk mengubah citra, lihatKalibrasi Kamera: Ekstrinsik, Intrinsik dan Distorsi.

Kendaraan Ego

Untuk mengumpulkan data untuk aplikasi mengemudi otonom, pengukuran yang digunakan untuk menghasilkan data cloud titik dan diambil dari sensor yang dipasang pada kendaraan, atauKendaraan ego. Untuk memproyeksikan penyesuaian label ke dan dari adegan 3D dan gambar 2D, Ground Truth membutuhkan pose kendaraan ego Anda dalam sistem koordinat dunia. Pose kendaraan ego terdiri dari koordinat posisi dan orientasi quaternion.

Ground Truth menggunakan pose kendaraan ego Anda untuk menghitung rotasi dan transformasi matriks. Rotasi dalam 3 dimensi dapat diwakili oleh urutan 3 rotasi di sekitar urutan sumbu. Secara teori, tiga sumbu yang mencakup ruang 3D Euclidean sudah cukup. Dalam prakteknya, sumbu rotasi dipilih untuk menjadi vektor dasar. Ketiga rotasi tersebut diharapkan berada dalam kerangka acuan global (ekstrinsik). Ground Truth tidak dukungan tubuh kerangka berpusat referensi (intrinsik) yang melekat pada, dan bergerak dengan, objek di bawah rotasi. Untuk melacak objek, Ground Truth perlu mengukur dari referensi global di mana semua kendaraan bergerak. Saat menggunakan pekerjaan pelabelan titik awan Ground Truth 3D, z menentukan sumbu rotasi (rotasi ekstrinsik) dan sudut Euler yaw berada dalam radian (sudut rotasi).

Pose

Ground Truth menggunakan informasi pose untuk visualisasi 3D dan fusi sensor. Pose informasi yang Anda masukan melalui file manifes Anda digunakan untuk menghitung matriks ekstrinsik. Jika Anda sudah memiliki matriks ekstrinsik, Anda dapat menggunakannya untuk mengekstrak data sensor dan pose kamera.

Misalnya dalam set data KITTI mengemudi otonom,pykittimodul python dapat digunakan untuk memuat data KITTI. Dalam set data`dataset.oxts[i].T_w_imu`memberikan transformasi ekstrinsik LiDAR untuk`i`thframe dan dapat dikalikan dengan poin untuk mendapatkan mereka dalam bingkai dunia -`matmul(lidar_transform_matrix, points)`. Transformasi ini dapat diubah menjadi posisi (vektor terjemahan) dan pos (dalam quaternion) dari LiDAR untuk format JSON file manifes masukan. Transformasi ekstrinsik kamera untuk`cam0`diithframe dapat dihitung dengan`inv(matmul(dataset.calib.T_cam0_velo, inv(dataset.oxts[i].T_w_imu)))`dan ini dapat diubah menjadi pos dan posisi untuk`cam0`.

```
import numpy

rotation = [[ 9.96714314e-01, -8.09890350e-02,  1.16333982e-03],
 [ 8.09967396e-02,  9.96661051e-01, -1.03090934e-02],
 [-3.24531964e-04,  1.03694477e-02,  9.99946183e-01]]

origin= [1.71104606e+00,
          5.80000039e-01,
          9.43144935e-01]


from scipy.spatial.transform import Rotation as R

# position is the origin
position = origin
r = R.from_matrix(np.asarray(rotation))

# heading in WCS using scipy
heading = r.as_quat()
print(f"pose:{position}\nheading: {heading}")
```

POSISI

Dalam file manifes masukan,`position`mengacu pada posisi sensor sehubungan dengan kerangka dunia. Jika Anda tidak dapat menempatkan posisi perangkat dalam sistem koordinat dunia, Anda

dapat menggunakan data LiDAR dengan koordinat lokal. Demikian pula, untuk kamera video yang dipasang Anda dapat menentukan posisi dan pos dalam sistem koordinat dunia. Untuk kamera, jika Anda tidak memiliki informasi posisi, silakan gunakan (0, 0, 0).

Berikut ini adalah bidang dalam objek posisi:

1. x(float) - x koordinat kendaraan ego, sensor, atau posisi kamera dalam meter.
2. y(float) - y koordinat kendaraan ego, sensor, atau posisi kamera dalam meter.
3. z(float) - z koordinat kendaraan ego, sensor, atau posisi kamera dalam meter.

Berikut ini adalah contoh`position`Objek JSON:

```
{
    "position": {
        "y": -152.77584902657554,
        "x": 311.21505956090624,
        "z": -10.854137529636024
      }
}
```

HEADING

Dalam file manifes masukan,`heading`adalah objek yang mewakili orientasi perangkat sehubungan dengan kerangka dunia. Nilai judul harus dalam quaternion. SEBUAHkuaternionadalah representasi dari orientasi yang konsisten dengan sifat bola geodesik. Jika Anda tidak dapat menempatkan pos sensor dalam koordinat dunia, silakan gunakan kuaternion identitas`(qx = 0, qy = 0, qz = 0, qw = 1)`. Demikian pula, untuk kamera, tentukan judul dalam kuaternion. Jika Anda tidak dapat memperoleh parameter kalibrasi kamera ekstrinsik, harap juga gunakan kuaternion identitas.

Bidang`heading`objek adalah sebagai berikut:

1. `qx`(float) - x komponen kendaraan ego, sensor, atau orientasi kamera.
2. `qy`(float) - y komponen kendaraan ego, sensor, atau orientasi kamera.
3. `qz`(float) - komponen z kendaraan ego, sensor, atau orientasi kamera.
4. `qw`(float) - w komponen kendaraan ego, sensor, atau orientasi kamera.

Berikut ini adalah contoh`heading`Objek JSON:

```
{
    "heading": {
        "qy": -0.7046155108831117,
        "qx": 0.034278837280808494,
        "qz": 0.7070617895701465,
        "qw": -0.04904659893885366
      }
}
```

Untuk mempelajari selengkapnya, lihat Hitung Orientasi Kuarter dan Posisi.

Hitung Orientasi Kuarter dan Posisi

Ground Truth mensyaratkan bahwa semua orientasi, atau pos, data diberikan dalam kuaternion. SEBUAHkuaternionadalah representasi orientasi yang konsisten dengan sifat bola geodesik yang dapat digunakan untuk perkiraan rotasi. DibandingkanSudut Eulermereka lebih sederhana untuk menulis dan menghindari masalahKunci gimbal. Dibandingkan dengan matriks rotasi mereka lebih kompak, lebih stabil secara numerik, dan lebih efisien.

Anda dapat menghitung quaternions dari matriks rotasi atau matriks transformasi.

Jika Anda memiliki matriks rotasi (terdiri dari rotasi sumbu) dan vektor terjemahan (atau asal) dalam sistem koordinat dunia alih-alih matriks transformasi kaku 4x4 tunggal, maka Anda dapat langsung menggunakan matriks rotasi dan vektor terjemahan untuk menghitung kuaternion. PustakaSciPydanpyqaternionbisa membantu. Kode-blok berikut menunjukkan contoh menggunakan perpustakaan ini untuk menghitung kuaternion dari matriks rotasi.

```
import numpy

rotation = [[ 9.96714314e-01, -8.09890350e-02,  1.16333982e-03],
 [ 8.09967396e-02,  9.96661051e-01, -1.03090934e-02],
 [-3.24531964e-04,  1.03694477e-02,  9.99946183e-01]]

origin = [1.71104606e+00,
          5.80000039e-01,
          9.43144935e-01]


from scipy.spatial.transform import Rotation as R
# position is the origin
position = origin
```

```
r = R.from_matrix(np.asarray(rotation))
# heading in WCS using scipy
heading = r.as_quat()
print(f"position:{position}\nheading: {heading}")
```

Alat UI seperti[Konverter Rotasi 3D](#)juga bisa bermanfaat.

Jika Anda memiliki matriks transformasi ekstrinsik 4x4, perhatikan bahwa matriks transformasi ada dalam bentuk`[R T; 0 0 0 1]`di mana`R`adalah matriks rotasi dan`T`adalah vektor terjemahan asal. Itu berarti Anda dapat mengekstrak matriks rotasi dan vektor terjemahan dari matriks transformasi sebagai berikut.

```
import numpy as np

transformation
= [[ 9.96714314e-01, -8.09890350e-02,  1.16333982e-03, 1.71104606e+00],
   [ 8.09967396e-02,  9.96661051e-01, -1.03090934e-02, 5.80000039e-01],
   [-3.24531964e-04,  1.03694477e-02,  9.99946183e-01, 9.43144935e-01],
   [              0,               0,               0,              1]]

transformation  = np.array(transformation )
rotation = transformation[0:3][0:3]
translation= transformation[0:3][3]

from scipy.spatial.transform import Rotation as R
# position is the origin translation
position = translation
r = R.from_matrix(np.asarray(rotation))
# heading in WCS using scipy
heading = r.as_quat()
print(f"position:{position}\nheading: {heading}")
```

Dengan pengaturan Anda sendiri, Anda dapat menghitung matriks transformasi ekstrinsik menggunakan posisi dan orientasi GPS/IMU (lintang, bujur, ketinggian dan gulungan, pitch, yaw) sehubungan dengan sensor LiDAR pada kendaraan ego. Misalnya, Anda dapat menghitung pose dari data mentah KITTI menggunakan`pose = convertOxtsToPose(oxts)`untuk mengubah data lembu menjadi pose euclidean lokal, ditentukan oleh matriks transformasi kaku 4x4. Anda kemudian dapat mengubah matriks transformasi pose ini ke kerangka referensi global menggunakan matriks transformasi frame referensi dalam sistem koordinat dunia.

```
struct Quaternion
```

```
{
    double w, x, y, z;
};

Quaternion ToQuaternion(double yaw, double pitch, double roll) // yaw (Z), pitch (Y),
 roll (X)
{
    // Abbreviations for the various angular functions
    double cy = cos(yaw * 0.5);
    double sy = sin(yaw * 0.5);
    double cp = cos(pitch * 0.5);
    double sp = sin(pitch * 0.5);
    double cr = cos(roll * 0.5);
    double sr = sin(roll * 0.5);

    Quaternion q;
    q.w = cr * cp * cy + sr * sp * sy;
    q.x = sr * cp * cy - cr * sp * sy;
    q.y = cr * sp * cy + sr * cp * sy;
    q.z = cr * cp * sy - sr * sp * cy;

    return q;
}
```

Transformasi Fusion Sensor Ground Truth

Bagian berikut membahas lebih detail tentang transformasi fusi sensor Ground Truth yang dilakukan menggunakan data pose yang Anda berikan.

Ektrinsik LiDAR

Untuk memproyeksikan ke dan dari adegan LiDAR 3D ke gambar kamera 2D, Ground Truth menghitung metrik proyeksi transformasi yang kaku menggunakan pose dan pos kendaraan ego. Ground Truth menghitung rotasi dan terjemahan dari koordinat dunia ke dalam pesawat 3D dengan melakukan urutan sederhana rotasi dan terjemahan.

Ground Truth menghitung metrik rotasi menggunakan kuaternion judul sebagai berikut:

$$M = \begin{pmatrix} 1 - 2y^2 - 2z^2 & 2xy + 2zw & 2xz - 2yw \\ 2xy - 2zw & 1 - 2x^2 - 2z^2 & 2yz + 2xw \\ 2xz + 2yw & 2yz - 2xw & 1 - 2x^2 - 2y^2 \end{pmatrix}$$

Di sini,`[x, y, z, w]`sesuai dengan parameter di`heading`Objek JSON,`[qx, qy, qz, qw]`. Ground Truth menghitung vektor kolom terjemahan sebagai`T = [poseX, poseY, poseZ]`. Maka metrik ekstrinsik adalah sebagai berikut:

```
LiDAR_extrinsic = [R T;0 0 0 1]
```

Kalibrasi Kamera: Ekstrinsik, Intrinsik dan Distorsi

Kalibrasi kamera geometri, juga disebut sebagaireseksi kamera, memperkirakan parameter lensa dan sensor gambar dari kamera gambar atau video. Anda dapat menggunakan parameter ini untuk mengoreksi distorsi lensa, mengukur ukuran objek dalam satuan dunia, atau menentukan lokasi kamera dalam pemandangan. Parameter kamera mencakup intrinsik dan koefisien distorsi.

Kamera Extrinsik

Jika pose kamera diberikan, maka Ground Truth menghitung ekstrinsik kamera berdasarkan transformasi kaku dari bidang 3D ke bidang kamera. Perhitungannya sama dengan yang digunakan untukEktrinsik LiDAR, kecuali bahwa Ground Truth menggunakan pose kamera (`position`dan`heading`) dan menghitung ekstrinsik terbalik.

```
camera_inverse_extrinsic = inv([Rc Tc;0 0 0 1]) #where Rc and Tc are camera pose
components
```

Intrinsik dan Distorsi

Beberapa kamera, seperti kamera lubang jarum atau fisheye, dapat menimbulkan distorsi yang signifikan pada foto. Distorsi ini dapat diperbaiki menggunakan koefisien distorsi dan panjang fokus kamera. Untuk mempelajari lebih lanjut, lihatKalibrasi Kamera Dengan OpenCVdalam dokumentasi OpenCV.

Ada dua jenis distorsi Ground Truth yang dapat diperbaiki: distorsi radial dan distorsi tangensial.

Distorsi Radialterjadi ketika sinar cahaya menekuk lebih dekat tepi lensa daripada yang mereka lakukan di pusat optiknya. Semakin kecil lensa, semakin besar distorsi. Kehadiran distorsi radial bermanifestasi dalam bentukbarelataumata-ikanefek dan Ground Truth menggunakan Formula 1 untuk undistorsi itu.

Formula 1:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Distorsi tangensialterjadi karena lensa yang digunakan untuk mengambil gambar tidak parallel sempurna dengan bidang pencitraan. Ini dapat diperbaiki dengan Formula 2.

Formula 2:

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

Dalam file manifes masukan, Anda dapat memberikan koefisien distorsi dan Ground Truth akan mengubah gambar Anda. Semua koefisien distorsi adalah pelampung.

- `k1,k2,k3,k4`- Koefisien distorsi radial. Didukung untuk model kamera fisheye dan lubang jarum.
- `p1,p2`- Koefisien distorsi tangensial. Didukung untuk model kamera lubang jarum.

Jika gambar sudah tidak terdistorsi, semua koefisien distorsi harus 0 dalam manifes masukan Anda.

Untuk merekonstruksi gambar yang dikoreksi dengan benar, Ground Truth melakukan konversi unit gambar berdasarkan panjang fokus. Jika panjang fokus umum digunakan dengan rasio aspek tertentu untuk kedua sumbu, seperti 1, dalam rumus atas kita akan memiliki panjang fokus tunggal. Matriks yang berisi empat parameter ini disebut sebagaidalam matriks kalibrasi intrinsik kamera.

$$\begin{Bmatrix} x \\ y \\ w \end{Bmatrix} = \begin{Bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$$

Meskipun koefisien distorsi sama terlepas dari resolusi kamera yang digunakan, ini harus diskalakan dengan resolusi saat ini dari resolusi yang dikalibrasi.

Berikut ini adalah nilai float.

- `fx`- panjang fokus dalam arah x.
- `fy`- panjang fokus dalam arah y.
- `cx`- x koordinat titik utama.
- `cy`- y koordinat titik utama.

Ground Truth menggunakan kamera ekstrinsik dan intrinsik kamera untuk menghitung metrik tampilan seperti yang ditunjukkan pada blok kode berikut untuk mengubah label antara adegan 3D dan gambar 2D.

```
def generate_view_matrix(intrinsic_matrix, extrinsic_matrix):
    intrinsic_matrix = np.c_[intrinsic_matrix, np.zeros(3)]
    view_matrix = np.matmul(intrinsic_matrix, extrinsic_matrix)
    view_matrix = np.insert(view_matrix, 2, np.array((0, 0, 0, 1)), 0)
    return view_matrix
```

Data Input Bingkai Video

Saat Anda membuat deteksi objek bingkai video atau pekerjaan pelabelan pelacakan objek, Anda dapat memilih file video (file MP4) atau bingkai video untuk data input. Semua tugas pekerja dibuat menggunakan bingkai video, jadi jika Anda memilih file video, gunakan alat ekstraksi bingkai Ground Truth untuk mengekstrak bingkai video (gambar) dari file video Anda.

Untuk kedua opsi ini, Anda dapat menggunakanPengaturan data otomatisopsi di bagian Ground Truth di Amazon SageMaker konsol untuk mengatur koneksi antara Ground Truth dan data input Anda di Amazon S3 sehingga Ground Truth tahu di mana harus mencari data input Anda saat membuat tugas pelabelan Anda. Ini membuat dan menyimpan file manifes masukan di lokasi set data input Amazon S3 Anda. Untuk mempelajari selengkapnya, lihat Pengaturan Data Input Bingkai Video Otomatis.

Atau, Anda dapat secara manual membuat file urutan untuk setiap urutan bingkai video yang ingin diberi label dan menyediakan lokasi Amazon S3 dari file manifes masukan yang mereferensikan setiap file urutan ini menggunakan`source-ref`kunci. Untuk mempelajari selengkapnya, lihat Membuat File Manifes Masukan Bingkai Video.

Topik

- Pilih File Video atau Bingkai Video untuk Input Data
- Pengaturan data

Pilih File Video atau Bingkai Video untuk Input Data

Saat Anda membuat deteksi objek bingkai video atau pekerjaan pelabelan pelacakan objek, Anda dapat memberikan urutan bingkai video (gambar) atau Anda dapat menggunakan Amazon SageMaker konsol untuk memiliki Ground Truth secara otomatis mengekstrak bingkai video dari file video Anda. Gunakan bagian berikut untuk mempelajari lebih lanjut tentang opsi ini.

Menyediakan Bingkai video

Bingkai video adalah urutan gambar yang diekstrak dari file video. Anda dapat membuat pekerjaan pelabelan Ground Truth agar pekerja memberi label beberapa urutan bingkai video. Setiap urutan terdiri dari gambar yang diekstrak dari satu video.

Untuk membuat pekerjaan pelabelan menggunakan urutan bingkai video, Anda harus menyimpan setiap urutan menggunakan yang unikprefiks nama kuncidi Amazon S3. Di konsol Amazon S3, prefiks nama kunci adalah folder. Jadi di konsol Amazon S3, setiap urutan bingkai video harus ditempatkan di foldernya sendiri di Amazon S3.

Misalnya, jika Anda memiliki dua urutan bingkai video, Anda dapat menggunakan awalan nama kuncisequence1/dansequence2/untuk mengidentifikasi urutan Anda. Dalam contoh ini, urutan Anda mungkin terletak dis3://*DOC-EXAMPLE-BUCKET*/video-frames/sequence1/dans3://*DOC-EXAMPLE-BUCKET*/video-frames/sequence2/.

Jika Anda menggunakan konsol Ground Truth untuk membuat file manifes masukan, semua awalan nama kunci urutan harus berada di lokasi yang sama di Amazon S3. Misalnya, di konsol Amazon S3, setiap urutan dapat berada dalam folder dis3://*DOC-EXAMPLE-BUCKET*/video-frames/. Dalam contoh ini, urutan pertama Anda frame video (gambar) mungkin terletak dis3://*DOC-EXAMPLE-BUCKET*/video-frames/sequence1/dan urutan kedua Anda mungkin terletak dis3://*DOC-EXAMPLE-BUCKET*/video-frames/sequence2/.

> **Important**
>
> Bahkan jika Anda hanya memiliki satu urutan bingkai video yang ingin diberi label oleh pekerja, urutan tersebut harus memiliki awalan nama kunci di Amazon S3. Jika Anda menggunakan konsol Amazon S3, ini berarti urutan Anda terletak di folder. Itu tidak dapat ditemukan di root bucket S3 Anda.

Saat membuat tugas pekerja menggunakan urutan bingkai video, Ground Truth menggunakan satu urutan per tugas. Dalam setiap tugas, Ground Truth memerintahkan bingkai video Anda menggunakanUTF-8urutan biner.

Misalnya, bingkai video mungkin dalam urutan berikut di Amazon S3:

```
[0001.jpg, 0002.jpg, 0003.jpg, ..., 0011.jpg]
```

Mereka diatur dalam urutan yang sama dalam tugas pekerja:`0001.jpg, 0002.jpg, 0003.jpg, ..., 0011.jpg`.

Frame mungkin juga dipesan menggunakan konvensi penamaan seperti berikut ini:

```
[frame1.jpg, frame2.jpg, ..., frame11.jpg]
```

Dalam kasus ini,`frame10.jpg`dan`frame11.jpg`datang sebelum`frame2.jpg`dalam tugas pekerja. Pekerja Anda melihat bingkai video Anda dalam urutan berikut:`frame1.jpg, frame10.jpg, frame11.jpg, frame2.jpg, ..., frame9.jpg`.

Menyediakan File Video

Anda dapat menggunakan fitur pemisahan bingkai Ground Truth saat membuat pekerjaan pelabelan baru di konsol untuk mengekstrak bingkai video dari file video (file MP4). Serangkaian bingkai video yang diekstrak dari satu file video disebut sebagaiurutan frame video.

Anda dapat memiliki Ground Truth secara otomatis mengekstrak semua frame, hingga 2.000, dari video, atau Anda dapat menentukan frekuensi untuk ekstraksi bingkai. Misalnya, Anda dapat memiliki ekstrak Ground Truth setiap 10^{th} bingkai dari video Anda.

Anda dapat menyediakan hingga 50 video saat Anda menggunakan pengaturan data otomatis untuk mengekstrak bingkai, namun file manifes masukan Anda tidak dapat mereferensikan lebih dari 10 file urutan bingkai video saat Anda membuat pelacakan objek bingkai video dan pekerjaan pelabelan deteksi objek bingkai video. Jika Anda menggunakan alat konsol pengaturan data otomatis untuk mengekstrak bingkai video dari lebih dari 10 file video, Anda perlu memodifikasi file manifes yang dihasilkan alat atau membuat yang baru untuk menyertakan 10 file urutan bingkai video atau kurang. Untuk mempelajari lebih lanjut tentang kuota ini, lihatKuota Job Pelabelan 3D Point Cloud dan Bingkai Video.

Untuk menggunakan alat ekstraksi bingkai video, lihatPengaturan Data Input Bingkai Video Otomatis.

Ketika semua frame video Anda telah berhasil diekstraksi dari video Anda, Anda akan melihat hal berikut di lokasi set data masukan S3 Anda:

- Prefiks nama kunci (folder di konsol Amazon S3) diberi nama dari setiap video. Masing-masing awalan ini mengarah ke:
  - Urutan bingkai video yang diekstrak dari video yang digunakan untuk memberi nama awalan itu.
  - File urutan yang digunakan untuk mengidentifikasi semua gambar yang membentuk urutan itu.
- File manifes masukan dengan ekstensi.manifest. Ini mengidentifikasi semua file urutan yang akan digunakan untuk membuat pekerjaan pelabelan Anda.

Semua frame yang diekstrak dari satu file video digunakan untuk tugas pelabelan. Jika Anda mengekstrak bingkai video dari beberapa file video, beberapa tugas dibuat untuk pekerjaan pelabelan Anda, satu untuk setiap urutan bingkai video.

Ground Truth menyimpan setiap urutan bingkai video yang diekstraknya di lokasi Amazon S3 Anda untuk set data input menggunakan yang unikprefiks nama kunci. Di konsol Amazon S3, prefiks nama kunci adalah folder.

Pengaturan data

Saat Anda membuat pekerjaan pelabelan bingkai video, Anda perlu memberi tahu Ground Truth di mana harus mencari data input Anda. Anda dapat melakukannya dengan salah satu dari dua cara berikut:

- Anda dapat menyimpan data input Anda di Amazon S3 dan meminta Ground Truth secara otomatis mendeteksi kumpulan data input yang digunakan untuk pekerjaan pelabelan Anda. LihatPengaturan Data Input Bingkai Video Otomatisuntuk mempelajari lebih lanjut tentang opsi ini.
- Anda dapat membuat file manifes masukan dan file urutan dan mengunggahnya ke Amazon S3. LihatPengaturan Data Input Manualuntuk mempelajari lebih lanjut tentang opsi ini.

Topik

Pengaturan Data Input Bingkai Video Otomatis

Anda dapat menggunakan pengaturan data otomatis Ground Truth untuk mendeteksi file video secara otomatis di bucket Amazon S3 Anda dan mengekstrak bingkai video dari file tersebut. Untuk mempelajari caranya, lihat Menyediakan File Video.

Jika Anda sudah memiliki bingkai video di Amazon S3, Anda dapat menggunakan pengaturan data otomatis untuk menggunakan bingkai video ini dalam pekerjaan pelabelan Anda. Untuk opsi ini, semua bingkai video dari satu video harus disimpan menggunakan awalan unik. Untuk mempelajari tentang persyaratan untuk menggunakan opsi ini, lihatMenyediakan Bingkai video.

Pilih salah satu bagian berikut untuk mempelajari cara mengatur koneksi set data input otomatis Anda dengan Ground Truth.

Menyediakan File Video dan Frame Ekstrak

Gunakan prosedur berikut untuk menghubungkan file video Anda dengan Ground Truth dan secara otomatis mengekstrak bingkai video dari file-file tersebut untuk deteksi objek bingkai video dan pekerjaan pelabelan pelacakan objek.

> **Note**
>
> Jika Anda menggunakan alat konsol pengaturan data otomatis untuk mengekstrak bingkai video dari lebih dari 10 file video, Anda perlu memodifikasi file manifes yang dihasilkan alat atau membuat yang baru untuk menyertakan 10 file urutan bingkai video atau kurang. Untuk mempelajari selengkapnya, lihat Menyediakan File Video.

Pastikan file video Anda disimpan dalam bucket Amazon S3 yang samaAWSWilayah tempat Anda melakukan pengaturan data otomatis.

Hubungkan file video Anda secara otomatis di Amazon S3 dengan Ground Truth dan ekstrak bingkai video:

1. Arahkan keBuat tugas pelabelanhalaman di Amazon SageMakerkonsol:https://console.aws.amazon.com/sagemaker/groundtruth.

   Bucket S3 input dan output harus ditempatkan di bucket S3 yang samaAWSWilayah tempat Anda membuat pekerjaan pelabelan Anda. Link ini menempatkan Anda di North Virginia (us-east-1)AWSWilayah. Jika data input Anda berada di bucket Amazon S3 di Wilayah lain, beralihlah ke Wilayah tersebut. Untuk mengubahAWSWilayah, dibar navigasi, pilih nama Wilayah yang ditampilkan saat ini.
2. PilihBuat tugas pelabelan.
3. ENTER aNama tugas.
4. Dalam Bagian IniPengaturan data, pilihPengaturan data otomatis.
5. Masukkan URI Amazon S3 untukLokasi S3 untuk kumpulan data input. URI S3 terlihat seperti berikut ini: `s3://DOC-EXAMPLE-BUCKET/path-to-files/`. URI ini harus mengarah ke lokasi Amazon S3 di mana file video Anda disimpan.
6. TentukanLokasi S3 untuk kumpulan data keluaran. Di sinilah data output Anda disimpan. Anda dapat memilih untuk menyimpan data keluaran Anda diLokasi yang sama dengan set data masukanatauTentukan lokasi barudan memasukkan URI S3 dari lokasi yang ingin Anda simpan data keluaran Anda.
7. PilihFile videountukJenis datamenggunakan daftar dropdown.
8. PilihYa, ekstrak frame untuk pelacakan objek dan tugas deteksi.
9. Pilih metodeEkstraksi bingkai.

   - Bila Anda memilihGunakan semua frame yang diekstrak dari video untuk membuat tugas pelabelan, Ground Truth mengekstrak semua frame dari setiap video diLokasi S3 untuk kumpulan data input, hingga 2.000 frame. Jika video dalam kumpulan data masukan Anda berisi lebih dari 2.000 frame, 2.000 frame pertama diekstraksi dan digunakan untuk tugas pelabelan tersebut.
   - Bila Anda memilihGunakan setiap*x*bingkai dari video untuk membuat tugas pelabelan, Ground Truth mengekstrak setiapx^{th}bingkai dari setiap video diLokasi S3 untuk kumpulan data input.

Misalnya, jika video Anda berdurasi 2 detik, dan memilikilaju bingkaidari 30 frame per detik, ada 60 frame di video Anda. Jika Anda menentukan 10 di sini, Ground Truth mengekstrak setiap 10^{th}bingkai dari video Anda. Ini artinyast, 10^{th}, 20^{th}, 30^{th}, 40^{th}, 50^{th}, dan 60^{th}frame diekstraksi.

10. Pilih atau buat peran eksekusi IAM. Pastikan peran ini memiliki izin untuk mengakses lokasi Amazon S3 Anda untuk data input dan output yang ditentukan dalam langkah 5 dan 6.
11. PilihSelesaikan pengaturan data.

Menyediakan Bingkai video

Gunakan prosedur berikut untuk menghubungkan urutan bingkai video Anda dengan Ground Truth untuk deteksi objek bingkai video dan pekerjaan pelabelan pelacakan objek.

Pastikan bingkai video Anda disimpan dalam bucket Amazon S3 yang samaAWSWilayah tempat Anda melakukan pengaturan data otomatis. Setiap urutan bingkai video harus memiliki awalan yang unik. Misalnya, jika Anda memiliki dua urutan yang disimpan dis3://*DOC-EXAMPLE-BUCKET*/video-frames/sequences/, masing-masing harus memiliki awalan yang unik sepertisequence1dansequence2dan keduanya harus ditempatkan langsung di bawah/sequences/prefiks. Dalam contoh di atas, lokasi dari dua urutan ini adalah:s3://*DOC-EXAMPLE-BUCKET*/video-frames/sequences/sequence1/dans3://*DOC-EXAMPLE-BUCKET*/video-frames/sequences/sequence2/.

Secara otomatis menghubungkan bingkai video Anda di Amazon S3 dengan Ground Truth:

1. Arahkan keBuat tugas pelabelanhalaman di Amazon SageMaker konsol:https://console.aws.amazon.com/sagemaker/groundtruth.

   Bucket S3 input dan output harus ditempatkan di bucket S3 yang samaAWSWilayah tempat Anda membuat pekerjaan pelabelan Anda. Link ini menempatkan Anda di North Virginia (us-east-1)AWSWilayah. Jika data input Anda berada di bucket Amazon S3 di Wilayah lain, beralihlah ke Wilayah tersebut. Untuk mengubahAWSWilayah, dibar navigasi, pilih nama Wilayah yang ditampilkan saat ini.
2. PilihBuat tugas pelabelan.
3. ENTER aNama tugas.
4. Dalam Bagian IniPengaturan data, pilihPengaturan data otomatis.
5. Masukkan URI Amazon S3 untukLokasi S3 untuk kumpulan data input.

Lokasi Amazon S3 di mana urutan Anda disimpan. Misalnya, jika Anda memiliki dua urutan yang disimpan dis3://*DOC-EXAMPLE-BUCKET*/video-frames/sequences/sequence1/,s3://*DOC-EXAMPLE-BUCKET*/video-frames/sequences/sequence2/, masukkans3://*DOC-EXAMPLE-BUCKET*/video-frames/sequences/di sini.

6. TentukanLokasi S3 untuk kumpulan data keluaran. Di sinilah data output Anda disimpan. Anda dapat memilih untuk menyimpan data keluaran Anda diLokasi yang sama dengan set data masukanatauTentukan lokasi barudan memasukkan URI S3 dari lokasi yang ingin Anda simpan data keluaran Anda.
7. PilihFrame videountukJenis datamenggunakan daftar dropdown.
8. Pilih atau buat peran eksekusi IAM. Pastikan peran ini memiliki izin untuk mengakses lokasi Amazon S3 Anda untuk data input dan output yang ditentukan dalam langkah 5 dan 6.
9. PilihSelesaikan pengaturan data.

Prosedur ini akan membuat manifes masukan di lokasi Amazon S3 untuk kumpulan data input yang Anda tentukan pada langkah 5. Jika Anda membuat pekerjaan pelabelan menggunakan SageMaker API atau,AWS CLI, atauAWSSDK, gunakan URI Amazon S3 untuk file manifes masukan ini sebagai masukan ke parameter`ManifestS3Uri`.

Pengaturan Data Input Manual

Pilih opsi penyiapan data manual jika Anda telah membuat file urutan untuk setiap urutan bingkai video Anda, dan daftar file manifes referensi ke file urutan tersebut.

Membuat File Manifes Masukan Bingkai Video

Ground Truth menggunakan file manifes masukan untuk mengidentifikasi lokasi kumpulan data input Anda saat membuat tugas pelabelan. Untuk deteksi objek bingkai video dan pekerjaan pelabelan pelacakan objek, setiap baris dalam file manifes masukan mengidentifikasi lokasi file urutan bingkai video. Setiap file urutan mengidentifikasi gambar yang disertakan dalam satu urutan bingkai video.

Gunakan halaman ini untuk mempelajari cara membuat file urutan bingkai video dan file manifes masukan untuk pelacakan objek bingkai video dan pekerjaan pelabelan deteksi objek.

Jika Anda ingin Ground Truth secara otomatis menghasilkan file urutan dan memasukkan file manifes Anda, lihatPengaturan Data Input Bingkai Video Otomatis.

Membuat Manifes Input Urutan Bingkai Video

Dalam file manifes masukan urutan bingkai video, setiap baris dalam manifes adalah objek JSON, dengan`"source-ref"`kunci yang mereferensikan file urutan. Setiap file urutan mengidentifikasi lokasi urutan frame video. Ini adalah format file manifes yang diperlukan untuk semua pekerjaan pelabelan bingkai video.

Contoh berikut menunjukkan sintaksis yang digunakan untuk file manifes input:

```
{"source-ref": "s3://DOC-EXAMPLE-BUCKET/example-folder/seq1.json"}
{"source-ref": "s3://DOC-EXAMPLE-BUCKET/example-folder/seq2.json"}
```

Buat File Urutan Bingkai Video

Data untuk setiap urutan frame video perlu disimpan dalam objek data JSON. Berikut ini adalah contoh format yang Anda gunakan untuk file urutan. Informasi tentang setiap frame disertakan sebagai objek JSON dan tercantum dalam`frames`Daftar. JSON berikut telah diperluas untuk keterbacaan.

```
{
 "seq-no": 1,
 "prefix": "s3://mybucket/prefix/video1/",
 "number-of-frames": 3,
 "frames":[
   {"frame-no": 1, "unix-timestamp": 1566861644, "frame": "frame0001.jpg" },
   {"frame-no": 2, "unix-timestamp": 1566861644, "frame": "frame0002.jpg" },
   {"frame-no": 3, "unix-timestamp": 1566861644, "frame": "frame0003.jpg" }
 ]
}
```

Tabel berikut memberikan rincian tentang parameter yang ditunjukkan dalam contoh kode ini.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`seq-no`	Ya	Bulat	Nomor urutan yang dipesan.
`prefix`	Ya	String Nilai yang Diterima:	Lokasi Amazon S3 di mana file urutan berada.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
		`s3://<bucket-name> /<prefix>/`	Prefiks harus diakhiri dengan garis miring ke depan:`/`.
`number-of-frames`	Ya	Bulat	Jumlah total frame termasuk dalam file urutan. Jumlah ini harus sesuai dengan jumlah frame yang tercantum dalam`frames`parameter di baris berikutnya.

`frames`	Ya	Daftar objek JSON
		
		Wajib:
		
		`frame-no`, `frame`
		
		Opsional:
		
		`unix-timestamp`
			Daftar data bingkai. Panjang daftar harus sama`number-of-frames` . Di UI pekerja, frame dalam urutan diurutkan UTF-8urutan biner. Untuk mempelajari tentang pemesanan ini, lihatMenyediakan Bingkai video.

`frame-no`	Ya	Bulat	Nomor pesanan bingkai. Ini akan menentukan urutan bingkai dalam urutan.
`unix-timestamp`	Tidak	Bulat	Unix timestamp dari frame. Jumlah detik sejak 1 Januari 1970 hingga waktu UTC ketika frame ditangkap.

Parameter	Diperlukan	Nilai yang Diterima	Deskripsi
`frame`	Ya	String	Nama file gambar bingkai video.

Data Output

Output dari pekerjaan pelabelan ditempatkan di lokasi Amazon S3 yang Anda tentukan di konsol atau dalam panggilan ke CreateLabelingJoboperasi. Data keluaran muncul di lokasi ini ketika pekerja telah mengirimkan satu atau lebih tugas, atau saat tugas kedaluwarsa. Perhatikan bahwa mungkin diperlukan beberapa menit agar data keluaran muncul di Amazon S3 setelah pekerja mengirimkan tugas atau tugas berakhir.

Setiap baris dalam file data keluaran identik dengan file manifes dengan penambahan atribut dan nilai untuk label yang ditetapkan ke objek input. Nama atribut untuk nilai didefinisikan di konsol atau dalam panggilan ke`CreateLabelingJob` operasi. Anda tidak dapat menggunakan`-metadata` nama atribut label. Jika Anda menjalankan segmentasi semantik gambar, segmentasi semantik awan titik 3D, atau pekerjaan pelacakan objek awan titik 3D, atribut label harus diakhiri dengan`-ref`. Untuk jenis pekerjaan lainnya, nama atribut tidak dapat diakhiri dengan`-ref`.

Output dari pekerjaan pelabelan adalah nilai pasangan kunci-nilai dengan label. Label dan nilai menimpa data JSON yang ada dalam file input dengan nilai baru.

Misalnya, berikut ini adalah output dari pekerjaan pelabelan klasifikasi gambar tempat file data input disimpan di Amazon S3*AWSDOC-EXAMPLE-BUCKET* dan nama atribut label didefinisikan sebagai *`sport`*. Dalam contoh ini objek JSON diformat untuk dibaca, dalam file output yang sebenarnya objek JSON adalah pada satu baris. Untuk informasi lebih lanjut tentang format data, lihat JSON Lines.

```
{
    "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/image_example.png",
    "sport":0,
    "sport-metadata":
    {
        "class-name": "football",
        "confidence": 0.00,
        "type":"groundtruth/image-classification",
        "job-name": "identify-sport",
        "human-annotated": "yes",
        "creation-date": "2018-10-18T22:18:13.527256"
```

```
    }
}
```

Nilai label dapat berupa JSON yang valid. Dalam hal ini nilai label adalah indeks kelas dalam daftar klasifikasi. Jenis pekerjaan lain, seperti kotak pembatas, memiliki nilai yang lebih kompleks.

Setiap pasangan kunci-nilai dalam file manifes masukan selain atribut label tidak berubah dalam file keluaran. Anda dapat menggunakan ini untuk meneruskan data ke aplikasi Anda.

Output dari pekerjaan pelabelan dapat digunakan sebagai masukan ke pekerjaan pelabelan lain. Anda dapat menggunakan ini ketika Anda merantai bersama pelabelan pekerjaan. Misalnya, Anda dapat mengirim satu pekerjaan pelabelan untuk menentukan olahraga yang sedang dimainkan. Kemudian Anda mengirim yang lain menggunakan data yang sama untuk menentukan apakah olahraga sedang dimainkan di dalam atau di luar ruangan. Dengan menggunakan data keluaran dari pekerjaan pertama sebagai manifes untuk pekerjaan kedua, Anda dapat mengkonsolidasikan hasil dari dua pekerjaan menjadi satu file keluaran agar lebih mudah diproses oleh aplikasi Anda.

File data output ditulis ke lokasi output secara berkala saat pekerjaan sedang berlangsung. File perantara ini berisi satu baris untuk setiap baris dalam file manifes. Jika sebuah objek diberi label, label disertakan. Jika objek belum diberi label, itu ditulis ke file output menengah identik dengan file manifes.

Direktori Output

Ground Truth membuat beberapa direktori di jalur keluaran Amazon S3 Anda. Direktori ini berisi hasil pekerjaan pelabelan Anda dan artefak pekerjaan lainnya. Direktori tingkat atas untuk pekerjaan pelabelan diberi nama yang sama dengan pekerjaan pelabelan Anda; direktori keluaran ditempatkan di bawahnya. Misalnya, jika Anda menamai pekerjaan pelabelan Anda**`find-people`**, output Anda akan berada di direktori berikut:

```
s3://AWSDOC-EXAMPLE-BUCKET/find-people/activelearning
s3://AWSDOC-EXAMPLE-BUCKET/find-people/annotations
s3://AWSDOC-EXAMPLE-BUCKET/find-people/inference
s3://AWSDOC-EXAMPLE-BUCKET/find-people/manifests
s3://AWSDOC-EXAMPLE-BUCKET/find-people/training
```

Setiap direktori berisi output sebagai berikut:

Direktori Pembelajaran Aktif

`activelearning`Direktori ini hanya ada ketika Anda menggunakan pelabelan data otomatis. Ini berisi set validasi input dan output untuk pelabelan data otomatis, dan folder input dan output untuk data berlabel otomatis.

Direktori Anotasi

`annotations`Direktori berisi semua anotasi yang dibuat oleh tenaga kerja. Ini adalah tanggapan dari pekerja individu yang belum dikonsolidasikan ke dalam satu label untuk objek data.

Ada tiga subdirektori dalam`annotations` direktori.

- Yang pertama`worker-response`,, berisi tanggapan dari pekerja individu. Ini berisi subdirektori untuk setiap iterasi, yang pada gilirannya berisi subdirektori untuk setiap objek data dalam iterasi itu. Data respons pekerja untuk setiap objek data disimpan dalam file JSON stempel waktu yang berisi jawaban yang dikirimkan oleh setiap pekerja untuk objek data tersebut, dan jika Anda menggunakan tenaga kerja pribadi, metadata tentang pekerja tersebut. Untuk mempelajari selengkapnya tentang metadata ini, lihatMetadata Pekerja.
- Yang kedua,`consolidated-annotation`, berisi informasi yang diperlukan untuk mengkonsolidasikan anotasi dalam batch saat ini ke dalam label untuk objek data Anda.
- Yang ketiga,`intermediate`, berisi manifes keluaran untuk batch saat ini dengan label yang telah selesai. File ini diperbarui sebagai label untuk setiap objek data selesai.

> **Note**
>
> Kami menyarankan Anda untuk tidak menggunakan file yang tidak disebutkan dalam dokumentasi.

Direktori

`inference`Direktori ini hanya ada ketika Anda menggunakan pelabelan data otomatis. Direktori ini berisi file input dan output untuk SageMaker batch transform yang digunakan saat pelabelan objek data.

Direktori Manifes

`manifest`Direktori berisi manifes keluaran dari pekerjaan pelabelan Anda. Ada satu subdirektori dalam direktori manifes,`output`. `output`Direktori berisi file manifes keluaran untuk pekerjaan pelabelan Anda. File tersebut dinamai`output.manifest`.

Direktori Pelatihan

`training`Direktori ini hanya ada ketika Anda menggunakan pelabelan data otomatis. Direktori ini berisi file input dan output yang digunakan untuk melatih model pelabelan data otomatis.

Skor Keyakinan

Bila Anda memiliki lebih dari satu pekerja membubuhi anotasi satu tugas, label Anda akan dihasilkan dari konsolidasi anotasi. Ground Truth menghitung skor kepercayaan untuk setiap label. Skor kepercayaan adalah angka antara 0 dan 1 yang menunjukkan seberapa percaya diri Ground Truth dalam label. Anda dapat menggunakan skor kepercayaan diri untuk membandingkan objek data berlabel satu sama lain, dan untuk mengidentifikasi label yang paling tidak atau paling percaya diri.

Anda tidak boleh menafsirkan nilai skor kepercayaan sebagai nilai absolut, atau membandingkan skor kepercayaan di seluruh pekerjaan pelabelan. Misalnya, jika semua skor kepercayaan antara 0,98 dan 0,998, Anda hanya harus membandingkan objek data satu sama lain dan tidak bergantung pada skor kepercayaan tinggi.

Anda tidak boleh membandingkan skor kepercayaan objek data berlabel manusia dan objek data berlabel otomatis. Skor kepercayaan untuk manusia dihitung menggunakan fungsi konsolidasi anotasi untuk tugas, sedangkan skor kepercayaan untuk pelabelan otomatis dihitung menggunakan model yang menggabungkan fitur objek. Kedua model umumnya memiliki skala yang berbeda dan kepercayaan rata-rata.

Untuk pekerjaan pelabelan kotak pembatas, Ground Truth menghitung skor kepercayaan per kotak. Anda dapat membandingkan skor kepercayaan dalam satu gambar atau di seluruh gambar untuk jenis pelabelan yang sama (manusia atau auto). Anda tidak dapat membandingkan skor kepercayaan di seluruh pekerjaan pelabelan.

Jika pekerja tunggal memberi anotasi tugas (`NumberOfHumanWorkersPerDataObject`diatur ke`1` atau di konsol, Anda memasukkan 1 untuk Jumlah pekerja per objek set data), skor kepercayaan diatur ke`0.00`.

Metadata Pekerja

Ground Truth memberikan informasi yang dapat Anda gunakan untuk melacak pekerja individu dalam data keluaran tugas. Data berikut terletak di direktori di bawah yang`worker-response` terletak di[Direktori Anotasi](#):

- `acceptanceTime`Ini adalah waktu bahwa pekerja menerima tugas. Format cap tanggal dan waktu ini adalah`YYYY-MM-DDTHH:MM:SS.mmmZ` untuk tahun (YYYY), bulan (), hari (MMDD), jam (HH), menit (MM), detik (SS) dan milidetik (mmm). Tanggal dan waktu dipisahkan oleh T.
- Itu`submissionTime` adalah waktu dimana pekerja mengirimkan anotasi mereka menggunakan tombol Submit. Format cap tanggal dan waktu ini adalah`YYYY-MM-DDTHH:MM:SS.mmmZ` untuk tahun (YYYY), bulan (), hari (MMDD), jam (HH), menit (MM), detik (SS) dan milidetik (mmm). Tanggal dan waktu dipisahkan oleh T.
- `timeSpentInSeconds`melaporkan total waktu, dalam hitungan detik, bahwa seorang pekerja secara aktif mengerjakan tugas itu. Metrik ini tidak mencakup waktu ketika pekerja berhenti sejenak atau beristirahat.
- `workerId`Ini unik untuk setiap pekerja.
- Jika Anda menggunakan [tenaga kerja pribadi](#), di`workerMetadata`, Anda melihat berikut ini.
  - `identityProviderType`Ini adalah layanan yang digunakan untuk mengelola tenaga kerja pribadi.
  - `issuer`Itu adalah kumpulan pengguna Cognito atau penerbit Penyedia Identitas OIDC (IdP) yang terkait dengan tim kerja yang ditugaskan untuk tugas peninjauan manusia ini.
  - Sebuah`sub` identifier unik mengacu pada pekerja. Jika Anda membuat tenaga kerja menggunakan Amazon Cognito, Anda dapat mengambil detail tentang pekerja ini (seperti nama atau nama pengguna) menggunakan ID ini menggunakan Amazon Cognito. Untuk mempelajari caranya, lihat [Mengelola dan Mencari Akun Pengguna](#) di [Panduan Pengembang Amazon Cognito](#).

Berikut ini adalah contoh output yang mungkin Anda lihat jika Anda menggunakan Amazon Cognito untuk membuat tenaga kerja pribadi. Hal ini diidentifikasi dalam`identityProviderType`.

```
"submissionTime": "2020-12-28T18:59:58.321Z",
"acceptanceTime": "2020-12-28T18:59:15.191Z",
"timeSpentInSeconds": 40.543,
"workerId": "a12b3cdefg4h5i67",
"workerMetadata": {
    "identityData": {
```

```
        "identityProviderType": "Cognito",
        "issuer": "https://cognito-idp.aws-region.amazonaws.com/aws-region_123456789",
        "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
    }
}
```

Berikut ini adalah contoh dari`workerMetadata` Anda mungkin melihat apakah Anda menggunakan IdP OIDC Anda sendiri untuk membuat tenaga kerja pribadi:

```
"workerMetadata": {
        "identityData": {
            "identityProviderType": "Oidc",
            "issuer": "https://example-oidc-ipd.com/adfs",
            "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
        }
}
```

Untuk mempelajari lebih lanjut tentang menggunakan tenaga kerja pribadi, lihat[Menggunakan Tenaga Kerja Pribadi](#).

Metadata Output

Output dari setiap pekerjaan berisi metadata tentang label yang ditetapkan ke objek data. Unsur-unsur ini sama untuk semua pekerjaan dengan variasi kecil. Contoh berikut menunjukkan elemen metadata:

```
    "confidence": 0.00,
    "type": "groundtruth/image-classification",
    "job-name": "identify-animal-species",
    "human-annotated": "yes",
    "creation-date": "2020-10-18T22:18:13.527256"
```

Unsur-unsur memiliki arti sebagai berikut:

- `confidence`- Keyakinan bahwa Ground Truth memiliki labelnya benar. Untuk informasi selengkapnya, lihat Skor Keyakinan.
- `type`— Jenis pekerjaan klasifikasi. Untuk jenis pekerjaan, lihat[Jenis Tugas Bawaan](#).
- `job-name`- Nama yang ditetapkan untuk pekerjaan ketika dibuat.
- `human-annotated`- Apakah objek data diberi label oleh manusia atau dengan pelabelan data otomatis. Untuk informasi selengkapnya, lihat Pelabelan Data.

- `creation-date`- Tanggal dan waktu label dibuat.

Output

Berikut ini adalah contoh output (file manifes keluaran) dari pekerjaan klasifikasi gambar dan tugas klasifikasi teks. Mereka termasuk label yang Ground Truth ditugaskan ke objek data, nilai untuk label, dan metadata yang menggambarkan label.

Selain elemen metadata standar, metadata untuk pekerjaan klasifikasi mencakup nilai teks kelas label. Untuk informasi selengkapnya, lihat Klasifikasi Gambar - MXNet.

Teks merah dan dicetak miring pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

```
{
    "source-ref":"s3://AWSDOC-EXAMPLE-BUCKET/example_image.jpg",
    "species":"0",
    "species-metadata":
    {
        "class-name": "dog",
        "confidence": 0.00,
        "type": "groundtruth/image-classification",
        "job-name": "identify-animal-species",
        "human-annotated": "yes",
        "creation-date": "2018-10-18T22:18:13.527256"
    }
}
```

```
{
    "source":"The food was delicious",
    "mood":"1",
    "mood-metadata":
    {
        "class-name": "positive",
        "confidence": 0.8,
        "type": "groundtruth/text-classification",
        "job-name": "label-sentiment",
        "human-annotated": "yes",
        "creation-date": "2020-10-18T22:18:13.527256"
    }
}
```

Output Job Klasifikasi Multi-label

Berikut ini adalah contoh keluaran file manifes dari pekerjaan klasifikasi gambar multi-label dan pekerjaan klasifikasi teks multi-label. Mereka menyertakan label yang Ground Truth ditugaskan ke objek data (misalnya, gambar atau potongan teks) dan metadata yang menggambarkan label yang dilihat pekerja saat menyelesaikan tugas pelabelan.

Parameter nama atribut label (misalnya,`image-label-attribute-name`) berisi larik semua label yang dipilih oleh setidaknya satu pekerja yang menyelesaikan tugas ini. Array ini berisi kunci integer (misalnya,`[1,0,8]`) yang sesuai dengan label yang ditemukan di`class-map`. Dalam contoh klasifikasi gambar multi-label`bicycle`,`person`,, dan`clothing` dipilih oleh setidaknya salah satu pekerja yang menyelesaikan tugas pelabelan untuk gambar,`exampleimage.jpg`.

Ini`confidence-map` menunjukkan skor kepercayaan yang diberikan Ground Truth untuk setiap label yang dipilih oleh pekerja. Untuk mempelajari lebih lanjut tentang skor kepercayaan Ground Truth, lihatSkor Keyakinan.

Teks merah dan dicetak miring pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

Berikut ini adalah contoh file manifes keluaran klasifikasi gambar multi-label.

```
{
    "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/example_image.jpg",
    "image-label-attribute-name":[1,0,8],
    "image-label-attribute-name-metadata":
       {
        "job-name":"labeling-job/image-label-attribute-name",
        "class-map":
            {
                "1":"bicycle","0":"person","8":"clothing"
            },
        "human-annotated":"yes",
        "creation-date":"2020-02-27T21:36:25.000201",
        "confidence-map":
            {
                "1":0.95,"0":0.77,"8":0.2
            },
        "type":"groundtruth/image-classification-multilabel"
        }
}
```

Berikut ini adalah contoh file manifes keluaran klasifikasi teks multi-label. Dalam contoh ini`approving`,,`sad` dan`critical` dipilih oleh setidaknya salah satu pekerja yang menyelesaikan tugas pelabelan untuk objek yang`exampletext.txt` ditemukan di`AWSDOC-EXAMPLE-BUCKET`.

```
{
    "source-ref": "AWSDOC-EXAMPLE-BUCKET/text_file.txt",
    "text-label-attribute-name":[1,0,4],
    "text-label-attribute-name-metadata":
       {
        "job-name":"labeling-job/text-label-attribute-name",
        "class-map":
            {
                "1":"approving","0":"sad","4":"critical"
            },
        "human-annotated":"yes",
        "creation-date":"2020-02-20T21:36:25.000201",
        "confidence-map":
            {
                "1":0.95,"0":0.77,"4":0.2
            },
        "type":"groundtruth/text-classification-multilabel"
        }
}
```

Output Job Kotak Bounding

Berikut ini adalah contoh output (output manifest file) dari pekerjaan kotak pembatas. Untuk tugas ini, tiga kotak pembatas dikembalikan. Nilai label berisi informasi tentang ukuran gambar, dan lokasi kotak pembatas.

`class_id`Elemen adalah indeks kelas kotak dalam daftar kelas yang tersedia untuk tugas. Elemen`class-map` metadata berisi teks kelas.

Metadata memiliki skor kepercayaan terpisah untuk setiap kotak pembatas. Metadata juga mencakup`class-map` elemen yang memetakan`class_id` ke nilai teks kelas. Untuk informasi selengkapnya, lihat Deteksi.

Teks merah dan dicetak miring pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

```
{
    "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/example_image.png",
```

```
    "bounding-box-attribute-name":
    {
        "image_size": [{ "width": 500, "height": 400, "depth":3}],
        "annotations":
        [
            {"class_id": 0, "left": 111, "top": 134,
                    "width": 61, "height": 128},
            {"class_id": 5, "left": 161, "top": 250,
                     "width": 30, "height": 30},
            {"class_id": 5, "left": 20, "top": 20,
                     "width": 30, "height": 30}
        ]
    },
    "bounding-box-attribute-name-metadata":
    {
        "objects":
        [
            {"confidence": 0.8},
            {"confidence": 0.9},
            {"confidence": 0.9}
        ],
        "class-map":
        {
            "0": "dog",
            "5": "bone"
        },
        "type": "groundtruth/object-detection",
        "human-annotated": "yes",
        "creation-date": "2018-10-18T22:18:13.527256",
        "job-name": "identify-dogs-and-toys"
    }
}
```

Output dari pekerjaan penyesuaian kotak pembatas terlihat seperti JSON berikut. Perhatikan bahwa JSON asli tetap utuh dan dua pekerjaan baru dicantumkan, masing-masing dengan “adjust-” ditambahkan ke nama atribut asli.

```
{
    "source-ref": "S3 bucket location",
    "bounding-box-attribute-name":
    {
        "image_size": [{ "width": 500, "height": 400, "depth":3}],
        "annotations":
```

```
        [
            {"class_id": 0, "left": 111, "top": 134,
                    "width": 61, "height": 128},
            {"class_id": 5, "left": 161, "top": 250,
                     "width": 30, "height": 30},
            {"class_id": 5, "left": 20, "top": 20,
                     "width": 30, "height": 30}
        ]
    },
    "bounding-box-attribute-name-metadata":
    {
        "objects":
        [
            {"confidence": 0.8},
            {"confidence": 0.9},
            {"confidence": 0.9}
        ],
        "class-map":
        {
            "0": "dog",
            "5": "bone"
        },
        "type": "groundtruth/object-detection",
        "human-annotated": "yes",
        "creation-date": "2018-10-18T22:18:13.527256",
        "job-name": "identify-dogs-and-toys"
    },
    "adjusted-bounding-box":
    {
        "image_size": [{ "width": 500, "height": 400, "depth":3}],
        "annotations":
        [
            {"class_id": 0, "left": 110, "top": 135,
                    "width": 61, "height": 128},
            {"class_id": 5, "left": 161, "top": 250,
                     "width": 30, "height": 30},
            {"class_id": 5, "left": 10, "top": 10,
                     "width": 30, "height": 30}
        ]
    },
    "adjusted-bounding-box-metadata":
    {
        "objects":
        [
```

```
            {"confidence": 0.8},
            {"confidence": 0.9},
            {"confidence": 0.9}
        ],
        "class-map":
        {
            "0": "dog",
            "5": "bone"
        },
        "type": "groundtruth/object-detection",
        "human-annotated": "yes",
        "creation-date": "2018-11-20T22:18:13.527256",
        "job-name": "adjust-bounding-boxes-on-dogs-and-toys",
        "adjustment-status": "adjusted"
    }
}
```

Dalam output ini, pekerjaan`type` tidak berubah, tetapi`adjustment-status` bidang ditambahkan. Bidang ini memiliki nilai`adjusted` atau`unadjusted`. Jika beberapa pekerja telah meninjau objek dan setidaknya satu menyesuaikan label, statusnya adalah`adjusted`.

Dinamakan Entitas Pengakuan

Berikut ini adalah contoh keluaran file manifes dari bernama entity recognition (NER) pelabelan tugas. Untuk tugas ini, tujuh`entities` dikembalikan.

Dalam manifes keluaran, objek JSON`annotations`,, menyertakan daftar`labels` (kategori label) yang Anda berikan.

Tanggapan pekerja berada dalam daftar bernama`entities`. Setiap entitas dalam daftar ini adalah objek JSON yang berisi`label` nilai yang cocok dengan satu dalam`labels` daftar,`startOffset` nilai integer untuk offset Unicode awal rentang berlabel, dan`endOffset` nilai integer untuk offset Unicode akhir.

Metadata memiliki skor kepercayaan terpisah untuk setiap entitas. Jika pekerja tunggal memberi label pada setiap objek data, nilai kepercayaan untuk setiap entitas akan menjadi nol.

Teks merah dan dicetak miring dalam contoh di bawah ini tergantung pada input pekerjaan pelabelan dan tanggapan pekerja.

```
{
```

```
    "source": "Amazon SageMaker is a cloud machine-learning platform that was launched
 in November 2017. SageMaker enables developers to create, train, and deploy machine-
learning (ML) models in the cloud. SageMaker also enables developers to deploy ML
 models on embedded systems and edge-devices",
    "ner-labeling-job-attribute-name": {
        "annotations": {
            "labels": [
                {
                    "label": "Date",
                    "shortDisplayName": "dt"
                },
                {
                    "label": "Verb",
                    "shortDisplayName": "vb"
                },
                {
                    "label": "Thing",
                    "shortDisplayName": "tng"
                },
                {
                    "label": "People",
                    "shortDisplayName": "ppl"
                }
            ],
            "entities": [
                {
                    "label": "Thing",
                    "startOffset": 22,
                    "endOffset": 53
                },
                {
                    "label": "Thing",
                    "startOffset": 269,
                    "endOffset": 281
                },
                {
                    "label": "Verb",
                    "startOffset": 63,
                    "endOffset": 71
                },
                {
                    "label": "Verb",
                    "startOffset": 228,
                    "endOffset": 234
```

```
                },
                {
                    "label": "Date",
                    "startOffset": 75,
                    "endOffset": 88
                },
                {
                    "label": "People",
                    "startOffset": 108,
                    "endOffset": 118
                },
                {
                    "label": "People",
                    "startOffset": 214,
                    "endOffset": 224
                }
            ]
        }
    },
    "ner-labeling-job-attribute-name-metadata": {
        "job-name": "labeling-job/example-ner-labeling-job",
        "type": "groundtruth/text-span",
        "creation-date": "2020-10-29T00:40:39.398470",
        "human-annotated": "yes",
        "entities": [
            {
                "confidence": 0
            },
            {
                "confidence": 0
            },
            {
                "confidence": 0
            },
            {
                "confidence": 0
            },
            {
                "confidence": 0
            },
            {
                "confidence": 0
            },
            {
```

```
                "confidence": 0
            }
        ]
    }
}
```

Verifikasi Label

Output (file manifes keluaran) dari pekerjaan verifikasi kotak pembatas terlihat berbeda dari output dari pekerjaan anotasi kotak pembatas. Itu karena pekerja memiliki jenis tugas yang berbeda. Mereka tidak memberi label objek, tetapi mengevaluasi keakuratan pelabelan sebelumnya, membuat penilaian, dan kemudian memberikan penilaian itu dan mungkin beberapa komentar.

Jika pekerja manusia memverifikasi atau menyesuaikan label kotak pembatas sebelumnya, output dari pekerjaan verifikasi akan terlihat seperti JSON berikut. Teks merah dan dicetak miring pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

```
{
    "source-ref":"s3://AWSDOC-EXAMPLE-BUCKET/image_example.png",
    "bounding-box-attribute-name":
    {
        "image_size": [{ "width": 500, "height": 400, "depth":3}],
        "annotations":
        [
            {"class_id": 0, "left": 111, "top": 134,
                    "width": 61, "height": 128},
            {"class_id": 5, "left": 161, "top": 250,
                     "width": 30, "height": 30},
            {"class_id": 5, "left": 20, "top": 20,
                     "width": 30, "height": 30}
        ]
    },
    "bounding-box-attribute-name-metadata":
    {
        "objects":
        [
            {"confidence": 0.8},
            {"confidence": 0.9},
            {"confidence": 0.9}
        ],
        "class-map":
        {
            "0": "dog",
```

```
            "5": "bone"
        },
        "type": "groundtruth/object-detection",
        "human-annotated": "yes",
        "creation-date": "2018-10-18T22:18:13.527256",
        "job-name": "identify-dogs-and-toys"
    },
    "verify-bounding-box-attribute-name":"1",
    "verify-bounding-box-attribute-name-metadata":
    {
        "class-name": "bad",
        "confidence": 0.93,
        "type": "groundtruth/label-verification",
        "job-name": "verify-bounding-boxes",
        "human-annotated": "yes",
        "creation-date": "2018-11-20T22:18:13.527256",
        "worker-feedback": [
            {"comment": "The bounding box on the bird is too wide on the right side."},
            {"comment": "The bird on the upper right is not labeled."}
        ]
    }
}
```

Meskipuntype pada output kotak pembatas asli adalahgroundtruth/object-detection, yang barutype adalahgroundtruth/label-verification. Perhatikan juga bahwaworker-feedback array menyediakan komentar pekerja. Jika pekerja tidak memberikan komentar, kolom kosong dikecualikan selama konsolidasi.

Output

Berikut ini adalah file manifes keluaran dari pekerjaan pelabelan segmentasi semantik. Nilai label untuk pekerjaan ini adalah referensi ke file PNG dalam bucket Amazon S3.

Selain elemen standar, metadata untuk label menyertakan peta warna yang mendefinisikan warna mana yang digunakan untuk memberi label pada gambar, nama kelas yang terkait dengan warna, dan skor kepercayaan untuk setiap warna. Untuk informasi selengkapnya, lihat Algoritme segmentasi semantik.

Teks merah dan dicetak miring pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

```
{
```

```
    "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/example_city_image.png",
    "city-streets-ref": "S3 bucket location",
    "city-streets-ref-metadata": {
      "internal-color-map": {
        "0": {
           "class-name": "BACKGROUND",
           "confidence": 0.9,
           "hex-color": "#ffffff"
        },
        "1": {
           "class-name": "buildings",
           "confidence": 0.9,
           "hex-color": "#2acf59"
        },
        "2":  {
           "class-name": "road",
           "confidence": 0.9,
           "hex-color": "#f28333"
       }
     },
     "type": "groundtruth/semantic-segmentation",
     "human-annotated": "yes",
     "creation-date": "2018-10-18T22:18:13.527256",
     "job-name": "label-city-streets",
     },
     "verify-city-streets-ref":"1",
     "verify-city-streets-ref-metadata":
     {
        "class-name": "bad",
        "confidence": 0.93,
        "type": "groundtruth/label-verification",
        "job-name": "verify-city-streets",
        "human-annotated": "yes",
        "creation-date": "2018-11-20T22:18:13.527256",
        "worker-feedback": [
            {"comment": "The mask on the leftmost building is assigned the wrong side
 of the road."},
            {"comment": "The curb of the road is not labeled but the instructions say
 otherwise."}
        ]
     }
}
```

Keyakinan dinilai berdasarkan per gambar. Skor kepercayaan sama di semua kelas dalam gambar.

Output pekerjaan penyesuaian segmentasi semantik terlihat serupa dengan JSON berikut.

```
{
    "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/example_city_image.png",
    "city-streets-ref": "S3 bucket location",
    "city-streets-ref-metadata": {
      "internal-color-map": {
        "0": {
           "class-name": "BACKGROUND",
           "confidence": 0.9,
           "hex-color": "#ffffff"
        },
        "1": {
           "class-name": "buildings",
           "confidence": 0.9,
           "hex-color": "#2acf59"
        },
        "2":  {
           "class-name": "road",
           "confidence": 0.9,
           "hex-color": "#f28333"
       }
     },
     "type": "groundtruth/semantic-segmentation",
     "human-annotated": "yes",
     "creation-date": "2018-10-18T22:18:13.527256",
     "job-name": "label-city-streets",
     },
     "adjusted-city-streets-ref": "s3://AWSDOC-EXAMPLE-BUCKET/example_city_image.png",
     "adjusted-city-streets-ref-metadata": {
      "internal-color-map": {
        "0": {
           "class-name": "BACKGROUND",
           "confidence": 0.9,
           "hex-color": "#ffffff"
        },
        "1": {
           "class-name": "buildings",
           "confidence": 0.9,
           "hex-color": "#2acf59"
        },
        "2":  {
```

```
            "class-name": "road",
            "confidence": 0.9,
            "hex-color": "#f28333"
        }
      },
      "type": "groundtruth/semantic-segmentation",
      "human-annotated": "yes",
      "creation-date": "2018-11-20T22:18:13.527256",
      "job-name": "adjust-label-city-streets",
      }
}
```

Output Deteksi Objek Bingkai Video

Berikut ini adalah file manifes keluaran dari pekerjaan pelabelan deteksi objek bingkai video. *Teks merah dan dicetak miring* pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

Selain elemen standar, metadata menyertakan peta kelas yang mencantumkan setiap kelas yang memiliki setidaknya satu label dalam urutan. Metadata juga mencakup nama`job-name` yang Anda tetapkan untuk pekerjaan pelabelan. Untuk tugas penyesuaian, Jika satu atau beberapa kotak pembatas diubah, ada`adjustment-status` parameter dalam metadata untuk alur kerja audit yang disetel ke`adjusted`.

```
{
    "source-ref": "s3://DOC-EXAMPLE-BUCKET/example-path/input-manifest.json",
    "CarObjectDetection-ref": "s3://AWSDOC-EXAMPLE-BUCKET/output/labeling-job-name/
annotations/consolidated-annotation/output/0/SeqLabel.json",
    "CarObjectDetection-ref-metadata": {
        "class-map": {
            "0": "car",
            "1": "bus"
        },
        "job-name": "labeling-job/labeling-job-name",
        "human-annotated": "yes",
        "creation-date": "2021-09-29T05:50:35.566000",
        "type": "groundtruth/video-object-detection"
        }
}
```

Ground Truth Kebenaran menciptakan satu berkas urutan output untuk setiap urutan frame video yang diberi label. Setiap file urutan output berisi berikut:

- Semua anotasi untuk semua frame secara berurutan dalam`detection-annotations` daftar objek JSON.
- Untuk setiap frame yang dianotasi oleh pekerja, nama file frame (`frame`), number (`frame-no`), daftar objek JSON yang berisi anotasi (`annotations`), dan jika berlaku,`frame-attributes`. Nama daftar ini didefinisikan oleh jenis tugas yang Anda gunakan:`polylines`,,`polygons``keypoints`, dan untuk kotak pembatas,`annotations`.

  Setiap objek JSON berisi informasi tentang anotasi tunggal dan label terkait. Tabel berikut menguraikan parameter yang akan Anda lihat untuk setiap jenis tugas bingkai video.

Jenis Tugas	Parameter
Kotak pembatas	Dimensi kotak:`height` dan`width` Kotak atas, lokasi piksel sudut kiri:`top` dan`left`
Keypoint	Simpul keypoint:`{ "x": int, "y": int }`
Polygon	Daftar simpul poligon:`vertices` Simpul poligon:`{ "x": int, "y": int }` Sebuah poligon adalah bentuk tertutup dan jadi titik pertama juga akan mewakili titik terakhir.
Polyline	Daftar simpul polyline:`vertices` Simpul polyline:`{ "x": int, "y": int }`

  Selain nilai spesifik tipe tugas, Anda akan melihat yang berikut di setiap objek JSON:

  - Nilai dari setiap`label-category-attributes` yang ditentukan untuk label itu.
  - `class-id`Dari kotak. Gunakan file manifes keluaran untuk melihat kategori label yang dipetakan ID ini.`class-map`

Berikut ini adalah contoh`SeqLabel.json` file dari bounding box bingkai video objek deteksi pekerjaan pelabelan. File ini akan berlokasi di bawah`s3://your-output-bucket/output-prefix/annotations/consolidated-annotation/output/annotation-number/`

```
{
    "detection-annotations": [
        {
            "annotations": [
                {
                    "height": 41,
                    "width": 53,
                    "top": 152,
                    "left": 339,
                    "class-id": "1",
                    "label-category-attributes": {
                        "occluded": "no",
                        "size": "medium"
                    }
                },
                {
                    "height": 24,
                    "width": 37,
                    "top": 148,
                    "left": 183,
                    "class-id": "0",
                    "label-category-attributes": {
                        "occluded": "no",
                    }
                }
            ],
            "frame-no": 0,
            "frame": "frame_0000.jpeg",
            "frame-attributes": {name: value, name: value}
        },
        {
            "annotations": [
                {
                    "height": 41,
                    "width": 53,
                    "top": 152,
                    "left": 341,
                    "class-id": "0",
                    "label-category-attributes": {}
```

```
                },
                {
                    "height": 24,
                    "width": 37,
                    "top": 141,
                    "left": 177,
                    "class-id": "0",
                    "label-category-attributes": {
                        "occluded": "no",
                    }
                }
            ],
            "frame-no": 1,
            "frame": "frame_0001.jpeg",
            "frame-attributes": {name: value, name: value}
        }
    ]
}
```

Output Pelacakan Objek Bingkai Video

Berikut ini adalah file manifes keluaran dari pekerjaan pelabelan pelacakan objek bingkai video. *Teks merah dan dicetak miring* pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

Selain elemen standar, metadata menyertakan peta kelas yang mencantumkan setiap kelas yang memiliki setidaknya satu label dalam urutan bingkai. Metadata juga mencakup nama`job-name` yang Anda tetapkan untuk pekerjaan pelabelan. Untuk tugas penyesuaian, Jika satu atau beberapa kotak pembatas diubah, ada`adjustment-status` parameter dalam metadata untuk alur kerja audit yang disetel ke`adjusted`.

```
{
    "source-ref": "s3://DOC-EXAMPLE-BUCKET/example-path/input-manifest.json",
    "CarObjectTracking-ref": "s3://AWSDOC-EXAMPLE-BUCKET/output/labeling-job-name/
annotations/consolidated-annotation/output/0/SeqLabel.json",
    "CarObjectTracking-ref-metadata": {
        "class-map": {
            "0": "car",
            "1": "bus"
        },
        "job-name": "labeling-job/labeling-job-name",
        "human-annotated": "yes",
```

```
        "creation-date": "2021-09-29T05:50:35.566000",
        "type": "groundtruth/video-object-tracking"
        }
}
```

Ground Truth Kebenaran menciptakan satu berkas urutan output untuk setiap urutan frame video yang diberi label. Setiap file urutan output berisi berikut:

- Semua anotasi untuk semua frame secara berurutan dalam`tracking-annotations` daftar objek JSON.
- Untuk setiap frame yang dianotasi oleh pekerja, frame (`frame`), number (`frame-no`), daftar objek JSON yang berisi anotasi (`annotations`), dan jika berlaku, atribut bingkai (`frame-attributes`). Nama daftar ini didefinisikan oleh jenis tugas yang Anda gunakan:`polylines`,,`polygons``keypoints`, dan untuk kotak pembatas,`annotations`.

  Setiap objek JSON berisi informasi tentang anotasi tunggal dan label terkait. Tabel berikut menguraikan parameter yang akan Anda lihat untuk setiap jenis tugas bingkai video.

Jenis Tugas	Parameter
Kotak pembatas	Dimensi kotak:`height` dan`width` Kotak atas, lokasi piksel sudut kiri:`top` dan`left`
Keypoint	Simpul keypoint:`{ "x": int, "y": int }`
Polygon	Daftar simpul poligon:`vertices` Simpul poligon:`{ "x": int, "y": int }` Sebuah poligon adalah bentuk tertutup dan jadi titik pertama juga akan mewakili titik terakhir.
Polyline	Daftar simpul polyline:`vertices` Simpul polyline:`{ "x": int, "y": int }`

Selain nilai spesifik tipe tugas, Anda akan melihat yang berikut di setiap objek JSON:

- Nilai dari setiap`label-category-attributes` yang ditentukan untuk label itu.
- `class-id`Dari kotak. Gunakan file manifes keluaran untuk melihat kategori label yang dipetakan ID ini.`class-map`
- Sebuah`object-id` yang mengidentifikasi sebuah instance dari label. ID ini akan sama di seluruh frame jika pekerja mengidentifikasi instance objek yang sama dalam beberapa frame. Misalnya, jika mobil muncul dalam beberapa bingkai, semua kotak pembatas digunakan untuk mengidentifikasi mobil itu akan memiliki hal yang sama`object-id`.
- Yang`object-name` merupakan contoh ID dari anotasi itu.

Berikut ini adalah contoh`SeqLabel.json` file dari bounding box video frame objek pelacakan pekerjaan pelabelan. File ini akan berlokasi di bawah`s3://`*`your-output-bucket`*`/`*`output-prefix`*`/annotations/consolidated-annotation/output/`*`annotation-number`*`/`

```
{
    "tracking-annotations": [
        {
            "annotations": [
                {
                    "height": 36,
                    "width": 46,
                    "top": 178,
                    "left": 315,
                    "class-id": "0",
                    "label-category-attributes": {
                        "occluded": "no"
                    },
                    "object-id": "480dc450-c0ca-11ea-961f-a9b1c5c97972",
                    "object-name": "car:1"
                }
            ],
            "frame-no": 0,
            "frame": "frame_0001.jpeg",
            "frame-attributes": {}
        },
        {
            "annotations": [
                {
                    "height": 30,
                    "width": 47,
                    "top": 163,
```

```
                    "left": 344,
                    "class-id": "1",
                    "label-category-attributes": {
                        "occluded": "no",
                        "size": "medium"
                    },
                    "object-id": "98f2b0b0-c0ca-11ea-961f-a9b1c5c97972",
                    "object-name": "bus:1"
                },
                {
                    "height": 28,
                    "width": 33,
                    "top": 150,
                    "left": 192,
                    "class-id": "0",
                    "label-category-attributes": {
                        "occluded": "partially"
                    },
                    "object-id": "480dc450-c0ca-11ea-961f-a9b1c5c97972",
                    "object-name": "car:1"
                }
            ],
            "frame-no": 1,
            "frame": "frame_0002.jpeg",
            "frame-attributes": {name: value, name: value}
        }
    ]
}
```

Output Segmentasi Semantik Awan Titik 3D

Berikut ini adalah file manifes keluaran dari pekerjaan pelabelan segmentasi semantik titik 3D.

Selain elemen standar, metadata untuk label menyertakan peta warna yang mendefinisikan warna mana yang digunakan untuk memberi label pada gambar, nama kelas yang terkait dengan warna, dan skor kepercayaan untuk setiap warna. Selain itu, ada`adjustment-status` parameter dalam metadata untuk alur kerja audit yang diatur ke`adjusted` jika topeng warna dimodifikasi. Jika Anda menambahkan satu atau lebih`frameAttributes` ke file konfigurasi kategori label, respons pekerja untuk atribut frame ada di objek JSON`dataset-object-attributes`.

`your-label-attribute``-ref`Parameter berisi lokasi file terkompresi dengan ekstensi.zlib. Ketika Anda uncompress file ini, itu berisi array. Setiap indeks dalam array sesuai dengan indeks

titik anotasi di awan titik input. Nilai array pada indeks tertentu memberikan kelas titik pada indeks yang sama di titik awan, berdasarkan peta warna semantik yang ditemukan dalam`color-map` parameter`metadata`.

Anda dapat menggunakan kode Python yang serupa dengan yang berikut ini untuk mendekompresi file.zlib:

```
import zlib
from array import array

# read the label file
compressed_binary_file = open(zlib_file_path/file.zlib, 'rb').read()

# uncompress the label file
binary_content = zlib.decompress(compressed_binary_file)

# load labels to an array
my_int_array_data = array('B', binary_content);

print(my_int_array_data)
```

Blok kode di atas akan menghasilkan output yang serupa dengan yang berikut ini. Setiap elemen dari array dicetak berisi kelas titik pada indeks yang di titik awan. Misalnya,`my_int_array_data[0] = 1` berartipoint`[0]` di titik input awan memiliki kelas`1`. Dalam contoh file manifes keluaran berikut, kelas`0` sesuai dengan`"Background"`,`1` dengan`Car`, dan`2` dengan`Pedestrian`.

```
>> array('B', [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

Berikut ini adalah contoh file manifest keluaran keluaran awan titik 3D semantik. Teks merah dan dicetak miring pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran.

```
{
   "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/examplefolder/frame1.bin",
   "source-ref-metadata":{
      "format": "binary/xyzi",
      "unix-timestamp": 1566861644.759115,
      "ego-vehicle-pose":{...},
      "prefix": "s3://AWSDOC-EXAMPLE-BUCKET/lidar_singleframe_dataset/prefix",
```

```
      "images": [{...}]
    },
    "lidar-ss-label-attribute-ref": "s3://your-output-bucket/labeling-job-name/
annotations/consolidated-annotation/output/dataset-object-id/filename.zlib",
    "lidar-ss-label-attribute-ref-metadata": {
        'color-map': {
            "0": {
                "class-name": "Background",
                "hex-color": "#ffffff",
                "confidence": 0.00
            },
            "1": {
                "class-name": "Car",
                "hex-color": "#2ca02c",
                "confidence": 0.00
            },
            "2": {
                "class-name": "Pedestrian",
                "hex-color": "#1f77b4",
                "confidence": 0.00
            },
            "3": {
                "class-name": "Tree",
                "hex-color": "#ff7f0e",
                "confidence": 0.00
            }
        },
        'type': 'groundtruth/point_cloud_single_frame_semantic_segmentation',
        'human-annotated': 'yes',
        'creation-date': '2019-11-12T01:18:14.271944',
        'job-name': 'labeling-job-name',
        //only present for adjustment audit workflow
        "adjustment-status": "adjusted", // "adjusted" means the label was adjusted
        "dataset-object-attributes": {name: value, name: value}
    }
}
```

Output Deteksi Objek Awan Titik 3D

Berikut ini adalah output sampel dari pekerjaan deteksi keberatan awan titik 3D. Untuk jenis tugas ini, data tentang kubus 3D dikembalikan dalam`3d-bounding-box` parameter, dalam daftar bernama`annotations`. Dalam daftar ini, setiap kubus 3D dijelaskan menggunakan informasi berikut.

- Setiap kelas, atau kategori label, yang Anda tentukan dalam manifes masukan dikaitkan dengan `class-id`. Gunakan`class-map` untuk mengidentifikasi kelas yang terkait dengan setiap ID kelas.
- Kelas-kelas ini digunakan untuk memberikan setiap kubus 3D`object-name` dalam format`<class>:<integer>` di mana`integer` adalah nomor unik untuk mengidentifikasi bahwa berbentuk kubus dalam bingkai.
- `center-x`,`center-y`, dan`center-z` merupakan koordinat pusat berbentuk kubus, dalam sistem koordinat yang sama dengan data input cloud titik 3D yang digunakan dalam pekerjaan pelabelan Anda.
- `length`,`width`, dan`height` menggambarkan dimensi berbentuk kubus.
- `yaw`digunakan untuk menggambarkan orientasi (heading) berbentuk kubus dalam radian.

> **Note**
>
> `yaw`Sekarang dalam sistem Cartesian tangan kanan. Karena fitur ini ditambahkan pada 02 September 2022 19:02:17 UTC, Anda dapat mengonversi`yaw` pengukuran dalam data keluaran sebelum menggunakan yang berikut (semua unit dalam radian):
>
> ```
> old_yaw_in_output = pi - yaw
> ```

- Dalam definisi kami, +x ada di sebelah kanan, +y ke depan, dan+z naik dari bidang tanah. Urutan rotasi adalah x - y - z. Itu`roll`,`pitch` dan`yaw` diwakili dalam sistem Cartesian tangan kanan. Dalam ruang 3D,`roll` adalah sepanjang sumbu x,`pitch` adalah sepanjang sumbu y dan`yaw` sepanjang sumbu z. Ketiganya berlawanan arah jarum jam.
- Jika Anda menyertakan atribut label dalam file manifes masukan untuk kelas tertentu,`label-category-attributes` parameter disertakan untuk semua cuboid untuk atribut label yang dipilih pekerja.

Jika satu atau beberapa kuboid dimodifikasi, ada`adjustment-status` parameter dalam metadata untuk alur kerja audit yang disetel ke`adjusted`. Jika Anda menambahkan satu atau lebih`frameAttributes` ke file konfigurasi kategori label, respons pekerja untuk atribut frame ada di objek JSON`dataset-object-attributes`.

`Teks merah dan dicetak miring` pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran. Elips (`...`) menunjukkan kelanjutan dari daftar itu, di mana objek tambahan dengan format yang sama dengan objek yang melanjutkan dapat muncul.

```
{
   "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/examplefolder/frame1.txt",
   "source-ref-metadata":{
      "format": "text/xyzi",
      "unix-timestamp": 1566861644.759115,
      "prefix": "s3://AWSDOC-EXAMPLE-BUCKET/lidar_singleframe_dataset/prefix",
      "ego-vehicle-pose": {
            "heading": {
                "qx": -0.02111296123795955,
                "qy": -0.006495469416730261,
                "qz": -0.008024565904865688,
                "qw": 0.9997181192298087
            },
            "position": {
                "x": -2.7161461413869947,
                "y": 116.25822288149078,
                "z": 1.8348751887989483
            }
       },
       "images": [
            {
                "fx": 847.7962624528487,
                "fy": 850.0340893791985,
                "cx": 576.2129134707038,
                "cy": 317.2423573573745,
                "k1": 0,
                "k2": 0,
                "k3": 0,
                "k4": 0,
                "p1": 0,
                "p2": 0,
                "skew": 0,
                "unix-timestamp": 1566861644.759115,
                "image-path": "images/frame_0_camera_0.jpg",
                "position": {
                    "x": -2.2722515189268138,
                    "y": 116.86003310568965,
                    "z": 1.454614668542299
                },
                "heading": {
                    "qx": 0.7594754093069037,
                    "qy": 0.02181790885672969,
                    "qz": -0.02461725233103356,
```

```
                    "qw": -0.6496916273040025
                },
                "camera_model": "pinhole"
            }
        ]
    },
    "3d-bounding-box":
    {
        "annotations": [
            {
                "label-category-attributes": {
                    "Occlusion": "Partial",
                    "Type": "Sedan"
                },
                "object-name": "Car:1",
                "class-id": 0,
                "center-x": -2.616382013657516,
                "center-y": 125.04149850484193,
                "center-z": 0.311272296465834,
                "length": 2.993000265181146,
                "width": 1.8355260519692056,
                "height": 1.3233490884304047,
                "roll": 0,
                "pitch": 0,
                "yaw": 1.6479308313703527
            },
            {
                "label-category-attributes": {
                    "Occlusion": "Partial",
                    "Type": "Sedan"
                },
                "object-name": "Car:2",
                "class-id": 0,
                "center-x": -5.188984560617168,
                "center-y": 99.7954483288783,
                "center-z": 0.2226435567445657,
                "length": 4,
                "width": 2,
                "height": 2,
                "roll": 0,
                "pitch": 0,
                "yaw": 1.6243170732068055
            }
        ]
```

```
    },
    "3d-bounding-box-metadata":
    {
        "objects": [],
        "class_map":
        {
            "0": "Car",
        },
        "type": "groundtruth/point_cloud_object_detection",
        "human-annotated": "yes",
        "creation-date": "2018-10-18T22:18:13.527256",
        "job-name": "identify-3d-objects",
        "adjustment-status": "adjusted",
        "dataset-object-attributes": {name: value, name: value}
    }
}
```

Output Pelacakan Objek Awan Titik 3D

Berikut ini adalah contoh file manifes keluaran dari tugas pelabelan pelacakan objek awan titik 3D. *Teks merah dan dicetak miring* pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran. Elips (`...`) menunjukkan kelanjutan dari daftar itu, di mana objek tambahan dengan format yang sama dengan objek yang melanjutkan dapat muncul.

Selain elemen standar, metadata menyertakan peta kelas yang mencantumkan setiap kelas yang memiliki setidaknya satu label dalam urutan. Jika satu atau beberapa kuboid dimodifikasi, ada`adjustment-status` parameter dalam metadata untuk alur kerja audit yang disetel ke`adjusted`.

```
{
   "source-ref": "s3://AWSDOC-EXAMPLE-BUCKET/myfolder/seq1.json",
    "lidar-label-attribute-ref": "s3://<CustomerOutputLocation>/<labelingJobName>/
annotations/consolidated-annotation/output/<datasetObjectId>/SeqLabel.json",
    "lidar-label-attribute-ref-metadata": {
        "objects":
        [
            {
                "frame-no": 300,
                "confidence": []
            },
            {
                "frame-no": 301,
```

```
                "confidence": []
            },
            ...
        ],
        'class-map': {'0': 'Car', '1': 'Person'},
        'type': 'groundtruth/point_cloud_object_tracking',
        'human-annotated': 'yes',
        'creation-date': '2019-11-12T01:18:14.271944',
        'job-name': 'identify-3d-objects',
        "adjustment-status": "adjusted"
    }
}
```

Dalam contoh di atas, data berbentuk kubus untuk setiap frameseq1.json beradaSeqLabel.json di lokasi Amazon S3,s3://*<customerOutputLocation>*/*<labelingJobName>*/annotations/consolidated-annotation/output/*<datasetObjectId>*/SeqLabel.json. Berikut ini adalah contoh file urutan label ini.

Untuk setiap frame dalam urutan, Anda melihatframe-number,frame-name, jika berlakuframe-attributes,, dan daftarannotations. Daftar ini berisi kubiod 3D yang digambar untuk bingkai itu. Setiap anotasi mencakup informasi berikut:

- Sebuahobject-name dalam format<class>:<integer> di manaclass mengidentifikasi kategori label daninteger ID unik di seluruh dataset.
- Ketika pekerja menggambar berbentuk kubus, hal ini terkait dengan unikobject-id yang dikaitkan dengan semua kubus yang mengidentifikasi objek yang sama di beberapa frame.
- Setiap kelas, atau kategori label, yang Anda tentukan dalam manifes masukan dikaitkan dengan aclass-id. Gunakanclass-map untuk mengidentifikasi kelas yang terkait dengan setiap ID kelas.
- center-x,center-y, dancenter-z merupakan koordinat pusat berbentuk kubus, dalam sistem koordinat yang sama dengan data input cloud titik 3D yang digunakan dalam pekerjaan pelabelan Anda.
- lengthwidth,, danheight menggambarkan dimensi berbentuk kubus.
- yawdigunakan untuk menggambarkan orientasi (heading) berbentuk kubus dalam radian.

> **Note**
>
> `yaw`Sekarang dalam sistem Cartesian tangan kanan. Karena fitur ini ditambahkan pada 02 September 2022 19:02:17 UTC, Anda dapat mengonversi`yaw` pengukuran dalam data keluaran sebelum menggunakan yang berikut (semua unit dalam radian):
>
> ```
> old_yaw_in_output = pi - yaw
> ```

- Dalam definisi kami, +x ada di sebelah kanan, +y ke depan, dan+z naik dari bidang tanah. Urutan rotasi adalah x - y - z. Ituroll,`pitch` dan`yaw` diwakili dalam sistem Cartesian tangan kanan. Dalam ruang 3D,`roll` adalah sepanjang sumbu x,`pitch` adalah sepanjang sumbu y dan`yaw` sepanjang sumbu z. Ketiganya berlawanan arah jarum jam.
- Jika Anda menyertakan atribut label dalam file manifes masukan untuk kelas tertentu,`label-category-attributes` parameter disertakan untuk semua cuboid untuk atribut label yang dipilih pekerja.

```
{
    "tracking-annotations": [
        {
            "frame-number": 0,
            "frame-name": "0.txt.pcd",
            "frame-attributes": {name: value, name: value},
            "annotations": [
                {
                    "label-category-attributes": {},
                    "object-name": "Car:4",
                    "class-id": 0,
                    "center-x": -2.2906369208300674,
                    "center-y": 103.73924823843463,
                    "center-z": 0.37634114027023313,
                    "length": 4,
                    "width": 2,
                    "height": 2,
                    "roll": 0,
                    "pitch": 0,
                    "yaw": 1.5827222214406014,
                    "object-id": "ae5dc770-a782-11ea-b57d-67c51a0561a1"
                },
```

```
            {
                "label-category-attributes": {
                    "Occlusion": "Partial",
                    "Type": "Sedan"
                },
                "object-name": "Car:1",
                "class-id": 0,
                "center-x": -2.6451293634707413,
                "center-y": 124.9534455706848,
                "center-z": 0.5020834081743839,
                "length": 4,
                "width": 2,
                "height": 2.080488827301309,
                "roll": 0,
                "pitch": 0,
                "yaw": -1.5963335581398077,
                "object-id": "06efb020-a782-11ea-b57d-67c51a0561a1"
            },
            {
                "label-category-attributes": {
                    "Occlusion": "Partial",
                    "Type": "Sedan"
                },
                "object-name": "Car:2",
                "class-id": 0,
                "center-x": -5.205611313118477,
                "center-y": 99.91731932137061,
                "center-z": 0.22917217081212138,
                "length": 3.8747142207671956,
                "width": 1.9999999999999918,
                "height": 2,
                "roll": 0,
                "pitch": 0,
                "yaw": 1.5672228760316775,
                "object-id": "26fad020-a782-11ea-b57d-67c51a0561a1"
            }
        ]
    },
    {
        "frame-number": 1,
        "frame-name": "1.txt.pcd",
        "frame-attributes": {},
        "annotations": [
            {
```

```
            "label-category-attributes": {},
            "object-name": "Car:4",
            "class-id": 0,
            "center-x": -2.2906369208300674,
            "center-y": 103.73924823843463,
            "center-z": 0.37634114027023313,
            "length": 4,
            "width": 2,
            "height": 2,
            "roll": 0,
            "pitch": 0,
            "yaw": 1.5827222214406014,
            "object-id": "ae5dc770-a782-11ea-b57d-67c51a0561a1"
        },
        {
            "label-category-attributes": {
                "Occlusion": "Partial",
                "Type": "Sedan"
            },
            "object-name": "Car:1",
            "class-id": 0,
            "center-x": -2.6451293634707413,
            "center-y": 124.9534455706848,
            "center-z": 0.5020834081743839,
            "length": 4,
            "width": 2,
            "height": 2.080488827301309,
            "roll": 0,
            "pitch": 0,
            "yaw": -1.5963335581398077,
            "object-id": "06efb020-a782-11ea-b57d-67c51a0561a1"
        },
        {
            "label-category-attributes": {
                "Occlusion": "Partial",
                "Type": "Sedan"
            },
            "object-name": "Car:2",
            "class-id": 0,
            "center-x": -5.221311072916759,
            "center-y": 100.4639841045424,
            "center-z": 0.22917217081212138,
            "length": 3.8747142207671956,
            "width": 1.9999999999999918,
```

```
                    "height": 2,
                    "roll": 0,
                    "pitch": 0,
                    "yaw": 1.5672228760316775,
                    "object-id": "26fad020-a782-11ea-b57d-67c51a0561a1"
                }
            ]
        }
    ]
}
```

Titik Pelacakan Objek 3D-2D Output Pelacakan Objek Cloud

Berikut ini adalah contoh file manifes keluaran dari tugas pelabelan pelacakan objek awan titik 3D. *Teks merah dan dicetak miring* pada contoh di bawah ini tergantung pada spesifikasi pekerjaan pelabelan dan data keluaran. Elips (`...`) menunjukkan kelanjutan dari daftar itu, di mana objek tambahan dengan format yang sama dengan objek yang melanjutkan dapat muncul.

Selain elemen standar, metadata menyertakan peta kelas yang mencantumkan setiap kelas yang memiliki setidaknya satu label dalam urutan. Jika satu atau beberapa kuboid dimodifikasi, ada`adjustment-status` parameter dalam metadata untuk alur kerja audit yang disetel ke`adjusted`.

```
{
  "source-ref": "s3://iad-groundtruth-lidar-test-bucket/artifacts/gt-point-cloud-demos/
sequences/seq2.json",
  "source-ref-metadata": {
    "json-paths": [
      "number-of-frames",
      "prefix",
      "frames{frame-no, frame}"
    ]
  },
  "3D2D-linking-ref": "s3://iad-groundtruth-lidar-test-bucket/xyz/3D2D-linking/
annotations/consolidated-annotation/output/0/SeqLabel.json",
  "3D2D-linking-ref-metadata": {
    "objects": [
      {
        "frame-no": 0,
        "confidence": []
      },
      {
```

```
        "frame-no": 1,
        "confidence": []
      },
      {
        "frame-no": 2,
        "confidence": []
      },
      {
        "frame-no": 3,
        "confidence": []
      },
      {
        "frame-no": 4,
        "confidence": []
      },
      {
        "frame-no": 5,
        "confidence": []
      },
      {
        "frame-no": 6,
        "confidence": []
      },
      {
        "frame-no": 7,
        "confidence": []
      },
      {
        "frame-no": 8,
        "confidence": []
      },
      {
        "frame-no": 9,
        "confidence": []
      }
    ],
    "class-map": {
      "0": "Car"
    },
    "type": "groundtruth/point_cloud_object_tracking",
    "human-annotated": "yes",
    "creation-date": "2023-01-19T02:55:10.206508",
    "job-name": "mcm-linking"
  },
```

```
  "3D2D-linking-chain-ref": "s3://iad-groundtruth-lidar-test-bucket/xyz/3D2D-linking-chain/annotations/consolidated-annotation/output/0/SeqLabel.json",
  "3D2D-linking-chain-ref-metadata": {
    "objects": [
      {
        "frame-no": 0,
        "confidence": []
      },
      {
        "frame-no": 1,
        "confidence": []
      },
      {
        "frame-no": 2,
        "confidence": []
      },
      {
        "frame-no": 3,
        "confidence": []
      },
      {
        "frame-no": 4,
        "confidence": []
      },
      {
        "frame-no": 5,
        "confidence": []
      },
      {
        "frame-no": 6,
        "confidence": []
      },
      {
        "frame-no": 7,
        "confidence": []
      },
      {
        "frame-no": 8,
        "confidence": []
      },
      {
        "frame-no": 9,
        "confidence": []
      }
```

```
    ],
    "class-map": {
      "0": "Car"
    },
    "type": "groundtruth/point_cloud_object_tracking",
    "human-annotated": "yes",
    "creation-date": "2023-01-19T03:29:49.149935",
    "job-name": "3d2d-linking-chain"
  }
}
```

Dalam contoh di atas, data berbentuk kubus untuk setiap frameseq2.json beradaSeqLabel.json di lokasi Amazon S3,s3://<customerOutputLocation>/<labelingJobName>/annotations/consolidated-annotation/output/<datasetObjectId>/SeqLabel.json. Berikut ini adalah contoh file urutan label ini.

Untuk setiap frame dalam urutan, Anda melihatframe-number,frame-name, jika berlakuframe-attributes,, dan daftarannotations. Daftar ini berisi kubiod 3D yang digambar untuk bingkai itu. Setiap anotasi mencakup informasi berikut:

- Sebuahobject-name dalam format<class>:<integer> di manaclass mengidentifikasi kategori label daninteger ID unik di seluruh dataset.
- Ketika pekerja menggambar berbentuk kubus, hal ini terkait dengan unikobject-id yang dikaitkan dengan semua kubus yang mengidentifikasi objek yang sama di beberapa frame.
- Setiap kelas, atau kategori label, yang Anda tentukan dalam manifes masukan dikaitkan dengan aclass-id. Gunakanclass-map untuk mengidentifikasi kelas yang terkait dengan setiap ID kelas.
- center-x,center-y, dancenter-z merupakan koordinat pusat berbentuk kubus, dalam sistem koordinat yang sama dengan data input cloud titik 3D yang digunakan dalam pekerjaan pelabelan Anda.
- lengthwidth,, danheight menggambarkan dimensi berbentuk kubus.
- yawdigunakan untuk menggambarkan orientasi (heading) berbentuk kubus dalam radian.

> **Note**
>
> yawSekarang dalam sistem Cartesian tangan kanan. Karena fitur ini ditambahkan pada 02 September 2022 19:02:17 UTC, Anda dapat mengonversiyaw pengukuran dalam data keluaran sebelum menggunakan yang berikut (semua unit dalam radian):

```
old_yaw_in_output = pi - yaw
```

- Dalam definisi kami, +x ada di sebelah kanan, +y ke depan, dan+z naik dari bidang tanah. Urutan rotasi adalah x - y - z. Ituroll,pitch danyaw diwakili dalam sistem Cartesian tangan kanan. Dalam ruang 3D,roll adalah sepanjang sumbu x,pitch adalah sepanjang sumbu y danyaw sepanjang sumbu z. Ketiganya berlawanan arah jarum jam.
- Jika Anda menyertakan atribut label dalam file manifes masukan untuk kelas tertentu,label-category-attributes parameter disertakan untuk semua cuboid untuk atribut label yang dipilih pekerja.


```
{
  "lidar": {
    "tracking-annotations": [
      {
        "frame-number": 0,
        "frame-name": "0.txt.pcd",
        "annotations": [
          {
            "label-category-attributes": {
              "Type": "Sedan"
            },
            "object-name": "Car:1",
            "class-id": 0,
            "center-x": 12.172361721602815,
            "center-y": 120.23067521992364,
            "center-z": 1.590525771183712,
            "length": 4,
            "width": 2,
            "height": 2,
            "roll": 0,
            "pitch": 0,
            "yaw": 0,
            "object-id": "505b39e0-97a4-11ed-8903-dd5b8b903715"
          },
          {
            "label-category-attributes": {},
            "object-name": "Car:4",
            "class-id": 0,
            "center-x": 17.192725195301094,
```

```
            "center-y": 114.55705365827872,
            "center-z": 1.590525771183712,
            "length": 4,
            "width": 2,
            "height": 2,
            "roll": 0,
            "pitch": 0,
            "yaw": 0,
            "object-id": "1afcb670-97a9-11ed-9a84-ff627d099e16"
          }
        ],
        "frame-attributes": {}
      },
      {
        "frame-number": 1,
        "frame-name": "1.txt.pcd",
        "annotations": [
          {
            "label-category-attributes": {
              "Type": "Sedan"
            },
            "object-name": "Car:1",
            "class-id": 0,
            "center-x": -1.6841480600695489,
            "center-y": 126.20198882749516,
            "center-z": 1.590525771183712,
            "length": 4,
            "width": 2,
            "height": 2,
            "roll": 0,
            "pitch": 0,
            "yaw": 0,
            "object-id": "505b39e0-97a4-11ed-8903-dd5b8b903715"
          },
          {
            "label-category-attributes": {},
            "object-name": "Car:4",
            "class-id": 0,
            "center-x": 17.192725195301094,
            "center-y": 114.55705365827872,
            "center-z": 1.590525771183712,
            "length": 4,
            "width": 2,
            "height": 2,
```

```
            "roll": 0,
            "pitch": 0,
            "yaw": 0,
            "object-id": "1afcb670-97a9-11ed-9a84-ff627d099e16"
          }
        ],
        "frame-attributes": {}
      },
      {
        "frame-number": 2,
        "frame-name": "2.txt.pcd",
        "annotations": [
          {
            "label-category-attributes": {
              "Type": "Sedan"
            },
            "object-name": "Car:1",
            "class-id": 0,
            "center-x": -1.6841480600695489,
            "center-y": 126.20198882749516,
            "center-z": 1.590525771183712,
            "length": 4,
            "width": 2,
            "height": 2,
            "roll": 0,
            "pitch": 0,
            "yaw": 0,
            "object-id": "505b39e0-97a4-11ed-8903-dd5b8b903715"
          },
          {
            "label-category-attributes": {},
            "object-name": "Car:4",
            "class-id": 0,
            "center-x": 17.192725195301094,
            "center-y": 114.55705365827872,
            "center-z": 1.590525771183712,
            "length": 4,
            "width": 2,
            "height": 2,
            "roll": 0,
            "pitch": 0,
            "yaw": 0,
            "object-id": "1afcb670-97a9-11ed-9a84-ff627d099e16"
          }
```

```
        ],
        "frame-attributes": {}
      }
    ]
  },
  "camera-0": {
    "tracking-annotations": [
      {
        "frame-no": 0,
        "frame": "0.txt.pcd",
        "annotations": [
          {
            "label-category-attributes": {
              "Occlusion": "Partial"
            },
            "object-name": "Car:2",
            "class-id": 0,
            "width": 223,
            "height": 164,
            "top": 225,
            "left": 486,
            "object-id": "5229df60-97a4-11ed-8903-dd5b8b903715"
          }
        ],
        "frame-attributes": {}
      },
      {
        "frame-no": 1,
        "frame": "1.txt.pcd",
        "annotations": [
          {
            "label-category-attributes": {},
            "object-name": "Car:4",
            "class-id": 0,
            "width": 252,
            "height": 246,
            "top": 237,
            "left": 473,
            "object-id": "1afcb670-97a9-11ed-9a84-ff627d099e16"
          }
        ],
        "frame-attributes": {}
      }
    ]
```

```
  }
}
```

Kotak berbentuk kubus dan pembatas untuk suatu objek dihubungkan melalui id objek umum.

Pelabelan Data yang ditingkatkan

Amazon SageMaker Ground Truth mengelola pengiriman objek data Anda ke pekerja untuk diberi label. Pelabelan setiap objek data adalahtugas. Pekerja menyelesaikan setiap tugas sampai seluruh pekerjaan pelabelan selesai. Ground Truth membagi jumlah total tugas menjadi lebih kecilbatchyang dikirim ke pekerja. Batch baru dikirim ke pekerja ketika yang sebelumnya selesai.

Ground Truth menyediakan dua fitur yang membantu meningkatkan keakuratan label data Anda dan mengurangi total biaya pelabelan data Anda:

- Konsolidasi Anotasimembantu meningkatkan akurasi label objek data Anda. Ini menggabungkan hasil tugas anotasi beberapa pekerja menjadi satu label kesetiaan tinggi.
- Pelabelan data otomatismenggunakan pembelajaran mesin untuk memberi label pada bagian data Anda secara otomatis tanpa harus mengirimkannya ke pekerja manusia.

Topik

- Kontrol Aliran Objek Data yang Dikirim ke Pekerja
- Anotasi Terkonsolidasi
- Pelabelan Data
- Tugas Pelabelan Rantai

Kontrol Aliran Objek Data yang Dikirim ke Pekerja

Tergantung pada jenis pekerjaan pelabelan yang Anda buat, Amazon SageMaker Ground Truth mengirimkan objek data ke pekerja dalam batch atau dalam mode streaming. Anda dapat mengontrol aliran objek data ke pekerja dengan cara berikut:

- Untuk kedua jenis pekerjaan pelabelan, Anda dapat menggunakan`MaxConcurrentTaskCount`untuk mengontrol jumlah total objek data yang tersedia untuk semua pekerja pada titik waktu tertentu ketika pekerjaan pelabelan berjalan.

- Untuk pekerjaan pelabelan streaming, Anda dapat mengontrol aliran objek data ke pekerja dengan memantau dan mengontrol jumlah objek data yang dikirim ke Amazon SQS yang terkait dengan pekerjaan pelabelan Anda.

Gunakan bagian berikut untuk mempelajari lebih lanjut tentang opsi ini. Untuk mempelajari tentang tugas pelabelan streaming, lihatPekerjaan Pelabelan Streaming Ground Truth.

Topik

- Gunakan MaxConcurrentTaskCount untuk Mengontrol Aliran Objek Data
- Gunakan Amazon SQS untuk Mengontrol Aliran Objek Data ke Pekerjaan Pelabelan Streaming

Gunakan MaxConcurrentTaskCount untuk Mengontrol Aliran Objek Data

`MaxConcurrentTaskCount`mendefinisikan jumlah maksimum objek data yang dapat diberi label oleh pekerja manusia pada waktu yang sama. Jika Anda menggunakan konsol, parameter ini diatur ke 1.000. Jika Anda menggunakan`CreateLabelingJob`, Anda dapat mengatur parameter ini ke bilangan bulat antara 1 dan 1.000, inklusif.

Saat Anda memulai pekerjaan pelabelan menggunakan file manifes masukan, Ground Truth melakukan hal berikut:

1. Untuk setiap objek data yang tercantum dalam file manifes masukan Anda, satu atau beberapa tugas dibuat, tergantung pada nilai yang Anda tentukan`NumberOfHumanWorkersPerDataObject`. Misalnya, jika Anda menetapkan jumlah pekerja per objek data ke 3, 3 tugas akan dibuat untuk setiap objek dataset. Untuk ditandai sebagai berhasil diberi label, setidaknya satu pekerja harus memberi label pada objek. Atau, tugas dapat kedaluwarsa atau ditolak.
2. Jika Anda menggunakan tenaga kerja Mechanical Turk, Ground Truth pertama-tama mengirimkan sejumlah 10 objek set data ke pekerja Anda. Menggunakan batch kecil ini untuk mengatur pekerjaan pelabelan dan untuk memastikan bahwa pekerjaan dikonfigurasi dengan benar.
3. Berikutnya, Ground Truth mengirimkan`MaxConcurrentTaskCount`jumlah objek dataset untuk pekerja. Misalnya, jika Anda memiliki 2.000 objek data masukan dalam file manifes masukan Anda dan telah menetapkan jumlah pekerja per objek data ke 3 dan menetapkan`MaxConcurrentTaskCount`ke 900, 900 objek data pertama dalam manifes masukan Anda dikirim ke pekerja, sesuai dengan 2.700 tugas (900 x 3). Ini adalah kumpulan objek berukuran penuh pertama yang dikirim ke pekerja.

4. Apa yang terjadi selanjutnya tergantung pada jenis pekerjaan pelabelan yang Anda buat. Langkah ini mengasumsikan satu atau beberapa objek set data dalam file manifes masukan Anda, atau dikirim menggunakan sumber data input Amazon SNS (dalam tugas pelabelan streaming) tidak disertakan dalam kumpulan yang dikirim ke pekerja pada langkah 3.
   - Tugas pelabelan streaming: Selama jumlah total objek yang tersedia untuk pekerja sama dengan`MaxConcurrentTaskCount`, semua objek set data yang tersisa pada file manifes masukan Anda dan yang Anda kirim secara real time menggunakan Amazon SNS ditempatkan pada antrean Amazon SQS. Ketika jumlah total objek yang tersedia untuk pekerja jatuh di bawah`MaxConcurrentTaskCount`minus`NumberOfHumanWorkersPerDataObject`, objek data baru dari antrian digunakan untuk membuat`NumberOfHumanWorkersPerDataObject`-tugas, yang dikirim ke pekerja secara real time.
   - Pekerjaan pelabelan non-streaming: Saat pekerja selesai memberi label satu set objek, hingga`MaxConcurrentTaskCount`kali`NumberOfHumanWorkersPerDataObject`jumlah tugas baru akan dikirim ke pekerja. Proses ini diulang sampai semua objek data dalam file manifes masukan diberi label.

Gunakan Amazon SQS untuk Mengontrol Aliran Objek Data ke Pekerjaan Pelabelan Streaming

Saat Anda membuat pekerjaan pelabelan streaming, antrean Amazon SQS secara otomatis dibuat di akun Anda. Objek data hanya ditambahkan ke antrean Amazon SQS ketika jumlah total objek yang dikirim ke pekerja berada di atas`MaxConcurrentTaskCount`. Jika tidak, objek dikirim langsung ke pekerja.

Anda dapat menggunakan antrian ini untuk mengelola aliran objek data ke pekerjaan pelabelan Anda. Untuk mempelajari selengkapnya, lihat Mengelola Permintaan Pelabelan dengan Antrean Amazon SQS .

Anotasi Terkonsolidasi

Sesi*anotasi*adalah hasil dari tugas pelabelan pekerja tunggal. Konsolidasi Anotasi*menggabungkan* anotasi dua atau lebih pekerja menjadi satu label untuk objek data Anda. Label, yang ditugaskan untuk setiap objek dalam dataset, adalah perkiraan probabilistik dari apa label sebenarnya seharusnya. Setiap objek dalam dataset biasanya memiliki beberapa anotasi, tetapi hanya satu label atau set label.

Anda memutuskan berapa banyak pekerja yang membubuhi anotasi setiap objek dalam kumpulan data Anda. Menggunakan lebih banyak pekerja dapat meningkatkan akurasi label Anda, tetapi

juga meningkatkan biaya pelabelan. Untuk mempelajari tentang harga Ground Truth, lihatAmazon SageMaker Harga Ground Truth.

Jika Anda menggunakan Amazon SageMaker konsol untuk membuat pekerjaan pelabelan, berikut ini adalah default untuk jumlah pekerja yang dapat membubuhi keterangan objek:

- Klasifikasi teks — 3 pekerja
- Klasifikasi gambar — 3 pekerja
- Kotak pembatas — 5 pekerja
- Segmentasi semantik — 3 pekerja
- Pengenalan entitas bernama—3 pekerja

Saat Anda menggunakan`CreateLabelingJob`operasi, Anda mengatur jumlah pekerja untuk membubuhi keterangan setiap objek data dengan`NumberOfHumanWorkersPerDataObject`parameter. Anda dapat mengganti jumlah default pekerja yang membuat anotasi objek data menggunakan konsol atau`CreateLabelingJob`operasi.

Ground Truth menyediakan fungsi konsolidasi anotasi untuk setiap tugas pelabelan yang telah ditetapkan: kotak pembatas, klasifikasi gambar, pengenalan entitas nama, segmentasi semantik, dan klasifikasi teks. Ini adalah fungsi:

- Konsolidasi anotasi multi-kelas untuk klasifikasi gambar dan teks menggunakan varianMaksimalisasi Ekspektasipendekatan anotasi. Ini memperkirakan parameter untuk setiap pekerja dan menggunakan inferensi Bayesian untuk memperkirakan kelas sebenarnya berdasarkan anotasi kelas dari pekerja individu.
- Anotasi kotak pembatas mengkonsolidasikan kotak pembatas dari beberapa pekerja. Fungsi ini menemukan kotak yang paling mirip dari pekerja yang berbeda berdasarkanIndeks jaccard, atau persimpangan atas serikat, dari kotak dan rata-rata mereka.
- Konsolidasi anotasi segmentasi semantik memperlakukan setiap piksel dalam satu gambar sebagai klasifikasi multi-kelas. Fungsi ini memperlakukan anotasi piksel dari pekerja sebagai “suara,” dengan lebih banyak informasi dari piksel sekitarnya yang digabungkan dengan menerapkan fungsi smoothing ke gambar.
- Pengenalan entitas bernama cluster pilihan teks dengan kesamaan Jaccard dan menghitung batas pemilihan berdasarkan mode, atau median jika mode tidak jelas. Label memutuskan untuk label entitas yang paling ditugaskan di cluster, melanggar ikatan dengan pilihan acak.

Anda dapat menggunakan algoritma lain untuk mengkonsolidasikan anotasi. Untuk informasi, lihat Buat Fungsi Konsolidasi Anotasi Anda Sendiri.

Buat Fungsi Konsolidasi Anotasi Anda Sendiri

Anda dapat memilih untuk menggunakan fungsi konsolidasi anotasi Anda sendiri untuk menentukan label akhir untuk objek berlabel Anda. Ada banyak pendekatan yang mungkin untuk menulis fungsi dan pendekatan yang Anda ambil tergantung pada sifat anotasi untuk dikonsolidasikan. Secara umum, fungsi konsolidasi melihat anotasi dari pekerja, mengukur kesamaan di antara mereka, dan kemudian menggunakan beberapa bentuk penilaian probabilistik untuk menentukan apa label yang paling mungkin seharusnya.

Jika Anda ingin menggunakan algoritme lain untuk membuat fungsi konsolidasi anotasi, Anda dapat menemukan respons pekerja di`[project-name]`/annotations/worker-responsefolder bucket Amazon S3 tempat Anda mengarahkan output tugas.

Nilai Kesamaan

Untuk menilai kesamaan antara label, Anda dapat menggunakan salah satu strategi berikut, atau Anda dapat menggunakan salah satu yang memenuhi kebutuhan pelabelan data Anda:

- Untuk ruang label yang terdiri dari kategori terpisah dan saling eksklusif, seperti klasifikasi multi-kelas, menilai kesamaan bisa sangat mudah. Label diskrit cocok atau tidak cocok.
- Untuk spasi label yang tidak memiliki nilai diskrit, seperti anotasi kotak pembatas, temukan ukuran kesamaan yang luas. Untuk kotak pembatas, salah satu ukuran tersebut adalah indeks Jaccard. Ini mengukur rasio persimpangan dua kotak dengan penyatuan kotak untuk menilai seberapa mirip mereka. Misalnya, jika ada tiga anotasi, maka bisa ada fungsi yang menentukan anotasi mana yang mewakili objek yang sama dan harus dikonsolidasikan.

Menilai Label Paling Kemungkinan

Dengan salah satu strategi yang dirinci dalam bagian sebelumnya dalam pikiran, buat semacam penilaian probabilistik tentang apa label konsolidasi seharusnya. Dalam kasus kategori diskrit dan saling eksklusif, ini bisa sangat mudah. Salah satu cara paling umum untuk melakukan ini adalah dengan mengambil hasil pemungutan suara mayoritas di antara anotasi. Ini bobot anotasi sama.

Beberapa pendekatan mencoba memperkirakan keakuratan anotator yang berbeda dan membebani anotasi mereka secara proporsional dengan probabilitas kebenaran. Contohnya adalah metode Expectation Maximization, yang digunakan dalam fungsi konsolidasi Ground Truth default untuk anotasi multi-kelas.

Untuk informasi selengkapnya tentang cara membuat fungsi konsolidasi anotasi, lihatLangkah 3: Memproses dengan AWS Lambda.

Pelabelan Data

Jika Anda memilih, Amazon SageMaker Ground Truth dapat menggunakan pembelajaran aktif untuk mengotomatiskan pelabelan data input Anda untuk jenis tugas bawaan tertentu. Pembelajaran aktifadalah teknik pembelajaran mesin yang mengidentifikasi data yang harus diberi label oleh pekerja Anda. Dalam Ground Truth, fungsi ini disebut pelabelan data otomatis. Pelabelan data otomatis membantu mengurangi biaya dan waktu yang diperlukan untuk memberi label pada kumpulan data Anda dibandingkan dengan hanya menggunakan manusia. Saat Anda menggunakan pelabelan otomatis, Anda dikenakan SageMaker biaya pelatihan dan inferensi.

Sebaiknya gunakan pelabelan data otomatis pada kumpulan data besar karena jaringan saraf yang digunakan dengan pembelajaran aktif memerlukan sejumlah besar data untuk setiap kumpulan data baru. Biasanya, saat Anda memberikan lebih banyak data, potensi prediksi akurasi tinggi naik. Data hanya akan diberi label otomatis jika jaringan saraf yang digunakan dalam model pelabelan otomatis dapat mencapai tingkat akurasi yang sangat tinggi. Oleh karena itu, dengan kumpulan data yang lebih besar, ada lebih banyak potensi untuk memberi label data secara otomatis karena jaringan saraf dapat mencapai akurasi yang cukup tinggi untuk pelabelan otomatis. Pelabelan data otomatis paling tepat bila Anda memiliki ribuan objek data. Jumlah minimum objek yang diizinkan untuk pelabelan data otomatis adalah 1.250, tetapi kami sangat menyarankan untuk menyediakan minimal 5.000 objek.

Pelabelan data otomatis hanya tersedia untuk jenis tugas bawaan Ground Truth berikut:

- Klasifikasi Gambar (Label Tunggal)
- Segmentasi Semantic
- Deteksi objek (Kotak pembatas)
- Klasifikasi Teks (Label Tunggal)

Tugas pelabelan streamingtidak mendukung pelabelan data otomatis.

Untuk mempelajari cara membuat alur kerja pembelajaran aktif kustom menggunakan model Anda sendiri, lihatSiapkan alur kerja pembelajaran aktif dengan model Anda sendiri.

Kuota data input berlaku untuk pekerjaan pelabelan data otomatis. LihatKuotauntuk informasi tentang ukuran dataset, ukuran data input dan batas resolusi.

> **Note**
>
> Sebelum Anda menggunakan model pelabelan otomatis dalam produksi, Anda perlu menyempurnakan atau mengujinya, atau keduanya. Anda dapat menyempurnakan model (atau membuat dan menyetel model lain yang diawasi pilihan Anda) pada kumpulan data yang dihasilkan oleh pekerjaan pelabelan Anda untuk mengoptimalkan arsitektur dan hyperparameter model. Jika Anda memutuskan untuk menggunakan model untuk inferensi tanpa menyempurnakannya, kami sangat menyarankan untuk memastikan bahwa Anda mengevaluasi keakuratannya pada bagian representatif (misalnya, dipilih secara acak) dari kumpulan data yang diberi label Ground Truth dan sesuai dengan harapan Anda.

Cara Kerjanya

Anda mengaktifkan pelabelan data otomatis saat membuat pekerjaan pelabelan. Beginilah cara kerjanya:

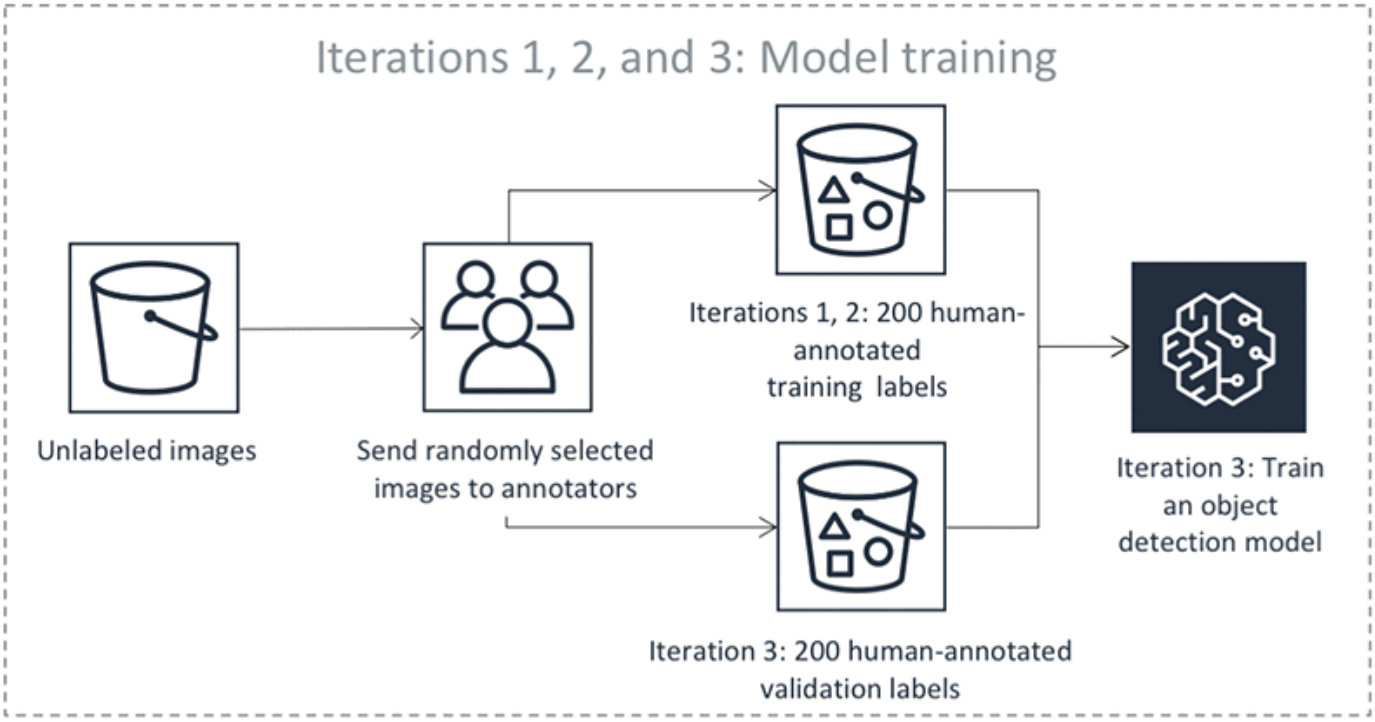
1. Ketika Ground Truth memulai pekerjaan pelabelan data otomatis, ia memilih sampel acak dari objek data input dan mengirimkannya ke pekerja manusia. Jika lebih dari 10% dari objek data ini gagal, pekerjaan pelabelan akan gagal. Jika pekerjaan pelabelan gagal, selain meninjau pesan kesalahan Ground Truth kembali, periksa apakah data input Anda ditampilkan dengan benar di UI pekerja, instruksinya jelas, dan bahwa Anda telah memberi pekerja cukup waktu untuk menyelesaikan tugas.
2. Ketika data berlabel dikembalikan, itu digunakan untuk membuat set pelatihan dan set validasi. Ground Truth menggunakan kumpulan data ini untuk melatih dan memvalidasi model yang digunakan untuk pelabelan otomatis.
3. Ground Truth menjalankan batch mengubah pekerjaan, menggunakan model divalidasi untuk inferensi pada data validasi. Inferensi Batch menghasilkan skor kepercayaan dan metrik kualitas untuk setiap objek dalam data validasi.
4. Komponen pelabelan auto akan menggunakan metrik kualitas dan skor kepercayaan ini untuk membuatambang batas tingkat kepercayaanyang memastikan label berkualitas.
5. Ground Truth menjalankan batch mengubah pekerjaan pada data unlabeled dalam dataset, menggunakan model divalidasi yang sama untuk inferensi. Ini menghasilkan skor kepercayaan untuk setiap objek.
6. Komponen pelabelan auto Ground Truth menentukan apakah skor kepercayaan yang dihasilkan pada langkah 5 untuk setiap objek memenuhi ambang batas yang diperlukan yang ditentukan

pada langkah 4. Jika skor kepercayaan memenuhi ambang batas, kualitas pelabelan otomatis yang diharapkan melebihi tingkat akurasi yang diminta dan objek tersebut dianggap diberi label otomatis.

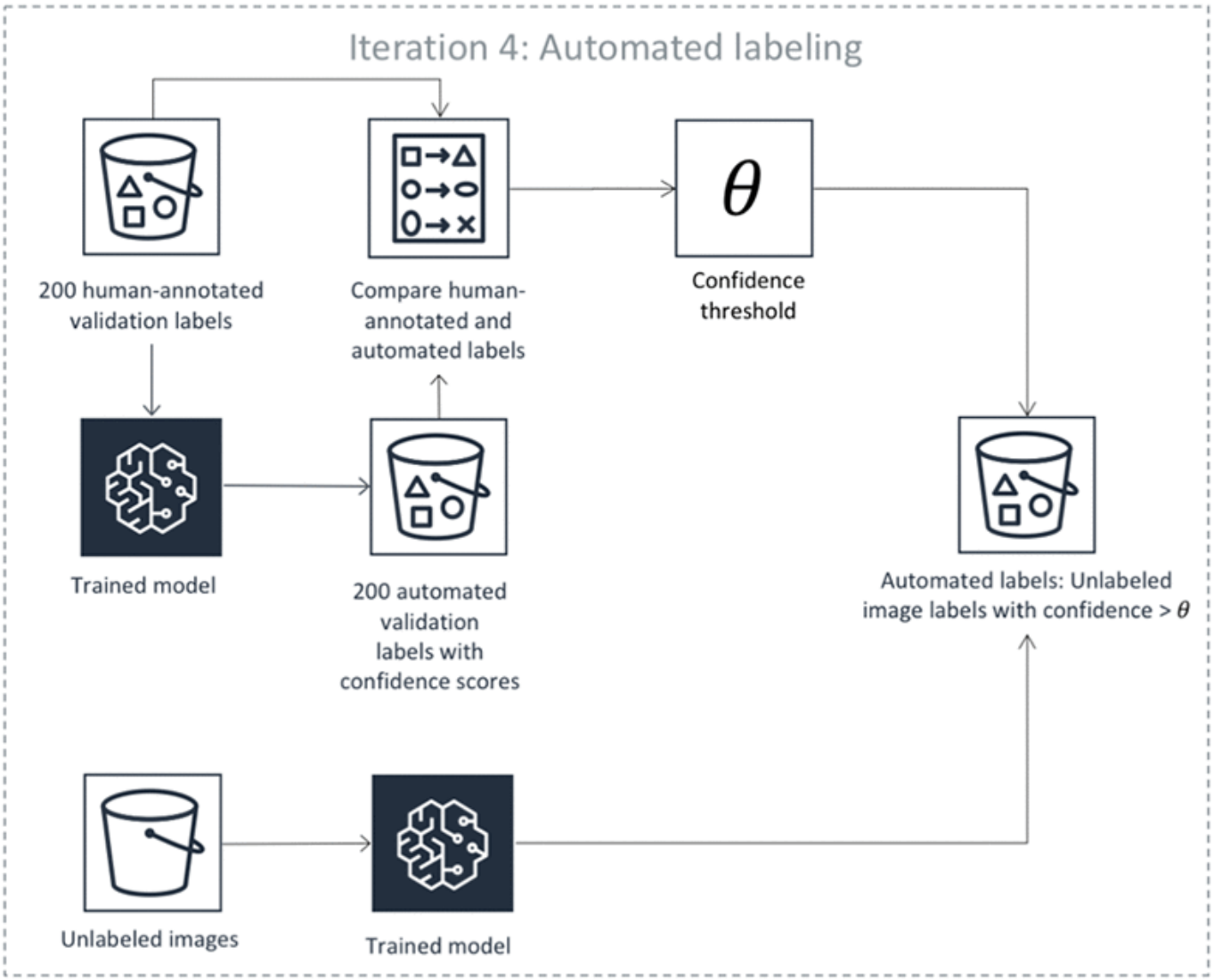
7. Langkah 6 menghasilkan dataset data yang tidak berlabel dengan skor keyakinan. Ground Truth memilih titik data dengan skor kepercayaan rendah dari kumpulan data ini dan mengirimkannya ke pekerja manusia.
8. Ground Truth menggunakan data berlabel manusia yang ada dan data berlabel tambahan ini dari pekerja manusia untuk memperbarui model.
9. Proses ini diulang sampai dataset sepenuhnya diberi label atau sampai kondisi berhenti lainnya terpenuhi. Misalnya, pelabelan otomatis berhenti jika anggaran anotasi manusia Anda tercapai.

Langkah-langkah sebelumnya terjadi dalam iterasi. Pilih setiap tab dalam tabel berikut untuk melihat contoh proses yang terjadi di setiap iterasi untuk pekerjaan pelabelan otomatis deteksi objek. Jumlah objek data yang digunakan dalam langkah tertentu dalam gambar-gambar ini (misalnya, 200) khusus untuk contoh ini. Jika ada kurang dari 5.000 objek untuk label, ukuran set validasi adalah 20% dari seluruh dataset. Jika ada lebih dari 5.000 objek dalam dataset input Anda, ukuran set validasi adalah 10% dari seluruh dataset. Anda dapat mengontrol jumlah label manusia yang dikumpulkan per iterasi pembelajaran aktif dengan mengubah nilai`MaxConcurrentTaskCount`saat menggunakan operasi API`CreateLabelingJob`. Nilai ini diatur ke 1.000 saat Anda membuat pekerjaan pelabelan menggunakan konsol. Dalam alur pembelajaran aktif diilustrasikan di bawahPembelajaran Aktiftab, nilai ini diatur ke 200.

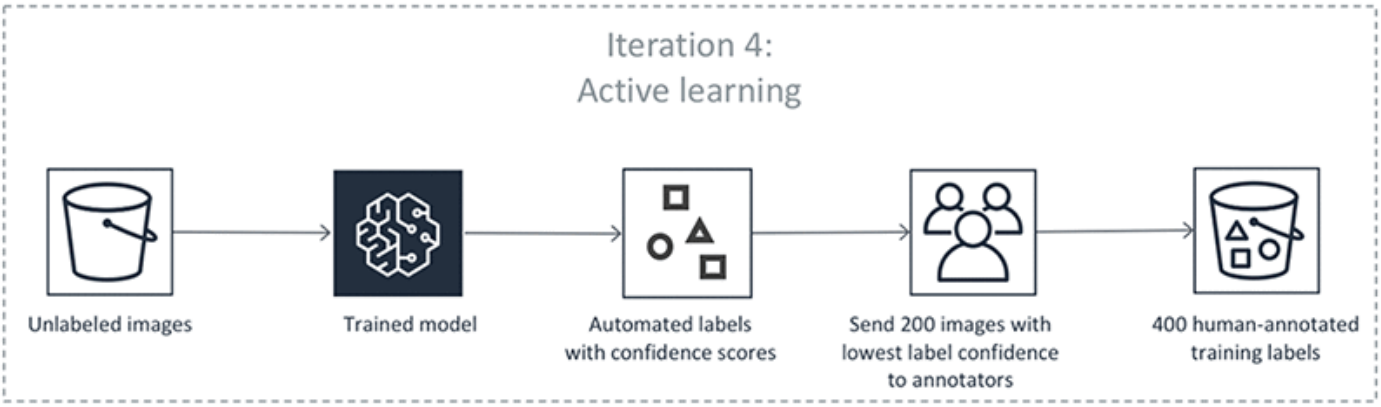
Model Training



Automated Labeling



Active Learning



Akurasi Label Otomatis

Definisiketepatantergantung pada jenis tugas bawaan yang Anda gunakan dengan pelabelan otomatis. Untuk semua jenis tugas, persyaratan akurasi ini ditentukan sebelumnya oleh Ground Truth dan tidak dapat dikonfigurasi secara manual.

- Untuk klasifikasi gambar dan klasifikasi teks, Ground Truth menggunakan logika untuk menemukan tingkat kepercayaan prediksi label yang sesuai dengan setidaknya 95% akurasi label. Ini berarti Ground Truth mengharapkan keakuratan label otomatis setidaknya 95% jika dibandingkan dengan label yang akan diberikan pelabel manusia untuk contoh-contoh tersebut.
- Untuk kotak pembatas, rata-rata yang diharapkanPersimpangan Over Union (IoU)dari gambar berlabel otomatis adalah 0,6. Untuk menemukan IoU rata-rata, Ground Truth menghitung ioU rata-rata dari semua kotak yang diprediksi dan tidak terjawab pada gambar untuk setiap kelas, dan kemudian rata-rata nilai-nilai ini di seluruh kelas.
- Untuk segmentasi semantik, rata-rata IoU yang diharapkan dari gambar berlabel otomatis adalah 0,7. Untuk menemukan mean IoU, Ground Truth mengambil rata-rata nilai IoU dari semua kelas dalam gambar (tidak termasuk latar belakang).

Pada setiap iterasi Pembelajaran Aktif (langkah 3-6 dalam daftar di atas), ambang kepercayaan ditemukan menggunakan set validasi anotasi manusia sehingga akurasi yang diharapkan dari objek berlabel otomatis memenuhi persyaratan akurasi tertentu yang telah ditentukan sebelumnya.

Membuat Job Pelabelan Data Otomatis (Konsol)

Untuk membuat pekerjaan pelabelan yang menggunakan pelabelan otomatis di SageMaker konsol, gunakan prosedur berikut.

Untuk membuat pekerjaan pelabelan data otomatis (konsol)

1. Buka Ground TruthLowongan kerja labelbagian SageMakerkonsol:https://console.aws.amazon.com/sagemaker/groundtruth.
2. MenggunakanMembuat Job Pelabelan (Konsol)sebagai panduan, lengkapiGambaran umum JobdanJenis tugasbagian. Perhatikan bahwa pelabelan auto tidak didukung untuk jenis tugas khusus.
3. Di bawahPekerja, pilih jenis tenaga kerja Anda.
4. Di bagian yang sama, pilihAktifkan pelabelan data otomatis.

5. MenggunakanLangkah 4: Konfigurasikan Alat Kotak Boundingsebagai panduan, buat instruksi pekerja di bagian***Tipe tugas***alat pelikan. Misalnya, jika Anda memilihSegmentasi semantiksebagai jenis pekerjaan pelabelan Anda, bagian ini disebutAlat pelabelan segmentasi semantik.
6. Untuk melihat pratinjau instruksi dan dasbor pekerja Anda, pilihPratinjau.
7. Pilih Create (Buat). Ini menciptakan dan memulai pekerjaan pelabelan Anda dan proses pelabelan auto.

Anda dapat melihat pekerjaan pelabelan Anda muncul diLowongan kerja labelbagian SageMaker konsol. Data keluaran Anda muncul di bucket Amazon S3 yang Anda tentukan saat membuat tugas pelabelan. Untuk informasi selengkapnya tentang format dan struktur file dari data keluaran tugas pelabelan Anda, lihatData Output.

Buat Job Pelabelan Data Otomatis (API)

Untuk membuat pekerjaan pelabelan data otomatis menggunakan SageMaker API, gunakan`LabelingJobAlgorithmsConfig`parameter`CreateLabelingJob`operasi. Untuk mempelajari cara memulai pekerjaan pelabelan menggunakan`CreateLabelingJob`operasi, lihatBuat Job Pelabelan (API).

Amazon Resource Name (ARN) dari algoritme yang Anda gunakan untuk pelabelan data otomatis diLabelingJobAlgorithmSpecificationArnparameter. Pilih dari salah satu dari empat algoritma bawaan Ground Truth yang didukung dengan pelabelan otomatis:

- Klasifikasi Gambar (Label Tunggal)
- Segmentasi Semantic
- Deteksi objek (Kotak pembatas)
- Klasifikasi Teks (Label Tunggal)

Ketika pekerjaan pelabelan data otomatis selesai, Ground Truth mengembalikan ARN dari model yang digunakannya untuk pekerjaan pelabelan data otomatis. Gunakan model ini sebagai model awal untuk jenis pekerjaan pelabelan otomatis serupa dengan menyediakan ARN, dalam format string, diInitialActiveLearningModelArnparameter. Untuk mengambil ARN model, gunakanAWS Command Line Interface(AWS CLI) perintah yang serupa dengan berikut ini.

```
# Fetch the mARN of the model trained in the final iteration of the previous labeling
 job.Ground Truth
```

```
pretrained_model_arn = sagemaker_client.describe_labeling_job(LabelingJobName=job_name)
['LabelingJobOutput']['FinalActiveLearningModelArn']
```

Untuk mengenkripsi data pada volume penyimpanan yang melekat pada instans komputasi MLyang digunakan dalam pelabelan otomatis, sertakanAWS Key Management Service(AWS KMS) kunci di`VolumeKmsKeyId`parameter. Untuk informasi tentangAWSKunci KMS, lihatApaAWSLayanan Manajemen Kunci?di dalamAWSPanduan Developer Layanan Manajemen Kunci.

Untuk contoh yang menggunakan`CreateLabelingJob`operasi untuk membuat pekerjaan pelabelan data otomatis, lihatobject_detection_tutorialcontohSageMaker Contoh,Tugas pelabelan Ground Truthbagian SageMaker Instans notebook. Untuk mempelajari cara membuat dan membuka instans notebook, lihatMembuat Instance Notebook. Untuk mempelajari cara mengakses SageMaker contoh notebook, lihatContoh Notebook.

Instans Amazon EC2 Diperlukan untuk Pelabelan Data Otomatis

Tabel berikut mencantumkan instans Amazon Elastic Compute Cloud (Amazon EC2) yang Anda perlukan untuk menjalankan pelabelan data otomatis untuk tugas pelatihan dan inferensi batch.

Jenis Job Pelabelan Data Otomatis	Jenis Instance Training	Jenis Inferensi
Klasifikasi gambar	ml.p3.xlarge	ml.c5.xlarge
Deteksi objek (kotak pembatas)	ml.p3.xlarge	ml.c5.4xlarge
Klasifikasi teks	ml.c5.2xlarge	db.m4.xlarge
Segmentasi semantik	ml.p3.xlarge	ml.p3.xlarge

* Di Wilayah Asia Pacific (Mumbai) (ap-south-1) gunakan ml.p2.8xlarge sebagai gantinya.

Ground Truth mengelola instance yang Anda gunakan untuk pekerjaan pelabelan data otomatis. Ini menciptakan, mengkonfigurasi, dan mengakhiri contoh yang diperlukan untuk melakukan pekerjaan Anda. Instans ini tidak muncul di dasbor instans Amazon EC2 Anda.

Siapkan alur kerja pembelajaran aktif dengan model Anda sendiri

Anda dapat membuat alur kerja pembelajaran aktif dengan algoritme Anda sendiri untuk menjalankan pelatihan dan kesimpulan dalam alur kerja tersebut untuk memberi label otomatis pada data Anda. Notebook bring_your_own_model_for_sagemaker_labeling_workflows_with_active_learning.ipynb menunjukkan ini menggunakan SageMaker algoritma bawaan,BlazingText. Notebook ini menyediakanAWS CloudFormationtumpukan yang dapat Anda gunakan untuk menjalankan alur kerja ini menggunakanAWS Step Functions. Anda dapat menemukan notebook dan file pendukung dalam hal iniGitHub repositori.

Anda juga dapat menemukan notebook ini di SageMaker Contoh repositori. LihatMenggunakan Notebook contohuntuk mempelajari cara menemukan Amazon SageMaker notebook.

Tugas Pelabelan Rantai

Amazon SageMaker Ground Truth dapat menggunakan kembali kumpulan data dari pekerjaan sebelumnya dengan dua cara: kloning dan rantai.

Kloningmenyalin penyiapan pekerjaan pelabelan sebelumnya dan memungkinkan Anda untuk membuat perubahan tambahan sebelum mengaturnya untuk dijalankan.

Rantaitidak hanya menggunakan pengaturan pekerjaan sebelumnya, tetapi juga hasilnya. Ini memungkinkan Anda untuk melanjutkan pekerjaan yang tidak lengkap dan menambahkan label atau objek data ke pekerjaan yang telah selesai. Chaining adalah operasi yang lebih kompleks.

Untuk pemrosesan data:

- Kloning menggunakan pekerjaan sebelumnyamemasukkanmanifes, dengan modifikasi opsional, sebagai manifes masukan pekerjaan baru.
- Chaining menggunakan pekerjaan sebelumnyakeluaranmanifes sebagai manifes masukan pekerjaan baru.

Rantai berguna ketika Anda perlu:

- Lanjutkan pekerjaan pelabelan yang dihentikan secara manual.
- Lanjutkan pekerjaan pelabelan yang gagal di tengah pekerjaan, setelah memperbaiki masalah.
- Beralih ke pelabelan data otomatis setelah memberi label secara manual pada bagian pekerjaan (atau sebaliknya).

- Tambahkan lebih banyak objek data ke pekerjaan yang telah selesai dan mulai pekerjaan dari sana.
- Tambahkan anotasi lain ke pekerjaan yang telah selesai. Misalnya, Anda memiliki kumpulan frasa berlabel untuk topik, lalu ingin menjalankan set lagi, mengkategorikannya berdasarkan audiens tersirat topik.

Amazon SageMaker Ground Truth Anda dapat mengkonfigurasi pekerjaan pelabelan dirantai dengan konsol atau API.

Istilah kunci: Nama atribut

Klasternama atribut label(`LabelAttributeName`di API) adalah string yang digunakan sebagai kunci untuk pasangan kunci-nilai yang dibentuk dengan label yang diberikan pekerja ke objek data.

Aturan berikut berlaku untuk nama atribut label:

- Itu tidak bisa berakhir dengan`-metadata`.
- Nama-nama`source`dan`source-ref`dicadangkan dan tidak dapat digunakan.
- Untuk pekerjaan pelabelan segmentasi semantik,, itu harus diakhiri dengan`-ref`. Untuk semua pekerjaan pelabelan lainnya, itutidak bisadiakhiri`-ref`. Jika Anda menggunakan konsol untuk membuat pekerjaan, Amazon SageMaker Ground Truth secara otomatis menambahkan`-ref`untuk semua nama atribut label kecuali untuk pekerjaan segmentasi semantik.
- Untuk pekerjaan pelabelan yang dirantai, jika Anda menggunakan nama atribut label yang sama dari pekerjaan asal dan Anda mengonfigurasi pekerjaan yang dirantai untuk menggunakan pelabelan otomatis, maka jika sudah dalam mode pelabelan otomatis pada titik mana pun, Ground Truth menggunakan model dari pekerjaan asal.

Dalam manifes keluaran, nama atribut label muncul serupa dengan berikut ini.

```
"source-ref": "<S3 URI>",
"<label attribute name>": {
  "annotations": [{
    "class_id": 0,
    "width": 99,
    "top": 87,
    "height": 62,
    "left": 175
  }],
```

```
    "image_size": [{
      "width": 344,
      "depth": 3,
      "height": 234
    }]
  },
  "<label attribute name>-metadata": {
    "job-name": "<job name>",
    "class-map": {
      "0": "<label attribute name>"
    },
    "human-annotated": "yes",
    "objects": [{
      "confidence": 0.09
    }],
    "creation-date": "<timestamp>",
    "type": "groundtruth/object-detection"
  }
```

Jika Anda membuat pekerjaan di konsol dan tidak secara eksplisit menetapkan nilai nama atribut label, Ground Truth menggunakan nama pekerjaan sebagai nama atribut label untuk pekerjaan tersebut.

Memulai Job Dirantai (Konsol)

Pilih pekerjaan pelabelan yang berhenti, gagal, atau selesai dari daftar pekerjaan Anda yang ada. Hal ini memungkinkanTindakanmenu.

DariTindakanmenu, pilihRantai.

Panel Gambaran umum Job

DiGambaran umum Jobpanel, baruNama tugasdiatur berdasarkan judul pekerjaan tempat Anda merantai yang satu ini. Anda dapat mengubahnya.

Anda juga dapat menentukan nama atribut label yang berbeda dari nama pekerjaan pelabelan.

Jika Anda merantai dari pekerjaan yang telah selesai, nama atribut label menggunakan nama pekerjaan baru yang Anda konfigurasi. Untuk mengubah nama, pilih kotak centang.

Jika Anda merantai dari pekerjaan yang dihentikan atau gagal, nama atribut label digunakan untuk nama pekerjaan tempat Anda merantai. Sangat mudah untuk melihat dan mengedit nilai karena kotak centang nama dicentang.

Pertimbangan penamaan label atribut

- Bawaanmenggunakan nama atribut label Ground Truth telah dipilih. Semua objek data tanpa data yang terhubung ke nama atribut label diberi label.
- Menggunakan nama atribut labeltidak hadir dalam manifes menyebabkan pekerjaan untuk memprosessegalaobjek dalam set data.

Klasterlokasi set data inputdalam hal ini secara otomatis dipilih sebagai manifes output dari pekerjaan dirantai. Kolom input tidak tersedia, sehingga Anda tidak dapat mengubahnya.

Menambahkan objek data ke pekerjaan pelabelan

Anda tidak dapat menentukan berkas manifes alternatif. Edit manifes keluaran secara manual dari tugas sebelumnya untuk menambahkan item baru sebelum memulai pekerjaan yang dirantai. URI Amazon S3 membantu Anda menemukan lokasi penyimpanan manifes di bucket Amazon S3 Anda. Unduh file manifes dari sana, edit secara lokal di komputer Anda, lalu unggah versi baru untuk menggantinya. Pastikan Anda tidak memperkenalkan kesalahan selama pengeditan. Kami sarankan Anda menggunakan JSON linter untuk memeriksa JSON Anda. Banyak editor teks populer dan IDE memiliki linter plugin yang tersedia.

Memulai Job Dirantai (API)

Prosedurnya hampir sama dengan menyiapkan pekerjaan pelabelan baru`CreateLabelingJob`, kecuali dua perbedaan utama:

- Lokasi manifes: Daripada menggunakan manifes asli Anda dari pekerjaan sebelumnya, nilai untuk`ManifestS3Uri`di dalam`DataSource`harus menunjuk ke URI Amazon S3manifesdari pekerjaan pelabelan sebelumnya.
- Nama atribut label: Mengatur yang benar`LabelAttributeName`nilai penting di sini. Ini adalah bagian kunci dari pasangan kunci-nilai di mana pelabelan data adalah nilai. Kasus penggunaan sampel meliputi:
  - Menambahkan label baru atau lebih spesifik ke pekerjaan yang telah selesai- Tetapkan nama atribut label baru.
  - Pelabelan item yang tidak berlabel dari pekerjaan sebelumnya- Gunakan nama atribut label dari pekerjaan sebelumnya.

Menggunakan Dataset Berlabel Sebagian

Anda bisa mendapatkan beberapa manfaat rantai jika Anda menggunakan manifes tambahan yang telah diberi label sebagian. PemeriksaanNama atributcentang kotak dan atur nama sehingga cocok dengan nama dalam manifes Anda.

Jika Anda menggunakan API, instruksinya sama dengan instruksi untuk memulai pekerjaan yang dirantai. Namun, pastikan untuk mengunggah manifes Anda ke bucket Amazon S3 dan menggunakannya alih-alih menggunakan manifes keluaran dari pekerjaan sebelumnya.

KlasterNama atributnilai dalam manifes harus sesuai dengan pertimbangan penamaan dibahas sebelumnya.

Keamanan dan Izin Kebenaran Dasar

Gunakan topik di halaman ini untuk mempelajari tentang fitur keamanan Ground Truth dan cara mengonfigurasi izinAWS Identity and Access Management (IAM) untuk memungkinkan pengguna atau peran membuat pekerjaan pelabelan. Selain itu, pelajari cara membuat peran eksekusi. Peran eksekusi adalah peran yang Anda tentukan saat Anda membuat tugas pelabelan. Peran ini digunakan untuk memulai pekerjaan pelabelan Anda.

Jika Anda adalah pengguna baru dan ingin memulai dengan cepat, atau jika Anda tidak memerlukan izin terperinci, lihatMenggunakan Kebijakan Terkelola IAM dengan Ground Truth.

Untuk informasi selengkapnya tentang pengguna dan peran IAM, lihat Identitas (Pengguna, Grup, dan Peran) dalam Panduan Pengguna IAM.

Untuk mempelajari lebih lanjut tentang menggunakan IAM dengan SageMaker, lihatIdentity and Access Management untuk Amazon SageMaker.

Topik

- Persyaratan Izin
- Tetapkan Izin IAM untuk Menggunakan Ground Truth
- Menggunakan Amazon SageMaker Ground Truth di Amazon Virtual Private Cloud
- Data Keluaran dan Enkripsi Volume Penyimpanan
- Otentikasi dan Pembatasan Tenaga Kerja

Persyaratan Izin

Sebelumnya pada tahun 2020, browser yang banyak digunakan seperti Chrome dan Firefox mengubah perilaku default mereka untuk memutar gambar berdasarkan metadata gambar, disebut sebagai data EXIF. Sebelumnya, browser akan selalu menampilkan gambar dengan cara yang persis di mana mereka disimpan pada disk, yang biasanya tidak diputar. Setelah perubahan, gambar sekarang berputar sesuai dengan sepotong metadata gambar yang disebut nilai orientasi. Ini memiliki implikasi penting bagi seluruh komunitas machine learning (ML/machine learning). Misalnya, jika aplikasi yang membuat anotasi gambar tidak mempertimbangkan orientasi EXIF, aplikasi tersebut dapat menampilkan gambar dalam orientasi yang tidak terduga, sehingga menghasilkan label yang salah.

Dimulai dengan Chrome 89, tidakAWS bisa lagi secara otomatis mencegah rotasi gambar karena standar web kelompok W3C telah memutuskan bahwa kemampuan untuk mengontrol rotasi gambar melanggar kebijakan sama-asal web. Oleh karena itu, untuk memastikan pekerja manusia membuat anotasi gambar masukan Anda dalam orientasi yang dapat diprediksi saat Anda mengirimkan permintaan untuk membuat pekerjaan pelabelan, Anda harus menambahkan kebijakan header CORS ke bucket Amazon S3 yang berisi gambar masukan Anda.

> **Important**
>
> Jika Anda tidak menambahkan konfigurasi CORS ke bucket Amazon S3 yang berisi data input Anda, tugas pelabelan untuk objek data input tersebut akan gagal.

Jika Anda membuat tugas melalui konsol Ground Truth, Jika semua data input Anda tidak terletak di bucket Amazon S3 yang sama dengan file manifes masukan Anda, Anda harus menambahkan konfigurasi CORS ke semua bucket Amazon S3 yang berisi data input menggunakan petunjuk berikut.

Jika Anda menggunakan`CreateLabelingJob` API untuk membuat pekerjaan pelabelan Ground Truth, Anda dapat menambahkan kebijakan CORS ke bucket Amazon S3 yang berisi data input di konsol S3. Untuk mengatur header CORS yang diperlukan pada bucket Amazon S3 yang berisi gambar masukan Anda di konsol Amazon S3, ikuti petunjuk yang terperinci dalam Bagaimana cara menambahkan berbagi sumber daya lintas domain dengan CORS? . Gunakan kode konfigurasi CORS berikut untuk bucket yang meng-host gambar Anda. Jika Anda menggunakan konsol Amazon S3 untuk menambahkan kebijakan ke bucket Anda, Anda harus menggunakan format JSON.

> **Important**
>
> Jika Anda membuat cloud titik 3D atau pekerjaan pelabelan bingkai video, Anda harus menambahkan aturan tambahan ke konfigurasi CORS Anda. Untuk mempelajari lebih lanjut, lihatPersyaratan Izin Job Pelabelan 3D Point Cloud danPersyaratan Izin Job Bingkai Video masing-masing.

JSON

```
[{
   "AllowedHeaders": [],
   "AllowedMethods": ["GET"],
   "AllowedOrigins": ["*"],
   "ExposeHeaders": ["Access-Control-Allow-Origin"]
}]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>*</AllowedOrigin>
   <AllowedMethod>GET</AllowedMethod>
   <ExposeHeader>Access-Control-Allow-Origin</ExposeHeader>
 </CORSRule>
</CORSConfiguration>
```

Tetapkan Izin IAM untuk Menggunakan Ground Truth

Gunakan topik di bagian ini untuk mempelajari cara menggunakan kebijakan terkelola dan kustomAWS Identity and Access Management (IAM) untuk mengelola akses ke Ground Truth dan sumber daya terkait.

Anda dapat menggunakan bagian di halaman ini untuk mempelajari hal-hal berikut:

- Cara membuat kebijakan IAM yang memberikan izin pengguna atau peran untuk membuat pekerjaan pelabelan. Administrator dapat menggunakan kebijakan IAM untuk membatasi akses ke Amazon SageMaker danAWS layanan lain yang khusus untuk Ground Truth.

- Cara membuat peran SageMaker eksekusi. Peran eksekusi adalah peran yang Anda tentukan saat Anda membuat tugas pelabelan. Peran ini digunakan untuk memulai dan mengelola pekerjaan pelabelan Anda.

Berikut ini adalah ikhtisar topik yang akan Anda temukan di halaman ini:

- Jika Anda mulai menggunakan Ground Truth, atau Anda tidak memerlukan izin terperinci untuk kasus penggunaan Anda, disarankan agar Anda menggunakan kebijakan terkelola IAM yang dijelaskan dalamMenggunakan Kebijakan Terkelola IAM dengan Ground Truth.
- Pelajari tentang izin yang diperlukan untuk menggunakan konsol Ground Truth diBerikan Izin IAM untuk Menggunakan Amazon SageMaker Ground Truth Console. Bagian ini mencakup contoh kebijakan yang memberikan izin entitas IAM untuk membuat dan memodifikasi tim kerja pribadi, berlangganan tim kerja vendor, dan membuat alur kerja pelabelan khusus.
- Ketika Anda membuat tugas pelabelan, Anda harus memberikan peran eksekusi. GunakanBuat Peran SageMaker Eksekusi untuk Job Pelabelan Ground Truth untuk mempelajari tentang izin yang diperlukan untuk peran ini.

Menggunakan Kebijakan Terkelola IAM dengan Ground Truth

SageMaker dan Ground Truth menyediakan kebijakanAWS terkelola yang dapat Anda gunakan untuk membuat pekerjaan pelabelan. Jika Anda mulai menggunakan Ground Truth dan Anda tidak memerlukan izin terperinci untuk kasus penggunaan Anda, disarankan agar Anda menggunakan kebijakan berikut:

- `AmazonSageMakerFullAccess`- Gunakan kebijakan ini untuk memberikan izin kepada pengguna atau peran untuk membuat pekerjaan pelabelan. Ini adalah kebijakan luas yang memberikan izin entitas untuk menggunakan SageMaker fitur, serta fiturAWS layanan yang diperlukan melalui konsol dan API. Kebijakan ini memberikan izin entitas untuk membuat pekerjaan pelabelan dan untuk membuat dan mengelola tenaga kerja menggunakan Amazon Cognito. Untuk mempelajari lebih lanjut, lihat AmazonSageMakerFullAccess Kebijakan.
- `AmazonSageMakerGroundTruthExecution`- Untuk membuat peran eksekusi, Anda dapat melampirkan kebijakan`AmazonSageMakerGroundTruthExecution` ke peran. Peran eksekusi adalah peran yang Anda tentukan saat membuat pekerjaan pelabelan dan digunakan untuk memulai pekerjaan pelabelan Anda. Kebijakan ini memungkinkan Anda untuk membuat pekerjaan pelabelan streaming dan non-streaming, dan membuat pekerjaan pelabelan menggunakan jenis tugas apa pun. Perhatikan batasan kebijakan terkelola berikut ini.

- Izin Amazon S3: Kebijakan ini memberikan izin peran eksekusi untuk mengakses bucket Amazon S3 dengan string berikut dalam nama:`GroundTruth`,,`Groundtruth`,`groundtruth`,`SageMakerSagemaker`, dan`sagemaker` atau bucket dengan tag objek yang menyertakan`SageMaker` dalam nama (tidak sensitif huruf). Pastikan nama bucket input dan output Anda menyertakan string ini, atau tambahkan izin tambahan ke peran eksekusi Anda untuk memberikannya izin mengakses bucket Amazon S3 Anda. Anda harus memberikan izin peran ini untuk melakukan tindakan berikut pada bucket Amazon S3 Anda:`AbortMultipartUpload`,`GetObject`, dan`PutObject`.
- Alur Kerja Kustom: Saat Anda membuat alur kerja pelabelan khusus, peran eksekusi ini dibatasi untuk memanggilAWS Lambda fungsi dengan salah satu string berikut sebagai bagian dari nama fungsi:`GtRecipe`,`SageMaker`,`Sagemaker`,`sagemaker`, atau`LabelingFunction`. Ini berlaku untuk fungsi Lambda pra-anotasi dan pasca-anotasi Anda. Jika Anda memilih untuk menggunakan nama tanpa string tersebut, Anda harus secara eksplisit memberikan`lambda:InvokeFunction` izin untuk peran eksekusi yang digunakan untuk membuat pekerjaan pelabelan.

Untuk mempelajari cara melampirkan kebijakanAWS terkelola ke pengguna atau peran, lihat Menambahkan dan menghapus izin identitas IAM di Panduan Pengguna IAM.

Berikan Izin IAM untuk Menggunakan Amazon SageMaker Ground Truth Console

Untuk menggunakan area Ground Truth SageMaker konsol, Anda perlu memberikan izin kepada entitas untuk mengakses SageMaker danAWS layanan lain yang berinteraksi dengan Ground Truth. Izin yang diperlukan untuk mengaksesAWS layanan lain tergantung pada kasus penggunaan Anda:

- Izin Amazon S3 diperlukan untuk semua kasus penggunaan. Izin ini harus memberikan akses ke bucket Amazon S3 yang berisi data input dan output.
- AWS Marketplaceizin diperlukan untuk menggunakan tenaga kerja vendor.
- Izin Amazon Cognito diperlukan untuk pengaturan tim kerja pribadi.
- AWS KMSizin diperlukan untuk melihatAWS KMS kunci yang tersedia yang dapat digunakan untuk enkripsi data keluaran.
- Izin IAM diperlukan untuk mencantumkan peran eksekusi yang sudah ada sebelumnya, atau untuk membuat yang baru. Selain itu, Anda harus menggunakan menambahkan`PassRole` izin untuk memungkinkan SageMaker untuk menggunakan peran eksekusi yang dipilih untuk memulai pekerjaan pelabelan.

Bagian berikut mencantumkan kebijakan yang mungkin ingin Anda berikan pada peran untuk menggunakan satu atau lebih fungsi Kebenaran Ground Truth.

Topik

Izin Konsol Ground

Untuk memberikan izin kepada pengguna atau peran untuk menggunakan area Ground Truth SageMaker konsol untuk membuat pekerjaan pelabelan, lampirkan kebijakan berikut ini kepada pengguna atau peran. Kebijakan berikut akan memberikan izin peran IAM untuk membuat tugas pelabelan menggunakan jenis tugas tipe tugas bawaan. Jika Anda ingin membuat alur kerja pelabelan kustom, tambahkan kebijakanIzin Alur Kerja Pelabelan Kustom ke kebijakan berikut. Masing-masing`Statement` termasuk dalam kebijakan berikut dijelaskan di bawah blok kode ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SageMakerApis",
            "Effect": "Allow",
            "Action": [
                "sagemaker:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "KmsKeysForCreateForms",
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey",
                "kms:ListAliases"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AccessAwsMarketplaceSubscriptions",
```

```
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ViewSubscriptions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SecretsManager",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListAndCreateExecutionRoles",
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassRoleForExecutionRoles",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        },
        {
            "Sid": "GroundTruthConsole",
            "Effect": "Allow",
            "Action": [
```

```
                "groundtruthlabeling:*",
                "lambda:InvokeFunction",
                "lambda:ListFunctions",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "s3:GetBucketCors",
                "s3:PutBucketCors",
                "s3:ListAllMyBuckets",
                "cognito-idp:AdminAddUserToGroup",
                "cognito-idp:AdminCreateUser",
                "cognito-idp:AdminDeleteUser",
                "cognito-idp:AdminDisableUser",
                "cognito-idp:AdminEnableUser",
                "cognito-idp:AdminRemoveUserFromGroup",
                "cognito-idp:CreateGroup",
                "cognito-idp:CreateUserPool",
                "cognito-idp:CreateUserPoolClient",
                "cognito-idp:CreateUserPoolDomain",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolClient",
                "cognito-idp:ListGroups",
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListUsers",
                "cognito-idp:ListUsersInGroup",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:UpdateUserPool",
                "cognito-idp:UpdateUserPoolClient"
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan ini mencakup pernyataan berikut. Anda dapat cakupan bawah salah satu pernyataan ini dengan menambahkan sumber daya tertentu ke`Resource` daftar untuk pernyataan itu.

`SageMakerApis`

Pernyataan ini mencakup`sagemaker:*`, yang memungkinkan pengguna untuk melakukan semua tindakanSageMaker API. Anda dapat mengurangi cakupan kebijakan ini dengan membatasi

pengguna melakukan tindakan yang tidak digunakan untuk membuat dan memantau pekerjaan pelabelan.

`KmsKeysForCreateForms`

Anda hanya perlu menyertakan pernyataan ini jika Anda ingin memberikan izin pengguna untuk mendaftar dan memilihAWS KMS kunci di konsol Ground Truth yang akan digunakan untuk enkripsi data keluaran. Kebijakan di atas memberikan izin pengguna untuk mendaftar dan memilih kunci apa pun di akunAWS KMS. Untuk membatasi kunci yang dapat didaftarkan dan dipilih pengguna, tentukan ARN kunci tersebut`Resource`.

`SecretsManager`

Pernyataan ini memberikan izin pengguna untuk menjelaskan, daftar, dan membuat sumber dayaAWS Secrets Manager yang diperlukan untuk membuat pekerjaan pelabelan.

`ListAndCreateExecutionRoles`

Pernyataan ini memberikan izin pengguna untuk membuat daftar (`ListRoles`) dan membuat (`CreateRole`) peran IAM di akun Anda. Hal ini juga memberikan izin pengguna untuk membuat (`CreatePolicy`) kebijakan dan melampirkan (`AttachRolePolicy`) kebijakan untuk entitas. Ini diperlukan untuk daftar, pilih, dan jika diperlukan, membuat peran eksekusi di konsol.

Jika Anda telah membuat peran eksekusi, dan ingin mempersempit cakupan pernyataan ini sehingga pengguna hanya dapat memilih peran itu di konsol, tentukan ARN peran yang Anda inginkan agar pengguna memiliki izin untuk melihat`Resource` dan menghapus tindakan`CreateRole`,`CreatePolicy`, dan`AttachRolePolicy`.

`AccessAwsMarketplaceSubscriptions`

Izin ini diperlukan untuk melihat dan memilih tim kerja vendor yang sudah Anda langgani saat membuat pekerjaan pelabelan. Untuk memberikan izin pengguna untuk berlangganan tim kerja vendor, tambahkan pernyataanIzin Tenaga Kerja Vendor ke kebijakan di atas

`PassRoleForExecutionRoles`

Hal ini diperlukan untuk memberikan izin pembuat pekerjaan pelabelan untuk melihat pratinjau UI pekerja dan memverifikasi bahwa data input, label, dan instruksi ditampilkan dengan benar. Pernyataan ini memberikan izin entitas untuk meneruskan peran eksekusi IAM yang digunakan untuk membuat pekerjaan pelabelan SageMaker untuk membuat dan melihat pratinjau UI pekerja. Untuk

mempersempit cakupan kebijakan ini, tambahkan peran ARN dari peran eksekusi yang digunakan untuk membuat pekerjaan pelabelan di bawah`Resource`.

`GroundTruthConsole`

- `groundtruthlabeling`- Ini memungkinkan pengguna untuk melakukan tindakan yang diperlukan untuk menggunakan fitur tertentu dari konsol Ground Truth. Ini termasuk izin untuk mendeskripsikan status pekerjaan pelabelan (`DescribeConsoleJob`), mencantumkan semua objek set data dalam file manifes masukan (`ListDatasetObjects`), memfilter kumpulan data jika pengambilan sampel set data dipilih (`RunFilterOrSampleDatasetJob`), dan untuk menghasilkan file manifes masukan jika pelabelan data otomatis digunakan (`RunGenerateManifestByCrawlingJob`). Tindakan ini hanya tersedia saat menggunakan konsol Ground Truth dan tidak dapat dipanggil langsung menggunakan API.
- `lambda:InvokeFunction`dan`lambda:ListFunctions` — tindakan ini memberi pengguna izin untuk membuat daftar dan memanggil fungsi Lambda yang digunakan untuk menjalankan alur kerja pelabelan khusus.
- `s3:*`— Semua izin Amazon S3 yang disertakan dalam pernyataan ini digunakan untuk melihat bucket Amazon S3 untuk pengaturan data otomatis (`ListAllMyBuckets`), mengakses data input di Amazon S3 (`ListBucket`,`GetObject`), memeriksa dan membuat kebijakan CORS di Amazon S3 jika diperlukan (`GetBucketCorsdanPutBucketCors`), dan menulis pelabelan file output pekerjaan ke S3 (`PutObject`).
- `cognito-idp`- Izin ini digunakan untuk membuat, melihat, dan mengelola serta tenaga kerja pribadi menggunakan Amazon Cognito. Untuk mempelajari lebih lanjut tentang tindakan ini, lihat Referensi API Amazon Cognito.

Izin Alur Kerja Pelabelan Kustom

Tambahkan pernyataan berikut ke kebijakan yang serupa dengan yang ada untuk memberikan izin penggunaIzin Konsol Ground untuk memilih fungsi Lambda pra-anotasi dan pasca-anotasi yang sudah ada sebelumnya sambil membuat alur kerja pelabelan khusus.

```
{
    "Sid": "GroundTruthConsoleCustomWorkflow",
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction",
        "lambda:ListFunctions"
    ],
```

```
    "Resource": "*"
}
```

Untuk mempelajari cara memberikan izin entitas untuk membuat dan menguji fungsi Lambda pra-anotasi dan pasca-anotasi, lihat Izin yang Diperlukan Untuk Menggunakan Lambda Dengan Ground Truth.

Izin Tenaga Kerja Pribadi

Saat ditambahkan ke kebijakan izin, izin berikut memberikan akses untuk membuat dan mengelola tenaga kerja pribadi dan tim kerja menggunakan Amazon Cognito. Izin ini tidak diperlukan untuk menggunakan tenaga kerja IdP OIDC.

```
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:AdminAddUserToGroup",
        "cognito-idp:AdminCreateUser",
        "cognito-idp:AdminDeleteUser",
        "cognito-idp:AdminDisableUser",
        "cognito-idp:AdminEnableUser",
        "cognito-idp:AdminRemoveUserFromGroup",
        "cognito-idp:CreateGroup",
        "cognito-idp:CreateUserPool",
        "cognito-idp:CreateUserPoolClient",
        "cognito-idp:CreateUserPoolDomain",
        "cognito-idp:DescribeUserPool",
        "cognito-idp:DescribeUserPoolClient",
        "cognito-idp:ListGroups",
        "cognito-idp:ListIdentityProviders",
        "cognito-idp:ListUsers",
        "cognito-idp:ListUsersInGroup",
        "cognito-idp:ListUserPoolClients",
        "cognito-idp:ListUserPools",
        "cognito-idp:UpdateUserPool",
        "cognito-idp:UpdateUserPoolClient"
        ],
    "Resource": "*"
}
```

Untuk mempelajari lebih lanjut tentang membuat tenaga kerja pribadi menggunakan Amazon Cognito, lihatMembuat dan Mengelola Amazon Cognito Workforce.

Izin Tenaga Kerja Vendor

Anda dapat menambahkan pernyataan berikut ke kebijakanBerikan Izin IAM untuk Menggunakan Amazon SageMaker Ground Truth Console untuk memberikan izin entitas untuk berlangganan tenaga kerja vendor.

```
{
    "Sid": "AccessAwsMarketplaceSubscriptions",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:Subscribe",
        "aws-marketplace:Unsubscribe",
        "aws-marketplace:ViewSubscriptions"
    ],
    "Resource": "*"
}
```

Buat Peran SageMaker Eksekusi untuk Job Pelabelan Ground Truth

Ketika Anda mengkonfigurasi pekerjaan pelabelan Anda, Anda perlu memberikan peran eksekusi, yang merupakan peran yang SageMaker memiliki izin untuk mengasumsikan untuk memulai dan menjalankan pekerjaan pelabelan Anda.

Peran ini harus memberikan izin Ground Truth untuk mengakses hal-hal berikut:

- Amazon S3 untuk mengambil data input Anda dan menulis data keluaran ke bucket Amazon S3 untuk bucket Amazon S3. Anda dapat memberikan izin untuk peran IAM untuk mengakses seluruh bucket dengan menyediakan ARN bucket, atau Anda dapat memberikan akses ke peran tersebut untuk mengakses sumber daya tertentu dalam bucket. Misalnya, ARN untuk bucket mungkin terlihat mirip dengan`arn:aws:s3:::awsexamplebucket1` dan ARN sumber daya dalam bucket Amazon S3 mungkin terlihat mirip dengan`arn:aws:s3:::awsexamplebucket1/prefix/file-name.png`. Untuk menerapkan tindakan ke semua sumber daya dalam bucket Amazon S3, Anda dapat menggunakan kartu liar:`*`. Sebagai contoh, `arn:aws:s3:::awsexamplebucket1/prefix/*`. Untuk informasi selengkapnya, lihat Amazon Amazon S3 Resources dalam Panduan Pengguna Amazon Simple Storage Storage Service.
- CloudWatch untuk mencatat metrik pekerja dan memberi label status pekerjaan.
- AWS KMSuntuk enkripsi data. (Opsional)
- AWS Lambdauntuk memproses data input dan output saat Anda membuat alur kerja khusus.

Selain itu, jika Anda membuat pekerjaan pelabelan streaming, peran ini harus memiliki izin untuk mengakses:

- Amazon SQS untuk membuat interaksi dengan antrean SQS yang digunakan untuk mengelola permintaan pelabelan.
- Amazon SNS untuk berlangganan dan mengambil pesan dari topik input Amazon SNS Anda dan mengirim pesan ke topik keluaran Amazon SNS Anda.

Semua izin ini dapat diberikan dengan kebijakan yang`AmazonSageMakerGroundTruthExecution` dikelola kecuali:

- Enkripsi volume data dan penyimpanan bucket Amazon S3 Anda. Untuk mempelajari cara mengonfigurasi izin ini, lihatEnkripsi Data Output dan Volume Penyimpanan dengan AWS KMS.
- Izin untuk memilih dan memanggil fungsi Lambda yang tidak menyertakan`GtRecipe`,`SageMaker`,`Sagemaker`,`sagemaker`,, atau`LabelingFunction` dalam nama fungsi.
- Bucket Amazon S3 yang tidak menyertakan`GroundTruth`,`Groundtruth`,`groundtruth`,`SageMakerSagemaker`,,, dan`sagemaker` dalam nama awalan atau bucket atau tag objek yang disertakan`SageMaker` dalam nama (tidak sensitif huruf).

Jika Anda memerlukan izin yang lebih terperinci daripada yang disediakan`AmazonSageMakerGroundTruthExecution`, gunakan contoh kebijakan berikut untuk membuat peran eksekusi yang sesuai dengan kasus penggunaan spesifik Anda.

Topik

- Tipe Tugas Bawaan (Non-streaming) Persyaratan Peran Eksekusi
- Built-In Jenis Tugas (Streaming) Persyaratan Peran Eksekusi
- Persyaratan Peran Eksekusi untuk Jenis Tugas Kustom
- Persyaratan Izin Pelabelan Data Otomatis

Tipe Tugas Bawaan (Non-streaming) Persyaratan Peran Eksekusi

Kebijakan berikut memberikan izin untuk membuat tugas pelabelan untuk jenis tugas bawaan. Kebijakan eksekusi ini tidak menyertakan izin untuk enkripsi atau dekripsiAWS KMS data. Ganti setiap ARN berwarna merah dan dicetak miring dengan ARN Amazon S3 Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ViewBuckets",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::<input-bucket-name>",
                "arn:aws:s3:::<output-bucket-name>"
            ]
        },
        {
            "Sid": "S3GetPutObjects",
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::<input-bucket-name>/*",
                "arn:aws:s3:::<output-bucket-name>/*"
            ]
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Built-In Jenis Tugas (Streaming) Persyaratan Peran Eksekusi

Jika Anda membuat pekerjaan pelabelan streaming, Anda harus menambahkan kebijakan yang serupa dengan yang berikut ini ke peran eksekusi yang Anda gunakan untuk membuat pekerjaan pelabelan. Untuk mempersempit cakupan kebijakan, ganti* in`Resource` denganAWS sumber daya tertentu yang ingin Anda berikan izin peran IAM untuk mengakses dan menggunakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::<input-bucket-name>/*",
                "arn:aws:s3:::<output-bucket-name>/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "s3:ExistingObjectTag/SageMaker": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<input-bucket-name>",
                "arn:aws:s3:::<output-bucket-name>"
            ]
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "StreamingQueue",
            "Effect": "Allow",
            "Action": [
                "sqs:CreateQueue",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage",
                "sqs:SendMessageBatch",
                "sqs:SetQueueAttributes"
            ],
            "Resource": "arn:aws:sqs:*:*:*GroundTruth*"
        },
        {
            "Sid": "StreamingTopicSubscribe",
            "Effect": "Allow",
            "Action": "sns:Subscribe",
            "Resource": [
                "arn:aws:sns:<aws-region>:<aws-account-number>:<input-topic-name>",
                "arn:aws:sns:<aws-region>:<aws-account-number>:<output-topic-name>"
            ],
            "Condition": {
```

```
                "StringEquals": {
                    "sns:Protocol": "sqs"
                },
                "StringLike": {
                    "sns:Endpoint": "arn:aws:sns:<aws-region>:<aws-account-
number>:*GroundTruth*"
                }
            }
        },
        {
            "Sid": "StreamingTopic",
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": [
                "arn:aws:sns:<aws-region>:<aws-account-number>:<input-topic-name>",
                "arn:aws:sns:<aws-region>:<aws-account-number>:<output-topic-name>"
            ]
        },
        {
            "Sid": "StreamingTopicUnsubscribe",
            "Effect": "Allow",
            "Action": [
                "sns:Unsubscribe"
            ],
            "Resource": [
                "arn:aws:sns:<aws-region>:<aws-account-number>:<input-topic-name>",
                "arn:aws:sns:<aws-region>:<aws-account-number>:<output-topic-name>"
            ]
        }
    ]
}
```

Persyaratan Peran Eksekusi untuk Jenis Tugas Kustom

Jika Anda ingin membuat alur kerja pelabelan kustom, tambahkan pernyataan berikut ke kebijakan peran eksekusi seperti yang ditemukan diTipe Tugas Bawaan (Non-streaming) Persyaratan Peran Eksekusi atauBuilt-In Jenis Tugas (Streaming) Persyaratan Peran Eksekusi.

Kebijakan ini memberikan izin peran eksekusi ke fungsi Lambda pra-anotasi dan pasca-anotasi`Invoke` Anda.

```
{
    "Sid": "LambdaFunctions",
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": [
        "arn:aws:lambda:<region>:<account-id>:function:<pre-annotation-lambda-name>",
        "arn:aws:lambda:<region>:<account-id>:function:<post-annotation-lambda-name>"
    ]
}
```

Persyaratan Izin Pelabelan Data Otomatis

Jika Anda ingin membuat tugas pelabelan dengan pelabelan data otomatis diaktifkan, Anda harus 1) menambahkan satu kebijakan ke kebijakan IAM yang dilampirkan pada peran eksekusi dan 2) memperbarui kebijakan kepercayaan dari peran eksekusi.

Pernyataan berikut memungkinkan peran eksekusi IAM diteruskan SageMaker sehingga dapat digunakan untuk menjalankan pekerjaan pelatihan dan inferensi yang digunakan untuk pembelajaran aktif dan pelabelan data otomatis masing-masing. Tambahkan pernyataan ini ke kebijakan peran eksekusi seperti yang ditemukan diTipe Tugas Bawaan (Non-streaming) Persyaratan Peran Eksekusi atauBuilt-In Jenis Tugas (Streaming) Persyaratan Peran Eksekusi. Ganti*arn:aws:iam::<account-number>:role/<role-name>* dengan peran eksekusi ARN. Anda dapat menemukan ARN peran IAM Anda di konsol IAM di bawah Peran.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::<account-number>:role/<execution-role-name>",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "sagemaker.amazonaws.com"
            ]
        }
    }
}
```

Pernyataan berikut memungkinkan SageMaker untuk mengasumsikan peran eksekusi untuk membuat dan mengelola pekerjaan SageMaker pelatihan dan inferensi. Kebijakan ini harus ditambahkan ke hubungan kepercayaan dari peran eksekusi. Untuk mempelajari cara menambahkan atau mengubah kebijakan kepercayaan peran IAM, lihat Memodifikasi peran dalam Panduan Pengguna IAM.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {"Service": "sagemaker.amazonaws.com" },
        "Action": "sts:AssumeRole"
    }
}
```

Enkripsi Data Output dan Volume Penyimpanan denganAWS KMS

Anda dapat menggunakanAWS Key Management Service (AWS KMS) untuk mengenkripsi data keluaran dari pekerjaan pelabelan dengan menentukan kunci yang dikelola pelanggan saat Anda membuat pekerjaan pelabelan. Jika Anda menggunakan operasi API`CreateLabelingJob` untuk membuat tugas pelabelan yang menggunakan pelabelan data otomatis, Anda juga dapat menggunakan kunci yang dikelola pelanggan untuk mengenkripsi volume penyimpanan yang dilampirkan ke instans komputasi ML. untuk menjalankan tugas pelatihan dan inferensi.

Bagian ini menjelaskan kebijakan IAM yang harus Anda lampirkan ke kunci yang dikelola pelanggan untuk mengaktifkan enkripsi data keluaran dan kebijakan yang harus Anda lampirkan ke kunci yang dikelola pelanggan dan peran eksekusi untuk menggunakan enkripsi volume penyimpanan. Untuk mempelajari selengkapnya tentang opsi ini, lihat Data Keluaran dan Enkripsi Volume Penyimpanan.

Enkripsi Data Output menggunakan KMS

Jika Anda menentukan kunci yang dikelolaAWS KMS pelanggan untuk mengenkripsi data keluaran, Anda harus menambahkan kebijakan IAM yang serupa dengan yang berikut ke kunci tersebut. Kebijakan ini memberikan peran eksekusi IAM yang Anda gunakan untuk membuat izin pekerjaan pelabelan untuk menggunakan kunci ini untuk melakukan semua tindakan yang tercantum dalam`"Action"`. Untuk mempelajari lebih lanjut tentang tindakan ini, lihat AWS KMSizin di PanduanAWS Key Management Service Developer.

Untuk menggunakan kebijakan ini, ganti ARN peran layanan IAM`"Principal"` dengan ARN peran eksekusi yang Anda gunakan untuk membuat tugas pelabelan. Saat Anda membuat Job pelabelan di

konsol, ini adalah peran yang Anda tentukan untuk Peran IAM di bawah bagian Ringkasan pekerjaan. Saat Anda membuat pekerjaan pelabelan menggunakan`CreateLabelingJob`, ini adalah ARN yang Anda tentukan `RoleArn`.

```
{
    "Sid": "AllowUseOfKmsKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/service-role/example-role"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Enkripsi Pelabelan Data Otomatis Volume Penyimpanan Instans Komputasi

Jika Anda menentukan `VolumeKmsKeyId`untuk mengenkripsi volume penyimpanan yang dilampirkan ke instans komputasi yang digunakan untuk pelatihan dan inferensi pelabelan data otomatis, Anda harus melakukan hal berikut:

- Lampirkan izin yang dijelaskan padaEnkripsi Data Output menggunakan KMS kunci yang dikelola pelanggan.
- Lampirkan kebijakan yang serupa dengan yang berikut ini dengan peran eksekusi IAM yang Anda gunakan untuk membuat tugas pelabelan Anda. Ini adalah peran IAM yang Anda tentukan`CreateLabelingJob`. `RoleArn` Untuk mempelajari lebih lanjut tentang`"kms:CreateGrant"` tindakan yang diizinkan kebijakan ini, lihat `CreateGrant`di ReferensiAWS Key Management Service API.

```
{
"Version": "2012-10-17",
"Statement":
 [
   {
    "Effect": "Allow",
```

```
    "Action": [
       "kms:CreateGrant"
    ],
    "Resource": "*"
  }
]
}
```

Untuk mempelajari lebih lanjut tentang enkripsi volume penyimpanan Ground Truth, lihatGunakan Kunci KMS Anda untuk Mengenkripsi Volume Penyimpanan Pelabelan Data Otomatis (Hanya API).

Menggunakan Amazon SageMaker Ground Truth di Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) adalah layanan yang dengannya Anda dapat meluncurkan AWS sumber daya dalam jaringan virtual yang terisolasi secara logis yang Anda tentukan. Anda dapat membuat dan menjalankan pekerjaan pelabelan Ground Truth di dalam VPC Amazon alih-alih menghubungkan melalui internet. Saat Anda meluncurkan pekerjaan pelabelan di Amazon VPC, komunikasi antara VPC dan Ground Truth Anda dilakukan sepenuhnya dan aman di dalam jaringan. AWS

Pekerjaan pelabelan Ground Truth di Amazon VPC mengikuti perilaku yang sama dengan PrivateLink Endpoint. Untuk informasi selengkapnya, lihat Subnet bersama.

Panduan ini menunjukkan bagaimana Anda dapat menggunakan Ground Truth di Amazon VPC dengan cara berikut:

1. Jalankan Pekerjaan Pelabelan Kebenaran SageMaker Tanah Amazon di Amazon Virtual Private Cloud
2. Menggunakan Mode Amazon VPC dari Portal Pekerja Pribadi

Jalankan Pekerjaan Pelabelan Kebenaran SageMaker Tanah Amazon di Amazon Virtual Private Cloud

Amazon SageMaker Ground Truth mendukung fungsionalitas berikut.

- Anda dapat menggunakan kebijakan bucket Amazon S3 untuk mengontrol akses ke bucket dari titik akhir Amazon VPC tertentu, atau VPC tertentu. Jika Anda meluncurkan tugas pelabelan dan data input Anda terletak di bucket Amazon S3 dengan akses terbatas pada pengguna di VPC Anda, Anda dapat menambahkan kebijakan bucket untuk juga memberikan izin titik akhir Ground

Truth untuk mengakses bucket. Untuk mempelajari selengkapnya, lihat Izinkan Ground Truth untuk Mengakses Bucket Amazon S3 yang Dibatasi VPC.

- Anda dapat meluncurkan pekerjaan pelabelan data otomatis di VPC Anda. Anda menggunakan konfigurasi VPC untuk menentukan subnet VPC dan grup keamanan. SageMakermenggunakan konfigurasi ini untuk meluncurkan pekerjaan pelatihan dan inferensi yang digunakan untuk pelabelan data otomatis di VPC Anda. Untuk mempelajari selengkapnya, lihat Membuat Pekerjaan Pelabelan Data Otomatis di VPC.

Anda mungkin ingin menggunakan opsi ini dengan salah satu cara berikut.

- Anda dapat menggunakan kedua metode ini untuk meluncurkan tugas pelabelan menggunakan bucket Amazon S3 yang dilindungi VPC dengan pelabelan data otomatis diaktifkan.
- Anda dapat meluncurkan pekerjaan pelabelan menggunakan jenis tugas bawaan apa pun menggunakan bucket yang dilindungi VPC.
- Anda dapat meluncurkan alur kerja pelabelan khusus menggunakan bucket yang dilindungi VPC. Ground Truth berinteraksi dengan fungsi Lambda pra-anotasi dan pasca-anotasi Anda menggunakan titik akhir. AWS PrivateLink

Kami menyarankan Anda meninjau Prasyarat untuk Menjalankan Pekerjaan Pelabelan Kebenaran Tanah di VPC sebelum membuat pekerjaan pelabelan di Amazon VPC.

Prasyarat untuk Menjalankan Pekerjaan Pelabelan Kebenaran Tanah di VPC

Tinjau prasyarat berikut sebelum Anda membuat pekerjaan pelabelan Ground Truth di Amazon VPC.

- Jika Anda adalah pengguna baru Ground Truth, tinjau Memulai untuk mempelajari cara membuat pekerjaan pelabelan.
- Jika data input Anda terletak di bucket Amazon S3 yang dilindungi VPC, pekerja Anda harus mengakses portal pekerja dari VPC Anda.

> **Note**
>
> Ketika Anda meluncurkan pekerjaan pelabelan di VPC Anda, Anda harus menggunakan tim kerja pribadi. Untuk mempelajari lebih lanjut cara membuat tim kerja pribadi, lihat Menggunakan Tenaga Kerja Pribadi.

- Jika Anda ingin meluncurkan pekerjaan pelabelan data otomatis di VPC Anda, tinjau prasyarat berikut.
  - Gunakan petunjuk dalam Membuat Endpoint Amazon S3 VPC. Kontainer pelatihan dan inferensi yang digunakan dalam alur kerja pelabelan data otomatis menggunakan titik akhir ini untuk berkomunikasi dengan bucket Anda di Amazon S3.
  - Tinjau Pelabelan Data Otomatis untuk mempelajari lebih lanjut tentang fitur ini. Perhatikan bahwa pelabelan data otomatis didukung untuk jenis tugas bawaan berikut: Klasifikasi Gambar (Label Tunggal), Segmentasi Semantik Gambar, Kotak Bounding, dan Klasifikasi Teks (Label Tunggal). Pekerjaan pelabelan streaming tidak mendukung pelabelan data otomatis.

- Tinjau bagian Keamanan dan Izin Kebenaran Dasar dan pastikan Anda telah memenuhi ketentuan berikut.
  - Pengguna yang membuat pekerjaan pelabelan memiliki semua izin yang diperlukan
  - Anda telah membuat peran eksekusi IAM dengan izin yang diperlukan. Jika Anda tidak memerlukan izin yang disesuaikan untuk kasus penggunaan Anda, kami sarankan Anda menggunakan kebijakan terkelola IAM yang dijelaskan dalam Memberikan Izin Umum Untuk Memulai Menggunakan Kebenaran Dasar.
  - Izinkan VPC Anda memiliki akses ke bucket `sagemaker-labeling-data-`*`region`* dan `sm-bxcb-`*`region`*`-saved-task-states` S3. Ini adalah bucket S3 regional milik sistem yang diakses dari portal pekerja saat pekerja sedang mengerjakan tugas. Kami menggunakan bucket ini untuk berinteraksi dengan data yang dikelola sistem.

Izinkan Ground Truth untuk Mengakses Bucket Amazon S3 yang Dibatasi VPC

Bagian berikut memberikan detail tentang izin yang diperlukan Ground Truth untuk meluncurkan pekerjaan pelabelan menggunakan bucket Amazon S3 yang memiliki akses terbatas pada titik akhir VPC dan VPC Anda. Untuk mempelajari cara membatasi akses ke bucket Amazon S3 ke VPC, lihat Mengontrol akses dari titik akhir VPC dengan kebijakan bucket dalam panduan Pengguna Amazon Simple Storage Service. Untuk mempelajari cara menambahkan kebijakan ke bucket S3, lihat Menambahkan kebijakan bucket menggunakan konsol Amazon S3.

> **Note**
>
> Memodifikasi kebijakan pada bucket yang ada dapat menyebabkan pekerjaan `IN_PROGRESS` Ground Truth gagal. Kami sarankan Anda memulai pekerjaan baru menggunakan ember

baru. Jika Anda ingin terus menggunakan bucket yang sama, Anda dapat melakukan salah satu dari berikut ini.

- Tunggu `IN_PROGRESS` pekerjaan selesai.
- Mengakhiri pekerjaan menggunakan konsol atau. AWS CLI

Anda dapat membatasi akses bucket Amazon S3 ke pengguna di VPC Anda menggunakan titik akhir. AWS PrivateLink Misalnya, kebijakan bucket S3 berikut memungkinkan akses ke bucket tertentu, dari*<bucket-name>*, *<vpc>* dan titik akhir saja*<vpc-endpoint>*. Saat memodifikasi kebijakan ini, Anda harus mengganti semua *teks beritalisasi merah* dengan sumber daya dan spesifikasi Anda.

Note

Kebijakan berikut menolak semua entitas selain pengguna dalam VPC untuk melakukan tindakan yang tercantum dalam. `Action` Jika Anda tidak menyertakan tindakan dalam daftar ini, tindakan tersebut masih dapat diakses oleh entitas apa pun yang memiliki akses ke bucket ini dan izin untuk melakukan tindakan tersebut. Misalnya, jika pengguna memiliki izin untuk melakukan `GetBucketLocation` di bucket Amazon S3 Anda, kebijakan di bawah ini tidak membatasi pengguna untuk melakukan tindakan ini di luar VPC Anda.

```
{
 "Version": "2012-10-17",
 "Id": "Policy1415115909152",
 "Statement": [
  {
   "Sid": "Access-to-specific-VPCE-only",
   "Principal": "*",
   "Action": [
    "s3:GetObject",
    "s3:PutObject"
   ],
   "Effect": "Deny",
   "Resource": [
    "arn:aws:s3:::<bucket-name>",
    "arn:aws:s3:::<bucket-name>/*"
   ],
   "Condition": {
```

```
    "StringNotEquals": {
     "aws:sourceVpce": [
      "<vpc-endpoint>",
      "<vpc>"
     ]
    }
   }
  }
 ]
}
```

Ground Truth harus dapat melakukan tindakan Amazon S3 berikut pada bucket S3 yang Anda gunakan untuk mengonfigurasi tugas pelabelan.

```
"s3:AbortMultipartUpload",
"s3:GetObject",
"s3:PutObject",
"s3:ListBucket",
"s3:GetBucketLocation"
```

Anda dapat melakukan ini dengan menambahkan titik akhir Ground Truth ke kebijakan bucket seperti yang disebutkan sebelumnya. Tabel berikut mencakup titik akhir layanan Ground Truth untuk setiap AWS Wilayah. Tambahkan titik akhir di AWSWilayah yang sama yang Anda gunakan untuk menjalankan tugas pelabelan ke kebijakan bucket Anda.

Wilayah AWS	Titik akhir Kebenaran Dasar
us–east-2	vpce-02569ba1c40aad0bc
us-east-1	vpce-08408e335ebf95b40
us-west-2	vpce-0ea07aa498eb78469
ca-central-1	vpce-0d46ea4c9ff55e1b7
eu-central-1	vpce-0865e7194a099183d
eu-west-2	vpce-0bccd56798f4c5df0
eu-west-1	vpce-0788e7ed8628e595d

Wilayah AWS	Titik akhir Kebenaran Dasar
ap-south-1	vpce-0d7fcda14e1783f11
ap-southeast-2	vpce-0b7609e6f305a77d4
ap-southeast-1	vpce-0e7e67b32e9efed27
ap-northeast-2	vpce-007893f89e05f2bbf
ap-northeast-1	vpce-0247996a1a1807dbd

Misalnya, kebijakan berikut membatasi `GetObject` dan `PutObject` tindakan pada:

- Bucket Amazon S3 untuk pengguna dalam VPC () `<vpc>`
- Titik akhir VPC () *`<vpc-endpoint>`*
- Titik akhir layanan Kebenaran Tanah () *`<ground-truth-endpoint>`*

```
{
    "Version": "2012-10-17",
    "Id": "1",
    "Statement": [
        {
            "Sid": "DenyAccessFromNonGTandCustomerVPC",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket-name>",
                "arn:aws:s3:::<bucket-name>/*"
            ],
            "Condition": {
                "ForAllValues:StringNotEquals": {
                    "aws:sourceVpce": [
                        "<vpc-endpoint>",
                        "<ground-truth-endpoint>"
                    ],
```

```
                    "aws:SourceVpc": "<vpc>"
                }
            }
        }
    ]
}
```

Jika Anda ingin pengguna memiliki izin untuk meluncurkan pekerjaan pelabelan menggunakan konsol Ground Truth, Anda juga harus menambahkan ARN pengguna ke kebijakan bucket menggunakan `aws:PrincipalArn` kondisi tersebut. Pengguna ini juga harus memiliki izin untuk melakukan tindakan Amazon S3 berikut pada bucket yang Anda gunakan untuk meluncurkan tugas pelabelan.

```
"s3:GetObject",
"s3:PutObject",
"s3:ListBucket",
"s3:GetBucketCors",
"s3:PutBucketCors",
"s3:ListAllMyBuckets",
```

Kode berikut adalah contoh kebijakan bucket yang membatasi izin untuk melakukan tindakan yang tercantum dalam `Action` bucket S3 <`bucket-name`> ke yang berikut ini.

- <role-name>
- Endpoint VPC yang tercantum dalam `aws:sourceVpce`
- Pengguna dalam VPC bernama <vpc>

```
{
    "Version": "2012-10-17",
    "Id": "1",
    "Statement": [
        {
            "Sid": "DenyAccessFromNonGTandCustomerVPC",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket-name>/*",
```

```
                "arn:aws:s3:::<bucket-name>"
            ],
            "Condition": {
                "ForAllValues:StringNotEquals": {
                    "aws:sourceVpce": [
                        "<vpc-endpoint>",
                        "<ground-truth-endpoint>"
                    ],
                    "aws:PrincipalArn": "arn:aws:iam::<aws-account-id>:role/<role-
name>",
                    "aws:SourceVpc": "<vpc>"
                }
            }
        }
    ]
}
```

> **Note**
>
> Titik akhir antarmuka Amazon VPC dan bucket Amazon S3 yang dilindungi yang Anda gunakan untuk data input dan output harus ditempatkan di AWS Wilayah yang sama dengan yang Anda gunakan untuk membuat tugas pelabelan.

Setelah Anda memberikan izin Ground Truth untuk mengakses bucket Amazon S3 Anda, Anda dapat menggunakan salah satu topik di Buat Pekerjaan Pelabelan untuk meluncurkan pekerjaan pelabelan. Tentukan bucket Amazon S3 yang dibatasi VPC untuk bucket data input dan output Anda.

Membuat Pekerjaan Pelabelan Data Otomatis di VPC

Untuk membuat pekerjaan pelabelan data otomatis menggunakan Amazon VPC, Anda menyediakan konfigurasi VPC menggunakan konsol Ground Truth atau operasi `CreateLabelingJob` API. SageMakermenggunakan subnet dan grup keamanan yang Anda berikan untuk meluncurkan pekerjaan pelatihan dan kesimpulan yang digunakan untuk pelabelan otomatis.

> **Important**
>
> Sebelum Anda meluncurkan pekerjaan pelabelan data otomatis dengan konfigurasi VPC, pastikan Anda telah membuat titik akhir Amazon S3 VPC menggunakan VPC yang ingin Anda gunakan untuk pekerjaan pelabelan. Untuk mempelajari caranya, lihat Membuat Endpoint Amazon S3 VPC.

Selain itu, jika Anda membuat pekerjaan pelabelan data otomatis menggunakan bucket Amazon S3 yang dibatasi VPC, Anda harus mengikuti instruksi Izinkan Ground Truth untuk Mengakses Bucket Amazon S3 yang Dibatasi VPC untuk memberikan izin Ground Truth untuk mengakses bucket.

Gunakan prosedur berikut untuk mempelajari cara menambahkan konfigurasi VPC ke permintaan pekerjaan pelabelan Anda.

Tambahkan konfigurasi VPC ke pekerjaan pelabelan data otomatis (konsol):

1. Ikuti petunjuk di Buat Pekerjaan Pelabelan (Konsol) dan selesaikan setiap langkah dalam prosedur, hingga langkah 15.
2. Di bagian Pekerja, pilih kotak centang di sebelah Aktifkan pelabelan data otomatis.
3. Maksimalkan bagian konfigurasi VPC konsol dengan memilih panah.
4. Tentukan Virtual private cloud (VPC) yang ingin Anda gunakan untuk pekerjaan pelabelan data otomatis Anda.
5. Pilih daftar dropdown di bawah Subnet dan pilih satu atau lebih subnet.
6. Pilih daftar tarik-turun di bawah Grup keamanan dan pilih satu atau beberapa grup.
7. Selesaikan semua langkah prosedur yang tersisa di Buat Pekerjaan Pelabelan (Konsol).

Tambahkan konfigurasi VPC ke pekerjaan pelabelan data otomatis (API):

Untuk mengonfigurasi pekerjaan pelabelan menggunakan operasi Ground Truth API`CreateLabelingJob`, ikuti petunjuk di Buat Pekerjaan Pelabelan Data Otomatis (API) untuk mengonfigurasi permintaan Anda. Selain parameter yang dijelaskan dalam dokumentasi ini, Anda harus menyertakan `VpcConfig` parameter `LabelingJobResourceConfig` untuk menentukan satu atau lebih subnet dan grup keamanan menggunakan skema berikut.

```
"LabelingJobAlgorithmsConfig": {
      "InitialActiveLearningModelArn": "string",
      "LabelingJobAlgorithmSpecificationArn": "string",
      "LabelingJobResourceConfig": {
         "VolumeKmsKeyId": "string",
         "VpcConfig": {
            "SecurityGroupIds": [ "string" ],
            "Subnets": [ "string" ]
```

```
        }
    }
}
```

Berikut ini adalah contoh permintaan AWS Python SDK (Boto3) untuk membuat pekerjaan pelabelan data otomatis di Wilayah AS Timur (Virginia Utara) menggunakan tenaga kerja pribadi. Ganti semua *teks miring merah* dengan sumber daya dan spesifikasi pekerjaan pelabelan Anda. Untuk mempelajari lebih lanjut tentang `CreateLabelingJob` operasi, lihat tutorial Create a Labeling Job (API) dan dokumentasi CreateLabelingJobAPI.

```
import boto3
client = boto3.client(service_name='sagemaker')

response = client.create_labeling_job(
    LabelingJobName="example-labeling-job",
    LabelAttributeName="label",
    InputConfig={
        'DataSource': {
            'S3DataSource': {
                'ManifestS3Uri': "s3://bucket/path/manifest-with-input-data.json"
            }
        }
    },
    "LabelingJobAlgorithmsConfig": {
      "LabelingJobAlgorithmSpecificationArn": "arn:aws:sagemaker:us-
east-1:027400017018:labeling-job-algorithm-specification/tasktype",
      "LabelingJobResourceConfig": {
         "VpcConfig": {
            "SecurityGroupIds": [ "sg-01233456789", "sg-987654321" ],
            "Subnets": [ "subnet-e0123456", "subnet-e7891011" ]
         }
      }
    },
    OutputConfig={
        'S3OutputPath': "s3://bucket/path/file-to-store-output-data",
        'KmsKeyId': "string"
    },
    RoleArn="arn:aws:iam::*:role/*",
    LabelCategoryConfigS3Uri="s3://bucket/path/label-categories.json",
    StoppingConditions={
        'MaxHumanLabeledObjectCount': 123,
        'MaxPercentageOfInputDatasetLabeled': 123
    },
```

```
    HumanTaskConfig={
        'WorkteamArn': "arn:aws:sagemaker:region:*:workteam/private-crowd/*",
        'UiConfig': {
            'UiTemplateS3Uri': "s3://bucket/path/custom-worker-task-template.html"
        },
        'PreHumanTaskLambdaArn': "arn:aws:lambda:us-
east-1:432418664414:function:PRE-tasktype",
        'TaskKeywords': [
            "Images",
            "Classification",
            "Multi-label"
        ],
        'TaskTitle': "Add task title here",
        'TaskDescription': "Add description of task here for workers",
        'NumberOfHumanWorkersPerDataObject': 1,
        'TaskTimeLimitInSeconds': 3600,
        'TaskAvailabilityLifetimeInSeconds': 21600,
        'MaxConcurrentTaskCount': 1000,
        'AnnotationConsolidationConfig': {
            'AnnotationConsolidationLambdaArn': "arn:aws:lambda:us-
east-1:432418664414:function:ACS-tasktype"
        },
    Tags=[
        {
            'Key': "string",
            'Value': "string"
        },
    ]
)
```

Menggunakan Mode Amazon VPC dari Portal Pekerja Pribadi

Untuk membatasi akses portal pekerja ke pelabel yang bekerja di dalam VPC Amazon, Anda dapat menambahkan konfigurasi VPC saat membuat tenaga kerja pribadi Ground Truth. Anda juga dapat menambahkan konfigurasi VPC ke tenaga kerja pribadi yang ada. Ground Truth secara otomatis membuat titik akhir antarmuka VPC di VPC Anda dan mengatur AWS PrivateLink antara titik akhir VPC Anda dan layanan Ground Truth. URL portal pekerja yang terkait dengan tenaga kerja dapat diakses dari VPC Anda. URL portal pekerja juga dapat diakses dari internet publik hingga Anda menetapkan pembatasan di internet publik. Ketika Anda menghapus tenaga kerja atau menghapus konfigurasi VPC dari tenaga kerja Anda, Ground Truth secara otomatis menghapus titik akhir VPC yang terkait dengan tenaga kerja.

> **Note**
>
> Hanya ada satu VPC yang didukung untuk tenaga kerja.

Point Cloud dan tugas video tidak mendukung pemuatan melalui VPC.

Panduan ini menunjukkan cara menyelesaikan langkah-langkah yang diperlukan untuk menambah dan menghapus konfigurasi Amazon VPC ke tenaga kerja Anda, dan memenuhi prasyarat.

Prasyarat

Untuk menjalankan pekerjaan pelabelan Ground Truth di Amazon VPC, tinjau prasyarat berikut.

- Anda memiliki Amazon VPC yang dikonfigurasi yang dapat Anda gunakan. Jika Anda belum mengkonfigurasi VPC, ikuti petunjuk ini untuk membuat VPC.
- Bergantung pada cara Template Tugas Pekerja ditulis, data pelabelan yang disimpan dalam bucket Amazon S3 dapat diakses langsung dari Amazon S3 selama tugas pelabelan. Dalam kasus ini, jaringan VPC harus dikonfigurasi untuk memungkinkan lalu lintas dari perangkat yang digunakan oleh pemberi label manusia ke bucket S3 yang berisi data pelabelan.
- Ikuti Lihat dan perbarui atribut DNS untuk VPC Anda untuk mengaktifkan nama host DNS dan resolusi DNS untuk VPC Anda.

> **Note**
>
> Ada dua cara untuk mengonfigurasi VPC Anda untuk tenaga kerja Anda. Anda dapat melakukan ini melalui konsol atau AWS SageMaker CLI.

Menggunakan SageMaker konsol untuk mengelola konfigurasi VPC

Anda dapat menggunakan SageMakerkonsol untuk menambah atau menghapus konfigurasi VPC. Anda juga dapat menghapus tenaga kerja yang ada.

Menambahkan konfigurasi VPC ke tenaga kerja Anda

Buat tenaga kerja pribadi

- Membuat tenaga kerja pribadi menggunakan Amazon Cognito

- Buat tenaga kerja pribadi menggunakan OpenID Connect (OIDC) Identity Provider (iDP).

Setelah Anda membuat tenaga kerja pribadi Anda, tambahkan konfigurasi VPC ke dalamnya.

1. Arahkan ke Amazon SageMaker Runtime di konsol Anda.
2. Pilih Pelabelan tenaga kerja di panel kiri.
3. Pilih Pribadi untuk mengakses tenaga kerja pribadi Anda. Setelah status Tenaga Kerja Anda Aktif, pilih Tambah di samping VPC.
4. Ketika Anda diminta untuk mengkonfigurasi VPC Anda, berikan yang berikut ini:
   a. VPC Anda
   b. Subnet
      i. Pastikan VPC Anda memiliki subnet yang sudah ada
   c. Grup keamanan
      i.

      > **Note**
      >
      > Anda tidak dapat memilih lebih dari 5 grup keamanan.

   d. Setelah mengisi informasi ini, pilih Konfirmasi.
5. Setelah Anda memilih Konfirmasi, Anda akan diarahkan kembali ke halaman Pribadi di bawah Tenaga kerja pelabelan. Anda akan melihat spanduk hijau di bagian atas yang membaca pembaruan tenaga kerja pribadi Anda dengan konfigurasi VPC berhasil diinisialisasi. Status tenaga kerja adalah Memperbarui. Di samping tombol Hapus tenaga kerja adalah tombol Refresh, yang dapat digunakan untuk mengambil status Tenaga Kerja terbaru. Setelah status tenaga kerja berubah menjadi Aktif, ID titik akhir VPC juga diperbarui.

Menghapus konfigurasi VPC dari tenaga kerja Anda

Gunakan informasi berikut untuk menghapus konfigurasi VPC dari tenaga kerja Anda menggunakan konsol.

1. Arahkan ke Amazon SageMaker Runtime di konsol Anda.
2. Pilih Pelabelan tenaga kerja di panel kiri.
3. Temukan dan pilih tenaga kerja Anda.
4. Di bawah Ringkasan tenaga kerja pribadi, temukan VPC dan pilih Hapus di sebelahnya.

5. Pilih Hapus.

Menghapus tenaga kerja melalui konsol

Jika Anda menghapus tenaga kerja, Anda seharusnya tidak memiliki tim yang terkait dengannya. Anda dapat menghapus tenaga kerja hanya jika status tenaga kerja Aktif atau Gagal.

Gunakan informasi berikut untuk menghapus tenaga kerja menggunakan konsol.

1. Arahkan ke Amazon SageMaker Runtime di konsol Anda.
2. Pilih Pelabelan tenaga kerja di panel kiri.
3. Temukan dan pilih tenaga kerja Anda.
4. Pilih Hapus tenaga kerja.
5. Pilih Delete (Hapus).

Menggunakan SageMaker AWS API untuk mengelola konfigurasi VPC

Download file berikut untuk menggunakan `VpcConfig` parameter baru ke dalam CLI SageMaker tenaga kerja:

sagemaker-2017-07-24.normal.json

sagemaker-2017-07-24.paginators.json

sagemaker-2017-07-24.pelayan-2.json

Setelah mengunduh file, jalankan perintah berikut di CLI Anda:

```
aws configure add-model --service-model file://./sagemaker-2017-07-24.normal.json --
service-name sagemaker
```

```
cp ./sagemaker-2017-07-24.paginators.json ~/.aws/models/sagemaker/2017-07-24/
paginators.json
```

```
cp ./sagemaker-2017-07-24.waiters-2.json ~/.aws/models/sagemaker/2017-07-24/
waiters-2.json
```

Anda sekarang dapat menguji perubahan API Anda menggunakan AWS CLI. Anda dapat membuat tenaga kerja baru dengan konfigurasi VPC atau memperbarui tenaga kerja yang ada untuk menambahkan konfigurasi VPC. Anda juga dapat menghapus konfigurasi VPC dari tenaga kerja yang ada.

Buat tenaga kerja dengan konfigurasi VPC

Jika akun sudah memiliki tenaga kerja, maka Anda harus menghapusnya terlebih dahulu. Anda juga dapat memperbarui tenaga kerja dengan konfigurasi VPC.

```
aws sagemaker create-workforce --cognito-config '{"ClientId": "app-client-
id","UserPool": "Pool_ID",}' --workforce-vpc-config \
" {\"VpcId\": \"vpc-id\", \"SecurityGroupIds\": [\"sg-0123456789abcdef0\"], \"Subnets
\": [\"subnet-0123456789abcdef0\"]}" --workforce-name workforce-name
{
    "WorkforceArn": "arn:aws:sagemaker:us-west-2:xxxxxxxxx:workforce/workforce-name"
}
```

Jelaskan tenaga kerja dan pastikan statusnya. `Initializing`

```
aws sagemaker describe-workforce --workforce-name workforce-name
{
    "Workforce": {
        "WorkforceName": "workforce-name",
        "WorkforceArn": "arn:aws:sagemaker:us-west-2:xxxxxxxxx:workforce/workforce-
name",
        "LastUpdatedDate": 1622151252.451,
        "SourceIpConfig": {
            "Cidrs": []
        },
        "SubDomain": "subdomain.us-west-2.sagamaker.aws.com",
        "CognitoConfig": {
            "UserPool": "Pool_ID",
            "ClientId": "app-client-id"
        },
        "CreateDate": 1622151252.451,
        "WorkforceVpcConfig": {
            "VpcId": "vpc-id",
            "SecurityGroupIds": [
```

```
                "sg-0123456789abcdef0"
            ],
            "Subnets": [
                "subnet-0123456789abcdef0"
            ]
        },
        "Status": "Initializing"
    }
}
```

Arahkan ke konsol Amazon VPC. Pilih Endpoint dari panel kiri. Harus ada dua titik akhir VPC yang dibuat di akun Anda.

Menambahkan konfigurasi VPC tenaga kerja Anda

Perbarui tenaga kerja pribadi non-VPC dengan konfigurasi VPC menggunakan perintah berikut.

```
aws sagemaker update-workforce --workforce-name workforce-name\
--workforce-vpc-config "{\"VpcId\": \"vpc-id\", \"SecurityGroupIds\":
 [\"sg-0123456789abcdef0\"], \"Subnets\": [\"subnet-0123456789abcdef0\"]}"
```

Jelaskan tenaga kerja dan pastikan statusnya. `Updating`

```
aws sagemaker describe-workforce --workforce-name workforce-name
{
    "Workforce": {
        "WorkforceName": "workforce-name",
        "WorkforceArn": "arn:aws:sagemaker:us-west-2:xxxxxxxxx:workforce/workforce-
name",
        "LastUpdatedDate": 1622151252.451,
        "SourceIpConfig": {
            "Cidrs": []
        },
        "SubDomain": "subdomain.us-west-2.sagamaker.aws.com",
        "CognitoConfig": {
            "UserPool": "Pool_ID",
            "ClientId": "app-client-id"
        },
        "CreateDate": 1622151252.451,
```

```
        "WorkforceVpcConfig": {
            "VpcId": "vpc-id",
            "SecurityGroupIds": [
                "sg-0123456789abcdef0"
            ],
            "Subnets": [
                "subnet-0123456789abcdef0"
            ]
        },
        "Status": "Updating"
    }
}
```

Arahkan ke konsol Amazon VPC Anda. Pilih Endpoint dari panel kiri. Harus ada dua titik akhir VPC yang dibuat di akun Anda.

Menghapus konfigurasi VPC dari tenaga kerja Anda

Perbarui tenaga kerja pribadi VPC dengan konfigurasi VPC kosong untuk menghapus sumber daya VPC.

```
aws sagemaker update-workforce --workforce-name workforce-name\
--workforce-vpc-config "{}"
```

Jelaskan tenaga kerja dan pastikan statusnya. `Updating`

```
aws sagemaker describe-workforce --workforce-name workforce-name
{
    "Workforce": {
        "WorkforceName": "workforce-name",
        "WorkforceArn": "arn:aws:sagemaker:us-west-2:xxxxxxxxx:workforce/workforce-
name",
        "LastUpdatedDate": 1622151252.451,
        "SourceIpConfig": {
            "Cidrs": []
        },
        "SubDomain": "subdomain.us-west-2.sagamaker.aws.com",
        "CognitoConfig": {
```

```
            "UserPool": "Pool_ID",
            "ClientId": "app-client-id"
        },
        "CreateDate": 1622151252.451,
        "Status": "Updating"
    }
}
```

Naviagate ke konsol Amazon VPC Anda. Pilih Endpoint dari panel kiri. Kedua titik akhir VPC harus dihapus.

Batasi akses publik ke portal pekerja sambil mempertahankan akses melalui VPC

Para pekerja di portal pekerja VPC atau non-VPC dapat melihat tugas pekerjaan pelabelan yang ditugaskan kepada mereka. Penugasan ini berasal dari menugaskan pekerja dalam tim kerja melalui kelompok OIDC. Merupakan tanggung jawab pelanggan untuk membatasi akses ke portal pekerja publik mereka dengan menetapkan tenaga kerja mereka. `sourceIpConfig`

> **Note**
>
> Anda dapat membatasi akses ke portal pekerja hanya melalui SageMaker API. Ini tidak dapat dilakukan melalui konsol.

Gunakan perintah berikut untuk membatasi akses publik ke portal pekerja.

```
aws sagemaker update-workforce --region us-west-2 \
--workforce-name workforce-demo --source-ip-config '{"Cidrs":["10.0.0.0/16"]}'
```

Setelah `sourceIpConfig` diatur pada tenaga kerja, para pekerja dapat mengakses portal pekerja di VPC tetapi tidak melalui internet publik.

> **Note**
>
> Anda tidak dapat mengatur `sourceIP` pembatasan untuk portal pekerja di VPC.

Data Keluaran dan Enkripsi Volume Penyimpanan

Dengan Amazon SageMaker Ground Truth, Anda dapat memberi label pada data yang sangat sensitif, tetap memegang kendali atas data Anda, dan menerapkan praktik terbaik keamanan. Saat tugas pelabelan Anda berjalan, Ground Truth mengenkripsi data saat transit dan saat tidak aktif. Selain itu, Anda dapat menggunakanAWS Key Management Service (AWS KMS) dengan Ground Truth untuk melakukan hal berikut:

- Gunakan kunci yang dikelola pelanggan untuk mengenkripsi data keluaran Anda.
- Gunakan kunci yang dikelolaAWS KMS pelanggan dengan tugas pelabelan data otomatis Anda untuk mengenkripsi volume penyimpanan yang dilampirkan ke instans komputasi yang digunakan untuk pelatihan model dan inferensi.

Gunakan topik di halaman ini untuk mempelajari lebih lanjut tentang fitur keamanan Ground Truth ini.

Gunakan Kunci KMS Anda untuk Mengenkripsi Data Keluaran

Secara opsional, Anda dapat memberikan kunci yang dikelolaAWS KMS pelanggan saat membuat pekerjaan pelabelan, yang digunakan Ground Truth untuk mengenkripsi data keluaran Anda.

Jika Anda tidak memberikan kunci terkelola pelanggan, Amazon SageMaker menggunakan defaultKunci yang dikelola AWS untuk Amazon S3 untuk akun peran Anda untuk mengenkripsi data keluaran Anda.

Jika Anda memberikan kunci terkelola pelanggan, Anda harus menambahkan izin yang diperlukan ke kunci yang dijelaskan diEnkripsi Data Output dan Volume Penyimpanan denganAWS KMS. Saat Anda menggunakan operasi API`CreateLabelingJob`, Anda dapat menentukan ID kunci yang dikelola pelanggan menggunakan parameter`KmsKeyId`. Lihat prosedur berikut untuk mempelajari cara menambahkan kunci yang dikelola pelanggan saat Anda membuat pekerjaan pelabelan menggunakan konsol.

Untuk menambahkanAWS KMS kunci untuk mengenkripsi data keluaran (konsol):

1. Selesaikan 7 langkah pertamaMembuat Job Pelabelan (Konsol).
2. Pada langkah 8, pilih panah di sebelah Konfigurasi tambahan untuk memperluas bagian ini.
3. Untuk kunci Encryption, pilihAWS KMS kunci yang ingin Anda gunakan untuk mengenkripsi data keluaran.

4. Selesaikan langkah-langkah lainnyaMembuat Job Pelabelan (Konsol) untuk membuat pekerjaan pelabelan.

Gunakan Kunci KMS Anda untuk Mengenkripsi Volume Penyimpanan Pelabelan Data Otomatis (Hanya API)

Ketika Anda membuat tugas pelabelan dengan pelabelan data otomatis menggunakan operasi`CreateLabelingJob` API, Anda memiliki opsi untuk mengenkripsi volume penyimpanan yang melekat pada instans komputasi MS yang menjalankan tugas pelatihan dan inferensi. Untuk menambahkan enkripsi ke volume penyimpanan Anda, gunakan parameter`VolumeKmsKeyId` untuk memasukkan kunci yang dikelolaAWS KMS pelanggan. Untuk informasi selengkapnya tentang parameter ini, lihat`LabelingJobResourceConfig`.

Jika Anda menentukan ID kunci atau ARN untuk`VolumeKmsKeyId`, peran SageMaker eksekusi Anda harus menyertakan izin untuk memanggil`kms:CreateGrant`. Untuk mempelajari cara menambahkan izin ini ke peran eksekusi, lihatBuat Peran SageMaker Eksekusi untuk Job Pelabelan Ground Truth.

> **Note**
>
> Jika Anda menentukan kunci yang dikelolaAWS KMS pelanggan saat membuat pekerjaan pelabelan di konsol, kunci tersebut hanya digunakan untuk mengenkripsi data keluaran Anda. Ini tidak digunakan untuk mengenkripsi volume penyimpanan yang dilampirkan ke instans komputasi yang digunakan untuk pelabelan data otomatis.

Otentikasi dan Pembatasan Tenaga Kerja

Ground Truth memungkinkan Anda menggunakan tenaga kerja pribadi Anda sendiri untuk mengerjakan pekerjaan pelabelan. Tenaga kerja pribadi adalah konsep abstrak yang mengacu pada seperangkat orang yang bekerja untuk Anda. Setiap pekerjaan pelabelan dibuat menggunakan tim kerja, terdiri dari pekerja di tenaga kerja Anda. Ground Truth mendukung pembuatan tenaga kerja pribadi menggunakan Amazon Cognito.

Tenaga kerja Ground Truth memetakan ke kolam pengguna Amazon Cognito. Tim kerja Ground Truth memetakan tim pengguna Amazon Cognito. Amazon Cognito mengelola autentikasi pekerja. Amazon Cognito mendukung koneksi Open ID (OIDC) dan pelanggan dapat mengatur federasi Amazon Cognito dengan penyedia identitas mereka sendiri (IdP).

Ground Truth hanya mengizinkan satu tenaga kerja per akun perAWS Wilayah. Setiap tenaga kerja memiliki URL login portal Ground Truth yang berdedikasi.

Anda juga dapat membatasi pekerja ke rentang alamat Blok/IP Classless Inter-Domain Routing (CIDR). Ini berarti anotator harus berada di jaringan tertentu untuk mengakses situs anotasi. Anda dapat menambahkan hingga sepuluh blok CIDR untuk satu tenaga kerja. Untuk mempelajari selengkapnya, lihat Mengelola Tenaga Kerja Pribadi Menggunakan Amazon SageMaker API.

Untuk mempelajari bagaimana Anda dapat membuat tenaga kerja pribadi, lihatMembuat Tenaga Kerja Pribadi (Amazon Cognito).

Batasi Akses ke Jenis Tenaga Kerja

Tim kerja Amazon SageMaker Ground Truth termasuk dalam salah satu dari tiga jenis tenaga kerja: publik (dengan Amazon Mechanical Turk), swasta, dan vendor. Untuk membatasi akses pengguna ke tim kerja tertentu menggunakan salah satu jenis ini atau tim kerja ARN, gunakan`sagemaker:WorkteamType` dan/atau kunci`sagemaker:WorkteamArn` kondisi. Untuk kunci`sagemaker:WorkteamType` kondisi, menggunakan operator kondisi string. Untuk kunci`sagemaker:WorkteamArn` ketentuan, gunakan operator ketentuan Amazon Resource Name (ARN). Jika pengguna mencoba membuat pekerjaan pelabelan dengan tim kerja terbatas, SageMaker mengembalikan kesalahan akses yang ditolak.

Kebijakan di bawah ini menunjukkan berbagai cara untuk menggunakan`sagemaker:WorkteamType` dan kunci`sagemaker:WorkteamArn` kondisi dengan operator kondisi yang sesuai dan nilai kondisi yang valid.

Contoh berikut menggunakan kunci`sagemaker:WorkteamType` kondisi dengan operator`StringEquals` kondisi untuk membatasi akses ke tim kerja umum. Ia menerima nilai-nilai kondisi dalam format berikut:*`workforcetype`*`-crowd`, di mana *`workforcetype`* dapat sama`public`,`private`, atau`vendor`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RestrictWorkteamType",
            "Effect": "Deny",
            "Action": "sagemaker:CreateLabelingJob",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "sagemaker:WorkteamType": "public-crowd"
                }
            }
        }
    ]
}
```

Kebijakan berikut menunjukkan cara membatasi akses ke tim kerja umum menggunakan kunci`sagemaker:WorkteamArn` kondisi. Yang pertama menunjukkan bagaimana menggunakannya dengan IAM regex-varian valid dari tim kerja ARN dan operator`ArnLike` kondisi. Yang kedua menunjukkan bagaimana menggunakannya dengan operator`ArnEquals` kondisi dan tim kerja ARN.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RestrictWorkteamType",
            "Effect": "Deny",
            "Action": "sagemaker:CreateLabelingJob",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "sagemaker:WorkteamArn": "arn:aws:sagemaker:*:*:workteam/public-
crowd/*"
                }
            }
        }
    ]
}
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RestrictWorkteamType",
            "Effect": "Deny",
            "Action": "sagemaker:CreateLabelingJob",
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "sagemaker:WorkteamArn": "arn:aws:sagemaker:us-
west-2:394669845002:workteam/public-crowd/default"
```

```
                }
            }
        }
    ]
}
```

Status Job Pelabelan Monitor

Untuk memantau status tugas pelabelan Anda, Anda dapat menyiapkanAmazon CloudWatch Peristiwa(CloudWatch Acara) aturan untuk Amazon SageMaker Ground Truth (Ground Truth) untuk mengirim acara ke CloudWatch Peristiwa saat status pekerjaan pelabelan berubah menjadi`Completed`,`Failed`, atau`Stopped`atau ketika pekerja menerima, menolak, mengirimkan, atau mengembalikan tugas.

Setelah Anda membuat aturan, Anda dapat menambahkan`target`untuk itu. CloudWatch Peristiwa menggunakan target ini untuk memanggil yang lainAWSlayanan untuk memproses acara. Misalnya, Anda dapat membuat target menggunakan topik Amazon Simple Notification Service (Amazon SNS) untuk mengirim notifikasi ke email ketika status tugas pelabelan berubah.

Prasyarat:

Untuk membuat CloudWatch Aturan acara, Anda akan memerlukanAWS Identity and Access Management(IAM) peran dengan kebijakan kepercayaan events.amazonaws.com terlampir. Berikut ini adalah contoh kebijakan kepercayaan events.amazonaws.com.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Topik

Kirim Acara ke CloudWatch Peristiwa

Untuk mengonfigurasi CloudWatch Aturan peristiwa untuk mendapatkan pembaruan status, atauacara, untuk pekerjaan pelabelan Ground Truth Anda, gunakanAWS Command Line Interface(AWS CLI)`put-rule`perintah. Anda dapat memfilter peristiwa yang dikirim ke aturan Anda berdasarkan perubahan status. Misalnya, Anda dapat membuat aturan yang memberi tahu Anda hanya jika status tugas pelabelan berubah menjadi`Completed`. Saat menggunakan`put-rule`perintah, tentukan berikut untuk menerima status pekerjaan pelabelan:

- `\"source\":[\"aws.sagemaker\"]`
- `\"detail-type\":[\"SageMaker Ground Truth Labeling Job State Change\"]`

Untuk mengonfigurasi CloudWatch Aturan peristiwa untuk menonton semua perubahan status, gunakan perintah berikut dan ganti teks placeholder. Misalnya, ganti*"GTLabelingJobStateChanges"*dengan unik CloudWatch Nama aturan peristiwa dan*"arn:aws:iam::111122223333:role/MyRoleForThisRule"*dengan Amazon Resource Number (ARN) dari peran IAM dengan kebijakan kepercayaan events.amazonaws.com terlampir.

```
aws events put-rule --name "GTLabelingJobStateChanges"
    --event-pattern "{\"source\":[\"aws.sagemaker\"],\"detail-type\":[\"SageMaker
 Ground Truth Labeling Job State Change\"]}"
    --role-arn "arn:aws:iam::111122223333:role/MyRoleForThisRule"
    --region "region"
```

Untuk memfilter berdasarkan status pekerjaan, gunakan`\"detail\":{\"LabelingJobStatus\":[\"`*Status*`\"]}}`"sintaks. Nilai yang valid untuk *Status* adalah `Completed`, `Failed`, dan `Stopped`.

Contoh berikut membuat CloudWatch Aturan peristiwa yang memberi tahu Anda saat pekerjaan pelabelan di us-west-2 (Oregon) berubah menjadi`Completed`.

```
aws events put-rule --name "LabelingJobCompleted"
    --event-pattern "{\"source\":[\"aws.sagemaker\"],\"detail-type\":[\"SageMaker
 Ground Truth Labeling Job State Change\"], \"detail\":{\"LabelingJobStatus\":
[\"Completed\"]}}"
    --role-arn "arn:aws:iam::111122223333:role/MyRoleForThisRule"
    --region us-west-2
```

Contoh berikut membuat CloudWatch Aturan acara yang memberi tahu Anda saat pekerjaan pelabelan di us-east-1 (Virginia) berubah`Completed`atau`Failed`.

```
aws events put-rule --name "LabelingJobCompletedOrFailed"
    --event-pattern "{\"source\":[\"aws.sagemaker\"],\"detail-type\":[\"SageMaker
 Ground Truth Labeling Job State Change\"], \"detail\":{\"LabelingJobStatus\":
[\"Completed\", \"Failed\"]}}"
    --role-arn "arn:aws:iam::111122223333:role/MyRoleForThisRule"
    --region us-east-1
```

Untuk mempelajari lebih lanjut tentang`put-rule`permintaan, lihatPola Peristiwa di CloudWatch Peristiwadi dalamAmazon CloudWatch Panduan Pengguna Events.

Menyiapkan Target untuk Memproses Acara

Setelah Anda membuat aturan, peristiwa yang mirip dengan berikut ini dikirim ke CloudWatch Peristiwa. Dalam contoh ini, pekerjaan pelabelan`test-labeling-job`status berubah menjadi`Completed`.

```
{
    "version": "0",
    "id": "111e1111-11d1-111f-b111-1111b11dcb11",
    "detail-type": "SageMaker Ground Truth Labeling Job State Change",
    "source": "aws.sagemaker",
    "account": "111122223333",
    "time": "2018-10-06T12:26:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:sagemaker:us-east-1:111122223333:labeling-job/test-labeling-job"
    ],
    "detail": {
        "LabelingJobStatus": "Completed"
    }
}
```

Untuk memproses peristiwa, Anda perlu menyiapkan target. Misalnya, jika Anda ingin menerima email saat status pekerjaan pelabelan Anda berubah, gunakan prosedur diMengatur Pemberitahuan Amazon SNSdi dalamAmazon CloudWatch Panduan Penggunauntuk menyiapkan topik Amazon SNS dan berlangganlah email Anda ke topik tersebut. Setelah Anda membuat topik, Anda dapat menggunakannya untuk membuat target.

Untuk menambahkan target ke target Anda CloudWatch Aturan acara

1. Buka CloudWatch konsol:https://console.aws.amazon.com/cloudwatch/home
2. Di panel navigasi, pilih Aturan.
3. Pilih aturan yang ingin Anda tambahkan target ke.
4. Pilih Tindakan, dan kemudian pilih Edit.
5. Di bawahTarget, chooseTambah Targetdan pilihlahAWSlayanan yang ingin Anda lakukan saat peristiwa perubahan status pekerjaan pelabelan terdeteksi.
6. Konfigurasi target Anda. Untuk instruksi, lihat topik untuk mengonfigurasi target diAWSdokumentasi untuk layanan tersebut.
7. Pilih Konfigurasikan detail.
8. UntukNama, masukkan nama dan, secara opsional, berikan detail tentang tujuan aturan diDeskripsi.
9. Pastikan kotak centang kotak di sampingnegara bagiandipilih sehingga aturan Anda terdaftar sebagaiDiaktifkan.
10. PilihMemperbarui aturan pembaruan.

Masa kedaluwarsa Job pelabelan

Jika pekerjaan pelabelan Anda tidak selesai setelah 30 hari, itu akan kedaluwarsa. Jika pekerjaan pelabelan Anda kedaluwarsa, Anda dapat merantai pekerjaan untuk membuat pekerjaan pelabelan baru yang hanya akan mengirim data tanpa label kepada pekerja. Untuk informasi lebih lanjut, dan untuk mempelajari cara membuat pekerjaan pelabelan menggunakan chaining, lihatTugas Pelabelan Rantai.

Tugas menurun

Pekerja dapat menolak tugas.

Pekerja menolak tugas jika instruksi tidak jelas, data input tidak ditampilkan dengan benar, atau jika mereka mengalami masalah lain dengan tugas tersebut. Jika jumlah pekerja per objek

dataset (NumberOfHumanWorkersPerDataObject) Menolak tugas, objek data ditandai sebagai kadaluarsa dan tidak akan dikirim ke pekerja tambahan.

Gunakan Amazon SageMaker Ground Truth Plus untuk Label Data

Amazon SageMaker Ground Truth Plus adalah layanan pelabelan data turnkey yang menggunakan tenaga kerja ahli untuk memberikan anotasi berkualitas tinggi dengan cepat dan mengurangi biaya hingga 40%. Menggunakan SageMaker Ground Truth Plus, ilmuwan data dan manajer bisnis, seperti manajer operasi data dan manajer program, dapat membuat kumpulan data pelatihan berkualitas tinggi tanpa harus membangun aplikasi pelabelan dan mengelola tenaga kerja pelabelan sendiri. Anda dapat memulai Amazon SageMaker Ground Truth Plus dengan mengunggah data bersama dengan persyaratan pelabelan di Amazon S3.

Mengapa menggunakan SageMaker Ground Truth Plus?

Untuk melatih model pembelajaran mesin (ML), ilmuwan data membutuhkan kumpulan data berlabel yang besar dan berkualitas tinggi. Seiring berkembangnya adopsi ML, kebutuhan pelabelan meningkat. Ini memaksa ilmuwan data untuk menghabiskan waktu berminggu-minggu untuk membangun alur kerja pelabelan data dan mengelola tenaga kerja pelabelan data. Sayangnya, ini memperlambat inovasi dan meningkatkan biaya. Untuk memastikan ilmuwan data dapat menghabiskan waktu mereka membangun, melatih, dan menerapkan model ML, ilmuwan data biasanya menugaskan tim internal lainnya yang terdiri dari manajer operasi data dan manajer program untuk menghasilkan kumpulan data pelatihan berkualitas tinggi. Namun, tim-tim ini biasanya tidak memiliki akses ke keterampilan yang diperlukan untuk memberikan kumpulan data pelatihan berkualitas tinggi, yang memengaruhi hasil ML. Hasilnya, Anda mencari mitra pelabelan data yang dapat membantu mereka membuat kumpulan data pelatihan berkualitas tinggi dalam skala besar tanpa menghabiskan sumber daya internal mereka.

Saat Anda mengunggah data, SageMaker Ground Truth Plus menyiapkan alur kerja pelabelan data dan mengoperasikannya atas nama Anda. Dari sana, tenaga kerja ahli yang dilatih pada berbagai tugas pembelajaran mesin (ML) melakukan pelabelan data. SageMaker Ground Truth Plus saat ini menawarkan dua jenis tenaga kerja ahli: tenaga kerja yang dipekerjakan Amazon dan daftar vendor pihak ketiga yang dikuratori. SageMaker Ground Truth Plus memberi Anda fleksibilitas untuk memilih tenaga kerja pelabelan. AWSpara ahli memilih tenaga kerja pelabelan terbaik berdasarkan kebutuhan proyek Anda. Misalnya, jika Anda membutuhkan orang yang mahir dalam memberi label file audio, tentukan itu dalam pedoman yang diberikan kepada SageMaker Ground Truth Plus, dan layanan secara otomatis memilih pelabel dengan keterampilan tersebut.

> **Important**
>
> SageMaker Ground Truth Plus tidak mendukung data bersertifikat PHI, PCI atau FedRAMP, dan Anda tidak boleh memberikan data ini ke Ground Truth Plus. SageMaker

Bagaimana cara kerja SageMaker Ground Truth Plus?

Ada lima komponen utama dalam alur kerja.

- Meminta proyek
- Membuat tim proyek
- Mengakses portal proyek untuk memantau kemajuan kumpulan data pelatihan dan meninjau data berlabel
- Membuat batch
- Menerima data berlabel

Bagaimana cara menggunakan SageMaker Ground Truth Plus?

Jika Anda adalah pengguna pertama kali SageMaker Ground Truth Plus, gunakan Memulai dengan Amazon SageMaker Ground Truth Plus. start. Untuk mengakses SageMaker Ground Truth Plus menggunakan SageMaker konsol, Anda harus berada di US East (Virginia N.) (`us-east-1`).

Memulai dengan Amazon SageMaker Ground Truth Plus.

Panduan ini menunjukkan cara menyelesaikan langkah-langkah yang diperlukan untuk memulai proyek Amazon SageMaker Ground Truth Plus, meninjau label, dan memenuhi prasyarat SageMaker Ground Truth Plus.

Untuk mulai menggunakan SageMaker Ground Truth Plus, tinjau Mengatur Prasyarat Amazon SageMaker Ground Truth Plus danKomponen Inti dari Amazon SageMaker Ground Truth Plus.

Mengatur Prasyarat Amazon SageMaker Ground Truth Plus

Gunakan informasi berikut untuk mendaftar AWS akun. Jika sudah memiliki akun AWS, Anda dapat melewati langkah ini.

Mendaftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk secara online.

   Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

   Saat Anda mendaftar Akun AWS, Pengguna root akun AWS akan dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, tetapkan akses administratif ke pengguna administratif, dan hanya gunakan pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Membuat pengguna administratif

Setelah mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat sebuah pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke AWS Management Console sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Di halaman berikutnya, masukkan kata sandi Anda.

   Untuk bantuan masuk menggunakan pengguna root, lihat Masuk sebagai pengguna root dalam Panduan Pengguna AWS Sign-In.
2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuknya, silakan lihat Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda (konsol) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center.
2. Di Pusat Identitas IAM, berikan akses administratif ke sebuah pengguna administratif.

   Untuk mendapatkan tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, silakan lihat Mengonfigurasi akses pengguna dengan Direktori Pusat Identitas IAM default di Panduan Pengguna AWS IAM Identity Center.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal akses AWS dalam Panduan Pengguna AWS Sign-In.

Komponen Inti dari Amazon SageMaker Ground Truth Plus

Istilah-istilah berikut adalah kunci untuk memahami kemampuan SageMaker Ground Truth Plus:

- Proyek: Setiap keterlibatan yang memenuhi syarat dengan seorang AWS ahli menghasilkan proyek SageMaker Ground Truth Plus. Sebuah proyek dapat dalam tahap percontohan atau produksi.
- Batch: Batch adalah kumpulan objek data berulang serupa seperti gambar, bingkai video, dan teks yang akan diberi label. Sebuah proyek dapat memiliki beberapa batch.
- Metrik: Metrik adalah data tentang proyek SageMaker Ground Truth Plus Anda untuk tanggal tertentu atau lebih dari rentang tanggal.
- Jenis tugas: SageMaker Ground Truth Plus mendukung lima tipe tugas untuk pelabelan data. Anda juga dapat memiliki jenis tugas khusus. Ini termasuk teks, gambar, video, audio, dan cloud titik 3D.
- Objek data: Item individual yang akan diberi label.

Minta Proyek

Untuk menggunakan AmazonSageMakerGround Truth Plus, mulailah dengan meminta proyek.

1. Di bawah tab Kebenaran Tanah AmazonSageMaker, pilihDitambah.
2. PadaSageMakerTanah Kebenaran Ditambahhalaman, pilihMinta proyek.
3. Sebuah halaman berjudulMeminta proyekterbuka. Halaman ini mencakup bidang untukInformasi UmumdanIkhtisar proyek. Masukkan informasi berikut

   a. Di bawahInformasi Umum, masukkanNama pertama,Nama belakangdanAlamat email bisnis. SebuahAWSahli menggunakan informasi ini untuk menghubungi Anda untuk mendiskusikan proyek setelah Anda mengirimkan permintaan.

   b. Di bawahIkhtisar proyek, masukkanNama proyekdanDeskripsi proyek. PilihJenis tugasberdasarkan data dan kasus penggunaan Anda. Anda juga dapat menunjukkan apakah data Anda berisi informasi identitas pribadi (PII).

   c. Membuat atau memilih peran IAM yang memberikanSageMakerIzin Ground Truth Plus untuk melakukan pekerjaan pelabelan dengan memilih salah satu opsi di bawah ini.

      i. Anda bisaMembuat peran IAMyang menyediakan akses ke bucket S3 yang Anda tentukan.

      ii. Anda bisaMasukkan peran IAM kustom ARN.

      iii. Anda dapat memilih peran yang ada.

      iv. Jika Anda menggunakan peran yang ada atau peran IAM kustom ARN, pastikan Anda memiliki peran IAM berikut dan kebijakan kepercayaan.

      IAM role

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::your-bucket-name",
                "arn:aws:s3:::your-bucket-name/*"
                //Ex: "arn:aws:s3:::input-data-to-label/*"
            ]
```

```
        }
    ]
}
```

Kebijakan kepercayaan

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker-ground-truth-plus.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

4. PilihMeminta proyek.

Setelah Anda membuat proyek, Anda dapat menemukannya diSageMakerTanah Kebenaran Ditambahhalaman, di bawah bagian Proyek. Status proyek harusTinjau dalam proses

> **Note**
>
> Anda tidak dapat memiliki lebih dari 5 proyek denganTinjau sedang berlangsungstatus.

Buat proyek proyek

Tim proyek menyediakan akses ke anggota dari organisasi atau tim Anda untuk melacak proyek, melihat metrik, dan meninjau anotasi. Anda dapat membuat tim proyek SageMaker Ground Truth Plus setelah Anda membagikan data Anda di bucket Amazon S3.

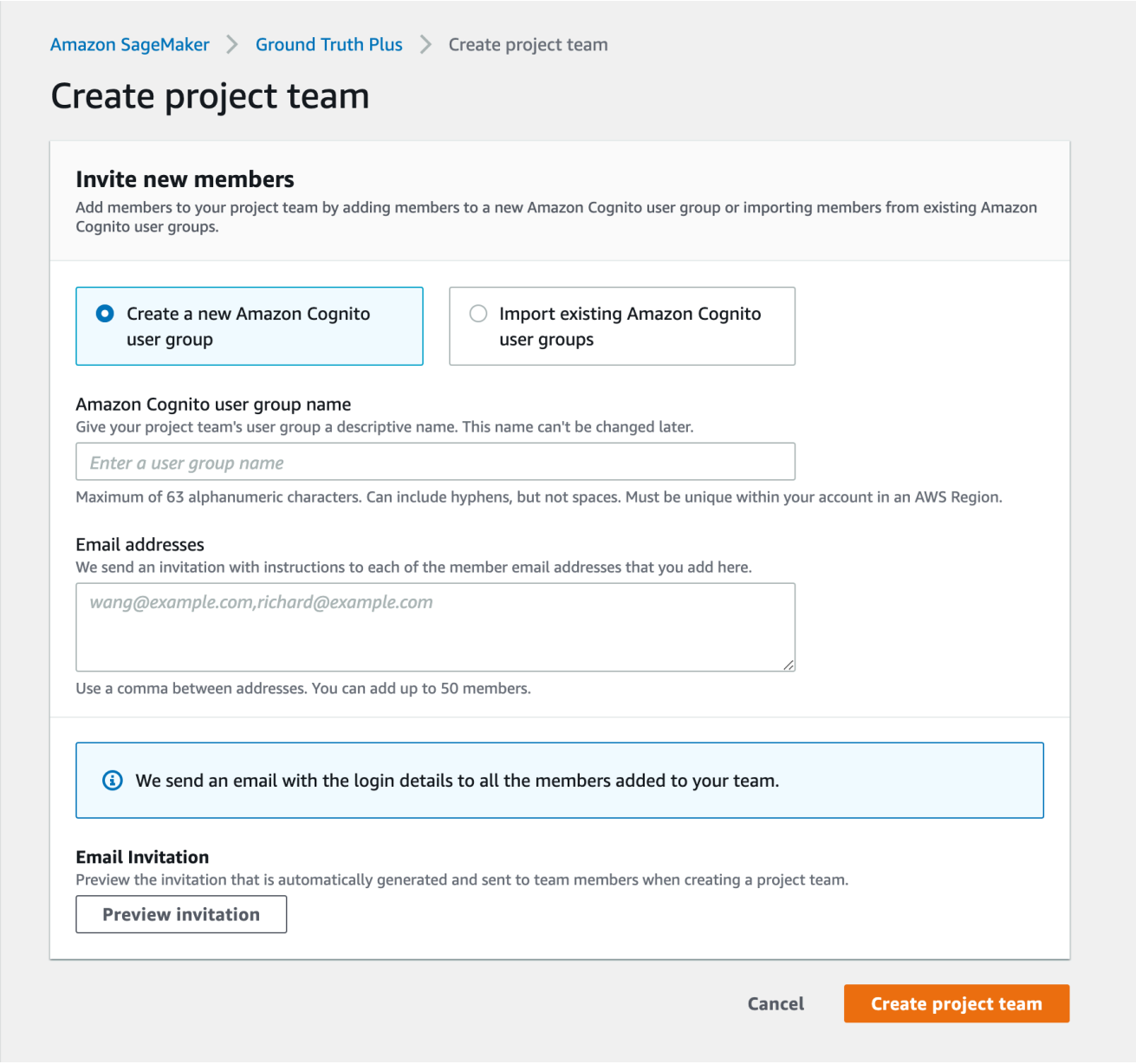
Untuk menambahkan anggota tim menggunakan Amazon Cognito, Anda memiliki dua opsi:

1. Buat grup pengguna Amazon Cognito baru

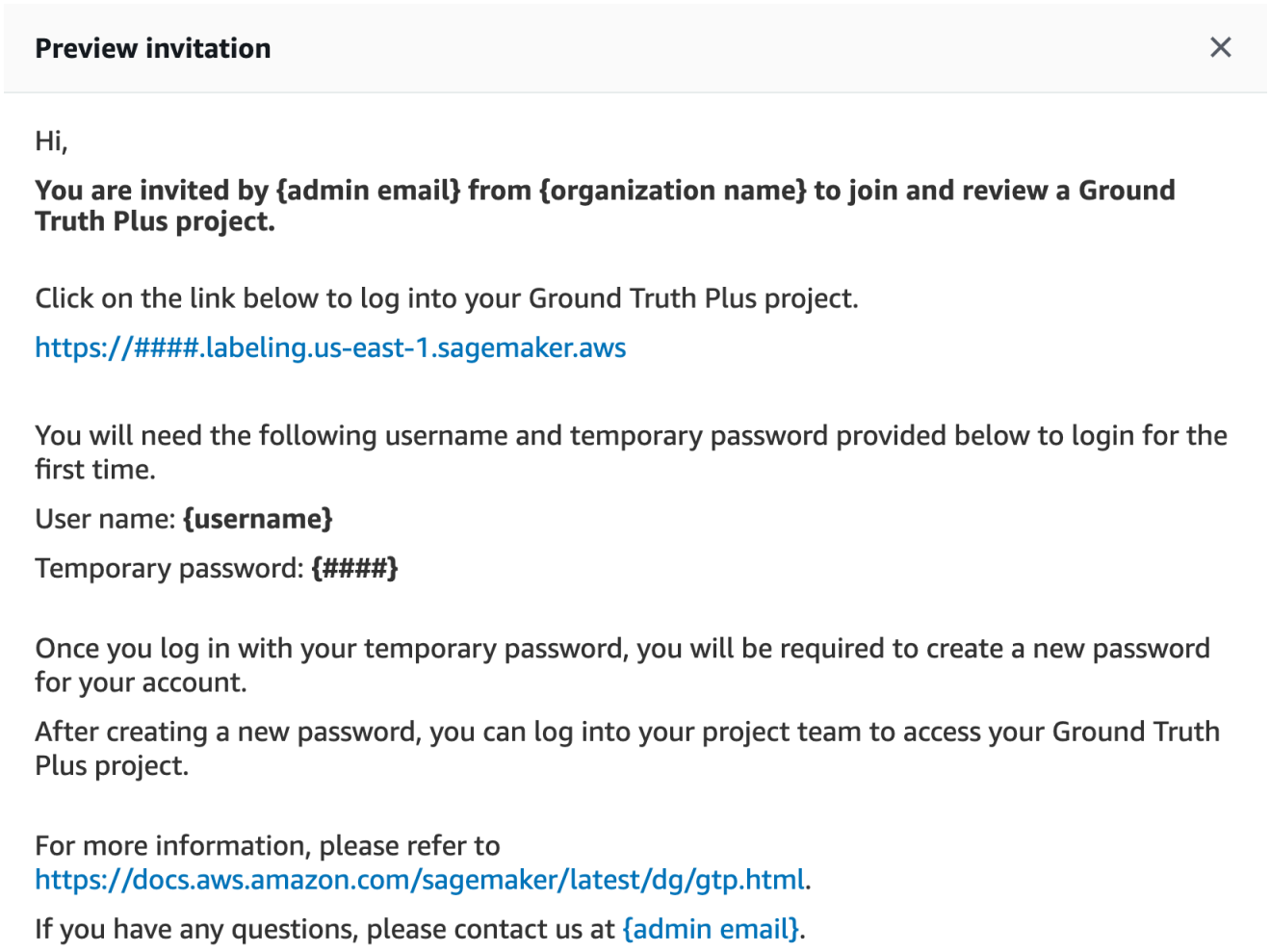
    a. Masukkan nama grup pengguna Amazon Cognito. Nama ini tidak dapat diubah.

b. Masukkan alamat email hingga 50 anggota tim di bidang Alamat email. Alamat harus dipisahkan dengan koma.

c. Pilih Buat tim proyek.



d. Anggota tim Anda menerima email yang mengundang mereka untuk bergabung dengan tim proyek SageMaker Ground Truth Plus seperti yang ditunjukkan pada gambar berikut.



2. Impor anggota tim dari grup pengguna Amazon Cognito yang ada.

   a. Pilih kumpulan pengguna yang telah Anda buat. Kumpulan pengguna memerlukan domain dan grup pengguna yang ada. Jika Anda mendapatkan kesalahan karena domain hilang, setel di opsi nama Domain di halaman integrasi aplikasi konsol Amazon Cognito untuk grup Anda.

   b. Pilih klien aplikasi. Sebaiknya gunakan klien yang dihasilkan oleh Amazon SageMaker.

   c. Pilih grup pengguna dari pangkalan Anda untuk mengimpor anggotanya.

   d. Pilih Buat tim proyek.

Anda dapat melihat dan mengelola daftar anggota tim melaluiAWS konsol.

Untuk menambahkan anggota tim setelah membuat tim proyek:

1. Pilih Undang anggota baru di bagian Anggota.
2. Masukkan alamat email hingga 50 anggota tim di bidang Alamat email. Alamat harus dipisahkan dengan koma.
3. Pilih Undang anggota baru

Untuk menghapus anggota tim yang ada:

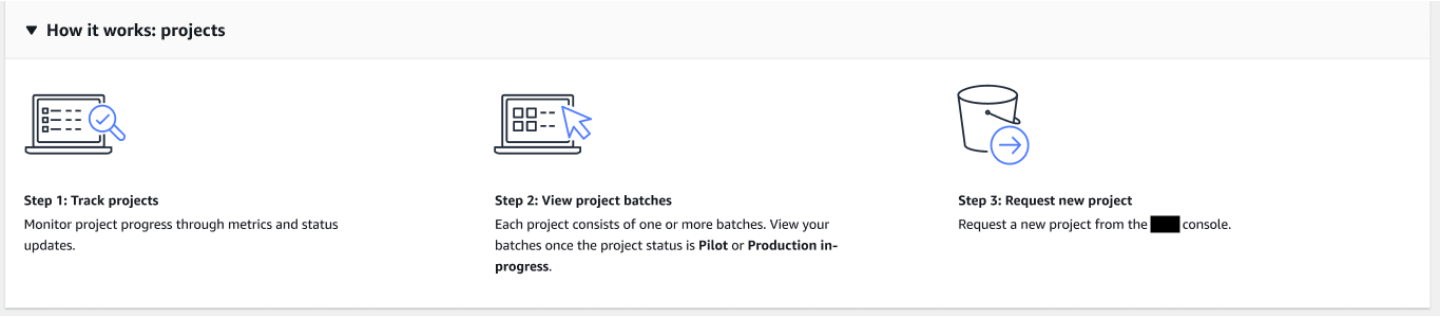
1. Pilih anggota tim yang akan dihapus di bagian Anggota.
2. Pilih Delete (Hapus).

Setelah Anda menambahkan anggota ke tim proyek Anda, Anda dapat membuka portal proyek untuk mengakses proyek Anda.

Buka Portal Proyek

Setelah Anda berhasil mengirimkan formulir asupan dan membuat tim proyek, Anda dapat mengakses proyek SageMaker Ground Truth Plus dengan memilih portal proyek Terbuka diAWS konsol.

Setiap satu atau beberapa kumpulan terdiri dari satu atau beberapa. Batch adalah kumpulan berulang objek data serupa (teks, gambar, bingkai video, dan titik awan) yang akan diberi label. Portal proyek memberi Anda transparansi ke dalam proses pelabelan data. Anda dapat tetap diperbarui tentang proyek, membuat batch dalam proyek, meninjau kemajuan kumpulan data di beberapa proyek, dan menganalisis metrik proyek. Portal proyek juga memungkinkan Anda untuk meninjau bagian dari data berlabel dan memberikan umpan balik. Anda dapat mengkonfigurasi kolom yang ditampilkan dalam proyek dan tabel batch Anda.



Anda dapat menggunakan portal proyek SageMaker Ground Truth Plus untuk melacak detail berikut tentang proyek Anda.

Nama proyek: Setiap proyek diidentifikasi menggunakan nama yang unik.

Status: Proyek SageMaker Ground Truth Plus memiliki salah satu tipe status berikut:

1. Tinjauan yang sedang berlangsung: Anda telah berhasil mengirimkan formulir permintaan proyek. SeorangAWS ahli saat ini sedang meninjau permintaan Anda.
2. Permintaan disetujui: Permintaan proyek Anda disetujui. Anda sekarang dapat membagikan data Anda dengan membuat batch baru dari portal proyek.
3. Desain alur kerja dan kemajuan penyiapan: SeorangAWS ahli sedang menyiapkan proyek Anda.
4. Pilot sedang berlangsung: Pelabelan objek untuk proyek di tahap percontohan saat ini sedang berlangsung.
5. Pilot selesai: Pelabelan objek selesai dan data berlabel disimpan di bucket Amazon S3 Anda.
6. Harga selesai: SeorangAWS ahli berbagi harga untuk proyek produksi dengan Anda.
7. Kontrak dieksekusi: Kontrak selesai.
8. Produksi sedang berlangsung: Pelabelan untuk proyek dalam tahap produksi sedang berlangsung.
9. Produksi selesai: Pelabelan objek selesai dan data berlabel disimpan di bucket Amazon S3 Anda.
10. Dijeda: Proyek saat ini dijeda atas permintaan Anda.

Jenis tugas: SageMaker Ground Truth Plus memungkinkan Anda memberi label lima jenis tugas yang mencakup teks, gambar, video, audio, dan titik awan.

Batch: Jumlah total batch dalam suatu proyek.

Tanggal pembuatan proyek: Tanggal mulai proyek.

Total objek: Jumlah total objek yang akan diberi label di semua batch.

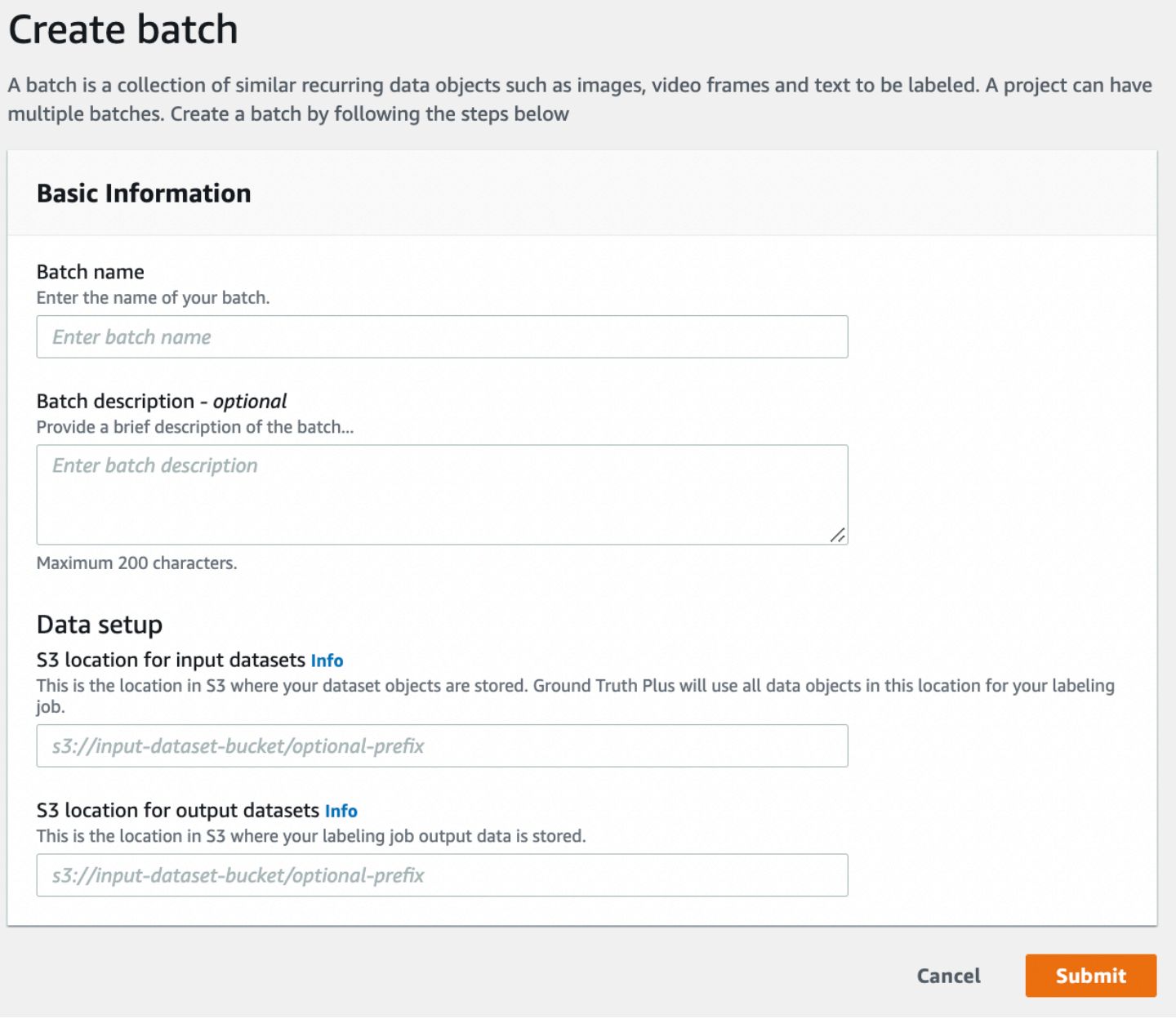
Objek selesai: Jumlah objek berlabel.

Objek yang tersisa: Jumlah benda yang tersisa untuk diberi label.

Objek gagal: Jumlah objek yang tidak dapat diberi label karena masalah dengan data input.

Batch

Anda dapat menggunakan portal proyek untuk membuat batch untuk proyek setelah status proyek diubah menjadi Permintaan disetujui.



Untuk membuat bagi, lakukan hal berikut.

1. Pilih proyek dengan memilih nama proyek.
2. Halaman berjudul dengan nama proyek terbuka. Di bawah bagian Batch, pilih Buat batch.
3. Masukkan nama Batch, deskripsi Batch, lokasi S3 untuk set data input, dan lokasi S3 untuk kumpulan data keluaran.

4. Pilih Submit (Kirim).

Untuk membuat bagi, pastikan Anda memenuhi kriteria berikut:

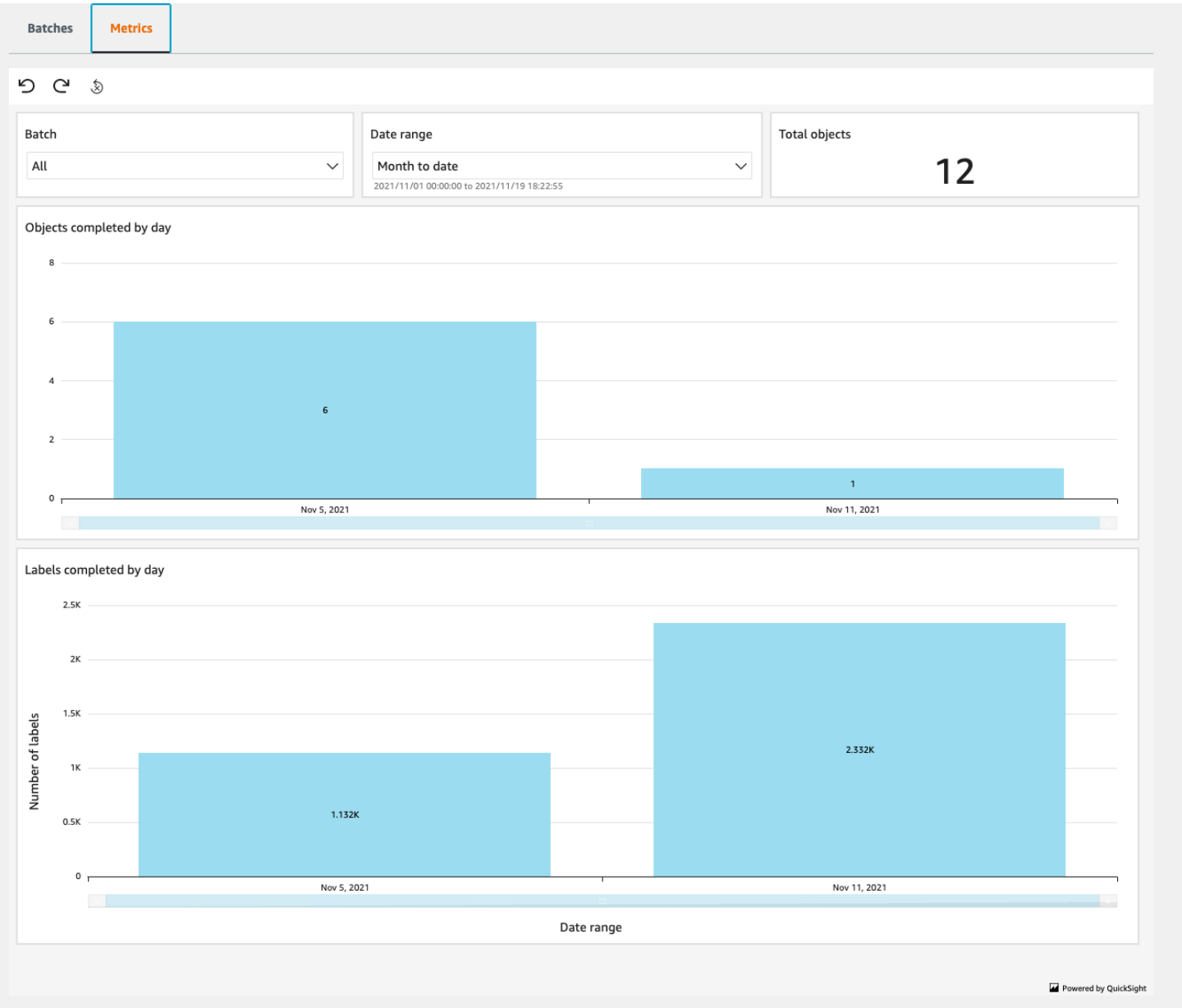
- Data Anda di US East (N. Virginia).
- Ukuran maksimum untuk setiap file tidak lebih dari 2 gigabyte.
- Jumlah maksimum file dalam batch adalah 10.000.
- Ukuran total batch kurang dari 100 gigabyte.
- Anda memiliki tidak lebih dari 5 batch dengan status transfer data yang sedang berlangsung.

> **Note**
>
> Anda tidak dapat membuat kumpulan sebelum status proyek berubah Permintaan disetujui.

Memeriksa Metrik

Metrik adalah data tentang proyek SageMaker Ground Truth Plus Anda untuk tanggal tertentu atau lebih dari rentang tanggal.

Anda dapat meninjau metrik untuk semua batch atau memilih kumpulan pilihan Anda seperti yang ditunjukkan pada gambar berikut.



Anda dapat meninjau metrik berikut tentang kumpulan:

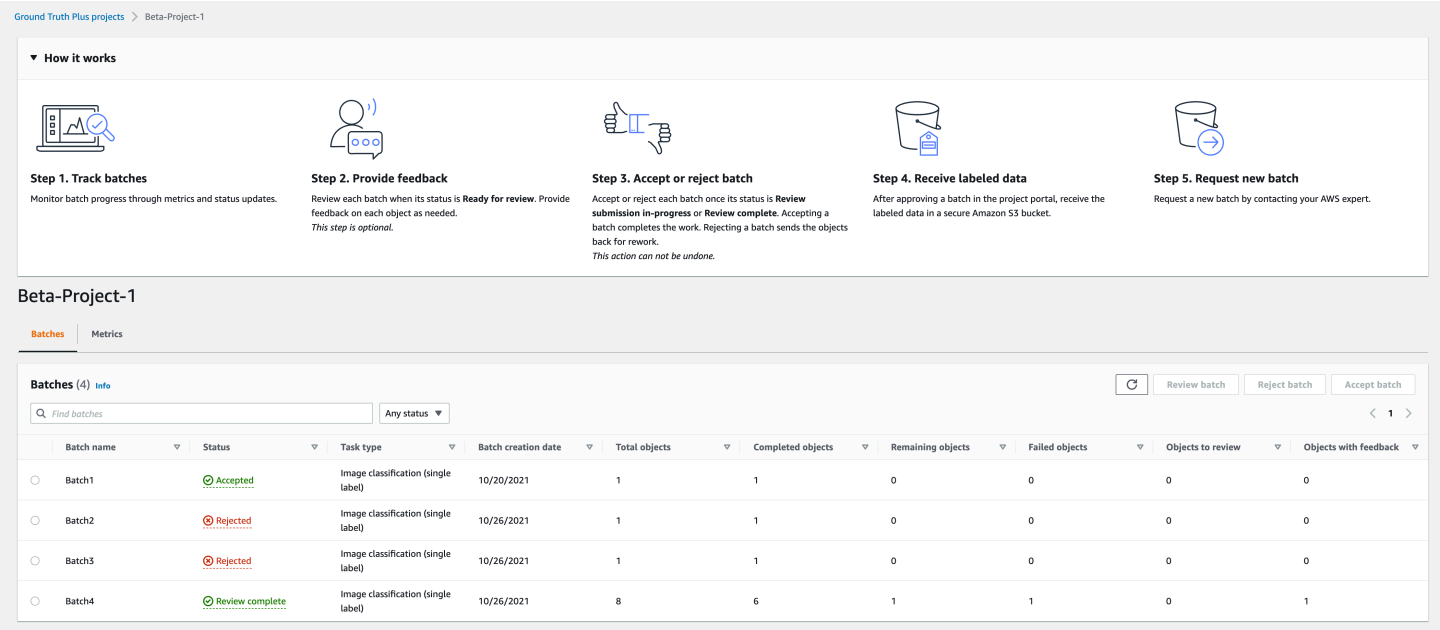
Total objek: Jumlah total objek dalam batch atau di semua batch.

Objek diselesaikan berdasarkan hari: Jumlah total objek yang diberi label pada tanggal tertentu atau di atas rentang tanggal.

Label diselesaikan berdasarkan hari: Jumlah total label yang diselesaikan pada tanggal tertentu atau lebih dari rentang tanggal. Objek dapat memiliki lebih dari satu label.

Tinjau Batch

Setiap proyek Amazon SageMaker Ground Truth Plus terdiri dari satu atau lebih batch. Setiap batch terdiri dari objek data yang akan diberi label. Anda dapat melihat semua batch untuk proyek Anda menggunakan portal proyek seperti yang ditunjukkan pada gambar berikut.



Anda dapat menggunakan portal proyek SageMaker Ground Truth Plus untuk melacak detail berikut tentang setiap batch:

Nama Batch: Setiap batch diidentifikasi dengan nama batch yang unik.

Status: Batch SageMaker Ground Truth Plus memiliki salah satu dari jenis status berikut:

1. Permintaan dikirimkan: Anda telah berhasil mengirimkan batch baru.
2. Transfer data gagal: Transfer data gagal dengan kesalahan. Periksa alasan kesalahan dan buat batch baru setelah memperbaiki kesalahan.
3. Data yang diterima: Kami telah menerima data input Anda yang tidak berlabel.
4. Dalam proses: Pelabelan data sedang berlangsung.
5. Siap untuk ditinjau: Pelabelan data selesai. Subset objek berlabel dari batch siap untuk Anda tinjau. Ini adalah langkah opsional.
6. Kiriman ulasan sedang berlangsung: Umpan balik ulasan saat ini sedang diproses.
7. Tinjau selesai: Anda telah berhasil meninjau batch. Selanjutnya, Anda harus menerima atau menolaknya. Tindakan ini tidak dapat dibatalkan.

8. Diterima: Anda telah menerima data berlabel dan akan segera menerimanya di bucket Amazon S3 Anda.
9. Ditolak: Data berlabel perlu dikerjakan ulang.
10. Dikirim untuk pengerjaan ulang: Data berlabel dikirim untuk pengerjaan ulang. Anda dapat meninjau batch setelah statusnya berubah menjadi Siap untuk ditinjau.
11. Siap untuk pengiriman: Data berlabel siap ditransfer ke bucket Amazon S3 Anda.
12. Data yang dikirim: Pelabelan objek selesai dan data berlabel disimpan di bucket Amazon S3 Anda.
13. Dijeda: Batch dijeda sesuai permintaan Anda.

Jenis tugas: SageMaker Ground Truth Plus memungkinkan Anda memberi label lima jenis tugas yang mencakup teks, gambar, video, audio, dan titik awan.

Tanggal pembuatan Batch: Tanggal kapan batch dibuat.

Total objek: Jumlah total objek yang akan diberi label di seluruh batch.

Objek selesai: Jumlah objek berlabel.

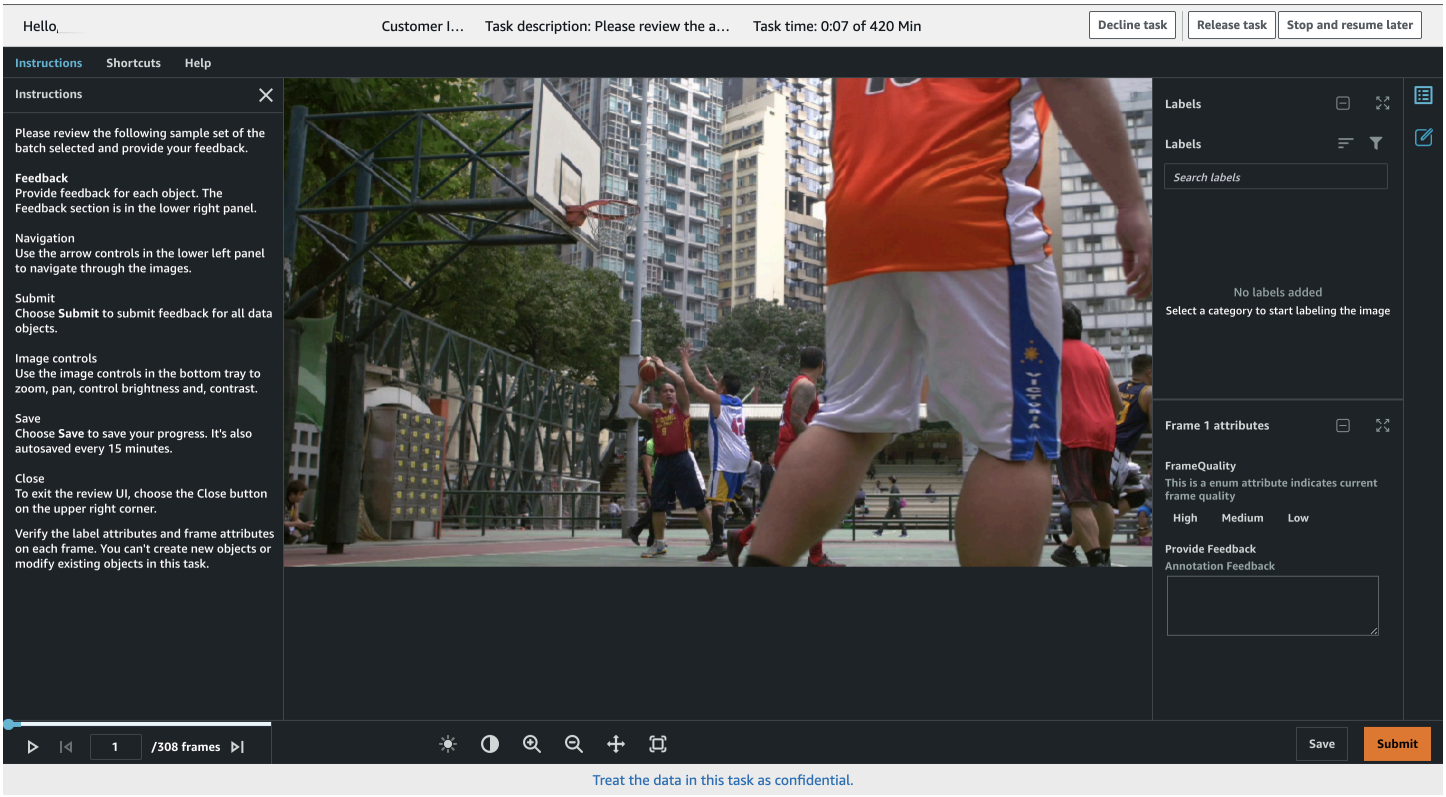
Objek yang tersisa: Jumlah benda yang tersisa untuk diberi label.

Objek gagal: Jumlah objek yang tidak dapat diberi label karena masalah dengan data input.

Objek untuk ditinjau: Jumlah objek yang siap untuk ditinjau.

Objek dengan umpan balik: Jumlah objek yang mendapat umpan balik dari anggota tim.

SageMaker Ground Truth Plus memungkinkan Anda meninjau kumpulan sampel data berlabel Anda (ditentukan selama panggilan konsultasi awal) melalui UI ulasan yang ditunjukkan pada gambar berikut.



Portal ini memungkinkan anggota tim proyek Anda dan Anda untuk meninjau satu set sampel kecil objek berlabel untuk setiap batch. Anda dapat memberikan umpan balik untuk setiap objek berlabel dalam subset tersebut melalui UI ini. UI ulasan memungkinkan Anda untuk menavigasi seluruh subset objek berlabel dan memberikan umpan balik untuk objek berlabel tersebut.

Anda dapat melakukan tindakan berikut menggunakan UI tinjauan.

- Gunakan kontrol panah di kiri bawah untuk menavigasi melalui objek data.
- Anda dapat memberikan umpan balik untuk setiap objek. Bagian Umpan Balik ada di panel kanan. Pilih Kirim untuk mengirimkan umpan balik untuk semua gambar.
- Gunakan kontrol gambar di baki bawah untuk memperbesar, menggeser, dan mengontrol kontras.
- Jika Anda berencana untuk kembali untuk menyelesaikan ulasan Anda, pilih Berhenti dan lanjutkan nanti di kanan atas.
- Pilih Simpan untuk menyimpan kemajuan Anda. Kemajuan Anda juga disimpan secara otomatis setiap 15 menit.
- Untuk keluar dari UI ulasan, pilih Tutup di sudut kanan atas UI ulasan.
- Anda dapat memverifikasi atribut Label dan atribut Frame pada setiap frame menggunakan panel di sebelah kanan. Anda tidak dapat membuat objek baru atau memodifikasi objek yang ada dalam tugas ini.

Terima atau Tolak Batch

Setelah Anda meninjau batch, Anda harus memilih untuk menerima atau menolaknya.

Jika Anda menerima batch, output dari pekerjaan pelabelan ini ditempatkan di bucket Amazon S3 yang Anda tentukan. Setelah data dikirim ke bucket S3 Anda, status batch Anda berubah dari Diterima ke Data yang dikirim.

Jika Anda menolak batch, Anda dapat memberikan umpan balik dan menjelaskan alasan Anda untuk menolak batch.

SageMaker Ground Truth Plus memungkinkan Anda untuk memberikan umpan balik pada tingkat objek data serta tingkat batch. Anda dapat memberikan umpan balik untuk objek data melalui UI tinjauan. Anda dapat menggunakan portal proyek untuk memberikan umpan balik untuk setiap batch. Ketika Anda menolak batch, seorangAWS ahli menghubungi Anda untuk menentukan proses pengerjaan ulang dan langkah selanjutnya untuk batch.

> **Note**
>
> Menerima atau menolak batch adalah tindakan satu kali dan tidak dapat dibatalkan. Hal ini diperlukan untuk menerima atau menolak setiap batch proyek.

Gunakan Amazon SageMaker Ground Truth Synthetic Data untuk Menghasilkan dan Label Data

Data sintetis Amazon SageMaker Ground Truth adalah layanan pembuatan dan pelabelan data turnkey yang membuatnya lebih cepat dan lebih hemat biaya bagi para ilmuwan machine learning (ML) untuk memperoleh gambar yang digunakan untuk melatih model computer vision (CV). Untuk melatih model CV, ilmuwan ML membutuhkan kumpulan data berlabel besar, berkualitas tinggi, dan berlabel. Dengan data sintetis Ground Truth, ilmuwan MLdapat menghasilkan dan memberi label ribuan gambar dalam beberapa hari. Data sintetis Ground Truth menggunakan model 3D yang dihasilkan komputer untuk menciptakan lingkungan virtual yang mewakili skenario dunia nyata, menghasilkan gambar sintetis yang diambil dari lingkungan ini, dan secara otomatis memberi anotasi setiap gambar dengan label. Anda dapat menggunakan gambar sintetis berlabel denganAWS layanan pelatihan model CV seperti Amazon SageMaker dan Amazon Lookout for Vision.

Mengapa menggunakan Data Sintetis Ground Truth?

Mengumpulkan dan memberi label data dalam lingkungan dinamis dengan variasi ukuran objek, bentuk, warna, posisi, latar belakang, dan pencahayaan seringkali merupakan proses yang memakan waktu dan mahal. Untuk melatih model secara efektif agar dapat beroperasi di lingkungan yang dinamis, para ilmuwan ML-nya harus mengumpulkan sejumlah besar gambar dunia nyata untuk merepresentasikan semua skenario yang memungkinkan, sebuah proses yang dapat memakan waktu berbulan-bulan. Untuk skenario yang tidak sering terjadi, seperti cacat produk langka dan penempatan produk yang salah, perlu waktu bertahun-tahun untuk menangkap gambar dalam jumlah yang cukup untuk melatih model CV. Untuk memperoleh gambar dengan cacat produk, ilmuwan ML dapat dengan sengaja merusak produk untuk mendapatkan gambar yang rusak. Data sintetis Ground Truth membuatnya lebih cepat dan lebih hemat biaya bagi ilmuwan ML. untuk dengan cepat memperoleh gambar berlabel yang mewakili skenario dunia nyata, persyaratan inti untuk melatih model CV. Ilmuwan MLdapat menggunakan data sintetis Ground Truth untuk menghasilkan ribuan gambar sintetis dari lingkungan virtual 3D yang mewakili skenario dunia nyata dalam beberapa jam, bukan bulan. Ground Truth menyediakan kesetiaan gambar sintetis dan laporan keragaman dan file manifes bersama dengan data sintetis berlabel. Laporan kesetiaan dan keragaman gambar sintetis menyediakan statistik dan plot yang membantu Anda lebih memahami gambar sintetis yang dihasilkan. File manifes berisi informasi tentang gambar dan label gambar yang dapat Anda gunakan untuk melatih dan menguji model.

> **Note**
>
> Data sintetis Ground Truth tidak mendukung data bersertifikat PHI, PCI, atau FedRAMP, dan Anda tidak boleh memberikan data ini ke data sintetis Ground Truth.

Data sintetis Ground Truth memiliki fungsionalitas berikut.

- Adegan 3D penuh dan beberapa kamera dalam sebuah adegan
- Peta kedalaman kebenaran tanah yang menyediakan data kedalaman 3D untuk semua gambar yang dihasilkan
- Urutan gambar (video) dari beberapa kamera yang disinkronkan
- Sabuk konveyor bergerak yang mendukung pemandangan dinamis

Bagaimana cara menggunakan Ground Truth

Jika Anda baru pertama kali menggunakan data GroundMemulai dengan Amazon SageMaker Ground Truth Synthetic Data

Memulai dengan Amazon SageMaker Ground Truth Synthetic Data

Panduan ini menunjukkan cara menyelesaikan langkah-langkah yang diperlukan untuk memenuhi prasyarat, memulai proyek data sintetis Ground Truth, dan meninjau label.

Untuk mulai menggunakan data sintetis, tinjau Mengatur Prasyarat Data Sintetis Amazon SageMaker Ground Truth danKomponen Inti dari Data Sintetis Amazon SageMaker Ground Truth.

Mengatur Prasyarat Data Sintetis Amazon SageMaker Ground Truth

Untuk menggunakan data sintetis Ground Truth, Anda memerlukan AWS akun. Jika sudah memiliki akun AWS, Anda dapat melewati langkah ini.

Mendaftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk secara online.

   Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

   Saat Anda mendaftar Akun AWS, Pengguna root akun AWS akan dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, tetapkan akses administratif ke pengguna administratif, dan hanya gunakan pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Membuat pengguna administratif

Setelah mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat sebuah pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke AWS Management Console sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Di halaman berikutnya, masukkan kata sandi Anda.

   Untuk bantuan masuk menggunakan pengguna root, lihat Masuk sebagai pengguna root dalam Panduan Pengguna AWS Sign-In.
2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuknya, silakan lihat Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda (konsol) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center.
2. Di Pusat Identitas IAM, berikan akses administratif ke sebuah pengguna administratif.

   Untuk mendapatkan tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, silakan lihat Mengonfigurasi akses pengguna dengan Direktori Pusat Identitas IAM default di Panduan Pengguna AWS IAM Identity Center.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal akses AWS dalam Panduan Pengguna AWS Sign-In.

Komponen Inti dari Data Sintetis Amazon SageMaker Ground Truth

Istilah-istilah berikut adalah kunci untuk memahami kemampuan data sintetis Ground Truth:

- Proyek: Setiap keterlibatan yang memenuhi syarat dengan seorang AWS ahli menghasilkan proyek data sintetis Ground Truth.

- Batch: Batch adalah kumpulan gambar berlabel serupa. Sebuah proyek dapat memiliki beberapa batch. Batch dapat dalam tahap pengujian atau produksi. Sebuah proyek dapat memiliki beberapa batch.
- Laporan Kesetiaan dan Keanekaragaman Gambar Sintetis: Data sintetis Ground Truth menyediakan laporan metrik yang membantu Anda membandingkan gambar sintetis yang dihasilkan dengan kumpulan data tipikal Anda.

Minta Proyek

Untuk memulai dengan data sintetis Amazon SageMaker Ground Truth, buka SageMaker konsol dan lengkapi formulir intake.

Setelah Anda mengirimkan formulir asupan di AWS konsol, seorang AWS ahli dari tim data sintetis Ground Truth menghubungi untuk mendiskusikan persyaratan dan harga proyek pelabelan data Anda.

Berbagi Data dari Bucket Amazon S3 Anda

SetelahAWSahli menjangkau untuk mendiskusikan proyek Anda, Anda mungkin diminta untuk mengisi formulir asupan dengan pertanyaan khusus untuk persyaratan data sintetis Anda. Formulir asupan, bersama dengan aset yang dibagikan dengan data sintetis Ground Truth, memungkinkan tim data sintetis Ground Truth untuk mengevaluasi proyek Anda dan perkiraan pekerjaan yang diperlukan untuk menyelesaikan proyek Anda.

Buat bucket Amazon S3 untuk membagikan aset proyek Anda dengan data sintetis Ground Truth dan simpan data keluaran proyek Anda.

Untuk membuat bucket Amazon S3 dan membagikannya kepada kami:

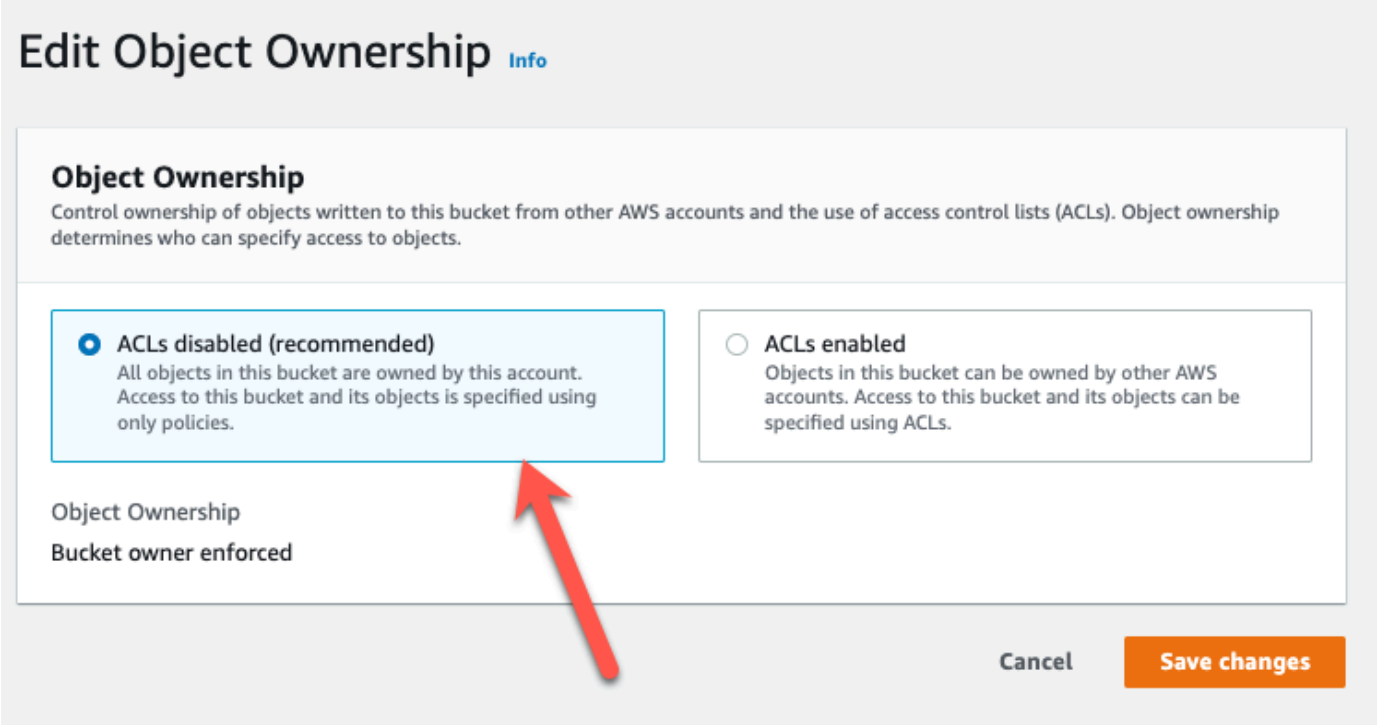
1. Ikuti petunjuknya diBuat Bucketdi dalamPanduan Pengguna Amazon Simple Storage Service Console.
2. Kami merekomendasikan penggunaan konvensi penamaan berikut sambil menyimpan data Anda di bucket Amazon S3.

   a. Klaster*nama bucket*harus berisi kurang dari 63 karakter.

   b. Klaster*nama bucket*dapat mencakup tanda hubung, tetapi tidak ada spasi dan garis bawah.
3. DiBucketlist, pilih nama bucket yang Anda buat.

4. Pilih Izin.
5. Di bagian Kebijakan bucket, pilih Edit.
6. Config ituACL dinonaktifkandipilih.

> **Note**
>
> Di bawahKepemilikan objekseharusnya ACL dinonaktifkan seperti yang ditunjukkan pada gambar di bawah ini.



7. Pilih Save changes (Simpan perubahan).

> **Note**
>
> Jika Anda memiliki persyaratan tambahan untuk mengakses data Anda di bucket Amazon S3, silakan hubungiAWSexpert.

Untuk membagikan aset proyek Anda dengan tim data sintetis Ground Truth untuk evaluasi proyek, estimasi pekerjaan, dan pembuatan data sintetis, ikuti langkah-langkah diKirim Data Proyek ke Data Sintetis Ground Truthbagian di bawah ini.

Setelah menerima formulir asupan dan aset proyek Anda, kami mengembalikan pernyataan pekerjaan (SOW) dalam 5 hari kerja. SOW menguraikan keterlibatan Anda dengan pembuatan dan pelabelan data sintetis Ground Truth. Setelah Anda menyetujui SOW, tim data sintetis Ground Truth menghasilkan batch uji yang terdiri dari 50 gambar sintetis. SesiAWSahli bertemu dengan Anda untuk meninjau batch pengujian, menyetujui atau menolak gambar, dan menyelesaikan produksi akhir. Garis waktu untuk ini didasarkan pada tanggapan dalam formulir asupan Anda.

Kirim Data Proyek ke Data Sintetis Ground Truth

SetelahAWSahli telah ditugaskan untuk proyek Anda, Anda dapat mengirim data proyek ke tim data sintetis Ground Truth untuk membantu dalam evaluasi proyek, estimasi pekerjaan, dan pembuatan data sintetis.

Untuk mengirim data proyek ke data sintetis Ground Truth:

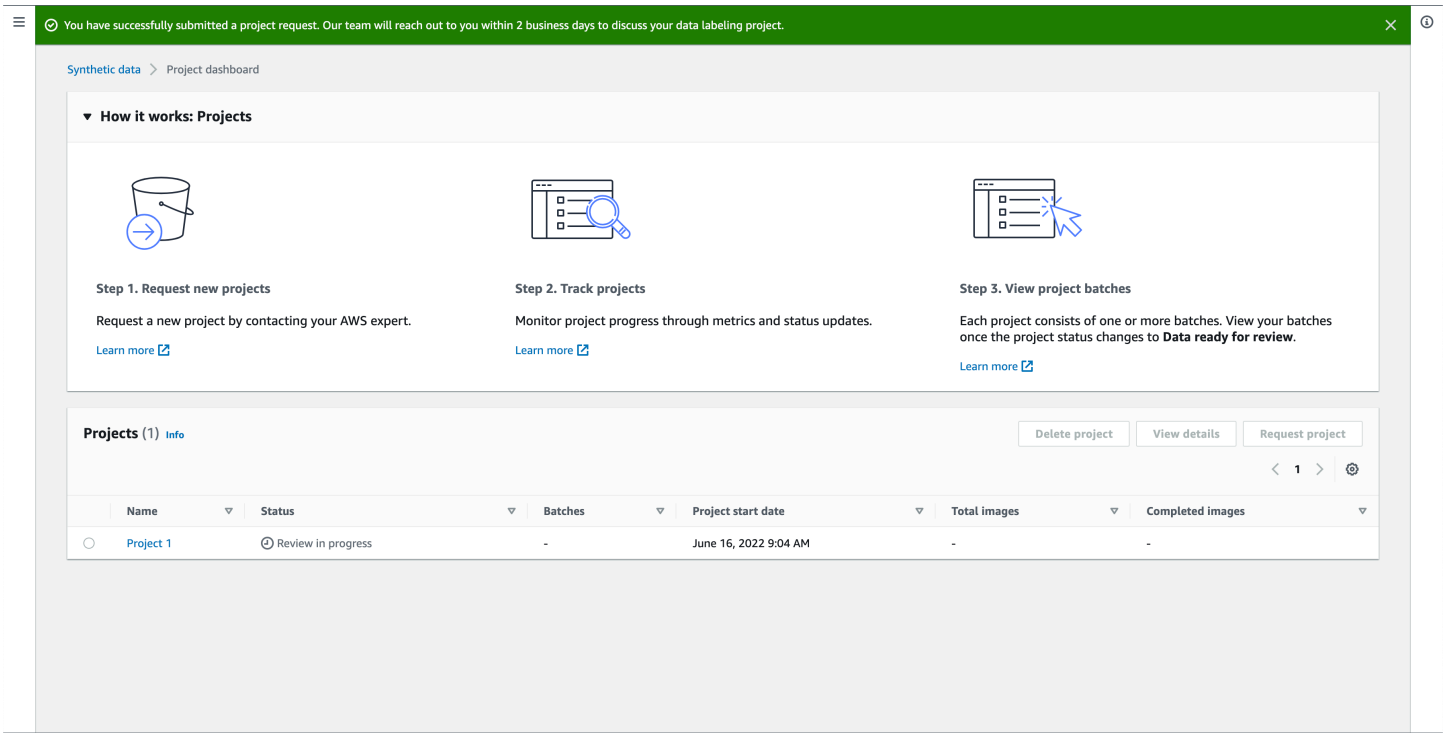
1. Di bawahTransfer data proyektabel di portal proyek, pilihMengirim data proyek.
2. Masukkan nama bucket S3 tempat Anda ingin mengirim data proyek sebagai lokasi sumber Amazon S3 dari transfer data proyek.
3. Pilih peran IAM untuk transfer data proyek. Jika Anda memilihOtomatis, Data sintetis Ground Truth menciptakan peran IAM di akun Anda dengan izin yang diperlukan untuk menjalankan transfer data proyek dan memanggil layanan lain atas nama Anda (disarankan). Jika Anda memilih peran IAM yang ada di akun Anda, data sintetis Ground Truth menggunakan peran IAM tersebut untuk menjalankan transfer data proyek dan memanggil layanan lain atas nama Anda.
4. PilihBuatuntuk membuat dan memulai transfer data proyek.

Setelah membuat transfer data proyek, Anda dapat melihat status transfer diTransfer data proyektabel di halaman rincian proyek di portal proyek. Ketika status transfer data proyekCompleted (Lengkap), data proyek tersedia untuk tim data sintetis Ground Truth.

Portal Proyek

Setiap project terdiri dari satu atau beberapa batch. Batch adalah kumpulan gambar yang dihasilkan dan diberi label serupa. Portal proyek memberi Anda akses ke proyek yang telah Anda kontrak dengan data sintetis Ground Truth. Anda dapat melihat status proyek Anda dan mengakses batch

yang telah selesai bersama dengan laporan kesetiaan dan keragaman gambar sintetis. Anda juga meninjau batch Anda untuk menerima atau menolaknya melalui portal proyek.



Anda dapat menggunakan portal proyek data sintetis Ground Truth untuk melacak detail berikut tentang proyek Anda:

Nama Proyek: Setiap proyek diidentifikasi menggunakan nama yang unik.

Status: Proyek data sintetis Ground Truth memiliki salah satu jenis status berikut:

1. Permintaan diserahkan: Anda telah berhasil mengirimkan formulir permintaan proyek. Selanjutnya, sebuahAWSahli jadwal panggilan dengan Anda untuk mendiskusikan rincian untuk proyek Anda.
2. Tinjau sedang berlangsung: Kami sedang meninjau proyek Anda. SesiAWSahli telah ditugaskan untuk proyek Anda.
3. Produksi sedang berlangsung: Saat ini kami sedang berupaya menghasilkan data berlabel untuk proyek Anda.
4. Data siap untuk ditinjau: Setidaknya satu batch siap untuk ditinjau.
5. Proyek Lengkap: Kami telah menyelesaikan pembuatan gambar berlabel yang diperlukan. Gambar disimpan di bucket Amazon S3.

Batch: Jumlah total batch dalam suatu proyek.

Tanggal mulai Proyek: Tanggal mulai dari sebuah proyek.

Total gambar: Jumlah gambar yang Anda minta.

Gambar Completed: Jumlah gambar berlabel yang dihasilkan di semua batch produksi yang diterima.

Menghapus Proyek

Anda dapat menghapus proyek menggunakan konsol jika status proyekPermintaan diserahkanatauProyek Lengkap. Untuk menghapus proyek dengan status lain, hubungiAWSAhli. Menghapus proyek data sintetis Ground Truth tidak menghapus data Anda dari bucket Amazon S3 dan dapat dikenakan biaya.

Anda dapat menghapus proyek berdasarkan statusnya sebagai berikut:

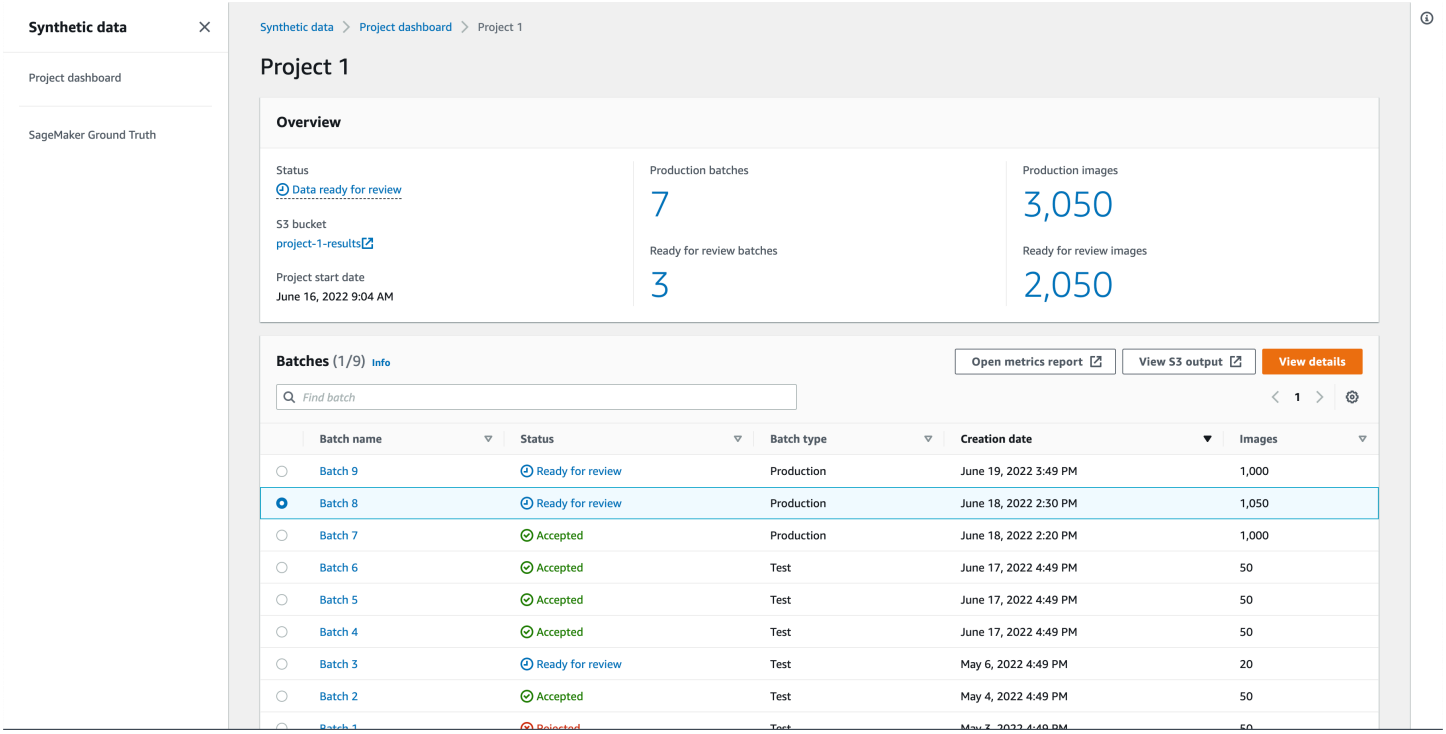
- Permintaan diserahkan: Menghapus proyek yang diminta menghapus semua proyek dan informasi pelanggan dari database data sintetis Ground Truth.
- Ulasan sedang berjalan/Produksi sedang berjalan/Data siap untuk ditinjau: Anda dapat memintaAWSahli untuk menghapus proyek yang memiliki salah satu status ini. Menghapus proyek menghapus semua proyek dan informasi pribadi dari database data sintetis Ground Truth dan ember S3.
- Proyek Completed: Setelah proyek ditandai sebagaiProyek Lengkap, kami menghapus semua informasi pelanggan dari database data sintetis Ground Truth dan ember S3. Anda dapat melihat proyek dan batch selama yang Anda inginkan, atau menghapusnya menggunakan konsol.

> **Note**
>
> Menghapus proyek tidak menghapus gambar dari bucket S3 Anda. Untuk mempelajari lebih lanjut tentang menghapus gambar dari bucket S3 Anda, lihatMenghapus objek Amazon S3.

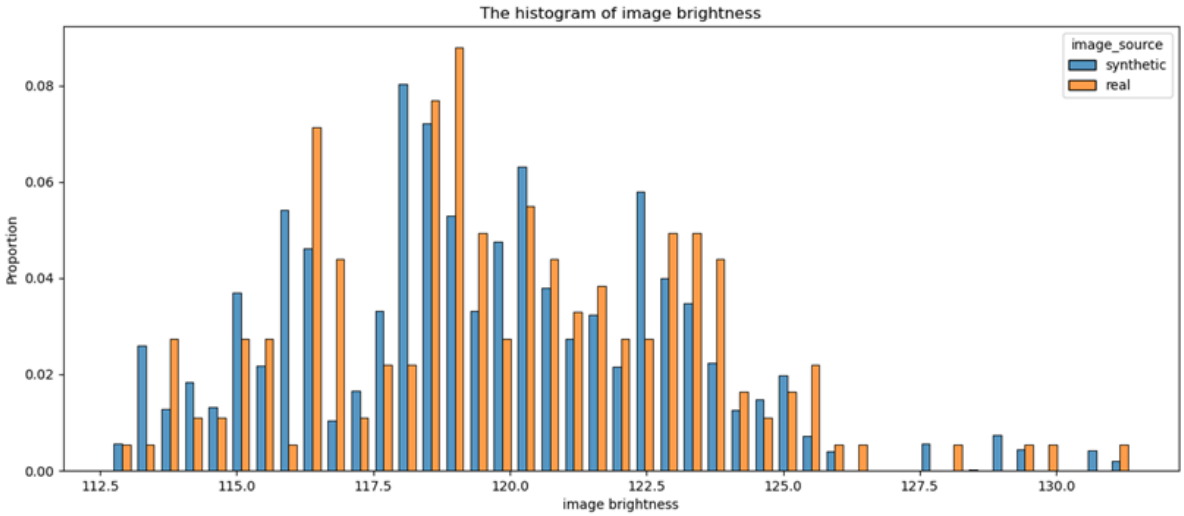
Batch tinjauan

Setiap Amazon SageMaker Proyek data sintetis Ground Truth terdiri dari satu atau beberapa batch. Setiap batch terdiri dari gambar sintetis berlabel. Batch terdiri dari dua jenis,Batch UjidanBatch Produksi. Batch pengujian menyediakan pratinjau kecil tentang bagaimana gambar sintetis terlihat menggunakan aset dan lingkungan 3D Anda. Gambar dalam batch pengujian tidak dihitung terhadap jumlah total gambar sintetis yang Anda kontrak. Setelah Anda menyetujui kumpulan pengujian untuk

konfigurasi gambar tertentu, data sintetis Ground Truth mulai menghasilkan gambar untuk batch produksi Anda. Gambar dalam batch produksi dihitung terhadap total gambar yang diperlukan.



Untuk setiap batch, data sintetis Ground Truth menyediakanLaporan Kesetiaan dan Keanekaragaman Gambar Sintetis. Laporan ini menyediakan statistik dan plot tingkat gambar dan objek yang membantu Anda memahami gambar sintetis yang dihasilkan. Statistik digunakan untuk menggambarkan keragaman dan kesetiaan gambar sintetis dan membandingkannya dengan gambar nyata. Contoh statistik dan plot yang disediakan adalah distribusi kelas objek, ukuran objek, kecerahan gambar, kontras gambar, serta plot yang mengevaluasi perbedaan antara gambar sintetis dan nyata. Data mentah untuk semua statistik set data yang dihitung juga disediakan sebagai file CSV untuk membantu Anda mempercepat debugging model dan mengaktifkan analisis lebih lanjut.



Anda dapat melihat semua batch untuk proyek Anda menggunakan portal proyek.

Anda dapat menggunakan portal proyek data sintetis Ground Truth untuk melacak detail berikut tentang setiap batch:

Nama Batch: Setiap batch diidentifikasi dengan nama batch yang unik.

Status: Kumpulan data sintetis Ground Truth memiliki salah satu jenis status berikut:

1. Sedang berlangsung: Kami saat ini menghasilkan gambar berlabel untuk batch ini. Ini akan segera siap untuk ulasan Anda.
2. Siap untuk ditinjau: Sekumpulan gambar sintetis berlabel sekarang siap untuk ditinjau. Ikuti langkah-langkah di[Mentransfer Data Batch ke bucket Amazon S3 Anda]bagian untuk melihat gambar dan meninjau batch.
3. Diterima: Anda telah menerima batch ini.
4. Ditolak: Anda telah menolak batch ini dan perlu dikerjakan ulang. Ketika Anda menolak batch,AWSahli menghubungi Anda untuk mendiskusikan hal ini lebih lanjut.

Tipe Batch: Batch dapat berupa batch uji atau batch produksi.

Tanggal pembuatan: Tanggal ketika batch dibuat.

Citra: Jumlah total citra dalam batch.

Mentransfer Data Batch ke bucket Amazon S3 Anda

Ketika status batchSiap untuk ditinjau, Anda harus mentransfer data batch ke bucket S3 Anda untuk melihat gambar dan meninjau batch.

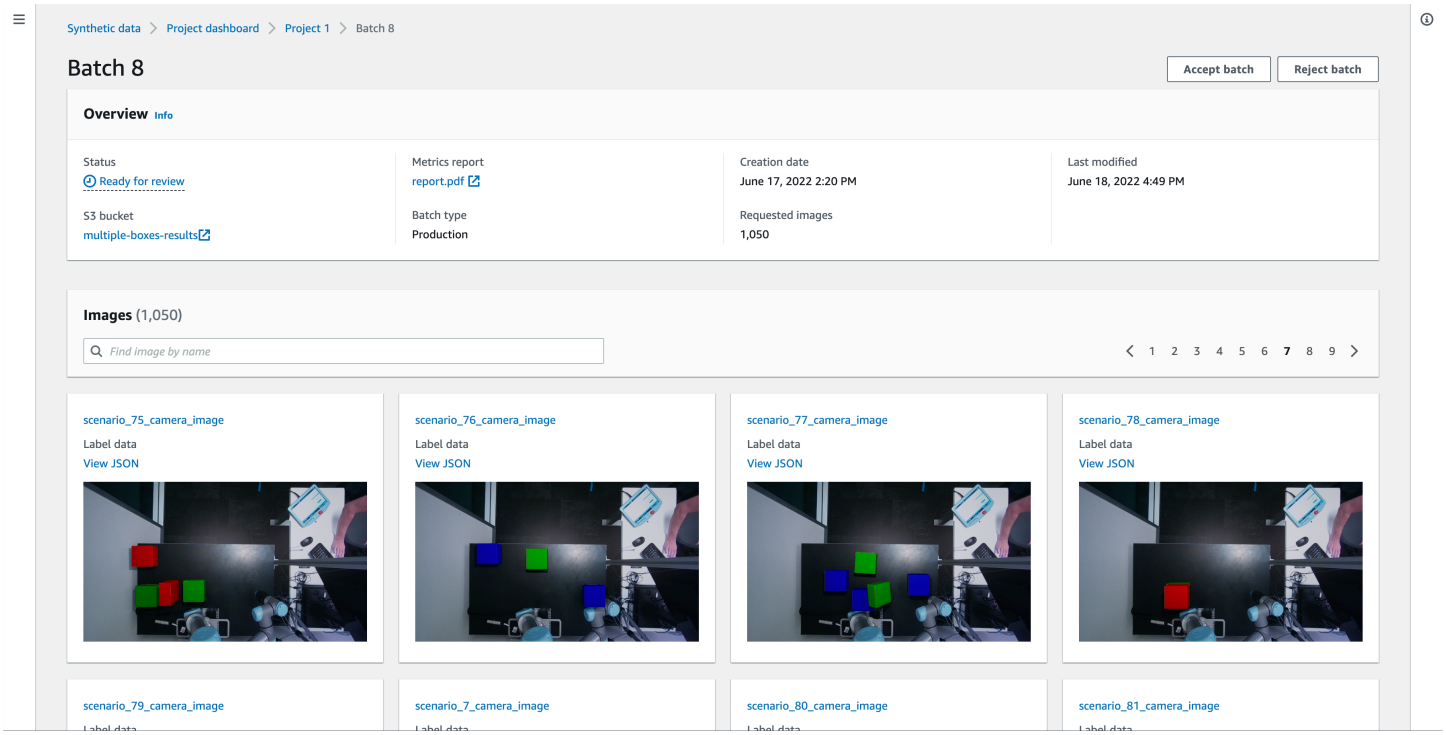
Untuk mentransfer data batch ke bucket S3 Anda:

1. Pada halaman rincian batch di portal proyek, pilihDapatkan data batch.
2. Di bawahLokasi tujuan tujuan S3Masukkan nama bucket S3 tempat Anda ingin menerima data batch.
3. Pilih peran IAM untuk transfer data proyek. Jika Anda memilihOtomatis, Data sintetis Ground Truth menciptakan peran IAM di akun Anda dengan izin yang diperlukan untuk menjalankan transfer data proyek dan memanggil layanan lain atas nama Anda (disarankan). Jika Anda memilih peran IAM yang ada di akun Anda, data sintetis Ground Truth menggunakan peran IAM tersebut untuk menjalankan transfer data proyek dan memanggil layanan lain atas nama Anda.
4. PilihBuatuntuk membuat dan memulai transfer data batch.

Setelah membuat transfer data batch, Anda dapat melihat status transfer diTransfer data Batchtabel pada halaman rincian batch di portal proyek. Ketika status transfer data batchCompleted (Lengkap), data batch tersedia di bucket S3 Anda, gambar batch dapat dilihat pada halaman detail batch di portal proyek, dan Anda dapat melanjutkan untuk meninjau batch.

Terima atau Tolak Batch

Setelah Anda meninjau batch, Anda dapat memilih untuk menerima atau menolaknya dari portal proyek seperti yang ditunjukkan di bawah ini.



Menerima batch menginformasikanAWSahli untuk melanjutkan atau menyelesaikan proyek, berdasarkan jumlah gambar yang tersisa yang Anda kontrak.

Ketika Anda menolak batch,AWSahli menghubungi Anda untuk menentukan proses pengerjaan ulang dan langkah selanjutnya untuk batch.

Menerima atau menolak batch adalah tindakan satu kali dan hanya dapat dibatalkan dengan menghubungi AndaAWSexpert Hal ini diperlukan untuk menerima atau menolak setiap batch proyek.

Membuat dan Mengelola Tenaga Kerja

SEBUAHtenaga kerjaadalah kelompok pekerja yang telah Anda pilih untuk memberi label pada kumpulan data Anda. Anda dapat memilih tenaga kerja Amazon Mechanical Turk, tenaga kerja yang dikelola vendor, atau Anda dapat membuat tenaga kerja pribadi Anda sendiri untuk memberi label atau meninjau kumpulan data Anda. Apapun jenis tenaga kerja yang Anda pilih, Amazon SageMaker mengurus pengiriman tugas ke pekerja.

Saat Anda menggunakan tenaga kerja pribadi, Anda juga membuattim kerja, sekelompok pekerja dari tenaga kerja Anda yang ditugaskan untuk spesifikpekerjaan—Amazon SageMaker Ground Truthtugas pelabelan atauAmazon Augmented AItugas review manusia. Anda dapat memiliki beberapa tim kerja dan dapat menetapkan satu atau lebih tim kerja untuk setiap pekerjaan.

Anda dapat menggunakan Amazon Cognito atau Penyedia Identitas (IdP) OpenID Connect (OIDC) pribadi Anda sendiri untuk mengelola tenaga kerja pribadi dan tim kerja Anda. Untuk informasi lebih lanjut tentang izin yang diperlukan untuk mengelola tenaga kerja Anda dengan cara ini, lihatIzin yang Diperlukan untuk Menggunakan Konsol Amazon SageMaker Ground Truth.

Topik

- Menggunakan Amazon Mechanical Turk Workforce
- Mengelola Tenaga Kerja Vendor
- Menggunakan Tenaga Kerja Pribadi

Menggunakan Amazon Mechanical Turk Workforce

Tenaga kerja Amazon Mechanical Turk (Mechanical Turk) menyediakan pekerja terbanyak untuk AndaAmazon SageMaker Ground Truthtugas pelabelan danAmazon Augmented AItugas review manusia. Tenaga kerja Amazon Mechanical Turk adalah sumber daya di seluruh dunia. Pekerja tersedia 24 jam sehari, 7 hari seminggu. Anda biasanya mendapatkan perputaran tercepat untuk tugas peninjauan manusia dan pekerjaan pelabelan saat Anda menggunakan tenaga kerja Amazon Mechanical Turk.

Setiap penagihan tenaga kerja Amazon Mechanical Turk ditangani sebagai bagian dari Ground Truth atau penagihan Amazon Augmented AI Anda. Anda tidak perlu membuat akun Mechanical Turk terpisah untuk menggunakan tenaga kerja Amazon Mechanical Turk.

> **Important**
>
> Anda tidak boleh membagikan informasi rahasia, informasi pribadi, atau informasi kesehatan yang dilindungi dengan tenaga kerja ini. Anda tidak boleh menggunakan tenaga kerja Amazon Mechanical Turk saat Anda menggunakan Amazon A2I bersamaan denganAWSLayanan yang memenuhi syarat HIPAA, seperti Amazon Textract dan Amazon Rekognition, untuk beban kerja yang berisi informasi kesehatan yang dilindungi.

Anda dapat memilih Mechanical Turk sebagai tenaga kerja Anda saat membuat pekerjaan pelabelan Ground Truth atau alur kerja tinjauan manusia Amazon A2I (definisi alur). Anda dapat membuat pekerjaan pelabelan dan alur kerja peninjauan manusia menggunakan SageMaker konsol dan API.

Saat Anda menggunakan operasi API untuk membuat pekerjaan pelabelan atau alur kerja peninjauan manusia, Anda menggunakan ARN berikut untuk tenaga kerja Amazon Mechanical Turk untuk

Anda`WorkteamArn`. Ganti*`region`*dengan AWS Wilayah yang Anda gunakan untuk membuat pekerjaan pelabelan atau loop manusia. Misalnya, jika Anda membuat tugas pelabelan di US West (Oregon), ganti*`region`*bersama`us-west-2`.

- `arn:aws:sagemaker:`*`region`*`:394669845002:workteam/public-crowd/default`

Ground Truth dan Amazon A2I memerlukan bahwa data input Anda bebas dari informasi pengenal pribadi (PII) saat Anda menggunakan Mechanical Turk. Jika Anda menggunakan tenaga kerja Mechanical Turk dan tidak menentukan bahwa data input Anda bebas dari PII, pekerjaan pelabelan Ground Truth dan tugas Augmented AI Anda akan gagal. Anda menentukan bahwa data input Anda bebas dari PII saat Anda membuat pekerjaan pelabelan Ground Truth dan saat Anda membuat loop manusia Amazon A2I menggunakan integrasi bawaan atau`StartHumanLoop`operasi operasi.

Gunakan bagian berikut untuk mempelajari cara menggunakan Mechanical Turk dengan layanan ini.

Topik

- Gunakan Mechanical Turk dengan Ground Truth
- Gunakan Mechanical Turk dengan Amazon A2I
- Kapan Mechanical Turk Tidak Didukung?

Gunakan Mechanical Turk dengan Ground Truth

Anda dapat menggunakan Mechanical Turk dengan Ground Truth saat Anda membuat pekerjaan pelabelan menggunakan konsol, atau`CreateLabelingJob`operasi operasi.

Saat Anda membuat pekerjaan pelabelan, kami sarankan Anda menyesuaikan jumlah pekerja yang membuat anotasi setiap objek data berdasarkan kompleksitas pekerjaan dan kualitas yang Anda butuhkan. Amazon SageMaker Ground Truth menggunakan konsolidasi anotasi untuk meningkatkan kualitas label. Lebih banyak pekerja dapat membuat perbedaan dalam kualitas label untuk pekerjaan pelabelan yang lebih kompleks, tetapi mungkin tidak membuat perbedaan untuk pekerjaan yang lebih sederhana. Untuk informasi selengkapnya, lihat Anotasi Terkonsolidasi. Perhatikan bahwa konsolidasi anotasi tidak didukung untuk alur kerja tinjauan manusia Amazon A2I.

Untuk menggunakan Mechanical Turk saat Anda membuat pekerjaan pelabelan (konsol):

1. Gunakan yang berikut ini untuk membuat pekerjaan pelabelan menggunakan area Ground Truth dari SageMaker konsol: Membuat Job Pelabelan (Konsol).
2. Saat Anda memilih Jenis pekerja di dalam Pekerja bagian, pilih Amazon Mechanical Turk.

3. Tentukan jumlah total pekerja waktu yang harus menyelesaikan tugas menggunakanKedaluwarsa tugas.
4. Tentukan jumlah total waktu tugas tetap tersedia untuk pekerjaKedaluwarsa tugas. Ini adalah berapa lama pekerja harus mengambil tugas sebelum gagal.
5. PilihHarga per tugasMenggunakan daftar dropdown. Ini adalah jumlah uang yang diterima pekerja untuk menyelesaikan satu tugas.
6. (Opsional) Jika berlaku, pilihDataset tidak berisi konten dewasa.SageMaker dapat membatasi pekerja Mechanical Turk yang dapat melihat tugas Anda jika berisi konten dewasa.
7. Anda harus membaca dan mengkonfirmasi pernyataan berikut dengan memilih kotak centang untuk menggunakan tenaga kerja Mechanical Turk. Jika data masukan Anda berisi informasi rahasia, informasi pribadi, atau informasi kesehatan yang dilindungi, Anda harus memilih tenaga kerja lain.

   Anda memahami dan menyetujui bahwa tenaga kerja Mechanical Turk terdiri dari kontraktor independen yang berlokasi di seluruh dunia dan bahwa Anda tidak boleh membagikan informasi rahasia, informasi pribadi, atau informasi kesehatan yang dilindungi dengan tenaga kerja ini.
8. (Opsional) Pilih kotak centang di sampingAktifkan pelabelan data otomatisjika Anda ingin mengaktifkan pelabelan data otomatis. Untuk mempelajari selengkapnya tentang fitur ini, lihatPelabelan Data.
9. Anda dapat menentukanJumlah pekerja per objek set datadi bawahKonfigurasi tambahan. Misalnya, jika Anda memasukkan 3 di bidang ini, setiap objek data akan diberi label oleh 3 pekerja.

Saat Anda membuat pekerjaan pelabelan dengan memilihBuat, tugas pelabelan Anda dikirim ke pekerja Mechanical Turk.

Untuk menggunakan Mechanical Turk saat Anda membuat pekerjaan pelabelan (API):

1. Gunakan berikut ini untuk membuat pekerjaan pelabelan menggunakan`CreateLabelingJob`Operasi operasi:Buat Job Pelabelan (API).
2. Gunakan berikut untuk`WorkteamArn`. Ganti*`region`*denganAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan.

   ```
   arn:aws:sagemaker:region:394669845002:workteam/public-crowd/default
   ```
3. Gunakan`TaskTimeLimitInSeconds`untuk menentukan jumlah total pekerja waktu harus menyelesaikan tugas.

4. GunakanTaskAvailabilityLifetimeInSecondsuntuk menentukan jumlah total waktu tugas tetap tersedia untuk pekerja. Ini adalah berapa lama pekerja harus mengambil tugas sebelum gagal.
5. GunakanNumberOfHumanWorkersPerDataObjectuntuk menentukan jumlah pekerja per objek dataset.
6. GunakanPublicWorkforceTaskPriceuntuk menetapkan harga per tugas. Ini adalah jumlah uang yang diterima pekerja untuk menyelesaikan satu tugas.
7. GunakanDataAttributesuntuk menentukan bahwa data masukan Anda bebas dari informasi rahasia, informasi pribadi, atau informasi kesehatan yang dilindungi.

   Ground Truthmemerlukanbahwa data input Anda bebas dari informasi identitas pribadi (PII) jika Anda menggunakan tenaga kerja Mechanical Turk. Jika Anda menggunakan Mechanical Turk dan tidak menentukan bahwa data input Anda bebas dari PII menggunakan`FreeOfPersonallyIdentifiableInformation`bendera, pekerjaan pelabelan Anda akan gagal.

   Gunakan`FreeOfAdultContent`tandai untuk menyatakan bahwa data masukan Anda bebas dari konten dewasa.SageMaker dapat membatasi pekerja Mechanical Turk yang dapat melihat tugas Anda jika berisi konten dewasa.

Anda dapat melihat contoh cara menggunakan API ini di notebook berikut, ditemukan di GitHub: Contoh Notebook Ground Truth Jupyter. Anda dapat mengakses notebook ini di bawah SageMaker Contoh Notebookdalam Instans notebook.

Gunakan Mechanical Turk dengan Amazon A2I

Anda dapat menentukan bahwa Anda ingin menggunakan Mechanical Turk dengan Amazon A2I saat Anda membuat alur kerja tinjauan manusia, juga disebut sebagaidefinisi alur, di konsol, atau dengan`CreateFlowDefinition`Operasi API API. Ketika Anda menggunakan alur kerja review manusia ini untuk mengkonfigurasi loop manusia, Anda harus menentukan bahwa data input Anda bebas dari PII.

Untuk menggunakan Mechanical Turk saat Anda membuat alur kerja tinjauan manusia (konsol):

1. Gunakan yang berikut untuk membuat alur kerja tinjauan manusia di bagian Augmented AI pada SageMaker konsol:Membuat Alur Kerja Tinjauan Manusia (Konsol).
2. Saat Anda memilihJenis pekerjadi dalamPekerjabagian, pilihAmazon Mechanical Turk.

3. PilihHarga per tugasMenggunakan daftar dropdown. Ini adalah jumlah uang yang diterima pekerja untuk menyelesaikan satu tugas.
4. (Opsional) Anda dapat menentukanJumlah pekerja per objek set datadi bawahKonfigurasi tambahan. Misalnya, jika Anda memasukkan 3 di bidang ini, setiap objek data akan diberi label oleh 3 pekerja.
5. (Opsional) Tentukan jumlah total pekerja waktu yang harus menyelesaikan tugas menggunakanKedaluwarsa tugas.
6. (Opsional) Tentukan jumlah total waktu tugas tetap tersedia untuk pekerja diKedaluwarsa tugas. Ini adalah berapa lama pekerja harus mengambil tugas sebelum gagal.
7. Setelah membuat alur kerja peninjauan manusia, Anda dapat menggunakannya untuk mengonfigurasi loop manusia dengan menyediakan Amazon Resource Name (ARN) dalam parameter`FlowDefinitionArn`. Anda mengonfigurasi loop manusia menggunakan salah satu operasi API dari jenis tugas bawaan, atau operasi API runtime Amazon A2I,`StartHumanLoop`. Untuk mempelajari selengkapnya, lihat Membuat dan Memulai Loop Manusia.

   Ketika Anda mengkonfigurasi loop manusia Anda, Anda harus menentukan bahwa data input Anda bebas dari informasi pribadi (PII) menggunakan`FreeOfPersonallyIdentifiableInformation`pengklasifikasi konten di`DataAttributes`. Jika Anda menggunakan Mechanical Turk dan tidak menentukan bahwa data input Anda bebas dari PII, tugas peninjauan manusia Anda akan gagal.

   Gunakan`FreeOfAdultContent`tandai untuk menyatakan bahwa data masukan Anda bebas dari konten dewasa.SageMaker dapat membatasi pekerja Mechanical Turk yang dapat melihat tugas Anda jika berisi konten dewasa.

Untuk menggunakan Mechanical Turk saat Anda membuat alur kerja tinjauan manusia (API):

1. Gunakan berikut ini untuk membuat alur kerja tinjauan manusia menggunakan`CreateFlowDefinition`Operasi operasi:Membuat Workflow Tinjauan Manusia (API).
2. Gunakan berikut untuk`WorkteamArn`. Ganti*`region`*denganAWSWilayah yang Anda gunakan untuk membuat pekerjaan pelabelan.

   ```
   arn:aws:sagemaker:region:394669845002:workteam/public-crowd/default
   ```

3. Gunakan`TaskTimeLimitInSeconds`untuk menentukan jumlah total pekerja waktu harus menyelesaikan tugas.

4. GunakanTaskAvailabilityLifetimeInSecondsuntuk menentukan jumlah total waktu tugas tetap tersedia untuk pekerja. Ini adalah berapa lama pekerja harus mengambil tugas sebelum gagal.
5. GunakanTaskCountuntuk menentukan jumlah pekerja per objek dataset. Misalnya, jika Anda menentukan 3 untuk parameter ini, setiap objek data akan diberi label oleh 3 pekerja.
6. GunakanPublicWorkforceTaskPriceuntuk menetapkan harga per tugas. Ini adalah jumlah uang yang diterima pekerja untuk menyelesaikan satu tugas.
7. Setelah membuat alur kerja peninjauan manusia, Anda dapat menggunakannya untuk mengonfigurasi loop manusia dengan menyediakan Amazon Resource Name (ARN) dalam parameterFlowDefinitionArn. Anda mengonfigurasi loop manusia menggunakan salah satu operasi API dari jenis tugas bawaan, atau operasi API runtime Amazon A2I,StartHumanLoop. Untuk mempelajari selengkapnya, lihat Membuat dan Memulai Loop Manusia.

   Ketika Anda mengkonfigurasi loop manusia Anda, Anda harus menentukan bahwa data input Anda bebas dari informasi pribadi (PII) menggunakanFreeOfPersonallyIdentifiableInformationpengklasifikasi konten diDataAttributes. Jika Anda menggunakan Mechanical Turk dan tidak menentukan bahwa data input Anda bebas dari PII, tugas peninjauan manusia Anda akan gagal.

   GunakanFreeOfAdultContenttandai untuk menyatakan bahwa data masukan Anda bebas dari konten dewasa.SageMaker dapat membatasi pekerja Mechanical Turk yang dapat melihat tugas Anda jika berisi konten dewasa.

Anda dapat melihat contoh cara menggunakan API ini di notebook berikut, ditemukan di GitHub: Contoh Notebooks Amazon A2I Jupyter.

Kapan Mechanical Turk Tidak Didukung?

Tenaga kerja ini tidak didukung di bawah skenario berikut. Dalam setiap skenario, Anda harus menggunakanprivatatauvendortenaga kerja.

- Tenaga kerja ini tidak didukung untuk pekerjaan pelabelan bingkai video Ground Truth dan pekerjaan pelabelan cloud titik 3D.
- Anda tidak dapat menggunakan tenaga kerja ini jika data input Anda berisi informasi pengenal pribadi (PII).
- Mechanical Turk tidak tersedia di beberapaAWSdaerah khusus. Jika berlaku, lihat dokumentasi untuk wilayah khusus Anda untuk informasi selengkapnya.

Mengelola Tenaga Kerja Vendor

Anda dapat menggunakan tenaga kerja yang dikelola vendor untuk memberi label pada data Anda menggunakan Amazon SageMaker Ground Truth (Ground Truth) dan Amazon Augmented AI). Vendor memiliki pengalaman luas dalam menyediakan layanan pelabelan data untuk tujuan pembelajaran mesin. Tenaga kerja vendor untuk kedua layanan ini harus dibuat dan dikelola secara terpisah melalui Amazon SageMaker konsol.

Vendor membuat layanan mereka tersedia melaluiAWSMarketplace. Anda dapat menemukan detail layanan vendor di halaman detail mereka, seperti jumlah pekerja dan jam kerja mereka. Anda dapat menggunakan rincian ini untuk membuat perkiraan berapa banyak pekerjaan pelabelan akan biaya dan jumlah waktu yang Anda dapat mengharapkan pekerjaan untuk mengambil. Setelah Anda memilih vendor, Anda berlangganan layanan mereka menggunakanAWSMarketplace.

Langganan adalah perjanjian antara Anda dan vendor. Perjanjian tersebut menjelaskan rincian perjanjian, seperti harga, jadwal, atau kebijakan pengembalian dana. Anda bekerja langsung dengan vendor jika ada masalah dengan pekerjaan pelabelan Anda.

Anda dapat berlangganan ke sejumlah vendor untuk memenuhi kebutuhan anotasi data Anda. Ketika Anda membuat pekerjaan pelabelan atau manusia review worklow Anda dapat menentukan bahwa pekerjaan akan dialihkan ke vendor tertentu.

> **Important**
>
> Sebelum mengirim data sensitif ke vendor, periksa praktik keamanan dan kepatuhan vendor di halaman detailnya dan tinjau perjanjian lisensi pengguna akhir (EULA) yang merupakan bagian dari perjanjian berlangganan Anda. Anda bertanggung jawab untuk memastikan bahwa vendor memenuhi persyaratan kepatuhan Anda untuk informasi pribadi atau rahasia. Jangan berbagi informasi kesehatan yang dilindungi dengan tenaga kerja ini.

Anda harus menggunakan konsol untuk berlangganan tenaga kerja vendor. Setelah berlangganan, Anda dapat menggunakan`ListSubscribedWorkteams`operasi untuk daftar vendor berlangganan Anda.

Untuk berlangganan tenaga kerja vendor

1. Buka SageMaker konsolhttps://console.aws.amazon.com/sagemaker/.
2. Pilih halaman yang sesuai dalam SageMaker konsol.

- Untuk pekerjaan pelabelan Ground Truth, pilihPelabelan tenaga kerja, pilihVendor, dan kemudian pilihTemukan layanan pelabelan data.
- Untuk alur kerja peninjauan manusia Amazon A2I, pilihTenaga kerja peninjauan manual, pilihVendor, dan kemudian pilihTemukan layanan peninjauan manusia.

3. Konsol membukaAWS Marketplacedengan:

   - kategori layanan pelabelan data dipilih untuk Ground Truth
   - kategori layanan peninjauan manusia dipilih untuk Amazon A2I

   Di sini Anda melihat daftar layanan vendor yang tersedia untuk layanan ini.
4. Pilih vendor. KlasterAWS Marketplacemenunjukkan informasi rinci tentang pelabelan data atau layanan peninjauan manusia. Gunakan informasi ini untuk menentukan apakah vendor memenuhi persyaratan Anda untuk tugas Anda.
5. Jika vendor memenuhi persyaratan Anda, pilihLanjutkan berlangganan.
6. Tinjau detail langganan. Jika Anda menyetujui persyaratan, pilihLangganannuntuk menyelesaikan langganan Anda ke layanan ini.

Menggunakan Tenaga Kerja Pribadi

SEBUAHtenaga kerja privatadalah sekelompok pekerja yangandamemilih. Ini bisa menjadi karyawan perusahaan Anda atau sekelompok ahli materi pelajaran dari industri Anda. Misalnya, jika tugasnya adalah memberi label gambar medis, Anda bisa menciptakan tenaga kerja pribadi orang-orang yang berpengetahuan luas tentang gambar yang dimaksud.

MASING-MASINGAWSakun memiliki akses ke satu tenaga kerja pribadi per wilayah, dan pemilik memiliki kemampuan untuk membuat beberapaprivat timdalam tenaga kerja itu. Satu tim kerja pribadi digunakan untuk menyelesaikan pekerjaan pelabelan atau tugas peninjauan manusia, ataupekerjaan. Anda dapat menetapkan setiap tim kerja ke pekerjaan terpisah atau menggunakan satu tim untuk beberapa pekerjaan. Seorang pekerja tunggal dapat berada di lebih dari satu tim kerja.

Tenaga kerja pribadi Anda dapat dibuat dan dikelola menggunakanAmazon Cognitoatau Penyedia Identitas OpenID Connect (OIDC) privat Anda.

Jika Anda adalah pengguna baruAmazon SageMaker Ground TruthatauAmazon Augmented AIdan tidak mengharuskan pekerja Anda dikelola dengan IdP Anda sendiri, disarankan agar Anda menggunakan Amazon Cognito untuk membuat dan mengelola tenaga kerja pribadi Anda.

Setelah membuat tenaga kerja, selain membuat dan mengelola tim kerja, Anda dapat melakukan hal berikut:

- Melacak performa worker
- Membuat topik Amazon SNSuntuk memberi tahu pekerja saat tugas pelabelan tersedia
- Mengelola Akses Tenaga Kerja Pribadi ke Tugas Menggunakan Alamat IP

> **Note**
>
> Tenaga kerja pribadi Anda dibagi antara Ground Truth dan Amazon A2I. Untuk membuat dan mengelola tim kerja pribadi yang digunakan oleh Augmented AI, gunakan bagian Ground Truth dari SageMaker konsol.

Topik

- Membuat dan Mengelola Amazon Cognito Workforce
- Membuat dan Mengelola Tenaga Kerja IdP OIDC
- Mengelola Tenaga Kerja Pribadi Menggunakan Amazon SageMaker API
- Melacak Kinerja Pekerja
- Membuat dan mengelola topik Amazon SNS untuk tim kerja Anda

Membuat dan Mengelola Amazon Cognito Workforce

Buat dan kelola tenaga kerja pribadi Anda menggunakan Amazon Cognito saat Anda ingin membuat tenaga kerja menggunakan Amazon SageMaker konsol atau Anda tidak ingin overhead mengelola kredensi pekerja dan otentikasi. Saat membuat tenaga kerja privat dengan Amazon Cognito, ia menyediakan autentikasi, otorisasi, dan pengelolaan pengguna untuk pekerja privat Anda.

Topik

- Membuat Tenaga Kerja Pribadi (Amazon Cognito)
- Mengelola Tenaga Kerja Pribadi (Amazon Cognito)

Membuat Tenaga Kerja Pribadi (Amazon Cognito)

Ketika Anda menggunakan Amazon Cognito, Anda dapat membuat tenaga kerja pribadi dengan salah satu cara berikut:

- Buat tenaga kerja baru saat Anda membuat pekerjaan pelabelan Anda. Untuk mempelajari caranya, lihat Membuat Tenaga Kerja Amazon Cognito Saat Membuat Job Pelabelan.
- Buat tenaga kerja baru sebelum Anda membuat pekerjaan pelabelan Anda. Untuk mempelajari caranya, lihat Membuat Amazon Cognito Workforce Menggunakan Halaman Labeling Workforce.
- Impor tenaga kerja yang ada setelah membuat pangkalan pengguna di konsol Amazon Cognito. Untuk mempelajari caranya, lihat Membuat Tenaga Kerja Pribadi (Amazon Cognito Console).

Setelah Anda membuat tenaga kerja pribadi, tenaga kerja dan semua tim kerja serta pekerja yang terkait dengannya tersedia untuk digunakan untuk semua tugas pekerjaan pelabelan Ground Truth dan tugas alur kerja tinjauan manusia Amazon Augmented AI.

Jika Anda baru ke Amazon SageMaker dan ingin menguji Ground Truth atau Amazon A2I, kami sarankan Anda membuat tim kerja pribadi yang terdiri dari orang-orang dari organisasi Anda menggunakan konsol. Gunakan tim kerja ini saat membuat alur kerja pelabelan atau tinjauan manusia (definisi alur) untuk menguji UI pekerja dan alur kerja Anda.

Topik

- Buat Tenaga Kerja Pribadi (Amazon SageMakerKonsol)
- Membuat Tenaga Kerja Pribadi (Amazon Cognito Console)

Buat Tenaga Kerja Pribadi (Amazon SageMakerKonsol)

Anda dapat membuat tenaga kerja pribadi di Amazon SageMaker konsol di salah satu dari dua cara berikut:

- Saat membuat pekerjaan pelabelan diLowongan kerja labelhalaman Amazon SageMaker Bagian Ground Truth Truth.
- MenggunakanPelabelan tenaga kerjahalaman Amazon SageMaker Bagian Ground Truth Truth. Jika Anda membuat tenaga kerja pribadi untuk alur kerja tinjauan manusia Amazon A2I, gunakan metode ini.

Kedua metode ini juga membuat tim kerja default yang berisi semua anggota tenaga kerja. Tenaga kerja pribadi ini tersedia untuk digunakan untuk pekerjaan Ground Truth dan Amazon Augmented AI.

Saat Anda membuat tenaga kerja pribadi menggunakan konsol, SageMaker menggunakan Amazon Cognito sebagai penyedia identitas untuk tenaga kerja Anda. Jika Anda ingin menggunakan OpenID Connect (OIDC) Identity Provider (IdP) Anda sendiri untuk membuat dan mengelola tenaga kerja pribadi Anda, Anda harus membuat tenaga kerja menggunakan SageMaker Operasi API`CreateWorkforce`. Untuk mempelajari selengkapnya, lihat Membuat Tenaga Kerja Pribadi (OIDC iDP).

Membuat Tenaga Kerja Amazon Cognito Saat Membuat Job Pelabelan

Jika Anda belum membuat tenaga kerja pribadi saat membuat pekerjaan pelabelan dan memilih untuk menggunakan pekerja swasta, Anda akan diminta untuk membuat tim kerja. Ini akan menciptakan tenaga kerja pribadi menggunakan Amazon Cognito.

Untuk membuat tenaga kerja sambil membuat pekerjaan pelabelan (konsol)

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilihLowongan kerja labeldan isi semua bidang yang diperlukan. Untuk petunjuk tentang cara memulai pekerjaan pelabelan, lihatMulai. Pilih Selanjutnya.
3. MemiilihPrivatuntuk jenis tenaga kerja.
4. DiPekerjabagian, masukkan:

    a. KlasterNama tim.

    b. Alamat email untuk hingga 100 anggota tenaga kerja. Alamat email peka huruf besar dan kecil. Pekerja Anda harus masuk menggunakan kasus yang sama dengan yang digunakan saat alamat awalnya dimasukkan. Anda dapat menambahkan anggota tenaga kerja tambahan setelah pekerjaan dibuat.

    c. Nama organisasi Anda. SageMaker menggunakan ini untuk menyesuaikan email yang dikirim ke pekerja.

    d. Alamat email kontak bagi pekerja untuk melaporkan masalah yang terkait dengan tugas.

Saat Anda membuat pekerjaan pelabelan, email dikirim ke setiap pekerja yang mengundang mereka untuk bergabung dengan tenaga kerja. Setelah membuat tenaga kerja, Anda dapat menambahkan, menghapus, dan menonaktifkan pekerja menggunakan SageMaker konsol atau konsol Amazon Cognito.

Membuat Amazon Cognito Workforce Menggunakan Halaman Labeling Workforce

Untuk membuat dan mengelola tenaga kerja pribadi Anda menggunakan Amazon Cognito, Anda dapat menggunakanPelabelan tenaga kerjahalaman. Saat mengikuti petunjuk di bawah ini, Anda memiliki opsi untuk membuat tenaga kerja pribadi dengan memasukkan email pekerja yang mengimpor tenaga kerja yang sudah ada sebelumnya dari kumpulan pengguna Amazon Cognito. Untuk mengimpor tenaga kerja, lihatMembuat Tenaga Kerja Pribadi (Amazon Cognito Console).

Untuk membuat tenaga kerja privat menggunakan email Pekerja

1. Buka Amazon SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilihPelabelan tenaga kerja.
3. MemiilihPrivat, lalu pilihBuat tim privat.
4. MemiilihUndang pekerja baru melalui email.
5. Tempel atau ketik daftar hingga 50 alamat email, dipisahkan dengan koma, ke dalam kotak alamat email.
6. Masukkan nama organisasi dan email kontak.
7. Secara opsional, pilih topik SNS yang akan berlangganan tim sehingga pekerja diberi tahu melalui email ketika pekerjaan pelabelan Ground Truth baru tersedia. Notifikasi Amazon SNS didukung oleh Ground Truth dan tidak didukung oleh Augmented AI. Jika Anda berlangganan pekerja untuk menerima pemberitahuan SNS, mereka hanya menerima pemberitahuan tentang pekerjaan pelabelan Ground Truth. Mereka tidak menerima pemberitahuan tentang tugas Augmented AI.
8. KlikBuat tim privattombol.

Setelah Anda mengimpor tenaga kerja pribadi Anda, refresh halaman. PadaRingkasan tenaga kerja pribadihalaman, Anda dapat melihat informasi tentang kumpulan pengguna Amazon Cognito untuk tenaga kerja Anda, daftar tim kerja untuk tenaga kerja Anda, dan daftar semua anggota tenaga kerja pribadi Anda.

> **Note**
>
> Jika Anda menghapus semua tim kerja pribadi Anda, Anda harus mengulangi proses ini untuk menggunakan tenaga kerja pribadi di wilayah tersebut.

Membuat Tenaga Kerja Pribadi (Amazon Cognito Console)

Amazon Cognito digunakan untuk menentukan dan mengelola tenaga kerja pribadi dan tim kerja Anda. Ini adalah layanan yang dapat Anda gunakan untuk membuat identitas bagi pekerja Anda dan mengotentikasi identitas ini dengan penyedia identitas. Tenaga kerja pribadi sesuai dengan satuKolam pengguna Amazon Cognito. Tim kerja pribadi sesuai denganGrup pengguna Amazon Cognitodalam kolam pengguna tersebut.

Contoh penyedia identitas yang didukung oleh Amazon Cognito:

- Penyedia login sosial seperti Facebook dan Google
- Penyedia OpenID Connect (OIDC)
- Security Assertion Markup Language (SAKL) penyedia seperti Active Directory
- Penyedia identitas bawaan Amazon Cognito

Untuk informasi selengkapnya, lihatApakah Amazon Cognito Itu?.

Untuk membuat tenaga kerja pribadi menggunakan Amazon Cognito, Anda harus memiliki kumpulan pengguna Amazon Cognito yang berisi setidaknya satu grup pengguna. LihatTutorial: Membuat Kolam Penggunauntuk mempelajari cara membuat kolam pengguna. LihatMenambahkan grup ke kolam penggunauntuk mempelajari cara menambahkan grup pengguna ke pool.

Setelah kumpulan pengguna Anda dibuat, ikuti langkah-langkah di bawah ini untuk membuat tenaga kerja pribadi dengan mengimpor kumpulan pengguna tersebut ke Amazon SageMaker.

Untuk membuat tenaga kerja pribadi dengan mengimpor kumpulan pengguna Amazon Cognito

1. Buka SageMaker konsolhttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilihPelabelan tenaga kerja.
3. PilihPrivat.
4. PilihMembuat tim pribadi. Ini menciptakan tenaga kerja pribadi dan tim kerja.
5. PilihImpor pekerja dari grup pengguna Amazon Cognito yang ada.
6. Pilih kumpulan pengguna yang telah Anda buat. Kumpulan pengguna memerlukan domain dan grup pengguna yang ada. Jika Anda mendapatkan kesalahan bahwa domain hilang, setel diNama domainpilihan padaIntegrasi aplikasihalaman konsol Amazon Cognito untuk grup Anda.
7. Pilih klien aplikasi. Kami merekomendasikan penggunaan klien yang dihasilkan oleh SageMaker.
8. Pilih grup pengguna dari pangkalan Anda untuk mengimpor anggotanya.

9. Pilih topik Amazon Simple Notification Service (Amazon SNS) tempat berlangganan tim sehingga pekerja diberi tahu melalui email ketika tugas pelabelan baru tersedia. Notifikasi Amazon SNS didukung oleh Ground Truth dan tidak didukung oleh Augmented AI. Jika Anda berlangganan pekerja untuk menerima pemberitahuan SNS, mereka hanya menerima pemberitahuan tentang pekerjaan pelabelan Ground Truth. Mereka tidak menerima pemberitahuan tentang tugas Augmented AI.
10. PilihMembuat tim pribadi.

> **Important**
>
> Setelah Anda membuat tenaga kerja menggunakan pangkalan pengguna Amazon Cognito, pangkalan pengguna Amazon Cognito tidak boleh dihapus tanpa terlebih dahulu menghapus semua tim kerja yang terkait dengan pangkalan tersebut di SageMaker konsol

Setelah Anda mengimpor tenaga kerja pribadi Anda, menyegarkan halaman untuk melihatRingkasan tenaga kerja pribadihalaman. Di halaman ini, Anda dapat melihat informasi tentang kumpulan pengguna Amazon Cognito untuk tenaga kerja Anda, daftar tim kerja untuk tenaga kerja Anda, dan daftar semua anggota tenaga kerja pribadi Anda. Tenaga kerja ini sekarang tersedia untuk digunakan di Amazon Augmented AI dan Amazon SageMaker Ground Truth untuk tugas peninjauan manusia dan pekerjaan pelabelan data masing-masing.

Mengelola Tenaga Kerja Pribadi (Amazon Cognito)

Setelah membuat tenaga kerja pribadi menggunakan Amazon Cognito, Anda dapat membuat dan mengelola tim kerja menggunakan Amazon SageMaker operasi konsol dan API.

Anda dapat melakukan hal berikut dengan menggunakanSageMakerkonsolatauKonsol Amazon Cognito.

- Menambah dan menghapus tim kerja.
- Tambahkan pekerja ke tenaga kerja Anda dan satu atau lebih tim kerja.
- Nonaktifkan atau hapus pekerja dari tenaga kerja Anda dan satu atau beberapa tim kerja. Jika Anda menambahkan pekerja ke tenaga kerja menggunakan konsol Amazon Cognito, Anda harus menggunakan konsol yang sama untuk menghapus pekerja dari tenaga kerja.

Anda dapat membatasi akses ke tugas untuk pekerja di alamat IP tertentu menggunakan SageMaker API Untuk informasi selengkapnya, lihat Mengelola Tenaga Kerja Pribadi Menggunakan Amazon SageMaker API.

Topik

- Mengelola Tenaga Kerja (Amazon SageMakerKonsol)
- Mengelola Tenaga Kerja Pribadi (Amazon Cognito Console)

Mengelola Tenaga Kerja (Amazon SageMakerKonsol)

Anda dapat menggunakan Amazon SageMaker konsol untuk membuat dan mengelola tim kerja dan pekerja individu yang membentuk tenaga kerja pribadi.

Gunakan tim kerja untuk menugaskan anggota tenaga kerja pribadi Anda ke pelabelan atau peninjauan manusiapekerjaan. Saat Anda membuat tenaga kerja menggunakan SageMaker konsol, ada tim kerja yang disebutEveryone-in-private-workforceyang memungkinkan Anda untuk menetapkan seluruh tenaga kerja Anda ke suatu pekerjaan. Karena kumpulan pengguna Amazon Cognito yang diimpor mungkin berisi anggota yang tidak ingin Anda sertakan dalam tim kerja Anda, tim kerja serupa tidak dibuat untuk kumpulan pengguna Amazon Cognito.

Anda memiliki dua pilihan untuk membuat tim kerja baru:

- Anda dapat membuat tim kerja di SageMaker konsol dan menambahkan anggota dari tenaga kerja Anda ke tim.
- Anda dapat membuat grup pengguna dengan menggunakan konsol Amazon Cognito dan kemudian membuat tim kerja dengan mengimpor grup pengguna. Anda dapat mengimpor lebih dari satu grup pengguna ke dalam setiap tim kerja. Anda mengelola anggota tim kerja dengan memperbarui grup pengguna di konsol Amazon Cognito. Lihat Mengelola Tenaga Kerja Pribadi (Amazon Cognito Console) untuk informasi selengkapnya.

Buat Tim Kerja Menggunakan SageMaker Konsol

Anda dapat membuat grup pengguna Amazon Cognito baru atau mengimpor grup pengguna yang ada menggunakan SageMakerkonsol, padaMelabel tenaga kerjahalaman. Untuk informasi selengkapnya tentang membuat grup pengguna di konsol Amazon Cognito, lihatMengelola Tenaga Kerja Pribadi (Amazon Cognito Console).

Untuk membuat tim kerja menggunakan SageMaker konsol

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. PilihMelabel tenaga kerjadari menu sebelah kiri.
3. Di bawahPrivat, PilihBuat tim pribadi.
4. Di bawahRincian tim, masukkanNama tim. Nama harus unik dalam akun Anda diAWSWilayah.
5. Di bawahMenambahkan pekerja, pilih metode untuk menambahkan pekerja ke tim menggunakan grup pengguna.

   - Jika Anda memilihBuat tim dengan menambahkan pekerja ke grup pengguna Amazon Cognito baru, pilih pekerja untuk ditambahkan ke tim.
   - Jika Anda memilihBuat tim dengan mengimpor grup pengguna Amazon Cognito yang ada, pilih grup pengguna yang merupakan bagian dari tim baru.
6. Jika Anda memilihTopik SNS, semua pekerja yang ditambahkan ke tim berlangganan topik Amazon SNS dan diberitahu ketika item pekerjaan baru tersedia untuk tim. Pilih dari daftar topik Amazon SNS terkait Ground Truth yang ada atau pilihBuat topik baruuntuk membuka dialog pembuatan topik.

   Notifikasi Amazon SNS didukung oleh Ground Truth dan tidak didukung oleh Augmented AI. Jika Anda berlangganan pekerja untuk menerima pemberitahuan SNS, mereka hanya menerima pemberitahuan tentang pekerjaan pelabelan Ground Truth. Mereka tidak menerima pemberitahuan tentang tugas Augmented AI.

Pekerja di tim kerja yang berlangganan topik menerima pemberitahuan ketika pekerjaan pelabelan Ground Truth baru untuk tim tersebut tersedia dan kapan seseorang akan kedaluwarsa.

BacaMembuat dan mengelola topik Amazon SNS untuk tim kerja Andauntuk informasi lebih lanjut tentang menggunakan topik Amazon SNS.

Langganan

Setelah Anda membuat tim kerja, Anda dapat melihat informasi lebih lanjut tentang tim dan mengubah atau mengatur topik Amazon SNS yang menjadi langganan anggotanya dengan mengunjungi konsol Amazon Cognito. Jika Anda menambahkan anggota tim sebelum berlangganan tim ke suatu topik, Anda perlu berlangganan secara manual anggota tersebut ke topik tersebut. BacaMembuat dan mengelola topik Amazon SNS untuk tim kerja Andauntuk informasi lebih lanjut tentang membuat dan mengelola topik Amazon SNS.

Menambahkan atau Menghapus Pekerja

SEBUAHTim kerjaadalah sekelompok pekerja dalam tenaga kerja Anda kepada siapa Anda dapat menetapkan pekerjaan. Seorang pekerja dapat ditambahkan ke lebih dari satu tim kerja. Setelah pekerja ditambahkan ke tim kerja, pekerja tersebut dapat dinonaktifkan atau dihapus.

Menambahkan Pekerja ke Tenaga Kerja

Menambahkan pekerja ke tenaga kerja memungkinkan Anda menambahkan pekerja itu ke tim kerja mana pun dalam tenaga kerja tersebut.

Untuk menambahkan pekerja menggunakan halaman ringkasan tenaga kerja pribadi

1. Buka Amazon SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. PilihMelabel tenaga kerjauntuk menavigasi ke halaman ringkasan tenaga kerja pribadi Anda.
3. PilihPrivat.
4. PilihMengundang pekerja baru.
5. Tempel atau ketik daftar alamat email, dipisahkan dengan koma, ke dalam kotak alamat email. Anda dapat memiliki hingga 50 alamat email dalam daftar ini.

Menambahkan Pekerja ke Tim Kerja

Seorang pekerja harus ditambahkan ke tenaga kerja sebelum ditambahkan ke tim kerja. Untuk menambahkan pekerja ke tim kerja, pertama-tama navigasikan keRingkasan tenaga kerja pribadihalaman menggunakan langkah-langkah di atas.

Untuk menambahkan pekerja ke tim kerja dari halaman ringkasan tenaga kerja pribadi

1. DiTim pribadibagian, pilih tim tempat Anda ingin menambahkan pekerja.
2. PilihPekerjaaTab.
3. PilihTambahkan pekerja ke timdan pilih kotak di samping pekerja yang ingin Anda tambahkan.
4. KlikTambahkan pekerja ke tim.

Nonaktifkan dan Hapus Pekerja dari Tenaga Kerja

Menonaktifkan pekerja menghentikan pekerja dari menerima pekerjaan. Tindakan ini tidak menghapus pekerja dari tenaga kerja, atau dari tim kerja mana pun yang terkait dengan pekerja.

Untuk menonaktifkan atau menghapus pekerja dari tim kerja, pertama-tama navigasikan ke halaman ringkasan tenaga kerja pribadi menggunakan langkah-langkah di atas.

Untuk menonaktifkan pekerja menggunakan halaman ringkasan tenaga kerja pribadi

1. DiPekerjaBagian, pilih pekerja yang ingin Anda nonaktifkan.
2. Pilih Disable (Nonaktifkan).

Jika diinginkan, Anda selanjutnya bisaAktifkanseorang pekerja setelah mereka dinonaktifkan.

Anda dapat menghapus pekerja dari tenaga kerja pribadi Anda langsung di SageMaker konsol jika pekerja yang ditambahkan di konsol ini. Jika Anda menambahkan pekerja (pengguna) di konsol Amazon Cognito, lihatMengelola Tenaga Kerja Pribadi (Amazon Cognito Console)untuk mempelajari cara menghapus pekerja di konsol Amazon Cognito.

Untuk menghapus pekerja menggunakan halaman ringkasan tenaga kerja pribadi

1. DiPekerjaBagian, pilih pekerja yang ingin Anda hapus.
2. Jika pekerja belum dinonaktifkan, pilihNonaktifkan.
3. Pilih pekerja dan pilihHapus.

Mengelola Tenaga Kerja Pribadi (Amazon Cognito Console)

Tenaga kerja pribadi sesuai dengan satuKolam pengguna Amazon Cognito. Tim kerja pribadi sesuai denganGrup pengguna Amazon Cognitodalam pengguna. Pekerja sesuai denganPengguna Amazon Cognitodalam kelompok-kelompok itu.

Setelah tenaga kerja Anda dibuat, Anda dapat menambahkan tim kerja dan pekerja individu melalui konsol Amazon Cognito. Anda juga dapat menghapus pekerja dari tenaga kerja pribadi Anda atau menghapusnya dari masing-masing tim di konsol Amazon Cognito.

> **Important**
>
> Anda tidak dapat menghapus tim kerja dari konsol Amazon Cognito. Menghapus grup pengguna Amazon Cognito yang dikaitkan dengan Amazon SageMaker tim kerja akan menghasilkan kesalahan. Untuk menghapus tim kerja, gunakan SageMakerkonsol.

Membuat Tim Kerja (Amazon Cognito Console)

Anda dapat membuat tim kerja baru untuk menyelesaikan pekerjaan dengan menambahkan grup pengguna Amazon Cognito ke kumpulan pengguna yang terkait dengan tenaga kerja pribadi Anda. Untuk menambahkan grup pengguna Amazon Cognito ke pangkalan pekerja yang ada, lihatMenambahkan grup ke Kolam pengguna.

Untuk membuat tim kerja menggunakan grup pengguna Amazon Cognito yang ada

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilihTenaga kerja.
3. UntukTim pribadi, pilihMembuat tim pribadi.
4. Di bawahRincian tim, beri nama tim. Nama harus unik dalam akun Anda dalamAWSWilayah.
5. UntukTambahkan pekerja,pilihImpor grup pengguna Amazon Cognito yang ada, dan pilih satu atau beberapa grup pengguna yang merupakan bagian dari tim baru.
6. Jika Anda memilihTopik SNS, semua pekerja yang ditambahkan ke tim berlangganan topik Amazon Simple Notification Service (Amazon SNS) dan diberi tahu ketika item pekerjaan baru tersedia untuk tim. Pilih dari daftar topik SNS Anda yang ada terkait SageMaker Ground Truth atau Amazon Augmented AI atau pilihMembuat topik baruuntuk membuat satu.

> **Note**
>
> Notifikasi Amazon SNS didukung oleh Ground Truth dan tidak didukung oleh Augmented AI. Jika Anda berlangganan pekerja untuk menerima pemberitahuan SNS, mereka hanya menerima pemberitahuan tentang pekerjaan pelabelan Ground Truth. Mereka tidak menerima pemberitahuan tentang tugas Augmented AI.

Langganan

Setelah Anda membuat tim kerja, Anda dapat melihat informasi lebih lanjut tentang tim dan mengubah atau mengatur topik SNS yang anggotanya berlangganan menggunakan konsol Amazon Cognito. Jika Anda menambahkan anggota tim sebelum berlangganan tim ke suatu topik, Anda perlu berlangganan secara manual anggota tersebut ke topik tersebut. Untuk informasi selengkapnya, lihat Membuat dan mengelola topik Amazon SNS untuk tim kerja Anda.

Tambah dan Hapus Pekerja (Amazon Cognito Console)

Saat menggunakan konsol Amazon Cognito untuk menambahkan pekerja ke tim kerja, Anda harus menambahkan pengguna ke pangkalan pengguna yang terkait dengan tenaga kerja sebelum menambahkan pengguna tersebut ke grup pengguna. Pengguna dapat ditambahkan ke kumpulan pengguna dengan berbagai cara. Untuk informasi selengkapnya, lihatMendaftar dan Mengonfirmasi Akun Pengguna.

Menambahkan Pekerja ke Tim Kerja

Setelah pengguna ditambahkan ke pool, pengguna dapat dikaitkan dengan grup pengguna di dalam pool tersebut. Setelah pengguna ditambahkan ke grup pengguna, pengguna tersebut menjadi pekerja di tim kerja apa pun yang dibuat menggunakan grup pengguna tersebut.

Untuk menambahkan pengguna ke grup pengguna

1. Buka konsol Amazon Cognito:https://console.aws.amazon.com/cognito/.
2. Pilih Kelola Kolam Pengguna.
3. Pilih pangkalan pengguna yang terkait dengan SageMaker tenaga kerja.
4. Di bawahPengaturan umum, pilihPengguna dan grupdan lakukan salah satu langkah berikut:
   - PilihGrupPilih grup yang ingin Anda tambahkan pengguna, dan pilihTambahkan pengguna. Pilih pengguna yang ingin Anda tambahkan dengan memilih ikon plus di sebelah kanan nama pengguna.
   - PilihPenggunaPilih pengguna yang ingin Anda tambahkan ke grup pengguna, dan pilihTambahkan ke grup. Dari menu pilihan menurun, pilih grup dan pilihTambahkan ke grup.

Menonaktifkan dan Menghapus Pekerja Dari Tim Kerja

Menonaktifkan pekerja menghentikan pekerja dari menerima pekerjaan. Tindakan ini tidak menghapus pekerja dari tenaga kerja, atau dari tim kerja mana pun yang terkait dengan pekerja. Untuk menghapus pengguna dari tim kerja di Amazon Cognito, Anda menghapus pengguna dari grup pengguna yang terkait dengan tim tersebut.

Untuk menonaktifkan pekerja (konsol Amazon Cognito)

1. Buka konsol Amazon Cognito:https://console.aws.amazon.com/cognito/.
2. Pilih Kelola Kolam Pengguna.
3. Pilih pangkalan pengguna yang terkait dengan SageMaker tenaga kerja.

4. Di bawahPengaturan umum, pilihPengguna dan grup.
5. Pilih pengguna yang ingin Anda nonaktifkan.
6. PilihMenonaktifkan pengguna.

Anda dapat mengaktifkan pengguna yang dinonaktifkan dengan memilihAktifkan Pengguna.

Untuk menghapus pengguna dari grup pengguna (konsol Amazon Cognito)

1. Buka konsol Amazon Cognito:https://console.aws.amazon.com/cognito/.
2. Pilih Kelola Kolam Pengguna.
3. Pilih pangkalan pengguna yang terkait dengan SageMaker tenaga kerja.
4. Di bawahPengaturan umum, pilihPengguna dan grup.
5. UntukPenggunatab, pilihXikon di sebelah kanan grup yang ingin Anda hapus pengguna.

Membuat dan Mengelola Tenaga Kerja IdP OIDC

Buat tenaga kerja pribadi menggunakan OpenID Connect (OIDC) Identity Provider (iDP) ketika Anda ingin mengelola dan mengautentikasi pekerja Anda menggunakan IdP OIDC Anda sendiri. Kredensi pekerja individu dan data lainnya akan dijaga kerahasiaannya. Ground Truth dan Amazon A2I hanya akan memiliki visibilitas ke informasi pekerja yang Anda berikan melalui klaim yang Anda kirim ke layanan ini. Untuk membuat tenaga kerja menggunakan IdP OIDC, IdP Anda harus mendukungkelompokkarena Ground Truth dan Amazon A2I memetakan satu atau lebih grup di IdP Anda ke tim kerja. Untuk mempelajari selengkapnya, lihat Kirim Klaim yang Diperlukan dan Opsional ke Ground Truth dan Amazon A2I.

Jika Anda adalah pengguna baru Ground Truth atau Amazon A2I, Anda dapat menguji UI pekerja dan alur kerja Anda dengan membuat tim kerja pribadi dan menambahkan diri Anda sebagai pekerja. Gunakan tim kerja ini saat Anda membuat pekerjaan pelabelan atau alur kerja peninjauan manusia. Pertama, buat tenaga kerja IdP OIDC pribadi menggunakan instruksi diMembuat Tenaga Kerja Pribadi (OIDC iDP). Selanjutnya, lihatMengelola Tenaga Kerja Pribadi (OIDC iDP)untuk mempelajari cara membuat tim kerja.

Topik

- Membuat Tenaga Kerja Pribadi (OIDC iDP)
- Mengelola Tenaga Kerja Pribadi (OIDC iDP)

Membuat Tenaga Kerja Pribadi (OIDC iDP)

Buat tenaga kerja pribadi menggunakan OpenID Connect (OIDC) Identity Provider (IdP) ketika Anda ingin mengautentikasi dan mengelola pekerja menggunakan penyedia identitas Anda sendiri. Gunakan halaman ini untuk mempelajari cara mengonfigurasi IdP Anda untuk berkomunikasi dengan Amazon SageMaker Ground Truth (Ground Truth) atau Amazon Augmented AI (Amazon A2I) dan untuk mempelajari cara membuat tenaga kerja menggunakan IdP Anda sendiri.

Untuk membuat tenaga kerja menggunakan IdP OIDC, IdP Anda harus mendukungkelompokkarena Ground Truth dan Amazon A2I menggunakan satu atau beberapa grup yang Anda tentukan untuk membuat tim kerja. Anda menggunakan tim kerja untuk menentukan pekerja untuk pekerjaan pelabelan Anda dan tugas peninjauan manusia. Karena kelompok bukanKlaim standar standar, IdP Anda mungkin memiliki konvensi penamaan yang berbeda untuk sekelompok pengguna (pekerja). Oleh karena itu, Anda harus mengidentifikasi satu atau beberapa grup pengguna tempat pekerja berada menggunakan klaim khusussagemaker:groupsyang dikirim ke Ground Truth atau Amazon A2I dari IdP Anda. Untuk mempelajari selengkapnya, lihat Kirim Klaim yang Diperlukan dan Opsional ke Ground Truth dan Amazon A2I.

Anda membuat tenaga kerja IdP OIDC menggunakan SageMaker Operasi APICreateWorkforce. Setelah Anda membuat tenaga kerja pribadi, tenaga kerja dan semua tim kerja serta pekerja yang terkait dengannya tersedia untuk digunakan untuk semua tugas pekerjaan pelabelan Ground Truth dan tugas alur kerja tinjauan manusia Amazon A2I. Untuk mempelajari selengkapnya, lihat Membuat Tenaga kerja OIDC IdP OIDC.

Kirim Klaim yang Diperlukan dan Opsional ke Ground Truth dan Amazon A2I

Bila Anda menggunakan IdP Anda sendiri, Ground Truth dan Amazon A2I gunakanIssuer,ClientId, danClientSecretuntuk mengautentikasi pekerja dengan mendapatkan KODE otentikasi dariAuthorizationEndpoint.

Ground Truth dan Amazon A2I akan menggunakan KODE ini untuk mendapatkan klaim khusus dari IDP AndaTokenEndpointatauUserInfoEndpoint. Anda dapat mengkonfigurasiTokenEndpointuntuk mengembalikan token web JSON (JWT) atauUserInfoEndpointuntuk mengembalikan sebuah objek JSON. Objek JWT atau JSON harus berisi klaim wajib dan opsional yang Anda tentukan. SEBUAHklaimadalah pasangan kunci-nilai yang berisi informasi tentang pekerja atau metadata tentang layanan OIDC. Tabel berikut mencantumkan klaim yang harus disertakan, dan yang secara opsional dapat dimasukkan dalam objek JWT atau JSON yang dikembalikan IdP Anda.

> **Note**
>
> Beberapa parameter dalam tabel berikut dapat ditentukan dengan menggunakan `:` atau `-`. Misalnya, Anda dapat menentukan grup yang dimiliki pekerja `sagemaker:groups` atau `sagemaker-groups` dalam klaim Anda.

Nama	Diperluka n	Format dan Nilai yang Diterima	Deskripsi	Contoh
`sagemaker :groups` atau `sagemaker- groups`	Ya	Jenis data: Jika pekerja termasuk dalam satu kelompok, identifikasi grup menggunakan string. Jika pekerja termasuk dalam beberapa kelompok, gunakan daftar yang berisi hingga 10 string. Karakter yang diizinkan yang diizinkan: Regex: [}}}}}}}} }}}}}}}}}}}}}}}}}}}}}}}} }}}}}}}}, Kuota: 10 kelompok per pekerja	Menetapkan pekerja untuk satu atau lebih kelompok. Grup digunakan untuk memetakan pekerja ke dalam tim kerja.	Contoh pekerja yang termasuk dalam satu kelompok: `"work_team1"` Contoh pekerja yang termasuk dalam lebih dari satu kelompok: `["work_team1", "work_team2"]`

Nama	Diperluka n	Format dan Nilai yang Diterima	Deskripsi	Contoh
		63 karakter per nama grup		
`sagemaker :sub` atau `sagemaker- sub`	Ya	Jenis data: String	Ini adalah wajib untuk melacak identitas pekerja di dalam platform Ground Truth untuk audit dan untuk mengidentifikasi tugas-tugas yang dikerjakan oleh pekerja itu. Untuk ADFS: Pelanggan harus menggunakan Primary Security Identifier (SID).	`"11101110 1-1234567 89-368705 6437-1111"`
`sagemaker :client_id` atau `sagemaker -client_id`	Ya	Jenis data: String Karakter yang diizinkan yang diizinkan: Regex: [\ w+-] + Tanda kutip: 128 karakter	ID klien. Semua token harus dikeluark an untuk ID klien ini.	`"00b600bb -1f00-05d 0-bd00-00 be00fbd0e0"`

Nama	Diperluka n	Format dan Nilai yang Diterima	Deskripsi	Contoh
`sagemaker :name` atau `sagemaker- name`	Ya	Jenis data: String	Nama pekerja yang akan ditampilkan di portal pekerja.	`"Jane Doe"`
`email`	Tidak	Jenis data: String	Email pekerja. Ground Truth menggunakan email ini untuk memberi tahu pekerja bahwa mereka telah diundang untuk mengerjakan tugas pelabelan. Ground Truth juga akan menggunakan email ini untuk memberi tahu pekerja Anda saat tugas pelabelan tersedia jika Anda menyiapkan topik Amazon SNS untuk tim kerja tempat pekerja ini aktif.	`"example- email@dom ain.com"`
`email_ver ified`	Tidak	Jenis data: Bool Nilai yang Diterima: `True`, `False`	Menunjukkan apakah email pengguna telah diverifikasi atau tidak.	`True`

Berikut ini adalah contoh sintaks objek JSON Anda`UserInfoEndpoint`dapat kembali.

```
{
    "sub":"122",
    "exp":"10000",
    "sagemaker-groups":["group1","group2"]
    "sagemaker-name":"name",
    "sagemaker-sub":"122",
    "sagemaker-client_id":"123456"
}
```

Ground Truth atau Amazon A2I membandingkan grup yang terdaftar `disagemaker:groups`atau`sagemaker-groups`untuk memverifikasi bahwa pekerja Anda milik tim kerja yang ditentukan dalam pekerjaan pelabelan atau tugas peninjauan manusia. Setelah tim kerja diverifikasi, tugas pelabelan atau peninjauan manusia dikirim ke pekerja itu.

Membuat Tenaga kerja OIDC IdP OIDC

Anda dapat membuat tenaga kerja menggunakan SageMaker Operasi API`CreateWorkforce`dan SDK khusus bahasa terkait. Menentukan`WorkforceName`dan informasi tentang IDP OIDC Anda dalam parameter`OidcConfig`. Dianjurkan agar Anda mengonfigurasi OIDC dengan URI pengalihan place-holder, lalu memperbarui URI dengan URL portal pekerja setelah Anda membuat tenaga kerja. Untuk mempelajari selengkapnya, lihat Konfigurasikan IdP OIDC Anda.

Berikut ini adalah contoh permintaan. Lihat`CreateWorkforce`untuk mempelajari lebih lanjut tentang setiap parameter dalam permintaan ini.

```
CreateWorkforceRequest: {
    #required fields
    WorkforceName: "example-oidc-workforce",
    OidcConfig: {
        ClientId: "clientId",
        ClientSecret: "secret",
        Issuer: "https://example-oidc-idp.com/adfs",
        AuthorizationEndpoint: "https://example-oidc-idp.com/adfs/oauth2/authorize",
        TokenEndpoint: "https://example-oidc-idp.com/adfs/oauth2/token",
        UserInfoEndpoint: "https://example-oidc-idp.com/adfs/oauth2/userInfo",
        LogoutEndpoint: "https://example-oidc-idp.com/adfs/oauth2/log-out",
        JwksUri: "https://example-oidc-idp.com/adfs/discovery/keys"
    },
    SourceIpConfig: {
        Cidrs: ["string", "string"]
    }
```

```
}
```

Konfigurasikan IdP OIDC Anda

Bagaimana Anda mengkonfigurasi IdP OIDC Anda tergantung pada IdP yang Anda gunakan, dan persyaratan bisnis Anda.

Ketika Anda mengkonfigurasi IdP Anda, Anda harus menentukan callback atau redirect URI. Setelah Ground Truth atau Amazon A2I mengautentikasi pekerja, URI ini akan mengarahkan pekerja ke portal pekerja tempat pekerja dapat mengakses tugas pelabelan atau peninjauan manusia. Untuk membuat URL portal pekerja, Anda perlu membuat tenaga kerja dengan detail IdP OIDC Anda menggunakan`CreateWorkforce`Operasi API API API API API API Secara khusus, Anda harus mengonfigurasi IdP OIDC Anda dengan klaim sagemaker khusus yang diperlukan (lihat bagian selanjutnya untuk lebih jelasnya). Oleh karena itu, disarankan agar Anda mengonfigurasi OIDC Anda dengan URI pengalihan place-holder, dan kemudian memperbarui URI setelah Anda membuat tenaga kerja. LihatMembuat Tenaga kerja OIDC IdP OIDCuntuk mempelajari cara membuat tenaga kerja menggunakan API ini.

Anda dapat melihat URL portal pekerja Anda di SageMaker Konsol Ground Truth, atau menggunakan SageMaker Operasi API API API API API API`DescribeWorkforce`. URL portal pekerja ada di`SubDomain`parameter dalam respon.

> **Important**
>
> Pastikan Anda menambahkan subdomain tenaga kerja ke daftar izinkan IdP OIDC Anda. Ketika Anda menambahkan subdomain ke daftar izinkan Anda, itu harus diakhiri dengan`/oauth2/idpresponse`.

Untuk melihat URL portal pekerja Anda setelah membuat tenaga kerja pribadi (Konsol):

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilihPelabelan tenaga kerja pelikan.
3. PilihPrivatTab.
4. MasukRingkasan tenaga kerja pribadi pribadikamu akan melihatURL masuk portal pelabelan label. Ini adalah URL portal pekerja Anda.

Untuk melihat URL portal pekerja Anda setelah membuat tenaga kerja pribadi (API):

Saat Anda membuat tenaga kerja pribadi menggunakan`CreateWorkforce`, Anda menentukan`WorkforceName`. Gunakan nama ini untuk menelepon`DescribeWorkforce`. Tabel berikut mencakup contoh permintaan yang menggunakanAWS CLIdanAWS SDK for Python (Boto3).

SDK for Python (Boto3)

```
response = client.describe_workforce(WorkforceName='string')
print(f'The workforce subdomain is: {response['SubDomain']}')
```

AWS CLI

```
$ C:\>  describe-workforce --workforce-name 'string'
```

Validasi Respons Otentikasi Tenaga Kerja IdP OIDC Anda

Setelah Anda membuat tenaga kerja OIDC IdP OIDC Anda, Anda dapat menggunakan prosedur berikut untuk memvalidasi alur kerja autentikasi menggunakan Curl. Prosedur ini mengasumsikan Anda memiliki akses ke terminal, dan bahwa Anda telah menginstal cURL.

Untuk memvalidasi respons otorisasi IdP OIDC Anda:

1. Dapatkan kode otorisasi menggunakan URI yang dikonfigurasi sebagai berikut:

   ```
   {AUTHORIZE ENDPOINT}?client_id={CLIENT ID}&redirect_uri={REDIRECT
    URI}&scope={SCOPE}&response_type=code
   ```

   a. Ganti`{AUTHORIZE ENDPOINT}`dengan endpoint otorisasi untuk IDP OIDC Anda.

   b. Ganti`{CLIENT ID}`dengan ID Klien dari klien OAuth Anda.

   c. Ganti`{REDIRECT URI}`dengan URL portal pekerja. Jika belum ada, Anda harus menambahkan`/oauth2/idpresponse`ke akhir URL.

   d. Jika Anda memiliki cakupan khusus, gunakan untuk mengganti`{SCOPE}`. Jika Anda tidak memiliki ruang lingkup khusus, ganti`{SCOPE}`bersama`openid`.

   Berikut ini adalah contoh URI setelah modifikasi di atas dibuat:

   ```
   https://example.com/authorize?
   client_id=f490a907-9bf1-4471-97aa-6bfd159f81ac&redirect_uri=https%3A%2F%2F
   ```

```
%2Fexample.labeling.sagemaker.aws
%2Foauth2%2Fidpresponse&response_type=code&scope=openid
```

2. Salin dan tempel URI yang dimodifikasi dari langkah 1 ke browser Anda dan tekan Enter pada keyboard Anda.
3. Otentikasi menggunakan IdP Anda.
4. Salin parameter kueri kode otentikasi di URI. Parameter ini makhluk dengan`code=`. Berikut ini adalah contoh seperti apa respons. Dalam contoh ini, salin`code=MCNYDB...`dan segala sesudahnya

```
https://example.labeling.sagemaker.aws/oauth2/idpresponse?code=MCNYDB....
```

5. Buka terminal dan masukkan perintah berikut setelah melakukan modifikasi yang diperlukan yang tercantum di bawah ini:

```
curl --request POST \
  --url '{TOKEN ENDPOINT}' \
  --header 'content-type: application/x-www-form-urlencoded' \
  --data grant_type=authorization_code \
  --data 'client_id={CLIENT ID}' \
  --data client_secret={CLIENT SECRET} \
  --data code={CODE} \
  --data 'redirect_uri={REDIRECT URI}'
```

   a. Ganti*{TOKEN ENDPOINT}*dengan titik akhir token untuk IdP OIDC Anda.

   b. Ganti*{CLIENT ID}*dengan ID Klien dari klien OAuth Anda.

   c. Ganti*{CLIENT SECRET}*dengan Rahasia Klien dari klien OAuth Anda.

   d. Ganti*{CODE}*dengan parameter kueri kode otentikasi yang Anda salin pada langkah 4.

   e. Ganti*{REDIRECT URI}*dengan URL portal pekerja.

   Berikut ini adalah contoh permintaan Curl setelah melakukan modifikasi yang dijelaskan di atas:

```
curl --request POST \
  --url 'https://example.com/token' \
  --header 'content-type: application/x-www-form-urlencoded' \
  --data grant_type=authorization_code \
  --data 'client_id=f490a907-9bf1-4471-97aa-6bfd159f81ac' \
  --data client_secret=client-secret \
```

```
  --data code=MCNYDB... \
  --data 'redirect_uri=https://example.labeling.sagemaker.aws/oauth2/idpresponse'
```

6. Langkah ini tergantung jenis`access_token`pengembalian IdP Anda, token akses teks biasa atau token akses JWT.

   - Jika IdP Anda tidak mendukung token akses JWT,`access_token`mungkin teks biasa (misalnya, UUID). Tanggapan yang Anda lihat mungkin terlihat seperti berikut ini. Dalam hal ini, pindah ke langkah 7.

     ```
     {
       "access_token":"179c144b-fccb-4d96-a28f-eea060f39c13",
       "token_type":"Bearer",
       "expires_in":3600,
       "refresh_token":"ef43e52e-9b4f-410c-8d4c-d5c5ee57631a",
       "scope":"openid"
     }
     ```

   - Jika IdP Anda mendukung token akses JWT, langkah 5 harus menghasilkan token akses dalam format JWT. Misalnya, respons mungkin terlihat seperti berikut:

     ```
     {
         "access_token":"eyJh...JV_adQssw5c",
         "refresh_token":"i6mapTIAVSp2oJkgUnCACKKfZxt_H5MBLiqcybBBd04",
         "refresh_token_expires_in":6327,
         "scope":"openid",
         "id_token":"eyJ0eXAiOiJK9...-rDaQzUHl6cQQWNiDpWOl_lxXjQEvQ"
     }
     ```

     Salin JWT dan decode itu. Anda dapat menggunakan skrip python atau situs web pihak ketiga untuk memecahkan kode itu. Misalnya, Anda dapat mengunjungi situs webhttps://jwt.io/dan tempel JWT keEncodedkotak untuk memecahkan kode itu.

     Pastikan respons yang diterjemahkan berisi yang berikut ini:

     - KlasterWajib SageMaker klaim dalam tabel yang ditemukan diKirim Klaim yang Diperlukan dan Opsional ke Ground Truth dan Amazon A2I. Jika tidak, Anda harus mengkonfigurasi ulang IdP OIDC Anda untuk memuat klaim ini.
     - KlasterPenerbitAnda menentukan ketika Anda mengatur tenaga kerja IdP.

7. Di terminal dan masukkan perintah berikut setelah membuat modifikasi yang diperlukan tercantum di bawah ini:

```
curl -X POST -H 'Authorization: Bearer {ACCESS TOKEN}' -d '' -k -v {USERINFO
 ENDPOINT}
```

a. Ganti*{USERINFO ENDPOINT}*dengan endpoint info pengguna untuk IDP OIDC Anda.

b. Ganti*{ACCESS TOKEN}*dengan token akses dalam respons yang Anda terima pada langkah 7. Ini adalah entri untuk`"access_token"`parameter.

Berikut ini adalah contoh permintaan Curl setelah melakukan modifikasi yang dijelaskan di atas:

```
 curl -X POST -H 'Authorization: Bearer eyJ0eX...' -d '' -k -v https://example.com/
userinfo
```

8. Respon terhadap langkah terakhir dalam prosedur di atas mungkin terlihat mirip dengan blok kode berikut.

   Jika`access_token`kembali pada langkah 6 adalah teks biasa, Anda harus memverifikasi bahwa respon ini berisi informasi yang diperlukan. Dalam kasus ini, respons harus berisiWajib SageMaker klaim dalam tabel yang ditemukan di[Kirim Klaim yang Diperlukan dan Opsional ke Ground Truth dan Amazon A2I](). Misalnya,`sagemaker-groups`,`sagamaker-name`.

```
{
    "sub":"122",
    "exp":"10000",
    "sagemaker-groups":["group1","group2"]
    "sagemaker-name":"name",
    "sagemaker-sub":"122",
    "sagemaker-client_id":"123456"
}
```

Langkah Selanjutnya

Setelah Anda membuat tenaga kerja pribadi menggunakan IdP Anda dan memverifikasi respons otentikasi IdP Anda, Anda dapat membuat tim kerja menggunakan grup IdP Anda. Untuk mempelajari selengkapnya, lihat [Mengelola Tenaga Kerja Pribadi (OIDC iDP)]().

Anda dapat membatasi akses pekerja ke tugas ke alamat IP tertentu, dan memperbarui atau menghapus tenaga kerja Anda menggunakan SageMaker API API API. Untuk mempelajari informasi lebih lanjut, lihat [Mengelola Tenaga Kerja Pribadi Menggunakan Amazon SageMaker API]().

Mengelola Tenaga Kerja Pribadi (OIDC iDP)

Setelah Anda membuat tenaga kerja pribadi menggunakan OpenID Connect (OIDC) Identity Provider (IdP), Anda dapat mengelola pekerja Anda menggunakan IdP Anda. Misalnya, Anda dapat menambahkan, menghapus, dan mengelompokkan pekerja secara langsung melalui IdP.

Untuk menambahkan pekerja ke Amazon SageMaker Pekerjaan pelabelan Ground Truth (Ground Truth) atau tugas peninjauan manusia Amazon Augmented AI (Amazon A2I), Anda membuat tim kerja menggunakan 1-10 grup IdP dan menetapkan tim kerja itu ke pekerjaan atau tugas. Anda menetapkan tim kerja ke pekerjaan atau tugas dengan menentukan tim kerja tersebut saat Anda membuat pekerjaan pelabelan (Ground Truth) atau alur kerja peninjauan manusia (Amazon A2I).

Anda hanya dapat menetapkan satu tim untuk setiap pekerjaan pelabelan atau alur kerja peninjauan manusia. Anda dapat menggunakan tim yang sama untuk membuat beberapa pekerjaan pelabelan atau tugas peninjauan manusia. Anda juga dapat membuat beberapa tim kerja untuk mengerjakan pekerjaan pelabelan yang berbeda atau tugas peninjauan manusia.

Prasyarat

Untuk membuat dan mengelola tim kerja pribadi menggunakan grup IdP OIDC Anda, pertama-tama Anda harus membuat tenaga kerja menggunakan SageMaker Operasi API`CreateWorkforce`. Untuk mempelajari selengkapnya, lihat Membuat Tenaga Kerja Pribadi (OIDC iDP).

Menambah tim kerja

Anda dapat menggunakan SageMaker konsol untuk membuat tim kerja pribadi menggunakan tenaga kerja IdP OIDC Anda diPelabelan tenagahalaman di bawahGround Truth. Jika Anda membuat pekerjaan pelabelan Ground Truth, Anda juga dapat membuat tim kerja pribadi sambil membuat pekerjaan pelabelan.

> **Note**
>
> Anda membuat dan mengelola tim kerja untuk Amazon A2I di area Ground Truth SageMakerKonsol.

Anda juga dapat menggunakan SageMaker API dan SDK khusus bahasa terkait untuk membuat tim kerja pribadi.

Gunakan prosedur berikut untuk mempelajari cara membuat tim kerja pribadi menggunakan SageMakerkonsol dan API.

Untuk membuat tim kerja pribadi di halaman Labeling workforce (konsol)

1. Pergi ke area Ground Truth dari SageMaker Konsol:https://console.aws.amazon.com/sagemaker/groundtruth.
2. PilihPelabelan tenaga.
3. PilihPrivat.
4. DiTim privatbagian, pilihBuat tim privat.
5. DiRincian timbagian, masukkanNama tim.
6. DiMenambah pekerjaBagian, masukkan nama grup pengguna tunggal. Semua pekerja yang terkait dengan grup ini di IdP Anda ditambahkan ke tim kerja ini.
7. Untuk menambahkan lebih dari satu grup pengguna, pilihTambahkan grup pengguna barudan masukkan nama grup pengguna yang ingin Anda tambahkan ke tim kerja ini. Masukkan satu grup pengguna per baris.
8. (Opsional) Untuk pekerjaan pelabelan Ground Truth, jika Anda memberikan email untuk pekerja di JWT Anda, Ground Truth memberi tahu pekerja ketika tugas pelabelan baru tersedia jika Anda memilih topik SNS.
9. PilihBuat tim privat.

Untuk membuat tim kerja pribadi sambil membuat pekerjaan pelabelan Ground Truth (konsol)

1. Pergi ke area Ground Truth dari SageMaker Konsol:https://console.aws.amazon.com/sagemaker/groundtruth.
2. PilihLowongan pelabelan.
3. Gunakan petunjuknyaMembuat Job Pelabelan (Konsol)untuk membuat pekerjaan pelabelan. Berhenti ketika Anda sampai kePekerjabagian pada halaman kedua.
4. PilihPrivatuntuk tipe pekerja Anda.
5. MasukkanNama tim.
6. DiMenambah pekerjabagian, masukkan nama grup pengguna tunggal di bawahGrup pengguna. Semua pekerja yang terkait dengan grup ini di IdP Anda ditambahkan ke tim kerja ini.

> **Important**
>
> Nama grup yang Anda tentukanGrup penggunaharus sesuai dengan nama grup yang ditentukan dalam IdP OIDC Anda.

7. Untuk menambahkan lebih dari satu grup pengguna, pilihTambahkan grup pengguna barudan masukkan nama grup pengguna yang ingin Anda tambahkan ke tim kerja ini. Masukkan satu grup pengguna per baris.
8. Selesaikan semua langkah yang tersisa untuk membuat pekerjaan pelabelan Anda.

Tim pribadi yang Anda buat digunakan untuk pekerjaan pelabelan ini, dan tercantum dalamPelabelan tenagabagian SageMaker Konsol.

Untuk membuat tim kerja pribadi menggunakan SageMaker API

Anda dapat membuat tim kerja pribadi menggunakan SageMaker Operasi API`CreateWorkteam`.

Saat Anda menggunakan operasi ini, cantumkan semua grup pengguna yang ingin Anda sertakan dalam tim kerja di`OidcMemberDefinition`parameter`Groups`.

> **Important**
>
> Nama grup yang Anda tentukan`Groups`harus sesuai dengan nama grup yang ditentukan dalam IdP OIDC Anda.

Misalnya, jika nama grup pengguna Anda`group1`,`group2`, dan`group3`di IdP OIDC Anda, konfigurasikan`OidcMemberDefinition`sebagai berikut:

```
"OidcMemberDefinition": {
   "Groups": ["group1", "group2", "group3"]
 }
```

Selain itu, Anda harus memberi nama tim kerja menggunakan`WorkteamName`parameter.

Menambah atau menghapus grup IdP dari tim kerja

Setelah membuat tim kerja, Anda dapat menggunakan SageMaker API untuk mengelola tim kerja itu. Gunakan`UpdateWorkteam`operasi untuk memperbarui grup pengguna IdP yang disertakan dalam tim kerja itu.

- Gunakan`WorkteamName`parameter untuk mengidentifikasi tim kerja yang ingin Anda perbarui.
- Saat Anda menggunakan operasi ini, cantumkan semua grup pengguna yang ingin Anda sertakan dalam tim kerja di`OidcMemberDefinition`parameter`Groups`. Jika grup pengguna dikaitkan

dengan tim kerja dan Anda melakukannyatidakmemasukkannya dalam daftar ini, bahwa kelompok pengguna tidak lagi terkait dengan tim kerja ini.

Menghapus tim kerja

Anda dapat menghapus tim kerja menggunakan SageMaker konsol dan SageMaker API

Untuk menghapus tim kerja privat di SageMaker konsol

1. Pergi ke area Ground Truth dari SageMaker Konsol:https://console.aws.amazon.com/sagemaker/groundtruth.
2. PilihPelabelan tenaga.
3. PilihPrivat.
4. DiTim privatBagian, pilih tim kerja yang ingin Anda hapus.
5. Pilih Hapus.

Untuk menghapus tim kerja privat (API)

Anda dapat menghapus tim kerja privat menggunakan SageMaker Operasi API`DeleteWorkteam`.

Mengelola Pekerja Perorangan

Ketika Anda membuat tenaga kerja menggunakan IdP OIDC Anda sendiri, Anda tidak dapat menggunakan Ground Truth atau Amazon A2I untuk mengelola pekerja individu.

- Untuk menambahkan pekerja ke tim kerja, tambahkan pekerja tersebut ke grup yang terkait dengan tim kerja tersebut.
- Untuk menghapus pekerja dari tim kerja, hapus pekerja tersebut dari semua grup pengguna yang terkait dengan tim kerja tersebut.

Perbarui, Hapus, dan Jelaskan Tenaga Kerja Anda

Anda dapat memperbarui, menghapus, dan menjelaskan tenaga kerja IdP OIDC Anda menggunakan SageMaker API Berikut ini adalah daftar operasi API yang dapat Anda gunakan untuk mengelola tenaga kerja. Untuk detail tambahan, termasuk bagaimana Anda dapat menemukan nama tenaga kerja Anda, lihatMengelola Tenaga Kerja Pribadi Menggunakan Amazon SageMaker API.

- `UpdateWorkforce`- Anda mungkin ingin memperbarui tenaga kerja yang dibuat menggunakan IdP OIDC Anda sendiri untuk menentukan titik akhir otorisasi, titik akhir token, atau penerbit yang berbeda. Anda dapat memperbarui parameter apa pun yang ditemukan di`OidcConfig`menggunakan operasi ini.

  Anda hanya dapat memperbarui konfigurasi IdP OIDC Anda ketika tidak ada tim kerja yang terkait dengan tenaga kerja Anda. Untuk mempelajari cara menghapus tim kerja, lihatMenghapus tim kerja.
- `DeleteWorkforce`- Gunakan operasi ini untuk menghapus tenaga kerja pribadi Anda. Jika Anda memiliki tim kerja yang terkait dengan tenaga kerja Anda, Anda harus menghapus tim kerja tersebut sebelum menghapus tenaga kerja Anda. Untuk informasi selengkapnya, lihat Menghapus tim kerja.
- `DescribeWorkforce`— Gunakan operasi ini untuk mencantumkan informasi tenaga kerja pribadi, termasuk nama tenaga kerja, Amazon Resource Name (ARN), dan, jika berlaku, rentang alamat IP yang diizinkan (CIDR).

Mengelola Tenaga Kerja Pribadi Menggunakan Amazon SageMaker API

Anda dapat menggunakan Amazon SageMaker Operasi API untuk mengelola, memperbarui, dan menghapus tenaga kerja privat Anda. Untuk setiap operasi API yang ditautkan di halaman ini, Anda dapat menemukan daftar SDK khusus bahasa yang didukung dan dokumentasinya diLihat Jugabagian dokumentasi API.

Temukan Nama Tenaga Kerja Anda

Beberapa SageMaker operasi API yang berhubungan dengan tenaga kerja memerlukan nama tenaga kerja Anda sebagai masukan. Anda dapat melihat nama pribadi dan tenaga kerja vendor Amazon Cognito atau OIDC IdP diAWSWilayah menggunakan`ListWorkforces`Operasi APIAWSWilayah.

Jika Anda membuat tenaga kerja menggunakan IdP OIDC Anda sendiri, Anda dapat menemukan nama tenaga kerja Anda di area Ground Truth di SageMaker konsol.

Untuk menemukan nama tenaga kerja Anda di SageMaker konsol

1. Pergi ke area Ground Truth dari SageMaker konsol:https://console.aws.amazon.com/sagemaker/groundtruth.
2. PilihPelabelan.

3. PilihPrivat.
4. DiRingkasan tenaga kerja privatbagian, cari ARN tenaga kerja Anda. Nama tenaga kerja Anda terletak di ujung ARN ini. Misalnya, jika ARN`arn:aws:sagemaker:us-east-2:111122223333:workforce/example-workforce`, nama tenaga kerja adalah`example-workforce`.

Batasi Akses Pekerja ke Tugas ke Alamat IP yang Diijinkan

Secara default, tenaga kerja tidak terbatas pada alamat IP tertentu. Anda dapat menggunakan`UpdateWorkforce`operasi untuk mengharuskan pekerja menggunakan rentang alamat IP tertentu (CIDR) untuk mengakses tugas. Jika Anda menentukan satu atau lebih CIDR, pekerja yang mencoba mengakses tugas menggunakan alamat IP apa pun di luar rentang yang ditentukan akan ditolak dan akan mendapatkan pesan galat HTTP 204 No Content di portal pekerja. Anda dapat menentukan hingga 10 nilai CIDR menggunakan`UpdateWorkforce`.

Setelah Anda membatasi tenaga kerja Anda untuk satu atau lebih CIDR, output dari`UpdateWorkforce`mencantumkan semua CIDR yang diijinkan. Anda juga dapat menggunakan`DescribeWorkforce`operasi untuk melihat semua CIDR yang diijinkan untuk tenaga kerja.

Perbarui Konfigurasi Tenaga Kerja Penyedia Identitas OIDC

Anda mungkin ingin memperbarui tenaga kerja yang dibuat menggunakan IdP OIDC Anda sendiri untuk menentukan titik akhir otorisasi, titik akhir token, atau penerbit yang berbeda. Anda dapat memperbarui parameter apa pun yang ditemukan di`OidcConfig`menggunakan`UpdateWorkforce` operation.

> **Important**
>
> Anda hanya dapat memperbarui konfigurasi IdP OIDC Anda ketika tidak ada tim kerja yang terkait dengan tenaga kerja Anda. Anda dapat menghapus tim kerja privat menggunakan`DeleteWorkteam` operation.

Menghapus Tenaga Kerja Pribadi

Anda hanya dapat memiliki satu tenaga kerja privatAWSWilayah. Anda mungkin ingin menghapus tenaga kerja pribadi Anda diAWSWilayah saat:

- Anda ingin membuat tenaga kerja menggunakan kumpulan pengguna Amazon Cognito baru.
- Anda telah membuat tenaga kerja pribadi menggunakan Amazon Cognito dan Anda ingin membuat tenaga kerja menggunakan OpenID Connect (OIDC) Identity Provider (IdP) Anda sendiri.

Untuk menghapus tenaga kerja privat, gunakan`DeleteWorkforce`Operasi API. Jika Anda memiliki tim kerja yang terkait dengan tenaga kerja Anda, Anda harus menghapus tim kerja tersebut sebelum menghapus tenaga kerja Anda. Anda dapat menghapus tim kerja privat menggunakan`DeleteWorkteam` operation.

Melacak Kinerja Pekerja

Amazon SageMaker Ground Truth log peristiwa pekerja ke Amazon CloudWatch, seperti ketika pekerja memulai atau mengirimkan tugas. Gunakan Amazon CloudWatch metrik untuk mengukur dan melacak throughput di seluruh tim atau untuk pekerja individu.

> **Important**
>
> Pelacakan peristiwa pekerja tidak tersedia untuk alur kerja tinjauan manusia Amazon Augmented AI.

Mengaktifkan Pelacakan

Selama proses pengaturan untuk tim kerja baru, izin untuk Amazon CloudWatch penebangan peristiwa pekerja dibuat. Karena fitur ini ditambahkan pada Agustus 2019, tim kerja yang dibuat sebelumnya mungkin tidak memiliki izin yang benar. Jika semua tim kerja Anda dibuat sebelum Agustus 2019, buat tim kerja baru. Itu tidak memerlukan anggota dan dapat dihapus setelah dibuat, tetapi dengan membuatnya, Anda menetapkan izin dan menerapkannya ke semua tim kerja Anda, terlepas dari kapan mereka dibuat.

Memeriksa Logs

Setelah pelacakan diaktifkan, aktivitas pekerja Anda dicatat. Buka Amazon CloudWatchkonsol dan pilihBeberapa catatandi panel navigasi. Anda akan melihat grup log bernama/aws/sagemaker/groundtruth/WorkerActivity.

Setiap tugas selesai diwakili oleh entri log, yang berisi informasi tentang pekerja, tim mereka, pekerjaan, ketika tugas diterima, dan ketika itu diserahkan.

Example Entri log

```
{
  "worker_id": "cd449a289e129409",
  "cognito_user_pool_id": "us-east-2_IpicJXXXX",
  "cognito_sub_id": "d6947aeb-0650-447a-ab5d-894db61017fd",
  "task_accepted_time": "Wed Aug 14 16:00:59 UTC 2019",
  "task_submitted_time": "Wed Aug 14 16:01:04 UTC 2019",
  "task_returned_time": "",
  "task_declined_time": "",
  "workteam_arn": "arn:aws:sagemaker:us-east-2:############:workteam/private-crowd/
Sample-labeling-team",
  "labeling_job_arn": "arn:aws:sagemaker:us-east-2:############:labeling-job/metrics-
demo",
  "work_requester_account_id": "############",
  "job_reference_code": "############",
  "job_type": "Private",
  "event_type": "TasksSubmitted",
  "event_timestamp": "1565798464"
}
```

Titik data yang berguna dalam setiap peristiwa adalah`cognito_sub_id`. Anda dapat mencocokkannya dengan pekerja individu.

1. Buka Amazon SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di bawahGround Truthbagian, pilihTenaga kerja.
3. MemiilihPrivat.
4. Pilih nama sebuah tim diTim pribadiBagian
5. DiRingkasan timBagian, pilih grup pengguna yang diidentifikasi di bawahGrup pengguna Amazon Cognito. Itu akan membawa Anda ke grup di konsol Amazon Cognito.
6. KlasterGruphalaman mencantumkan pengguna dalam grup. Pilih tautan pengguna mana pun diNama penggunakolom untuk melihat informasi lebih lanjut tentang pengguna, termasuk yang uniksubID

Untuk mendapatkan informasi tentang semua anggota tim, gunakanListUserstindakan (contohdi API Amazon Cognito.

Gunakan Metrik Log

Jika Anda tidak ingin menulis skrip Anda sendiri untuk memproses dan memvisualisasikan informasi log mentah, Amazon CloudWatch metrik memberikan wawasan tentang aktivitas pekerja untuk Anda.

Melihat metrik

1. Buka CloudWatch konsol dihttps://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Metrik.
3. Pilih`AWS/SageMaker/Workteam`ruang nama, lalu jelajahimetrik yang tersedia. Misalnya, memilihTim KerjadanTenaga Kerjametrik memungkinkan Anda menghitung waktu rata-rata per tugas yang dikirimkan untuk pekerjaan pelabelan tertentu.

Untuk informasi selengkapnya, lihatMenggunakan Amazon CloudWatch Metrik.

Membuat dan mengelola topik Amazon SNS untuk tim kerja Anda

Gunakan prosedur dalam topik ini saat Anda ingin:

- Buat topik yang Anda inginkan untuk berlangganan tim kerja yang ada.
- Buat topik sebelum Anda membuat tim kerja.
- Buat atau ubah tim tugas dengan panggilan API, dan tentukan topik Amazon Resource Name (ARN).

Jika Anda membuat tim kerja menggunakan konsol, konsol menyediakan opsi untuk membuat topik baru untuk tim sehingga Anda tidak perlu melakukan langkah-langkah ini.

> **Important**
>
> Fitur Amazon SNS tidak didukung oleh Amazon A2I. Jika Anda berlangganan tim kerja Anda ke topik Amazon SNS, pekerja hanya akan menerima pemberitahuan tentang pekerjaan pelabelan Ground Truth. Pekerja tidak akan menerima pemberitahuan tentang tugas peninjauan manusia Amazon A2I baru.

Buat topik Amazon SNS

Langkah-langkah untuk membuat topik Amazon SNS untuk pemberitahuan tim kerja serupa dengan langkah-langkah diMemulaidi dalamPanduan Developer Amazon SNS, dengan satu tambahan yang signifikan—Anda harus menambahkan kebijakan akses sehingga Amazon SageMaker dapat mempublikasikan pesan ke topik atas nama Anda.

Untuk menambahkan kebijakan saat membuat topik

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. MasukMembuat Topik, masukkan nama topik Anda dan kemudian pilihLangkah selanjutnya.
3. Di Kebijakan akses, pilih Lanjutan.
4. DiPenyunting JSON, find`Resource`properti, yang menampilkan ARN topik.
5. Salin`Resource`Nilai ARN.
6. Sebelum penjepit penutupan akhir (`]`), tambahkan kebijakan berikut.

```
   , {
       "Sid": "AwsSagemaker_SnsAccessPolicy",
       "Effect": "Allow",
       "Principal": {
           "Service": "sagemaker.amazonaws.com"
       },
       "Action": "sns:Publish",
       "Resource": "arn:partition:sns:region:111122223333:MyTopic", # ARN of the
topic you copied in the previous step
       "Condition": {
           "ArnLike": {
               "aws:SourceArn":
"arn:partition:sagemaker:region:111122223333:workteam/*" # Workteam ARN
           },
           "StringEquals": {
               "aws:SourceAccount": "111122223333" # SNS topic account
           }
       }
   }
```

7. Buat topik.

Setelah Anda membuat topik, topik muncul diTopiklayar ringkasan. Untuk informasi selengkapnya tentang cara membuat topik, lihatMembuat Topikdi dalamPanduan Developer Amazon SNS.

Kelola langganan pekerja

Jika Anda berlangganan tim kerja ke topik setelah Anda telah membuat tim kerja, anggota tim kerja individu yang ditambahkan ke tim ketika tim kerja dibuat tidak secara otomatis berlangganan topik. Untuk informasi tentang cara berlangganan alamat email pekerja ke topik, lihatBerlangganan Titik Akhir untuk Topik Amazon SNSdi dalamPanduan Developer Amazon SNS.

Satu-satunya situasi di mana pekerja secara otomatis berlangganan topik Anda adalah ketika Anda membuat atau mengimpor grup pengguna Amazon Cognito pada saat Anda membuat tim kerjadanAnda mengatur langganan topik saat Anda membuat tim kerja itu. Untuk informasi selengkapnya tentang cara membuat dan mengelola tim kerja Anda dengan Amazon Cognito, lihatMembuat Tim Kerja (Amazon Cognito Console).

Crowd HTML Elemen Referensi

Crowd HTML Elements adalah komponen web, standar web yang abstrak markup HTML, CSS, dan JavaScript fungsionalitas menjadi tag HTML atau set tag. Amazon SageMaker menyediakan pelanggan dengan kemampuan untuk merancang template tugas kustom mereka sendiri dalam HTML.

Sebagai titik awal, Anda dapat menggunakan template yang dibangun menggunakan Crowd HTML Elements dari salah satu dari berikut GitHub repositori:

- Contoh UI tugas untuk Amazon SageMaker Ground Truth
- Lebih dari 60 contoh UI tugas untuk Amazon Augmented AI (A2I)

Repositori ini mencakup template yang dirancang untuk audio, gambar, teks, video, dan jenis pelabelan data dan tugas anotasi lainnya.

Untuk informasi selengkapnya tentang cara menerapkan templat khusus di Amazon SageMaker Ground Truth, lihatMembuat Alur Kerja Pelabelan Kustom. Untuk mempelajari lebih lanjut tentang template kustom di Amazon Augmented AI, lihatBuat Templat Tugas Pekerja Kustom.

SageMaker Elemen Crowd

Berikut ini adalah daftar Crowd HTML Elements yang membuat membangun template kustom lebih mudah dan menyediakan UI akrab bagi pekerja. Elemen-elemen ini didukung di Ground Truth, Augmented AI, dan Mechanical Turk.

Topik

- peringatan kerumunan
- kerumunan lencana
- Tombol kerumunan
- kerumunan-batas-kotak
- Kartu kerumunan
- kerumunan-kotak centang
- klasifikasi kerumunan
- crowd-classifier-multi-pilih
- kerumunan-entitas-anotasi
- kerumunan-fab
- Formulir kerumunan
- crowd-icon-tombol
- kerumunan-gambar-classifier
- crowd-image-classifier-multi-pilih
- Masukan kerumunan
- kerumunan-instance-segmentasi
- Instruksi kerumunan
- kerumunan-keypoint
- garis kerumunan
- kerumunan-modal
- poligon
- kerumunan-polyline
- tombol radio kerumunan
- kelompok radio kerumunan
- kerumunan-semantik-segmentasi
- kerumunan-slider
- Tab kerumunan

- kerumunan-tab
- kerumunan-teks-daerah
- kerumunan bersulang
- crowd-toggle-tombol

peringatan kerumunan

Pesan yang mengingatkan pekerja untuk situasi saat ini.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-alert>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <div id="errorBox"></div>

  <crowd-keypoint
    src="{{ task.input.taskObject | grant_read_access }}"
    labels="['Item A', 'Item B', 'Item C']"
    header="Please locate the centers of each item."
    name="annotatedResult">
    <short-instructions>
      Describe your task briefly here and give examples
    </short-instructions>
    <full-instructions>
      Give additional instructions and good/bad examples here
    </full-instructions>
  </crowd-keypoint>
</crowd-form>

<script>
  var num_obj = 1;

  document.querySelector('crowd-form').onsubmit = function(e) {
    const keypoints = document.querySelector('crowd-keypoint').value.keypoints ||
 document.querySelector('crowd-keypoint')._submittableValue.keypoints;
    const labels = keypoints.map(function(p) {
```

```
      return p.label;
    });

    // 1. Make sure total number of keypoints is correct.
    var original_num_labels = document.getElementsByTagName("crowd-keypoint")
[0].getAttribute("labels");

    original_num_labels = original_num_labels.substring(2, original_num_labels.length -
 2).split("\",\"");
    var goalNumKeypoints = num_obj*original_num_labels.length;
    if (keypoints.length != goalNumKeypoints) {
      e.preventDefault();
      errorBox.innerHTML = '<crowd-alert type="error" dismissible>You must add all
 keypoint annotations and use each label only once.</crowd-alert>';
      errorBox.scrollIntoView();
      return;
    }

    // 2. Make sure all labels are unique.
    labelCounts = {};
    for (var i = 0; i < labels.length; i++) {
      if (!labelCounts[labels[i]]) {
        labelCounts[labels[i]] = 0;
      }
      labelCounts[labels[i]]++;
    }
    const goalNumSingleLabel = num_obj;

    const numLabels = Object.keys(labelCounts).length;

    Object.entries(labelCounts).forEach(entry => {
      if (entry[1] != goalNumSingleLabel) {
        e.preventDefault();
        errorBox.innerHTML = '<crowd-alert type="error" dismissible>You must use each
 label only once.</crowd-alert>';
        errorBox.scrollIntoView();
      }
    })
  };
</script>
```

Atribut

Atribut berikut didukung oleh elemen ini.

tidak dapat diterima

Sebuah saklar Boolean yang, jika ada, memungkinkan pesan yang akan ditutup oleh pekerja.

jenis

String yang menentukan jenis pesan yang akan ditampilkan. Nilai yang mungkin adalah “info” (default), “sukses”, “error”, dan “warning”.

Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen Anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan lencana

Ikon yang mengapung di sudut kanan atas elemen lain yang dilekatkan.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat yang menggunakan`<crowd-badge>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-image-classifier
    name="crowd-image-classifier"
    src="https://unsplash.com/photos/NLUkAA-nDdE"
    header="Choose the correct category for this image."
```

```
    categories="['Person', 'Umbrella', 'Chair', 'Dolphin']"
  >
    <full-instructions header="Classification Instructions">
      <p>Read the task carefully and inspect the image.</p>
      <p>Choose the appropriate label that best suits the image.</p>
    </full-instructions>

    <short-instructions id="short-instructions">
      <p>Read the task carefully and inspect the image.</p>
      <p>Choose the appropriate label that best suits the image.</p>
      <crowd-badge icon="star" for="short-instructions"/>
    </short-instructions>
  </crowd-image-classifier>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

untuk

Sebuah string yang menentukan ID dari elemen yang lencana terpasang.

ikon

String yang menentukan ikon yang akan ditampilkan di lencana. String harus berupa nama ikon dari open-sourcebesi-ikonset, yang sudah dimuat sebelumnya, atau URL ke ikon kustom.

Atribut ini menimpalabelatribut.

Berikut ini adalah contoh sintaks yang dapat Anda gunakan untuk menambahkan besi-ikon ke`<crowd-badge>`Elemen HTML. Ganti*`icon-name`*dengan nama ikon yang ingin Anda gunakanIkon.

```
<crowd-badge icon="icon-name" for="short-instructions"/>
```

label

Teks yang akan ditampilkan di lencana. Tiga karakter atau kurang dianjurkan karena teks yang terlalu besar akan meluap area lencana. Ikon dapat ditampilkan alih-alih teks dengan mengaturikonatribut.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

Tombol kerumunan

Sebuah tombol gaya yang mewakili beberapa tindakan.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat yang menggunakan`<crowd-button>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-image-classifier
    name="crowd-image-classifier"
    src="https://unsplash.com/photos/NLUkAA-nDdE"
    header="Please select the correct category for this image"
    categories="['Person', 'Umbrella', 'Chair', 'Dolphin']"
  >
    <full-instructions header="Classification Instructions">
      <p>Read the task carefully and inspect the image.</p>
      <p>Choose the appropriate label that best suits the image.</p>
    </full-instructions>
    <short-instructions>
      <p>Read the task carefully and inspect the image.</p>
      <p>Choose the appropriate label that best suits the image.</p>
      <crowd-button>
        <iron-icon icon="question-answer"/>
      </crowd-button>
    </short-instructions>
```

```
  </crowd-image-classifier>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan tombol sebagai dinonaktifkan dan mencegah klik.

tindakan-formulir

Sebuah switch yang baik mengirimkan induknyaFormulir kerumunanelemen, jika diatur ke “submit”, atau me-reset <crowd-form>elemen induknya, jika diatur ke “reset”.

href

URL ke sumber daya online. Gunakan properti ini jika Anda memerlukan tautan yang ditata sebagai tombol.

ikon

String yang menentukan ikon yang akan ditampilkan di samping teks tombol. String harus berupa nama ikon dari open-sourcebesi-ikonset, yang sudah dimuat sebelumnya. Misalnya, untuk memasukkanmencaribesi-icon, gunakan yang berikut ini:

```
<crowd-button>
    <iron-icon icon="search"/>
</crowd-button>
```

Ikon diposisikan ke kiri atau kanan teks, seperti yang ditentukan olehikon-menyelaraskanAtribut.

Cara menggunakan ikon kustom lihaturl.

ikon-menyelaraskan

Posisi kiri atau kanan ikon relatif terhadap teks tombol. Defaultnya adalah “left”.

url

URL untuk gambar kustom untuk ikon. Sebuah gambar kustom dapat digunakan sebagai pengganti ikon standar yang ditentukan olehikonAtribut.

memuat

Sebuah saklar Boolean yang, jika ada, menampilkan tombol sebagai berada dalam keadaan pemuatan. Atribut ini memiliki prioritas atasdinonaktifkanatribut jika kedua atribut yang hadir.

target

Bila Anda menggunakan`href`atribut untuk membuat tombol bertindak sebagai hyperlink ke URL tertentu,`target`atribut opsional menargetkan bingkai atau jendela di mana URL terkait harus memuat.

varian

Gaya umum tombol. Gunakan “primer” untuk tombol utama, “normal” untuk tombol sekunder, “tautan” untuk tombol tersier, atau “ikon” untuk hanya menampilkan ikon tanpa teks.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen Anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-batas-kotak

Widget untuk menggambar persegi panjang pada gambar dan menetapkan label ke bagian gambar yang tertutup di setiap persegi panjang.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-bounding-box>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk

melihat pratinjau dan berinteraksi dengan template ini. Untuk contoh lainnya, lihat iniRepositori GitHub.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-bounding-box
    name="annotatedResult"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Draw bounding boxes around all the cats and dogs in this image"
    labels="['Cat', 'Dog']"
  >
    <full-instructions header="Bounding Box Instructions" >
      <p>Use the bounding box tool to draw boxes around the requested target of
 interest:</p>
      <ol>
        <li>Draw a rectangle using your mouse over each instance of the target.</li>
        <li>Make sure the box does not cut into the target, leave a 2 - 3 pixel
 margin</li>
        <li>
          When targets are overlapping, draw a box around each object,
          include all contiguous parts of the target in the box.
          Do not include parts that are completely overlapped by another object.
        </li>
        <li>
          Do not include parts of the target that cannot be seen,
          even though you think you can interpolate the whole shape of the target.
        </li>
        <li>Avoid shadows, they're not considered as a part of the target.</li>
        <li>If the target goes off the screen, label up to the edge of the image.</li>
      </ol>
    </full-instructions>

    <short-instructions>
      Draw boxes around the requested target of interest.
    </short-instructions>
  </crowd-bounding-box>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nilai awal

Array objek JSON, yang masing-masing menetapkan kotak pembatas ketika komponen dimuat. Setiap objek JSON dalam array berisi properti berikut. Kotak pembatas diatur melalui`initial-value`properti dapat disesuaikan dan apakah atau tidak jawaban pekerja disesuaikan dilacak melalui`initialValueModified`boolean dalam output jawaban pekerja.

- tingginya- Tinggi kotak dalam piksel.
- label— Teks yang ditugaskan ke kotak sebagai bagian dari tugas pelabelan. Teks ini harus sesuai dengan salah satu label yang didefinisikan dalamlabelatribut <crowd-bounding-box>elemen.
- kiri— Jarak sudut kiri atas kotak dari sisi kiri gambar, diukur dalam piksel.
- teratas— Jarak sudut kiri atas kotak dari atas gambar, diukur dalam piksel.
- Lebar- Lebar kotak dalam piksel.

  Anda dapat mengekstrak nilai awal kotak pembatas dari file manifes dari pekerjaan sebelumnya dalam template kustom menggunakan bahasa template Liquid:

```
initial-value="[
  {% for box in task.input.manifestLine.label-attribute-name-from-prior-
job.annotations %}
    {% capture class_id %}{{ box.class_id }}{% endcapture %}
    {% assign label = task.input.manifestLine.label-attribute-name-from-prior-job-
metadata.class-map[class_id] %}
    {
      label: {{label | to_json}},
      left: {{box.left}},
      top: {{box.top}},
      width: {{box.width}},
      height: {{box.height}},
    },
  {% endfor %}
 ]"
```

label

Sebuah array JSON diformat string, yang masing-masing adalah label yang pekerja dapat menetapkan untuk bagian gambar tertutup oleh persegi panjang. Batas:label 10.

nama

Nama widget ini. Ini digunakan sebagai kunci untuk input widget dalam output form.

src

URL gambar yang menggambar kotak pembatas.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak:instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut diperlukan oleh elemen ini.

instruksi lengkap

Petunjuk umum tentang cara menggambar kotak pembatas.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Output

Output berikut didukung oleh elemen ini.

boundingBoxes

Array objek JSON, yang masing-masing menentukan kotak pembatas yang telah dibuat oleh pekerja. Setiap objek JSON dalam array berisi properti berikut.

- tingginya- Tinggi kotak dalam piksel.

- label— Teks yang ditugaskan ke kotak sebagai bagian dari tugas pelabelan. Teks ini harus sesuai dengan salah satu label yang didefinisikan dalamlabelatribut <crowd-bounding-box>elemen.
- kiri— Jarak sudut kiri atas kotak dari sisi kiri gambar, diukur dalam piksel.
- teratas— Jarak sudut kiri atas kotak dari atas gambar, diukur dalam piksel.
- Lebar- Lebar kotak dalam piksel.

InputImageProperties

Sebuah objek JSON yang menentukan dimensi gambar yang sedang dianotasi oleh pekerja. Objek ini berisi properti berikut.

- tingginya— Tinggi, dalam piksel, gambar.
- Lebar— Lebar, dalam piksel, gambar.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari skenario penggunaan umum untuk elemen ini.

Label Tunggal, Kotak Tunggal/Beberapa Label, Kotak Tunggal

```
[
  {
    "annotatedResult": {
      "boundingBoxes": [
        {
          "height": 401,
          "label": "Dog",
          "left": 243,
          "top": 117,
          "width": 187
        }
      ],
      "inputImageProperties": {
        "height": 533,
        "width": 800
      }
    }
  }
]
```

Label Tunggal, Beberapa Kotak

```
[
  {
    "annotatedResult": {
      "boundingBoxes": [
        {
          "height": 401,
          "label": "Dog",
          "left": 243,
          "top": 117,
          "width": 187
        },
        {
          "height": 283,
          "label": "Dog",
          "left": 684,
          "top": 120,
          "width": 116
        }
      ],
      "inputImageProperties": {
        "height": 533,
        "width": 800
      }
    }
  }
]
```

Beberapa Label, Beberapa Kotak

```
[
  {
    "annotatedResult": {
      "boundingBoxes": [
        {
          "height": 395,
          "label": "Dog",
          "left": 241,
          "top": 125,
          "width": 158
        },
        {
```

```
          "height": 298,
          "label": "Cat",
          "left": 699,
          "top": 116,
          "width": 101
        }
      ],
      "inputImageProperties": {
        "height": 533,
        "width": 800
      }
    }
  }
]
```

Anda bisa memiliki banyak label yang tersedia, tetapi hanya yang digunakan muncul dalam output.

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

Kartu kerumunan

Sebuah kotak dengan tampilan tinggi untuk menampilkan informasi.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh dari template yang dirancang untuk tugas-tugas analisis sentimen yang menggunakan`<crowd-card>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<style>
  h3 {
    margin-top: 0;
  }
```

```
  crowd-card {
    width: 100%;
  }

  .card {
    margin: 10px;
  }

  .left {
    width: 70%;
    margin-right: 10px;
    display: inline-block;
    height: 200px;
  }

  .right {
    width: 20%;
    height: 200px;
    display: inline-block;
  }
</style>

<crowd-form>
  <short-instructions>
    Your short instructions here.
  </short-instructions>

  <full-instructions>
    Your full instructions here.
  </full-instructions>

  <div class="left">
    <h3>What sentiment does this text convey?</h3>
    <crowd-card>
      <div class="card">
        Nothing is great.
      </div>
    </crowd-card>
  </div>

  <div class="right">
    <h3>Select an option</h3>

    <select name="sentiment1" style="font-size: large" required>
```

```
      <option value="">(Please select)</option>
      <option>Negative</option>
      <option>Neutral</option>
      <option>Positive</option>
      <option>Text is empty</option>
    </select>
  </div>

  <div class="left">
    <h3>What sentiment does this text convey?</h3>
    <crowd-card>
      <div class="card">
        Everything is great!
      </div>
    </crowd-card>
  </div>

  <div class="right">
    <h3>Select an option</h3>

    <select name="sentiment2" style="font-size: large" required>
      <option value="">(Please select)</option>
      <option>Negative</option>
      <option>Neutral</option>
      <option>Positive</option>
      <option>Text is empty</option>
    </select>
  </div>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

menuju

Teks ditampilkan di bagian atas kotak.

gambar

Sebuah URL untuk gambar yang akan ditampilkan dalam kotak.

Elemen Hierarki

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen Anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-kotak centang

Komponen UI yang dapat dicentang atau dicentang memungkinkan pengguna memilih beberapa opsi dari satu set.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-checkbox>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>

  <p>Find the official website for: <strong>{{ task.input.company }}</strong></p>
  <p>Do not give Yelp pages, LinkedIn pages, etc.</p>
  <p>Include the http:// prefix from the website</p>
  <crowd-input name="website" placeholder="http://example.com"></crowd-input>

  <crowd-checkbox name="website-found">Website Found</crowd-checkbox>

</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

memeriksa

Sebuah saklar Boolean yang, jika ada, menampilkan kotak centang seperti yang dicentang.

Berikut ini adalah contoh syntx yang digunakan untuk mencentang kotak centang secara default.

```
  <crowd-checkbox name="checkedBox" value="checked" checked>This box is checked</crowd-
checkbox>
```

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan kotak centang sebagai dinonaktifkan dan mencegah dari yang dicentang.

Berikut ini adalah contoh sintaksis yang digunakan untuk menonaktifkan kotak centang.

```
  <crowd-checkbox name="disabledCheckBox" value="Disabled" disabled>Cannot be
 selected</crowd-checkbox>
```

nama

String yang digunakan untuk mengidentifikasi jawaban yang diajukan oleh pekerja. Nilai ini akan cocok dengan kunci dalam objek JSON yang menentukan jawabannya.

dibutuhkan

Sebuah saklar Boolean yang, jika ada, membutuhkan pekerja untuk memberikan masukan.

Berikut ini adalah contoh sintaksis yang digunakan untuk memerlukan kotak centang yang dipilih.

```
  <crowd-checkbox name="work_verified" required>Instructions were clear</crowd-
checkbox>
```

nilai

Sebuah string yang digunakan sebagai nama untuk negara kotak centang dalam output. Default untuk “on” jika tidak ditentukan.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak: tidak ada

Output

Menyediakan objek JSON. Parameter`name`string adalah nama objek dan`value`string adalah nama properti untuk nilai Boolean berdasarkan keadaan kotak centang; true jika dicentang, false jika tidak dicentang.

Example : Contoh Elemen Output

Menggunakan yang sama**`name`**nilai untuk beberapa kotak.

```
<!-- INPUT -->
<div><crowd-checkbox name="image_attributes" value="blurry"> Blurry </crowd-checkbox></
div>
<div><crowd-checkbox name="image_attributes" value="dim"> Too Dim </crowd-checkbox></
div>
<div><crowd-checkbox name="image_attributes" value="exposed"> Too Bright </crowd-
checkbox></div>
```

```
//Output with "blurry" and "dim" checked
[
  {
    "image_attributes": {
      "blurry": true,
      "dim": true,
      "exposed": false
    }
  }
]
```

Perhatikan bahwa ketiga nilai warna adalah sifat dari satu objek.

Menggunakan yang berbeda**`name`**nilai untuk setiap kotak.

```
<!-- INPUT -->
<div><crowd-checkbox name="Stop" value="Red"> Red </crowd-checkbox></div>
<div><crowd-checkbox name="Slow" value="Yellow"> Yellow </crowd-checkbox></div>
<div><crowd-checkbox name="Go" value="Green"> Green </crowd-checkbox></div>
```

```
//Output with "Red" checked
[
  {
```

```
    "Go": {
      "Green": false
    },
    "Slow": {
      "Yellow": false
    },
    "Stop": {
      "Red": true
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

klasifikasi kerumunan

Widget untuk mengklasifikasikan konten non-gambar, seperti audio, video, atau teks.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh template tugas pekerja HTML yang dibangun menggunakan`crowd-classifier`. Contoh ini menggunakanBahasa template cairuntuk mengotomatiskan:

- Kategori label di`categories`parameter
- Benda-benda yang sedang diklasifikasikan dalam`classification-target`parameter.

Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
    <crowd-classifier
      name="category"
      categories="{{ task.input.labels | to_json | escape }}"
```

```
      header="What type of a document is this?"
    >
      <classification-target>
        <iframe style="width: 100%; height: 600px;" src="{{ task.input.taskObject  |
 grant_read_access }}" type="application/pdf"></iframe>
      </classification-target>

      <full-instructions header="Document Classification Instructions">
        <p>Read the task carefully and inspect the document.</p>
        <p>Choose the appropriate label that best suits the document.</p>
      </full-instructions>

      <short-instructions>
        Please choose the correct category for the document
      </short-instructions>
    </crowd-classifier>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

kategori

Sebuah array JSON diformat string, masing-masing adalah kategori yang pekerja dapat menetapkan untuk teks. Anda harus menyertakan “lain” sebagai kategori, jika pekerja saya tidak dapat memberikan jawaban.

header

Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nama

Nama widget ini. Hal ini digunakan sebagai kunci untuk input widget dalam output form.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen Induk:Formulir kerumunan
- Elemen Anak:klasifikasi target,instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut didukung oleh elemen ini.

klasifikasi target

Konten yang akan diklasifikasikan oleh pekerja. Ini bisa berupa teks biasa atau HTML. Contoh bagaimana HTML dapat digunakan meliputinamun tidak terbatas padamenyematkan pemutar video atau audio, menyematkan PDF, atau melakukan perbandingan dua atau lebih gambar.

instruksi lengkap

Petunjuk umum tentang bagaimana melakukan klasifikasi teks.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Output

Output dari elemen ini adalah objek menggunakan ditentukan`name`nilai sebagai nama properti, dan string dari`categories`sebagai nilai properti.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
[
  {
    "<name>": {
      "label": "<value>"
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

crowd-classifier-multi-pilih

Widget untuk mengklasifikasikan berbagai bentuk konten—seperti audio, video, atau teks — ke dalam satu atau beberapa kategori. Konten untuk mengklasifikasikan disebut sebagaiobjek.

Lihat contoh interaktif dari template HTML yang menggunakan ini Crowd HTML Element diCodePen.

Berikut ini adalah contoh template tugas pekerja HTML dibangun menggunakan elemen ini. Salin kode berikut dan simpan dalam file dengan ekstensi `.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
    <crowd-classifier-multi-select
      name="category"
      categories="['Positive', 'Negative', 'Neutral']"
      header="Select the relevant categories"
      exclusion-category="{ text: 'None of the above' }"
    >
      <classification-target>
        {{ task.input.taskObject }}
      </classification-target>

      <full-instructions header="Text Categorization Instructions">
        <p><strong>Positive</strong> sentiment include: joy, excitement, delight</p>
        <p><strong>Negative</strong> sentiment include: anger, sarcasm, anxiety</p>
        <p><strong>Neutral</strong>: neither positive or negative, such as stating a
 fact</p>
        <p><strong>N/A</strong>: when the text cannot be understood</p>
        <p>When the sentiment is mixed, such as both joy and sadness, choose both
 labels.</p>
      </full-instructions>

      <short-instructions>
       Choose all categories that are expressed by the text.
      </short-instructions>
    </crowd-classifier-multi-select>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh`crowd-classifier-multi-select`elemen. Setiap atribut menerima nilai string atau nilai string.

kategori

Diperlukan. Sebuah array JSON-diformat string, masing-masing adalah kategori yang pekerja dapat menetapkan ke objek.

header

Diperlukan. Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nama

Diperlukan. Nama widget ini. Dalam output form, nama digunakan sebagai kunci untuk input widget.

kategori eksklusion-

Tidak wajib. Sebuah string JSON-diformat dengan format berikut: `"{ text: 'default-value' }"`. Atribut ini menetapkan nilai default yang dapat dipilih oleh pekerja jika tidak ada label yang berlaku pada objek yang ditampilkan di UI pekerja.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut:

- Elemen induk:Formulir kerumunan
- Elemen Anak:klasifikasi target,instruksi lengkap,instruksi singkat

Wilayah

Elemen ini menggunakan daerah-daerah berikut.

Klasifikasi target

Konten yang akan diklasifikasikan oleh pekerja. Konten dapat berupa teks biasa atau objek yang Anda tentukan dalam template menggunakan HTML. Misalnya, Anda dapat menggunakan elemen

HTML untuk menyertakan pemutar video atau audio, menyematkan file PDF, atau menyertakan perbandingan dua atau lebih gambar.

instruksi lengkap

Petunjuk umum tentang cara mengklasifikasikan teks.

instruksi singkat

Instruksi khusus tugas penting. Instruksi ini ditampilkan secara jelas.

Output

Output dari elemen ini adalah objek yang menggunakan ditentukan`name`nilai sebagai nama properti, dan string dari`categories`sebagai nilai properti.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
[
  {
    "<name>": {
        labels: ["label_a", "label_b"]
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat yang berikut:

- Klasifikasi Teks (Multi-label)
- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-entitas-anotasi

Widget untuk memberi label kata, frasa, atau string karakter dalam teks yang lebih panjang. Pekerja memilih label, dan menyorot teks yang diterapkan label.

> **Penting: Widget mandiri**
>
> Jangan gunakan`<crowd-entity-annotation>`Elemen dengan`<crowd-form>`Elemen. Ini berisi logika formulir pengajuan sendiri danKirimtombol.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat yang menggunakan`<crowd-entity-annotation>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-entity-annotation
  name="crowd-entity-annotation"
  header="Highlight parts of the text below"
  labels="[{'label': 'person', 'shortDisplayName': 'per', 'fullDisplayName': 'Person'},
 {'label': 'date', 'shortDisplayName': 'dat', 'fullDisplayName': 'Date'}, {'label':
 'company', 'shortDisplayName': 'com', 'fullDisplayName': 'Company'}]"
  text="Amazon SageMaker Ground Truth helps you build highly accurate training datasets
 for machine learning quickly."
>
  <full-instructions header="Named entity recognition instructions">
    <ol>
      <li><strong>Read</strong> the text carefully.</li>
      <li><strong>Highlight</strong> words, phrases, or sections of the text.</li>
      <li><strong>Choose</strong> the label that best matches what you have
 highlighted.</li>
      <li>To <strong>change</strong> a label, choose highlighted text and select a new
 label.</li>
      <li>To <strong>remove</strong> a label from highlighted text, choose the X next
 to the abbreviated label name on the highlighted text.</li>
      <li>You can select all of a previously highlighted text, but not a portion of
 it.</li>
    </ol>
  </full-instructions>

  <short-instructions>
    Apply labels to words or phrases.
  </short-instructions>
```

```
    <div id="additionalQuestions" style="margin-top: 20px">
      <h3>
        What is the overall subject of this text?
      </h3>
      <crowd-radio-group>
        <crowd-radio-button name="tech" value="tech">Technology</crowd-radio-button>
        <crowd-radio-button name="politics" value="politics">Politics</crowd-radio-
button>
      </crowd-radio-group>
    </div>
</crowd-entity-annotation>

<script>
  document.addEventListener('all-crowd-elements-ready', () => {
    document
      .querySelector('crowd-entity-annotation')
      .shadowRoot
      .querySelector('crowd-form')
      .form
      .appendChild(additionalQuestions);
  });
</script>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nilai inisial

Sebuah array JSON diformat objek, yang masing-masing mendefinisikan anotasi untuk diterapkan ke teks pada inisialisasi. Objek mengandung`label`nilai yang cocok satu di`labels`atribut, integer`startOffset`nilai untuk offset unicode mulai span berlabel, dan integer`endOffset`nilai untuk offset unicode akhir.

Example

```
[
```

```
  {
    label: 'person',
    startOffset: 0,
    endOffset: 16
  },
  ...
]
```

label

Array JSON diformat, masing-masing berisi:

- **`label`**(wajib): Nama yang digunakan untuk mengidentifikasi entitas.
- **`fullDisplayName`**(opsional) Digunakan untuk daftar label di widget tugas. Default untuk nilai label jika tidak ditentukan.
- **`shortDisplayName`**(opsional) Singkatan dari 3-4 huruf untuk ditampilkan di atas entitas yang dipilih. Default untuk nilai label jika tidak ditentukan.

> **`shortDisplayName`**sangat dianjurkan
>
> Nilai yang ditampilkan di atas pilihan dapat tumpang tindih dan menciptakan kesulitan mengelola entitas berlabel di ruang kerja. Menyediakan karakter 3-4`shortDisplayName`untuk setiap label sangat dianjurkan untuk mencegah tumpang tindih dan menjaga ruang kerja dikelola untuk pekerja Anda.

Example

```
[
  {
    label: 'person',
    shortDisplayName: 'per',
    fullDisplayName: 'person'
  }
]
```

nama

Berfungsi sebagai nama widget di DOM. Hal ini juga digunakan sebagai nama label atribut dalam bentuk output dan output manifest.

teks

Teks yang akan dianotasi. Sistem template lolos dari kutipan dan string HTML secara default. Jika kode Anda sudah lolos atau sebagian lolos, lihatFilter variabeluntuk lebih banyak cara untuk mengontrol melarikan diri.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen anak:instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut didukung oleh elemen ini.

instruksi lengkap

Petunjuk umum tentang cara bekerja dengan widget.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Output

Output berikut didukung oleh elemen ini.

pengada

Sebuah objek JSON yang menentukan awal, akhir, dan label anotasi. Objek ini berisi sifat berikut.

- label— Label yang ditugaskan.
- startOffset— Unicode offset dari awal teks yang dipilih.
- EndOffset— Unicode offset karakter pertama setelah seleksi.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
{
  "myAnnotatedResult": {
```

```
    "entities": [
      {
        "endOffset": 54,
        "label": "person",
        "startOffset": 47
      },
      {
        "endOffset": 97,
        "label": "event",
        "startOffset": 93
      },
      {
        "endOffset": 219,
        "label": "date",
        "startOffset": 212
      },
      {
        "endOffset": 271,
        "label": "location",
        "startOffset": 260
      }
    ]
  }
}
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-fab

Tombol mengambang dengan gambar di tengahnya.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh template Liquid yang dirancang untuk klasifikasi gambar`<crowd-fab>`Elemen. Template ini menggunakan JavaScript untuk memungkinkan pekerja melaporkan masalah dengan UI pekerja. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
    <crowd-image-classifier
        src="${image_url}"
        categories="['Cat', 'Dog', 'Bird', 'None of the Above']"
        header="Choose the correct category for the image"
        name="category">


        <short-instructions>
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.</p>
            <p>If there is an issue with the image or tools, please select
              <b>None of the Above</b>, describe the issue in the text box and click
 the
              button below.</p>
            <crowd-input label="Report an Issue" name="template-issues"></crowd-input>
            <crowd-fab id="button1" icon="report-problem" title="Issue"/>
        </short-instructions>

        <full-instructions header="Classification Instructions">
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.
            Use the <b>None of the Above</b> option if none of the other labels suit
 the image.</p>
        </full-instructions>

    </crowd-image-classifier>
</crowd-form>

<script>
  [
    button1,
  ].forEach(function(button) {
    button.addEventListener('click', function() {
      document.querySelector('crowd-form').submit();
    });
  });
</script>
```

Atribut

Atribut berikut didukung oleh elemen ini.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan tombol mengambang sebagai dinonaktifkan dan mencegah klik.

ikon

String yang menentukan ikon yang akan ditampilkan di tengah tombol. String harus berupa nama ikon dari open-sourcebesi-ikonset, yang sudah dimuat sebelumnya, atau URL ke ikon kustom.

Berikut ini adalah contoh sintaks yang dapat Anda gunakan untuk menambahkan besi-ikon ke`<crowd-fab>`Elemen HTML. Ganti*`icon-name`*dengan nama ikon yang ingin Anda gunakanIkon.

```
<crowd-fab "id="button1" icon="icon-name" title="Issue"/>
```

label

Sebuah string yang terdiri dari satu karakter yang dapat digunakan sebagai pengganti ikon. Emoji atau beberapa karakter dapat mengakibatkan tombol menampilkan elipsis sebagai gantinya.

judul

String yang akan ditampilkan sebagai tip alat saat mouse melayang di atas tombol.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen Anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

Formulir kerumunan

Formulir pembungkus untuk semua tugas kustom. Menetapkan dan menerapkan tindakan penting untuk pengiriman yang tepat dari data formulir Anda.

JikaTombol kerumunantipe “submit” tidak termasuk dalam`<crowd-form>`elemen, secara otomatis akan ditambahkan dalam`<crowd-form>`Elemen.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh klasifikasi gambar yang menggunakan`<crowd-form>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
    <crowd-image-classifier
        src="${image_url}"
        categories="['Cat', 'Dog', 'Bird', 'None of the Above']"
        header="Choose the correct category for the image"
        name="category">


        <short-instructions>
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.</p>
        </short-instructions>


        <full-instructions header="Classification Instructions">
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.
            Use the <b>None of the Above</b> option if none of the other labels suit
 the image.</p>
        </full-instructions>

    </crowd-image-classifier>
</crowd-form>
```

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk: tidak ada
- Elemen anak: Salah satuTemplate UIelemen

Acara Elemen

Parametercrowd-formElemen memperluasHTML standarformelemendan mewarisi peristiwa-peristiwa, sepertionclickdanonsubmit.

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

crowd-icon-tombol

Sebuah tombol dengan gambar ditempatkan di tengah. Ketika pengguna menyentuh tombol, efek riak berasal dari tengah tombol.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh template Liquid yang dirancang untuk klasifikasi gambar<crowd-icon-button>Elemen. Template ini menggunakan JavaScript untuk memungkinkan pekerja melaporkan masalah dengan UI pekerja. Salin kode berikut dan simpan dalam file dengan ekstensi.html. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
    <crowd-image-classifier
        src="${image_url}"
        categories="['Cat', 'Dog', 'Bird', 'None of the Above']"
        header="Choose the correct category for the image"
        name="category">


        <short-instructions>
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.</p>
            <p>If there is an issue with the image or tools, please select
```

```
              <b>None of the Above</b>, describe the issue in the text box and click
 the
              button below.</p>
            <crowd-input label="Report an Issue" name="template-issues"/></crowd-input>
            <crowd-icon-button id="button1" icon="report-problem" title="Issue"/>
        </short-instructions>

        <full-instructions header="Classification Instructions">
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.
            Use the <b>None of the Above</b> option if none of the other labels suit
 the image.</p>
        </full-instructions>

    </crowd-image-classifier>
</crowd-form>

<script>
  [
    button1,
  ].forEach(function(button) {
    button.addEventListener('click', function() {
      document.querySelector('crowd-form').submit();
    });
  });
</script>
```

Atribut

Atribut berikut didukung oleh elemen ini.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan tombol sebagai dinonaktifkan dan mencegah klik.

ikon

String yang menentukan ikon yang akan ditampilkan di tengah tombol. String harus berupa nama ikon dari open-sourcebesi-ikonset, yang sudah dimuat sebelumnya, atau URL ke ikon kustom.

Berikut ini adalah contoh sintaks yang dapat Anda gunakan untuk menambahkan besi-ikon ke`<crowd-icon-button>`Elemen HTML. Ganti*`icon-name`*dengan nama ikon yang ingin Anda gunakan dari ikon iniIkon ditetapkan.

```
<crowd-icon-button id="button1" icon="icon-name" title="Issue"/>
```

Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen Induk:Formulir kerumunan
- Elemen Anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-gambar-classifier

Widget untuk mengklasifikasikan gambar. Gunakan salah satu format gambar yang didukung berikut: APNG, BMP, GIF, ICO, JPEG, PNG, SVG. Gambar tidak memiliki batas ukuran.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh klasifikasi gambar yang menggunakan`<crowd-image-classifier>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
    <crowd-image-classifier
        src="${image_url}"
        categories="['Cat', 'Dog', 'Bird', 'None of the Above']"
        header="Choose the correct category for the image"
        name="category">


        <short-instructions>
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.</p>
```

```
        </short-instructions>


        <full-instructions header="Classification Instructions">
            <p>Read the task carefully and inspect the image.</p>
            <p>Choose the appropriate label that best suits the image.
            Use the <b>None of the Above</b> option if none of the other labels suit
 the image.</p>
        </full-instructions>

    </crowd-image-classifier>
</crowd-form>
```

Atribut

Atribut berikut diperlukan oleh elemen ini.

kategori

Sebuah array JSON diformat string, masing-masing adalah kategori yang pekerja dapat menetapkan untuk gambar. Anda harus menyertakan “lain” sebagai kategori, sehingga pekerja dapat memberikan jawaban. Anda dapat menentukan hingga 10 kategori.

header

Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nama

Nama widget ini. Hal ini digunakan sebagai kunci untuk input widget dalam output form.

hamparan

Informasi yang akan dilapis pada gambar sumber. Ini adalah untuk alur kerja verifikasi pembatas kotak, semantik-segmentasi, dan tugas-tugas instance-segmentasi.

Ini adalah objek JSON yang berisi objek dengan nama tugas-jenis di CamelCase sebagai kunci. Nilai kunci itu adalah objek yang berisi label dan informasi lain yang diperlukan dari tugas sebelumnya.

Contoh`crowd-image-classifier`elemen dengan atribut untuk memverifikasi tugas batas-kotak berikut:

```
<crowd-image-classifier
    name="boundingBoxClassification"
    header="Rate the quality of the annotations based on the background section
       in the instructions on the left hand side."
    src="https://i.imgur.com/CIPKVJo.jpg"
    categories="['good', 'bad', 'okay']"
    overlay='{
        "boundingBox": {
            labels: ["bird", "cat"],
            value: [
                {
                  height: 284,
                  label: "bird",
                  left: 230,
                  top: 974,
                  width: 223
                },
                {
                  height: 69,
                  label: "bird",
                  left: 79,
                  top: 889,
                  width: 247
                }
            ]
        },
    }'
> ... </crowd-image-classifier>
```

Sebuah tugas verifikasi segmentasi semantik akan menggunakan`overlay`sebagai berikut:

```
<crowd-image-classifier
  name='crowd-image-classifier'
  categories='["good", "bad"]'
  src='URL of image to be classified'
  header='Please classify'
  overlay='{
    "semanticSegmentation": {
      "labels": ["Cat", "Dog", "Bird", "Cow"],
      "labelMappings": {
        "Bird": {
          "color": "#ff7f0e"
        },
```

```
        "Cat": {
          "color": "#2ca02c"
        },
        "Cow": {
          "color": "#d62728"
        },
        "Dog": {
          "color": "#2acf59"
        }
      },
      "src": "URL of overlay image",
    }
  }'
> ... </crowd-image-classifier>
```

Sebuah tugas instance-segmentasi akan menggunakan`overlay`sebagai berikut:

```
<crowd-image-classifier
  name='crowd-image-classifier'
  categories='["good", "bad"]'
  src='URL of image to be classified'
  header='Please classify instances of each category'
  overlay='{
    "instanceSegmentation": {
       "labels": ["Cat", "Dog", "Bird", "Cow"],
       "instances": [
        {
         "color": "#2ca02c",
         "label": "Cat"
        },
        {
         "color": "#1f77b4",
         "label": "Cat"
        },
        {
         "color": "#d62728",
         "label": "Dog"
        }
       ],
       "src": "URL of overlay image",
    }
  }'
> ... </crowd-image-classifier>
```

src

URL gambar yang akan diklasifikasikan.

Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen Anak:instruksi lengkap,instruksi singkat,worker-comment

Wilayah

Daerah berikut digunakan oleh elemen ini.

instruksi lengkap

Petunjuk umum untuk pekerja tentang cara mengklasifikasikan gambar.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

worker-comment

Gunakan ini dalam alur kerja verifikasi ketika Anda membutuhkan pekerja untuk menjelaskan mengapa mereka membuat pilihan yang mereka lakukan. Gunakan teks antara tag pembuka dan penutup untuk memberikan instruksi bagi pekerja tentang informasi apa yang harus disertakan dalam komentar.

Ini menggunakan atribut berikut:

header

Sebuah frase dengan ajakan bertindak untuk meninggalkan komentar. Digunakan sebagai teks judul untuk jendela modal di mana komentar ditambahkan.

Tidak wajib. Default untuk “Tambahkan komentar.”

teks-link

Teks ini muncul di bawah kategori di widget. Ketika diklik, membuka jendela modal di mana pekerja dapat menambahkan komentar.

Tidak wajib. Default untuk “Tambahkan komentar.”

placeholder

Contoh teks di area teks komentar yang ditimpa ketika pekerja mulai mengetik. Ini tidak muncul dalam output jika pekerja meninggalkan lapangan kosong.

Tidak wajib. Default ke kosong.

Output

Output dari elemen ini adalah string yang menentukan salah satu nilai yang didefinisikan dalamkategoriatribut <crowd-image-classifier>elemen.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
[
  {
    "<name>": {
      "label": "<value>"
      "workerComment": "Comment - if no comment is provided, this field will not be
 present"
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

crowd-image-classifier-multi-pilih

Widget untuk mengklasifikasikan gambar menjadi satu atau lebih kategori. Gunakan salah satu format gambar yang didukung berikut: APNG, BMP, GIF, ICO, JPEG, PNG, SVG. Gambar tidak memiliki batas ukuran.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh dari template tugas pekerja HTML dibangun menggunakan elemen kerumunan ini. Salin kode berikut dan simpan dalam file dengan ekstensi `.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-image-classifier-multi-select
    name="animals"
    categories="['Cat', 'Dog', 'Horse', 'Pig', 'Bird']"
    src="https://images.unsplash.com/photo-1509205477838-a534e43a849f?
ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1998&q=80"
    header="Please identify the animals in this image"
    exclusion-category="{ text: 'None of the above' }"
  >
    <full-instructions header="Classification Instructions">
      <p>If more than one label applies to the image, select multiple labels.</p>
      <p>If no labels apply, select <b>None of the above</b></p>
    </full-instructions>

    <short-instructions>
      <p>Read the task carefully and inspect the image.</p>
      <p>Choose the appropriate label(s) that best suit the image.</p>
    </short-instructions>
  </crowd-image-classifier-multi-select>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh`crowd-image-classifier-multi-select`Elemen. Setiap atribut menerima nilai string atau nilai string.

kategori

Diperlukan. Sebuah array JSON-diformat string, masing-masing adalah kategori yang pekerja dapat menetapkan untuk gambar. Seorang pekerja harus memilih setidaknya satu kategori dan dapat memilih semua kategori.

header

Diperlukan. Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nama

Diperlukan. Nama widget ini. Dalam output form, nama digunakan sebagai kunci untuk input widget.

src

Diperlukan. URL gambar yang akan diklasifikasikan.

kategori eksklusion-

Tidak wajib. Sebuah string JSON-diformat dengan format berikut: `"{ text: 'default-value' }"`. Atribut ini menetapkan nilai default yang dapat dipilih oleh pekerja jika tidak ada label yang berlaku pada gambar yang ditampilkan di UI pekerja.

Elemen Hirarki

Elemen ini memiliki elemen induk dan anak berikut:

- Elemen induk:Formulir kerumunan
- Elemen anak:instruksi lengkap,instruksi singkat,worker-comment

Wilayah

Elemen ini menggunakan daerah berikut

instruksi lengkap

Petunjuk umum untuk pekerja tentang cara mengklasifikasikan gambar.

instruksi singkat

Instruksi khusus tugas penting. Instruksi ini ditampilkan secara jelas.

Output

Output dari elemen ini adalah string yang menentukan satu atau lebih dari nilai-nilai yang didefinisikan dalam`categories`Atribut`<crowd-image-classifier-multi-select>`Elemen.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
[
```

```
  {
    "<name>": {
        labels: ["label_a", "label_b"]
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat yang berikut:

- Klasifikasi Gambar (Multi-label)
- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

Masukan kerumunan

Sebuah kotak yang menerima input data.

> **Tidak dapat menutup diri**
>
> Berbeda dengan`input`elemen dalam standar HTML, elemen ini tidak dapat ditutup sendiri dengan menempatkan garis miring sebelum braket akhir, misalnya`<crowd-input ... />`. Ini harus diikuti dengan`</crowd-input>`untuk menutup elemen.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-input>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <img style="max-width: 35vw; max-height: 50vh" src="{{ task.input.taskObject |
 grant_read_access }}">
  <crowd-input name="tag1" label="Word/phrase 1" required></crowd-input>
  <crowd-input name="tag2" label="Word/phrase 2" required></crowd-input>
  <crowd-input name="tag3" label="Word/phrase 3" required></crowd-input>
```

```
  <short-instructions>
    Your custom quick instructions and examples
  </short-instructions>

  <full-instructions>
    Your custom detailed instracutions and more examples
  </full-instructions>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

pola diperbolehkankan

Ekspresi reguler yang digunakan denganvalidasi otomatisatribut untuk mengabaikan karakter non-pencocokan sebagai jenis pekerja.

fokus otomatis

Ketika nilai diatur ke betul, browser menempatkan fokus di dalam area input setelah loading. Dengan cara ini, pekerja dapat mulai mengetik tanpa harus memilihnya terlebih dahulu.

validasi otomatis

Sebuah saklar Boolean yang, jika ada, menyalakan validasi input. Perilaku validator dapat dimodifikasi olehpesan kesalahandanpola diperbolehkankanatribut.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan area input sebagai dinonaktifkan.

pesan kesalahan

Teks yang akan ditampilkan di bawah bidang input, di sisi kiri, jika validasi gagal.

label

Sebuah string yang ditampilkan di dalam bidang teks.

Teks ini menyusut dan naik di atas bidang teks ketika pekerja mulai mengetik di lapangan atau ketikanilaiatribut diatur.

panjang maks

Jumlah maksimum karakter input akan menerima. Masukan di luar batas ini diabaikan.

panjang min

Panjang minimum untuk input di lapangan

nama

Menetapkan nama input yang akan digunakan dalam DOM dan output dari bentuk.

placeholder

Sebuah nilai string yang digunakan sebagai teks placeholder, ditampilkan sampai pekerja mulai memasukkan data ke dalam input, Hal ini tidak digunakan sebagai nilai default.

dibutuhkan

Sebuah saklar Boolean yang, jika ada, membutuhkan pekerja untuk memberikan masukan.

jenis

Membawa string untuk mengatur HTML5`input-type`perilaku untuk input. Contohnya termasuk `file` dan `date`.

nilai

Sebuah preset yang menjadi default jika pekerja tidak memberikan masukan. Preset muncul di bidang teks.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen Induk:Formulir kerumunan
- Elemen Anak: tidak ada

Output

Menyediakan`name`string sebagai nama properti, dan teks yang dimasukkan di lapangan sebagai nilainya.

Example Sampel: Output JSON

Nilai-nilai untuk beberapa elemen adalah output dalam objek yang sama, dengan merekanamenilai atribut sebagai nama properti mereka. Elemen tanpa input tidak muncul dalam output. Sebagai contoh, mari kita gunakan tiga input:

```
<crowd-input name="tag1" label="Word/phrase 1"></crowd-input>
<crowd-input name="tag2" label="Word/phrase 2"></crowd-input>
<crowd-input name="tag3" label="Word/phrase 3"></crowd-input>
```

Ini adalah output jika hanya dua memiliki masukan:

```
[
  {
    "tag1": "blue",
    "tag2": "red"
  }
]
```

Ini berarti kode apa pun yang dibuat untuk mengurai hasil ini harus mampu menangani ada tidaknya setiap masukan dalam jawaban.

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-instance-segmentasi

Widget untuk mengidentifikasi contoh individu objek tertentu dalam gambar dan menciptakan hamparan berwarna untuk setiap contoh berlabel.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-instance-segmentation>`. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
```

```
<crowd-form>
  <crowd-instance-segmentation
    name="annotatedResult"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Please label each of the requested objects in this image"
    labels="['Cat', 'Dog', 'Bird']"
  >
    <full-instructions header="Segmentation Instructions">
      <ol>
          <li><strong>Read</strong> the task carefully and inspect the image.</li>
          <li><strong>Read</strong> the options and review the examples provided to
 understand more about the labels.</li>
          <li><strong>Choose</strong> the appropriate label that best suits the
 image.</li>
      </ol>
    </full-instructions>

    <short-instructions>
      <p>Use the tools to label all instances of the requested items in the image</p>
    </short-instructions>
  </crowd-instance-segmentation>
</crowd-form>
```

Gunakan template yang mirip dengan berikut ini untuk memungkinkan pekerja menambahkan kategori mereka sendiri (label).

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-instance-segmentation
    id="annotator"
    name="myTexts"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Click Instructions to add new labels."
    labels="['placeholder']"
  >
    <short-instructions>
      <h3>Add a label to describe each type of object in this image.</h3>
      <h3>Cover each instance of each object with a segmentation mask.</h3>
      <br>
      <h3>
        Add new label
      </h3>
```

```
      <crowd-input name="_customLabel" id="customLabel"></crowd-input>
      <crowd-button id="addLabel">Add</crowd-button>

      <br><br><br>
      <h3>
      Manage labels
      </h3>
      <div id="labelsSection"></div>
    </short-instructions>

    <full-instructions>
      Describe your task in more detail here.
    </full-instructions>
  </crowd-instance-segmentation>
</crowd-form>

<script>
  document.addEventListener('all-crowd-elements-ready', function(event) {
    document.querySelector('crowd-instance-segmentation').labels = [];
  });

  function populateLabelsSection() {
    labelsSection.innerHTML = '';
    annotator.labels.forEach(function(label) {
      const labelContainer = document.createElement('div');
      labelContainer.innerHTML = label + ' <a href="javascript:void(0)">(Delete)</a>';
      labelContainer.querySelector('a').onclick = function() {
        annotator.labels = annotator.labels.filter(function(l) {
          return l !== label;
        });
        populateLabelsSection();
      };
      labelsSection.appendChild(labelContainer);
    });
  }

  addLabel.onclick = function() {
    annotator.labels = annotator.labels.concat([customLabel.value]);
    customLabel.value = null;

    populateLabelsSection();
  };
</script>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

label

Sebuah array JSON diformat string, yang masing-masing adalah label yang pekerja dapat menetapkan ke instance dari objek dalam gambar. Pekerja dapat menghasilkan warna overlay yang berbeda untuk setiap instance yang relevan dengan memilih “add instance” di bawah label di tool.

nama

Nama widget ini. Hal ini digunakan sebagai kunci untuk data pelabelan dalam output bentuk.

src

URL gambar yang akan diberi label.

nilai awal

Objek JSON yang berisi pemetaan warna pekerjaan segmentasi instance sebelumnya dan tautan ke output gambar overlay oleh pekerjaan sebelumnya. Sertakan ini ketika Anda ingin pekerja manusia untuk memverifikasi hasil pekerjaan pelabelan sebelumnya dan menyesuaikannya jika perlu.

Atribut akan muncul sebagai berikut:

```
initial-value="{
  "instances": [
    {
      "color": "#2ca02c",
      "label": "Cat"
    },
    {
      "color": "#1f77b4",
      "label": "Cat"
    },
    {
```

```
        "color": "#d62728",
        "label": "Dog"
      }
    ],
    "src": {{ "S3 file URL for image" | grant_read_access }}
  }"
```

Hirarki

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen Induk:Formulir kerumunan
- Elemen Anak:instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut didukung oleh elemen ini.

instruksi lengkap

Petunjuk umum tentang bagaimana melakukan segmentasi gambar.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Output

Output berikut didukung oleh elemen ini.

labeledImage

Objek JSON yang berisi PNG yang dikodekan Base64 dari label.

Instans

Sebuah JSON Array berisi objek dengan label instance dan warna.

- warna- Nilai heksadesimal dari warna RGB label di`labeledImage`PNG.
- label— Label yang diberikan untuk overlay (s) menggunakan warna itu. Nilai ini mungkin berulang, karena contoh label yang berbeda diidentifikasi oleh warna unik mereka.

InputImageProperties

Sebuah objek JSON yang menentukan dimensi gambar yang sedang dianotasi oleh pekerja. Objek ini berisi properti berikut.

- tingginya— Tinggi, dalam piksel, gambar.
- Lebar— Lebar, dalam piksel, gambar.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
[
  {
    "annotatedResult": {
      "inputImageProperties": {
        "height": 533,
        "width": 800
      },
      "instances": [
        {
          "color": "#1f77b4",
          "label": "<Label 1>":
        },
        {
          "color": "#2ca02c",
          "label": "<Label 1>":
        },
        {
          "color": "#ff7f0e",
          "label": "<Label 3>":
        },
      ],
      "labeledImage": {
        "pngImageData": "<Base-64 Encoded Data>"
      }
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

Instruksi kerumunan

Sebuah elemen yang menampilkan petunjuk pada tiga halaman tab,Ringkasan,Petunjuk terperinci, danContoh, ketika pekerja mengklik link atau tombol.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-instructions>`elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
   <crowd-instructions link-text="View instructions" link-type="button">
    <short-summary>
      <p>Given an image, write three words or short phrases that summarize its
 contents.</p>
    </short-summary>
    <detailed-instructions>
      <p>Imagine that you are describing an image to a friend or tagging it for a news
 website. Provide three specific words or short phrases that describe it.</p>
    </detailed-instructions>
    <positive-example>
      <p><img src="https://s3.amazonaws.com/cv-demo-images/highway.jpg"/></p>
      <p>
       <ul>
        <li>Highway</li>
        <li>Cars</li>
        <li>Gas station</li>
       </ul>
      </p>
    </positive-example>
    <negative-example>
```

```
      <p><img src="https://s3.amazonaws.com/cv-demo-images/highway.jpg"/></p>
      <p>
       These are not specific enough:
       <ol>
        <li>Trees</li>
        <li>Outside</li>
        <li>Daytime</li>
       </ol>
      </p>
    </negative-example>
 </crowd-instructions>
    <p><strong>Instructions: </strong>Given an image, write three words or short
 phrases that summarize its contents.</p>
    <p>If someone were to see these three words or phrases, they should understand the
 subject and context of the image, as well as any important actions.</p>
 <p>View the instructions for detailed instructions and examples.</p>
 <p><img style="max-width: 100%; max-height: 100%" src="{{ task.input.taskObject |
 grant_read_access }}"></p>
  <crowd-input name="tag1" label="Word/phrase 1" required></crowd-input>
  <crowd-input name="tag2" label="Word/phrase 2" required></crowd-input>
  <crowd-input name="tag3" label="Word/phrase 3" required></crowd-input>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

teks-tautan

Teks yang akan ditampilkan untuk membuka instruksi. Default-nya adalahPetunjuk klik.

link-jenis

Sebuah string yang menentukan jenis pemicu untuk petunjuk. Nilai yang mungkin adalah “link” (default) dan “tombol”.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak: tidak ada

Wilayah

Daerah berikut didukung oleh elemen ini.

instruksi detail

Konten yang menyediakan instruksi khusus untuk tugas. Ini muncul di halaman tab “Instruksi Terperinci”.

negatif-contoh

Konten yang memberikan contoh penyelesaian tugas yang tidak memadai. Ini muncul di halaman tab “Contoh”. Lebih dari satu contoh dapat diberikan dalam elemen ini.

contoh positif

Konten yang memberikan contoh penyelesaian tugas yang tepat. Ini muncul di halaman tab “Contoh”.

ringkasan singkat

Sebuah pernyataan singkat yang merangkum tugas yang harus diselesaikan. Ini muncul di halaman tab “Ringkasan”. Lebih dari satu contoh dapat diberikan dalam elemen ini.

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-keypoint

Menghasilkan alat untuk memilih dan memberi anotasi poin kunci pada gambar.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-keypoint>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
```

```
  <div id="errorBox"></div>

  <crowd-keypoint
    src="{{ task.input.taskObject | grant_read_access }}"
    labels="['Item A', 'Item B', 'Item C']"
    header="Please locate the centers of each item."
    name="annotatedResult">
    <short-instructions>
      Describe your task briefly here and give examples
    </short-instructions>
    <full-instructions>
      Give additional instructions and good/bad examples here
    </full-instructions>
  </crowd-keypoint>
</crowd-form>

<script>
  var num_obj = 1;

  document.querySelector('crowd-form').onsubmit = function(e) {
    const keypoints = document.querySelector('crowd-keypoint').value.keypoints ||
 document.querySelector('crowd-keypoint')._submittableValue.keypoints;
    const labels = keypoints.map(function(p) {
      return p.label;
    });

    // 1. Make sure total number of keypoints is correct.
    var original_num_labels = document.getElementsByTagName("crowd-keypoint")
[0].getAttribute("labels");

    original_num_labels = original_num_labels.substring(2, original_num_labels.length -
 2).split("\",\"");
    var goalNumKeypoints = num_obj*original_num_labels.length;
    if (keypoints.length != goalNumKeypoints) {
      e.preventDefault();
      errorBox.innerHTML = '<crowd-alert type="error" dismissible>You must add all
 keypoint annotations and use each label only once.</crowd-alert>';
      errorBox.scrollIntoView();
      return;
    }

    // 2. Make sure all labels are unique.
    labelCounts = {};
    for (var i = 0; i < labels.length; i++) {
```

```
      if (!labelCounts[labels[i]]) {
        labelCounts[labels[i]] = 0;
      }
      labelCounts[labels[i]]++;
    }
    const goalNumSingleLabel = num_obj;

    const numLabels = Object.keys(labelCounts).length;

    Object.entries(labelCounts).forEach(entry => {
      if (entry[1] != goalNumSingleLabel) {
        e.preventDefault();
        errorBox.innerHTML = '<crowd-alert type="error" dismissible>You must use each
 label only once.</crowd-alert>';
        errorBox.scrollIntoView();
      }
    })
  };
</script>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nilai awal

Array, dalam format JSON, dari keypoints yang akan diterapkan pada gambar pada awal. Misalnya:

```
initial-value="[
  {
    'label': 'Left Eye',
    'x': 1022,
    'y': 429
  },
  {
    'label': 'Beak',
    'x': 941,
    'y': 403
  }
```

```
]
```

> **Note**
>
> Harap dicatat bahwa nilai label yang digunakan dalam atribut ini harus memiliki nilai yang cocok di`labels`atribut atau titik tidak akan diberikan.

label

Array, dalam format JSON, string yang akan digunakan sebagai label anotasi keypoint.

nama

Sebuah string yang digunakan untuk mengidentifikasi jawaban yang disampaikan oleh pekerja. Nilai ini akan cocok dengan kunci dalam objek JSON yang menentukan jawabannya.

src

Sumber URI gambar yang akan dianotasi.

Elemen Hierarki

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen Anak:instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut diperlukan oleh elemen ini.

instruksi lengkap

Petunjuk umum tentang cara membuat anotasi gambar.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Output

Output berikut didukung oleh elemen ini.

InputImageProperties

Sebuah objek JSON yang menentukan dimensi gambar yang sedang dianotasi oleh pekerja. Objek ini berisi properti berikut.

- tingginya— Tinggi, dalam piksel, gambar.
- lebar— Lebar, dalam piksel, gambar.

keypoint

Array objek JSON yang berisi koordinat dan label keypoint. Setiap objek berisi properti berikut.

- label— Label yang ditugaskan untuk keypoint.
- x— Koordinat X, dalam piksel, dari keypoint pada gambar.
- y— Koordinat Y, dalam piksel, dari keypoint pada gambar.

> **Note**
>
> Koordinat X dan Y didasarkan pada 0,0 menjadi sudut kiri atas gambar.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari menggunakan elemen ini.

```
[
  {
    "crowdKeypoint": {
      "inputImageProperties": {
        "height": 1314,
        "width": 962
      },
      "keypoints": [
        {
          "label": "dog",
          "x": 155,
          "y": 275
        },
        {
          "label": "cat",
```

```
          "x": 341,
          "y": 447
        },
        {
          "label": "cat",
          "x": 491,
          "y": 513
        },
        {
          "label": "dog",
          "x": 714,
          "y": 578
        },
        {
          "label": "cat",
          "x": 712,
          "y": 763
        },
        {
          "label": "cat",
          "x": 397,
          "y": 814
        }
      ]
    }
  }
]
```

Anda mungkin memiliki banyak label yang tersedia, tetapi hanya yang digunakan muncul dalam output.

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

garis kerumunan

Widget untuk menggambar garis pada gambar. Setiap baris dikaitkan dengan label, dan data output akan melaporkan titik awal dan akhir dari setiap baris.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-line>`elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini. Untuk contoh lainnya, lihat iniRepositori GitHub.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-line
    name="crowdLine"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Add header here to describe the task"
    labels="['car','pedestrian','street car']"
  >
    <short-instructions>
        <p>Read the task carefully and inspect the image.</p>
        <p>Choose the appropriate label that best suits the image.</p>
        <p>Draw a line on each objects that the label applies to.</p>
    </short-instructions>

    <full-instructions>
        <p>Read the task carefully and inspect the image.</p>
        <p>Choose the appropriate label that best suits the image.
        <p>Draw a line along each object that the image applies to.
            Make sure that the line does not extend beyond the boundaries
            of the object.
        </p>
        <p>Each line is defined by a starting and ending point. Carefully
        place the starting and ending points on the boundaries of the object.</p>
    </full-instructions>

  </crowd-line>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Tidak wajib. Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nilai-awal

Tidak wajib. Array objek JSON, yang masing-masing menetapkan garis ketika komponen dimuat. Setiap objek JSON dalam array berisi properti berikut:

- label— Teks yang ditugaskan ke baris sebagai bagian dari tugas pelabelan. Teks ini harus sesuai dengan salah satu label yang didefinisikan dalamlabelAtribut<crowd-line>elemen.
- simpul— yangxdanypixel corrdinates dari titik awal dan titik akhir dari garis, relatif terhadap sudut kiri atas gambar.

```
initial-value="{
    lines: [
    {
        label: 'sideline', // label of this line annotation
        vertices:[         // an array of vertices which decide the position of the
 line
        {
            x: 84,
            y: 110
        },
        {
            x: 60,
            y: 100
        }
        ]
    },
    {
        label: 'yardline',
        vertices:[
        {
            x: 651,
            y: 498
        },
        {
            x: 862,
            y: 869
        }
        ]
    }
   ]
}"
```

Garis diatur melalui`initial-value`properti dapat disesuaikan. Apakah jawaban pekerja disesuaikan atau tidak dilacak melalui`initialValueModified`boolean dalam output jawaban pekerja.

label

Diperlukan. Sebuah array JSON diformat string, yang masing-masing adalah label yang pekerja dapat menetapkan ke baris.

Batas:label 10

label-warna

Tidak wajib. Susunan rangkaian. Setiap string adalah heksadesimal (hex) kode untuk label.

nama

Diperlukan. Nama widget ini. Ini digunakan sebagai kunci untuk input widget dalam output form.

src

Diperlukan. URL gambar yang menggambar garis.

Wilayah

Daerah berikut diperlukan oleh elemen ini.

instruksi lengkap

Petunjuk umum tentang cara menggambar garis.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak:instruksi singkat,instruksi lengkap

Output

InputImageProperties

Sebuah objek JSON yang menentukan dimensi gambar yang sedang dianotasi oleh pekerja. Objek ini berisi properti berikut.

- tingginya— Tinggi, dalam piksel, gambar.
- lebar— Lebar, dalam piksel, gambar.

lini

Sebuah JSON Array berisi objek dengan label garis dan simpul.

- label— Label yang diberikan ke garis.
- simpul — yangxdanypixel corrdinates dari titik awal dan titik akhir dari garis, relatif terhadap sudut kiri atas gambar.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
{
    "crowdLine": { //This is the name you set for the crowd-line
      "inputImageProperties": {
        "height": 1254,
        "width": 2048
      },
      "lines": [
        {
          "label": "yardline",
          "vertices": [
            {
              "x": 58,
              "y": 295
            },
            {
              "x": 1342,
              "y": 398
            }
          ]
```

```
        },
        {
          "label": "sideline",
          "vertices": [
            {
              "x": 472,
              "y": 910
            },
            {
              "x": 1480,
              "y": 600
            }
          ]
        }
      ]
    }
  }
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-modal

Sebuah jendela kecil yang muncul di layar ketika dibuka.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh sintaksis yang dapat Anda gunakan dengan`<crowd-modal>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-modal
link-text = "See Examples"
link-type = "button">
Example Modal Text</crowd-modal>
```

Atribut

Atribut berikut didukung oleh elemen ini.

teks-link

Teks yang akan ditampilkan untuk membuka modal. Defaultnya adalah “Klik untuk membuka modal”.

link-jenis

Sebuah string yang menentukan jenis pemicu untuk modal. Nilai yang mungkin adalah “link” (default) dan “tombol”.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

poligon

Widget untuk menggambar poligon pada gambar dan menetapkan label ke bagian gambar yang tertutup di setiap poligon.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-polygon>`elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
```

```
  <crowd-polygon
    name="annotatedResult"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Draw a polygon around each of the requested target(s) of interest"
    labels="['Cat', 'Dog', 'Bird']"
  >
    <full-instructions header="Polygon instructions">
      <ul>
        <li>Make the polygon tight around the object</li>
        <li>You need to select a label before starting a polygon</li>
        <li>You will need to select a label again after completing a polygon</li>
        <li>To select a polygon, you can click on its borders</li>
        <li>You can start drawing a polygon from inside another polygon</li>
        <li>You can undo and redo while you're drawing a polygon to go back and forth
 between points you've placed</li>
        <li>You are prevented from drawing lines that overlap other lines from the same
 polygon</li>
      </ul>
    </full-instructions>

    <short-instructions>
      <p>Draw a polygon around each of the requested target(s) of interest</p>
      <p>Make the polygon tight around the object</p>
    </short-instructions>
  </crowd-polygon>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Teks yang akan ditampilkan di atas citra. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

label

Sebuah array JSON diformat string, yang masing-masing adalah label yang pekerja dapat menetapkan untuk bagian gambar tertutup oleh poligon.

nama

Nama widget ini. Ini digunakan sebagai kunci untuk input widget dalam output form.

src

URL gambar yang menggambar poligon.

nilai-awal

Array objek JSON, yang masing-masing mendefinisikan poligon yang akan ditarik ketika komponen dimuat. Setiap objek JSON dalam array berisi properti berikut.

- label— Teks yang ditugaskan ke poligon sebagai bagian dari tugas pelabelan. Teks ini harus sesuai dengan salah satu label yang didefinisikan dalamlabelatribut <crowd-polygon>elemen.
- simpul— Susunan rangkaian. Setiap objek berisi nilai koordinat x dan y untuk titik dalam poligon.

Example

Sesi`initial-value`Atribut mungkin terlihat seperti ini.

```
initial-value =
  '[
     {
     "label": "dog",
     "vertices":
       [
         {
            "x": 570,
            "y": 239
         },
        ...
         {
            "x": 759,
            "y": 281
         }
       ]
     }
  ]'
```

Karena ini akan berada dalam elemen HTML, array JSON harus tertutup dalam tanda kutip tunggal atau ganda. Contoh di atas menggunakan tanda kutip tunggal untuk merangkum JSON dan tanda kutip ganda dalam JSON itu sendiri. Jika Anda harus mencampur tanda kutip tunggal dan ganda di dalam JSON Anda, ganti dengan kode entitas HTML mereka (`"`untuk kutipan ganda,`'`untuk single) untuk melarikan diri dengan aman.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak:instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut diperlukan.

instruksi lengkap

Petunjuk umum tentang cara menggambar poligon.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Output

Output berikut didukung oleh elemen ini.

poligon

Array objek JSON, yang masing-masing menggambarkan poligon yang telah dibuat oleh pekerja. Setiap objek JSON dalam array berisi properti berikut.

- label— Teks yang ditugaskan ke poligon sebagai bagian dari tugas pelabelan.
- simpul— Susunan rangkaian. Setiap objek berisi nilai koordinat x dan y untuk titik dalam poligon. Sudut kiri atas citra 0,0.

InputImageProperties

Sebuah objek JSON yang menentukan dimensi gambar yang sedang dianotasi oleh pekerja. Objek ini berisi properti berikut.

- tingginya— Tinggi, dalam piksel, gambar.
- lebar— Lebar, dalam piksel, gambar.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari skenario penggunaan umum untuk elemen ini.

Label Tunggal, Poligon Tunggal

```
{
    "annotatedResult":
    {
      "inputImageProperties": {
        "height": 853,
        "width": 1280
      },
      "polygons":
      [
        {
          "label": "dog",
          "vertices":
          [
            {
              "x": 570,
              "y": 239
            },
            {
              "x": 603,
              "y": 513
            },
            {
              "x": 823,
              "y": 645
            },
            {
              "x": 901,
              "y": 417
            },
            {
              "x": 759,
              "y": 281
            }
          ]
        }
      ]
    }
  }
```

```
]
```

Label Tunggal, Beberapa Poligon

```
[
  {
    "annotatedResult": {
      "inputImageProperties": {
        "height": 853,
        "width": 1280
      },
      "polygons": [
        {
          "label": "dog",
          "vertices": [
            {
              "x": 570,
              "y": 239
            },
            {
              "x": 603,
              "y": 513
            },
            {
              "x": 823,
              "y": 645
            },
            {
              "x": 901,
              "y": 417
            },
            {
              "x": 759,
              "y": 281
            }
          ]
        },
        {
          "label": "dog",
          "vertices": [
            {
              "x": 870,
              "y": 278
```

```
            },
            {
              "x": 908,
              "y": 446
            },
            {
              "x": 1009,
              "y": 602
            },
            {
              "x": 1116,
              "y": 519
            },
            {
              "x": 1174,
              "y": 498
            },
            {
              "x": 1227,
              "y": 479
            },
            {
              "x": 1179,
              "y": 405
            },
            {
              "x": 1179,
              "y": 337
            }
          ]
        }
      ]
    }
  }
]
```

Beberapa Label, Beberapa Poligon

```
[
  {
    "annotatedResult": {
      "inputImageProperties": {
        "height": 853,
```

```
        "width": 1280
      },
      "polygons": [
        {
          "label": "dog",
          "vertices": [
            {
              "x": 570,
              "y": 239
            },
            {
              "x": 603,
              "y": 513
            },
            {
              "x": 823,
              "y": 645
            },
            {
              "x": 901,
              "y": 417
            },
            {
              "x": 759,
              "y": 281
            }
          ]
        },
        {
          "label": "cat",
          "vertices": [
            {
              "x": 870,
              "y": 278
            },
            {
              "x": 908,
              "y": 446
            },
            {
              "x": 1009,
              "y": 602
            },
            {
```

```
              "x": 1116,
              "y": 519
            },
            {
              "x": 1174,
              "y": 498
            },
            {
              "x": 1227,
              "y": 479
            },
            {
              "x": 1179,
              "y": 405
            },
            {
              "x": 1179,
              "y": 337
            }
          ]
        }
      ]
    }
  }
]
```

Anda bisa memiliki banyak label yang tersedia, tetapi hanya yang digunakan muncul dalam output.

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-polyline

Widget untuk menggambar polylines atau garis pada gambar. Setiap polyline dikaitkan dengan label dan dapat mencakup dua atau lebih simpul. Sebuah polyline dapat memotong dirinya sendiri dan titik awal dan akhir dapat ditempatkan di mana saja pada gambar.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat yang menggunakan`<crowd-polyline>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini. Untuk contoh lainnya, lihat iniRepositori GitHub.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-polyline
    name="crowdPolyline"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Add header here to describe the task"
    labels="['car','pedestrian','street car']"
  >
    <full-instructions>
        <p>Read the task carefully and inspect the image.</p>
        <p>Choose the appropriate label that best suits the image.</p>
        <p>Draw a polyline around the boundaries of all objects
        that the label applies to.</p>
        <p>Use the <b>Enter</b> key to complete a polyline.</p>
        <p>Make sure that the polyline fits tightly around the boundary
        of the object.</p>
    </full-instructions>

    <short-instructions>
        <p>Read the task carefully and inspect the image.</p>
        <p>Review the tool guide to learn how to use the polyline tool.</p>
        <p>Choose the appropriate label that best suits the image.</p>
        <p>To draw a polyline, select a label that applies to an object of interest
            and add a single point to the photo by clicking on that point. Continue to
            draw the polyline around the object by adding additional points
            around the object boundary.</p>
        <p>After you place the final point on the polyline, press <b>Enter</b> on your
        keyboard to complete the polyline.</p>

    </short-instructions>
  </crowd-polyline>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Tidak wajib. Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nilai awal

Tidak wajib. Array objek JSON, yang masing-masing menetapkan polyline ketika komponen dimuat. Setiap objek JSON dalam array berisi properti berikut:

- label— Teks yang ditugaskan ke polyline sebagai bagian dari tugas pelabelan. Teks ini harus sesuai dengan salah satu label yang didefinisikan dalamlabelAtribut`<crowd-polyline>`Elemen.
- simpul— yang`x`dany`y`kordinat piksel dari simpul polyline, relatif terhadap sudut kiri atas gambar.

```
initial-value= "{
   polylines: [
   {
       label: 'sideline', // label of this line annotation
       vertices:[         // an array of vertices which decide the position of the
line
       {
           x: 84,
           y: 110
       },
       {
           x: 60,
           y: 100
       }
       ]
   },
   {
       label: 'yardline',
       vertices:[
       {
           x: 651,
           y: 498
       },
       {
           x: 862,
           y: 869
       },
       {
```

```
            x: 1000,
            y: 869
        }
        ]
    }
   ]
}"
```

Polylines diatur melaluiinitial-valueproperti dapat disesuaikan. Apakah jawaban pekerja disesuaikan atau tidak dilacak melaluiinitialValueModifiedboolean dalam output jawaban pekerja.

label

Diperlukan. Sebuah array JSON diformat string, yang masing-masing adalah label yang pekerja dapat menetapkan ke baris.

Batas:Label 10

label

Tidak wajib. Susunan rangkaian. Setiap string adalah heksadesimal (hex) kode untuk label.

nama

Diperlukan. Nama widget ini. Ini digunakan sebagai kunci untuk input widget dalam output form.

src

Diperlukan. URL gambar yang menggambar polylines.

Wilayah

Daerah berikut diperlukan oleh elemen ini.

instruksi lengkap

Petunjuk umum tentang cara menggambar polylines.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Hirarki

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak:instruksi singkat,instruksi lengkap

Output

InputImageProperties

Sebuah objek JSON yang menentukan dimensi gambar yang sedang dianotasi oleh pekerja. Objek ini berisi properti berikut.

- tingginya— Tinggi, dalam piksel, gambar.
- Lebar— Lebar, dalam piksel, gambar.

polylines

JSON Array berisi objek dengan label polylines dan simpul.

- label— Label yang diberikan ke garis.
- simpul — yangxdanykordinat piksel dari simpul polyline, relatif terhadap sudut kiri atas gambar.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
{
   "crowdPolyline": { //This is the name you set for the crowd-polyline
     "inputImageProperties": {
       "height": 1254,
       "width": 2048
     },
     "polylines": [
       {
         "label": "sideline",
         "vertices": [
           {
             "x": 651,
             "y": 498
           },
           {
             "x": 862,
```

```
              "y": 869
            },
            {
              "x": 1449,
              "y": 611
            }
          ]
        },
        {
          "label": "yardline",
          "vertices": [
            {
              "x": 1148,
              "y": 322
            },
            {
              "x": 1705,
              "y": 474
            },
            ,
            {
              "x": 1755,
              "y": 474
            }
          ]
        }
      ]
    }
  }
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

tombol radio kerumunan

Sebuah tombol yang dapat dicentang atau dicentang. Ketika tombol radio berada di dalam grup radio, tepat satu tombol radio dalam grup dapat diperiksa kapan saja. Berikut ini adalah contoh cara mengkonfigurasi`crowd-radio-button`elemen dalam`crowd-radio-group`Elemen.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh sintaksis yang dapat Anda gunakan dengan`<crowd-radio-button>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
<crowd-radio-group>
    <crowd-radio-button name="tech" value="tech">Technology</crowd-radio-button>
    <crowd-radio-button name="politics" value="politics">Politics</crowd-radio-button>
</crowd-radio-group>
</crowd-form>
```

Contoh sebelumnya dapat dilihat dalam template tugas pekerja kustom dalam contoh GitHub ini:Templat Kustom Pekerjaan Pelabelan Pengakuan Entitas.

Tombol radio Crowd HTML Element tidak mendukung tag HTML,`required`. Untuk membuat pemilihan tombol radio diperlukan, gunakan`<input type="radio">`elemen untuk membuat tombol radio dan menambahkan`required`Tanda. Parameter`name`atribut untuk semua`<input>`elemen yang termasuk dalam kelompok tombol radio yang sama harus sama. Misalnya, template berikut mengharuskan pengguna untuk memilih tombol radio di`animal-type`kelompok sebelum mengirimkan.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <p>Select an animal type:</p>
<img src="https://images.unsplash.com/photo-1537151608828-ea2b11777ee8?
ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1539&q=80"
 style="height: 500; width: 400;"/>
<br><br>
<div>
  <input type="radio" id="cat" name="animal-type" value="cat" required>
  <label for="cat">Cat</label>
</div>
<div>
  <input type="radio" id="dog" name="animal-type" value="dog">
  <label for="dog">Dog</label>
</div>
<div>
  <input type="radio" id="unknown" name="animal-type" value="unknown">
  <label for="unknown">Unknown</label>
```

```
</div>
    <full-instructions header="Classification Instructions">
      <p>Read the task carefully and inspect the image.</p>
      <p>Choose the appropriate label that best suits the image.</p>
    </full-instructions>
    <short-instructions>
      <p>Read the task carefully and inspect the image.</p>
      <p>Choose the appropriate label that best suits the image.</p>
    </short-instructions>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

memeriksa

Sebuah saklar Boolean yang, jika ada, menampilkan tombol radio seperti yang dicentang.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan tombol sebagai dinonaktifkan dan mencegah dari yang diperiksa.

nama

String yang digunakan untuk mengidentifikasi jawaban yang diajukan oleh pekerja. Nilai ini akan cocok dengan kunci dalam objek JSON yang menentukan jawabannya.

> **Note**
>
> Jika Anda menggunakan tombol di luarkelompok radio kerumunanelemen, tetapi dengannamestring dan berbedavaluestring,nameobjek dalam output akan berisi nilai Boolean untuk masing-masingvaluestring. Untuk memastikan bahwa hanya satu tombol dalam kelompok yang dipilih, membuat mereka anak-anak darikelompok radio kerumunanelemen dan menggunakan nilai nama yang berbeda.

nilai

Sebuah nama properti untuk nilai boolean elemen. Jika tidak ditentukan, ia menggunakan “on” sebagai default, mis.`{ "<name>": { "<value>": <true or false> } }`.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:kelompok radio kerumunan
- Elemen Anak: tidak ada

Output

Output objek dengan pola berikut: `{ "<name>": { "<value>": <true or false> } }`. Jika Anda menggunakan tombol di luarkelompok radio kerumunanelemen, tetapi dengan`name`string dan berbeda`value`string, nama objek akan berisi nilai Boolean untuk masing-masing`value`string. Untuk memastikan bahwa hanya satu dalam kelompok tombol yang dipilih, membuat mereka anak-anak darikelompok radio kerumunanelemen dan menggunakan nilai nama yang berbeda.

Example Contoh output dari elemen ini

```
[
  {
    "btn1": {
      "yes": true
    },
    "btn2": {
      "no": false
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kelompok radio kerumunan

Sekelompok tombol radio. Hanya satu tombol radio dalam grup yang dapat dipilih. Memilih satu tombol radio membersihkan tombol radio yang dipilih sebelumnya dalam kelompok yang sama.

Untuk contoh template UI kustom yang menggunakan`crowd-radio-group`elemen, lihat iniTemplat Kustom Pekerjaan Pelabelan Pengakuan Entitas.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh sintaksis yang dapat Anda gunakan dengan`<crowd-radio-group>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<style>
 body {
  padding-left: 20px;
  margin-bottom: 20px;
 }
 #outer-container {
     display: flex;
     justify-content: space-around;
     max-width: 900px;
     margin-left: 100px;
 }
 .left-container {
     margin-right: auto;
     padding-right: 50px;
 }
 .right-container {
     margin-left: auto;
     padding-left: 50px;
 }
 #vertical-separator {
     border: solid 1px #d5dbdb;
 }
</style>

<crowd-form>
    <div>
        <h1>Instructions</h1>
 Lorem ipsum...
    </div>
    <div>
        <h2>Background</h2>
```

```
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
 exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
   </div>
   <div id="outer-container">
<span class="left-container">
    <h2>Option 1</h2>
    <p>Nulla facilisi morbi tempus iaculis urna. Orci dapibus ultrices in iaculis nunc
 sed augue lacus.</p>
</span>
<span id="vertical-separator"></span>
<span class="right-container">
    <h2>Option 2</h2>
    <p>Ultrices vitae auctor eu augue ut. Pellentesque massa placerat duis ultricies
 lacus sed turpis tincidunt id.</p>
</span>
   </div>
   <div>
       <h2>Question</h2>
    <p>Which do you agree with?</p>
<crowd-radio-group>
    <crowd-radio-button name="option1" value="Option 1">Option 1</crowd-radio-button>
    <crowd-radio-button name="option2" value="Option 2">Option 2</crowd-radio-button>
</crowd-radio-group>

    <p>Why did you choose this answer?</p>
<crowd-text-area name="explanation" placeholder="Explain how you reached your
 conclusion..."></crowd-text-area>
   </div>
</crowd-form>
```

Atribut

Tidak ada atribut khusus yang didukung oleh elemen ini.

Elemen Hierarki

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak:tombol radio kerumunan

Output

Output array objek yang mewakilitombol radio kerumunanelemen di dalamnya.

Example Contoh dari Elemen Output

```
[
  {
    "btn1": {
      "yes": true
    },
    "btn2": {
      "no": false
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-semantik-segmentasi

Widget untuk segmentasi gambar dan menetapkan label ke setiap segmen gambar.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-semantic-segmentation>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-semantic-segmentation
    name="annotatedResult"
    src="{{ task.input.taskObject | grant_read_access }}"
    header="Please label each of the requested objects in this image"
```

```
    labels="['Cat', 'Dog', 'Bird']"
  >
    <full-instructions header="Segmentation Instructions">
      <ol>
          <li><strong>Read</strong> the task carefully and inspect the image.</li>
          <li><strong>Read</strong> the options and review the examples provided to
 understand more about the labels.</li>
          <li><strong>Choose</strong> the appropriate label that best suits the
 image.</li>
      </ol>
    </full-instructions>

    <short-instructions>
      <p>Use the tools to label the requested items in the image</p>
    </short-instructions>
  </crowd-semantic-segmentation>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Teks yang akan ditampilkan di atas gambar. Ini biasanya pertanyaan atau instruksi sederhana untuk pekerja.

nilai awal

Objek JSON yang berisi pemetaan warna dari pekerjaan segmentasi semantik sebelumnya dan tautan ke output gambar overlay oleh pekerjaan sebelumnya. Sertakan ini ketika Anda ingin pekerja manusia untuk memverifikasi hasil pekerjaan pelabelan sebelumnya dan menyesuaikannya jika perlu.

Atribut akan muncul sebagai berikut:

```
  initial-value='{
    "labelMappings": {
        "Bird": {
          "color": "#ff7f0e"
        },
        "Cat": {
```

```
          "color": "#2ca02c"
        },
        "Cow": {
          "color": "#d62728"
        },
        "Dog": {
          "color": "#1f77b4"
        }
      },
    "src": {{ "S3 file URL for image" | grant_read_access }}
  }'
```

Saat menggunakan Ground Truthtipe yang dibangunbersamakonsolidasi(di mana lebih dari satu pekerja label gambar tunggal), pemetaan label termasuk dalam catatan output pekerja individu, namun hasil keseluruhan direpresentasikan sebagai`internal-color-map`dalam hasil konsolidasi.

Anda dapat mengkonversi`internal-color-map`kepada`label-mappings`dalam template kustom menggunakan bahasa template Liquid:

```
initial-value="{
  'src' : '{{ task.input.manifestLine.label-attribute-name-from-prior-job|
 grant_read_access }}',
  'labelMappings': {
     {% for box in task.input.manifestLine.label-attribute-name-from-prior-job-
metadata.internal-color-map %}
       {% if box[1]['class-name'] != 'BACKGROUND' %}
         {{ box[1]['class-name'] | to_json }}: {
           'color': {{ box[1]['hex-color'] | to_json }}
         },
       {% endif %}
     {% endfor %}
   }
}"
```

label

Sebuah array JSON diformat string, yang masing-masing adalah label yang pekerja dapat menetapkan ke segmen gambar.

nama

Nama widget ini. Hal ini digunakan sebagai kunci untuk input widget dalam output form.

src

URL gambar yang akan tersegmentasi.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen Anak:instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut didukung oleh elemen ini.

instruksi lengkap

Petunjuk umum tentang bagaimana melakukan segmentasi gambar.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Output

Output berikut didukung oleh elemen ini.

labeledImage

Objek JSON yang berisi PNG yang dikodekan Base64 dari label.

LabelMappings

Objek JSON yang berisi objek dengan nama dengan label segmentasi.

- warna- Nilai heksadesimal warna RGB label di`labeledImage`PNG.

initialValueModified

Sebuah boolean mewakili apakah nilai awal telah dimodifikasi. Ini hanya disertakan ketika output dari tugas penyesuaian.

InputImageProperties

Sebuah objek JSON yang menentukan dimensi gambar yang sedang dijelaskan oleh pekerja. Objek ini berisi properti berikut.

- tingginya— Tinggi, dalam piksel, gambar.
- lebar— Lebar, dalam piksel, gambar.

Example : Contoh Elemen Output

Berikut ini adalah contoh output dari elemen ini.

```
[
  {
    "annotatedResult": {
      "inputImageProperties": {
        "height": 533,
        "width": 800
      },
      "labelMappings": {
        "<Label 2>": {
          "color": "#ff7f0e"
        },
        "<label 3>": {
          "color": "#2ca02c"
        },
        "<label 1>": {
          "color": "#1f77b4"
        }
      },
      "labeledImage": {
        "pngImageData": "<Base-64 Encoded Data>"
      }
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data

- Crowd HTML Elemen Referensi

kerumunan-slider

Sebuah bar dengan tombol geser yang memungkinkan pekerja untuk memilih nilai dari berbagai nilai dengan memindahkan kenop. Slider menjadikannya pilihan tepat untuk pengaturan yang mencerminkan tingkat intensitas, seperti volume, kecerahan, atau saturasi warna.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat survei yang menggunakan`<crowd-slider>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
<crowd-instructions link-text="View instructions" link-type="button">
  <short-summary>
    <p>Provide a brief instruction here</p>
  </short-summary>

  <detailed-instructions>
    <h3>Provide more detailed instructions here</h3>
    <p>Include additional information</p>
  </detailed-instructions>

  <positive-example>
    <p>Provide an example of a good answer here</p>
    <p>Explain why it's a good answer</p>
  </positive-example>

  <negative-example>
    <p>Provide an example of a bad answer here</p>
    <p>Explain why it's a bad answer</p>
  </negative-example>
</crowd-instructions>

<div>
  <p>What is your favorite color for a bird?</p>
  <crowd-input name="favoriteColor" placeholder="example: pink" required></crowd-input>
```

```
</div>

<div>
  <p>Check this box if you like birds</p>
  <crowd-checkbox name="likeBirds" checked="true" required></crowd-checkbox>
</div>

<div>
  <p>On a scale of 1-10, how much do you like birds?</p>
  <crowd-slider name="howMuch" min="1" max="10" step="1" pin="true" required></crowd-
slider>
</div>

<div>
  <p>Write a short essay describing your favorite bird</p>
  <crowd-text-area name="essay" rows="4" placeholder="Lorem ipsum..." required></crowd-
text-area>
</div>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan slider sebagai dinonaktifkan.

diedit

Sebuah saklar Boolean yang, jika ada, menampilkan tombol atas/bawah yang dapat dipilih untuk memilih nilai.

Memilih nilai melalui tombol atas/bawah adalah alternatif untuk memilih nilai dengan memindahkan kenop pada slider. Tombol pada slider akan bergerak serentak dengan pilihan tombol atas/bawah.

max

Sejumlah yang menentukan nilai maksimum pada slider.

min

Sejumlah yang menentukan nilai minimum pada slider.

nama

String yang digunakan untuk mengidentifikasi jawaban yang diajukan oleh pekerja. Nilai ini akan cocok dengan kunci dalam objek JSON yang menentukan jawabannya.

pin

Sebuah saklar Boolean yang, jika ada, menampilkan nilai saat ini di atas kenop sebagai kenop dipindahkan.

dibutuhkan

Sebuah saklar Boolean yang, jika ada, membutuhkan pekerja untuk memberikan masukan.

Kemajuan sekunder

Bila digunakan dengan`crowd-slider-secondary-color`Atribut CSS, progress bar diwarnai ke titik yang diwakili oleh`secondary-progress`. Misalnya, jika ini mewakili kemajuan pada video streaming,`value`akan mewakili tempat penampil berada di timeline video. Parameter`secondary-progress`nilai akan mewakili titik pada timeline yang video telah buffered.

langkah

Sejumlah yang menentukan perbedaan antara nilai-nilai yang dapat dipilih pada slider.

nilai

Sebuah preset yang menjadi default jika pekerja tidak memberikan masukan.

Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen Induk:Formulir kerumunan
- Elemen Anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

Tab kerumunan

Komponen ditata agar terlihat seperti tab dengan informasi di bawah ini.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut adalah contoh`<crowd-tab>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-tabs>
    <crowd-tab header="Tab 1">
      <h2>Image</h2>

      <img
        src="https://images.unsplash.com/photo-1478382188900-5bb598fe27d3?
ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1351&q=80"
        style="max-width: 40%"
      >

      <h2>Text</h2>
      <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua.
      </p>
      <p>
      Sed risus ultricies tristique nulla aliquet enim tortor at auctor. Tempus egestas
 sed sed risus.
      </p>
    </crowd-tab>

    <crowd-tab header="Tab 2">
      <h2>Description</h2>
      <p>
      Sed risus ultricies tristique nulla aliquet enim tortor at auctor. Tempus egestas
 sed sed risus.
      </p>
    </crowd-tab>

    <crowd-tab header="Tab 3">
```

```
      <div style="width: 40%; display: inline-block">
        <img
          src="https://images.unsplash.com/photo-1472747459646-91fd6f13995f?
ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1350&q=80"
          style="max-width: 80%"
        >
        <crowd-input label="Input inside tab" name="inputInsideTab"></crowd-input>
        <input type="checkbox" name="checkbox" value="foo">Foo
        <input type="checkbox" name="checkbox" value="bar">Bar
        <crowd-button>Some button</crowd-button>
      </div>

      <div style="width: 40%; display: inline-block; vertical-align: top">
        Lorem ipsum dolor sit amet, lorem a wisi nibh, in pulvinar, consequat praesent
 vestibulum tellus ante felis auctor, vitae lobortis dictumst mauris.
        Pellentesque nulla ipsum ante quisque quam augue.
        Class lacus id euismod, blandit tempor mauris quisque tortor mauris,
 urna gravida nullam pede libero, ut suscipit orci faucibus lacus varius ornare,
 pellentesque ipsum.
        At etiam suspendisse est elementum luctus netus, vel sem nulla sodales, potenti
 magna enim ipsum diam tortor rutrum,
        quam donec massa elit ac, nam adipiscing sed at leo ipsum consectetuer.
 Ac turpis amet wisi, porttitor sint lacus ante, turpis accusantium, ac maecenas
 deleniti,
        nisl leo sem integer ac dignissim. Lobortis etiam luctus lectus odio auctor.
 Justo vitae, felis integer id, bibendum accumsan turpis eu est mus eros, ante id
 eros.
      </div>
    </crowd-tab>

  </crowd-tabs>

  <crowd-input label="Input outside tabs" name="inputOutsideTab"></crowd-input>

  <short-instructions>
    <p>Sed risus ultricies tristique nulla aliquet enim tortor at auctor. Tempus
 egestas sed sed risus.</p>
</short-instructions>

<full-instructions header="Classification Instructions">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua.</p>
    <p> Tempus egestas sed sed risus.</p>
</full-instructions>
```

```
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

header

Teks yang muncul di tab. Ini biasanya beberapa nama deskriptif singkat indikasi informasi yang terkandung di bawah tab.

Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:kerumunan-tab
- Elemen anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-tab

Sebuah wadah untuk informasi tab.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut adalah contoh`<crowd-tabs>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <crowd-tabs>
    <crowd-tab header="Tab 1">
      <h2>Image</h2>
```

```
      <img
        src="https://images.unsplash.com/photo-1478382188900-5bb598fe27d3?
ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1351&q=80"
        style="max-width: 40%"
      >

      <h2>Text</h2>
      <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua.
      </p>
      <p>
      Sed risus ultricies tristique nulla aliquet enim tortor at auctor. Tempus egestas
 sed sed risus.
      </p>
    </crowd-tab>

    <crowd-tab header="Tab 2">
      <h2>Description</h2>
      <p>
      Sed risus ultricies tristique nulla aliquet enim tortor at auctor. Tempus egestas
 sed sed risus.
      </p>
    </crowd-tab>

    <crowd-tab header="Tab 3">
      <div style="width: 40%; display: inline-block">
        <img
          src="https://images.unsplash.com/photo-1472747459646-91fd6f13995f?
ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1350&q=80"
          style="max-width: 80%"
        >
        <crowd-input label="Input inside tab" name="inputInsideTab"></crowd-input>
        <input type="checkbox" name="checkbox" value="foo">Foo
        <input type="checkbox" name="checkbox" value="bar">Bar
        <crowd-button>Some button</crowd-button>
      </div>

      <div style="width: 40%; display: inline-block; vertical-align: top">
        Lorem ipsum dolor sit amet, lorem a wisi nibh, in pulvinar, consequat praesent
 vestibulum tellus ante felis auctor, vitae lobortis dictumst mauris.
        Pellentesque nulla ipsum ante quisque quam augue.
```

```
        Class lacus id euismod, blandit tempor mauris quisque tortor mauris,
 urna gravida nullam pede libero, ut suscipit orci faucibus lacus varius ornare,
 pellentesque ipsum.
        At etiam suspendisse est elementum luctus netus, vel sem nulla sodales, potenti
 magna enim ipsum diam tortor rutrum,
        quam donec massa elit ac, nam adipiscing sed at leo ipsum consectetuer.
 Ac turpis amet wisi, porttitor sint lacus ante, turpis accusantium, ac maecenas
 deleniti,
        nisl leo sem integer ac dignissim. Lobortis etiam luctus lectus odio auctor.
 Justo vitae, felis integer id, bibendum accumsan turpis eu est mus eros, ante id
 eros.
     </div>
   </crowd-tab>

  </crowd-tabs>

  <crowd-input label="Input outside tabs" name="inputOutsideTab"></crowd-input>

  <short-instructions>
    <p>Sed risus ultricies tristique nulla aliquet enim tortor at auctor. Tempus
 egestas sed sed risus.</p>
</short-instructions>

<full-instructions header="Classification Instructions">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua.</p>
    <p> Tempus egestas sed sed risus.</p>
</full-instructions>

</crowd-form>
```

Atribut

Elemen ini tidak memiliki atribut.

Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak:Tab kerumunan

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan-teks-daerah

Sebuah bidang untuk input teks.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh template Liquid yang dirancang untuk menuliskan klip audio yang menggunakan`<crowd-text-area>`elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <audio controls>
      <source src="{{ task.input.taskObject | grant_read_access }}" type="audio/mpeg">
      Your browser does not support the audio element.
  </audio>
  <h3>Instructions</h3>
  <p>Transcribe the audio</p>
  <p>Ignore "umms", "hmms", "uhs" and other non-textual phrases</p>
  <crowd-text-area name="transcription" rows="4"></crowd-text-area>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

pola diperbolehkankan

Ekspresi reguler yang digunakan denganvalidasi otomatisatribut untuk mengabaikan karakter non-pencocokan sebagai jenis pekerja.

fokus otomatis

Sebuah saklar Boolean yang, jika ada, menempatkan kursor dalam elemen ini on-load sehingga pengguna dapat segera mulai mengetik tanpa harus mengklik di dalam elemen.

validasi otomatis

Sebuah saklar Boolean yang, jika ada, menyalakan validasi input. Perilaku validator dapat dimodifikasi olehpesan kesalahandanpola diperbolehkankanatribut.

penghitung

Sebuah saklar Boolean yang, jika ada, menempatkan bidang teks kecil di bawah sudut kanan bawah elemen, menampilkan jumlah karakter di dalam elemen.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan area input sebagai dinonaktifkan.

pesan kesalahan

Teks yang akan ditampilkan di bawah bidang input, di sisi kiri, jika validasi gagal.

label

Sebuah string yang ditampilkan di dalam bidang teks.

Teks ini menyusut dan naik di atas bidang teks ketika pekerja mulai mengetik di lapangan atau ketikanilaiatribut diatur.

max-length

Integer yang menentukan jumlah maksimum karakter diperbolehkan oleh elemen. Karakter yang diketik atau disisipkan di luar maksimum diabaikan.

baris maks

Integer yang menentukan jumlah maksimum baris teks yang diperbolehkan dalam kerumunan teks-area. Biasanya elemen mengembang untuk mengakomodasi baris baru. Jika ini diatur, setelah jumlah baris melebihi itu, konten menggulir ke atas dari tampilan dan kontrol scrollbar muncul.

nama

Sebuah string yang digunakan untuk mewakili data elemen dalam output.

placeholder

Sebuah string disajikan kepada pengguna sebagai teks placeholder. Ini menghilang setelah pengguna menempatkan sesuatu di area input.

baris

Integer yang menentukan ketinggian elemen dalam baris teks.

nilai

Sebuah preset yang menjadi default jika pekerja tidak memberikan masukan. Preset muncul di bidang teks.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak: tidak ada

Output

Elemen ini outputnameSebagai nama properti dan isi teks elemen sebagai nilai. Carriage kembali dalam teks direpresentasikan sebagai\n.

Example Output sampel untuk elemen ini

```
[
  {
    "textInput1": "This is the text; the text that\nmakes the crowd go wild."
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

kerumunan bersulang

Notifikasi halus yang muncul sementara di layar. Hanya satu keramaian roti yang terlihat.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Berikut ini adalah contoh templat Liquid yang menggunakan`<crowd-toast>`Elemen. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <p>Find the official website for: <strong>{{ task.input.company }}</strong></p>
  <p>Do not give Yelp pages, LinkedIn pages, etc.</p>
  <p>Include the http:// prefix from the website</p>
  <crowd-input name="website" placeholder="http://example.com"></crowd-input>

  <crowd-toast duration="10000" opened>
    This is a message that you want users to see when opening the template. This
 message will disappear in 10 seconds.
   </crowd-toast>

</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

durasi

Angka yang menentukan durasi, dalam milidetik, bahwa notifikasi muncul di layar.

teks

Teks yang akan ditampilkan dalam notifikasi.

Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen induk:Formulir kerumunan
- Elemen anak: tidak ada

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

crowd-toggle-tombol

Sebuah tombol yang bertindak sebagai saklar ON/OFF, toggling state.

Lihat contoh interaktif dari template HTML yang menggunakan Crowd HTML Element ini diCodePen.

Contoh berikut menunjukkan berbagai cara Anda dapat menggunakan`<crowd-toggle-button>`Elemen HTML. Salin kode berikut dan simpan dalam file dengan ekstensi`.html`. Buka file di browser apa pun untuk melihat pratinjau dan berinteraksi dengan template ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
  <!--Toggle button without value-->
  <crowd-toggle-button name="toggleButtonWithoutValue"></crowd-toggle-button>

  <!--Toggle button with value-->
  <crowd-toggle-button name="toggleButtonWithValue" value="someValue"></crowd-toggle-
button>

  <!--Toggle button disabled-->
  <crowd-toggle-button name="toggleButtonDisabled" disabled></crowd-toggle-button>

  <!--Toggle button marked invalid-->
  <crowd-toggle-button name="toggleButtonInvalid" invalid></crowd-toggle-button>

  <!--Toggle button marked required-->
  <crowd-toggle-button name="toggleButtonRequired" required></crowd-toggle-button>
</crowd-form>
```

Atribut

Atribut berikut didukung oleh elemen ini.

memeriksa

Sebuah saklar Boolean yang, jika ada, menampilkan tombol beralih ke posisi ON.

dinonaktifkan

Sebuah saklar Boolean yang, jika ada, menampilkan tombol sebagai dinonaktifkan dan mencegah Toggling.

tidak valid

Ketika dalam posisi off, tombol menggunakan atribut ini, akan ditampilkan dalam warna peringatan. Standar merah, tetapi dapat diubah dalam CSS. Ketika dinyalakan, tombol akan ditampilkan dalam warna yang sama dengan tombol lain di posisi on.

nama

String yang digunakan untuk mengidentifikasi jawaban yang diajukan oleh pekerja. Nilai ini cocok dengan kunci dalam objek JSON yang menentukan jawabannya.

dibutuhkan

Sebuah saklar Boolean yang, jika ada, membutuhkan pekerja untuk memberikan masukan.

nilai

Sebuah nilai yang digunakan dalam output sebagai nama properti untuk negara Boolean elemen. Default untuk “on” jika tidak disediakan.

Elemen Hierarki

Elemen ini memiliki elemen induk dan anak berikut.

- Elemen Induk:Formulir kerumunan
- Elemen Anak: tidak ada

Output

Elemen ini outputnamesebagai nama objek, yang berisivaluesebagai nama properti dan negara elemen sebagai nilai Boolean untuk properti. Jika tidak ada nilai untuk elemen ditentukan, default nama properti untuk “on.”

Example Output sampel untuk elemen ini

```
[
  {
    "theToggler": {
      "on": true
    }
  }
]
```

Lihat Juga

Untuk informasi selengkapnya, lihat berikut ini.

- Gunakan Amazon SageMaker Ground Truth untuk Label Data
- Crowd HTML Elemen Referensi

Augmented AI Crowd HTML Elemen

Elemen HTML Crowd berikut ini hanya tersedia untuk tugas alur kerja manusia Amazon Augmented AI.

Topik

- kerumunan-textract-analisis-dokumen
- kerumunan-rekognition-deteksi-moderasi-label

kerumunan-textract-analisis-dokumen

Widget untuk mengaktifkan peninjauan manusia atas hasil analisis dokumen Amazon Textract Texact.

Atribut

Atribut berikut didukung oleh elemen ini.

header

Ini adalah teks yang ditampilkan sebagai header.

src

Ini adalah link ke gambar yang akan dianalisis oleh pekerja.

initialValue

Ini menetapkan nilai awal untuk atribut yang ditemukan di UI pekerja.

Berikut ini adalah contoh dari`initialValue`masukan:

```
[
            {
                "blockType": "KEY_VALUE_SET",
                "confidence": 38.43309020996094,
                "geometry": {
                    "boundingBox": {
                        "width": 0.32613086700439453,
                        "weight": 0.0942094624042511,
                        "left": 0.4833833575248718,
                        "top": 0.5227988958358765
                    },
                    "polygon": [
                        {"x": 0.123, "y": 0.345}, ...
                    ]
                }
                "id": "8c97b240-0969-4678-834a-646c95da9cf4",
                "relationships": [
                    {
                        "type": "CHILD",
                        "ids": [
                            "7ee7b7da-ee1b-428d-a567-55a3e3affa56",
                            "4d6da730-ba43-467c-a9a5-c6137ba0c472"
                        ]
                    },
                    {
                        "type": "VALUE",
                        "ids": [
                            "6ee7b7da-ee1b-428d-a567-55a3e3affa54"
                        ]
```

```
                    }
                ],
                "entityTypes": [
                    "KEY"
                ],
                "text": "Foo bar"
            },
]
```

BlockTypes

Hal ini menentukan jenis analisis pekerja dapat melakukan. HANYAKEY_VALUE_SETsaat ini didukung.

keys

Ini menentukan kunci baru dan nilai teks terkait pekerja dapat menambahkan. Nilai input untuk`keys`dapat mencakup unsur-unsur berikut:

- `importantFormKey`menerima string, dan digunakan untuk menentukan satu kunci.
- `importantFormKeyAliases`dapat digunakan untuk menentukan alias yang alternatif yang dapat diterima untuk kunci yang disediakan. Gunakan elemen ini untuk mengidentifikasi ejaan alternatif atau presentasi kunci Anda. Parameter ini menerima daftar satu atau beberapa string.

Berikut ini adalah contoh dari masukan untuk`keys`.

```
[
      {
        importantFormKey: 'Address',
        importantFormKeyAliases: [
          'address',
          'Addr.',
          'Add.',
        ]
      },
      {
        importantFormKey: 'Last name',
        importantFormKeyAliases: ['Surname']
      }
]
```

tanpa kunci-edit

Hal ini mencegah para pekerja mengedit kunci anotasi yang dilewati`initialValue`. Hal ini mencegah pekerja mengedit kunci yang telah terdeteksi pada dokumen Anda. Ini wajib diisi.

no-geometry-edit

Hal ini mencegah pekerja mengedit poligon anotasi yang dilewati`initialValue`. Misalnya, ini akan mencegah pekerja mengedit kotak pembatas di sekitar kunci yang diberikan. Ini wajib diisi.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- elemen induk - kerumunan-bentuk
- Elemen anak -instruksi lengkap,instruksi singkat

Wilayah

Daerah berikut didukung oleh elemen ini. Anda dapat menggunakan kode HTML dan CSS khusus di wilayah ini untuk memformat instruksi Anda kepada pekerja. Misalnya, gunakan`short-instructions`bagian untuk memberikan contoh yang baik dan buruk tentang bagaimana menyelesaikan tugas.

instruksi lengkap

Petunjuk umum tentang cara bekerja dengan widget.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Contoh Template Pekerja Menggunakan Element kerumunan

Contoh template pekerja menggunakan elemen kerumunan ini akan terlihat seperti berikut ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
{% capture s3_uri %}http://s3.amazonaws.com/
{{ task.input.aiServiceRequest.document.s3Object.bucket }}/
{{ task.input.aiServiceRequest.document.s3Object.name }}{% endcapture %}

<crowd-form>
  <crowd-textract-analyze-document
```

```
    src="{{ s3_uri | grant_read_access }}"
    initial-value="{{ task.input.selectedAiServiceResponse.blocks }}"
    header="Review the key-value pairs listed on the right and correct them if they
 don't match the following document."
    no-key-edit
    no-geometry-edit
    keys="{{ task.input.humanLoopContext.importantFormKeys }}"
    block-types="['KEY_VALUE_SET']"
  >
    <short-instructions header="Instructions">
      <style>
        .instructions {
          white-space: pre-wrap;
        }
        .instructionsImage {
          display: inline-block;
          max-width: 100%;
        }
      </style>
      <p class='instructions'>Click on a key-value block to highlight the corresponding
 key-value pair in the document.

If it is a valid key-value pair, review the content for the value. If the content is
 incorrect, correct it.

The text of the value is incorrect, correct it.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/
correct-value-text.png" />

A wrong value is identified, correct it.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/
correct-value.png" />

If it is not a valid key-value relationship, choose No.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/not-a-
key-value-pair.png" />

If you can’t find the key in the document, choose Key not found.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/key-is-
not-found.png" />

If the content of a field is empty, choose Value is blank.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/value-
is-blank.png" />
```

```
<b>Examples</b>
Key and value are often displayed next or below to each other.

Key and value displayed in one line.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/sample-
key-value-pair-1.png" />

Key and value displayed in two lines.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/sample-
key-value-pair-2.png" />

If the content of the value has multiple lines, enter all the text without line break.
 Include all value text even if it extends beyond the highlight box.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/
multiple-lines.png" /></p>
    </short-instructions>

    <full-instructions header="Instructions"></full-instructions>
  </crowd-textract-analyze-document>
</crowd-form>
```

Output

Berikut ini adalah contoh output dari elemen ini. Anda dapat menemukan penjelasan rinci tentang output ini di Amazon TextractAnalyzeDocumentDokumentasi API.

```
{
  "AWS/Textract/AnalyzeDocument/Forms/V1": {
    blocks: [
      {
        "blockType": "KEY_VALUE_SET",
        "id": "8c97b240-0969-4678-834a-646c95da9cf4",
        "relationships": [
          {
            "type": "CHILD",
            "ids": ["7ee7b7da-ee1b-428d-a567-55a3e3affa56", "4d6da730-ba43-467c-a9a5-
c6137ba0c472"]
          },
          {
            "type": "VALUE",
            "ids": ["6ee7b7da-ee1b-428d-a567-55a3e3affa54"]
          }
```

```
        ],
        "entityTypes": ["KEY"],
        "text": "Foo bar baz"
      }
    ]
  }
}
```

kerumunan-rekognition-deteksi-moderasi-label

Widget untuk mengaktifkan peninjauan manusia atas hasil moderasi gambar Amazon Rekognition.

Atribut

Atribut berikut didukung oleh elemen ini.

header

Ini adalah teks yang ditampilkan sebagai header.

src

Ini adalah link ke gambar yang akan dianalisis oleh pekerja.

kategori

Mendukung ini`categories`sebagai array stringatauarray objek di mana setiap objek memiliki`name`bidang.

Jika kategori masuk sebagai objek, hal-hal berikut berlaku:

- Kategori yang ditampilkan adalah nilai`name`bidang.
- Jawaban yang dikembalikan berisipenuhobjek dari setiap kategori yang dipilih.

Jika kategori masuk sebagai string, hal-hal berikut berlaku:

- Jawaban yang dikembalikan adalah array dari semua string yang dipilih.

Kategori eksklusion-

Dengan menetapkan atribut ini, Anda membuat tombol di bawah kategori di UI.

- Ketika pengguna memilih tombol, semua kategori tidak dipilih dan dinonaktifkan.

- Memilih tombol lagi mengaktifkan kembali kategori sehingga pengguna dapat memilih mereka.
- Jika Anda mengirimkan setelah memilih tombol, ia mengembalikan array kosong.

Hirarki Elemen

Elemen ini memiliki elemen induk dan anak berikut.

- elemen induk - kerumunan-bentuk
- Elemen anak -instruksi lengkap,instruksi singkat

Wilayah AWS

Berikut iniAWSDaerah didukung oleh elemen ini. Anda dapat menggunakan kode HTML dan CSS khusus dalam Wilayah ini untuk memformat instruksi Anda kepada pekerja. Misalnya, gunakan`short-instructions`bagian untuk memberikan contoh yang baik dan buruk tentang bagaimana menyelesaikan tugas.

instruksi lengkap

Petunjuk umum tentang cara bekerja dengan widget.

instruksi singkat

Instruksi khusus tugas penting yang ditampilkan di tempat yang menonjol.

Contoh Pekerja Template dengan kerumunan Element

Contoh template pekerja menggunakan elemen kerumunan akan terlihat seperti berikut ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
{% capture s3_uri %}http://s3.amazonaws.com/
{{ task.input.aiServiceRequest.image.s3Object.bucket }}/
{{ task.input.aiServiceRequest.image.s3Object.name }}{% endcapture %}

<crowd-form>
  <crowd-rekognition-detect-moderation-labels
    categories='[
      {% for label in task.input.selectedAiServiceResponse.moderationLabels %}
        {
          name: "{{ label.name }}",
          parentName: "{{ label.parentName }}",
        },
```

```
      {% endfor %}
    ]'
    src="{{ s3_uri | grant_read_access }}"
    header="Review the image and choose all applicable categories."
  >
    <short-instructions header="Instructions">
      <style>
        .instructions {
          white-space: pre-wrap;
        }
      </style>
      <p class='instructions'>Review the image and choose all applicable categories.
If no categories apply, choose None.

<b>Nudity</b>
Visuals depicting nude male or female person or persons

<b>Graphic Male Nudity</b>
Visuals depicting full frontal male nudity, often close ups

<b>Graphic Female Nudity</b>
Visuals depicting full frontal female nudity, often close ups

<b>Sexual Activity</b>
Visuals depicting various types of explicit sexual activities and pornography

<b>Illustrated Nudity or Sexual Activity</b>
Visuals depicting animated or drawn sexual activity, nudity or pornography

<b>Adult Toys</b>
Visuals depicting adult toys, often in a marketing context

<b>Female Swimwear or Underwear</b>
Visuals depicting female person wearing only swimwear or underwear

<b>Male Swimwear Or Underwear</b>
Visuals depicting male person wearing only swimwear or underwear

<b>Partial Nudity</b>
Visuals depicting covered up nudity, for example using hands or pose

<b>Revealing Clothes</b>
Visuals depicting revealing clothes and poses, such as deep cut dresses
```

```
<b>Graphic Violence or Gore</b>
Visuals depicting prominent blood or bloody injuries

<b>Physical Violence</b>
Visuals depicting violent physical assault, such as kicking or punching

<b>Weapon Violence</b>
Visuals depicting violence using weapons like firearms or blades, such as shooting

<b>Weapons</b>
Visuals depicting weapons like firearms and blades

<b>Self Injury</b>
Visuals depicting self-inflicted cutting on the body, typically in distinctive patterns
 using sharp objects

<b>Emaciated Bodies</b>
Visuals depicting extremely malnourished human bodies

<b>Corpses</b>
Visuals depicting human dead bodies

<b>Hanging</b>
Visuals depicting death by hanging</p>
    </short-instructions>

    <full-instructions header="Instructions"></full-instructions>
  </crowd-rekognition-detect-moderation-labels>
</crowd-form>
```

Output

Berikut ini adalah contoh output dari elemen ini. Untuk detail tentang output ini, lihat Amazon RekognitionDetectModerationLabelsDokumentasi API.

```
{
  "AWS/Rekognition/DetectModerationLabels/Image/V3": {
    "ModerationLabels": [
        { name: 'Gore', parentName: 'Violence' },
        { name: 'Corpses', parentName: 'Violence' },
    ]
  }
}
```

Menggunakan Amazon Augmented AI untuk Human Review

Saat Anda menggunakan aplikasi AI seperti Amazon Rekognition, Amazon Textract, atau model machine learning (ML) khusus, Anda dapat menggunakan Amazon Augmented AI untuk mendapatkan ulasan manusia terhadap prediksi kepercayaan rendah atau sampel prediksi acak.

Apa itu Amazon Augmented AI?

Amazon Augmented AI (Amazon A2I) adalah layanan yang membawa ulasan manusia terhadap prediksi MLke semua pengembang dengan menghapus beban berat yang terkait dengan membangun sistem tinjauan manusia atau mengelola sejumlah besar pengulas manusia.

Banyak aplikasi ML mengharuskan manusia untuk meninjau prediksi kepercayaan rendah untuk memastikan hasilnya benar. Misalnya, mengekstraksi informasi dari formulir aplikasi hipotek yang dipindai dapat memerlukan peninjauan manusia karena pemindaian berkualitas rendah atau tulisan tangan yang buruk. Membangun sistem peninjauan manusia dapat memakan waktu dan mahal karena melibatkan penerapan proses atau alur kerja yang kompleks, menulis perangkat lunak khusus untuk mengelola tugas dan hasil peninjauan, dan mengelola kelompok besar pengulas.

Amazon A2I menyederhanakan membangun dan mengelola ulasan manusia untuk aplikasi ML. Amazon A2I menyediakan alur kerja peninjauan manusia bawaan untuk kasus penggunaan MS umum, seperti moderasi konten dan ekstraksi teks dari dokumen. Anda juga dapat membuat alur kerja Anda sendiri untuk model ML yang dibangun di atas SageMaker atau alat lainnya. Dengan menggunakan Amazon A2I, Anda dapat mengizinkan pengulas manusia untuk masuk ketika model tidak dapat membuat prediksi dengan kepercayaan tinggi atau mengaudit prediksinya secara berkelanjutan.

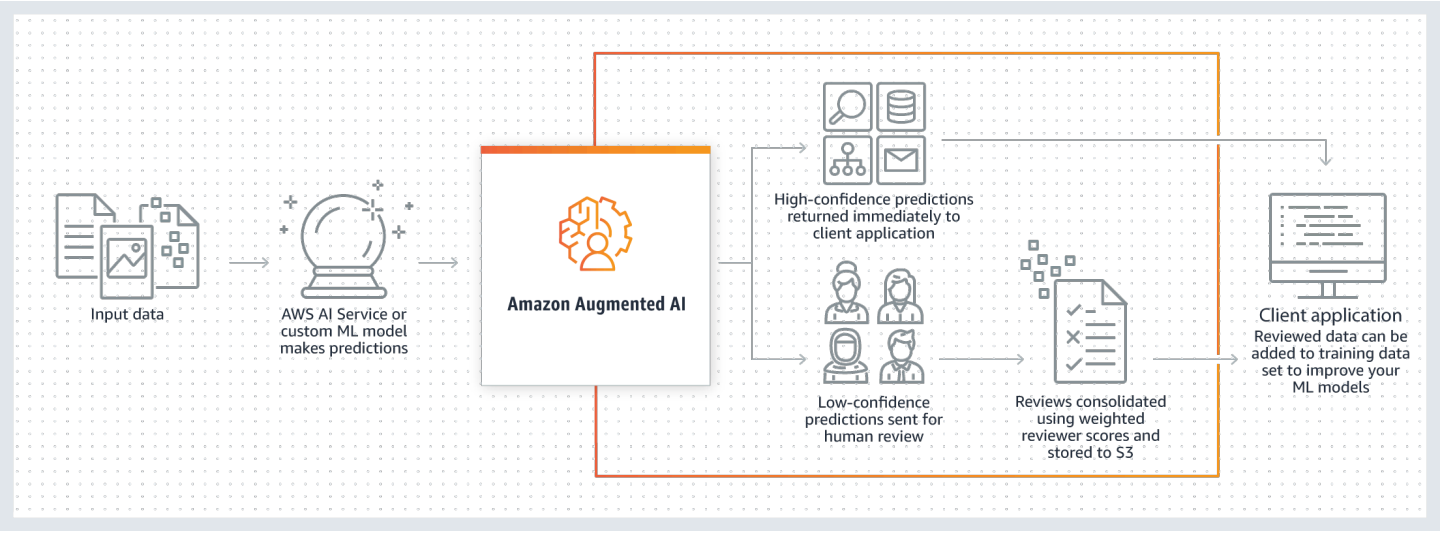
Contoh Kasus Penggunaan Amazon A2I

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Amazon A2I untuk mengintegrasikan loop ulasan manusia ke dalam aplikasi ML-mu. Untuk masing-masing contoh ini, Anda dapat menemukan Notebook Jupyter yang menunjukkan alur kerja tersebut. Contoh Kasus Penggunaan Amazon A2I

- Gunakan Amazon A2I dengan Amazon Textract — Minta manusia meninjau pasangan nilai kunci penting dalam dokumen satu halaman atau minta Amazon Textract secara acak mencicipi dan mengirim dokumen dari kumpulan data Anda ke manusia untuk ditinjau.
- Gunakan Amazon A2I dengan Amazon Rekognition — Minta manusia meninjau gambar yang tidak aman untuk konten dewasa atau kekerasan eksplisit jika Amazon Rekognition mengembalikan

skor kepercayaan rendah, atau meminta Amazon Rekognition secara acak mengambil sampel dan mengirim gambar dari kumpulan data Anda ke manusia untuk ditinjau.

- Gunakan Amazon A2I untuk meninjau inferensi ML—Gunakan Amazon A2I untuk meninjau kesimpulan real-time dan rendah kepercayaan yang dibuat oleh model yang diterapkan ke titik akhir yang SageMaker dihosting dan secara bertahap melatih model Anda menggunakan data keluaran Amazon A2I.
- Gunakan Amazon A2I dengan Amazon Comprehend — Minta manusia meninjau kesimpulan Amazon Comprehend tentang data teks seperti analisis sentimen, sintaks teks, dan deteksi entitas.
- Gunakan Amazon A2I dengan Amazon Transcribe — Minta manusia meninjau transkripsi Amazon Transcribe file video atau audio. Gunakan hasil loop tinjauan manusia transkripsi untuk membuat kosakata khusus dan meningkatkan transkripsi konten video atau audio serupa di masa mendatang.
- Gunakan Amazon A2I dengan Amazon Translate — Minta manusia meninjau terjemahan dengan kepercayaan rendah yang dikembalikan dari Amazon Translate.
- Gunakan Amazon A2I untuk meninjau data tabular — Gunakan Amazon A2I untuk mengintegrasikan loop tinjauan manusia ke dalam aplikasi MLyang menggunakan data tabular.



Topik

- Memulai dengan Amazon Augmented AI
- Contoh Kasus Penggunaan Amazon A2I
- Buat Alur Kerja Tinjauan Manusia
- Menghapus Alur Kerja Tinjauan Manusia

- Membuat dan Memulai Loop Manusia
- Menghapus Loop Manusia
- Membuat dan Mengelola Template Tugas Pekerja
- Pantau dan Kelola Loop Manusia Anda
- Data Output Amazon A2I
- Izin dan Keamanan di Amazon Augmented AI
- GunakanAmazon CloudWatch EventsAmazon Augmented AI
- Menggunakan API di Amazon Augmented AI

Memulai dengan Amazon Augmented AI

Untuk mulai menggunakan Amazon Augmented AI, tinjauKomponen Inti Amazon A2I danPrasyarat untuk Menggunakan Augmented AI. Kemudian, gunakan dokumentasi berikut untuk mempelajari cara menggunakan konsol dan API Amazon A2I.

- Tutorial: Mulai Konsol Amazon A2I
- Tutorial: Mulai Menggunakan API Amazon A2I

Anda juga bisa menatap menggunakan Amazon A2I API dengan mengikuti tutorial Jupyter Notebook. LihatContoh Kasus Penggunaan Amazon A2I daftar notebook dan kasus penggunaan.

Komponen Inti Amazon A2I

Tinjau persyaratan berikut untuk membiasakan diri dengan komponen inti Amazon A2I.

Jenis Tugas

Alur kerja AI/ML tempat Anda mengintegrasikan Amazon A2I mendefinisikan jenis tugas Amazon A2I.

Amazon A2I mendukung:

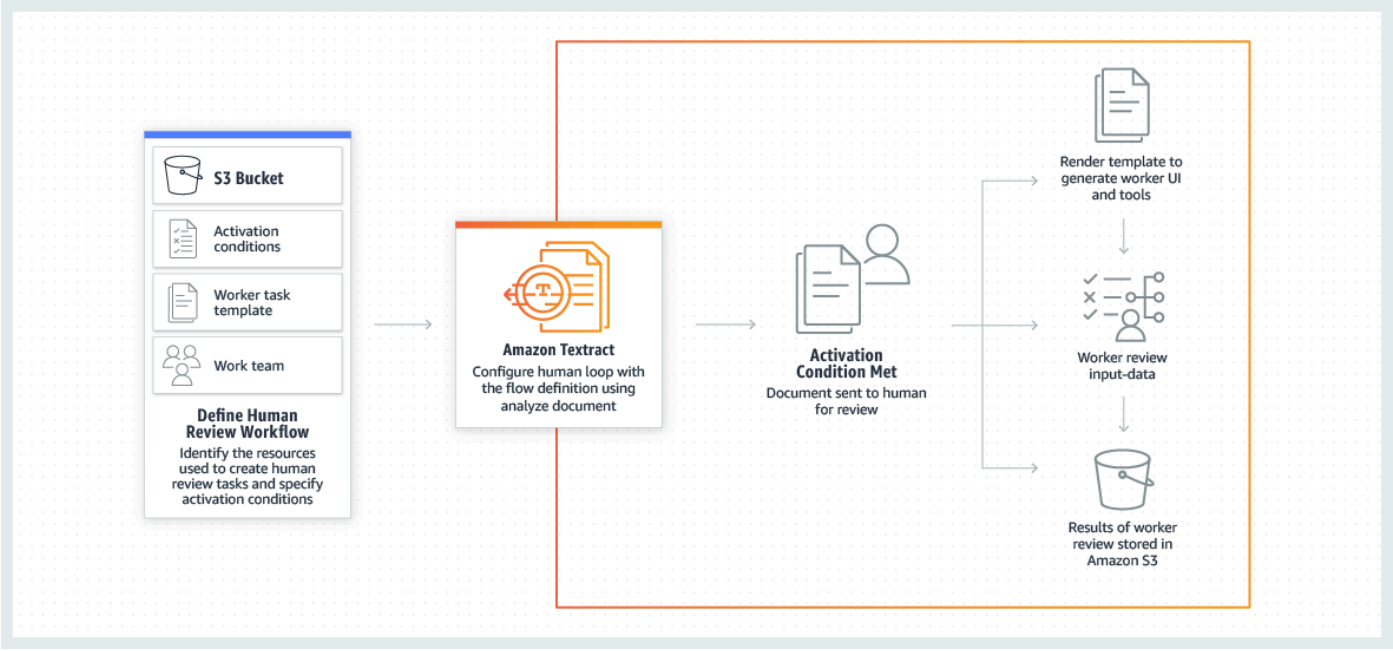
- Dua jenis tugas bawaan: Ekstraksi pasangan nilai kunci Amazon Textract dan moderasi gambar Amazon Rekognition.
- Jenis tugas khusus: Gunakan jenis tugas khusus untuk mengintegrasikan loop tinjauan manusia ke dalam alur kerja machine learning apa pun. Anda dapat menggunakan jenis tugas khusus untuk mengintegrasikan Amazon A2I denganAWS layanan lain seperti Amazon Comprehend, Amazon

Transcribe, dan Amazon Translate, serta alur kerja machine learning khusus Anda sendiri. Untuk mempelajari selengkapnya, lihat Contoh Kasus Penggunaan Amazon A2I.

Pilih tab di tabel berikut untuk melihat diagram yang menggambarkan cara kerja Amazon A2I dengan setiap jenis tugas. Pilih halaman jenis tugas menggunakan tautan dalam daftar sebelumnya untuk mempelajari lebih lanjut tentang jenis tugas tersebut.

Amazon Textract – Key-value pair extraction

Gambar ini menggambarkan alur kerja bawaan Amazon A2I dengan Amazon Textract. Di sebelah kiri, sumber daya yang diperlukan untuk membuat alur kerja tinjauan manusia Amazon Textract digambarkan: bucket Amazon S3, kondisi aktivasi, templat tugas pekerja, dan tim kerja. Sumber daya ini digunakan untuk membuat alur kerja tinjauan manusia, atau definisi alur. Panah mengarah ke langkah berikutnya dalam alur kerja: menggunakan Amazon Textract untuk mengonfigurasi loop manusia dengan alur kerja tinjauan manusia. Panah kedua menunjuk langsung dari langkah ini ke langkah di mana kondisi aktivasi yang ditentukan dalam alur kerja tinjauan manusia terpenuhi. Ini memulai penciptaan loop manusia. Di sebelah kanan gambar, loop manusia digambarkan dalam tiga langkah: 1) UI pekerja dan alat yang dihasilkan dan tugas dibuat tersedia untuk pekerja, 2) pekerja meninjau data masukan, dan akhirnya, 3) hasil disimpan di Amazon S3.

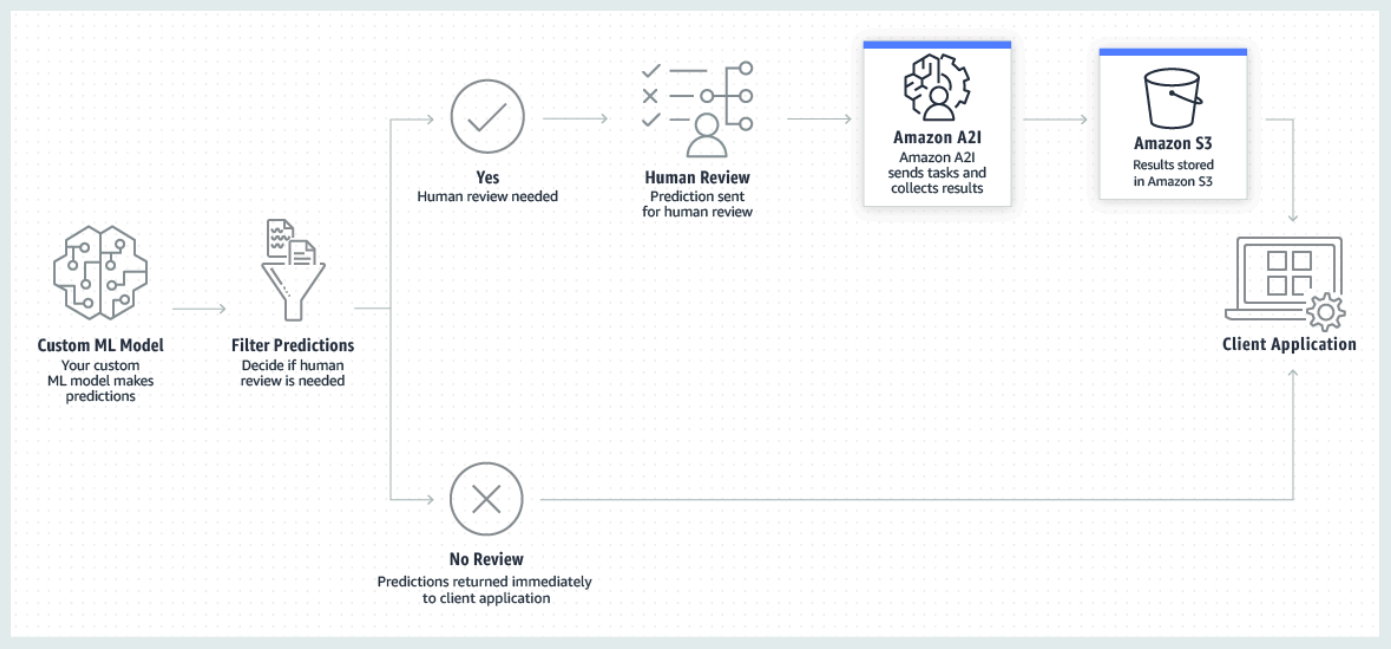


Amazon Rekognition – Image moderation

Gambar ini menggambarkan alur kerja bawaan Amazon A2I dengan Amazon Rekognition. Di sebelah kiri, sumber daya yang diperlukan untuk membuat alur kerja tinjauan manusia Amazon Rekognition digambarkan: bucket Amazon S3, kondisi aktivasi, templat tugas pekerja, dan tim kerja. Sumber daya ini digunakan untuk membuat alur kerja tinjauan manusia, atau definisi alur. Panah mengarah ke langkah berikutnya dalam alur kerja: menggunakan Amazon Rekognition untuk mengonfigurasi loop manusia dengan alur kerja tinjauan manusia. Panah kedua menunjuk langsung dari langkah ini ke langkah di mana kondisi aktivasi yang ditentukan dalam alur kerja tinjauan manusia terpenuhi. Ini memulai penciptaan loop manusia. Di sebelah kanan gambar, loop manusia digambarkan dalam tiga langkah: 1) UI pekerja dan alat yang dihasilkan dan tugas dibuat tersedia untuk pekerja, 2) pekerja meninjau data masukan, dan akhirnya, 3) hasil disimpan di Amazon S3.

Custom Task Type

Gambar berikut menggambarkan alur kerja kustom Amazon A2I. Model ML khusus digunakan untuk menghasilkan prediksi. Aplikasi klien menyaring prediksi ini menggunakan kriteria yang ditetapkan pengguna dan menentukan apakah peninjauan manusia diperlukan. Jika demikian, prediksi ini dikirim ke Amazon A2I untuk peninjauan manusia. Amazon A2I mengumpulkan hasil tinjauan manusia di Amazon S3, yang dapat diakses oleh aplikasi klien. Jika filter menentukan bahwa tidak diperlukan peninjauan manusia, prediksi dapat diumpankan langsung ke aplikasi klien.



Alur Kerja Tinjauan Manusia (Definisi Aliran)

Anda menggunakan alur kerja peninjauan manusia untuk menentukan tim kerja manusia, menyiapkan UI pekerja menggunakan templat tugas pekerja, dan untuk memberikan informasi tentang bagaimana pekerja harus menyelesaikan tugas peninjauan.

Untuk jenis tugas bawaan, Anda juga menggunakan alur kerja tinjauan manusia untuk mengidentifikasi kondisi di mana loop manusia dimulai. Misalnya, Amazon Rekognition dapat melakukan moderasi konten gambar menggunakan machine learning. Anda dapat menggunakan alur kerja peninjauan manusia untuk menentukan bahwa gambar dikirim ke manusia untuk peninjauan moderasi konten jika kepercayaan Amazon Rekognition terlalu rendah.

Anda dapat menggunakan alur kerja tinjauan manusia untuk membuat beberapa loop manusia.

Anda dapat membuat definisi aliran di SageMaker konsol atau dengan SageMaker API. Untuk mempelajari lebih lanjut tentang kedua opsi ini, lihatBuat Alur Kerja Tinjauan Manusia.

Tim Kerja

Tim kerja adalah sekelompok pekerja manusia yang Anda kirimkan tugas peninjauan manusia Anda.

Saat membuat alur kerja tinjauan manusia, Anda menentukan satu tim kerja.

Tim kerja Anda dapat berasal dari tenaga kerja Amazon Mechanical Turk, tenaga kerja yang dikelola vendor, atau tenaga kerja pribadi Anda sendiri. Saat Anda menggunakan tenaga kerja pribadi, Anda

dapat membuat beberapa tim kerja. Setiap tim kerja dapat digunakan dalam beberapa alur kerja tinjauan manusia. Untuk mempelajari cara membuat tenaga kerja dan tim kerja, lihatMembuat dan Mengelola Tenaga Kerja.

Template Tugas Pekerja dan UI Tugas Manusia

Anda menggunakan template tugas pekerja untuk membuat UI pekerja (UI tugas manusia) untuk tugas peninjauan manusia.

UI tugas manusia menampilkan data input Anda, seperti dokumen atau gambar, dan instruksi kepada pekerja. Ini juga menyediakan alat interaktif yang digunakan pekerja untuk menyelesaikan tugas Anda.

Untuk jenis tugas bawaan, Anda harus menggunakan template tugas pekerja Amazon A2I yang disediakan untuk jenis tugas tersebut.

Loop Manusia

Sebuah loop manusia digunakan untuk membuat pekerjaan review manusia tunggal. Untuk setiap pekerjaan review manusia, Anda dapat memilih jumlah pekerja yang dikirim tugas untuk meninjau objek data tunggal. Misalnya, jika Anda menetapkan jumlah pekerja per objek3 untuk pekerjaan pelabelan klasifikasi gambar, tiga pekerja mengklasifikasikan setiap gambar masukan. Meningkatkan jumlah pekerja per objek dapat meningkatkan akurasi label.

Sebuah loop manusia dibuat menggunakan alur kerja review manusia sebagai berikut:

- Untuk tipe tugas bawaan, kondisi yang ditentukan dalam alur kerja tinjauan manusia menentukan kapan loop manusia dibuat.
- Tugas peninjauan manusia dikirim ke tim kerja yang ditentukan dalam alur kerja peninjauan manusia.
- Template tugas pekerja yang ditentukan dalam alur kerja tinjauan manusia digunakan untuk membuat UI tugas manusia.

Kapan loop manusia bisa dibuat?

Saat Anda menggunakan salah satu jenis tugas bawaan,AWS layanan terkait akan membuat dan memulai loop manusia atas nama Anda ketika kondisi yang ditentukan dalam alur kerja tinjauan manusia terpenuhi. Sebagai contoh:

- Saat Anda menggunakan Augmented AI dengan Amazon Textract, Anda dapat mengintegrasikan Amazon A2I ke dalam tugas peninjauan dokumen menggunakan operasi API`AnalyzeDocument`. Loop manusia dibuat setiap kali Amazon Textract mengembalikan inferensi tentang pasangan nilai kunci yang memenuhi ketentuan yang Anda tentukan dalam alur kerja peninjauan manusia.
- Saat Anda menggunakan Augmented AI dengan Amazon Rekognition, Anda dapat mengintegrasikan Amazon A2I ke dalam tugas moderasi gambar menggunakan operasi API`DetectModerationLabels`. Loop manusia dibuat setiap kali Amazon Rekognition mengembalikan inferensi tentang konten gambar yang memenuhi ketentuan yang Anda tentukan dalam alur kerja peninjauan manusia.

Saat menggunakan jenis tugas khusus, Anda memulai loop manusia menggunakan Amazon Augmented AI Runtime API. Ketika Anda meneleponStartHumanLoop dalam aplikasi kustom Anda, tugas dikirim ke pengulas manusia.

Untuk mempelajari cara membuat dan memulai loop manusia, lihatMembuat dan Memulai Loop Manusia.

Untuk menghasilkan sumber daya ini dan membuat alur kerja tinjauan manusia, Amazon A2I mengintegrasikan beberapa API, termasuk Amazon Augmented AI Runtime Model, SageMaker API, dan API yang terkait dengan jenis tugas Anda. Untuk mempelajari selengkapnya, lihat Menggunakan API di Amazon Augmented AI.

> **Note**
>
> AWSKetersediaan wilayah mungkin berbeda saat Anda menggunakan Augmented AI denganAWS layanan lain, seperti Amazon Textract. Buat sumber daya Augmented AI diAWS Wilayah yang sama dengan yang Anda gunakan untuk berinteraksi denganAWS layanan tersebut. Untuk ketersediaanAWS Wilayah untuk semua layanan, lihat Tabel Wilayah.

Prasyarat untuk Menggunakan Augmented AI

Amazon A2I menggunakan sumber daya dalam IAM, SageMaker, dan Amazon S3 untuk membuat dan menjalankan alur kerja peninjauan manusia Anda. Anda dapat membuat beberapa sumber daya ini di konsol Amazon A2I saat membuat alur kerja peninjauan manusia. Untuk mempelajari caranya, lihat Tutorial: Mulai Konsol Amazon A2I.

Untuk menggunakan Amazon A2I, Anda memerlukan sumber daya berikut:

- Satu atau beberapa bucket Amazon S3 diAWS Wilayah yang sama dengan alur kerja untuk data input dan output Anda. Untuk membuat bucket, ikuti petunjuk di Buat Bucket di Panduan Pengguna Amazon Simple Storage Console.
- Peran IAM dengan izin yang diperlukan untuk membuat alur kerja tinjauan manusia dan pengguna IAM atau peran dengan izin untuk mengakses Augmented AI. Untuk informasi selengkapnya, lihat Izin dan Keamanan di Amazon Augmented AI.
- Tenaga kerja publik, swasta, atau vendor untuk alur kerja peninjauan manusia Anda. Jika Anda berencana untuk menggunakan tenaga kerja pribadi, Anda perlu mengaturnya sebelumnya diAWS Wilayah yang sama dengan alur kerja Amazon A2I Anda. Untuk mempelajari lebih banyak tentang jenis tenaga kerja ini, lihatMembuat dan Mengelola Tenaga Kerja.

> **Important**
>
> Untuk mempelajari program kepatuhan yang mencakup Amazon Augmented AI, lihat AWSLayanan di Cakupan Layanan berdasarkan Program Kepatuhan. Jika Anda menggunakan Amazon Augmented AI bersama denganAWS layanan lain (seperti Amazon Rekognition dan Amazon Textract), perhatikan bahwa Amazon Augmented AI mungkin tidak berada dalam lingkup untuk program kepatuhan yang sama dengan layanan lainnya. Anda bertanggung jawab atas cara Anda menggunakan Amazon Augmented AI, termasuk memahami cara layanan memproses atau menyimpan data pelanggan dan dampak apa pun terhadap kepatuhan lingkungan data Anda. Anda harus mendiskusikan sasaran dan sasaran beban kerja Anda dengan timAWS akun Anda; mereka dapat membantu Anda mengevaluasi apakah layanan tersebut cocok untuk kasus penggunaan dan arsitektur yang Anda usulkan.

Tutorial: Mulai Konsol Amazon A2I

Tutorial berikut menunjukkan kepada Anda cara memulai menggunakan Amazon A2I di konsol Amazon A2I.

Tutorial ini memberi Anda opsi untuk menggunakan Augmented AI dengan Amazon Textract untuk peninjauan dokumen atau Amazon Rekognition untuk peninjauan konten gambar.

Prasyarat

Untuk mulai menggunakan Amazon A2I, selesaikan prasyarat berikut.

- Buat bucket Amazon S3 di yangAWS Wilayah yang sama dengan data input dan output Anda. Misalnya, jika Anda menggunakan Amazon Textract A2I dengan Amazon Teast-1, buat bucket Anda di us-east-1, buat bucket Anda di us-east-1. Untuk membuat bucket, ikuti petunjuk di Buat Bucket di Panduan Pengguna Amazon Simple Storage Console.
- Lakukan salah satu dari berikut:
  - Jika Anda ingin menyelesaikan tutorial menggunakan Amazon Textract Teastract, unduh dokumen sampel ini dan letakkan di bucket Amazon S3 Anda.
  - Jika Anda ingin menyelesaikan tutorial menggunakan Amazon Rekognition, unduh gambar ini dan letakkan di bucket Amazon S3 Anda.

> **Note**
>
> Konsol Amazon A2I disematkan di SageMaker konsol.

Langkah 1: Buat Tim Kerja

Pertama, buat tim kerja di konsol Amazon A2I dan tambahkan diri Anda sebagai pekerja sehingga Anda dapat melihat pratinjau tugas tinjauan pekerja.

> **Important**
>
> Tutorial ini menggunakan tim kerja pribadi. Tenaga kerja pribadi Amazon A2I dikonfigurasi di area Ground Truth SageMaker konsol dan dibagi antara Amazon A2I dan Ground Truth.

Untuk membuat tenaga kerja privat menggunakan email pekerja

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilih Labeling workforce di bawah Ground Truth.
3. Pilih Pribadi, lalu pilih Buat tim pribadi.
4. Pilih Undang pekerja baru melalui email.
5. Untuk tutorial ini, masukkan email Anda dan yang lain yang Anda inginkan untuk dapat melihat pratinjau UI tugas manusia. Anda dapat menempelkan atau mengetikkan daftar hingga 50 alamat email, dipisahkan dengan koma, ke dalam kotak alamat email.
6. Masukkan nama organisasi dan email kontak.

7. Secara opsional, pilih topik Amazon SNS yang akan berlangganan tim sehingga pekerja diberi tahu melalui email ketika pekerjaan pelabelan Ground Truth baru tersedia. Notifikasi Amazon SNS didukung oleh Ground Truth dan tidak didukung oleh Augmented AI. Jika Anda berlangganan pekerja ke pemberitahuan Amazon SNS, mereka hanya menerima pemberitahuan tentang pekerjaan pelabelan Ground Truth. Mereka tidak menerima pemberitahuan tentang tugas Augmented AI.
8. Pilih Buat tim pribadi.

Jika Anda menambahkan diri Anda ke tim kerja pribadi, Anda menerima email dari `no-reply@verificationemail.com` dengan informasi login. Gunakan tautan di email ini untuk mengatur ulang kata sandi Anda dan masuk ke portal pekerja Anda. Di sinilah tugas peninjauan manusia Anda muncul saat Anda membuat loop manusia.

Langkah 2: Buat WorkWorkWorkReview Manusia

Pada langkah ini, Anda membuat alur kerja tinjauan manusia. Setiap alur kerja tinjauan manusia dibuat untuk jenis tugas tertentu. Tutorial ini memungkinkan Anda untuk memilih antara jenis tugas bawaan: Amazon Rekognition dan Amazon Textract.

Untuk membuat alur kerja tinjauan manusia:

1. Buka Augmented AI console di https://console.aws.amazon.com/a2i untuk mengakses halaman Alur kerja ulasan Manusia.
2. Pilih Buat alur kerja tinjauan manusia.
3. Dalam pengaturan Workflow, masukkan alur kerja Name, S3 bucket, dan peran IAM yang Anda buat untuk tutorial ini, dengan kebijakanAWS terkelola`AmazonAugmentedAIIntegratedAPIAccess` terlampir.
4. Untuk Jenis tugas, pilih Textract - Ekstraksi pasangan nilai kunci atau Rekognition - Moderasi gambar.
5. Pilih jenis tugas yang Anda pilih dari tabel berikut untuk petunjuk untuk jenis tugas tersebut.

   Amazon Textract – Key-value pair extraction

   1. Pilih Picu ulasan manusia untuk kunci formulir tertentu berdasarkan skor kepercayaan kunci formulir atau ketika kunci formulir tertentu hilang.

   2. Untuk Nama kunci, masukkan`Mail Address`.

3. Tetapkan ambang batas kepercayaan identifikasi antara`0` dan`99`.

4. Tetapkan ambang batas kepercayaan kualifikasi antara`0` dan`99`.

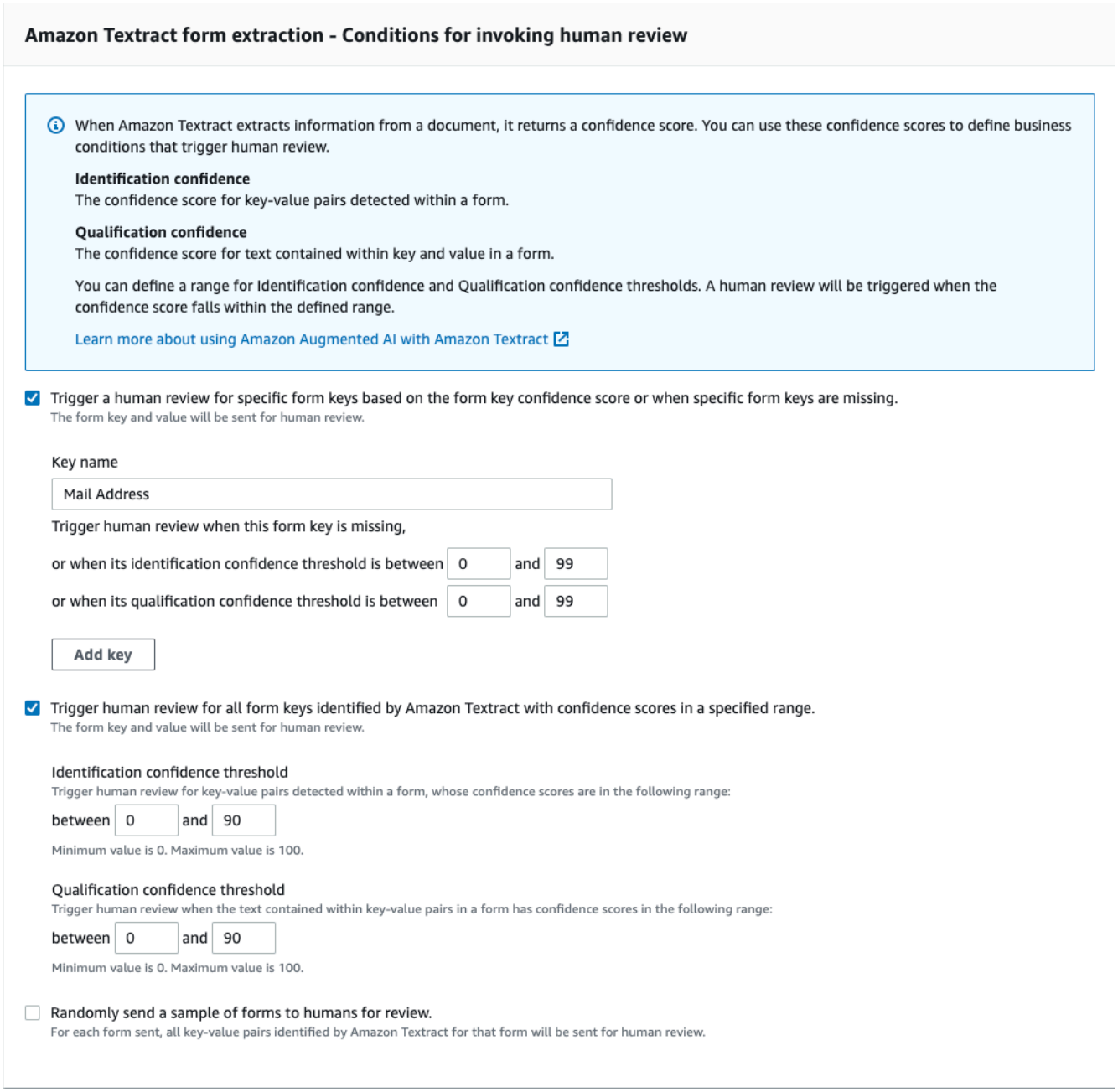
5. Pilih Trigger ulasan manusia untuk semua kunci formulir yang diidentifikasi oleh Amazon Textract dengan skor kepercayaan dalam rentang tertentu.

6. Tetapkan ambang batas kepercayaan identifikasi antara`0` dan`90`.

7. Tetapkan ambang batas kepercayaan kualifikasi antara`0` dan`90`.

Ini memulai tinjauan manusia jika Amazon Textract mengembalikan skor kepercayaan diri yang kurang dari`99` untuk`Mail Address` dan kuncinya, atau jika ia mengembalikan skor kepercayaan kurang dari`90` pasangan nilai kunci yang terdeteksi dalam dokumen.

Gambar berikut menunjukkan ekstraksi formulir Amazon Textract - Ketentuan untuk menerapkan bagian peninjauan manusia dari konsol Amazon A2I. Pada gambar, kotak centang untuk dua jenis pemicu yang dijelaskan dalam paragraf persidangan dicentang, dan`Mail Address` digunakan sebagai nama kunci untuk pemicu pertama. Ambang batas kepercayaan identifikasi didefinisikan menggunakan skor kepercayaan untuk pasangan kunci-nilai mendeteksi dalam bentuk dan ditetapkan antara 0 dan 99. Ambang batas kepercayaan kualifikasi didefinisikan menggunakan skor kepercayaan untuk teks yang terkandung dalam kunci dan nilai dalam bentuk dan ditetapkan antara 0 dan 99.



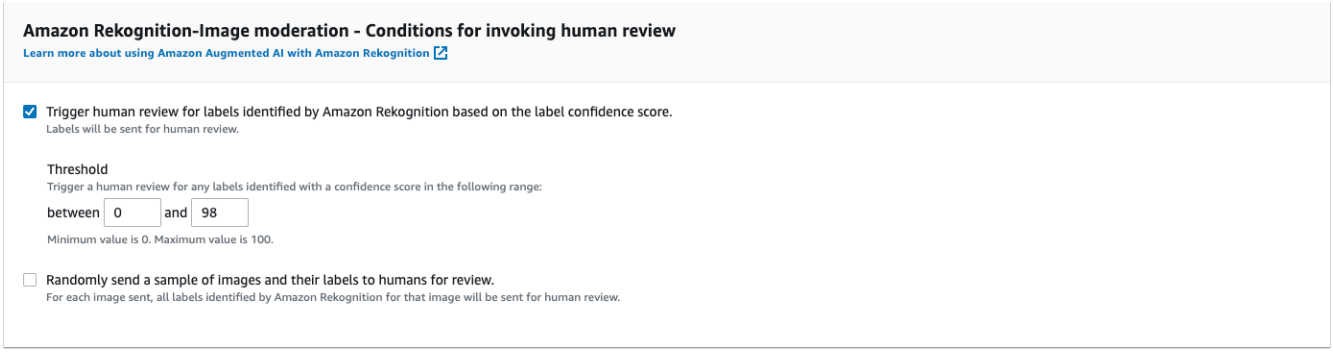
Amazon Rekognition – Image moderation

1. Pilih Trigger review manusia untuk label yang diidentifikasi oleh Amazon Rekognition berdasarkan skor kepercayaan label.

2. Mengatur Threshold antara0 dan98.

Ini memulai peninjauan manusia jika Amazon Rekognition mengembalikan skor kepercayaan yang kurang dari98 pekerjaan moderasi gambar.

Gambar berikut menunjukkan bagaimana Anda dapat memilih ulasan Trigger human untuk label yang diidentifikasi oleh Amazon Rekognition berdasarkan opsi skor kepercayaan label dan memasukkan Threshold antara 0 dan 98 di konsol Amazon A2I.



6. Di bawah Pembuatan template tugas Pekerja, pilih Buat dari template default.
7. Masukkan nama Template.
8. Di bidang Deskripsi tugas, masukkan teks berikut:

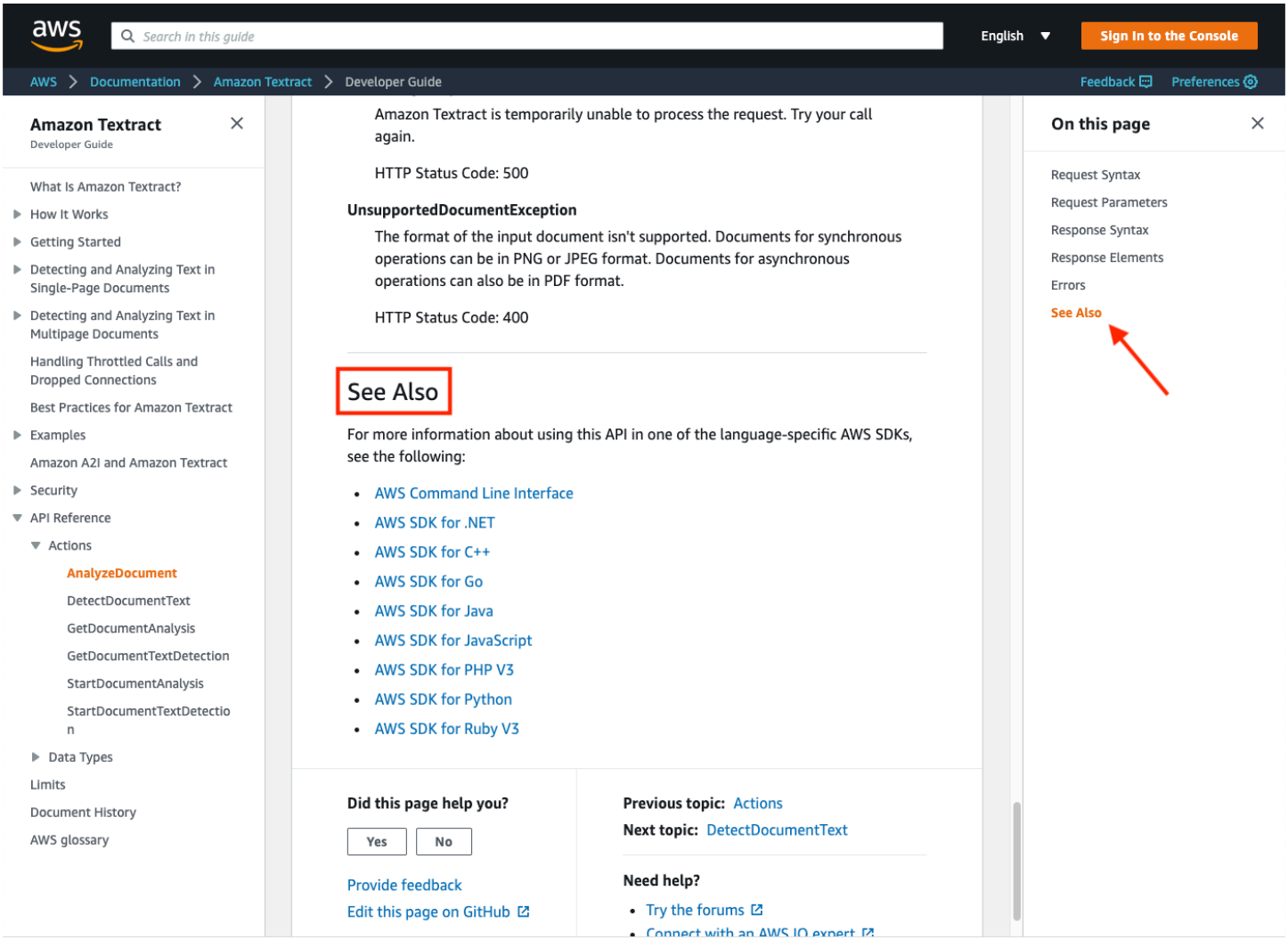
   ```
   Read the instructions carefully and complete the task.
   ```

9. Di bawah Pekerja, pilih Pribadi.
10. Pilih tim privat yang Anda buat.
11. Pilih Create (Buat).

Setelah alur kerja peninjauan manusia Anda dibuat, itu muncul di tabel di halaman Alur kerja ulasan Manusia. Ketika Status adalah`Active`, salin dan simpan ARN alur kerja. Anda memerlukannya untuk langkah berikutnya.

Langkah 3: Mulai Loop Manusia

Anda harus menggunakan operasi API untuk memulai loop manusia. Ada berbagai SDK khusus bahasa yang dapat Anda gunakan untuk berinteraksi dengan operasi API ini. Untuk melihat dokumentasi untuk masing-masing SDK ini, lihat bagian Lihat Juga dalam dokumentasi API, seperti yang ditunjukkan pada gambar berikut.



Dalam tutorial ini, Anda menggunakan salah satu API berikut:

- Jika Anda memilih jenis tugas Amazon Textract, Anda menggunakan`AnalyzeDocument` operasi.
- Jika Anda memilih jenis tugas Amazon Rekognition, Anda menggunakan`DetectModerationLabels` operasi.

Anda dapat berinteraksi dengan API ini menggunakan instance SageMaker notebook (direkomendasikan untuk pengguna baru) atauAWS Command Line Interface (AWS CLI). Pilih salah satu dari berikut untuk mempelajari lebih banyak tentang opsi ini:

- Untuk mempelajari lebih lanjut tentang dan menyiapkan instance notebook, lihatInstans SageMaker Notebook Amazon.
- Untuk mempelajari lebih lanjut tentang dan mulai menggunakanAWS CLI, lihat Apa itu Antarmuka BarisAWS Perintah? dalam PanduanAWS Command Line Interface Pengguna.

Pilih jenis tugas Anda dalam tabel berikut untuk melihat contoh permintaan Amazon Textract dan Amazon Rekognition menggunakanAWS SDK for Python (Boto3).

Amazon Textract – Key-value pair extraction

Contoh berikutAWS SDK for Python (Boto3) menggunakan panggilan`analyze_document` us-west-2. Ganti teks merah yang dicetak miring dengan sumber daya Anda. Sertakan `DataAttributes`parameter jika Anda menggunakan tenaga kerja Amazon Mechanical Turk. Untuk informasi lebih lanjut, lihat`analyze_document` dokumentasi di ReferensiAWS SDK for Python (Boto) API.

```
   response = client.analyze_document(
         Document={
                "S3Object": {
                    "Bucket": "AWSDOC-EXAMPLE-BUCKET",
                    "Name": "document-name.pdf"
                }
         },
         HumanLoopConfig={
            "FlowDefinitionArn":"arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
            "HumanLoopName":"human-loop-name",
            "DataAttributes" : {
                "ContentClassifiers":
["FreeOfPersonallyIdentifiableInformation","FreeOfAdultContent"]
            }
         },
         FeatureTypes=["TABLES", "FORMS"])
```

Amazon Rekognition – Image moderation

Contoh berikutAWS SDK for Python (Boto3) menggunakan panggilan`detect_moderation_labels` us-west-2. Ganti teks merah yang dicetak miring dengan sumber daya Anda. Sertakan `DataAttributes`parameter jika Anda menggunakan tenaga kerja Amazon Mechanical Turk. Untuk informasi lebih lanjut, lihat`detect_moderation_labels` dokumentasi di ReferensiAWS SDK for Python (Boto) API.

```
   response = client.detect_moderation_labels(
         Image={
```

```
                "S3Object":{
                    "Bucket": "AWSDOC-EXAMPLE-BUCKET",
                    "Name": "image-name.png"
                }
            },
            HumanLoopConfig={
               "FlowDefinitionArn":"arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
               "HumanLoopName":"human-loop-name",
               "DataAttributes":{
                    ContentClassifiers:
["FreeOfPersonallyIdentifiableInformation"|"FreeOfAdultContent"]
                }
             })
```

Langkah 4: Lihat Status Loop Manusia di Konsol

Saat memulai loop manusia, Anda dapat melihat statusnya di konsol Amazon A2I.

Untuk melihat status loop manusia

1. Buka Augmented AI console di https://console.aws.amazon.com/a2i untuk mengakses halaman Alur kerja ulasan Manusia.
2. Pilih alur kerja tinjauan manusia yang Anda gunakan untuk memulai loop manusia.
3. Di bagian Human loop, Anda dapat melihat loop manusia Anda. Lihat statusnya di kolom Status.

Langkah 5: Unduh Data Output

Data keluaran Anda disimpan dalam bucket Amazon S3 yang Anda tentukan saat membuat alur kerja tinjauan manusia.

Untuk melihat data keluaran Amazon A2I Anda

1. Buka konsol Amazon S3.
2. Pilih bucket Amazon S3 yang Anda tentukan saat membuat alur kerja peninjauan manusia pada langkah 2 dari contoh ini.
3. Dimulai dengan folder yang dinamai menurut alur kerja tinjauan manusia Anda, navigasikan ke data keluaran Anda dengan memilih folder dengan konvensi penamaan berikut:

```
s3://output-bucket-specified-in-human-review-workflow/human-review-workflow-
name/YYYY/MM/DD/hh/mm/ss/human-loop-name/output.json
```

4. Pilihoutput.json dan pilih Unduh.

Tutorial: Mulai Menggunakan API Amazon A2I

Tutorial ini menjelaskan operasi API yang dapat Anda gunakan untuk mulai menggunakan Amazon A2I.

Untuk menggunakan Notebook Jupyter untuk menjalankan operasi ini, pilih Notebook Jupyter dariContoh Kasus Penggunaan Amazon A2I dan gunakanGunakan Instans Notebook SageMaker dengan Notebook Amazon A2I Jupyter untuk mempelajari cara menggunakannya dalam instance SageMaker notebook.

Untuk mempelajari lebih lanjut tentang operasi API yang dapat Anda gunakan dengan Amazon A2I, lihatMenggunakan API di Amazon Augmented AI.

Buat Tim Kerja Pribadi

Anda dapat membuat tim kerja pribadi dan menambahkan diri Anda sebagai pekerja sehingga Anda dapat melihat pratinjau Amazon A2I.

Jika Anda tidak terbiasa dengan Amazon Cognito, kami menyarankan Anda menggunakan SageMaker konsol untuk membuat tenaga kerja privat dan menambahkan diri Anda sebagai pekerja privat. Untuk petunjuk, lihat Langkah 1: Buat Tim Kerja.

Jika Anda terbiasa dengan Amazon Cognito, Anda dapat menggunakan instruksi berikut untuk membuat tim kerja pribadi menggunakan SageMaker API. Setelah Anda membuat tim kerja, perhatikan tim kerja ARN (WorkteamArn).

Untuk mempelajari lebih banyak tentang tenaga kerja privat dan konfigurasi lain yang tersedia, lihatMenggunakan Tenaga Kerja Pribadi.

Buat tenaga kerja privat

Jika Anda belum membuat tenaga kerja pribadi, Anda dapat melakukannya menggunakan pangkalan pengguna Amazon Cognito. Pastikan bahwa Anda telah menambahkan diri Anda ke

kolam pengguna ini. Anda dapat membuat tim kerja pribadi menggunakanAWS SDK for Python (Boto3)create_workforce fungsi tersebut. Untuk SDK khusus bahasa lainnya, lihat daftar di CreateWorkforce.

```
response = client.create_workforce(
    CognitoConfig={
        "UserPool": "Pool_ID",
        "ClientId": "app-client-id"
    },
    WorkforceName="workforce-name"
)
```

Buat tim kerja privat

Setelah Anda membuat tenaga kerja pribadi diAWS Wilayah untuk mengonfigurasi dan memulai loop manusia, Anda dapat membuat tim kerja pribadi menggunakanAWS SDK for Python (Boto3)create_workteam fungsi tersebut. Untuk SDK khusus bahasa lainnya, lihat daftar diCreateWorkteam.

```
response = client.create_workteam(
    WorkteamName="work-team-name",
    WorkforceName= "workforce-name",
    MemberDefinitions=[
        {
            "CognitoMemberDefinition": {
                "UserPool": "<aws-region>_ID",
                "UserGroup": "user-group",
                "ClientId": "app-client-id"
            },
        }
    ]
)
```

Akses tim kerja Anda ARN sebagai berikut:

```
workteamArn = response["WorkteamArn"]
```

Cantumkan tim kerja pribadi di akun Anda

Jika Anda telah membuat tim kerja pribadi, Anda dapat mencantumkan semua tim kerja diAWS Wilayah tertentu di akun Anda menggunakanAWS SDK for Python (Boto3)`list_workteams` fungsi tersebut. Untuk SDK khusus bahasa lainnya, lihat daftar di`ListWorkteams`.

```
response = client.list_workteams()
```

Jika Anda memiliki banyak tim kerja di akun Anda, Anda mungkin ingin menggunakan`MaxResults`,`SortBy`, dan`NameContains` memfilter hasil Anda.

Buat Alur Kerja Tinjauan Manusia

Anda dapat membuat alur kerja peninjauan manusia menggunakan`CreateFlowDefinition` operasi Amazon A2I. Sebelum membuat alur kerja peninjauan manusia, Anda perlu membuat UI tugas manusia. Anda dapat melakukan ini dengan`CreateHumanTaskUi` operasi.

Jika Anda menggunakan Amazon A2I dengan integrasi Amazon Textract atau Amazon Rekognition, Anda dapat menentukan kondisi aktivasi menggunakan JSON.

Buat UI Tugas Manusia

Jika Anda membuat alur kerja tinjauan manusia untuk digunakan dengan integrasi Amazon Textract atau Amazon Rekognition, Anda perlu menggunakan dan memodifikasi templat tugas pekerja yang sudah dibuat sebelumnya. Untuk semua integrasi kustom, Anda dapat menggunakan template tugas pekerja kustom Anda sendiri. Gunakan tabel berikut untuk mempelajari cara membuat UI tugas manusia menggunakan template tugas pekerja untuk dua integrasi bawaan. Ganti template dengan template Anda sendiri untuk menyesuaikan permintaan ini.

Amazon Textract – Key-value pair extraction

Untuk mempelajari lebih banyak tentang template ini, lihatContoh kustom untuk Amazon Textract.

```
template = r"""
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
{% capture s3_uri %}http://s3.amazonaws.com/
{{ task.input.aiServiceRequest.document.s3Object.bucket }}/
{{ task.input.aiServiceRequest.document.s3Object.name }}{% endcapture %}
<crowd-form>
  <crowd-textract-analyze-document
      src="{{ s3_uri | grant_read_access }}"
      initial-value="{{ task.input.selectedAiServiceResponse.blocks }}"
```

```
      header="Review the key-value pairs listed on the right and correct them if
 they don"t match the following document."
      no-key-edit=""
      no-geometry-edit=""
      keys="{{ task.input.humanLoopContext.importantFormKeys }}"
      block-types='["KEY_VALUE_SET"]'>
    <short-instructions header="Instructions">
        <p>Click on a key-value block to highlight the corresponding key-value pair
 in the document.
        </p><p><br></p>
        <p>If it is a valid key-value pair, review the content for the value. If the
 content is incorrect, correct it.
        </p><p><br></p>
        <p>The text of the value is incorrect, correct it.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/correct-value-
text.png">
        </p><p><br></p>
        <p>A wrong value is identified, correct it.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/correct-value.png">
        </p><p><br></p>
        <p>If it is not a valid key-value relationship, choose No.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/not-a-key-value-
pair.png">
        </p><p><br></p>
        <p>If you can’t find the key in the document, choose Key not found.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/key-is-not-
found.png">
        </p><p><br></p>
        <p>If the content of a field is empty, choose Value is blank.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/value-is-
blank.png">
        </p><p><br></p>
        <p><strong>Examples</strong></p>
        <p>Key and value are often displayed next or below to each other.
        </p><p><br></p>
        <p>Key and value displayed in one line.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/sample-key-value-
pair-1.png">
        </p><p><br></p>
        <p>Key and value displayed in two lines.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/sample-key-value-
pair-2.png">
        </p><p><br></p>
```

```
        <p>If the content of the value has multiple lines, enter all the text
 without line break.
        Include all value text even if it extends beyond the highlight box.</p>
        <p><img src="https://assets.crowd.aws/images/a2i-console/multiple-
lines.png"></p>
    </short-instructions>
    <full-instructions header="Instructions"></full-instructions>
  </crowd-textract-analyze-document>
</crowd-form>
"""
```

Amazon Rekognition – Image moderation

Untuk mempelajari lebih banyak tentang template ini, lihatContoh kustom untuk Amazon Rekognition.

```
template = r"""
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
{% capture s3_uri %}http://s3.amazonaws.com/
{{ task.input.aiServiceRequest.image.s3Object.bucket }}/
{{ task.input.aiServiceRequest.image.s3Object.name }}{% endcapture %}

<crowd-form>
  <crowd-rekognition-detect-moderation-labels
    categories='[
      {% for label in task.input.selectedAiServiceResponse.moderationLabels %}
        {
          name: "{{ label.name }}",
          parentName: "{{ label.parentName }}",
        },
      {% endfor %}
    ]'
    src="{{ s3_uri | grant_read_access }}"
    header="Review the image and choose all applicable categories."
  >
    <short-instructions header="Instructions">
      <style>
        .instructions {
          white-space: pre-wrap;
        }
      </style>
      <p class="instructions">Review the image and choose all applicable categories.
If no categories apply, choose None.
```

```
<b>Nudity</b>
Visuals depicting nude male or female person or persons

<b>Partial Nudity</b>
Visuals depicting covered up nudity, for example using hands or pose

<b>Revealing Clothes</b>
Visuals depicting revealing clothes and poses

<b>Physical Violence</b>
Visuals depicting violent physical assault, such as kicking or punching

<b>Weapon Violence</b>
Visuals depicting violence using weapons like firearms or blades, such as shooting

<b>Weapons</b>
Visuals depicting weapons like firearms and blades
    </short-instructions>

    <full-instructions header="Instructions"></full-instructions>
  </crowd-rekognition-detect-moderation-labels>
</crowd-form>"""
```

Custom Integration

Berikut ini adalah contoh template yang dapat digunakan dalam integrasi kustom. Template ini digunakan dalam notebook ini, menunjukkan integrasi kustom dengan Amazon Comprehend.

```
template = r"""
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
    <crowd-classifier
      name="sentiment"
      categories='["Positive", "Negative", "Neutral", "Mixed"]'
      initial-value="{{ task.input.initialValue }}"
      header="What sentiment does this text convey?"
    >
      <classification-target>
        {{ task.input.taskObject }}
      </classification-target>
```

```
      <full-instructions header="Sentiment Analysis Instructions">
        <p><strong>Positive</strong> sentiment include: joy, excitement, delight</p>
        <p><strong>Negative</strong> sentiment include: anger, sarcasm, anxiety</p>
        <p><strong>Neutral</strong>: neither positive or negative, such as stating a
 fact</p>
        <p><strong>Mixed</strong>: when the sentiment is mixed</p>
      </full-instructions>

      <short-instructions>
       Choose the primary sentiment that is expressed by the text.
      </short-instructions>
    </crowd-classifier>
</crowd-form>
"""
```

Menggunakan template yang ditentukan di atas, Anda dapat membuat template menggunakanAWS SDK for Python (Boto3)`create_human_task_ui` fungsi. Untuk SDK khusus bahasa lainnya, lihat daftar di`CreateHumanTaskUi`.

```
response = client.create_human_task_ui(
    HumanTaskUiName="human-task-ui-name",
    UiTemplate={
        "Content": template
    }
)
```

Elemen respon ini berisi tugas manusia UI ARN. Simpan ini sebagai berikut:

```
humanTaskUiArn = response["HumanTaskUiArn"]
```

Buat JSON untuk menentukan kondisi aktivasi

Untuk integrasi bawaan Amazon Textract dan Amazon Rekognition, Anda dapat menyimpan kondisi aktivasi dalam objek JSON dan menggunakannya dalam`CreateFlowDefinition` permintaan Anda.

Selanjutnya, pilih tab untuk melihat contoh kondisi aktivasi yang dapat Anda gunakan untuk integrasi bawaan ini. Untuk informasi tambahan tentang opsi kondisi aktivasi, lihatSkema JSON untuk Kondisi Aktivasi Loop Manusia di Amazon Augmented AI.

Amazon Textract – Key-value pair extraction

Contoh ini menentukan kondisi untuk kunci tertentu (sepertiMail address) dalam dokumen. Jika kepercayaan Amazon Textract berada di luar ambang batas yang ditetapkan di sini, dokumen dikirim ke manusia untuk ditinjau, dengan kunci spesifik yang memulai loop manusia diminta ke pekerja.

```
     import json

     humanLoopActivationConditions = json.dumps(
       {
           "Conditions": [
               {
                 "Or": [

                   {
                       "ConditionType": "ImportantFormKeyConfidenceCheck",
                       "ConditionParameters": {
                           "ImportantFormKey": "Mail address",
                           "ImportantFormKeyAliases": ["Mail Address:","Mail
address:", "Mailing Add:","Mailing Addresses"],
                           "KeyValueBlockConfidenceLessThan": 100,
                           "WordBlockConfidenceLessThan": 100
                       }
                   },
                   {
                       "ConditionType": "MissingImportantFormKey",
                       "ConditionParameters": {
                           "ImportantFormKey": "Mail address",
                           "ImportantFormKeyAliases": ["Mail Address:","Mail
address:","Mailing Add:","Mailing Addresses"]
                       }
                   },
                   {
                       "ConditionType": "ImportantFormKeyConfidenceCheck",
                       "ConditionParameters": {
                           "ImportantFormKey": "Phone Number",
                           "ImportantFormKeyAliases": ["Phone number:", "Phone
No.:", "Number:"],
                           "KeyValueBlockConfidenceLessThan": 100,
                           "WordBlockConfidenceLessThan": 100
                       }
```

```
                    },
                    {
                      "ConditionType": "ImportantFormKeyConfidenceCheck",
                      "ConditionParameters": {
                        "ImportantFormKey": "*",
                        "KeyValueBlockConfidenceLessThan": 100,
                        "WordBlockConfidenceLessThan": 100
                      }
                    },
                    {
                      "ConditionType": "ImportantFormKeyConfidenceCheck",
                      "ConditionParameters": {
                        "ImportantFormKey": "*",
                        "KeyValueBlockConfidenceGreaterThan": 0,
                        "WordBlockConfidenceGreaterThan": 0
                      }
                    }
            ]
        }
            ]
        }
    )
```

Amazon Rekognition – Image moderation

Kondisi aktivasi loop manusia yang digunakan di sini disesuaikan dengan moderasi konten Amazon Rekognition; mereka didasarkan pada ambang batas kepercayaan untuk`Suggestive` dan label`Female Swimwear Or Underwear` moderasi.

```
        import json

        humanLoopActivationConditions = json.dumps(
        {
            "Conditions": [
                {
                  "Or": [
                    {
                        "ConditionType": "ModerationLabelConfidenceCheck",
                        "ConditionParameters": {
                            "ModerationLabelName": "Suggestive",
                            "ConfidenceLessThan": 98
                        }
```

```
                    },
                    {
                        "ConditionType": "ModerationLabelConfidenceCheck",
                        "ConditionParameters": {
                            "ModerationLabelName": "Female Swimwear Or Underwear",
                            "ConfidenceGreaterThan": 98
                        }
                    }
                ]
            }
        ]
    }
)
```

Buat alur kerja tinjauan manusia

Bagian ini memberikan contoh`CreateFlowDefinition`AWS SDK for Python (Boto3) permintaan menggunakan sumber daya yang dibuat di bagian sebelumnya. Untuk SDK khusus bahasa lainnya, lihat daftar di CreateFlowDefinition. Gunakan tab dalam tabel berikut untuk melihat permintaan untuk membuat alur kerja peninjauan manusia untuk integrasi bawaan Amazon Textract dan Amazon Rekognition.

Amazon Textract – Key-value pair extraction

Jika Anda menggunakan integrasi bawaan dengan Amazon Textract, Anda harus menentukan`"AWS/Textract/AnalyzeDocument/Forms/V1"` untuk`"AwsManagedHumanLoopRequestSource"` masuk`HumanLoopRequestSource`.

```
    response = client.create_flow_definition(
        FlowDefinitionName="human-review-workflow-name",
        HumanLoopRequestSource={
            "AwsManagedHumanLoopRequestSource": "AWS/Textract/AnalyzeDocument/Forms/
V1"
        },
        HumanLoopActivationConfig={
            "HumanLoopActivationConditionsConfig": {
                "HumanLoopActivationConditions": humanLoopActivationConditions
            }
        },
        HumanLoopConfig={
            "WorkteamArn": workteamArn,
```

```
            "HumanTaskUiArn": humanTaskUiArn,
            "TaskTitle": "Document entry review",
            "TaskDescription": "Review the document and instructions. Complete the
 task",
            "TaskCount": 1,
            "TaskAvailabilityLifetimeInSeconds": 43200,
            "TaskTimeLimitInSeconds": 3600,
            "TaskKeywords": [
                "document review",
            ],
        },
        OutputConfig={
            "S3OutputPath": "s3://DOC-EXAMPLE-BUCKET/prefix/",
        },
        RoleArn="arn:aws:iam::<account-number>:role/<role-name>",
        Tags=[
            {
                "Key": "string",
                "Value": "string"
            },
        ]
    )
```

Amazon Rekognition – Image moderation

Jika Anda menggunakan integrasi bawaan dengan Amazon Rekognition, Anda harus menentukan`"AWS/Rekognition/DetectModerationLabels/Image/V3"` untuk`"AwsManagedHumanLoopRequestSource"` masuk`HumanLoopRequestSource`.

```
    response = client.create_flow_definition(
        FlowDefinitionName="human-review-workflow-name",
        HumanLoopRequestSource={
            "AwsManagedHumanLoopRequestSource": "AWS/Rekognition/
DetectModerationLabels/Image/V3"
        },
        HumanLoopActivationConfig={
            "HumanLoopActivationConditionsConfig": {
                "HumanLoopActivationConditions": humanLoopActivationConditions
            }
        },
        HumanLoopConfig={
            "WorkteamArn": workteamArn,
```

```
            "HumanTaskUiArn": humanTaskUiArn,
            "TaskTitle": "Image content moderation",
            "TaskDescription": "Review the image and instructions. Complete the
task",
            "TaskCount": 1,
            "TaskAvailabilityLifetimeInSeconds": 43200,
            "TaskTimeLimitInSeconds": 3600,
            "TaskKeywords": [
                "content moderation",
            ],
        },
        OutputConfig={
            "S3OutputPath": "s3://DOC-EXAMPLE-BUCKET/prefix/",
        },
        RoleArn="arn:aws:iam::<account-number>:role/<role-name>",
        Tags=[
            {
                "Key": "string",
                "Value": "string"
            },
        ]
    )
```

Custom Integration

Jika Anda menggunakan integrasi kustom, mengecualikan parameter berikut:`HumanLoopRequestSource`,`HumanLoopActivationConfig`.

```
    response = client.create_flow_definition(
        FlowDefinitionName="human-review-workflow-name",
        HumanLoopConfig={
            "WorkteamArn": workteamArn,
            "HumanTaskUiArn": humanTaskUiArn,
            "TaskTitle": "Image content moderation",
            "TaskDescription": "Review the image and instructions. Complete the
task",
            "TaskCount": 1,
            "TaskAvailabilityLifetimeInSeconds": 43200,
            "TaskTimeLimitInSeconds": 3600,
            "TaskKeywords": [
                "content moderation",
            ],
```

```
        },
        OutputConfig={
            "S3OutputPath": "s3://DOC-EXAMPLE-BUCKET/prefix/",
        },
        RoleArn="arn:aws:iam::<account-number>:role/<role-name>",
        Tags=[
            {
                "Key": "string",
                "Value": "string"
            },
        ]
    )
```

Setelah membuat alur kerja tinjauan manusia, Anda dapat mengambil definisi alur ARN dari respons:

```
    humanReviewWorkflowArn = response["FlowDefinitionArn"]
```

Buat Loop Manusia

Operasi API yang Anda gunakan untuk memulai loop manusia bergantung pada integrasi Amazon A2I yang Anda gunakan.

- Jika Anda menggunakan integrasi bawaan Amazon Textract, Anda menggunakan AnalyzeDocumentoperasi.
- Jika Anda menggunakan integrasi bawaan Amazon Rekognition, Anda menggunakan DetectModerationLabelsoperasi.
- Jika Anda menggunakan integrasi kustom, Anda menggunakan StartHumanLoopoperasi.

Pilih jenis tugas Anda dalam tabel berikut untuk melihat contoh permintaan Amazon Textract dan Amazon Rekognition menggunakanAWS SDK for Python (Boto3).

Amazon Textract – Key-value pair extraction

Contoh berikutAWS SDK for Python (Boto3) menggunakan panggilan`analyze_document` us-west-2. Ganti teks merah yang dicetak miring dengan sumber daya Anda. Sertakan `DataAttributes`parameter jika Anda menggunakan tenaga kerja Amazon Mechanical Turk. Untuk informasi selengkapnya, lihat analyze_document documention di AWS SDK for Python (Boto)API Reference.

```
   response = client.analyze_document(
         Document={"S3Object": {"Bucket": "AWSDOC-EXAMPLE-BUCKET", "Name":
 "document-name.pdf"},
         HumanLoopConfig={
            "FlowDefinitionArn":"arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
            "HumanLoopName":"human-loop-name",
            "DataAttributes" : {ContentClassifiers:
["FreeOfPersonallyIdentifiableInformation"|"FreeOfAdultContent"]}
         }
         FeatureTypes=["FORMS"]
    )
```

Loop manusia hanya dibuat jika kepercayaan Amazon Textract untuk tugas analisis dokumen memenuhi kondisi aktivasi yang Anda tentukan dalam alur kerja peninjauan manusia. Anda dapat memeriksa`aresponse` elemen untuk menentukan apakah loop manusia telah dibuat. Untuk melihat semua yang termasuk dalam respons ini, lihat `HumanLoopActivationOutput`.

```
    if "HumanLoopArn" in analyzeDocumentResponse["HumanLoopActivationOutput"]:
        # A human loop has been started!
        print(f"A human loop has been started with ARN:
 {analyzeDocumentResponse["HumanLoopActivationOutput"]["HumanLoopArn"]}"
```

Amazon Rekognition – Image moderation

Contoh berikutAWS SDK for Python (Boto3) menggunakan panggilan`detect_moderation_labels` us-west-2. Ganti teks merah yang dicetak miring dengan sumber daya Anda. Sertakan `DataAttributes`parameter jika Anda menggunakan tenaga kerja Amazon Mechanical Turk. Untuk informasi selengkapnya, lihat documention detect_moderation_labels di ReferensiAWS SDK for Python (Boto) API.

```
   response = client.detect_moderation_labels(
            Image={"S3Object":{"Bucket": "AWSDOC-EXAMPLE-BUCKET", "Name": "image-
name.png"}},
            HumanLoopConfig={
               "FlowDefinitionArn":"arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
               "HumanLoopName":"human-loop-name",
```

```
                "DataAttributes":{ContentClassifiers:
["FreeOfPersonallyIdentifiableInformation"|"FreeOfAdultContent"]}
            }
    )
```

Loop manusia hanya dibuat jika kepercayaan Amazon Rekognition untuk tugas moderasi gambar memenuhi kondisi aktivasi yang Anda tentukan dalam alur kerja peninjauan manusia. Anda dapat memeriksa`response` elemen untuk menentukan apakah loop manusia telah dibuat. Untuk melihat semua yang termasuk dalam respons ini, lihat `HumanLoopActivationOutput`.

```
    if "HumanLoopArn" in response["HumanLoopActivationOutput"]:
        # A human loop has been started!
        print(f"A human loop has been started with ARN:
 {response["HumanLoopActivationOutput"]["HumanLoopArn"]}")
```

Custom Integration

Contoh berikutAWS SDK for Python (Boto3) menggunakan panggilan`start_human_loop` us-west-2. Ganti teks merah yang dicetak miring dengan sumber daya Anda. Sertakan `DataAttributes`parameter jika Anda menggunakan tenaga kerja Amazon Mechanical Turk. Untuk informasi selengkapnya, lihat dokumentasi start_human_loop di ReferensiAWS SDK for Python (Boto) API.

```
   response = client.start_human_loop(
        HumanLoopName= "human-loop-name",
        FlowDefinitionArn= "arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
        HumanLoopInput={"InputContent": inputContentJson},
        DataAttributes={"ContentClassifiers":
["FreeOfPersonallyIdentifiableInformation"|"FreeOfAdultContent"]}
   )
```

Contoh ini menyimpan konten masukan dalam variabel *`inputContentJson`*. Asumsikan bahwa konten masukan berisi dua elemen: uraian teks dan sentimen (sepertiPositiveNegative,, orNeutral), dan diformat sebagai berikut:

```
    inputContent = {
```

```
        "initialValue": sentiment,
         "taskObject": blurb
    }
```

Kunci`initialValue` dan`taskObject` harus sesuai dengan kunci yang digunakan dalam elemen cair dari template tugas pekerja. Lihat template kustomBuat UI Tugas Manusia untuk melihat contoh.

Untuk membuat*`inputContentJson`*, lakukan hal berikut:

```
import json

inputContentJson = json.dumps(inputContent)
```

Sebuah lingkaran manusia dimulai setiap kali Anda meneleponstart_human_loop. Untuk memeriksa status loop manusia, gunakan eas_human_loop:

```
human_loop_info = a2i.describe_human_loop(HumanLoopName="human_loop_name")
print(f"HumanLoop Status: {resp["HumanLoopStatus"]}")
print(f"HumanLoop Output Destination: {resp["HumanLoopOutput"]}")
```

Contoh Kasus Penggunaan Amazon A2I

Anda dapat menggunakan Amazon Augmented AI untuk memasukkan tinjauan manusia ke dalam alur kerja Andatipe bawaan, Amazon Textract Texact dan Amazon Rekognition, atau tugas kustom Anda sendiri menggunakanJenis tugas kustom.

Bila Anda membuat alur kerja tinjauan manusia menggunakan salah satu tipe tugas bawaan, Anda dapat menentukan kondisi, seperti ambang batas kepercayaan, yang memulai tinjauan manusia. Layanan (Amazon Rekognition atau Amazon Textract Texact) membuat loop manusia atas nama Anda ketika kondisi ini terpenuhi dan memasok data masukan Anda langsung ke Amazon A2I untuk dikirim ke pengulas manusia. Untuk mempelajari lebih lanjut tentang tipe tugas bawaan, gunakan yang berikut ini:

- Menggunakan Amazon Augmented AI dengan Amazon Textract
- Menggunakan Amazon Augmented AI dengan Amazon Rekognition

Saat Anda menggunakan tipe tugas khusus, Anda membuat dan memulai loop manusia menggunakan API Runtime Amazon A2I. Gunakan jenis tugas khusus untuk menggabungkan alur kerja tinjauan manusia dengan yang lainAWSlayanan atau aplikasi ML-mu sendiri.

- Untuk detail selengkapnya, lihat Menggunakan Amazon Augmented AI dengan Jenis Tugas Kustom

Tabel berikut menguraikan berbagai kasus penggunaan Amazon A2I yang dapat Anda jelajahi menggunakan Notebook SageMaker Jupyter. Untuk memulai dengan Notebook Jupyter, gunakan petunjuk diGunakan Instans Notebook SageMaker dengan Notebook Amazon A2I Jupyter. Untuk contoh lainnya, lihat iniRepositori GitHub.

Kasus Penggunaan	Deskripsi	Jenis tugas
Menggunakan Amazon A2I dengan Amazon Textract	Mintalah manusia meninjau dokumen satu halaman untuk meninjau pasangan nilai kunci bentuk penting, atau minta Amazon Textract secara acak mencicipi dan mengirim dokumen dari dataset Anda ke manusia untuk ditinjau.	Bawaan
Menggunakan Amazon A2I dengan Amazon Rekognition	Mintalah manusia meninjau gambar yang tidak aman untuk konten dewasa atau kekerasan eksplisit jika Amazon Rekogniti on mengembalikan skor kepercayaan rendah, atau minta Amazon Rekogniti on secara acak mencicipi dan mengirim gambar dari dataset Anda ke manusia untuk ditinjau.	Bawaan

Kasus Penggunaan	Deskripsi	Jenis tugas
Menggunakan Amazon A2I dengan Amazon Comprehend	Minta manusia meninjau kesimpulan Amazon Comprehend tentang data teks seperti analisis sentimen, sintaks teks, dan deteksi entitas.	Khusus
Menggunakan Amazon A2I dengan Amazon Transcribe	Mintalah manusia meninjau transkripsi Amazon Transcrib e file video atau audio. Gunakan hasil transkripsi loop tinjauan manusia untuk membuat kosakata kustom dan meningkatkan transkripsi masa depan video atau konten audio serupa.	Khusus
Menggunakan Amazon A2I dengan Amazon Translate	Mintalah manusia meninjau terjemahan dengan kepercaya an rendah yang dikembalikan dari Amazon Translate.	Khusus
Gunakan Amazon A2I untuk meninjau kesimpulan ML-real time	Gunakan Amazon A2I untuk meninjau kesimpulan dengan tingkat kepercayaan rendah waktu nyata yang dibuat oleh model yang dikerahkan ke titik akhir yang dihosting SageMaker dan melatih model Anda secara bertahap menggunakan data output Amazon A2I.	Khusus

Kasus Penggunaan	Deskripsi	Jenis tugas
Menggunakan Amazon A2I untuk meninjau data tabular	Gunakan Amazon A2I untuk mengintegrasikan loop tinjauan manusia ke dalam aplikasi ML-nya yang menggunakan data tabular.	Khusus

Topik

- Gunakan Instans Notebook SageMaker dengan Notebook Amazon A2I Jupyter
- Menggunakan Amazon Augmented AI dengan Amazon Textract
- Menggunakan Amazon Augmented AI dengan Amazon Rekognition
- Menggunakan Amazon Augmented AI dengan Jenis Tugas Kustom

Gunakan Instans Notebook SageMaker dengan Notebook Amazon A2I Jupyter

Untuk contoh end-to-end yang menunjukkan cara mengintegrasikan loop tinjauan manusia Amazon A2I ke dalam alur kerja machine learning, Anda dapat menggunakan Notebook Jupyter dari iniRepositori GitHubdalam contoh notebook SageMaker.

Untuk menggunakan notebook sampel tipe tugas kustom Amazon A2I di instans notebook Amazon SageMaker:

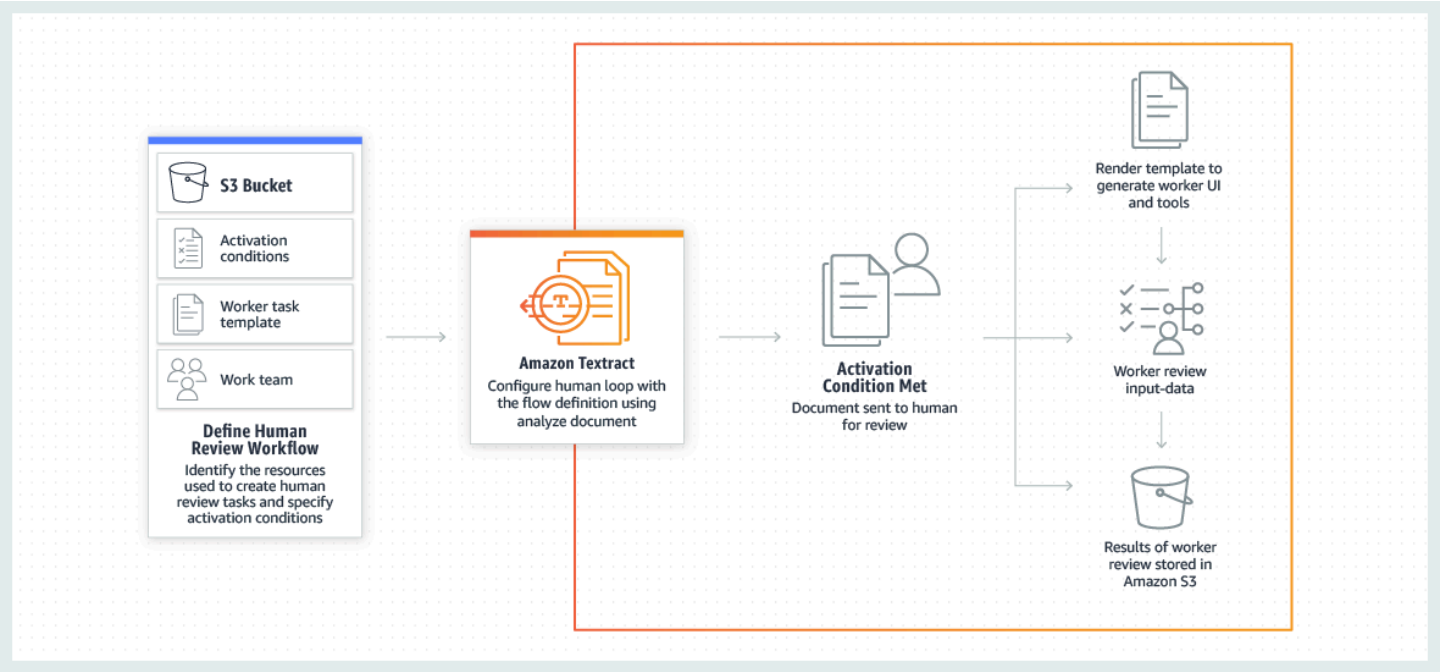
1. Jika Anda tidak memiliki instance notebook SageMaker yang aktif, buat dengan mengikuti petunjukLangkah 1: Buat Instans SageMaker Notebook Amazon.
2. Saat instans notebook Anda aktif, pilihBuka JupyterLabdi sebelah kanan nama contoh notebook. Perlu waktu beberapa saat agar JupyterLab dimuat.
3. Pilih  ikc untuk mengkloning repositori GitHub ke ruang kerja Anda.
4. Masukkanamazon-a2i-sampel-jupyter-notebookURL HTTPS repositori.
5. PilihPENGKLONAAN.
6. Buka notebook yang ingin Anda jalankan.
7. Ikuti petunjuk di notebook untuk mengkonfigurasi alur kerja tinjauan manusia Anda dan loop manusia dan menjalankan sel.

8. Untuk menghindari biaya yang tidak perlu, setelah selesai dengan demo, hentikan dan hapus instans notebook Anda selain bucket Amazon S3, peran IAM, dan sumber daya CloudWatch Events yang dibuat selama penelusuran.

Menggunakan Amazon Augmented AI dengan Amazon Textract

Amazon Textract memungkinkan Anda untuk menambahkan deteksi dan analisis teks dokumen untuk aplikasi Anda. Amazon Augmented AI (Amazon A2I) secara langsung terintegrasi dengan Amazon Textract`AnalyzeDocument`Operasi API. Anda dapat menggunakan`AnalyzeDocument`untuk menganalisis dokumen untuk hubungan antara item terdeteksi. Saat Anda menambahkan loop tinjauan manusia Amazon A2I ke`AnalyzeDocument`permintaan, Amazon A2I memonitor hasil Amazon Textract dan mengirimkan dokumen ke satu atau lebih pekerja manusia untuk ditinjau ketika kondisi yang ditentukan dalam definisi aliran Anda terpenuhi. Misalnya, jika Anda ingin manusia meninjau kunci tertentu seperti`Full name:`dan nilai-nilai masukan terkait mereka, Anda dapat membuat kondisi aktivasi yang memulai tinjauan manusia setiap saat`Full name:`kunci terdeteksi atau ketika keyakinan kesimpulan untuk kunci itu jatuh dalam kisaran yang Anda tentukan.

Gambar berikut menggambarkan alur kerja bawaan Amazon A2I dengan Amazon Textract. Di sebelah kiri, sumber daya yang diperlukan untuk membuat alur kerja peninjauan manusia Amazon Textract Texact digambarkan: dan bucket Amazon S3, kondisi aktivasi, template tugas pekerja, dan tim kerja. Sumber daya ini digunakan untuk membuat alur kerja tinjauan manusia, atau definisi aliran. Panah menunjuk tepat ke langkah berikutnya dalam alur kerja: menggunakan Amazon Textract untuk mengkonfigurasi loop manusia dengan alur kerja tinjauan manusia. Panah kedua menunjuk langsung dari langkah ini ke langkah di mana kondisi aktivasi yang ditentukan dalam alur kerja tinjauan manusia terpenuhi. Ini memulai penciptaan loop manusia. Di sebelah kanan gambar, loop manusia digambarkan dalam tiga langkah: 1) UI pekerja dan alat dihasilkan dan tugas dibuat tersedia bagi pekerja,2) pekerja meninjau data input, dan akhirnya, 3) hasil disimpan di Amazon S3.



Anda dapat menentukan kapan Amazon Textract Texact mengirimkan tugas ke pekerja manusia untuk ditinjau saat membuat alur kerja tinjauan manusia atau definisi alur kerja dengan menentukankondisi aktivasi.

Anda dapat mengatur kondisi aktivasi berikut saat menggunakan jenis tugas Amazon Textract Texact:

- Memulai tinjauan manusia untuk kunci formulir tertentu berdasarkan skor kepercayaan kunci formulir.
- Memulai tinjauan manusia ketika kunci formulir tertentu hilang.
- Memulai tinjauan manusia untuk semua kunci formulir yang diidentifikasi oleh Amazon Textract dengan skor percaya diri dalam kisaran tertentu.
- Secara acak mengirim sampel formulir ke manusia untuk ditinjau.

Ketika kondisi aktivasi Anda tergantung pada skor kepercayaan kunci bentuk, Anda dapat menggunakan dua jenis kepercayaan prediksi untuk memulai loop manusia:

- kepercayaan Identifikasi— Skor kepercayaan untuk pasangan kunci-nilai terdeteksi dalam bentuk.
- kepercayaan kualifikasi— Skor kepercayaan untuk teks yang terkandung dalam kunci dan nilai dalam bentuk.

Pada gambar di bagian berikut,Nama lengkap: Janepasangan nilai kunci,Nama lengkapadalah kuncinya, danJaneadalah nilai.

Anda dapat mengatur kondisi aktivasi ini menggunakan konsol Amazon SageMaker saat membuat alur kerja tinjauan manusia, atau dengan membuat JSON untuk kondisi aktivasi loop manusia dan menentukannya sebagai masukan dalam`HumanLoopActivationConditions`parameter`CreateFlowDefinition`Operasi API. Untuk mempelajari cara menentukan kondisi aktivasi dalam format JSON, lihatSkema JSON untuk Kondisi Aktivasi Loop Manusia di Amazon Augmented AIdanGunakan Kondisi Aktivasi Loop Manusia JSON Schema dengan Amazon Textract.

> **Note**
>
> Saat menggunakan Augmented AI dengan Amazon Textract, buat sumber daya Augmented AI dalam hal yang samaAWSWilayah yang Anda gunakan untuk meneleponAnalyzeDocument.

Memulai: Mengintegrasikan Tinjauan Manusia ke dalam Amazon Texact Analyze Document Job

Untuk mengintegrasikan tinjauan manusia ke dalam pekerjaan pendeteksian teks dan analisis Amazon Textract, Anda perlu membuat definisi aliran, dan kemudian menggunakan API Amazon Textract untuk mengintegrasikan definisi aliran tersebut ke dalam alur kerja Anda. Untuk mempelajari cara membuat definisi aliran menggunakan konsol SageMaker atau Augmented AI API, lihat topik berikut:

- Membuat Alur Kerja Tinjauan Manusia (Konsol)
- Membuat Workflow Tinjauan Manusia (API)

Setelah Anda membuat definisi alur, lihatMenggunakan Augmented AI dengan Amazon Textractuntuk mempelajari cara mengintegrasikan definisi aliran Anda ke dalam tugas Amazon Textract Anda.

Contoh end-to-end Menggunakan Amazon Textract dan Amazon A2I

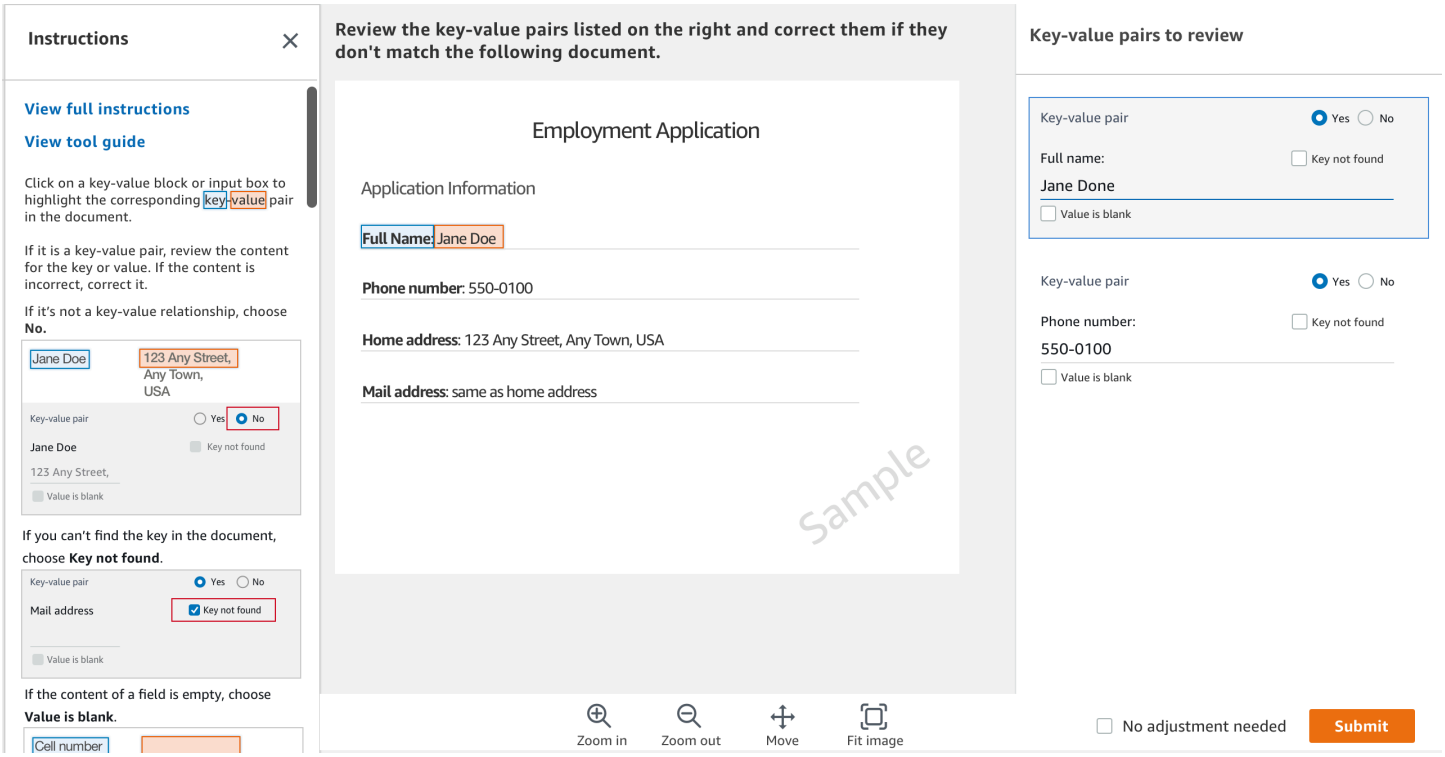
Untuk contoh menyeluruh yang menunjukkan cara menggunakan Amazon Textract dengan Amazon A2I menggunakan konsol, lihatTutorial: Mulai Konsol Amazon A2I.

Untuk mempelajari cara menggunakan API Amazon A2I untuk membuat dan memulai tinjauan manusia, Anda dapat menggunakanIntegrasi Amazon Augmented AI (Amazon A2I) dengan Dokumen

Analisis Amazon TExtract [Contoh]dalam contoh Notebook SageMaker. Untuk memulai, lihat Gunakan Instans Notebook SageMaker dengan Notebook Amazon A2I Jupyter.

Pratinjau Konsol Pekerja Textract A2I

Ketika mereka ditugaskan tugas peninjauan dalam alur kerja Amazon Textract Texact, pekerja mungkin melihat antarmuka pengguna yang mirip dengan berikut ini:



Anda dapat menyesuaikan antarmuka ini di konsol SageMaker saat membuat definisi tinjauan manusia, atau dengan membuat dan menggunakan templat khusus. Untuk mempelajari informasi lebih lanjut, lihat Membuat dan Mengelola Template Tugas Pekerja.

Menggunakan Amazon Augmented AI dengan Amazon Rekognition

Amazon Rekognition memudahkan untuk menambahkan analisis citra ke aplikasi Anda. Amazon Rekognition`DetectModerationLabels`Operasi API terintegrasi secara langsung dengan Amazon A2I sehingga Anda dapat dengan mudah membuat loop manual untuk meninjau citra yang tidak aman, seperti konten dewasa yang eksplisit atau kekerasan. Anda dapat menggunakan`DetectModerationLabels`untuk mengkonfigurasi loop manusia menggunakan ARN definisi aliran. Hal ini memungkinkan Amazon A2I menganalisis prediksi yang dibuat oleh Amazon Rekognition dan mengirim hasil kepada manusia untuk ditinjau untuk memastikan mereka memenuhi kondisi yang ditetapkan dalam definisi aliran Anda.

Gambar berikut menggambarkan alur kerja bawaan Amazon A2I dengan Amazon Rekognition. Di sebelah kiri, sumber daya yang diperlukan untuk membuat alur kerja peninjauan manusia Amazon Rekognition digambarkan: dan bucket Amazon S3, kondisi aktivasi, template tugas pekerja, dan tim kerja. Sumber daya ini digunakan untuk membuat alur kerja tinjauan manusia, atau definisi aliran. Panah menunjuk tepat ke langkah berikutnya dalam alur kerja: menggunakan Amazon Rekognition untuk mengkonfigurasi loop manusia dengan alur kerja tinjauan manusia. Panah kedua menunjuk langsung dari langkah ini ke langkah di mana kondisi aktivasi yang ditentukan dalam alur kerja tinjauan manusia terpenuhi. Ini memulai penciptaan loop manusia. Di sebelah kanan gambar, loop manusia digambarkan dalam tiga langkah: 1) UI pekerja dan alat dihasilkan dan tugas dibuat tersedia untuk pekerja, 2) pekerja meninjau data masukan, dan akhirnya, 3) hasil disimpan di Amazon S3.

Anda dapat mengatur kondisi aktivasi berikut saat menggunakan jenis tugas Amazon Rekognition:

- Memulai tinjauan manusia untuk label yang diidentifikasi oleh Amazon Rekognition berdasarkan skor kepercayaan label.
- Secara acak mengirim sampel gambar ke manusia untuk ditinjau.

Anda dapat mengatur kondisi aktivasi ini menggunakan konsol Amazon SageMaker saat membuat alur kerja tinjauan manusia, atau dengan membuat JSON untuk kondisi aktivasi loop manusia dan menentukannya sebagai masukan dalam`HumanLoopActivationConditions`parameter`CreateFlowDefinition`Operasi API. Untuk mempelajari cara menentukan kondisi aktivasi dalam format JSON, lihatSkema JSON untuk

Kondisi Aktivasi Loop Manusia di Amazon Augmented AIdanGunakan Kondisi Aktivasi Loop Manusia JSON Schema dengan Amazon Rekognition.

> **Note**
>
> Saat menggunakan Augmented AI dengan Amazon Rekognition, buat sumber daya Augmented AI yang samaAWSWilayah yang Anda gunakan untuk meneleponDetectModerationLabels.

Memulai: Mengintegrasikan Tinjauan Manusia ke dalam Job Moderasi Gambar Amazon Rekognition

Untuk integrasi peninjauan manual ke Amazon Rekognition, lihat topik berikut:

- Membuat Alur Kerja Tinjauan Manusia (Konsol)
- Membuat Workflow Tinjauan Manusia (API)

Setelah Anda membuat definisi alur, lihatMenggunakan Augmented AI dengan Amazon Rekognitionuntuk mempelajari cara mengintegrasikan definisi aliran Anda ke dalam tugas Amazon Rekognition Anda.

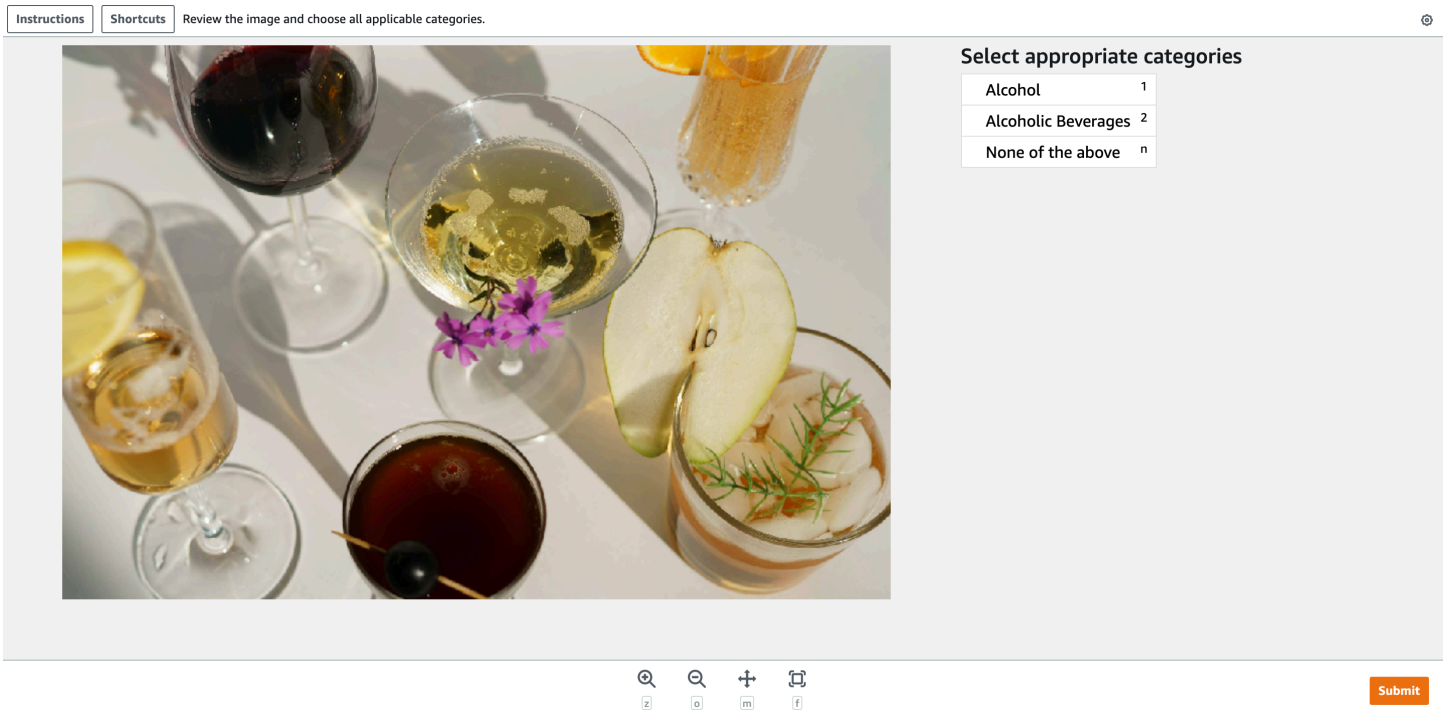
Demo end-to-end Menggunakan Amazon Rekognition dan Amazon A2I

Untuk contoh menyeluruh yang menunjukkan cara menggunakan Amazon Rekognition dengan Amazon A2I menggunakan konsol, lihatTutorial: Mulai Konsol Amazon A2I.

Untuk mempelajari cara menggunakan API Amazon A2I untuk membuat dan memulai tinjauan manusia, Anda dapat menggunakanIntegrasi Amazon Augmented AI (Amazon A2I) dengan Amazon Rekognition [Contoh]dalam Instans Notebook SageMaker. Untuk memulai, lihat Gunakan Instans Notebook SageMaker dengan Notebook Amazon A2I Jupyter.

Pratinjau Konsol Pekerja A2I Rekognition

Ketika mereka ditugaskan tugas peninjauan dalam alur kerja Amazon Rekognition, pekerja mungkin melihat antarmuka pengguna yang mirip dengan berikut ini:



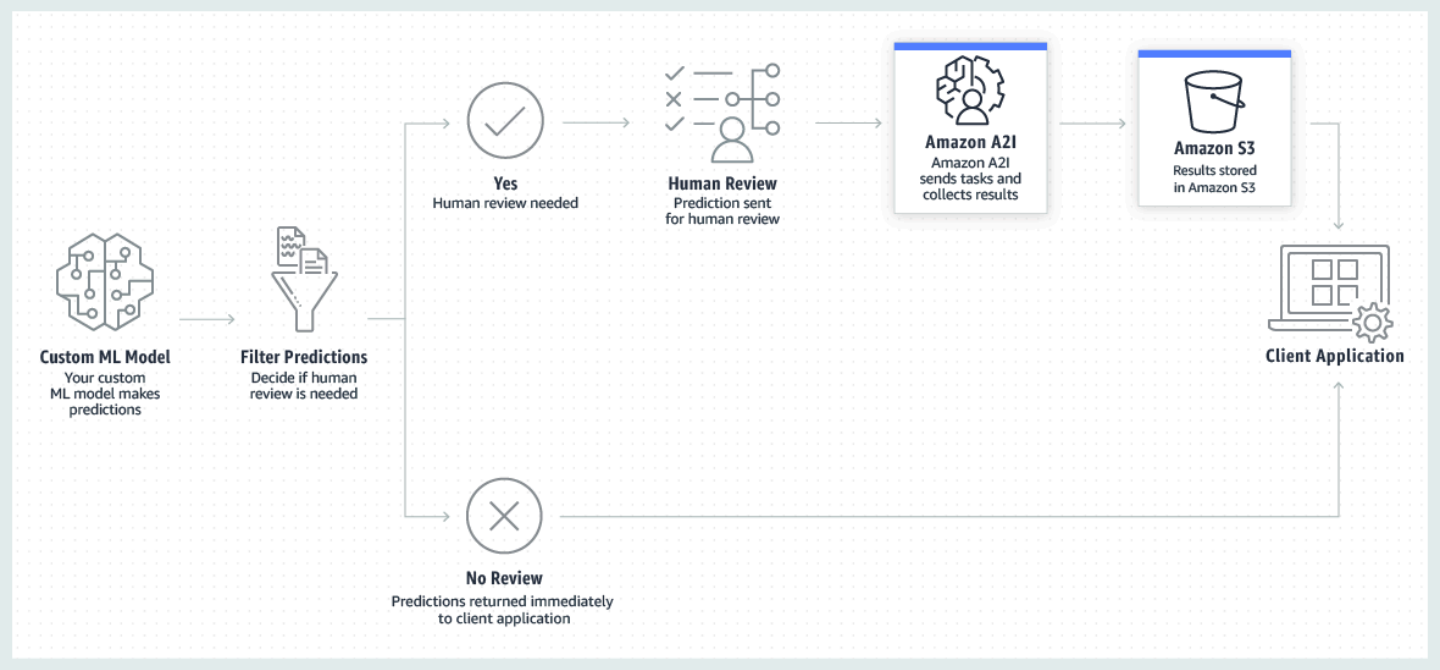
Anda dapat menyesuaikan antarmuka ini di konsol SageMaker saat membuat definisi tinjauan manusia, atau dengan membuat dan menggunakan templat khusus. Untuk mempelajari informasi lebih lanjut, lihat Membuat dan Mengelola Template Tugas Pekerja.

Menggunakan Amazon Augmented AI dengan Jenis Tugas Kustom

Anda dapat menggunakan Amazon Augmented AI (Amazon A2I) untuk memasukkan peninjauan manusia (human loop) keapa punalur kerja pembelajaran mesin menggunakanJenis tugas kustom. Opsi ini memberi Anda fleksibilitas paling banyak untuk menyesuaikan kondisi di mana objek data Anda dikirim ke manusia untuk ditinjau, serta tampilan dan nuansa antarmuka pengguna pekerja Anda.

Ketika Anda menggunakan jenis tugas kustom, Anda membuat alur kerja tinjauan manusia kustom dan menentukan kondisi di mana objek data dikirim untuk peninjauan manusia secara langsung di aplikasi Anda.

Gambar berikut menggambarkan alur kerja kustom Amazon A2I. Model ML-kustom digunakan untuk menghasilkan prediksi. Aplikasi klien menyaring prediksi ini menggunakan kriteria yang ditetapkan pengguna dan menentukan apakah tinjauan manusia diperlukan. Jika demikian, prediksi ini dikirim ke Amazon A2I untuk peninjauan manusia. Amazon A2I mengumpulkan hasil tinjauan manusia di Amazon S3, yang dapat diakses oleh aplikasi klien. Jika filter menentukan bahwa tidak ada tinjauan manusia diperlukan, prediksi dapat diumpankan langsung ke aplikasi klien.



Gunakan prosedur di halaman ini untuk mempelajari cara mengintegrasikan Amazon A2I ke dalam alur kerja machine learning apa pun menggunakan tipe tugas khusus.

Buat loop manusia menggunakan definisi aliran, integrasikan ke dalam aplikasi Anda, dan pantau hasilnya

1. Lengkapi Amazon A2IPrasyarat untuk Menggunakan Augmented AI. Perhatikan hal berikut:
   - Jalur ke bucket Amazon Simple Storage Service (Amazon S3) tempat Anda menyimpan data input dan output.
   - Amazon Resource Name (ARN) dariAWS Identity and Access Management(IAM) peran dengan izin yang diperlukan dilampirkan.
   - (Opsional) ARN tenaga kerja pribadi Anda, jika Anda berencana untuk menggunakannya.
2. Menggunakan elemen HTML, buat templat pekerja khusus yang digunakan Amazon A2I untuk menghasilkan UI tugas pekerja Anda. Untuk mempelajari cara membuat template kustom, lihatBuat Templat Tugas Pekerja Kustom.
3. Gunakan template pekerja khusus dari langkah 2 untuk menghasilkan template tugas pekerja di konsol Amazon SageMaker. Untuk mempelajari caranya, lihat Buat Template Tugas Pekerja.

   Pada langkah berikutnya, Anda membuat definisi aliran:

- Jika Anda ingin membuat definisi aliran menggunakan SageMaker API, perhatikan ARN template tugas pekerja ini untuk langkah berikutnya.
- Jika Anda membuat definisi aliran menggunakan konsol, template Anda secara otomatis muncul diTemplat tugas pekerjabagian ketika Anda memilihBuat alur kerja peninjauan manusia.

4. Saat membuat definisi alur, berikan jalur ke bucket S3, ARN peran IAM Anda, dan templat pekerja Anda.

   - Untuk mempelajari cara membuat definisi aliran menggunakan SageMaker`CreateFlowDefinition`API, lihatMembuat Workflow Tinjauan Manusia (API).
   - Untuk mempelajari cara membuat definisi aliran menggunakan konsol SageMaker, lihatMembuat Alur Kerja Tinjauan Manusia (Konsol).

5. Konfigurasikan loop manusia Anda menggunakanAmazon A2I. Untuk mempelajari caranya, lihat Membuat dan Memulai Loop Manusia.
6. Untuk mengontrol kapan tinjauan manusia dimulai dalam aplikasi Anda, tentukan kondisi di mana`StartHumanLoop`disebut dalam aplikasi Anda. Kondisi aktivasi loop manusia, seperti ambang kepercayaan yang memulai loop manusia, tidak tersedia saat menggunakan Amazon A2I dengan tipe tugas khusus. SETIAP`StartHumanLoop`hasil doa dalam tinjauan manusia.

Setelah memulai loop manusia, Anda dapat mengelola dan memantau loop menggunakan Amazon Augmented AI Runtime API dan Amazon EventBridge (juga dikenal sebagai Amazon CloudWatch Events). Untuk mempelajari selengkapnya, lihat Pantau dan Kelola Loop Manusia Anda.

Tutorial Akhir Menggunakan Jenis Tugas Kustom Amazon A2I

Untuk contoh end-to-end yang menunjukkan cara mengintegrasikan Amazon A2I ke dalam berbagai alur kerja ML, lihat tabel diContoh Kasus Penggunaan Amazon A2I. Untuk mulai menggunakan salah satu notebook ini, lihatGunakan Instans Notebook SageMaker dengan Notebook Amazon A2I Jupyter.

Buat Alur Kerja Tinjauan Manusia

Menggunakan Amazon Augmented AI (Amazon A2I)alur kerja tinjauan manusia, ataudefinisi alur, untuk menentukan hal berikut:

- Untuk jenis tugas bawaan Amazon TExtract dan Amazon Rekognition, kondisi di mana loop manusia Anda disebut

- Tenaga kerja tempat tugas Anda dikirim
- Set instruksi yang diterima tenaga kerja Anda, yang disebutTemplat tugas pekerja
- Konfigurasi tugas pekerja Anda, termasuk jumlah pekerja yang menerima tugas dan batas waktu untuk menyelesaikan tugas
- Tempat data output Anda disimpan

Anda dapat membuat alur kerja tinjauan manusia di konsol SageMaker atau menggunakan SageMaker`CreateFlowDefinition`operasi. Anda dapat membuat template tugas pekerja menggunakan konsol untuk jenis tugas Amazon Textract dan Amazon Rekognition sambil membuat definisi alur Anda.

> **Important**
>
> Kondisi aktivasi loop manusia, yang memulai loop manusia—misalnya, ambang kepercayaan diri—tidak tersedia untuk jenis tugas kustom Amazon A2I. Saat menggunakan konsol untuk membuat definisi aliran untuk jenis tugas khusus, Anda tidak dapat menentukan kondisi aktivasi. Saat menggunakan API Amazon A2I untuk membuat definisi aliran untuk jenis tugas khusus, Anda tidak dapat mengaturHumanLoopActivationConditionsatributHumanLoopActivationConditionsConfig Untuk mengontrol kapan tinjauan manusia dimulai, tentukan kondisi di manaStartHumanLoopdisebut dalam aplikasi kustom Anda. Dalam kasus ini, setiapStartHumanLoophasil doa dalam tinjauan manusia. Untuk informasi selengkapnya, lihat Menggunakan Amazon Augmented AI dengan Jenis Tugas Kustom.

Prasyarat

Untuk membuat definisi alur kerja tinjauan manusia, Anda harus telah menyelesaikan prasyarat yang dijelaskan dalamPrasyarat untuk Menggunakan Augmented AI.

Jika Anda menggunakan API untuk membuat definisi aliran untuk jenis tugas apa pun, atau jika Anda menggunakan tipe tugas khusus saat membuat definisi aliran di konsol, pertama buat template tugas pekerja. Untuk informasi selengkapnya, lihat Membuat dan Mengelola Template Tugas Pekerja.

Jika Anda ingin melihat pratinjau templat tugas pekerja saat membuat definisi alur untuk tipe tugas bawaan di konsol, pastikan bahwa Anda memberikan peran yang Anda gunakan untuk membuat izin definisi alur untuk mengakses bucket Amazon S3 yang berisi artefak template Anda menggunakan kebijakan seperti yang dijelaskan diAktifkan Pratinjau Template Tugas Pekerja .

Topik

Membuat Alur Kerja Tinjauan Manusia (Konsol)

Gunakan prosedur ini untuk membuat alur kerja peninjauan Amazon Augmented AI (Amazon A2I) menggunakan konsol SageMaker. Jika Anda baru mengenal Amazon A2I, sebaiknya Anda membuat tim kerja pribadi menggunakan orang-orang di organisasi Anda, dan gunakan ARN tim kerja ini saat membuat definisi alur Anda. Untuk mempelajari cara mengatur tenaga kerja pribadi dan membuat tim kerja, lihatBuat Tenaga Kerja Pribadi (Amazon SageMakerKonsol). Jika Anda telah menyiapkan tenaga kerja pribadi, lihatBuat Tim Kerja Menggunakan SageMaker Konsoluntuk mempelajari cara menambahkan tim kerja ke tenaga kerja itu.

Jika Anda menggunakan Amazon A2I dengan salah satu tipe tugas bawaan, Anda dapat membuat instruksi pekerja menggunakan template tugas pekerja default yang disediakan oleh Augmented AI sekaligus membuat alur kerja tinjauan manusia di konsol. Untuk melihat contoh template default yang disediakan oleh Augmented AI, lihat tipe tugas bawaan diContoh Kasus Penggunaan Amazon A2I.

Untuk membuat definisi aliran (konsol)

1. Buka konsol SageMaker di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, di bawah panelAugmented AIbagian, pilihAlur kerja peninjauandan kemudian pilihBuat alur kerja tinjauan manusia.
3. MasukIkhtisar, lakukan hal berikut:

    a. UntukNama, masukkan nama alur kerja yang unik. Nama harus huruf kecil, unik di dalamAWSWilayah di akun Anda, dan dapat memiliki hingga 63 karakter. Karakter yang valid meliputi: a-z, 0-9, dan - (tanda hubung).

    b. UntukLokasi S3 untuk output, masukkan bucket S3 tempat Anda ingin menyimpan hasil peninjauan manusia. Ember harus ditempatkan diAWSWilayah sebagai alur kerja.

    c. UntukPeran IAM, pilih peran yang memiliki izin yang diperlukan. Jika Anda memilih tipe tugas bawaan dan ingin melihat pratinjau templat pekerja Anda di konsol, berikan peran dengan jenis kebijakan yang dijelaskan dalamAktifkan Pratinjau Template Tugas Pekerja terpasang.

4. UntukJenis tugas, memilih jenis tugas yang Anda ingin pekerja manusia untuk melakukan.
5. Jika Anda memilih jenis tugas Amazon Rekognition atau Amazon Textract Texact, tentukan kondisi yang memanggil peninjauan manusia.

   - Untuk tugas moderasi gambar Amazon Rekognition, pilih interval ambang batas skor keyakinan inferensi yang memulai peninjauan manusia.
   - Untuk tugas Amazon Textract, Anda dapat memulai peninjauan manusia saat kunci formulir tertentu hilang atau ketika kepercayaan pendeteksian kunci formulir rendah. Anda juga dapat memulai tinjauan manusia jika, setelah mengevaluasi semua kunci formulir dalam teks, kepercayaan diri lebih rendah dari ambang batas yang diperlukan untuk kunci formulir apa pun. Dua variabel menentukan ambang kepercayaan Anda: kepercayaan Identifikasidankepercayaan kualifikasi. Untuk mempelajari lebih lanjut tentang variabel-variabel ini, lihatMenggunakan Amazon Augmented AI dengan Amazon Textract.
   - Untuk kedua jenis tugas, Anda dapat secara acak mengirim persentase objek data (gambar atau formulir) dan labelnya ke manusia untuk ditinjau.
6. Mengkonfigurasi dan menentukan template tugas pekerja Anda:

   a. Jika Anda menggunakan jenis tugas Amazon Rekognition atau Amazon Textract Texact:

      - DiMembuat templatBagian:

        - Untuk membuat instruksi bagi pekerja Anda menggunakan templat default Amazon A2I untuk jenis tugas Amazon Rekognition dan Amazon Textract Texact, pilihMembangun dari template default.
          - Jika Anda memilihMembangun dari template default, membuat instruksi Anda di bawahDesain tugas pekerja:
            - BerikanNama templatyang unik dalamAWSWilayah Anda berada di.
            - DiPetunjukbagian, memberikan petunjuk rinci tentang cara menyelesaikan tugas Anda. Untuk membantu pekerja mencapai akurasi yang lebih besar, berikan contoh yang baik dan buruk.
            - (Opsional) DalamPetunjuk tambahan, berikan informasi dan instruksi tambahan kepada pekerja Anda.

              Untuk informasi tentang membuat instruksi yang efektif, lihatMembuat Instruksi Pekerja yang Baik.

- Untuk memilih template kustom yang telah Anda buat, pilih dariTemplatmenu dan menyediakanDeskripsi tugasuntuk menjelaskan secara singkat tugas bagi pekerja Anda. Untuk mempelajari cara membuat templat kustom, lihatBuat Template Tugas Pekerja.

b. Jika Anda menggunakan jenis tugas kustom:

- DiTemplat tugas pekerjaBagian, pilih templat Anda dari daftar. Semua template yang telah Anda buat di konsol SageMaker muncul dalam daftar ini. Untuk mempelajari cara membuat templat untuk jenis tugas khusus, lihatMembuat dan Mengelola Template Tugas Pekerja.

7. (Opsional) Pratinjau template pekerja Anda:

   Untuk jenis tugas Amazon Rekognition dan Amazon Textract Texact, Anda memiliki opsi untuk memilihMelihat tugas pekerja sampeluntuk melihat pratinjau UI tugas pekerja Anda.

   Jika Anda membuat definisi aliran untuk jenis tugas kustom, Anda dapat melihat pratinjau UI tugas pekerja Anda menggunakan`RenderUiTemplate`operasi. Untuk informasi selengkapnya, lihat Pratinjau Template Tugas Pekerja.

8. UntukPekerja, pilih jenis tenaga kerja.
9. Pilih Create (Buat).

Langkah Selanjutnya

Setelah Anda membuat alur kerja tinjauan manusia, alur kerja tersebut muncul di konsol di bawahAlur kerja peninjauan. Untuk melihat Amazon Resource Name (ARN) dan detail konfigurasi Anda, pilih alur kerja dengan memilih namanya.

Jika Anda menggunakan tipe tugas built-in, Anda dapat menggunakan ARN definisi aliran untuk memulai loop manusia menggunakan ituAWSAPI layanan (misalnya, API Amazon Textract). Untuk jenis tugas kustom, Anda dapat menggunakan ARN untuk memulai loop manusia menggunakan Amazon Augmented AI Runtime API. Untuk mempelajari lebih lanjut tentang kedua opsi, lihatMembuat dan Memulai Loop Manusia.

Membuat Workflow Tinjauan Manusia (API)

Untuk membuat definisi aliran menggunakan SageMaker API, Anda menggunakan`CreateFlowDefinition`operasi. Setelah Anda menyelesaikanPrasyarat untuk

Menggunakan Augmented AI, gunakan prosedur berikut untuk mempelajari cara menggunakan operasi API ini.

Untuk ikhtisar`CreateFlowDefinition`operasi, dan rincian tentang tiap parameter, lihat`CreateFlowDefinition`.

Membuat definisi alur (API)

1. Untuk`FlowDe#nitionName`, masukkan nama yang unik. Nama harus unik dalamAWSWilayah di akun Anda, dan dapat memiliki hingga 63 karakter. Karakter yang valid meliputi: a-z, 0-9, dan - (tanda hubung).
2. Untuk`RoleArn`, masukkan ARN peran yang Anda konfigurasi untuk memberikan akses ke sumber data Anda.
3. Untuk`HumanLoopConfig`, masukkan informasi tentang pekerja dan apa yang harus mereka lihat. Untuk informasi tentang tiap parameter di`HumanLoopConfig`, lihatHumanLoopConfig.
4. (Opsional) Jika Anda menggunakan tipe tugas bawaan, berikan kondisi yang memulai loop manusia di`HumanLoopActivationCon#g`. Untuk mempelajari cara membuat input yang diperlukan untuk`HumanLoopActivationCon#g`parameter, lihatSkema JSON untuk Kondisi Aktivasi Loop Manusia di Amazon Augmented AI. Jika Anda tidak menentukan kondisi di sini, ketika Anda memberikan definisi aliran keAWSLayanan yang terkait dengan tipe tugas bawaan (misalnya, Amazon Textract atau Amazon Rekognition), layanan tersebut mengirimkan setiap tugas kepada pekerja manusia untuk ditinjau.

   Jika Anda menggunakan jenis tugas kustom,`HumanLoopActivationConfig`dinonaktifkan. Untuk mempelajari cara mengontrol kapan tugas dikirim ke pekerja manusia menggunakan jenis tugas khusus, lihatMenggunakan Amazon Augmented AI dengan Jenis Tugas Kustom.
5. (Opsional) Jika Anda menggunakan tipe tugas bawaan, tentukan sumber integrasi (misalnya, Amazon Rekognition atau Amazon Textract Texact) diHumanLoopRequestSourceparameter.
6. Untuk`OutputConfig`, menunjukkan tempat Amazon Simple Storage Service (Amazon S3) untuk menyimpan output loop manusia.
7. (Opsional) Gunakan`Tags`untuk memasukkan pasangan nilai kunci untuk membantu Anda mengategorikan dan mengatur definisi alur. Setiap tag terdiri atas sebuah kunci dan sebuah nilai opsional, yang keduanya Anda tentukan.

Amazon Textract – Key-value pair extraction

Berikut ini adalah contoh permintaan untuk membuat alur kerja peninjauan manusia Amazon Textract (definisi aliran) menggunakanAWS SDK for Python (Boto3). Anda harus menggunakan`'AWS/Textract/AnalyzeDocument/Forms/V1'`untuk membuat loop manusia Amazon Textract Texact. Hanya termasuk`PublicWorkforceTaskPrice`jika Anda menggunakan tenaga kerja Mechanical Turk.

```
sagemaker_client = boto3.client('sagemaker', aws_region)

response = sagemaker_client.create_flow_definition(
    FlowDefinitionName='ExampleFlowDefinition',
    HumanLoopRequestSource={
         'AwsManagedHumanLoopRequestSource': 'AWS/Textract/AnalyzeDocument/Forms/V1'
    },
    HumanLoopActivationConfig={
        'HumanLoopActivationConditionsConfig': {
            'HumanLoopActivationConditions': '{...}'
        }
    },
    HumanLoopConfig={
        'WorkteamArn': 'arn:aws:sagemaker:aws_region:aws_account_number:workteam/
private-crowd/workteam_name',
        'HumanTaskUiArn': 'arn:aws:sagemaker:aws_region:aws_account_number:human-
task-ui/template_name',
        'TaskTitle': 'Example task title',
        'TaskDescription': 'Example task description.',
        'TaskCount': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskKeywords': [
            'Keyword1','Keyword2'
        ],
        'PublicWorkforceTaskPrice': {
            'AmountInUsd': {
                'Dollars': 123,
                'Cents': 123,
                'TenthFractionsOfACent': 123
            }
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/',
```

```
        'KmsKeyId': '1234abcd-12ab-34cd-56ef-1234567890ab'
    },
    RoleArn='arn:aws:iam::aws_account_number:role/role_name',
    Tags=[
        {
            'Key': 'KeyName',
            'Value': 'ValueName'
        },
    ]
)
```

Amazon Rekognition – Image moderation

Berikut ini adalah contoh permintaan untuk membuat alur kerja peninjauan manusia Amazon Rekognition (definisi aliran) menggunakanAWS SDK for Python (Boto3). Anda harus menggunakan`'AWS/Rekognition/DetectModerationLabels/Image/V3'`untuk membuat definisi aliran Amazon Rekognition. Hanya termasuk`PublicWorkforceTaskPrice`jika Anda menggunakan tenaga kerja Mechanical Turk.

```
sagemaker_client = boto3.client('sagemaker', aws_region)

response = sagemaker_client.create_flow_definition(
    FlowDefinitionName='ExampleFlowDefinition',
    HumanLoopRequestSource={
         'AwsManagedHumanLoopRequestSource': 'AWS/Rekognition/
DetectModerationLabels/Image/V3'
    },
    HumanLoopActivationConfig={
        'HumanLoopActivationConditionsConfig': {
            'HumanLoopActivationConditions': '{...}'
        }
    },
    HumanLoopConfig={
        'WorkteamArn': 'arn:aws:sagemaker:aws_region:aws_account_number:workteam/
private-crowd/workteam_name',
        'HumanTaskUiArn': 'arn:aws:sagemaker:aws_region:aws_account_number:human-
task-ui/template_name',
        'TaskTitle': 'Example task title',
        'TaskDescription': 'Example task description.',
        'TaskCount': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskKeywords': [
```

```
            'Keyword1','Keyword2'
        ],
        'PublicWorkforceTaskPrice': {
            'AmountInUsd': {
                'Dollars': 123,
                'Cents': 123,
                'TenthFractionsOfACent': 123
            }
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/',
        'KmsKeyId': '1234abcd-12ab-34cd-56ef-1234567890ab'
    },
    RoleArn='arn:aws:iam::aws_account_number:role/role_name',
    Tags=[
        {
            'Key': 'KeyName',
            'Value': 'ValueName'
        },
    ]
)
```

Custom Workflow

Berikut ini adalah contoh permintaan untuk membuat alur kerja tinjauan manusia (definisi aliran) untuk integrasi kustom. Untuk membuat jenis alur kerja tinjauan manusia ini, hilangkan`HumanLoopRequestSource`dari permintaan definisi aliran. Anda hanya perlu menyertakan`PublicWorkforceTaskPrice`jika Anda menggunakan tenaga kerja Mechanical Turk.

```
sagemaker_client = boto3.client('sagemaker', aws_region)

response = sagemaker_client.create_flow_definition(
    FlowDefinitionName='ExampleFlowDefinition',
    HumanLoopActivationConfig={
        'HumanLoopActivationConditionsConfig': {
            'HumanLoopActivationConditions': '{...}'
        }
    },
    HumanLoopConfig={
        'WorkteamArn': 'arn:aws:sagemaker:aws_region:aws_account_number:workteam/
private-crowd/workteam_name',
```

```
        'HumanTaskUiArn': 'arn:aws:sagemaker:aws_region:aws_acount_number:human-
task-ui/template_name',
        'TaskTitle': 'Example task title',
        'TaskDescription': 'Example task description.',
        'TaskCount': 123,
        'TaskAvailabilityLifetimeInSeconds': 123,
        'TaskTimeLimitInSeconds': 123,
        'TaskKeywords': [
            'Keyword1','Keyword2'
        ],
        'PublicWorkforceTaskPrice': {
            'AmountInUsd': {
                'Dollars': 123,
                'Cents': 123,
                'TenthFractionsOfACent': 123
            }
        }
    },
    OutputConfig={
        'S3OutputPath': 's3://bucket/path/',
        'KmsKeyId': '1234abcd-12ab-34cd-56ef-1234567890ab'
    },
    RoleArn='arn:aws:iam::account_number:role/role_name',
    Tags=[
        {
            'Key': 'KeyName',
            'Value': 'ValueName'
        },
    ]
)
```

Langkah Selanjutnya

Nilai kembali dari panggilan sukses dari`CreateFlowDefinition`Operasi API adalah definisi alur Amazon Resource Name (ARN).

Jika Anda menggunakan tipe tugas built-in, Anda dapat menggunakan ARN definisi aliran untuk memulai loop manusia menggunakan ituAWSAPI layanan (yaitu Amazon Textract API). Untuk jenis tugas kustom, Anda dapat menggunakan ARN untuk memulai loop manusia menggunakan Amazon Augmented AI Runtime API. Untuk mempelajari lebih lanjut tentang kedua opsi ini, lihatMembuat dan Memulai Loop Manusia.

Skema JSON untuk Kondisi Aktivasi Loop Manusia di Amazon Augmented AI

KlasterHumanLoopActivationConditionsadalah parameter input dari[CreateFlowDefinition](#)API. Parameter ini adalah string berformat JSON. JSON memodelkan kondisi di mana loop manusia dibuat ketika kondisi tersebut dievaluasi terhadap respons dari API layanan AI yang terintegrasi (sepertiRekognition.DetectModerationLabelsatauTextract.AnalyzeDocument). Tanggapan ini disebut sebagaiinferensi. Misalnya, Amazon Rekognition mengirimkan inferensi label moderasi dengan skor kepercayaan terkait. Dalam contoh ini, inferensi adalah perkiraan terbaik model dari label yang sesuai untuk gambar. Untuk Amazon Textract, inferensi dibuat pada hubungan antara blok teks (pasangan nilai kunci), seperti hubungan antaraName:danSuedalam bentuk serta konten dalam blok teks, ataublok kata, seperti 'Nama'.

Berikut ini adalah skema untuk JSON. Di tingkat atas,HumanLoopActivationConditionsmemiliki array JSON,Conditions. Setiap anggota array ini adalah kondisi independen yang, jika dievaluasitrue, menghasilkan Amazon A2I membuat loop manusia. Setiap kondisi independen tersebut bisa menjadi kondisi sederhana atau kondisi yang kompleks. Sebuah kondisi sederhana memiliki atribut berikut:

- ConditionType: Atribut ini mengidentifikasi tipe kondisi. MASING-MASINGAWSAPI layanan AI yang terintegrasi dengan Amazon A2I mendefinisikan kumpulan yang diizinkanConditionTypes.
  - RekognitionDetectModerationLabels- API ini mendukungModerationLabelConfidenceCheckdanSampling ConditionTypevalues.
  - TextractAnalyzeDocument- API ini mendukungImportantFormKeyConfidenceCheck,MissingImportantFormKey, danSampling ConditionTypevalues.
- ConditionParameters- Ini adalah objek JSON yang parameterisasi kondisi. Himpunan atribut yang diizinkan dari objek ini tergantung pada nilaiConditionType. MASING-MASINGConditionTypemendefinisikan set sendiriConditionParameters.

Seorang anggotaConditionsarray dapat memodelkan kondisi yang kompleks. Hal ini dilakukan dengan menghubungkan kondisi sederhana secara logis menggunakanAnddanOroperator logis dan bersarang kondisi sederhana yang mendasarinya. Hingga dua tingkat bersarang didukung.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "definitions": {
```

```
        "Condition": {
            "type": "object",
            "properties": {
                "ConditionType": {
                    "type": "string"
                },
                "ConditionParameters": {
                    "type": "object"
                }
            },
            "required": [
                "ConditionType"
            ]
        },
        "OrConditionArray": {
            "type": "object",
            "properties": {
                "Or": {
                    "type": "array",
                    "minItems": 2,
                    "items": {
                        "$ref": "#/definitions/ComplexCondition"
                    }
                }
            }
        },
        "AndConditionArray": {
            "type": "object",
            "properties": {
                "And": {
                    "type": "array",
                    "minItems": 2,
                    "items": {
                        "$ref": "#/definitions/ComplexCondition"
                    }
                }
            }
        },
        "ComplexCondition": {
            "anyOf": [
                {
                    "$ref": "#/definitions/Condition"
                },
                {
```

```
                    "$ref": "#/definitions/OrConditionArray"
                },
                {
                    "$ref": "#/definitions/AndConditionArray"
                }
            ]
        }
    },
    "type": "object",
    "properties": {
        "Conditions": {
            "type": "array",
            "items": {
                "$ref": "#/definitions/ComplexCondition"
            }
        }
    }
}
```

> **Note**
>
> Kondisi aktivasi loop manusia tidak tersedia untuk alur kerja tinjauan manusia yang terintegrasi dengan jenis tugas khusus. KlasterHumanLoopActivationConditionsparameter dinonaktifkan untuk jenis tugas kustom.

Topik

- Gunakan Kondisi Aktivasi Loop Manusia JSON Schema dengan Amazon Textract
- Gunakan Kondisi Aktivasi Loop Manusia JSON Schema dengan Amazon Rekognition

Gunakan Kondisi Aktivasi Loop Manusia JSON Schema dengan Amazon Textract

Saat digunakan dengan Amazon A2I,AnalyzeDocumentoperasi mendukung input berikut diConditionTypeparameter:

- ImportantFormKeyConfidenceCheck- Gunakan kondisi ini untuk membuat loop manusia ketika keyakinan inferensi berada dalam kisaran tertentu untuk kunci formulir dokumen dan blok kata. SEBUAHkunci formuliradalah setiap kata dalam dokumen yang dikaitkan dengan masukan. Intputnya disebut anilai. Bersama-sama, kunci bentuk dan nilai disebut sebagaipasangan nilai

kunci. SEBUAHblok katamengacu pada kata-kata yang dikenali Amazon Textract di dalam blok teks yang terdeteksi. Untuk mempelajari selengkapnya tentang blok dokumen Amazon Textract, lihatDokumen dan Blok Objekdi dalamAmazon Textract Panduan Developer.

- `MissingImportantFormKey`- Gunakan kondisi ini untuk membuat loop manusia ketika Amazon Textract tidak mengidentifikasi kunci atau alias terkait dalam dokumen.
- `Sampling`- Gunakan kondisi ini untuk menentukan persentase formulir untuk dikirim ke manusia untuk ditinjau, terlepas dari skor kepercayaan inferensi. Gunakan kondisi ini untuk melakukan hal berikut:
  - Audit model MS Anda dengan mengambil sampel secara acak semua formulir yang dianalisis oleh model Anda dan mengirimkan persentase tertentu kepada manusia untuk ditinjau.
  - MenggunakanImportantFormKeyConfidenceCheckkondisi, sampel secara acak persentase dari kesimpulan yang memenuhi kondisi yang ditentukan dalamImportantFormKeyConfidenceCheckuntuk memulai loop manusia dan hanya mengirim persentase yang ditentukan kepada manusia untuk ditinjau.

> **Note**
>
> Jika Anda mengirim permintaan yang sama keAnalyzeDocumentbeberapa kali, hasilSamplingtidak berubah untuk inferensi input itu. Misalnya, jika Anda membuatAnalyzeDocumentpermintaan sekali, danSamplingtidak memulai loop manusia, permintaan berikutnya untukAnalyzeDocumentdengan konfigurasi yang sama tidak memulai loop manusia.

`ImportantFormKeyConfidenceCheck`Input dan Hasil

KlasterImportantFormKeyConfidenceCheck `ConditionType`mendukung berikut ini`ConditionParameters`:

- `ImportantFormKey`— String yang mewakili kunci dalam pasangan nilai kunci yang terdeteksi oleh Amazon Textract yang perlu ditinjau oleh pekerja manusia. Jika nilai parameter ini adalah nilai catch-all khusus (*), maka semua kunci dianggap cocok dengan kondisi. Anda dapat menggunakan ini untuk memodelkan kasus di mana setiap pasangan kunci-nilai yang memenuhi ambang kepercayaan tertentu membutuhkan peninjauan manusia.
- `ImportantFormKeyAliases`- Array yang mewakili ejaan alternatif atau setara logis untuk kunci formulir penting.

- `KeyValueBlockConfidenceEquals`
- `KeyValueBlockConfidenceLessThan`
- `KeyValueBlockConfidenceLessThanEquals`
- `KeyValueBlockConfidenceGreaterThan`
- `KeyValueBlockConfidenceGreaterThanEquals`
- `WordBlockConfidenceEquals`
- `WordBlockConfidenceLessThan`
- `WordBlockConfidenceLessThanEquals`
- `WordBlockConfidenceGreaterThan`
- `WordBlockConfidenceGreaterThanEquals`

Saat Anda menggunakan`ImportantFormKeyConfidenceCheck` `ConditionType`, Amazon A2I mengirimkan blok kunci-nilai dan blok kata kesimpulan dari blok kunci-nilai dan alias terkait yang Anda tentukan`ImportantFormKey`dan`ImportantFormKeyAliases`untuk review manusia.

Saat membuat definisi alur, jika Anda menggunakan template tugas pekerja default yang disediakan diAlur kerja peninjauan manusiabagian dari Amazon SageMakerconsole, key-value, dan block inferences yang dikirim untuk peninjauan manusia oleh kondisi aktivasi ini disertakan dalam UI worker. Jika Anda menggunakan template tugas pekerja kustom, Anda perlu menyertakan`{{ task.input.selectedAiServiceResponse.blocks }}`elemen untuk menyertakan data input nilai awal (inferensi) dari Amazon Textract. Untuk contoh template kustom yang menggunakan elemen masukan ini, lihatContoh kustom untuk Amazon Textract.

`MissingImportantFormKey`Input dan Hasil

Kluster`MissingImportantFormKey` `ConditionType`mendukung berikut ini`ConditionParameters`:

- `ImportantFormKey`— String yang mewakili kunci dalam pasangan nilai kunci yang terdeteksi oleh Amazon Textract yang perlu ditinjau oleh pekerja manusia.
- `ImportantFormKeyAliases`- Array yang mewakili ejaan alternatif atau setara logis untuk kunci formulir penting.

Saat Anda menggunakan`MissingImportantFormKey` `ConditionType`, jika kuncinya masuk`ImportantFormKey`alias atau alias di`ImportantFormKeyAliases`tidak termasuk

dalam inferensi Amazon Textract, formulir tersebut dikirim ke manusia untuk ditinjau dan tidak ada pasangan nilai kunci yang diprediksi disertakan. Misalnya, jika Amazon Textract hanya diidentifikasi`Address`dan`Phone`dalam bentuk, tapi hilang`ImportantFormKey Name`(dalam`MissingImportantFormKey`jenis kondisi) bahwa formulir akan dikirim ke manusia untuk ditinjau tanpa kunci formulir terdeteksi (`Address`dan`Phone`).

Jika Anda menggunakan template tugas pekerja default yang disediakan di SageMaker konsol, tugas dibuat meminta pekerja untuk mengidentifikasi kunci di`ImportantFormKey`dan nilai terkait. Jika Anda menggunakan template tugas pekerja kustom, Anda perlu menyertakan`<task.input.humanLoopContext>`elemen HTML kustom untuk mengkonfigurasi tugas ini.

Input dan Hasil Sampling

Klaster`Sampling ConditionType`mendukung`RandomSamplingPercentage ConditionParameters`. Input untuk`RandomSamplingPercentage`harus bilangan real antara 0,01 dan 100. Angka ini mewakili persentase data yang memenuhi syarat untuk tinjauan manusia dan dikirim ke manusia untuk ditinjau. Jika Anda menggunakan`Sampling`kondisi tanpa kondisi lain, angka ini mewakili persentase dari semua kesimpulan yang dihasilkan yang dibuat oleh`AnalyzeDocument`operasi dari satu permintaan yang dikirim ke manusia untuk ditinjau.

Jika Anda menentukan`Sampling`kondisi tanpa jenis kondisi lain, semua kunci-nilai dan blok kesimpulan dikirim ke pekerja untuk ditinjau.

Saat membuat definisi alur, jika Anda menggunakan template tugas pekerja default yang disediakan di*Alur kerja peninjauan manusia*bagian dari SageMakerkonsol, semua kunci-nilai dan blok inferensi yang dikirim untuk tinjauan manusia oleh kondisi aktivasi ini disertakan dalam UI pekerja. Jika Anda menggunakan template tugas pekerja kustom, Anda perlu menyertakan`{{ task.input.selectedAiServiceResponse.blocks }}`elemen untuk menyertakan data input nilai awal (inferensi) dari Amazon Textract. Untuk contoh template kustom yang menggunakan elemen masukan ini, lihat*Contoh kustom untuk Amazon Textract*.

Contoh

Sementara hanya satu kondisi yang perlu dievaluasi`true`untuk memulai loop manusia, Amazon A2I mengevaluasi semua kondisi untuk setiap objek yang dianalisis oleh Amazon Textract. Peninjau manusia diminta untuk meninjau kunci formulir penting untuk semua kondisi yang dievaluasi`true`.

Contoh 1: Mendeteksi kunci formulir penting dengan skor keyakinan dalam rentang tertentu yang memulai loop manusia

Contoh berikut menunjukkanHumanLoopActivationConditionsJSON yang memulai loop manusia jika salah satu dari tiga kondisi berikut terpenuhi:

- Amazon TextractAnalyzeDocumentAPI mengembalikan pasangan nilai kunci yang kuncinya adalah salah satuEmployee Name,Name, atauEmployeeName, dengan keyakinan blok kunci-nilai yang kurang dari 60 dan kepercayaan dari masing-masing blok kata membentuk kunci dan nilai yang kurang dari 85.
- Amazon TextractAnalyzeDocumentAPI mengembalikan pasangan nilai kunci yang kuncinya adalah salah satuPay Date,PayDate,DateOfPay, ataupay-date, dengan keyakinan blok kunci-nilai yang kurang dari 65 dan kepercayaan dari masing-masing blok kata membentuk kunci dan nilai yang kurang dari 85.
- Amazon TextractAnalyzeDocumentAPI mengembalikan pasangan nilai kunci kunci adalah salah satuGross Pay,GrossPay, atauGrossAmount, dengan keyakinan blok kunci-nilai yang kurang dari 60 dan kepercayaan dari masing-masing blok kata membentuk kunci dan nilai yang kurang dari 85.

```
{
    "Conditions": [
        {
            "ConditionType": "ImportantFormKeyConfidenceCheck",
            "ConditionParameters": {
                "ImportantFormKey": "Employee Name",
                "ImportantFormKeyAliases": [
                    "Name",
                    "EmployeeName"
                ],
                "KeyValueBlockConfidenceLessThan": 60,
                "WordBlockConfidenceLessThan": 85
            }
        },
        {
            "ConditionType": "ImportantFormKeyConfidenceCheck",
            "ConditionParameters": {
                "ImportantFormKey": "Pay Date",
                "ImportantFormKeyAliases": [
                    "PayDate",
                    "DateOfPay",
                    "pay-date"
                ],
                "KeyValueBlockConfidenceLessThan": 65,
```

```
                "WordBlockConfidenceLessThan": 85
            }
        },
        {
            "ConditionType": "ImportantFormKeyConfidenceCheck",
            "ConditionParameters": {
                "ImportantFormKey": "Gross Pay",
                "ImportantFormKeyAliases": [
                    "GrossPay",
                    "GrossAmount"
                ],
                "KeyValueBlockConfidenceLessThan": 60,
                "WordBlockConfidenceLessThan": 85
            }
        }
    ]
}
```

Contoh 2: Gunakan **`ImportantFormKeyConfidenceCheck`**

Dalam contoh berikut, jika Amazon Textract mendeteksi pasangan kunci-nilai yang percaya diri untuk blok kunci-nilai kurang dari 60 dan kurang dari 90 untuk setiap blok kata yang mendasarinya, itu menciptakan loop manusia. Peninjau manusia diminta untuk meninjau semua bentuk pasangan nilai kunci yang cocok dengan perbandingan nilai kepercayaan.

```
{
    "Conditions": [
        {
            "ConditionType": "ImportantFormKeyConfidenceCheck",
            "ConditionParameters": {
                "ImportantFormKey": "*",
                "KeyValueBlockConfidenceLessThan": 60,
                "WordBlockConfidenceLessThan": 90
            }
        }
    ]
}
```

Contoh 3: Gunakan Sampling

Dalam contoh berikut, 5% dari kesimpulan yang dihasilkan dari Amazon Textract`AnalyzeDocument`permintaan dikirim ke pekerja manusia untuk ditinjau. Semua pasangan nilai kunci yang terdeteksi yang dikembalikan oleh Amazon Textract dikirim ke pekerja untuk ditinjau.

```
{
  "Conditions": [
    {
      "ConditionType": "Sampling",
      "ConditionParameters": {
        "RandomSamplingPercentage": 5
      }
    }
  ]
}
```

Contoh 4: Gunakan **`MissingImportantFormKey`**

Pada contoh berikut, jika`Mailing Address`atau alias nya,`Mailing Address:`, hilang dari kunci yang terdeteksi oleh Amazon Textract, tinjauan manusia dimulai. Saat menggunakan template tugas pekerja default, UI pekerja meminta pekerja untuk mengidentifikasi kunci`Mailing Address`atau`Mailing Address:`dan nilai yang terkait.

```
{
    "ConditionType": "MissingImportantFormKey",
    "ConditionParameters": {
        "ImportantFormKey": "Mailing Address",
        "ImportantFormKeyAliases": ["Mailing Address:"]
    }
}
```

Contoh 5: Gunakan Sampling dan**`ImportantFormKeyConfidenceCheck`**dengan**`And`**operator

Dalam contoh ini, 5% pasangan kunci-nilai terdeteksi oleh Amazon Textract yang kuncinya adalah salah satunya`Pay Date`,`PayDate`,`DateOfPay`, atau`pay-date`, dengan keyakinan blok kunci-nilai kurang dari 65 dan kepercayaan masing-masing blok kata yang membentuk kunci dan nilai kurang dari 85, dikirim ke pekerja untuk ditinjau.

```
{
  "Conditions": [
    {
      "And": [
```

```
        {
          "ConditionType": "Sampling",
          "ConditionParameters": {
            "RandomSamplingPercentage": 5
          }
        },
        {
          "ConditionType": "ImportantFormKeyConfidenceCheck",
            "ConditionParameters": {
                "ImportantFormKey": "Pay Date",
                "ImportantFormKeyAliases": [
                    "PayDate",
                    "DateOfPay",
                    "pay-date"
                ],
                "KeyValueBlockConfidenceLessThan": 65,
                "WordBlockConfidenceLessThan": 85
            }
        }
      ]
    }
  ]
}
```

Contoh 6: Gunakan Sampling dan**`ImportantFormKeyConfidenceCheck`**dengan**`And`**operator

Gunakan contoh ini untuk mengonfigurasi alur kerja peninjauan manusia agar selalu mengirimkan kesimpulan kepercayaan rendah dari pasangan kunci-nilai tertentu untuk tinjauan manusia dan sampel inferensi keyakinan tinggi dari pasangan kunci-nilai pada tingkat tertentu.

Dalam contoh berikut, tinjauan manusia dimulai dengan salah satu cara berikut:

- Pasangan kunci-nilai terdeteksi yang kuncinya adalah salah satu`Pay Date`,`PayDate`,`DateOfPay`, atau`pay-date`, dengan key-value dan word block confidences kurang dari 60, dikirim untuk review manusia. Hanya`Pay Date`form key (dan alias) dan nilai terkait dikirim ke pekerja untuk ditinjau.
- 5% dari pasangan nilai kunci terdeteksi yang kunci adalah salah satu dari`Pay Date`,`PayDate`,`DateOfPay`, atau`pay-date`, dengan key-value dan word block confidences lebih besar dari 90, dikirim untuk review manusia. Hanya`Pay Date`form key (dan alias) dan nilai terkait dikirim ke pekerja untuk ditinjau.

```
{
```

```
  "Conditions": [
    {
      "Or": [
       {
          "ConditionType": "ImportantFormKeyConfidenceCheck",
            "ConditionParameters": {
                "ImportantFormKey": "Pay Date",
                "ImportantFormKeyAliases": [
                    "PayDate",
                    "DateOfPay",
                    "pay-date"
                ],
                "KeyValueBlockConfidenceLessThan": 60,
                "WordBlockConfidenceLessThan": 60
            }
       },
       {
            "And": [
                {
                    "ConditionType": "Sampling",
                    "ConditionParameters": {
                        "RandomSamplingPercentage": 5
                    }
                },
                {
                    "ConditionType": "ImportantFormKeyConfidenceCheck",
                        "ConditionParameters": {
                            "ImportantFormKey": "Pay Date",
                            "ImportantFormKeyAliases": [
                                "PayDate",
                                "DateOfPay",
                                "pay-date"
                        ],
                        "KeyValueBlockConfidenceLessThan": 90
                        "WordBlockConfidenceGreaterThan": 90
                    }
                }
            ]
        }
      ]
    }
  ]
}
```

Contoh 7: Gunakan Sampling dan**`ImportantFormKeyConfidenceCheck`**dengan**`Or`**operator

Pada contoh berikut, Amazon Textract`AnalyzeDocument`operasi mengembalikan pasangan nilai kunci yang merupakan salah satu`Pay Date`,`PayDate`,`DateOfPay`, atau`pay-date`, dengan keyakinan blok kunci-nilai kurang dari 65 dan kepercayaan dari masing-masing blok kata membentuk kunci dan nilai kurang dari 85. Selain itu, 5% dari semua bentuk lain memulai loop manusia. Untuk setiap formulir yang dipilih secara acak, semua pasangan nilai kunci yang terdeteksi untuk formulir tersebut dikirim ke manusia untuk ditinjau.

```
{
  "Conditions": [
    {
      "Or": [
        {
          "ConditionType": "Sampling",
          "ConditionParameters": {
            "RandomSamplingPercentage": 5
          }
        },
        {
           "ConditionType": "ImportantFormKeyConfidenceCheck",
            "ConditionParameters": {
                "ImportantFormKey": "Pay Date",
                "ImportantFormKeyAliases": [
                    "PayDate",
                    "DateOfPay",
                    "pay-date"
                ],
                "KeyValueBlockConfidenceLessThan": 65,
                "WordBlockConfidenceLessThan": 85
            }
          }
        }
      ]
    }
  ]
}
```

Gunakan Kondisi Aktivasi Loop Manusia JSON Schema dengan Amazon Rekognition

Saat digunakan dengan Amazon A2I, Amazon Rekognition`DetectModerationLabels`operasi mendukung input berikut di`ConditionType`parameter:

- `ModerationLabelConfidenceCheck`- Gunakan jenis kondisi ini untuk membuat loop manusia saat kepercayaan inferensi rendah untuk satu atau lebih label yang ditentukan.
- `Sampling`- Gunakan kondisi ini untuk menentukan persentase dari semua kesimpulan untuk dikirim ke manusia untuk ditinjau. Gunakan kondisi ini untuk melakukan hal berikut:
  - Audit model MS Anda dengan mengambil sampel secara acak semua kesimpulan model Anda dan mengirimkan persentase tertentu ke manusia untuk ditinjau.
  - Menggunakan`ModerationLabelConfidenceCheck`kondisi, sampel secara acak persentase dari kesimpulan yang memenuhi kondisi yang ditentukan dalam`ModerationLabelConfidenceCheck`untuk memulai loop manusia dan hanya mengirim persentase yang ditentukan kepada manusia untuk ditinjau.

> **Note**
>
> Jika Anda mengirim permintaan yang sama ke`DetectModerationLabels`beberapa kali, hasil`Sampling`tidak berubah untuk inferensi input itu. Misalnya, jika Anda membuat`DetectModerationLabels`permintaan sekali, dan`Sampling`tidak memulai loop manusia, permintaan berikutnya untuk`DetectModerationLabels`dengan konfigurasi yang sama tidak memulai loop manusia.

Saat membuat definisi alur, jika Anda menggunakan template tugas pekerja default yang disediakan di*Alur kerja peninjauan manusia*bagian dari Amazon SageMaker konsol, kesimpulan yang dikirim untuk peninjauan manusia oleh kondisi aktivasi ini disertakan dalam UI pekerja saat pekerja membuka tugas Anda. Jika Anda menggunakan template tugas pekerja kustom, Anda perlu menyertakan`<task.input.selectedAiServiceResponse.blocks>`elemen HTML kustom untuk mengakses kesimpulan ini. Untuk contoh template kustom yang menggunakan elemen HTML ini, lihat*Contoh kustom untuk Amazon Rekognition*.

`ModerationLabelConfidenceCheck`Input

Untuk`ModerationLabelConfidenceCheck ConditionType`, berikut ini`ConditionParameters`didukung:

- `ModerationLabelName`- Nama (peka huruf besar/kecil)*ModerationLabel*terdeteksi oleh Amazon Rekognition`DetectModerationLabels`operasi. Anda dapat menentukan nilai catch-all khusus (*) untuk menunjukkan label moderasi apa pun.
- `ConfidenceEquals`

- `ConfidenceLessThan`
- `ConfidenceLessThanEquals`
- `ConfidenceGreaterThan`
- `ConfidenceGreaterThanEquals`

Saat Anda menggunakan`ModerationLabelConfidenceCheck` `ConditionType`, Amazon A2I mengirimkan inferensi label untuk label yang Anda tentukan`ModerationLabelName`untuk review manusia.

Sampel Input

Kluster`Sampling` `ConditionType`mendukung`RandomSamplingPercentage` `ConditionParameters`. Input untuk`RandomSamplingPercentage`prameter harus bilangan real antara 0,01 dan 100. Angka ini mewakili persentase kesimpulan yang memenuhi syarat untuk tinjauan manusia yang dikirim ke manusia untuk ditinjau. Jika Anda menggunakan`Sampling`kondisi tanpa kondisi lain, angka ini mewakili persentase dari semua kesimpulan yang dihasilkan dari satu`DetectModerationLabel`permintaan yang dikirim ke manusia untuk ditinjau.

Contoh

Contoh 1: Gunakan**`ModerationLabelConfidenceCheck`**dengan**`And`**operator

Berikut ini adalah contoh dari`HumanLoopActivationConditions`kondisi memulai loop manusia ketika satu atau lebih dari kondisi berikut terpenuhi:

- Amazon Rekognition mendeteksi`Graphic Male Nudity`label moderasi dengan kepercayaan antara 90 dan 99.
- Amazon Rekognition mendeteksi`Graphic Female Nudity`label moderasi dengan kepercayaan antara 80 dan 99.

Perhatikan penggunaan`Or`dan`And`operator logis untuk model logika ini.

Meskipun hanya satu dari dua kondisi di bawah`Or`operator perlu mengevaluasi`true`agar loop manusia dibuat, Amazon Augmented AI mengevaluasi semua kondisi. Peninjau manusia diminta untuk meninjau label moderasi untuk semua kondisi yang dievaluasi`true`.

```
{
     "Conditions": [{
```

```
        "Or": [{
                "And": [{
                        "ConditionType": "ModerationLabelConfidenceCheck",
                        "ConditionParameters": {
                            "ModerationLabelName": "Graphic Male Nudity",
                            "ConfidenceLessThanEquals": 99
                        }
                    },
                    {
                        "ConditionType": "ModerationLabelConfidenceCheck",
                        "ConditionParameters": {
                            "ModerationLabelName": "Graphic Male Nudity",
                            "ConfidenceGreaterThanEquals": 90
                        }
                    }
                ]
            },
            {
                "And": [{
                        "ConditionType": "ModerationLabelConfidenceCheck",
                        "ConditionParameters": {
                            "ModerationLabelName": "Graphic Female Nudity",
                            "ConfidenceLessThanEquals": 99
                        }
                    },
                    {
                        "ConditionType": "ModerationLabelConfidenceCheck",
                        "ConditionParameters": {
                            "ModerationLabelName": "Graphic Female Nudity",
                            "ConfidenceGreaterThanEquals": 80
                        }
                    }
                ]
            }
        ]
    }]
}
```

Contoh 2: Gunakan**`ModerationLabelConfidenceCheck`**dengan nilai catch-all (*)

Pada contoh berikut, jika ada label moderasi dengan kepercayaan lebih besar dari atau sama dengan 75 terdeteksi, loop manusia dimulai. Peninjau manusia diminta untuk meninjau semua label moderasi dengan skor kepercayaan lebih besar dari atau sama dengan 75.

```
{
    "Conditions": [
        {
            "ConditionType": "ModerationLabelConfidenceCheck",
            "ConditionParameters": {
                "ModerationLabelName": "*",
                "ConfidenceGreaterThanEquals": 75
            }
        }
    ]
}
```

Contoh 3: Gunakan Sampling

Dalam contoh berikut, 5% dari Amazon Rekognition kesimpulan dari`DetectModerationLabels`permintaan dikirim ke pekerja manusia. Saat menggunakan template tugas pekerja default yang disediakan di SageMaker konsol, semua label moderasi yang dikembalikan oleh Amazon Rekognition dikirim ke pekerja untuk ditinjau.

```
{
  "Conditions": [
    {
      "ConditionType": "Sampling",
      "ConditionParameters": {
        "RandomSamplingPercentage": 5
      }
    }
  ]
}
```

Contoh 4: Gunakan Sampling dan**`ModerationLabelConfidenceCheck`**dengan**`And`**operator

Dalam contoh ini, 5% dari Amazon Rekognition kesimpulan dari`Graphic Male Nudity`label moderasi dengan keyakinan lebih besar dari 50 dikirim pekerja untuk ditinjau. Saat menggunakan template tugas pekerja default yang disediakan di SageMakerkonsol, hanya kesimpulan dari`Graphic Male Nudity`label dikirim ke pekerja untuk ditinjau.

```
{
  "Conditions": [
    {
      "And": [
```

```
        {
          "ConditionType": "Sampling",
          "ConditionParameters": {
            "RandomSamplingPercentage": 5
          }
        },
        {
          "ConditionType": "ModerationLabelConfidenceCheck",
          "ConditionParameters": {
            "ModerationLabelName": "Graphic Male Nudity",
            "ConfidenceGreaterThan": 50
          }
        }
      ]
    }
  ]
}
```

Contoh 5: Gunakan Sampling dan**`ModerationLabelConfidenceCheck`**dengan**`And`**operator

Gunakan contoh ini untuk mengonfigurasi alur kerja peninjauan manusia agar selalu mengirimkan inferensi rendah dari label tertentu untuk peninjauan manusia dan sampel inferensi kepercayaan tinggi label pada tingkat tertentu.

Dalam contoh berikut, tinjauan manusia dimulai dengan salah satu cara berikut:

- Kesimpulan untuk`Graphic Male Nudity`label moderasi dengan skor keyakinan kurang dari 60 selalu dikirim untuk ditinjau manusia. Hanya`Graphic Male Nudity`label dikirim ke pekerja untuk meninjau.
- 5% dari semua kesimpulan untuk`Graphic Male Nudity`label moderasi dengan skor keyakinan lebih besar dari 90 dikirim untuk peninjauan manusia. Hanya`Graphic Male Nudity`label dikirim ke pekerja untuk meninjau.

```
{
  "Conditions": [
    {
      "Or": [
        {
          "ConditionType": "ModerationLabelConfidenceCheck",
          "ConditionParameters": {
            "ModerationLabelName": "Graphic Male Nudity",
```

```
            "ConfidenceLessThan": 60
          }
        },
        {
            "And": [
                {
                    "ConditionType": "Sampling",
                    "ConditionParameters": {
                        "RandomSamplingPercentage": 5
                    }
                },
                {
                    "ConditionType": "ModerationLabelConfidenceCheck",
                    "ConditionParameters": {
                        "ModerationLabelName": "Graphic Male Nudity",
                        "ConfidenceGreaterThan": 90
                    }
                }
            ]
        }
      ]
    }
  ]
}
```

Contoh 6: Gunakan Sampling dan**`ModerationLabelConfidenceCheck`**dengan**`Or`**operator

Dalam contoh berikut, loop manusia dibuat jika respons inferensi Amazon Rekognition berisi label 'Ketelanjangan Pria Grafis' dengan keyakinan inferensi lebih besar dari 50. Selain itu, 5% dari semua kesimpulan lainnya memulai loop manusia.

```
{
  "Conditions": [
    {
      "Or": [
        {
          "ConditionType": "Sampling",
          "ConditionParameters": {
            "RandomSamplingPercentage": 5
          }
        },
        {
          "ConditionType": "ModerationLabelConfidenceCheck",
```

```
          "ConditionParameters": {
            "ModerationLabelName": "Graphic Male Nudity",
            "ConfidenceGreaterThan": 50
          }
        }
      ]
    }
  ]
}
```

Menghapus Alur Kerja Tinjauan Manusia

Saat Anda menghapus alur kerja tinjauan manusia atau menghapus alur kerjaAWSakun saat loop manusia sedang dalam proses, status alur kerja tinjauan manusia Anda berubah menjadi`Deleting`. Amazon A2I secara otomatis berhenti dan menghapus semua loop manusia terkait jika pekerja belum memulai tugas yang dibuat oleh loop manusia tersebut. Jika pekerja manusia sudah mengerjakan tugas, tugas itu terus tersedia sampai selesai atau berakhir. Selama pekerja masih mengerjakan tugas, status alur kerja tinjauan manusia Anda adalah`Deleting`. Jika tugas ini selesai, hasilnya akan disimpan dalam bucket Amazon S3 yang ditentukan dalam definisi aliran Anda.

Menghapus definisi aliran tidak menghapus jawaban pekerja dari bucket S3 Anda. Jika tugas selesai, tetapi Anda menghapusAWSakun, hasilnya disimpan dalam bucket layanan Augmented AI selama 30 hari dan kemudian dihapus secara permanen.

Setelah semua loop manusia telah dihapus, alur kerja tinjauan manusia dihapus secara permanen. Ketika alur kerja tinjauan manusia telah dihapus, Anda dapat menggunakan kembali namanya untuk membuat alur kerja tinjauan manusia baru.

Anda mungkin ingin menghapus alur kerja tinjauan manusia untuk salah satu alasan berikut:

- Anda telah mengirim data ke satu set pengulas manusia dan Anda ingin menghapus semua loop manusia yang tidak dimulai karena Anda tidak ingin pekerja tersebut bekerja pada tugas-tugas itu lagi.
- Template tugas pekerja yang digunakan untuk menghasilkan UI pekerja Anda tidak merender dengan benar atau tidak berfungsi seperti yang diharapkan.

Setelah Anda menghapus alur kerja manusia, perubahan berikut terjadi:

- Alur kerja tinjauan manusia tidak lagi muncul diAlur kerja manusiahalaman di area Augmented AI konsol Amazon SageMaker.

- Bila Anda menggunakan nama alur kerja tinjauan manusia sebagai masukan ke operasi API`DescribeFlowDefinition`atau`DeleteFlowDefinition`, Augmented AI mengembalikan`ResourceNotFound`kesalahan.
- Saat Anda menggunakan`ListFlowDefinitions`, alur kerja tinjauan manusia yang dihapus tidak disertakan dalam hasil.
- Bila Anda menggunakan alur kerja tinjauan manusia ARN sebagai masukan ke operasi Augmented AI Runtime API`ListHumanLoops`, Augmented AI mengembalikan`ResourceNotFoundException`.

Menghapus Definisi Aliran Menggunakan Konsol atau SageMaker API

Anda dapat menghapus alur kerja tinjauan manusia padaAlur kerja manusiahalaman di area Augmented AI dari konsol SageMaker atau dengan menggunakan SageMaker API.

Definisi aliran hanya dapat dihapus jika statusnya`Active`.

Menghapus alur kerja manusia (konsol)

1. Arahkan ke konsol Augmented AI dihttps://console.aws.amazon.com/a2i/.
2. Di panel navigasi, di bawahAugmented AIbagian, pilihAlur kerja manusia.
3. Pilih nama hyperlink dari alur kerja tinjauan manusia yang ingin Anda hapus.
4. PadaRingkasanhalaman alur kerja tinjauan manusia Anda, pilihHapus.
5. Di kotak dialog yang meminta Anda mengonfirmasi bahwa Anda ingin menghapus alur kerja manusia, pilihHapus.

Anda secara otomatis diarahkan keAlur kerja manusiahalaman. Saat alur kerja peninjauan manusia Anda dihapus, statusnyaMenghapusmuncul di kolom status untuk alur kerja itu. Setelah dihapus, itu tidak muncul dalam daftar alur kerja di halaman ini.

Menghapus alur kerja tinjauan manusia (API)

Anda dapat menghapus alur kerja tinjauan manusia (definisi aliran) menggunakan SageMakerDeleteFlowDefinitionOperasi API. Operasi API ini didukung melaluiAWS CLIdan aberbagai SDK bahasa tertentu. Tabel berikut menunjukkan permintaan contoh menggunakan SDK for Python (Boto3) danAWS CLIuntuk menghapus alur kerja tinjauan manusia,*`example-flow-definition`*.

AWS SDK for Python (Boto3)

Contoh permintaan berikut menggunakan SDK for Python (Boto3) untuk menghapus alur kerja manusia. Untuk informasi selengkapnya, lihatdelete_flow_definitiondiAWSReferensi API SDK for Python (Boto).

```
import boto3

sagemaker_client = boto3.client('sagemaker')
response = sagemaker_client.delete_flow_definition(FlowDefinitionName='example-flow-definition')
```

AWS CLI

Contoh permintaan berikut menggunakanAWSCLI untuk menghapus alur kerja tinjauan manusia. Untuk informasi selengkapnya, lihatmenghapus-flow-definitiondiAWS CLIReferensi Perintah.

```
$ aws sagemaker delete-flow-definition --flow-definition-name 'example-flow-definition'
```

Jika tindakan berhasil, Augmented AI mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Membuat dan Memulai Loop Manusia

Lingkaran manusia memulai alur kerja peninjauan manusia Anda dan mengirimkan tugas peninjauan data ke pekerja manusia. Saat Anda menggunakan salah satu jenis tugas bawaan Amazon A2I,AWS layanan terkait akan membuat dan memulai loop manusia atas nama Anda ketika kondisi yang ditentukan dalam definisi alur Anda terpenuhi. Jika tidak ada kondisi yang ditentukan dalam definisi aliran Anda, loop manusia dibuat untuk setiap objek. Saat menggunakan Amazon A2I untuk tugas khusus, loop manusia dimulai saat aplikasi Anda meneleponStartHumanLoop.

Gunakan petunjuk berikut untuk mengonfigurasi loop manusia dengan jenis tugas bawaan Amazon Rekognition atau Amazon Textract dan jenis tugas khusus.

Prasyarat

Untuk membuat dan memulai loop manusia, Anda harus melampirkanAmazonAugmentedAIFullAccess kebijakan ke penggunaAWS Identity and Access Management (IAM) atau peran yang mengkonfigurasi atau memulai loop

manusia. Ini adalah identitas yang Anda gunakan untuk mengkonfigurasi loop manusia menggunakan`HumanLoopConfig` untuk built-in jenis tugas. Untuk jenis tugas khusus, ini adalah identitas yang Anda gunakan untuk meneleponStartHumanLoop.

Selain itu, saat menggunakan tipe tugas bawaan, pengguna atau peran Anda harus memiliki izin untuk menjalankan operasi API dariAWS layanan yang terkait dengan jenis tugas Anda. Misalnya, jika Anda menggunakan Amazon Rekognition dengan Augmented AI, Anda harus melampirkan izin yang diperlukan untuk meneleponDetectModerationLabels. Untuk contoh kebijakan berbasis identitas yang dapat Anda gunakan untuk memberikan izin ini, lihat Contoh Kebijakan Berbasis Identitas Amazon Rekognition dan Contoh Kebijakan Berbasis Identitas Amazon Textract. Anda juga dapat menggunakan kebijakan yang lebih umumAmazonAugmentedAIIntegratedAPIAccess untuk memberikan izin ini. Untuk informasi selengkapnya, lihat Membuat Pengguna Dengan Izin untuk Memanggil Amazon A2I, Amazon Textract, dan Amazon Rekognition API Operations.

Untuk membuat dan memulai loop manusia, Anda memerlukan definisi aliran ARN. Untuk mempelajari cara membuat definisi alur (atau alur kerja tinjauan manusia), lihatBuat Alur Kerja Tinjauan Manusia.

> **Important**
>
> Amazon A2I mewajibkan semua bucket S3 yang berisi data gambar input loop manusia agar kebijakan CORS terpasang. Untuk mempelajari tentang perubahan ini, lihatCORS.

Membuat dan Memulai Loop Manusia untuk Tipe Tugas Built-in

Untuk memulai loop manusia menggunakan tipe tugas bawaan, gunakan API layanan terkait untuk menyediakan data input Anda dan untuk mengonfigurasi loop manusia. Untuk Amazon Textract, Anda menggunakan operasiAnalyzeDocument API. Untuk Amazon Rekognition, Anda menggunakan operasiDetectModerationLabels API. Anda dapat menggunakan SDK khusus bahasaAWS CLI atau untuk membuat permintaan menggunakan operasi API ini.

> **Important**
>
> Ketika Anda membuat loop manusia menggunakan built-in jenis tugas, Anda dapat menggunakanDataAttributes untuk menentukan satu setContentClassifiers terkait dengan masukan yang disediakan untukStartHumanLoop operasi. Gunakan pengklasifikasi konten untuk menyatakan bahwa konten Anda bebas dari informasi identitas pribadi atau konten dewasa.

Untuk menggunakan Amazon Mechanical Turk, pastikan data Anda bebas dari informasi yang dapat diidentifikasi secara pribadi, termasuk informasi kesehatan yang dilindungi di bawah HIPAA. Sertakan pengklasifikasi`FreeOfPersonallyIdentifiableInformation` konten. Jika Anda tidak menggunakan pengklasifikasi konten ini, SageMaker tidak mengirim tugas Anda ke Mechanical Turk. Jika data Anda bebas dari konten dewasa, sertakan juga`'FreeOfAdultContent'` pengklasifikasi. Jika Anda tidak menggunakan pengklasifikasi konten ini, SageMaker dapat membatasi pekerja Mechanical Turk yang dapat melihat tugas Anda.

Setelah Anda memulai pekerjaan dengan menggunakan APIAWS layanan tipe tugas bawaan, Amazon A2I memantau hasil inferensi layanan tersebut. Misalnya, saat menjalankan pekerjaan dengan Amazon Rekognition, Amazon A2I memeriksa skor kepercayaan inferensi untuk setiap gambar dan membandingkannya dengan ambang kepercayaan yang ditentukan dalam definisi alur Anda. Jika kondisi untuk memulai tugas peninjauan manusia terpenuhi, atau jika Anda tidak menentukan kondisi dalam definisi alur Anda, tugas peninjauan manusia akan dikirim ke pekerja.

Membuat Amazon Textract Human Loop

Amazon A2I terintegrasi dengan Amazon Textract sehingga Anda dapat mengonfigurasi dan memulai loop manusia menggunakan Amazon Textract API. Untuk mengirim file dokumen ke Amazon Textract untuk analisis dokumen, Anda menggunakan operasi Amazon Textract`AnalyzeDocument` API. Untuk menambahkan loop manusia ke pekerjaan analisis dokumen ini, Anda harus mengkonfigurasi parameter`HumanLoopConfig`.

Saat Anda mengonfigurasi loop manusia, definisi alur yang Anda tentukan`HumanLoopConfig` harus ditempatkan diAWS Wilayah yang sama dengan bucket yang diidentifikasi dalam`BucketDocument` parameter.`FlowDefinitionArn`

Tabel berikut menunjukkan contoh bagaimana menggunakan operasi ini denganAWS CLI danAWS SDK for Python (Boto3).

AWS SDK for Python (Boto3)

Contoh permintaan berikut menggunakan SDK for Python (Boto3). Untuk informasi selengkapnya, lihat analyze_document di AWSSDK for Python (Boto) API Reference.

```
import boto3
```

```
textract = boto3.client('textract', aws_region)

response = textract.analyze_document(
            Document={'S3Object': {'Bucket': bucket_name, 'Name': document_name}},
            FeatureTypes=["TABLES", "FORMS"],
            HumanLoopConfig={
                'FlowDefinitionArn':
 'arn:aws:sagemaker:aws_region:aws_account_number:flow-definition/flow_def_name',
                'HumanLoopName': 'human_loop_name',
                'DataAttributes': {'ContentClassifiers':
 ['FreeOfPersonallyIdentifiableInformation','FreeOfAdultContent']}
            }
          )
```

AWS CLI

Contoh permintaan berikut menggunakan AWS CLI. Untuk informasi selengkapnya, lihat analyze-document di AWS CLICommand Reference.

```
$ aws textract analyze-document \
     --document '{"S3Object":{"Bucket":"bucket_name","Name":"document_name"}}' \
     --human-loop-config
 HumanLoopName="human_loop_name",FlowDefinitionArn="arn:aws:sagemaker:aws-
region:aws_account_number:flow-
definition/
flow_def_name",DataAttributes='{ContentClassifiers=["FreeOfPersonallyIdentifiableInformation
 "FreeOfAdultContent"]}' \
     --feature-types '["TABLES", "FORMS"]'
```

```
$ aws textract analyze-document \
     --document '{"S3Object":{"Bucket":"bucket_name","Name":"document_name"}}' \
     --human-loop-config \

 '{"HumanLoopName":"human_loop_name","FlowDefinitionArn":"arn:aws:sagemaker:aws_region:aws_a
definition/flow_def_name","DataAttributes": {"ContentClassifiers":
["FreeOfPersonallyIdentifiableInformation","FreeOfAdultContent"]}}' \
     --feature-types '["TABLES", "FORMS"]'
```

Setelah Anda menjalankan`AnalyzeDocument` dengan loop manusia yang dikonfigurasi, Amazon A2I memantau hasilnya`AnalyzeDocument` dan memeriksanya terhadap kondisi aktivasi definisi

aliran. Jika skor kepercayaan inferensi Amazon Textract untuk satu atau beberapa pasangan nilai kunci memenuhi ketentuan untuk ditinjau, Amazon A2I memulai loop tinjauan manusia dan menyertakan `HumanLoopActivationOutput` objek dalam `AnalyzeDocument` respons.

Membuat Amazon Rekognition Human Loop

Amazon A2I terintegrasi dengan Amazon Rekognition sehingga Anda dapat mengonfigurasi dan memulai loop manusia menggunakan Amazon Rekognition API. Untuk mengirim gambar ke Amazon Rekognition untuk moderasi konten, Anda menggunakan operasi Amazon Rekognition `DetectModerationLabels` API. Untuk mengkonfigurasi loop manusia, atur `HumanLoopConfig` parameter saat Anda mengkonfigurasi `DetectModerationLabels`.

Saat Anda mengonfigurasi loop manusia, definisi aliran yang Anda tentukan `HumanLoopConfig` harus ditempatkan di AWS Wilayah yang sama dengan bucket S3 yang diidentifikasi dalam `BucketImage` parameter. `FlowDefinitionArn`

Tabel berikut menunjukkan contoh bagaimana menggunakan operasi ini dengan AWS CLI dan AWS SDK for Python (Boto3).

AWS SDK for Python (Boto3)

Contoh permintaan berikut menggunakan SDK for Python (Boto3). Untuk informasi selengkapnya, lihat detect_moderation_labels di AWSSDK for Python (Boto) API Reference.

```
import boto3

rekognition = boto3.client("rekognition", aws_region)

response = rekognition.detect_moderation_labels( \
        Image={'S3Object': {'Bucket': bucket_name, 'Name': image_name}}, \
        HumanLoopConfig={ \
            'HumanLoopName': 'human_loop_name', \
            'FlowDefinitionArn': , 
 "arn:aws:sagemaker:aws_region:aws_account_number:flow-definition/flow_def_name" \
            'DataAttributes': {'ContentClassifiers': 
 ['FreeOfPersonallyIdentifiableInformation','FreeOfAdultContent']}
         })
```

AWS CLI

Contoh permintaan berikut menggunakan AWS CLI. Untuk informasi selengkapnya, lihat detect-moderation-labels di Referensi AWS CLI Perintah.

```
$ aws rekognition detect-moderation-labels \
    --image "S3Object={Bucket='bucket_name',Name='image_name'}" \
    --human-loop-config
 HumanLoopName="human_loop_name",FlowDefinitionArn="arn:aws:sagemaker:aws_region:aws_account
definition/
flow_def_name",DataAttributes='{ContentClassifiers=["FreeOfPersonallyIdentifiableInformation
 "FreeOfAdultContent"]}'
```

```
$ aws rekognition detect-moderation-labels \
    --image "S3Object={Bucket='bucket_name',Name='image_name'}" \
    --human-loop-config \
        '{"HumanLoopName": "human_loop_name", "FlowDefinitionArn":
 "arn:aws:sagemaker:aws_region:aws_account_number:flow-
definition/flow_def_name", "DataAttributes": {"ContentClassifiers":
 ["FreeOfPersonallyIdentifiableInformation", "FreeOfAdultContent"]}}'
```

Setelah Anda menjalankan`DetectModerationLabels` dengan loop manusia yang dikonfigurasi, Amazon A2I memantau hasilnya`DetectModerationLabels` dan memeriksanya terhadap kondisi aktivasi definisi aliran. Jika skor kepercayaan inferensi Amazon Rekognition untuk gambar memenuhi ketentuan peninjauan, Amazon A2I memulai loop tinjauan manusia dan menyertakan elemen respons`HumanLoopActivationOutput` dalam`DetectModerationLabels` respons.

Membuat dan Memulai Loop Manusia untuk Jenis Tugas Kustom

Untuk mengonfigurasi loop manusia untuk tugas peninjauan manusia kustom, gunakan`StartHumanLoop` operasi dalam aplikasi Anda. Bagian ini memberikan contoh permintaan loop manusia menggunakanAWS SDK for Python (Boto3) and theAWS Command Line Interface (AWS CLI). Untuk dokumentasi tentang SDK khusus bahasa lain yang mendukung`StartHumanLoop`, gunakan bagian Lihat Juga StartHumanLoopdalam dokumentasi Amazon Augmented AI Runtime API. Lihat contoh yang menunjukkan cara menggunakan Amazon A2I dengan jenis tugas khusus.Contoh Kasus Penggunaan Amazon A2I

Prasyarat

Untuk menyelesaikan prosedur ini, Anda memerlukan:

- Masukan data diformat sebagai representasi string dari file JSON-diformat
- Amazon Resource Name (ARN) dari definisi aliran Anda

Untuk mengkonfigurasi loop manusia

1. Untuk`DataAttributes`, tentukan satu set`ContentClassifiers` terkait dengan input yang disediakan untuk`StartHumanLoop` operasi. Gunakan pengklasifikasi konten untuk menyatakan bahwa konten Anda bebas dari informasi identitas pribadi atau konten dewasa.

   Untuk menggunakan Amazon Mechanical Turk, pastikan data Anda bebas dari informasi yang dapat diidentifikasi secara pribadi, termasuk informasi kesehatan yang dilindungi di bawah HIPAA, dan menyertakan pengklasifikasi`FreeOfPersonallyIdentifiableInformation` konten. Jika Anda tidak menggunakan pengklasifikasi konten ini, SageMaker tidak mengirim tugas Anda ke Mechanical Turk. Jika data Anda bebas dari konten dewasa, sertakan juga`'FreeOfAdultContent'` pengklasifikasi. Jika Anda tidak menggunakan pengklasifikasi konten ini, SageMaker dapat membatasi pekerja Mechanical Turk yang dapat melihat tugas Anda.
2. Untuk`FlowDefinitionArn`, masukkan Amazon Resource Name (ARN) dari definisi aliran Anda.
3. Untuk`HumanLoopInput`, masukkan data masukan Anda sebagai representasi string dari file berformat JSON. Susun data input dan template tugas pekerja kustom Anda sehingga data input Anda ditampilkan dengan benar kepada pekerja manusia saat Anda memulai loop manusia. LihatPratinjau Template Tugas Pekerja untuk mempelajari cara melihat pratinjau template tugas pekerja kustom Anda.
4. Untuk`HumanLoopName`, masukkan nama untuk loop manusia. Nama harus unik dalam Wilayah di akun Anda dan dapat memiliki hingga 63 karakter. Karakter yang benar adalah a-z, 0-9, dan - (tanda hubung).

Untuk memulai loop manusia

- Untuk memulai loop manusia, kirimkan permintaan yang mirip dengan contoh berikut menggunakan SDK khusus bahasa pilihan Anda.

AWS SDK for Python (Boto3)

Contoh permintaan berikut menggunakan SDK for Python (Boto3). Untuk informasi selengkapnya, lihat Boto 3 Augmented AI Runtime di AWSSDK for Python (Boto) API Reference.

```
import boto3
```

```
a2i_runtime_client = boto3.client('sagemaker-a2i-runtime')

response = a2i_runtime_client.start_human_loop(
    HumanLoopName='human_loop_name',
    FlowDefinitionArn='arn:aws:sagemaker:aws-region:xyz:flow-
definition/flow_def_name',
    HumanLoopInput={
        'InputContent': '{"InputContent": {\"prompt\":\"What is the answer?\"}}'
    },
    DataAttributes={
        'ContentClassifiers': [
            'FreeOfPersonallyIdentifiableInformation'|'FreeOfAdultContent',
        ]
    }
)
```

AWS CLI

Contoh permintaan berikut menggunakan AWS CLI. Untuk informasi selengkapnya, lihat start-human-loopdi ReferensiAWS CLI Perintah.

```
$ aws sagemaker-a2i-runtime start-human-loop
        --flow-definition-arn 'arn:aws:sagemaker:aws_region:xyz:flow-
definition/flow_def_name' \
        --human-loop-name 'human_loop_name' \
        --human-loop-input '{"InputContent": "{\"prompt\":\"What is the answer?
\"}"}' \
        --data-attributes
 ContentClassifiers="FreeOfPersonallyIdentifiableInformation","FreeOfAdultContent" \
```

Ketika Anda berhasil memulai loop manusia dengan memohon`StartHumanLoop` secara langsung, respon termasuk`HumanLoopARN` dan`HumanLoopActivationResults` objek yang diatur ke`NULL`. Anda dapat menggunakan ini nama loop manusia untuk memantau dan mengelola loop manusia Anda.

Langkah Langkah Langkah Langkah Langkah Langkah Langkah Langkah Langkah

Setelah memulai loop manusia, Anda dapat mengelola dan memantaunya dengan Amazon Augmented AI Runtime API dan Amazon CloudWatch Events. Untuk mempelajari informasi lebih lanjut, lihat Pantau dan Kelola Loop Manusia Anda.

Menghapus Loop Manusia

Ketika Anda menghapus loop manusia, status berubah menjadi`Deleting`. Ketika loop manusia dihapus, tugas peninjauan manusia terkait tidak lagi tersedia untuk pekerja. Anda mungkin ingin menghapus loop manusia dalam salah satu keadaan berikut:

- Template tugas pekerja yang digunakan untuk menghasilkan antarmuka pengguna pekerja Anda tidak membuat benar atau tidak berfungsi seperti yang diharapkan.
- Sebuah objek data tunggal secara tidak sengaja dikirim ke pekerja beberapa kali.
- Anda tidak perlu lagi objek data yang ditinjau oleh manusia.

Jika status loop manusia`InProgress`, Anda harus menghentikan loop manusia sebelum menghapusnya. Ketika Anda menghentikan loop manusia, status berubah menjadi`Stopping`sementara sedang dihentikan. Saat status berubah menjadi`Stopped`, Anda bisa menghapus loop manusia.

Jika pekerja manusia sudah mengerjakan tugas ketika Anda menghentikan loop manusia terkait, tugas itu terus tersedia sampai selesai atau berakhir. Selama pekerja masih mengerjakan tugas, status lingkaran manusia Anda`Stopping`. Jika tugas ini selesai, hasilnya akan disimpan dalam URI bucket Amazon S3 yang ditentukan dalam alur kerja peninjauan manusia Anda. Jika pekerja meninggalkan tugas tanpa mengirimkan pekerjaan, itu dihentikan dan pekerja tidak dapat kembali ke tugas. Jika tidak ada pekerja yang mulai mengerjakan tugas, segera dihentikan.

Jika Anda menghapusAWSakun yang digunakan untuk membuat loop manusia, itu dihentikan dan dihapus secara otomatis.

Retensi dan Penghapusan Data Loop Manusia

Ketika pekerja manusia menyelesaikan tugas peninjauan manusia, hasilnya akan disimpan dalam bucket keluaran Amazon S3 yang Anda tentukan dalam alur kerja tinjauan manusia yang digunakan untuk membuat loop manusia. Menghapus atau menghentikan loop manusia tidak menghapus jawaban pekerja dari bucket S3 Anda.

Selain itu, Amazon A2I untuk sementara menyimpan data input dan output loop manusia secara internal karena alasan berikut:

- Jika Anda mengkonfigurasi loop manusia sehingga objek data tunggal dikirim ke beberapa pekerja untuk ditinjau, Amazon A2I tidak menulis data output ke bucket S3 Anda sampai semua pekerja

menyelesaikan tugas peninjauan. Amazon A2I menyimpan jawaban parsial-jawaban dari pekerja individu—secara internal sehingga dapat menulis hasil lengkap ke bucket S3 Anda.

- Jika Anda melaporkan hasil tinjauan manusia berkualitas rendah, Amazon A2I dapat menyelidiki dan menanggapi masalah Anda.
- Jika Anda kehilangan akses ke atau menghapus bucket S3 output yang ditentukan dalam alur kerja tinjauan manusia yang digunakan untuk membuat loop manusia, dan tugas tersebut telah dikirim ke satu atau lebih pekerja, Amazon A2I membutuhkan tempat untuk menyimpan sementara hasil tinjauan manusia.

Amazon A2I menghapus data ini secara internal 30 hari setelah perubahan status human loop menjadi salah satu hal berikut:`Deleted`,`Stopped`, atau`Completed`. Dengan kata lain, data dihapus 30 hari setelah loop manusia telah selesai, dihentikan, atau dihapus. Selain itu, data ini akan dihapus setelah 30 hari jika Anda menutupAWSakun yang digunakan untuk membuat loop manusia terkait.

Menghentikan dan Menghapus Definisi Aliran Menggunakan Konsol atau API Amazon A2I

Anda dapat menghentikan dan menghapus loop manusia di konsol Augmented AI atau dengan menggunakan SageMaker API. Ketika loop manusia telah dihapus, status berubah menjadi`Deleted`.

Menghapus loop manusia (konsol)

1. Arahkan ke konsol Augmented AI dihttps://console.aws.amazon.com/a2i/.
2. Di panel navigasi, di bawahAugmented AIbagian, pilihAlur kerja tinjauan manusia.
3. Pilih nama hyperlink dari alur kerja tinjauan manusia yang Anda gunakan untuk membuat loop manusia yang ingin Anda hapus.
4. DiLoop Manusiabagian di bagian bawah halaman, pilih loop manusia yang ingin Anda hentikan dan hapus.
5. Jika status loop manusia`Completed`,`Stopped`, atau`Failed`, pilihHapus.

   Jika loop manusiaStatusadalah`InProgress`, pilihBerhenti. Saat status berubah menjadiDihentikan, pilihHapus.

Menghapus loop manusia (API)

1. Periksa status loop manusia Anda menggunakan operasi Augmented AI Runtime API`DescribeHumanLoop`. Lihat contoh menggunakan operasi ini dalam tabel berikut.

   AWS SDK for Python (Boto3)

   Contoh berikut menggunakan SDK for Python (Boto3) untuk menggambarkan loop manusia bernama*`contoh-manusia-loop`*. Untuk informasi selengkapnya, lihatdescribe_human_loopdiAWSReferensi API SDK for Python (Boto).

   ```
   import boto3

   a2i_runtime_client = boto3.client('sagemaker-a2i-runtime')
   response = a2i_runtime_client.describe_human_loop(HumanLoopName='example-human-loop')
   human_loop_status = response['HumanLoopStatus']
   print(f'example-human-loop status is: {human_loop_status}')
   ```

   AWS CLI

   Contoh berikut menggunakanAWSCLI untuk menggambarkan lingkaran manusia bernama*`contoh-manusia-loop`*. Untuk informasi selengkapnya, lihatmenjelaskan-manusia-loopdiAWS CLIReferensi Perintah.

   ```
   $ aws sagemaker-a2i-runtime describe-human-loop --human-loop-name 'example-human-loop'
   ```

2. Jika status definisi aliran`Completed`,`Stopped`, atau`Failed`, hapus definisi aliran menggunakan operasi Augmented AI Runtime API`DeleteHumanLoop`.

   AWS SDK for Python (Boto3)

   Contoh berikut menggunakan SDK for Python (Boto3) untuk menghapus loop manusia bernama*`contoh-manusia-loop`*. Untuk informasi selengkapnya, lihatdelete_human_loopdiAWSReferensi API SDK for Python (Boto).

   ```
   import boto3

   a2i_runtime_client = boto3.client('sagemaker-a2i-runtime')
   ```

```
response = a2i_runtime_client.delete_human_loop(HumanLoopName='example-human-loop')
```

AWS CLI

Contoh berikut menggunakanAWSCLI untuk menghapus loop manusia bernama*contoh-manusia-loop*. Untuk informasi selengkapnya, lihatmenghapus-manusia-loopdiAWS CLIReferensi Perintah.

```
$ aws sagemaker-a2i-runtime delete-human-loop --human-loop-name 'example-human-loop'
```

Jika status loop manusia`InProgress`, hentikan loop manusia menggunakan`StopHumanLoop`dan kemudian menggunakan`DeleteHumanLoop`untuk menghapusnya.

AWS SDK for Python (Boto3)

Contoh berikut menggunakan SDK for Python (Boto3) untuk menggambarkan loop manusia bernama*contoh-manusia-loop*. Untuk informasi selengkapnya, lihatstop_human_loopdiAWSReferensi API SDK for Python (Boto).

```
import boto3

a2i_runtime_client = boto3.client('sagemaker-a2i-runtime')
response = a2i_runtime_client.stop_human_loop(HumanLoopName='example-human-loop')
```

AWS CLI

Contoh berikut menggunakanAWSCLI untuk menggambarkan lingkaran manusia bernama*contoh-manusia-loop*. Untuk informasi selengkapnya, lihatstop-manusia-loopdiAWS CLIReferensi Perintah.

```
$ aws sagemaker-a2i-runtime stop-human-loop --human-loop-name 'example-human-loop'
```

Membuat dan Mengelola Template Tugas Pekerja

Anda dapat membuat antarmuka pengguna tugas untuk pekerja Anda dengan membuatTemplat Tugas Pekerja. Template tugas pekerja adalah file HTML yang digunakan untuk menampilkan data input dan instruksi untuk membantu pekerja menyelesaikan tugas Anda.

Untuk jenis tugas Amazon Rekognition atau Amazon Textract Texact, Anda dapat menyesuaikan template tugas pekerja pra-dibuat menggunakan antarmuka pengguna grafis (GUI) dan menghindari berinteraksi dengan kode HTML. Untuk opsi ini, gunakan petunjuk diMembuat Alur Kerja Tinjauan Manusia (Konsol)untuk membuat alur kerja tinjauan manusia dan menyesuaikan template tugas pekerja Anda di konsol Amazon SageMaker. Setelah Anda membuat template menggunakan petunjuk ini, itu akan muncul di halaman template tugas pekerjaKonsol Augmented AI.

Jika Anda membuat alur kerja tinjauan manusia untuk jenis tugas kustom, Anda harus membuatTemplat Tugas Pekerja Khususmenggunakan kode HTML. Untuk informasi selengkapnya, lihat Buat Templat Tugas Pekerja Kustom.

Jika Anda membuat template Anda menggunakan HTML, Anda harus menggunakan template ini untuk menghasilkan Amazon A2IAmazon Resource Name (ARN)Konsol Amazon A2I. ARN ini memiliki format berikut: `arn:aws:sagemaker:<aws-region>:<aws-account-number>:human-task-ui/<template-name>`. ARN ini dikaitkan dengan sumber daya template tugas pekerja yang dapat Anda gunakan dalam satu atau lebih alur kerja tinjauan manusia (definisi aliran).

Menghasilkan tugas manusia UI ARN menggunakan template tugas pekerja dengan mengikuti petunjuk yang ditemukan diBuat Template Tugas Pekerjaatau dengan menggunakan`CreateHumanTaskUi`Operasi API.

Topik

- Membuat dan Menghapus Template Tugas Pekerja
- Buat Templat Tugas Pekerja Kustom
- Membuat Instruksi Pekerja yang Baik

Membuat dan Menghapus Template Tugas Pekerja

Anda dapat menggunakan template pekerja untuk menyesuaikan antarmuka dan instruksi yang dilihat pekerja Anda saat mengerjakan tugas Anda. Gunakan petunjuk di halaman ini untuk

membuat template tugas pekerja di area Augmented AI konsol Amazon SageMaker. Template starter disediakan untuk tugas Amazon Textract dan Amazon Rekognition. Untuk mempelajari cara menyesuaikan template Anda menggunakan elemen kerumunan HTML, lihatBuat Templat Tugas Pekerja Kustom.

Ketika Anda membuat template pekerja di halaman template tugas pekerja dari area Augmented AI konsol SageMaker, template tugas pekerja ARN dihasilkan. Gunakan ARN ini sebagai masukan untuk`HumanTaskUiArn`saat Anda membuat definisi aliran menggunakan operasi API`CreateFlowDefinition`. Anda dapat memilih template ini saat membuat alur kerja tinjauan manusia di halaman alur kerja tinjauan manusia konsol.

Jika Anda membuat sumber daya template tugas pekerja untuk jenis tugas Amazon Textract atau Amazon Rekognition, Anda dapat melihat pratinjau UI pekerja yang dihasilkan dari template Anda di halaman konsol template tugas pekerja. Anda harus melampirkan kebijakan yang dijelaskan dalamAktifkan Pratinjau Template Tugas Pekerja untuk peran IAM yang Anda gunakan untuk melihat pratinjau template.

Buat Template Tugas Pekerja

Anda dapat membuat template tugas pekerja menggunakan konsol SageMaker dan menggunakan operasi SageMaker API`CreateHumanTaskUi`.

Membuat templat tugas pekerja (konsol)

1. Buka konsol Amazon A2Ihttps://console.aws.amazon.com/a2i/.
2. Di bawahAmazon Augmented AIdi panel navigasi sebelah kiri, pilihPekerja tugas template.
3. Pilih Buat templat.
4. MasukNama templat, masukkan nama yang unik.
5. (Opsional) MasukkanPeran IAMyang memberikan Amazon A2I izin yang diperlukan untuk memanggil layanan atas nama Anda.
6. MasukJenis templat, pilih tipe templat dari daftar dropdown. Jika Anda membuat template untukEkstraksi bentuk teksatauModerasi gambar rekognisitugas, pilih opsi yang sesuai.
7. Masukkan elemen template kustom Anda sebagai berikut:

   - Jika Anda memilih templat tugas Amazon Textract atau Amazon Rekognition, makaEditor templatautopopulates dengan template default yang dapat Anda sesuaikan.
   - Jika Anda menggunakan template kustom, masukkan template yang telah ditentukan di editor.

8. (Opsional) Untuk menyelesaikan langkah ini, Anda harus memberikan peran IAM ARN dengan izin untuk membaca objek Amazon S3 yang diberikan pada antarmuka pengguna Anda diLangkah 5.

   Anda hanya dapat melihat pratinjau template Anda jika Anda membuat template untuk Amazon Textract Texact atau Amazon Rekognition.

   PilihLihat pratinjauuntuk melihat pratinjau antarmuka dan instruksi yang dilihat pekerja. Ini adalah pratinjau interaktif. Setelah Anda menyelesaikan tugas sampel dan memilihKirim, Anda melihat output yang dihasilkan dari tugas yang baru saja Anda lakukan.

   Jika Anda membuat template tugas pekerja untuk jenis tugas khusus, Anda dapat melihat pratinjau UI tugas pekerja Anda menggunakan`RenderUiTemplate`. Untuk informasi selengkapnya, lihat Pratinjau Template Tugas Pekerja.
9. Saat Anda puas dengan templat Anda, pilihBuat.

Setelah membuat template, Anda dapat memilih template tersebut saat membuat alur kerja tinjauan manusia di konsol. Template Anda juga muncul diAmazon Augmented AIbagian konsol SageMaker di bawahPekerja tugas template. Pilih template Anda untuk melihat ARN nya. Gunakan ARN ini saat menggunakan`CreateFlowDefinition`Operasi API.

Membuat template tugas pekerja menggunakan template tugas pekerja (API)

Untuk menghasilkan template tugas pekerja menggunakan operasi SageMaker API`CreateHumanTaskUi`, tentukan nama untuk UI Anda`HumanTaskUiName`edan masukan template HTML Anda di`Content`di bawah`UiTemplate`. Temukan dokumentasi pada SDK khusus bahasa yang mendukung operasi API ini diLihat Jugabagian`CreateHumanTaskUi`.

Menghapus Template Tugas Pekerja

Setelah Anda membuat template tugas pekerja, Anda dapat menghapusnya menggunakan konsol SageMaker atau operasi SageMaker API`DeleteHumanTaskUi`.

Ketika Anda menghapus template tugas pekerja, Anda tidak dapat menggunakan alur kerja tinjauan manusia (definisi aliran) yang dibuat menggunakan template tersebut untuk memulai loop manusia. Setiap loop manusia yang telah dibuat menggunakan template tugas pekerja yang Anda hapus terus diproses sampai selesai dan tidak terpengaruh.

Menghapus templat tugas pekerja (konsol)

1. Buka konsol Amazon A2Ihttps://console.aws.amazon.com/a2i/.
2. Di bagian Augmented AI Amazon di panel navigasi sebelah kiri, pilihPekerja tugas template.
3. Pilih templat yang ingin Anda hapus.
4. SeleteHapus.
5. Modal muncul untuk mengkonfirmasi pilihan Anda. Pilih Hapus.

Menghapus template tugas pekerja (API)

Untuk menghapus template tugas pekerja menggunakan operasi SageMaker APIDeleteHumanTaskUi, tentukan nama UI Anda diHumanTaskUiName.

Buat Templat Tugas Pekerja Kustom

Elemen HTMLadalah komponen web yang menyediakan sejumlah widget tugas dan elemen desain yang dapat Anda sesuaikan dengan pertanyaan yang ingin Anda tanyakan. Anda dapat menggunakan elemen kerumunan ini untuk membuat template pekerja khusus dan mengintegrasikannya dengan alur kerja peninjauan manusia Amazon Augmented AI (Amazon A2I) untuk menyesuaikan konsol dan instruksi pekerja.

Untuk daftar semua elemen kerumunan HTML yang tersedia untuk pengguna Amazon A2I, lihatCrowd HTML Elemen Referensi. Untuk contoh template, lihatAWSRepositori GitHub, yang berisi lebih dari 60 contoh tugas kustom template.

Mengembangkan Template Secara Lokal

Ketika di konsol untuk menguji bagaimana template Anda memproses data yang masuk, Anda dapat menguji tampilan dan nuansa HTML template Anda dan elemen kustom di browser Anda dengan menambahkan kode berikut ke bagian atas file HTML Anda.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
```

Ini memuat kode yang diperlukan untuk membuat elemen HTML kustom. Gunakan kode ini jika Anda ingin mengembangkan tampilan dan nuansa template Anda di editor pilihan Anda, bukan di konsol.

Kode ini tidak akan mengurai variabel Anda. Anda mungkin ingin menggantinya dengan konten sampel sambil mengembangkan secara lokal.

Menggunakan Aset Eksternal

Templat kustom Amazon Augmented AI memungkinkan Anda untuk menyematkan skrip eksternal dan style sheet. Misalnya, header berikut menyematkan`text/css`nama style sheet`stylesheet`terletak di`https://www.example.com/my-enhancement-styles.css`ke dalam template kustom.

Example

```
<script src="https://www.example.com/my-enhancment-script.js"></script>
<link rel="stylesheet" type="text/css" href="https://www.example.com/my-enhancement-
styles.css">
```

Jika Anda mengalami kesalahan, pastikan server asal Anda mengirimkan jenis MIME yang benar dan header pengkodean dengan aset.

Misalnya, MIME dan jenis pengkodean untuk skrip jarak jauh adalah`application/javascript;CHARSET=UTF-8`.

Jenis MIME dan encoding untuk stylesheet jarak jauh adalah`text/css;CHARSET=UTF-8`.

Melacak Variabel Anda

Ketika membuat template kustom, Anda harus menambahkan variabel untuk itu untuk mewakili potongan-potongan data yang mungkin berubah dari tugas ke tugas, atau pekerja ke pekerja. Jika Anda memulai dengan salah satu contoh template, Anda perlu memastikan bahwa Anda menyadari variabel yang sudah digunakan.

Misalnya, untuk template kustom yang mengintegrasikan loop tinjauan manusia Augmented AI dengan tugas peninjauan teks Amazon Texact,`{{ task.input.selectedAiServiceResponse.blocks }}`digunakan untuk inisial-nilai input data. Untuk integrasi Amazon Augmented AI (Amazon A2I) dengan Amazon Rekognition,`{{ task.input.selectedAiServiceResponse.moderationLabels }}`digunakan. Untuk jenis tugas khusus, Anda perlu menentukan parameter input untuk jenis tugas Anda. Gunakan`{{ task.input.`*`customInputValuesForStartHumanLoop`*`}}`di mana Anda menentukan*`customInputValuesForStartHumanLoop`*.

Contoh kustom untuk Amazon Textract

Semua template kustom dimulai dan diakhiri dengan`<crowd-form> </crowd-form>`elemen. Seperti HTML standar`<form>`elemen, semua kode formulir Anda harus pergi antara elemen-elemen ini.

Untuk tugas analisis dokumen Amazon Textract, gunakan`<crowd-textract-analyze-document>`elemen. Menggunakan atribut berikut:

- `src`— Menentukan URL dari file gambar yang akan dijelaskan.
- `initialValue`— Menetapkan nilai awal untuk atribut yang ditemukan di UI pekerja.
- `blockTypes`(diperlukan) - Menentukan jenis analisis yang dapat dilakukan pekerja. HANYA`KEY_VALUE_SET`saat ini didukung.
- `keys`(diperlukan) - Menentukan kunci baru dan nilai teks terkait bahwa pekerja dapat menambahkan.
- `no-key-edit`(diperlukan) - Mencegah para pekerja mengedit kunci anotasi yang dilewati`initialValue`.
- `no-geometry-edit`— Mencegah pekerja mengedit poligon anotasi yang dilewati`initialValue`.

Untuk anak-anak dari`<crowd-textract-analyze-document>`elemen, Anda harus memiliki dua Daerah. Anda dapat menggunakan elemen HTML dan CSS yang sewenang-wenang di Wilayah ini.

- `<full-instructions>`— Instruksi yang tersedia dariPetunjuk lengkaplink di alat. Anda dapat membiarkan ini kosong, tetapi kami menyarankan Anda memberikan instruksi lengkap untuk mendapatkan hasil yang lebih baik.
- `<short-instructions>`— Penjelasan singkat tentang tugas yang muncul di bilah sisi alat. Anda dapat membiarkan ini kosong, tetapi kami menyarankan Anda memberikan instruksi lengkap untuk mendapatkan hasil yang lebih baik.

Template Amazon Textract akan terlihat serupa dengan yang berikut ini.

Example

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
```

```
{% capture s3_uri %}http://s3.amazonaws.com/
{{ task.input.aiServiceRequest.document.s3Object.bucket }}/
{{ task.input.aiServiceRequest.document.s3Object.name }}{% endcapture %}

<crowd-form>
  <crowd-textract-analyze-document
    src="{{ s3_uri | grant_read_access }}"
    initial-value="{{ task.input.selectedAiServiceResponse.blocks }}"
    header="Review the key-value pairs listed on the right and correct them if they
 don't match the following document."
    no-key-edit
    no-geometry-edit
    keys="{{ task.input.humanLoopContext.importantFormKeys }}"
    block-types="['KEY_VALUE_SET']"
  >
    <short-instructions header="Instructions">
      <style>
        .instructions {
          white-space: pre-wrap;
        }
        .instructionsImage {
          display: inline-block;
          max-width: 100%;
        }
      </style>
      <p class='instructions'>Choose a key-value block to highlight the corresponding
 key-value pair in the document.

If it is a valid key-value pair, review the content for the value. If the content is
 incorrect, correct it.

The text of the value is incorrect, correct it.
<img class='instructionsImage' src="https://example-site/correct-value-text.png" />

A wrong value is identified, correct it.
<img class='instructionsImage' src="https://example-site/correct-value.png" />

If it is not a valid key-value relationship, choose No.
<img class='instructionsImage' src="https://example-site/not-a-key-value-pair.png" />

If you can’t find the key in the document, choose Key not found.
<img class='instructionsImage' src="https://example-site/key-is-not-found.png" />

If the content of a field is empty, choose Value is blank.
```

```
<img class='instructionsImage' src="https://example-site/value-is-blank.png" />

<b>Examples</b>
Key and value are often displayed next to or below to each other.

Key and value displayed in one line.
<img class='instructionsImage' src="https://example-site/sample-key-value-pair-1.png" /
>

Key and value displayed in two lines.
<img class='instructionsImage' src="https://example-site/sample-key-value-pair-2.png" /
>

If the content of the value has multiple lines, enter all the text without a line
 break. Include all value text even if it extends beyond the highlight box.
<img class='instructionsImage' src="https://assets.crowd.aws/images/a2i-console/
multiple-lines.png" /></p>
    </short-instructions>

    <full-instructions header="Instructions"></full-instructions>
  </crowd-textract-analyze-document>
</crowd-form>
```

Contoh kustom untuk Amazon Rekognition

Semua template kustom dimulai dan diakhiri dengan`<crowd-form> </crowd-form>`elemen. Seperti HTML standar`<form>`elemen, semua kode formulir Anda harus pergi antara elemen-elemen ini. Untuk template tugas kustom Amazon Rekognition, gunakan`<crowd-rekognition-detect-moderation-labels>`elemen. Elemen ini mendukung atribut berikut:

- `categories`— Array dari stringataususunan objek di mana setiap objek memilikinameBidang.
  - Jika kategori masuk sebagai objek, hal berikut berlaku:
    - Kategori yang ditampilkan adalah nilainameBidang.
    - Jawaban yang dikembalikan berisipenuhobjek dari setiap kategori yang dipilih.
  - Jika kategori masuk sebagai string, hal berikut berlaku:
    - Jawaban yang dikembalikan adalah array dari semua string yang dipilih.
- `exclusion-category`— Dengan menetapkan atribut ini, Anda membuat tombol di bawah kategori di UI. Ketika pengguna memilih tombol, semua kategori tidak dipilih dan dinonaktifkan. Jika pekerja memilih tombol lagi, Anda mengaktifkan kembali pengguna untuk memilih kategori.

Jika pekerja menyerahkan tugas dengan memilihKirimsetelah Anda memilih tombol, tugas itu mengembalikan array kosong.

Untuk anak-anak dari`<crowd-rekognition-detect-moderation-labels>`elemen, Anda harus memiliki dua Daerah.

- `<full-instructions>`— Instruksi yang tersedia dariPetunjuk lengkaplink di alat. Anda dapat membiarkan ini kosong, tetapi kami menyarankan Anda memberikan instruksi lengkap untuk mendapatkan hasil yang lebih baik.
- `<short-instructions>`— Deskripsi singkat tentang tugas yang muncul di sidebar alat ini. Anda dapat membiarkan ini kosong, tetapi kami menyarankan Anda memberikan instruksi lengkap untuk mendapatkan hasil yang lebih baik.

Template menggunakan elemen ini akan terlihat serupa dengan yang berikut ini.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
{% capture s3_uri %}http://s3.amazonaws.com/
{{ task.input.aiServiceRequest.image.s3Object.bucket }}/
{{ task.input.aiServiceRequest.image.s3Object.name }}{% endcapture %}

<crowd-form>
  <crowd-rekognition-detect-moderation-labels
    categories='[
      {% for label in task.input.selectedAiServiceResponse.moderationLabels %}
        {
          name: "{{ label.name }}",
          parentName: "{{ label.parentName }}",
        },
      {% endfor %}
    ]'
    src="{{ s3_uri | grant_read_access }}"
    header="Review the image and choose all applicable categories."
  >
    <short-instructions header="Instructions">
      <style>
        .instructions {
          white-space: pre-wrap;
        }
      </style>
      <p class='instructions'>Review the image and choose all applicable categories.
```

```
If no categories apply, choose None.

<b>Nudity</b>
Visuals depicting nude male or female person or persons

<b>Graphic Male Nudity</b>
Visuals depicting full frontal male nudity, often close ups

<b>Graphic Female Nudity</b>
Visuals depicting full frontal female nudity, often close ups

<b>Sexual Activity</b>
Visuals depicting various types of explicit sexual activities and pornography

<b>Illustrated Nudity or Sexual Activity</b>
Visuals depicting animated or drawn sexual activity, nudity, or pornography

<b>Adult Toys</b>
Visuals depicting adult toys, often in a marketing context

<b>Female Swimwear or Underwear</b>
Visuals depicting female person wearing only swimwear or underwear

<b>Male Swimwear Or Underwear</b>
Visuals depicting male person wearing only swimwear or underwear

<b>Partial Nudity</b>
Visuals depicting covered up nudity, for example using hands or pose

<b>Revealing Clothes</b>
Visuals depicting revealing clothes and poses, such as deep cut dresses

<b>Graphic Violence or Gore</b>
Visuals depicting prominent blood or bloody injuries

<b>Physical Violence</b>
Visuals depicting violent physical assault, such as kicking or punching

<b>Weapon Violence</b>
Visuals depicting violence using weapons like firearms or blades, such as shooting

<b>Weapons</b>
Visuals depicting weapons like firearms and blades
```

```
<b>Self Injury</b>
Visuals depicting self-inflicted cutting on the body, typically in distinctive patterns
 using sharp objects

<b>Emaciated Bodies</b>
Visuals depicting extremely malnourished human bodies

<b>Corpses</b>
Visuals depicting human dead bodies

<b>Hanging</b>
Visuals depicting death by hanging</p>
    </short-instructions>

    <full-instructions header="Instructions"></full-instructions>
  </crowd-rekognition-detect-moderation-labels>
</crowd-form>
```

Tambahkan Otomatisasi dengan Cairan

Sistem template kustom menggunakanCairanuntuk otomatisasi. Cairanadalah bahasa markup inline sumber terbuka. Untuk informasi lebih lanjut dan dokumentasi, lihatHomepage cair.

Dalam Liquid, teks antara kurung kurawal tunggal dan simbol persen adalah instruksi ataumenandaiyang melakukan operasi seperti aliran kontrol atau iterasi. Teks antara kurung kurawal ganda adalah variabel atauobjekyang output nilainya. Daftar berikut mencakup dua jenis tag cair yang mungkin berguna untuk mengotomatisasi pemrosesan data input template. Jika Anda memilih salah satu jenis tag berikut, Anda akan diarahkan ke dokumentasi Liquid.

- Alur kontrol: Termasuk operator logika pemrograman seperti`if/else`,`unless`, dan`case/when`.
- Iterasi: Memungkinkan Anda untuk menjalankan blok kode berulang kali menggunakan pernyataan seperti untuk loop.

  Misalnya, contoh kode berikut menunjukkan bagaimana Anda dapat menggunakan Liquid`for`tag untuk membuat`for`Loop. Contoh ini loop melalui`moderationLabels`dikembalikan dari Amazon Rekognition dan menampilkan`moderationLabels`atribut`name`dan`parentName`bagi pekerja untuk meninjau:

  ```
  {% for label in task.input.selectedAiServiceResponse.moderationLabels %}
     {
       name: &amp;quot;{{ label.name }}&amp;quot;,
  ```

```
      parentName: &amp;quot;{{ label.parentName }}&amp;quot;,
    },
{% endfor %}
```

Gunakan Filter Variabel

Selain standarFilter cairdan tindakan, Amazon Augmented AI (Amazon A2I) menawarkan filter tambahan. Anda menerapkan filter dengan menempatkan pipa (`|`) karakter setelah nama variabel, dan kemudian menentukan nama filter. Untuk filter rantai, gunakan format berikut.

Example

```
{{ <content> | <filter> | <filter> }}
```

Autoescape dan Eksplisit Luput

Secara default, input adalah HTML-lolos untuk mencegah kebingungan antara teks variabel dan HTML. Anda dapat secara eksplisit menambahkan`escape`filter untuk membuatnya lebih jelas bagi seseorang yang membaca sumber template Anda yang melarikan diri sedang dilakukan.

escape_once

`escape_once`memastikan bahwa jika Anda sudah lolos dari kode Anda, itu tidak mendapatkan kembali melarikan diri lagi. Sebagai contoh, memastikan bahwa`&`tidak menjadi`&`.

skip_autoescape

`skip_autoescape`berguna ketika konten Anda dimaksudkan untuk digunakan sebagai HTML. Misalnya, Anda mungkin memiliki beberapa paragraf teks dan beberapa gambar dalam instruksi lengkap untuk kotak pembatas.

> **Note**
>
> Gunakan`skip_autoescape`hemat. Sebagai praktik terbaik untuk template, hindari melewati kode fungsional atau markup dengan`skip_autoescape`kecuali Anda benar-benar yakin bahwa Anda memiliki kontrol ketat atas apa yang sedang berlalu. Jika Anda melewati masukan pengguna, Anda bisa membuka pekerja Anda hingga serangan skrip lintas situs.

to_json

`to_json`mengkodekan data yang Anda berikan ke JavaScript Object Notation (JSON). Jika Anda memberikan objek, itu serializes itu.

grant_read_access

`grant_read_access`URI Amazon Simple Storage Service (Amazon S3) dan mengkodekan ke URL HTTPS dengan token akses jangka pendek untuk sumber daya itu. Hal ini memungkinkan untuk menampilkan foto, audio, atau objek video yang disimpan dalam ember S3 yang tidak dapat diakses publik oleh pekerja.

Example Contoh filter to_json dan grant_read_access

Input

```
auto-escape: {{ "Have you read 'James & the Giant Peach'?" }}
explicit escape: {{ "Have you read 'James & the Giant Peach'?" | escape }}
explicit escape_once: {{ "Have you read 'James &amp; the Giant Peach'?" |
 escape_once }}
skip_autoescape: {{ "Have you read 'James & the Giant Peach'?" | skip_autoescape }}
to_json: {{ jsObject | to_json }}
grant_read_access: {{ "s3://examplebucket/myphoto.png" | grant_read_access }}
```

Example

Output

```
auto-escape: Have you read &#39;James &amp; the Giant Peach&#39;?
explicit escape: Have you read &#39;James &amp; the Giant Peach&#39;?
explicit escape_once: Have you read &#39;James &amp; the Giant Peach&#39;?
skip_autoescape: Have you read 'James & the Giant Peach'?
to_json: { "point_number": 8, "coords": [ 59, 76 ] }
grant_read_access: https://s3.amazonaws.com/examplebucket/myphoto.png?<access token and
 other params>
```

Example Contoh template klasifikasi otomatis.

Untuk mengotomatisasi sampel klasifikasi teks sederhana ini, sertakan tag Liquid`{{ task.input.source }}`. Contoh ini menggunakanklasifikasi kerumunanelemen.

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
```

```
<crowd-form>
  <crowd-classifier
    name="tweetFeeling"
    categories="['positive', 'negative', 'neutral', 'cannot determine']"
    header="Which term best describes this tweet?"
  >
    <classification-target>
       {{ task.input.source }}
    </classification-target>

    <full-instructions header="Analyzing a sentiment">
      Try to determine the feeling the author
      of the tweet is trying to express.
      If none seems to match, choose "other."
    </full-instructions>

    <short-instructions>
      Pick the term that best describes the sentiment
      of the tweet.
    </short-instructions>

  </crowd-classifier>
</crowd-form>
```

Pratinjau Template Tugas Pekerja

Untuk melihat pratinjau template tugas pekerja khusus, gunakan SageMaker`RenderUiTemplate`operasi. Anda dapat menggunakan`RenderUiTemplate`operasi dengan AWS CLI atau pilihan Anda AWS SDK. Untuk dokumentasi SDK khusus bahasa yang didukung untuk operasi API ini, lihat`See Also`bagian`RenderUiTemplate`.

Prasyarat

Untuk melihat pratinjau template tugas pekerja Anda, AWS Identity and Access Management (IAM) role Amazon Resource Name (ARN), atau`RoleArn`, yang Anda gunakan harus memiliki izin untuk mengakses ke objek S3 yang digunakan oleh template. Untuk mempelajari cara mengonfigurasi peran atau pengguna Aktifkan Pratinjau Template Tugas Pekerja .

Untuk melihat pratinjau template tugas pekerja Anda menggunakan **`RenderUiTemplate`** Operasi:

1. Berikan **`RoleArn`** dari peran dengan kebijakan yang diperlukan yang dilampirkan untuk melihat pratinjau template kustom Anda.

2. Di**`Input`**parameter**`Task`**, menyediakan objek JSON yang berisi nilai-nilai untuk variabel didefinisikan dalam template. Ini adalah variabel yang diganti untuk`task.input.`*`source`*variabel. Misalnya, jika Anda mendefinisikan variabel task.input.text dalam template Anda, Anda dapat menyediakan variabel dalam objek JSON sebagai`text:sample text`.
3. Di**`Content`**parameter**`UiTemplate`**, masukkan template Anda.

Setelah Anda mengonfigurasi`RenderUiTemplate`, gunakan SDK pilihan Anda atauAWS CLIuntuk mengirimkan permintaan untuk membuat template Anda. Jika permintaan Anda berhasil, tanggapannya termasuk`RenderedContent`, template Liquid yang membuat HTML untuk UI pekerja.

> **Important**
>
> Untuk melihat pratinjau template, Anda memerlukan peran IAM dengan izin untuk membaca objek Amazon S3 yang diberikan pada antarmuka pengguna Anda. Untuk contoh kebijakan yang dapat Anda lampirkan ke peran IAM Anda untuk memberikan izin ini, lihatAktifkan Pratinjau Template Tugas Pekerja .

Membuat Instruksi Pekerja yang Baik

Membuat instruksi yang baik untuk pekerjaan peninjauan manusia Anda meningkatkan akurasi pekerja Anda dalam menyelesaikan tugas mereka. Anda dapat memodifikasi instruksi default yang disediakan di konsol saat membuat alur kerja tinjauan manusia, atau Anda dapat menggunakan konsol untuk membuat template pekerja khusus dan menyertakan instruksi Anda dalam template ini. Instruksi ditampilkan kepada pekerja di halaman UI tempat mereka menyelesaikan tugas pelabelan mereka.

Buat Instruksi Good Worker

Ada tiga jenis instruksi di konsol Amazon Augmented AI:

- Deskripsi tugas— Deskripsi harus memberikan penjelasan ringkas tentang tugas.
- Petunjuk— Instruksi ini ditampilkan pada halaman web yang sama di mana pekerja menyelesaikan tugas. Instruksi ini harus memberikan referensi yang mudah untuk menunjukkan kepada pekerja cara yang benar untuk menyelesaikan tugas.
- Petunjuk tambahan- Instruksi ini ditampilkan dalam kotak dialog yang muncul ketika seorang pekerja memilihLihat instruksi lengkap. Kami menyarankan Anda memberikan petunjuk rinci untuk

menyelesaikan tugas, dan menyertakan beberapa contoh yang menunjukkan kasus tepi dan situasi sulit lainnya untuk pelabelan objek.

Tambahkan Contoh Gambar ke Petunjuk Anda

Gambar memberikan contoh yang berguna bagi pekerja Anda. Untuk menambahkan gambar yang dapat diakses publik ke instruksi Anda, lakukan hal berikut:

1. Tempatkan kursor tempat gambar harus masuk ke editor petunjuk.
2. Pilih ikon gambar di toolbar editor.
3. Masukkan URL gambar Anda.

Jika gambar instruksi Anda berada dalam bucket S3 yang tidak dapat diakses publik, lakukan hal berikut:

- Untuk URL gambar, masukkan:`{{ 'https://s3.amazonaws.com/your-bucket-name/image-file-name' | grant_read_access }}`.

Ini membuat URL gambar dengan kode akses satu kali berumur pendek yang ditambahkan sehingga browser pekerja dapat menampilkannya. Ikon gambar yang rusak ditampilkan di editor petunjuk, tetapi melihat pratinjau alat menampilkan gambar di pratinjau yang diberikan. Lihatgrant_read_accessuntuk informasi lebih lanjut tentang`grand_read_access`elemen.

Pantau dan Kelola Loop Manusia Anda

Setelah Anda memulai loop tinjauan manusia, Anda dapat memeriksa hasil tugas yang dikirim ke loop dan mengelolanya menggunakanAPI Amazon Augmented AI. Selain itu, Amazon A2I terintegrasi dengan Amazon EventBridge (juga dikenal sebagai Amazon CloudWatch Events) untuk mengingatkan Anda ketika status loop tinjauan manusia berubah menjadi`Completed`,`Failed`, atau`Stopped`. Pengiriman acara ini dijamin setidaknya sekali, yang berarti semua peristiwa yang dibuat ketika loop manusia selesai berhasil dikirim ke EventBridge.

Gunakan prosedur di bawah ini untuk mempelajari cara menggunakan Amazon A2I Runtime API untuk memantau dan mengelola loop manusia Anda. LihatGunakanAmazon CloudWatch EventsAmazon Augmented Aluntuk mempelajari bagaimana Amazon A2I terintegrasi dengan Amazon EventBridge.

Untuk memeriksa data output Anda:

1. Periksa hasil loop manusia Anda dengan memanggilDescribeHumanLoopoperasi. Hasil operasi API ini berisi informasi tentang alasan dan hasil aktivasi loop.
2. Periksa data output dari loop manusia Anda di Amazon Simple Storage Service (Amazon S3). Di jalur ke data,*YYYY*/*MM*/*DD*/*hh*/*mm*/*ss*mewakili tanggal penciptaan loop manusia dengan tahun (YYYY), bulan (MM), dan hari (DD), dan waktu penciptaan dengan jam (hh), menit (mm), dan kedua (ss).

   ```
   s3://customer-output-bucket-specified-in-flow-definition/flow-definition-name/YYYY/MM/DD/hh/mm/ss/human-loop-name/output.json
   ```

Anda dapat mengintegrasikan struktur ini denganAWS Glueatau Amazon Athena untuk partisi dan menganalisis data output Anda. Untuk informasi selengkapnya, lihatMengelola Partisi untuk Output ETL diAWSGlue.

Untuk mempelajari selengkapnya tentang format data keluaran Amazon A2I, lihatData Output Amazon A2I.

Untuk menghentikan dan menghapus loop manusia Anda:

1. Setelah loop manusia telah dimulai, Anda dapat menghentikan loop manusia Anda dengan memanggilStopHumanLoopoperasi menggunakanHumanLoopName. Jika loop manusia berhasil dihentikan, server mengirimkan kembali respons HTTP 200.
2. Untuk menghapus loop manusia yang statusnya samaFailed,Completed, atauStopped, gunakanDeleteHumanLoopoperasi.

Untuk daftar loop manusia:

1. Anda dapat daftar semua loop manusia aktif dengan memanggilListHumanLoopsoperasi. Anda dapat menyaring loop manusia dengan tanggal pembuatan loop menggunakanCreationTimeAfterdanCreateTimeBeforeparameter.
2. Jika berhasil,ListHumanLoopspulangHumanLoopSummariesdanNextTokenbenda dalam elemen respon.HumanLoopSummariesberisi informasi tentang loop manusia tunggal. Misalnya,daftarSstatus loop dan,jika berlaku,-nyaalasan kegagalan.

Gunakan string yang dikembalikan`NextToken`sebagai masukan dalam panggilan berikutnya untuk`ListHumanLoops`untuk melihat halaman berikutnya dari loop manusia.

Data Output Amazon A2I

Ketika alur kerja machine learning Anda mengirimkan Amazon A2I objek data,lingkaran manusiadiciptakan dan pengulas manusia menerimatugasuntuk meninjau bahwa objek data. Data output dari setiap tugas peninjauan manual disimpan di bucket output Amazon Simple Storage Service (Amazon S3) yang Anda tetapkan di alur kerja peninjauan manual. Di jalur ke data,*YYYY*/*MM*/*DD*/*hh*/*mm*/*ss*mewakili tanggal penciptaan loop manusia dengan tahun (YYYY), bulan (MM), dan hari (DD), dan waktu penciptaan dengan jam (hh), menit (mm), dan kedua (ss).

```
s3://customer-output-bucket-specified-in-flow-definition/flow-definition-
name/YYYY/MM/DD/hh/mm/ss/human-loop-name/output.json
```

Konten data output Anda tergantung pada jenistipe tugas(built-in atau custom) dan jenistenaga kerjaAnda menggunakan. Data output Anda selalu menyertakan respons dari pekerja manusia. Selain itu, data output mungkin termasuk metadata tentang loop manusia, reviewer manusia (pekerja), dan objek data.

Gunakan bagian berikut untuk mempelajari lebih lanjut tentang format data output Amazon A2I untuk berbagai jenis tugas dan tenaga kerja.

Output Data Dari Built-In Jenis Tugas

Jenis tugas bawaan Amazon A2I mencakup Amazon Textract dan Amazon Rekognition. Selain respon manusia, data output dari salah satu tugas ini mencakup rincian tentang alasan loop manusia diciptakan dan informasi tentang layanan terintegrasi yang digunakan untuk membuat loop manusia. Gunakan tabel berikut untuk mempelajari lebih lanjut tentang skema data output untuk semua jenis tugas built-in. Parameternilaiuntuk masing-masing parameter ini tergantung pada layanan yang Anda gunakan dengan Amazon A2I. Lihat tabel kedua di bagian ini untuk informasi lebih lanjut tentang nilai-nilai khusus layanan ini.

Parameter	Jenis Nilai	Nilai contoh	Deskripsi
`awsManagedHumanLoopRequestSource`	String	`AWS/Rekognition/DetectModerationLabels/Image/V3` atau `AWS/Textract/AnalyzeDocument/Forms/V1`	Operasi API dan terkaitAWSlayanan yang meminta Amazon A2I membuat loop manusia. Ini adalah operasi API yang Anda gunakan untuk mengonfigurasi loop manusia Amazon A2I Anda.
`flowDefinitionArn`	String	`arn:aws:sagemaker:us-west-2:` *`111122223333`* `:flow-definition/` *`flow-definition-name`*	Amazon Resource Number (ARN) dari alur kerja peninjauan manual (definisi aliran) yang digunakan untuk membuat lingkaran manusia.

`humanAnswers`	Daftar objek JSON	`{ "answerContent": { "AWS/Rekognition/DetectModerationLabels/Image/V3": { "moderationLabels": [...] } }, }`
		or
		`{`
			Daftar objek JSON yang berisi tanggapan pekerja`answerContent` .
			Objek ini juga berisi rincian pengiriman dan, jika tenaga kerja swasta digunakan, metadata pekerja. Untuk mempelajari selengkapnya, lihat Aktivitas Pekerja Track.

Parameter	Jenis Nilai	Nilai contoh	Deskripsi
		"answerContent": { "AWS/Textract/AnalyzeDocument/Forms/V1": { "blocks": [...] } },	Untuk data keluaran loop manusia yang dihasilkan dari Amazon Rekogniti onDetectModerationLabel meninjau tugas, parameter ini hanya berisi tanggapan positif. Misalnya, jika pekerja memilihTidak ada konten, respon ini tidak termasuk.
humanLoopName	String	'human-loop-name'	Nama lingkaran manusia.
inputContent	Objek JSON	{ "aiServiceRequest": {...}, "aiServiceResponse": {...}, "humanTaskActivationConditionResults": {...}, "selectedAiServiceResponse": {...} }	Konten inputAWSI ayanan dikirim ke Amazon A2I ketika diminta loop manusia dibuat.

Parameter	Jenis Nilai	Nilai contoh	Deskripsi

`aiServiceRequest`	Objek JSON	`{ "document": {...}, "featureTypes": [ ...], "humanLoopConfig": { ...} }`
		or
		`{ "image": {...}, "humanLoopConfig": { ...} }`
			Permintaan asli dikirim keAWSLayanan terintegrasi dengan Amazon A2I. Misalnya, jika Anda menggunakan Amazon Rekognition dengan Amazon A2I, ini termasuk permintaan yang dibuat melalui operasi API`DetectModerationLabels` . Untuk integrasi Amazon Textract, ini termasuk permintaan yang dibuat melalui`AnalyzeDocument` .

Parameter	Jenis Nilai	Nilai contoh	Deskripsi

`aiService Response`	Objek JSON	`{ "moderationLabels": [...], "moderationModelVersion": "3.0" }`
		or
		`{ "blocks": [...], "documentMetadata": {} }`
			Respon penuh dariAWSlayanan. Ini adalah data yang digunakan untuk menentukan apakah tinjauan manusia diperlukan. Objek ini mungkin berisi metadata tentang objek data yang tidak dibagi dengan pengulas manusia.

Parameter	Jenis Nilai	Nilai contoh	Deskripsi

`selectedAiServiceResponse`	Objek JSON	`{ "moderationLabels": [...], "moderationModelVersion": "3.0" }`
		
		or
		
		`{ "blocks": [...], "documentMetadata": {} }`
			Bagian dari`aiServiceResponse` yang sesuai dengan kondisi aktivasi di`ActivationConditions` .
			
			Semua objek data yang tercantum dalam`aiServiceResponse` tercantum dalam`selectedAiServiceResponse` ketika kesimpulan diambil sampel secara acak, atau semua kesimpulan dimulai kondisi aktivasi.

Parameter	Jenis Nilai	Nilai contoh	Deskripsi
`humanTaskActivationConditionResults`	Objek JSON	`{ "Conditions": [ ...] }`	Objek JSON di`inputContent` yang berisi alasan loop manusia diciptakan. Ini termasuk daftar kondisi aktivasi (`Conditions`) termasuk dalam alur kerja tinjauan manusia Anda (definisi aliran), dan hasil evaluasi untuk setiap kondisi—hasil ini baik`true`atau`false`. Untuk mempelajari selengkapnya tentang kondisi aktivasi, lihatSkema JSON untuk Kondisi Aktivasi Loop Manusia di Amazon Augmented AI.

Pilih tab pada tabel berikut untuk mempelajari tentang jenis tugas—parameter spesifik dan lihat contoh blok kode output-data untuk masing-masing jenis tugas bawaan.

Amazon Textract Task Type Output Data

> Saat Anda menggunakan integrasi bawaan Amazon Textract Texact, Anda akan melihat`'AWS/Textract/AnalyzeDocument/Forms/V1'`sebagai nilai untuk`awsManagedHumanLoopRequestSource`data output Anda.

Parameter`answerContent`parameter berisi`Block`objek yang mencakup respons manusia untuk semua blok yang dikirim ke Amazon A2I.

Parameter`aiServiceResponse`parameter juga termasuk`Block`objek dengan respons Amazon TExtract terhadap permintaan asli yang dikirim menggunakan`AnalyzeDocument`.

Untuk mempelajari lebih lanjut tentang parameter yang Anda lihat di objek blok, lihatBlokirdiPanduan Developer Amazon Textract Texact.

Berikut ini adalah contoh data output dari tinjauan manusia Amazon A2I atas kesimpulan analisis dokumen Amazon Textract.

```
{
    "awsManagedHumanLoopRequestSource": "AWS/Textract/AnalyzeDocument/Forms/V1",
    "flowDefinitionArn": "arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
    "humanAnswers": [
        {
            "answerContent": {
                "AWS/Textract/AnalyzeDocument/Forms/V1": {
                    "blocks": [...]
                }
            },
            "submissionTime": "2020-09-28T19:17:59.880Z",
            "workerId": "111122223333",
            "workerMetadata": {
                "identityData": {
                    "identityProviderType": "Cognito",
                    "issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111",
                    "sub": "c6aa8eb7-9944-42e9-a6b9-111122223333"
                }
            }
        }
    ],
    "humanLoopName": "humnan-loop-name",
    "inputContent": {
        "aiServiceRequest": {
            "document": {
                "s3Object": {
                    "bucket": "DOC-EXAMPLE-BUCKET1",
                    "name": "document-demo.jpg"
                }
```

```
            },
            "featureTypes": [
                "TABLES",
                "FORMS"
            ],
            "humanLoopConfig": {
                "dataAttributes": {
                    "contentClassifiers": [
                        "FreeOfPersonallyIdentifiableInformation"
                    ]
                },
                "flowDefinitionArn": "arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
                "humanLoopName": "humnan-loop-name"
            }
        },
        "aiServiceResponse": {
            "blocks": [...],
            "documentMetadata": {
                "pages": 1
            }
        },
        "humanTaskActivationConditionResults": {
            "Conditions": [
                {
                    "EvaluationResult": true,
                    "Or": [
                        {
                            "ConditionParameters": {
                                "ImportantFormKey": "Mail address",
                                "ImportantFormKeyAliases": [
                                    "Mail Address:",
                                    "Mail address:",
                                    "Mailing Add:",
                                    "Mailing Addresses"
                                ],
                                "KeyValueBlockConfidenceLessThan": 100,
                                "WordBlockConfidenceLessThan": 100
                            },
                            "ConditionType": "ImportantFormKeyConfidenceCheck",
                            "EvaluationResult": true
                        },
                        {
                            "ConditionParameters": {
```

```
                                "ImportantFormKey": "Mail address",
                                "ImportantFormKeyAliases": [
                                    "Mail Address:",
                                    "Mail address:",
                                    "Mailing Add:",
                                    "Mailing Addresses"
                                ]
                            },
                            "ConditionType": "MissingImportantFormKey",
                            "EvaluationResult": false
                        }
                    ]
                }
            ]
        },
        "selectedAiServiceResponse": {
            "blocks": [...]
        }
    }
}
```

Amazon Rekognition Task Type Output Data

Saat Anda menggunakan integrasi bawaan Amazon Textract Texact, Anda akan melihat string `'AWS/Rekognition/DetectModerationLabels/Image/V3'` sebagai nilai untuk `awsManagedHumanLoopRequestSource` data output Anda.

Parameter `answerContent` parameter berisi `moderationLabels` objek yang berisi respons manusia untuk semua label moderasi yang dikirim ke Amazon A2I.

Parameter `aiServiceResponse` parameter juga termasuk `moderationLabels` keberatan dengan respons Amazon Rekognition terhadap permintaan asli yang dikirim ke `DetectModerationLabels`.

Untuk mempelajari lebih lanjut tentang parameter yang Anda lihat di objek blok, lihat ModerationLabel di Panduan Pengembang Amazon Rekognition.

Berikut ini adalah contoh data output dari tinjauan manusia Amazon A2I atas kesimpulan moderasi gambar Amazon Rekognition.

```
{
    "awsManagedHumanLoopRequestSource": "AWS/Rekognition/DetectModerationLabels/
Image/V3",
```

```
    "flowDefinitionArn": "arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
    "humanAnswers": [
        {
            "answerContent": {
                "AWS/Rekognition/DetectModerationLabels/Image/V3": {
                    "moderationLabels": [...]
                }
            },
            "submissionTime": "2020-09-28T19:22:35.508Z",
            "workerId": "ef7294f850a3d9d1",
            "workerMetadata": {
                "identityData": {
                    "identityProviderType": "Cognito",
                    "issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111",
                    "sub": "c6aa8eb7-9944-42e9-a6b9-111122223333"
                }
            }
        }
    ],
    "humanLoopName": "humnan-loop-name",
    "inputContent": {
        "aiServiceRequest": {
            "humanLoopConfig": {
                "flowDefinitionArn": "arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
                "humanLoopName": "humnan-loop-name"
            },
            "image": {
                "s3Object": {
                    "bucket": "DOC-EXAMPLE-BUCKET1",
                    "name": "example-image.jpg"
                }
            }
        },
        "aiServiceResponse": {
            "moderationLabels": [...],
            "moderationModelVersion": "3.0"
        },
        "humanTaskActivationConditionResults": {
            "Conditions": [
                {
                    "EvaluationResult": true,
```

```
                    "Or": [
                        {
                            "ConditionParameters": {
                                "ConfidenceLessThan": 98,
                                "ModerationLabelName": "Suggestive"
                            },
                            "ConditionType": "ModerationLabelConfidenceCheck",
                            "EvaluationResult": true
                        },
                        {
                            "ConditionParameters": {
                                "ConfidenceGreaterThan": 98,
                                "ModerationLabelName": "Female Swimwear Or
 Underwear"
                            },
                            "ConditionType": "ModerationLabelConfidenceCheck",
                            "EvaluationResult": false
                        }
                    ]
                }
            ]
        },
        "selectedAiServiceResponse": {
            "moderationLabels": [
                {
                    "confidence": 96.7122802734375,
                    "name": "Suggestive",
                    "parentName": ""
                }
            ],
            "moderationModelVersion": "3.0"
        }
    }
}
```

Output Data Dari Jenis Tugas Kustom

Saat Anda menambahkan Amazon A2I ke alur kerja peninjauan manusia kustom, Anda akan melihat parameter berikut dalam data output yang dikembalikan dari tugas peninjauan manusia.

Parameter	Jenis Nilai	Deskripsi
`flowDefinitionArn`	String	Amazon Resource Number (ARN) dari alur kerja peninjauan manual (definisi aliran) yang digunakan untuk membuat lingkaran manusia.
`humanAnswers`	Daftar objek JSON	Daftar objek JSON yang berisi tanggapan pekerja`answerContent`. Nilai dalam parameter ini ditentukan oleh output yang diterima dari AndaTemplat Tugas Pekerja. Jika Anda menggunakan tenaga kerja pribadi, metadata pekerja disertakan. Untuk mempelajari selengkapnya, lihat Aktivitas Pekerja Track.
`humanLoopName`	String	Nama lingkaran manusia.
`inputContent`	Objek JSON	Konten masukan yang dikirim ke Amazon A2I dalam permintaan untuk`StartHumanLoop`.

Berikut ini adalah contoh data output dari integrasi kustom dengan Amazon A2I dan Amazon Transcribe. Dalam contoh ini,`inputContent`terdiri dari:

- Path ke file.mp4 di Amazon S3 dan judul video
- Transkripsi yang dikembalikan dari Amazon Transcribe (diurai dari data output Amazon Transcribe)
- Waktu mulai dan akhir yang digunakan oleh template tugas pekerja untuk klip file.mp4 dan menunjukkan pekerja bagian yang relevan dari video

```
{
    "flowDefinitionArn": "arn:aws:sagemaker:us-west-2:111122223333:flow-
definition/flow-definition-name",
    "humanAnswers": [
        {
            "answerContent": {
                "transcription": "use lambda to turn your notebook"
            },
            "submissionTime": "2020-06-18T17:08:26.246Z",
            "workerId": "ef7294f850a3d9d1",
            "workerMetadata": {
                "identityData": {
                    "identityProviderType": "Cognito",
                    "issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111",
                    "sub": "c6aa8eb7-9944-42e9-a6b9-111122223333"
                }
            }

        }
    ],
    "humanLoopName": "human-loop-name",
    "inputContent": {
        "audioPath": "s3://DOC-EXAMPLE-BUCKET1/a2i_transcribe_demo/Fully-Managed
 Notebook Instances with Amazon SageMaker - a Deep Dive.mp4",
        "end_time": 950.27,
        "original_words": "but definitely use Lambda to turn your ",
        "start_time": 948.51,
        "video_title": "Fully-Managed Notebook Instances with Amazon SageMaker - a Deep
 Dive.mp4"
    }
}
```

Aktivitas Pekerja Track

Amazon A2I menyediakan informasi yang dapat Anda gunakan untuk melacak individu pekerja dalam data output tugas. Untuk mengidentifikasi pekerja yang mengerjakan tugas peninjauan manusia, gunakan yang berikut dari data output di Amazon S3:

- Parameter`acceptanceTime`adalah waktu bahwa pekerja menerima tugas. Format cap tanggal dan waktu ini`YYYY-MM-DDTHH:MM:SS.mmm`Zuntuk tahun (`YYYY`), bulan (`MM`), hari (`DD`), jam (`HH`), menit (`MM`), kedua (`SS`), dan milidetik (`mmm`). Tanggal dan waktu dipisahkan oleh`T`.

- Parameter`submissionTime`adalah waktu bahwa pekerja menyerahkan anotasi mereka menggunakan`Kirimtombol. Format cap tanggal dan waktu ini`YYYY-MM-DDTHH:MM:SS.mmm`Zuntuk tahun (`YYYY`), bulan (`MM`), hari (`DD`), jam (`HH`), menit (`MM`), kedua (`SS`), dan milidetik (`mmm`). Tanggal dan waktu dipisahkan oleh`T.
- `timeSpentInSeconds`melaporkan total waktu, dalam hitungan detik, bahwa seorang pekerja secara aktif mengerjakan tugas itu. Metrik ini tidak termasuk waktu ketika seorang pekerja berhenti atau beristirahat.
- Parameter`workerId`unik untuk setiap pekerja.
- Jika Anda menggunakan[tenaga kerja privat](), di`workerMetadata`, Anda akan melihat pesan berikut.
  - Parameter`identityProviderType`adalah layanan yang digunakan untuk mengelola tenaga kerja swasta.
  - Parameter`issuer`adalah kumpulan pengguna Amazon Cognito atau OpenID Connect (OIDC) Identity Provider (IdP) penerbit yang terkait dengan tim kerja yang ditugaskan untuk tugas peninjauan manusia ini.
  - Unik`sub`pengidentifikasi mengacu pada pekerja. Jika Anda membuat tenaga kerja menggunakan Amazon Cognito, Anda dapat mengambil rincian tentang pekerja ini (seperti nama atau nama pengguna) yang terkait dengan ID ini menggunakan Amazon Cognito. Untuk mempelajari caranya, lihat[Mengelola dan Mencari Akun Pengguna]()di[Panduan Developer Amazon Cognito]().

Berikut ini adalah contoh output yang mungkin Anda lihat jika Anda menggunakan Amazon Cognito untuk membuat tenaga kerja privat. Hal ini diidentifikasi dalam`identityProviderType`.

```
"submissionTime": "2020-12-28T18:59:58.321Z",
"acceptanceTime": "2020-12-28T18:59:15.191Z",
"timeSpentInSeconds": 40.543,
"workerId": "a12b3cdefg4h5i67",
"workerMetadata": {
    "identityData": {
        "identityProviderType": "Cognito",
        "issuer": "https://cognito-idp.aws-region.amazonaws.com/aws-region_123456789",
        "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
    }
}
```

Berikut ini adalah contoh output yang mungkin Anda lihat jika Anda menggunakan IDP OIDC Anda sendiri untuk membuat tenaga kerja privat:

```
"workerMetadata": {
        "identityData": {
            "identityProviderType": "Oidc",
            "issuer": "https://example-oidc-ipd.com/adfs",
            "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
        }
}
```

Untuk mempelajari selengkapnya tentang menggunakan tenaga kerja privat, lihatMenggunakan Tenaga Kerja Pribadi.

Izin dan Keamanan di Amazon Augmented AI

Saat menggunakan Amazon Augmented AI (Amazon A2I) untuk membuat alur kerja peninjauan manusia untuk aplikasi ML/AI Anda, Anda membuat dan mengonfigurasi sumber daya di Amazon SageMaker seperti tenaga kerja manusia dan templat tugas pekerja. Untuk mengonfigurasi dan memulai loop manusia, Anda mengintegrasikan Amazon A2I denganAWS layanan lain seperti Amazon Textract atau Amazon Rekognition, atau menggunakan Amazon Augmented AI Runtime API. Untuk membuat alur kerja tinjauan manusia dan memulai loop manusia, Anda harus melampirkan kebijakan tertentu ke peran atau penggunaAWS Identity and Access Management (IAM) Anda. Secara khusus:

- Saat memulai loop manusia menggunakan data input gambar pada atau setelah 12 Januari 2020, Anda harus menambahkan kebijakan header CORS ke bucket Amazon S3 yang berisi data input Anda. Lihat CORS untuk mempelajari selengkapnya.
- Saat membuat definisi alur, Anda perlu memberikan peran yang memberikan izin Amazon A2I untuk mengakses Amazon S3 baik untuk membaca objek yang dirender dalam UI tugas manusia dan untuk menulis hasil tinjauan manusia.

  Kebijakan kepercayaan ini juga harus memiliki kebijakan kepercayaan untuk memberikan SageMaker izin untuk mengambil peran. Hal ini memungkinkan Amazon A2I untuk melakukan tindakan sesuai dengan izin yang Anda lampirkan ke peran.

  LihatTambahkan Izin ke Peran IAM yang Digunakan untuk Membuat Definisi Aliran contoh kebijakan yang dapat Anda ubah dan lampirkan ke peran yang Anda gunakan untuk membuat definisi alur. Ini adalah kebijakan yang dilampirkan pada peran IAM yang dibuat di bagian Alur kerja tinjauan Manusia di area Amazon A2I SageMaker konsol.

- Untuk membuat dan memulai loop manusia, Anda menggunakan operasi API dari jenis tugas bawaan (sepertiDetectModerationLabel atauAnalyzeDocument) atau operasi Amazon A2I Runtime APIStartHumanLoop dalam aplikasi MS kustom. Anda perlu melampirkan kebijakanAmazonAugmentedAIFullAccess terkelola kepada pengguna yang memanggil operasi API ini untuk memberikan izin ke layanan ini untuk menggunakan operasi Amazon A2I. Untuk mempelajari caranya, lihat Membuat Pengguna yang Dapat Memanggil Operasi API Amazon A2I.

  Kebijakan ini tidak memberikan izin untuk menjalankan operasi APIAWS layanan yang terkait dengan jenis tugas bawaan. Misalnya,AmazonAugmentedAIFullAccess tidak memberikan izin untuk memanggil operasi Amazon RekognitionDetectModerationLabel API atau operasi Amazon TextractAnalyzeDocument API. Anda dapat menggunakan kebijakan yang lebih umumAmazonAugmentedAIIntegratedAPIAccess,, untuk memberikan izin ini. Untuk informasi selengkapnya, lihat Membuat Pengguna Dengan Izin untuk Memanggil Amazon A2I, Amazon Textract, dan Amazon Rekognition API Operations. Ini adalah opsi yang baik ketika Anda ingin memberikan izin luas kepada pengguna untuk menggunakan Amazon A2I dan operasiAWS API layanan terintegrasi.

  Jika Anda ingin mengonfigurasi izin yang lebih terperinci, lihat Contoh Kebijakan Berbasis Identitas Amazon Rekognition dan Contoh Kebijakan Berbasis Identitas Amazon Textract untuk kebijakan berbasis identitas yang dapat Anda gunakan untuk memberikan izin untuk menggunakan layanan individual ini.

- Untuk melihat pratinjau template UI tugas pekerja kustom, Anda memerlukan peran IAM dengan izin untuk membaca objek Amazon S3 yang dirender di antarmuka pengguna Anda. Lihat contoh kebijakan diAktifkan Pratinjau Template Tugas Pekerja .

Topik

CORS

Sebelumnya pada tahun 2020, browser yang banyak digunakan seperti Chrome dan Firefox mengubah perilaku default mereka untuk memutar gambar berdasarkan metadata gambar, disebut sebagai data EXIF. Sebelumnya, gambar akan selalu ditampilkan di browser persis bagaimana mereka disimpan pada disk, yang biasanya tidak diputar. Setelah perubahan, gambar sekarang berputar sesuai dengan sepotong metadata gambar yang disebut nilai orientasi. Ini memiliki implikasi penting bagi seluruh komunitas machine learning (ML/machine learning). Misalnya, jika orientasi EXIF tidak dipertimbangkan, aplikasi yang digunakan untuk membuat anotasi gambar dapat menampilkan gambar dalam orientasi yang tidak terduga dan menghasilkan label yang salah.

Dimulai dengan Chrome 89, tidakAWS dapat lagi secara otomatis mencegah rotasi gambar karena kelompok standar web W3C telah memutuskan bahwa kemampuan untuk mengontrol rotasi gambar melanggar Kebijakan Same-Origin web. Oleh karena itu, untuk memastikan pekerja manusia membuat anotasi gambar masukan Anda dalam orientasi yang dapat diprediksi saat Anda mengirimkan permintaan untuk membuat loop manusia, Anda harus menambahkan kebijakan header CORS ke bucket S3 yang berisi gambar masukan Anda.

> **Important**
>
> Jika Anda tidak menambahkan konfigurasi CORS ke bucket S3 yang berisi data masukan Anda, tugas peninjauan manusia untuk objek data input tersebut gagal.

Anda dapat menambahkan kebijakan CORS ke bucket S3 yang berisi data input di konsol Amazon S3. Untuk mengatur header CORS yang diperlukan pada bucket S3 yang berisi gambar masukan Anda di konsol S3, ikuti petunjuk yang terperinci dalam Bagaimana cara menambahkan berbagi sumber daya lintas domain dengan CORS? . Gunakan kode konfigurasi CORS berikut untuk bucket yang meng-host gambar Anda. Jika Anda menggunakan konsol Amazon S3 untuk menambahkan kebijakan ke bucket, Anda harus menggunakan format JSON.

JSON

```
[{
   "AllowedHeaders": [],
   "AllowedMethods": ["GET"],
   "AllowedOrigins": ["*"],
   "ExposeHeaders": []
}]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>*</AllowedOrigin>
   <AllowedMethod>GET</AllowedMethod>
 </CORSRule>
</CORSConfiguration>
```

Tambahkan Izin ke Peran IAM yang Digunakan untuk Membuat Definisi Aliran

Untuk membuat definisi alur, lampirkan kebijakan di bagian ini ke peran yang Anda gunakan saat membuat alur kerja tinjauan manusia di SageMaker konsol, atau saat menggunakan operasi`CreateFlowDefinition` API.

- Jika Anda menggunakan konsol untuk membuat alur kerja tinjauan manusia, masukkan peran Amazon Resource Name (ARN) di bidang peran IAM saat membuat alur kerja tinjauan manusia di konsol.
- Saat membuat definisi aliran menggunakan API, lampirkan kebijakan ini ke peran yang diteruskan ke`RoleArn` parameter`CreateFlowDefinition` operasi.

Saat Anda membuat alur kerja tinjauan manusia (definisi alur), Amazon A2I memanggil Amazon S3 untuk menyelesaikan tugas Anda. Untuk memberikan izin Amazon A2I untuk mengambil dan menyimpan file Anda di bucket Amazon S3 Anda, buat kebijakan berikut dan lampirkan ke peran Anda. Misalnya, jika gambar, dokumen, dan file lain yang Anda kirimkan untuk peninjauan manusia disimpan dalam bucket S3 bernama`my_input_bucket`, dan jika Anda ingin ulasan manusia disimpan dalam bucket bernama`my_output_bucket`, buat kebijakan berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::my_input_bucket/*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::my_output_bucket/*"
            ]
        }
    ]
}
```

Selain itu, IAM role harus memiliki kebijakan kepercayaan berikut untuk memberikan SageMaker izin untuk mengambil peran tersebut. Untuk mempelajari lebih lanjut tentang kebijakan kepercayaan IAM, lihat bagian Kebijakan Berbasis Sumber Daya pada Kebijakan dan Izin dalam dokumentasi AWSIdentity and Access Management.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSageMakerToAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi lebih lanjut tentang membuat dan mengelola IAM role dan kebijakan IAM, lihat topik berikut di PanduanAWS Identity and Access Management Pengguna:

- Untuk membuat IAM role, lihat Membuat Peran untuk Mendelegasikan Izin ke Pengguna IAM.
- Untuk mempelajari cara membuat kebijakan IAM, lihat Membuat Kebijakan IAM.
- Untuk mempelajari cara melampirkan kebijakan IAM ke peran, lihat Menambahkan dan Menghapus Izin Identitas IAM.

Membuat Pengguna yang Dapat Memanggil Operasi API Amazon A2I

Untuk menggunakan Amazon A2I untuk membuat dan memulai loop manusia untuk Amazon Rekognition, Amazon Textract, atau API runtime Amazon A2I, Anda harus menggunakan pengguna yang memiliki izin untuk menjalankan operasi Amazon A2I. Untuk melakukannya, gunakan konsol IAM untuk melampirkan kebijakan `AmazonAugmentedAIFullAccess`terkelola ke pengguna baru atau yang sudah ada.

Kebijakan ini memberikan izin kepada pengguna untuk menjalankan operasi API dari SageMaker API untuk pembuatan dan pengelolaan definisi alur serta Amazon Augmented AI Runtime API untuk pembuatan dan pengelolaan loop manusia. Untuk mempelajari lebih lanjut tentang operasi API ini, lihat Menggunakan API di Amazon Augmented AI.

`AmazonAugmentedAIFullAccess`tidak memberikan izin untuk menggunakan operasi Amazon Rekognition atau Amazon Textract API.

> **Note**
>
> Anda juga dapat melampirkan`AmazonAugmentedAIFullAccess` kebijakan ke peran IAM yang digunakan untuk membuat dan memulai loop manusia.

Untuk menyediakan akses, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup diAWS IAM Identity Center:

  Buat set izin. Ikuti petunjuk di Buat set izin di PanduanAWS IAM Identity Center Pengguna.
- Pengguna yang dikelola dalam IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti petunjuk dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diasumsikan pengguna Anda. Ikuti petunjuk dalam Membuat peran untuk pengguna IAM di Panduan Pengguna IAM.
  - (Tidak disarankan) Lampirkan kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam Menambahkan izin ke pengguna (konsol) di Panduan Pengguna IAM.

Untuk informasi selengkapnya, lihat Menambahkan dan Menghapus Izin Identitas IAM di PanduanAWS Identity and Access Management Pengguna.

Membuat Pengguna Dengan Izin untuk Memanggil Amazon A2I, Amazon Textract, dan Amazon Rekognition API Operations

Untuk membuat pengguna yang memiliki izin untuk menjalankan operasi API yang digunakan oleh jenis tugas bawaan (yaitu,`DetectModerationLables` untuk Amazon Rekognition dan Amazon Textract) dan izin`AnalyzeDocument` untuk menggunakan semua operasi API Amazon A2I, lampirkan kebijakan yang dikelola IAM`AmazonAugmentedAIIntegratedAPIAccess`. Anda mungkin ingin menggunakan kebijakan ini ketika Anda ingin memberikan izin luas kepada pengguna menggunakan Amazon A2I dengan lebih dari satu jenis tugas. Untuk mempelajari lebih lanjut tentang operasi API ini, lihat Menggunakan API di Amazon Augmented AI.

> **Note**
>
> Anda juga dapat melampirkan`AmazonAugmentedAIIntegratedAPIAccess` kebijakan ke peran IAM yang digunakan untuk membuat dan memulai loop manusia.

Untuk menyediakan akses, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup diAWS IAM Identity Center:

  Buat set izin. Ikuti petunjuk di Buat set izin di PanduanAWS IAM Identity Center Pengguna.
- Pengguna yang dikelola dalam IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti petunjuk dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diasumsikan pengguna Anda. Ikuti petunjuk dalam Membuat peran untuk pengguna IAM di Panduan Pengguna IAM.
  - (Tidak disarankan) Lampirkan kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam Menambahkan izin ke pengguna (konsol) di Panduan Pengguna IAM.

Untuk informasi selengkapnya, lihat Menambahkan dan Menghapus Izin Identitas IAM di PanduanAWS Identity and Access Management Pengguna.

Aktifkan Pratinjau Template Tugas Pekerja

Untuk menyesuaikan antarmuka dan instruksi yang dilihat pekerja saat mengerjakan tugas, Anda membuat templat tugas pekerja. Anda dapat membuat template menggunakan `CreateHumanTaskUi`operasi atau SageMaker konsol.

Untuk melihat pratinjau template Anda, Anda memerlukan peran IAM dengan izin berikut untuk membaca objek Amazon S3 yang dirender di antarmuka pengguna Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::my_input_bucket/*"
            ]
        }
    ]
}
```

Untuk jenis tugas Amazon Rekognition dan Amazon Textract, Anda dapat melihat pratinjau template Anda menggunakan bagian Amazon Augmented AI pada SageMaker konsol. Untuk jenis tugas khusus, Anda melihat pratinjau template Anda dengan menjalankan `RenderUiTemplate`operasi. Untuk melihat pratinjau template Anda, ikuti petunjuk untuk jenis tugas Anda:

- Jenis tugas Amazon Rekognition dan Amazon Textract — Di SageMaker konsol, gunakan Amazon Resource Name (ARN) peran dalam prosedur yang didokumentasikan diBuat Template Tugas Pekerja.
- Jenis tugas khusus - Dalam`RenderUiTemplate` operasi, gunakan ARN peran dalam`RoleArn` parameter.

Menggunakan Amazon A2I dengan BucketAWS KMS Terenkripsi

Jika Anda menetapkanAWS Key Management Service (AWS KMS) kunci yang dikelola pelanggan untuk mengenkripsi data keluaran `CreateFlowDefinition`, Anda harus menambahkan kebijakan IAM yang serupa dengan yang berikut ke kunci tersebut.`OutputConfig` Kebijakan ini memberikan

peran eksekusi IAM yang Anda gunakan untuk membuat izin loop manusia untuk menggunakan kunci ini untuk melakukan semua tindakan yang tercantum`"Action"`. Untuk mempelajari lebih lanjut tentang tindakan ini, lihat AWS KMSizin di PanduanAWS Key Management Service Developer.

Untuk menggunakan kebijakan ini, ganti ARN peran layanan IAM`"Principal"` dengan ARN peran eksekusi yang Anda gunakan untuk membuat alur kerja tinjauan manusia (definisi alur). Saat Anda membuat pekerjaan pelabelan menggunakan`CreateFlowDefinition`, ini adalah ARN yang Anda tentukan `RoleArn`. Perhatikan bahwa Anda tidak dapat memberikan`KmsKeyId` ketika Anda membuat definisi aliran di konsol.

```
{
    "Sid": "AllowUseOfKmsKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/service-role/example-role"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Izin Tambahan dan Sumber Daya Keamanan

- the section called “Kontrol Akses ke SageMaker Sumber Daya dengan Menggunakan Tag”.
- the section called “Kebijakan Berbasis Identitas SageMaker”
- the section called “Kontrol Pembuatan Sumber SageMaker Daya dengan Kunci Kondisi”
- the section called “Referensi Izin SageMaker API Amazon”
- Konfigurasikan keamanan di Amazon SageMaker

GunakanAmazon CloudWatch EventsAmazon Augmented AI

Amazon Augmented AI menggunakan Amazon CloudWatch Events untuk mengingatkan Anda saat status loop tinjauan manusia berubah menjadi`Completed`,`Failed`, atau`Stopped`. Pengiriman acara

ini dijamin setidaknya sekali, yang berarti semua peristiwa yang dibuat ketika loop manusia selesai berhasil dikirim ke CloudWatch Events (Amazon EventBridge). Ketika loop tinjauan berubah ke salah satu status ini, Augmented AI akan mengirimkan acara ke Acara CloudWatch yang serupa dengan yang berikut ini.

```
{
    "version":"0",
    "id":"12345678-1111-2222-3333-12345EXAMPLE",
    "detail-type":"SageMaker A2I HumanLoop Status Change",
    "source":"aws.sagemaker",
    "account":"1111111111111",
    "time":"2019-11-14T17:49:25Z",
    "region":"us-east-1",
    "resources":["arn:aws:sagemaker:us-east-1:111111111111:human-loop/humanloop-
nov-14-1"],
    "detail":{
        "creationTime":"2019-11-14T17:37:36.740Z",
        "failureCode":null,
        "failureReason":null,
        "flowDefinitionArn":"arn:aws:sagemaker:us-east-1:111111111111:flow-definition/
flowdef-nov-12",
        "humanLoopArn":"arn:aws:sagemaker:us-east-1:111111111111:human-loop/humanloop-
nov-14-1",
        "humanLoopName":"humanloop-nov-14-1",
        "humanLoopOutput":{
            "outputS3Uri":"s3://customer-output-bucket-specified-in-flow-definition/
flowdef-nov-12/2019/11/14/17/37/36/humanloop-nov-14-1/output.json"
        },
        "humanLoopStatus":"Completed"
    }
}
```

Rincian dalam output JSON meliputi yang berikut ini:

`creationTime`

Cap waktu ketika Augmented AI menciptakan lingkaran manusia.

`failureCode`

Kode kegagalan yang menunjukkan jenis kegagalan tertentu.

`failureReason`

Alasan mengapa loop manusia telah gagal. Alasan kegagalan hanya dikembalikan ketika status loop tinjauan manusia adalah`failed`.

`flowDefinitionArn`

Amazon Resource Name (ARN) dari definisi aliran, ataualur kerja tinjauan manusia.

`humanLoopArn`

Amazon Resource Name (ARN) dari loop manusia.

`humanLoopName`

Nama loop manusia.

`humanLoopOutput`

Objek yang berisi informasi tentang output dari loop manusia.

`outputS3Uri`

Lokasi objek Amazon S3 tempat Augmented AI menyimpan output loop manusia Anda.

`humanLoopStatus`

Status loop manusia.

Kirim Acara dari Human Loop Anda ke Acara CloudWatch

Untuk mengkonfigurasi aturan CloudWatch Events untuk mendapatkan pembaruan status, atauacara, untuk loop manusia Amazon A2I Anda, gunakanAWS Command Line Interface(AWS CLI)`put-rule`perintah. Saat menggunakan`put-rule`perintah, menentukan berikut untuk menerima status lingkaran manusia:

- `\"source\":[\"aws.sagemaker\"]`
- `\"detail-type\":[\"SageMaker A2I HumanLoop Status Change\"]`

Untuk mengkonfigurasi aturan CloudWatch Events untuk melihat semua perubahan status, gunakan perintah berikut dan ganti teks placeholder. Misalnya, ganti*`"A2IHumanLoopStatusChanges"`*dengan nama aturan CloudWatch Events yang unik

dan*"arn:aws:iam::111122223333:role/MyRoleForThisRule"*Amazon Resource Number (ARN) dari peran IAM dengan kebijakan kepercayaan events.amazonaws.com terlampir. Ganti*daerah*denganAWSWilayah tempat Anda ingin membuat aturan.

```
aws events put-rule --name "A2IHumanLoopStatusChanges"
    --event-pattern "{\"source\":[\"aws.sagemaker\"],\"detail-type\":[\"SageMaker A2I
 HumanLoop Status Change\"]}"
    --role-arn "arn:aws:iam::111122223333:role/MyRoleForThisRule"
    --region "region"
```

Untuk mempelajari selengkapnya tentang`put-rule`permintaan, lihatPola Peristiwa di CloudWatch Eventsdi dalamPanduan Pengguna Amazon CloudWatch Events.

Mengatur Target untuk Memproses Acara

Untuk memproses peristiwa, Anda perlu menyiapkan target. Misalnya, jika Anda ingin menerima email saat status loop manusia berubah, gunakan prosedur diMengatur Pemberitahuan Amazon SNSdi dalamPanduan Pengguna Amazon CloudWatchuntuk menyiapkan topik Amazon SNS dan berlangganan topik tersebut. Setelah Anda membuat topik, Anda dapat menggunakannya untuk membuat target.

Untuk menambahkan target ke aturan CloudWatch Events

1. Buka konsol CloudWatch:https://console.aws.amazon.com/cloudwatch/home
2. Di panel navigasi, pilih Aturan.
3. Pilih aturan yang ingin Anda tambahkan target.
4. Pilih Tindakan, dan kemudian pilih Edit.
5. Di bawahTarget, pilihTambah Targetdan pilihAWSlayanan Anda ingin bertindak ketika kejadian perubahan status loop manusia terdeteksi.
6. Konfigurasi target Anda. Untuk instruksi, lihat topik untuk mengonfigurasi target diAWSdokumentasi untuk layanan tersebut.
7. Pilih Konfigurasikan detail.
8. UntukNama, masukkan nama dan, secara opsional tentang tujuan aturan diDeskripsi.
9. Pastikan kotak centang di sampingnegara bagiandipilih sehingga aturan Anda terdaftar sebagaiDiaktifkan.
10. MemiilihMemperbarui aturan.

Gunakan Output Tinjauan Manusia

Setelah Anda menerima hasil tinjauan manusia, Anda dapat menganalisis hasilnya dan membandingkannya dengan prediksi pembelajaran mesin. JSON yang disimpan dalam bucket Amazon S3 berisi prediksi machine learning dan hasil tinjauan manusia.

Informasi Selengkapnya

Mengotomatisasi Amazon SageMaker dengan Amazon EventBridge

Menggunakan API di Amazon Augmented AI

Anda dapat membuat alur kerja tinjauan manusia atau template tugas pekerja secara terprogram. API yang Anda gunakan bergantung pada apakah Anda membuat Amazon Rekognition, Amazon Textract, atau jenis tugas khusus. Topik ini menyediakan link ke dokumentasi referensi API untuk setiap jenis tugas dan tugas pemrograman.

API berikut dapat digunakan dengan Augmented AI:

Amazon Augmented AI

- Gunakan Augmented AI API untuk memulai, menghentikan, dan menghapus loop tinjauan manusia. Anda juga dapat mencantumkan semua loop tinjauan manusia dan mengembalikan informasi tentang loop tinjauan manusia di akun Anda.

  Pelajari lebih lanjut tentang API loop tinjauan manusia diReferensi API Amazon Augmented AI.

Amazon Rekognition

- MenggunakanHumanLoopConfigparameter `DetectModerationLabels`API untukmemulaialur kerja peninjauan manusia menggunakan Amazon Rekognition.

Amazon SageMaker

- Gunakan API Amazon SageMaker untuk membuat`FlowDefinition`, juga dikenal sebagaialur kerja tinjauan manusia. Anda juga dapat membuat`HumanTaskUi`atauTemplat Tugas Pekerja.

  Untuk informasi selengkapnya, lihat`CreateFlowDefinition`atau`CreateHumanTaskUi`Dokumentasi API.

Amazon Textract

- MenggunakanHumanLoopConfigparameterAnalyzeDocumentAPI untuk memulai alur kerja tinjauan manusia menggunakan Amazon Textract.

Tutorial Terprogram

Tutorial berikut memberikan kode contoh dan petunjuk langkah demi langkah untuk membuat alur kerja tinjauan manusia dan templat tugas pekerja secara terprogram.

- Tutorial: Mulai Menggunakan API Amazon A2I
- Membuat Workflow Tinjauan Manusia (API)
- Membuat dan Memulai Loop Manusia
- Menggunakan Amazon Augmented AI dengan Amazon Rekognitiondi dalamPanduan Developer Amazon Rekognition
- Menggunakan Amazon Augmented AI dengan Amazon Textract AnalyzeDocumentdi dalamPanduan Developer Amazon Textract Texact

Mempersiapkan data

Anda dapat menggunakan Amazon SageMaker Data Wrangler untuk mengimpor, menyiapkan, mengubah, memvisualisasikan, dan menganalisis data. Anda dapat mengintegrasikan Data Wrangler ke dalam alur kerja pembelajaran mesin Anda untuk menyederhanakan dan merampingkan pra-pemrosesan data dan rekayasa fitur menggunakan sedikit atau tanpa pengkodean. Anda juga dapat menambahkan skrip Python Anda sendiri dan transformasi untuk menyesuaikan alur kerja persiapan data Anda.

Impor data dari Amazon S3, Amazon RedshiftAmazon Athena, dan gunakan Data Wrangler untuk membuat alur kerja persiapan data pembelajaran mesin yang canggih dengan transformasi dan analisis data bawaan dan kustom termasuk kebocoran target fitur dan pemodelan cepat.

Setelah Anda menentukan alur kerja persiapan data, atau aliran data, Anda dapat mengintegrasikannya dengan SageMaker Processing, SageMaker Pipelines, dan SageMaker Feature Store, menyederhanakan tugas memproses, berbagi, dan menyimpan data pelatihan ML. Anda juga dapat mengekspor aliran data Anda ke skrip python dan membuat pipeline persiapan data HTML kustom.

Untuk informasi selengkapnya, lihat Siapkan Data ML dengan Amazon SageMaker Data Wrangler.

Untuk persiapan data yang cepat dalam skala besar, Amazon SageMaker Studio Classic menyediakan integrasi bawaan dengan Amazon EMR. Anda dapat menggunakan SageMaker Studio Classic untuk menyambungkan, menyediakan, atau mengelola klaster EMR Amazon dari antarmuka notebook untuk pemrosesan data skala petabyte, analitik interaktif, dan pembelajaran mesin. Amazon EMR menggunakan kerangka kerja sumber terbuka seperti Apache Spark, Apache Hive, atau Presto. Untuk informasi selengkapnya tentang menggunakan Amazon EMR dalam SageMaker Studio Classic, lihat. Siapkan data menggunakan Amazon EMR

Atau, Anda dapat menggunakan mesin tanpa server berbasis Apache Spark dari sesi AWS Glue interaktif untuk mengumpulkan dan mengubah data dari berbagai sumber. Anda dapat mengumpulkan dan mengubah data dari pipa analitik dan ETL (ekstrak, transform, and load) Anda tanpa perlu mengelola infrastruktur. Untuk informasi selengkapnya tentang penggunaan sesi AWS Glue interaktif dalam SageMaker Studio Classic, lihatSiapkan data menggunakan Sesi AWS Glue Interaktif.

Data yang Anda gunakan untuk melatih model pembelajaran mesin Anda mungkin mengandung bias. Bias dapat menghasilkan model pembelajaran mesin yang mendiskriminasi individu atau kelompok

tertentu. Anda dapat menggunakan Amazon SageMaker Clarify untuk menentukan apakah data yang Anda gunakan untuk melatih model atau model yang dihasilkan mengkodekan bias apa pun. SageMaker Clarify juga dapat membantu Anda menjelaskan model yang dibuat dengan data tabular, gambar atau NLP dengan plot ketergantungan sebagian, kepentingan fitur, dan banyak lagi. Untuk informasi lebih lanjut tentang SageMaker Clarify, lihatMendeteksi Bias Data Pra-pelatihan.

Topik

- Jelajahi, Analisis, dan Proses Data
- Siapkan Data ML dengan Amazon SageMaker Data Wrangler
- Persiapan data skala menggunakan Apache Spark, Hive, atau Presto di Amazon EMR atau AWS Glue dari notebook Amazon Studio Classic SageMaker

Jelajahi, Analisis, dan Proses Data

Sebelum menggunakan dataset untuk melatih model, ilmuwan data biasanya mengeksplorasi, menganalisis, dan memprosesnya terlebih dahulu.

Amazon SageMaker Processing memungkinkan pekerjaan yang berjalan untuk memproses data pra-proses dan pasca proses, melakukan rekayasa fitur, dan mengevaluasi model SageMaker dengan mudah dan dalam skala besar. Ketika dikombinasikan dengan tugas pembelajaran mesin penting lainnya yang disediakan oleh SageMaker, seperti pelatihan dan hosting, Processing memberi Anda manfaat dari lingkungan pembelajaran mesin yang dikelola sepenuhnya, termasuk semua dukungan keamanan dan kepatuhan yang ada di dalamnya SageMaker. Dengan Processing, Anda memiliki fleksibilitas untuk menggunakan wadah pemrosesan data bawaan atau membawa kontainer Anda sendiri dan mengirimkan pekerjaan khusus untuk dijalankan pada infrastruktur terkelola. Setelah Anda mengirimkan pekerjaan, SageMaker meluncurkan instans komputasi, memproses dan menganalisis data input, dan melepaskan sumber daya setelah selesai. Untuk informasi selengkapnya, lihat Memproses data.

- Untuk informasi tentang cara menjalankan skrip pemrosesan data Anda sendiri, lihatPemrosesan Data dengan scikit-learn.
- Untuk informasi tentang cara membuat container pemrosesan Anda sendiri untuk menjalankan skrip, lihatBangun Kontainer Pemrosesan Anda Sendiri (Skenario Lanjutan).
- Untuk informasi tentang cara melakukan analisis data eksplorasi (EDA) dengan antarmuka tanpa kode visual, lihat. Siapkan Data ML dengan Amazon SageMaker Data Wrangler

Siapkan Data ML dengan Amazon SageMaker Data Wrangler

> **Important**
>
> Amazon SageMaker Data Wrangler telah diintegrasikan ke dalam Amazon SageMaker Canvas. Dalam pengalaman Data Wrangler baru di SageMaker Canvas, Anda dapat menggunakan antarmuka bahasa alami untuk menjelajahi dan mengubah data Anda selain antarmuka visual. Untuk informasi selengkapnya tentang Data Wrangler di SageMaker Canvas, lihat. Siapkan data

Amazon SageMaker Data Wrangler (Data Wrangler) adalah fitur Amazon SageMaker Studio Classic yang menyediakan end-to-end solusi untuk mengimpor, menyiapkan, mengubah, menyesuaikan, dan menganalisis data. Anda dapat mengintegrasikan alur persiapan data Wrangler Data ke dalam alur kerja machine learning (ML) Anda untuk menyederhanakan dan merampingkan pra-pemrosesan data dan rekayasa fitur menggunakan sedikit atau tanpa pengkodean. Anda juga dapat menambahkan skrip dan transformasi Python Anda sendiri untuk menyesuaikan alur kerja.

Data Wrangler menyediakan fungsionalitas inti berikut untuk membantu Anda menganalisis dan menyiapkan data untuk aplikasi pembelajaran mesin.

- Impor — Hubungkan ke dan impor data dari Amazon Simple Storage Service (Amazon S3), Amazon Athena (Athena), Amazon Redshift, Snowflake, dan Databricks.
- Aliran Data - Buat aliran data untuk menentukan serangkaian langkah persiapan data ML. Anda dapat menggunakan alur untuk menggabungkan kumpulan data dari sumber data yang berbeda, mengidentifikasi jumlah dan jenis transformasi yang ingin Anda terapkan ke kumpulan data, dan menentukan alur kerja persiapan data yang dapat diintegrasikan ke dalam pipeline ML.
- Transform - Bersihkan dan ubah dataset Anda menggunakan transformasi standar seperti string, vektor, dan alat pemformatan data numerik. Faturisasi data Anda menggunakan transformasi seperti penyematan teks dan tanggal/waktu serta pengkodean kategoris.
- Hasilkan Wawasan Data — Secara otomatis memverifikasi kualitas data dan mendeteksi kelainan pada data Anda dengan Data Wrangler Data Insights and Quality Report.
- Analisis — Analisis fitur dalam kumpulan data Anda di setiap titik dalam alur Anda. Data Wrangler mencakup alat visualisasi data bawaan seperti plot pencar dan histogram, serta alat analisis data seperti analisis kebocoran target dan pemodelan cepat untuk memahami korelasi fitur.

- Ekspor - Ekspor alur kerja persiapan data Anda ke lokasi yang berbeda. Berikut ini adalah contoh lokasi:
  - bucket Amazon Simple Storage Service (Amazon S3)
  - Amazon SageMaker Model Building Pipelines — Gunakan SageMaker Pipelines untuk mengotomatiskan penerapan model. Anda dapat mengekspor data yang telah Anda ubah langsung ke saluran pipa.
  - Amazon SageMaker Feature Store — Simpan fitur dan datanya di toko terpusat.
  - Skrip Python — Simpan data dan transformasinya dalam skrip Python untuk alur kerja kustom Anda.

Untuk mulai menggunakan Data Wrangler, lihat. Memulai dengan Data Wrangler

> **Important**
>
> Data Wrangler tidak lagi mendukung Jupyter Lab Versi 1 (JL1). Untuk mengakses fitur dan pembaruan terbaru, perbarui ke Jupyter Lab Versi 3. Untuk informasi selengkapnya tentang meningkatkan, lihatLihat dan perbarui JupyterLab versi aplikasi dari konsol.

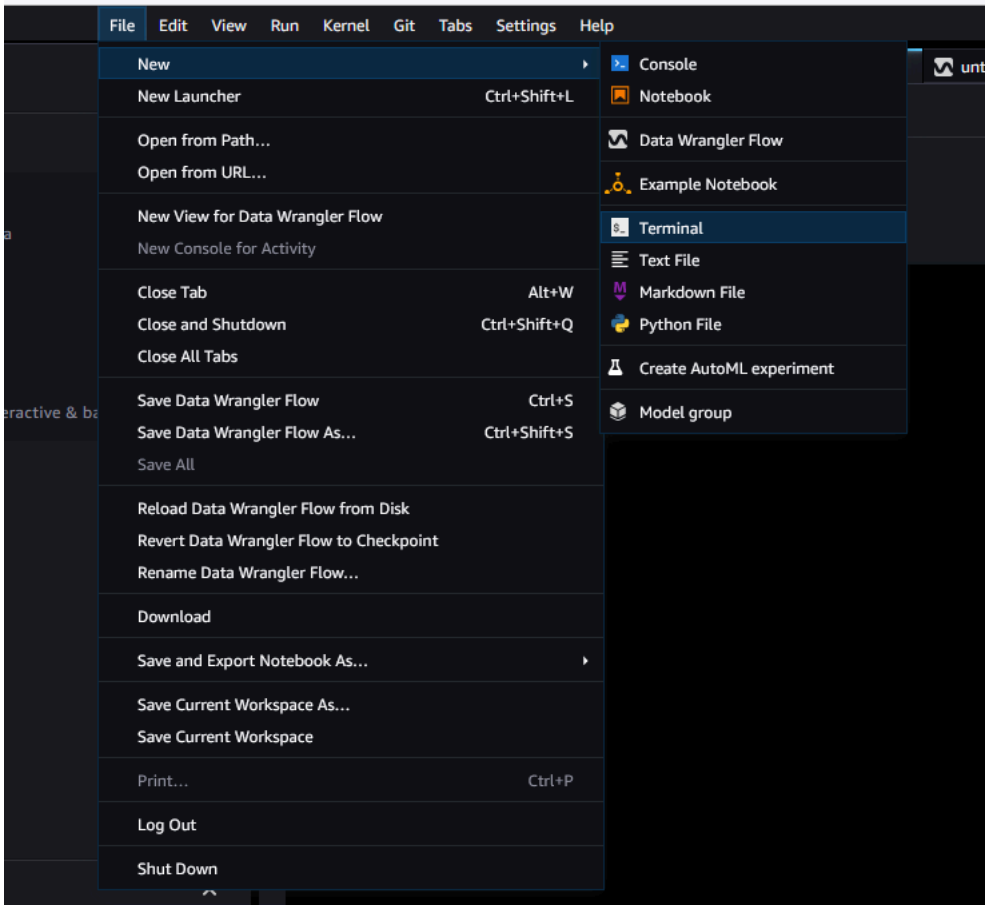
> **Important**
>
> Informasi dan prosedur dalam panduan ini menggunakan versi terbaru Amazon SageMaker Studio Classic. Untuk informasi tentang pembaruan Studio Classic ke versi terbaru, lihatIkhtisar UI Amazon SageMaker Studio Classic.

Anda harus menggunakan Studio Classic versi 1.3.0 atau yang lebih baru. Gunakan prosedur berikut untuk membuka Amazon SageMaker Studio Classic dan melihat versi mana yang Anda jalankan.

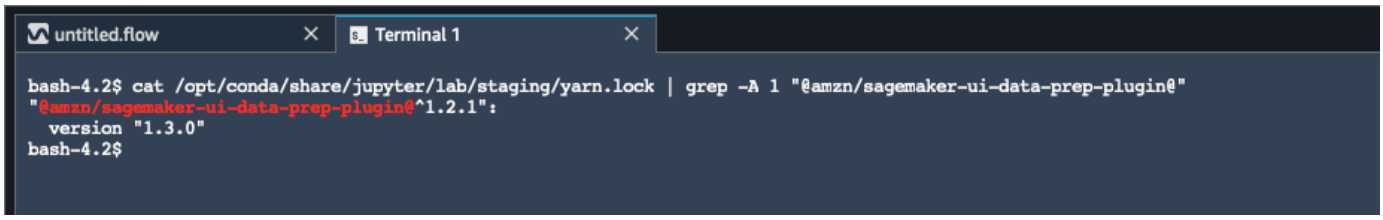
Untuk membuka Studio Classic dan memeriksa versinya, lihat prosedur berikut.

1. Gunakan langkah-langkah Prasyarat untuk mengakses Data Wrangler melalui Amazon SageMaker Studio Classic.
2. Di samping pengguna yang ingin Anda gunakan untuk meluncurkan Studio Classic, pilih Luncurkan aplikasi.
3. Pilih Studio.

4. Setelah Studio Classic dimuat, pilih File, lalu Baru, dan kemudian Terminal.



5. Setelah Anda meluncurkan Studio Classic, pilih File, lalu New, dan kemudian Terminal.
6. Masukkan `cat /opt/conda/share/jupyter/lab/staging/yarn.lock | grep -A 1 "@amzn/sagemaker-ui-data-prep-plugin@"` untuk mencetak versi instans Studio Classic Anda. Anda harus memiliki Studio Classic versi 1.3.0 untuk menggunakan Snowflake.



Anda dapat memperbarui Amazon SageMaker Studio Classic dari dalamAWS Management Console. Untuk informasi lebih lanjut tentang memperbarui Studio Classic, lihatIkhtisar UI Amazon SageMaker Studio Classic.

Topik

- Memulai dengan Data Wrangler

- Impor
- Membuat dan Menggunakan Data Wrangler Flow
- Dapatkan Wawasan Tentang Kualitas Data dan Data
- Secara Otomatis Melatih Model pada Alur Data Anda
- Memindahkan
- Analisis dan Memvisualisasikan
- Menggunakan Kembali Alur Data untuk Kumpulan Data yang Berbeda
- Ekspor
- Menggunakan Widget Persiapan Data Interaktif di Notebook Amazon SageMaker Studio Classic untuk Mendapatkan Wawasan Data
- Keamanan dan Izin
- Catatan rilis
- Pemecahan Masalah
- Meningkatkan Batas Instans Amazon EC2
- Perbarui Data Wrangler
- Matikan Data Wrangler

Memulai dengan Data Wrangler

Amazon SageMaker Data Wrangler adalah fitur di Amazon SageMaker Studio Classic. Gunakan bagian ini untuk mempelajari cara mengakses dan mulai menggunakan Data Wrangler. Lakukan hal-hal berikut:

1. Selesaikan setiap langkahPrasyarat.
2. Ikuti prosedur Akses Data untuk mulai menggunakan Data Wrangler.

Prasyarat

Untuk menggunakan Data Wrangler, Anda harus menyelesaikan prasyarat berikut.

1. Untuk menggunakan Data Wrangler, Anda memerlukan akses ke instans Amazon Elastic Compute Cloud (Amazon EC2). Untuk informasi selengkapnya tentang instans Amazon EC2 yang dapat Anda gunakan, lihat. Instans Untuk mempelajari cara melihat kuota Anda dan, jika perlu, minta peningkatan kuota, lihat kuota AWSlayanan.

2. Konfigurasikan izin yang diperlukan yang dijelaskan dalamKeamanan dan Izin.
3. Jika organisasi Anda menggunakan firewall yang memblokir lalu lintas internet, Anda harus memiliki akses ke URL berikut:
   - `https://ui.prod-1.data-wrangler.sagemaker.aws/`
   - `https://ui.prod-2.data-wrangler.sagemaker.aws/`
   - `https://ui.prod-3.data-wrangler.sagemaker.aws/`
   - `https://ui.prod-4.data-wrangler.sagemaker.aws/`

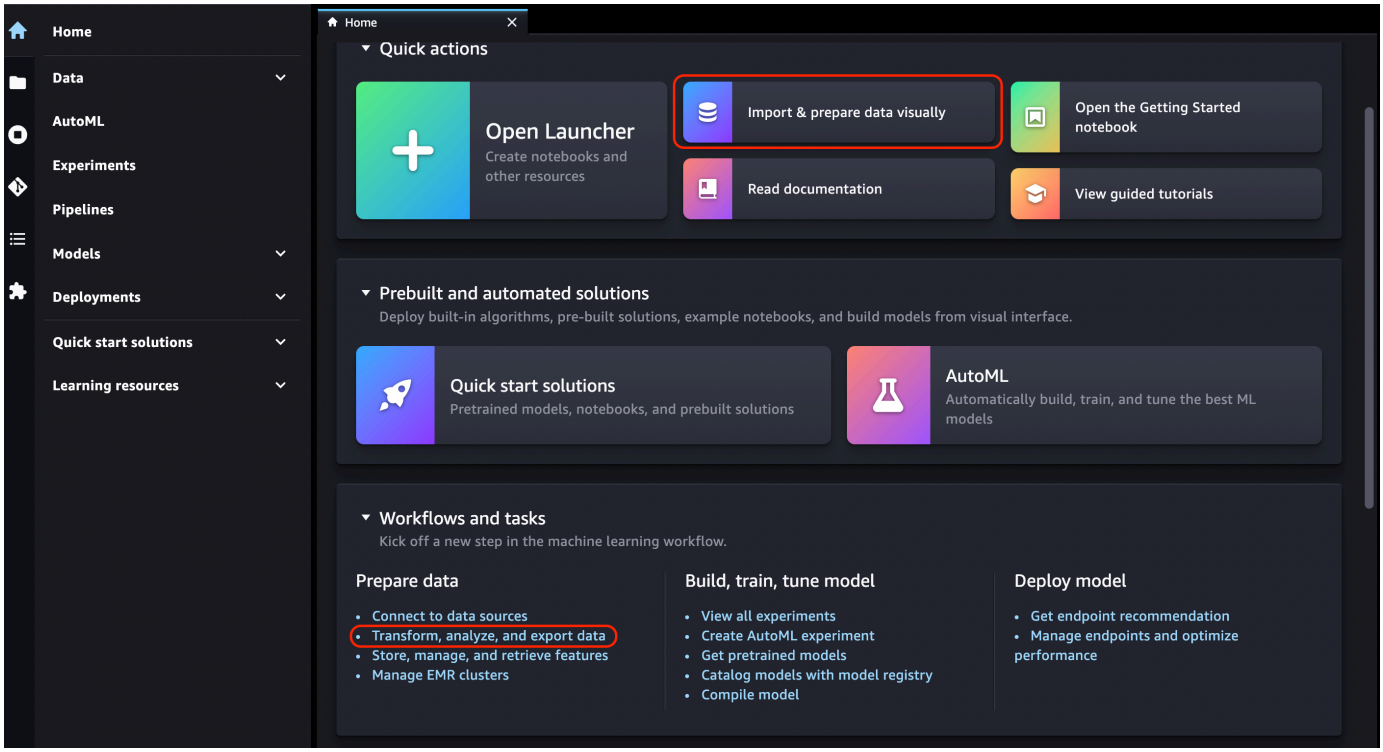
Untuk menggunakan Data Wrangler, Anda memerlukan instance Studio Classic yang aktif. Untuk mempelajari cara meluncurkan instance baru, lihatIkhtisar SageMaker Domain Amazon. Saat instans Studio Classic Anda Siap, gunakan instruksi diAkses Data.

Akses Data

Prosedur berikut mengasumsikan Anda telah menyelesaikan. Prasyarat

Untuk mengakses Data Wrangler di Studio Classic, lakukan hal berikut.

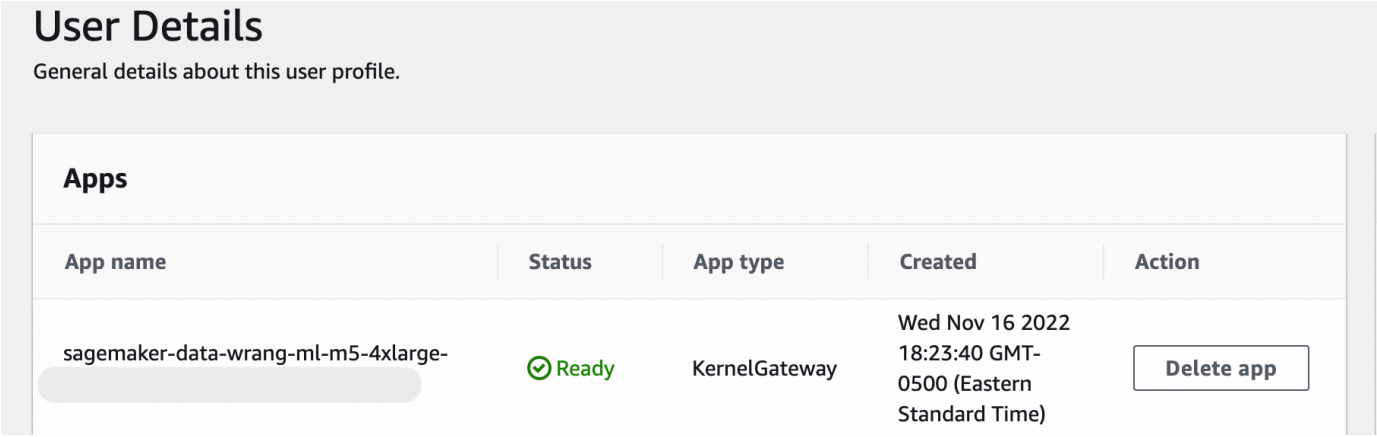
1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Anda juga dapat membuat alur Data Wrangler dengan melakukan hal berikut.
   a. Di bilah navigasi atas, pilih File.
   b. Pilih Baru.
   c. Pilih Data Wrangler Flow.



9. (Opsional) Ganti nama direktori baru dan file.flow.
10. Saat Anda membuat file.flow baru di Studio Classic, Anda mungkin melihat carousel yang memperkenalkan Anda ke Data Wrangler.

    Hal ini mungkin memerlukan waktu beberapa menit.

    Pesan ini tetap ada selama KernelGatewayaplikasi di halaman Detail Pengguna Anda Tertunda. Untuk melihat status aplikasi ini, di SageMaker konsol di halaman Amazon SageMaker Studio Classic, pilih nama pengguna yang Anda gunakan untuk mengakses Studio Classic. Pada halaman Detail Pengguna, Anda melihat KernelGatewayaplikasi di bawah Aplikasi. Tunggu hingga status aplikasi ini Siap untuk mulai menggunakan Data Wrangler. Ini bisa memakan waktu sekitar 5 menit saat pertama kali Anda meluncurkan Data Wrangler.



11. Untuk memulai, pilih sumber data dan gunakan untuk mengimpor dataset. Lihat Impor untuk mempelajari selengkapnya.

    Saat Anda mengimpor kumpulan data, itu muncul di aliran data Anda. Untuk mempelajari selengkapnya, lihat Membuat dan Menggunakan Data Wrangler Flow.

12. Setelah Anda mengimpor dataset, Data Wrangler secara otomatis menyimpulkan jenis data di setiap kolom. Pilih + di samping langkah Jenis data dan pilih Edit tipe data.

    > **Important**
    >
    > Setelah menambahkan transformasi ke langkah Tipe data, Anda tidak dapat memperbarui jenis kolom secara massal menggunakan jenis Perbarui.

13. Gunakan aliran data untuk menambahkan transformasi dan analisis. Untuk mempelajari lebih lanjut lihat Memindahkan danAnalisis dan Memvisualisasikan.

14. Untuk mengekspor aliran data lengkap, pilih Ekspor dan pilih opsi ekspor. Untuk mempelajari selengkapnya, lihat Ekspor.

15. Terakhir, pilih ikon Components and registries, dan pilih Data Wrangler dari daftar dropdown untuk melihat semua file.flow yang telah Anda buat. Anda dapat menggunakan menu ini untuk menemukan dan berpindah antar aliran data.

Setelah Anda meluncurkan Data Wrangler, Anda dapat menggunakan bagian berikut untuk menelusuri bagaimana Anda dapat menggunakan Data Wrangler untuk membuat aliran persiapan data ML.

Perbarui Data Wrangler

Kami menyarankan Anda memperbarui aplikasi Data Wrangler Studio Classic secara berkala untuk mengakses fitur dan pembaruan terbaru. Nama aplikasi Data Wrangler dimulai dengan. sagemaker-data-wrang Untuk mempelajari cara memperbarui aplikasi Studio Classic, lihatMatikan dan Perbarui Aplikasi Studio Classic.

Demo: Panduan Set Data Wrangler Titanic

Bagian berikut menyediakan panduan untuk membantu Anda memulai menggunakan Data Wrangler. Panduan ini mengasumsikan bahwa Anda telah mengikuti langkah-langkah Akses Data dan membuka file aliran data baru yang ingin Anda gunakan untuk demo. Anda mungkin ingin mengganti nama file.flow ini menjadi sesuatu yang mirip dengan. `titanic-demo.flow`

Panduan ini menggunakan dataset Titanic. Ini adalah versi modifikasi dari dataset Titanic yang dapat Anda impor ke aliran Data Wrangler Anda dengan lebih mudah. Kumpulan data ini berisi status kelangsungan hidup, usia, jenis kelamin, dan kelas (yang berfungsi sebagai proxy untuk status ekonomi) penumpang di atas pelayaran perdana RMS Titanic pada tahun 1912.

Dalam tutorial ini, Anda melakukan langkah-langkah berikut.

1. Lakukan salah satu dari cara berikut:
   - Buka alur Data Wrangler Anda dan pilih Use Sample Dataset.
   - Unggah dataset Titanic ke Amazon Simple Storage Service (Amazon S3), lalu impor dataset ini ke Data Wrangler.
2. Analisis dataset ini menggunakan analisis Data Wrangler.
3. Tentukan aliran data menggunakan transformasi data Wrangler Data.
4. Ekspor alur Anda ke Notebook Jupyter yang dapat Anda gunakan untuk membuat pekerjaan Data Wrangler.
5. Memproses data Anda, dan memulai pekerjaan SageMaker pelatihan untuk melatih XGBoost Binary Classifier.

Unggah Dataset ke S3 dan Impor

Untuk memulai, Anda dapat menggunakan salah satu metode berikut ini untuk mengimpor dataset Titanic ke Data Wrangler:

- Mengimpor dataset langsung dari aliran Data Wrangler

- Mengunggah kumpulan data ke Amazon S3 dan kemudian mengimpornya ke Data Wrangler

Untuk mengimpor dataset langsung ke Data Wrangler, buka alur dan pilih Use Sample Dataset.

Mengunggah kumpulan data ke Amazon S3 dan mengimpornya ke Data Wrangler lebih dekat dengan pengalaman Anda mengimpor data Anda sendiri. Informasi berikut memberi tahu Anda cara mengunggah kumpulan data Anda dan mengimpornya.

Sebelum Anda mulai mengimpor data ke Data Wrangler, unduh dataset Titanic dan unggah ke bucket Amazon S3 (Amazon S3) di AWS Wilayah tempat Anda ingin menyelesaikan demo ini.

Jika Anda adalah pengguna baru Amazon S3, Anda dapat melakukan ini menggunakan drag and drop di konsol Amazon S3. Untuk mempelajari caranya, lihat Mengunggah File dan Folder dengan Menggunakan Seret dan Jatuhkan di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

> **Important**
>
> Unggah kumpulan data Anda ke bucket S3 di AWS Wilayah yang sama yang ingin Anda gunakan untuk menyelesaikan demo ini.

Saat dataset Anda berhasil diunggah ke Amazon S3, Anda dapat mengimpornya ke Data Wrangler.

Impor dataset Titanic ke Data Wrangler

1. Pilih tombol Impor data di tab Aliran data Anda atau pilih tab Impor.
2. Pilih Amazon S3.
3. Gunakan tabel Impor kumpulan data dari S3 untuk menemukan bucket tempat Anda menambahkan kumpulan data Titanic. Pilih file CSV kumpulan data Titanic untuk membuka panel Detail.
4. Di bawah Detail, jenis File harus CSV. Periksa Baris pertama adalah header untuk menentukan bahwa baris pertama dari dataset adalah header. Anda juga dapat memberi nama kumpulan data dengan sesuatu yang lebih ramah, seperti**`Titanic-train`**.
5. Pilih tombol Impor.

Ketika dataset Anda diimpor ke Data Wrangler, itu muncul di tab Aliran Data Anda. Anda dapat mengklik dua kali pada node untuk memasukkan tampilan detail node, yang memungkinkan Anda

menambahkan transformasi atau analisis. Anda dapat menggunakan ikon plus untuk akses cepat ke navigasi. Di bagian selanjutnya, Anda menggunakan aliran data ini untuk menambahkan analisis dan mengubah langkah-langkah.

Aliran Data

Di bagian aliran data, satu-satunya langkah dalam aliran data adalah dataset Anda yang baru saja diimpor dan langkah tipe Data. Setelah menerapkan transformasi, Anda dapat kembali ke tab ini dan melihat seperti apa aliran datanya. Sekarang, tambahkan beberapa transformasi dasar di bawah tab Siapkan dan Analisis.

Persiapkan dan Visualisasi

Data Wrangler memiliki transformasi dan visualisasi bawaan yang dapat Anda gunakan untuk menganalisis, membersihkan, dan mengubah data Anda.

Tab Data dari tampilan detail node mencantumkan semua transformasi bawaan di panel kanan, yang juga berisi area di mana Anda dapat menambahkan transformasi khusus. Kasus penggunaan berikut menampilkan cara menggunakan transformasi ini.

Untuk mendapatkan informasi yang dapat membantu Anda dalam eksplorasi data dan rekayasa fitur, buat laporan kualitas data dan wawasan. Informasi dari laporan dapat membantu Anda membersihkan dan memproses data Anda. Ini memberi Anda informasi seperti jumlah nilai yang hilang dan jumlah outlier. Jika Anda memiliki masalah dengan data Anda, seperti kebocoran target atau ketidakseimbangan, laporan wawasan dapat membawa masalah tersebut ke perhatian Anda. Untuk informasi lebih lanjut tentang cara membuat laporan, lihatDapatkan Wawasan Tentang Kualitas Data dan Data.

Eksplorasi

Pertama, buat ringkasan tabel data menggunakan analisis. Lakukan hal-hal berikut:

1. Pilih + di sebelah langkah Tipe data dalam aliran data Anda dan pilih Tambahkan analisis.
2. Di area Analisis, pilih Ringkasan tabel dari daftar dropdown.
3. Berikan ringkasan tabel sebuah Nama.
4. Pilih Pratinjau untuk melihat pratinjau tabel yang akan dibuat.
5. Pilih Simpan untuk menyimpannya ke aliran data Anda. Itu muncul di bawah Semua Analisis.

Dengan menggunakan statistik yang Anda lihat, Anda dapat melakukan pengamatan yang serupa dengan yang berikut tentang kumpulan data ini:

- Rata-rata tarif (rata-rata) adalah sekitar $33, sedangkan maks lebih dari $500. Kolom ini kemungkinan memiliki outlier.
- Dataset ini menggunakan? untuk menunjukkan nilai yang hilang. Sejumlah kolom memiliki nilai yang hilang: cabin, embarked, dan home.dest
- Kategori usia tidak memiliki lebih dari 250 nilai.

Selanjutnya, bersihkan data Anda menggunakan wawasan yang diperoleh dari statistik ini.

Menghapus Daftar

Dengan menggunakan analisis dari bagian sebelumnya, bersihkan kumpulan data untuk mempersiapkannya untuk pelatihan. Untuk menambahkan transformasi baru ke aliran data Anda, pilih + di sebelah langkah Jenis data dalam aliran data Anda dan pilih Tambahkan transformasi.

Pertama, jatuhkan kolom yang tidak ingin Anda gunakan untuk pelatihan. Anda dapat menggunakan pustaka analisis data panda untuk melakukan ini, atau Anda dapat menggunakan salah satu transformasi bawaan.

Gunakan prosedur berikut untuk menjatuhkan kolom yang tidak digunakan.

Untuk menjatuhkan kolom yang tidak digunakan.

1. Buka alur Data Wrangler.
2. Ada dua node dalam aliran Data Wrangler Anda. Pilih + di sebelah kanan node tipe Data.
3. Pilih Tambahkan transformasi.
4. Di kolom Semua langkah, pilih Tambahkan langkah.
5. Dalam daftar Transformasi standar, pilih Kelola Kolom. Transformasi standar sudah jadi, transformasi bawaan. Pastikan kolom Drop dipilih.
6. Di bawah Kolom untuk dijatuhkan, periksa nama kolom berikut:
    - kabin
    - karcis
    - name
    - sibsp

- parch
- rumah.dest
- perahu
- body

7. Pilih Pratinjau.
8. Verifikasi bahwa kolom telah dijatuhkan, lalu pilih Tambah.

Untuk melakukannya menggunakan panda, ikuti langkah-langkah berikut.

1. Di kolom Semua langkah, pilih Tambahkan langkah.
2. Dalam daftar Custom transform, pilih Custom transform.
3. Berikan nama untuk transformasi Anda, dan pilih Python (Pandas) dari daftar dropdown.
4. Masukkan skrip Python berikut di kotak kode.

```
cols = ['name', 'ticket', 'cabin', 'sibsp', 'parch', 'home.dest','boat', 'body']
df = df.drop(cols, axis=1)
```

5. Pilih Pratinjau untuk melihat pratinjau perubahan, lalu pilih Tambah untuk menambahkan transformasi.

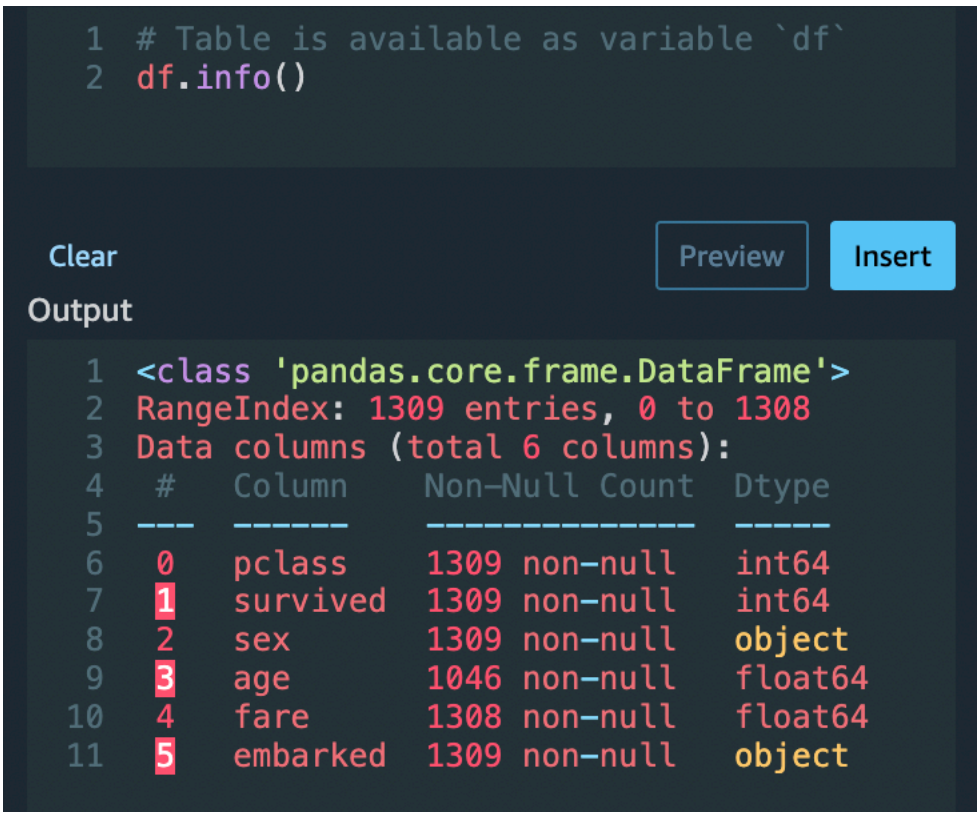
Bersihkan Nilai yang Hilang

Sekarang, bersihkan nilai yang hilang. Anda dapat melakukan ini dengan Menangani grup transformasi nilai yang hilang.

Sejumlah kolom memiliki nilai yang hilang. Dari kolom yang tersisa, usia dan tarif mengandung nilai yang hilang. Periksa ini menggunakan Custom Transform.

Menggunakan opsi Python (Pandas), gunakan yang berikut ini untuk meninjau dengan cepat jumlah entri di setiap kolom:

```
df.info()
```



Untuk menjatuhkan baris dengan nilai yang hilang dalam kategori usia, lakukan hal berikut:

1. Pilih Handle hilang.
2. Pilih Drop missing untuk Transformer.
3. Pilih usia untuk kolom Input.
4. Pilih Pratinjau untuk melihat bingkai data baru, lalu pilih Tambah untuk menambahkan transformasi ke alur Anda.
5. Ulangi proses yang sama untuk ongkos.

Anda dapat menggunakan `df.info()` di bagian Custom transform untuk mengonfirmasi bahwa semua baris sekarang memiliki 1.045 nilai.

Panda Kustom: Encode

Coba pengkodean datar menggunakan Pandas. Pengkodean data kategoris adalah proses menciptakan representasi numerik untuk kategori. Misalnya, jika kategori Anda `Dog` dan`Cat`, Anda dapat menyandikan informasi ini menjadi dua vektor: `[1,0]` untuk mewakili`Dog`, dan `[0,1]` untuk mewakili. `Cat`

1. Di bagian Custom Transform, pilih Python (Pandas) dari daftar dropdown.
2. Masukkan yang berikut ini di kotak kode.

```
import pandas as pd

dummies = []
cols = ['pclass','sex','embarked']
for col in cols:
    dummies.append(pd.get_dummies(df[col]))

encoded = pd.concat(dummies, axis=1)

df = pd.concat((df, encoded),axis=1)
```

3. Pilih Pratinjau untuk melihat pratinjau perubahan. Versi dikodekan dari setiap kolom akan ditambahkan ke dataset.
4. Pilih Tambah untuk menambahkan transformasi.

Kustom SQL: PILIH Kolom

Sekarang, pilih kolom yang ingin Anda gunakan SQL. Untuk demo ini, pilih kolom yang tercantum dalam `SELECT` pernyataan berikut. Karena bertahan adalah kolom target Anda untuk pelatihan, letakkan kolom itu terlebih dahulu.

1. Di bagian Custom Transform, pilih SQL (PySpark SQL) dari daftar dropdown.
2. Masukkan yang berikut ini di kotak kode.

```
SELECT survived, age, fare, 1, 2, 3, female, male, C, Q, S FROM df;
```

3. Pilih Pratinjau untuk melihat pratinjau perubahan. Kolom yang tercantum dalam `SELECT` pernyataan Anda adalah satu-satunya kolom yang tersisa.
4. Pilih Tambah untuk menambahkan transformasi.

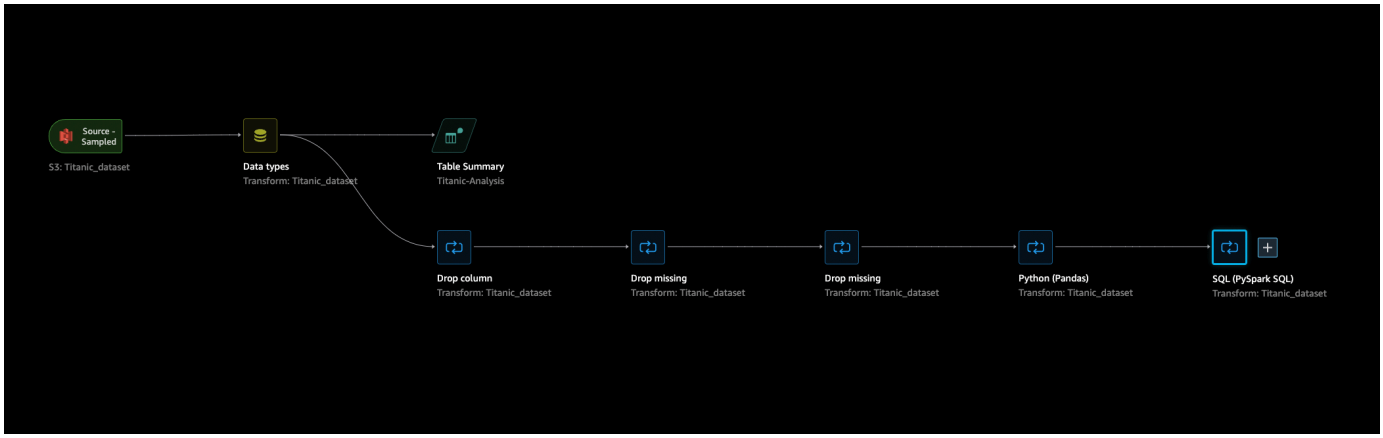
Ekspor ke Notebook Data Wrangler

Setelah selesai membuat aliran data, Anda memiliki sejumlah opsi ekspor. Bagian berikut menjelaskan cara mengekspor ke buku catatan pekerjaan Data Wrangler. Pekerjaan Data Wrangler digunakan untuk memproses data Anda menggunakan langkah-langkah yang ditentukan dalam aliran data Anda. Untuk mempelajari lebih lanjut tentang semua opsi ekspor, lihatEkspor.

Ekspor ke Data Wrangler Job Notebook

Saat Anda mengekspor aliran data menggunakan pekerjaan Data Wrangler, proses akan secara otomatis membuat Notebook Jupyter. Notebook ini secara otomatis terbuka di instans Studio Classic Anda dan dikonfigurasi untuk menjalankan pekerjaan SageMaker pemrosesan untuk menjalankan aliran data Wrangler Data Anda, yang disebut sebagai pekerjaan Data Wrangler.

1. Simpan aliran data Anda. Pilih File dan kemudian pilih Save Data Wrangler Flow.
2. Kembali ke tab Aliran Data, pilih langkah terakhir dalam aliran data Anda (SQL), lalu pilih + untuk membuka navigasi.
3. Pilih Ekspor, dan Amazon S3 (melalui Jupyter Notebook). Ini membuka Notebook Jupyter.



4. Pilih kernel Python 3 (Data Science) untuk Kernel.
5. Saat kernel dimulai, jalankan sel di buku catatan hingga Kick off SageMaker Training Job (Opsional).
6. Secara opsional, Anda dapat menjalankan sel di Kick off SageMaker Training Job (Opsional) jika Anda ingin membuat pekerjaan SageMaker pelatihan untuk melatih pengklasifikasi XGBoost. Anda dapat menemukan biaya untuk menjalankan pekerjaan SageMaker pelatihan di Amazon SageMaker Pricing.

   Atau, Anda dapat menambahkan blok kode yang ditemukan di notebook dan menjalankannya Pelatihan Pengklasifikasi XGBoost untuk menggunakan pustaka sumber terbuka XGBoost untuk melatih pengklasifikasi XGBoost.
7. Hapus komentar dan jalankan sel di bawah Pembersihan dan jalankan untuk mengembalikan SageMaker Python SDK ke versi aslinya.

Anda dapat memantau status pekerjaan Data Wrangler Anda di SageMaker konsol di tab Processing. Selain itu, Anda dapat memantau pekerjaan Data Wrangler Anda menggunakan Amazon.

CloudWatch Untuk informasi tambahan, lihat Memantau Pekerjaan SageMaker Pemrosesan Amazon dengan CloudWatch Log dan Metrik.

Jika Anda memulai pekerjaan pelatihan, Anda dapat memantau statusnya menggunakan SageMaker konsol di bawah Pekerjaan pelatihan di bagian Pelatihan.

Pelatihan Pengklasifikasi XGBoost

Anda dapat melatih XGBoost Binary Classifier menggunakan notebook Jupyter atau Amazon Autopilot. SageMaker Anda dapat menggunakan Autopilot untuk secara otomatis melatih dan menyetel model pada data yang telah Anda ubah langsung dari alur Data Wrangler Anda. Untuk informasi tentang Autopilot, lihat. Secara Otomatis Melatih Model pada Alur Data Anda

Di buku catatan yang sama yang memulai pekerjaan Data Wrangler, Anda dapat menarik data dan melatih XGBoost Binary Classifier menggunakan data yang disiapkan dengan persiapan data minimal.

1. Pertama, tingkatkan modul yang diperlukan menggunakan `pip` dan hapus file _SUCCESS (file terakhir ini bermasalah saat menggunakan`awswrangler`).

```
! pip install --upgrade awscli awswrangler boto sklearn
! aws s3 rm {output_path} --recursive  --exclude "*" --include "*_SUCCESS*"
```

2. Baca data dari Amazon S3. Anda dapat menggunakan `awswrangler` untuk membaca semua file CSV secara rekursif di awalan S3. Data kemudian dibagi menjadi fitur dan label. Label adalah kolom pertama dari kerangka data.

```
import awswrangler as wr

df = wr.s3.read_csv(path=output_path, dataset=True)
X, y = df.iloc[:,:-1],df.iloc[:,-1]
```

   - Terakhir, buat DMatrices (struktur primitif XGBoost untuk data) dan lakukan validasi silang menggunakan klasifikasi biner XGBoost.

```
import xgboost as xgb

dmatrix = xgb.DMatrix(data=X, label=y)

params = {"objective":"binary:logistic",'learning_rate': 0.1, 'max_depth': 5,
 'alpha': 10}
```

```
xgb.cv(
    dtrain=dmatrix,
    params=params,
    nfold=3,
    num_boost_round=50,
    early_stopping_rounds=10,
    metrics="rmse",
    as_pandas=True,
    seed=123)
```

Mematikan Data Wrangler

Setelah selesai menggunakan Data Wrangler, kami sarankan Anda mematikan instans yang dijalankan untuk menghindari biaya tambahan. Untuk mempelajari cara mematikan aplikasi Data Wrangler dan instance terkait, lihat. Matikan Data Wrangler

Impor

Anda dapat menggunakan Amazon SageMaker Data Wrangler untuk mengimpor data dari sumber data berikut: Amazon Simple Storage Service (Amazon S3), Amazon Athena, Amazon Redshift, dan Snowflake. Dataset yang Anda impor dapat menyertakan hingga 1000 kolom.

Topik

- Impor data dari Amazon S3
- Impor data dari Athena
- Pengimporan data dari Amazon Redshift
- Pengimporan data dari Amazon EMR
- Impor data dari Databricks (JDBC)
- Impor data dari Salesforce Data Cloud
- Impor data dari Snowflake
- Impor Data Dari Perangkat Lunak sebagai Platform Layanan (SaaS)
- Penyimpanan Data yang Diimpor

Beberapa sumber data memungkinkan Anda menambahkan beberapa koneksi data:

- Anda dapat terhubung ke beberapa cluster Amazon Redshift. Setiap klaster menjadi sumber data.

- Anda dapat menanyakan database Athena apa pun di akun Anda untuk mengimpor data dari database tersebut.

Ketika Anda mengimpor dataset dari sumber data, itu akan muncul dalam aliran data Anda. Data Wrangler secara otomatis menyimpulkan tipe data setiap kolom dalam kumpulan data Anda. Untuk mengubah jenis ini, pilih langkah Jenis data dan pilih Edit tipe data.

Saat Anda mengimpor data dari Athena atau Amazon Redshift, data yang diimpor secara otomatis disimpan di bucket S3 SageMaker default untuk Wilayah tempat Anda AWS menggunakan Studio Classic. Selain itu, Athena menyimpan data yang Anda pratinjau di Data Wrangler di bucket ini. Untuk mempelajari selengkapnya, lihat Penyimpanan Data yang Diimpor.

> **Important**
>
> Bucket Amazon S3 default mungkin tidak memiliki setelan keamanan yang paling tidak permisif, seperti kebijakan bucket dan enkripsi sisi server (SSE). Kami sangat menyarankan Anda Menambahkan Kebijakan Bucket Untuk Membatasi Akses ke Kumpulan Data yang Diimpor ke Data Wrangler.

> **Important**
>
> Selain itu, jika Anda menggunakan kebijakan terkelola untuk SageMaker, kami sangat menyarankan agar Anda mencakupnya ke kebijakan paling ketat yang memungkinkan Anda untuk melakukan kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Berikan Izin Peran IAM untuk Menggunakan Data Wrangler.

Semua sumber data kecuali Amazon Simple Storage Service (Amazon S3) mengharuskan Anda menentukan kueri SQL untuk mengimpor data Anda. Untuk setiap kueri, Anda harus menentukan hal berikut:

- Katalog data
- Basis data
- Tabel

Anda dapat menentukan nama database atau katalog data di menu drop-down atau dalam kueri. Berikut ini adalah contoh kueri:

- `select * from example-data-catalog-name.example-database-name.example-table-name`— Kueri tidak menggunakan apa pun yang ditentukan dalam menu tarik-turun antarmuka pengguna (UI) untuk dijalankan. Ini pertanyaan di `example-table-name` `example-database-name` dalam`example-data-catalog-name`.
- `select * from example-database-name.example-table-name`— Kueri menggunakan katalog data yang telah Anda tentukan di menu tarik-turun katalog Data untuk dijalankan. Ini query `example-table-name` dalam `example-database-name` dalam katalog data yang telah Anda tentukan.
- `select * from example-table-name`— Kueri mengharuskan Anda untuk memilih bidang untuk katalog Data dan menu tarik-turun nama Database. Ini query `example-table-name` dalam katalog data dalam database dan katalog data yang telah Anda tentukan.

Hubungan antara Data Wrangler dan sumber data adalah koneksi. Anda menggunakan koneksi untuk mengimpor data dari sumber data Anda.

Ada beberapa tipe koneksi berikut:

- Langsung
- Dikatalogkan

Data Wrangler selalu memiliki akses ke data terbaru dalam koneksi langsung. Jika data dalam sumber data telah diperbarui, Anda dapat menggunakan koneksi untuk mengimpor data. Misalnya, jika seseorang menambahkan file ke salah satu bucket Amazon S3 Anda, Anda dapat mengimpor file tersebut.

Koneksi yang dikatalogkan adalah hasil dari transfer data. Data dalam koneksi yang dikatalogkan tidak selalu memiliki data terbaru. Misalnya, Anda dapat mengatur transfer data antara Salesforce dan Amazon S3. Jika ada pembaruan pada data Salesforce, Anda harus mentransfer data lagi. Anda dapat mengotomatisasi proses transfer data. Untuk informasi selengkapnya tentang transfer data, lihatImpor Data Dari Perangkat Lunak sebagai Platform Layanan (SaaS).

Impor data dari Amazon S3

Anda dapat menggunakan Amazon Simple Storage Service (Amazon S3) untuk menyimpan dan mengambil data dalam jumlah berapa pun, kapan saja, dari mana saja di web. Anda dapat

menyelesaikan tugas ini menggunakanAWS Management Console, yang merupakan antarmuka web yang sederhana dan intuitif, dan Amazon S3 API. Jika Anda telah menyimpan kumpulan data secara lokal, kami sarankan Anda menambahkannya ke bucket S3 untuk diimpor ke Data Wrangler. Untuk mempelajari caranya, lihat Mengunggah objek ke bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Data Wrangler menggunakan S3 Select untuk memungkinkan Anda melihat pratinjau file Amazon S3 Anda di Data Wrangler. Anda dikenakan biaya standar untuk setiap pratinjau file. Untuk mempelajari lebih lanjut tentang harga, lihat tab Permintaan & pengambilan data pada harga Amazon S3.

Important

Jika Anda berencana untuk mengekspor aliran data dan meluncurkan pekerjaan Data Wrangler, memasukkan data ke dalam SageMaker feature store, atau membuat SageMaker pipeline, ketahuilah bahwa integrasi ini memerlukan data input Amazon S3 untuk ditempatkan di wilayah yang sama. AWS

Important

Jika Anda mengimpor file CSV, pastikan file tersebut memenuhi persyaratan berikut:

- Catatan dalam dataset Anda tidak bisa lebih panjang dari satu baris.
- Garis miring terbalik,\, adalah satu-satunya karakter pelarian yang valid.
- Dataset Anda harus menggunakan salah satu pembatas berikut:
  - Koma — `,`
  - Usus besar — `:`
  - Titik koma — `;`
  - Pipa — `|`
  - Tab — `[TAB]`

Untuk menghemat ruang, Anda dapat mengimpor file CSV terkompresi.

Data Wrangler memberi Anda kemampuan untuk mengimpor seluruh kumpulan data atau mengambil sampel sebagian darinya. Untuk Amazon S3, ini menyediakan opsi pengambilan sampel berikut:

- Tidak ada - Impor seluruh set data.
- K Pertama - Contoh baris K pertama dari kumpulan data, di mana K adalah bilangan bulat yang Anda tentukan.
- Acak - Mengambil sampel acak dari ukuran yang Anda tentukan.
- Bertingkat — Mengambil sampel acak bertingkat. Sampel bertingkat mempertahankan rasio nilai dalam kolom.

Setelah mengimpor data, Anda juga dapat menggunakan transformator sampling untuk mengambil satu atau lebih sampel dari seluruh kumpulan data Anda. Untuk informasi lebih lanjut tentang transformator sampel, lihatPengambilan sampel.

Anda dapat menggunakan salah satu pengidentifikasi sumber daya berikut untuk mengimpor data Anda:

- URI Amazon S3 yang menggunakan bucket Amazon S3 atau titik akses Amazon S3
- Alias titik akses Amazon S3
- Amazon Resource Name (ARN) yang menggunakan titik akses Amazon S3 atau bucket Amazon S3

Titik akses Amazon S3 diberi nama titik akhir jaringan yang terpasang ke bucket. Setiap titik akses memiliki izin dan kontrol jaringan yang berbeda yang dapat Anda konfigurasi. Untuk informasi selengkapnya tentang jalur akses, lihat Mengelola akses data dengan Amazon S3 access points.

> **Important**
>
> Jika Anda menggunakan Nama Sumber Daya Amazon (ARN) untuk mengimpor data Anda, itu harus untuk sumber daya yang terletak sama dengan Wilayah AWS yang Anda gunakan untuk mengakses Amazon SageMaker Studio Classic.

Anda dapat mengimpor satu file atau beberapa file sebagai kumpulan data. Anda dapat menggunakan operasi impor multifile ketika Anda memiliki kumpulan data yang dipartisi menjadi file terpisah. Dibutuhkan semua file dari direktori Amazon S3 dan mengimpornya sebagai kumpulan data tunggal. Untuk informasi tentang jenis file yang dapat Anda impor dan cara mengimpornya, lihat bagian berikut.

Single File Import

Anda dapat mengimpor file tunggal dalam format berikut:

- Nilai Dipisahkan Koma (CSV)
- Parquet
- Notasi Objek Javascript (JSON)
- Optimized Row Columnar (ORC)
- Gambar - Data Wrangler menggunakan OpenCV untuk mengimpor gambar. Untuk informasi selengkapnya tentang format gambar yang didukung, lihat Membaca dan menulis file gambar.

Untuk file yang diformat dalam JSON, Data Wrangler mendukung kedua baris JSON (.jsonl) dan dokumen JSON (.json). Saat Anda melihat pratinjau data Anda, secara otomatis menampilkan JSON dalam format tabel. Untuk dokumen JSON bersarang yang lebih besar dari 5 MB, Data Wrangler menunjukkan skema untuk struktur dan array sebagai nilai dalam kumpulan data. Gunakan operator array Flatten terstruktur dan Explode untuk menampilkan nilai bersarang dalam format tabel. Lihat informasi yang lebih lengkap di Fungsi JSON dan Penyempurnaan Data.

Saat memilih kumpulan data, Anda dapat mengganti namanya, menentukan jenis file, dan mengidentifikasi baris pertama sebagai header.

Anda dapat mengimpor kumpulan data yang telah dipartisi menjadi beberapa file di bucket Amazon S3 dalam satu langkah impor.

Untuk mengimpor dataset ke Data Wrangler dari satu file yang telah Anda simpan di Amazon S3:

1. Jika saat ini Anda tidak berada di tab Impor, pilih Impor.
2. Di bawah Tersedia, pilih Amazon S3.
3. Dari Impor tabular, gambar, atau data deret waktu dari S3, lakukan salah satu hal berikut:
    - Pilih bucket Amazon S3 dari tampilan tabel lalu arahkan ke file yang Anda impor.
    - Untuk sumber S3, tentukan bucket Amazon S3 atau URI Amazon S3 dan pilih Go. Amazon S3 URI dapat berupa salah satu dari berikut:
        - `s3://DOC-EXAMPLE-BUCKET/example-prefix/example-file`
        - *example-access-point- aqfqprnstn7aefdfbarligizwgyfouse1a - s3alias/datasets/contoh-file*

- `s3://arn:aws:s3:AWS-Region:111122223333:accesspoint/example-prefix/example-file`

4. Pilih kumpulan data untuk membuka panel Pengaturan impor.
5. Jika file CSV Anda memiliki header, pilih kotak centang di sebelah Tambahkan header ke tabel.
6. Gunakan tabel Pratinjau untuk melihat pratinjau kumpulan data Anda. Tabel ini menunjukkan hingga 100 baris.
7. Di panel Detail, verifikasi atau ubah Nama dan Jenis File untuk kumpulan data Anda. Jika Anda menambahkan Nama yang berisi spasi, spasi ini akan diganti dengan garis bawah saat dataset Anda diimpor.
8. Tentukan konfigurasi pengambilan sampel yang ingin Anda gunakan.
9. Pilih Impor.

Multifile Import

Berikut ini adalah persyaratan untuk mengimpor beberapa file:

- File harus berada di folder yang sama dengan bucket Amazon S3.
- File harus berbagi header yang sama atau tidak memiliki header.

Setiap file harus berada dalam salah satu format berikut:

- CSV
- Parquet
- Optimized Row Columnar (ORC)
- Gambar - Data Wrangler menggunakan OpenCV untuk mengimpor gambar. Untuk informasi selengkapnya tentang format gambar yang didukung, lihat Membaca dan menulis file gambar.

Gunakan prosedur berikut untuk mengimpor beberapa file.

Untuk mengimpor dataset ke Data Wrangler dari beberapa file yang telah disimpan di direktori Amazon S3

1. Jika saat ini Anda tidak berada di tab Impor, pilih Impor.

2. Di bawah Tersedia, pilih Amazon S3.
3. Dari Impor tabular, gambar, atau data deret waktu dari S3, lakukan salah satu hal berikut:
   - Pilih bucket Amazon S3 dari tampilan tabular dan navigasikan ke folder yang berisi file yang Anda impor.
   - Untuk sumber S3, tentukan bucket Amazon S3 atau URI Amazon S3 dengan file Anda, lalu pilih Go. Berikut ini adalah URI yang valid:
     - `s3://DOC-EXAMPLE-BUCKET/example-prefix/example-prefix`
     - `example-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-s3alias/example-prefix/`
     - `s3://arn:aws:s3:AWS-Region:111122223333:accesspoint/example-prefix`
4. Pilih folder yang berisi file yang ingin Anda impor. Setiap file harus berada dalam salah satu format yang didukung. File Anda harus memiliki tipe data yang sama.
5. Jika folder Anda berisi file CSV dengan header, pilih kotak centang di sebelah Baris pertama adalah header.
6. Jika file Anda bersarang di dalam folder lain, pilih kotak centang di samping Sertakan direktori bersarang.
7. (Opsional) Pilih Tambahkan kolom nama file tambahkan kolom ke kumpulan data yang menunjukkan nama file untuk setiap pengamatan.
8. (Opsional) Secara default, Data Wrangler tidak menampilkan pratinjau folder. Anda dapat mengaktifkan pratinjau dengan memilih tombol mati Pratinjau biru. Pratinjau menunjukkan 10 baris pertama dari 10 file pertama di folder.
9. Di panel Detail, verifikasi atau ubah Nama dan Jenis File untuk kumpulan data Anda. Jika Anda menambahkan Nama yang berisi spasi, spasi ini akan diganti dengan garis bawah saat dataset Anda diimpor.
10. Tentukan konfigurasi pengambilan sampel yang ingin Anda gunakan.
11. Pilih Impor dataset.

Anda juga dapat menggunakan parameter untuk mengimpor subset file yang cocok dengan pola. Parameter membantu Anda memilih file yang Anda impor secara lebih selektif. Untuk mulai menggunakan parameter, edit sumber data dan terapkan ke jalur yang Anda gunakan untuk mengimpor data. Untuk informasi selengkapnya, lihat Menggunakan Kembali Alur Data untuk Kumpulan Data yang Berbeda.

Impor data dari Athena

Gunakan Amazon Athena untuk mengimpor data Anda dari Amazon Simple Storage Service (Amazon S3) ke Data Wrangler. Di Athena, Anda menulis kueri SQL standar untuk memilih data yang Anda impor dari Amazon S3. Untuk informasi lebih lanjut, lihat Apa itu Amazon Athena?

Anda dapat menggunakan AWS Management Console untuk mengatur Amazon Athena. Anda harus membuat setidaknya satu basis data di Athena sebelum Anda mulai menjalankan kueri. Untuk informasi selengkapnya tentang memulai dengan Athena, lihat Memulai.

Athena terintegrasi langsung dengan Data Wrangler. Anda dapat menulis kueri Athena tanpa harus meninggalkan UI Data Wrangler.

Selain menulis kueri Athena sederhana di Data Wrangler, Anda juga dapat menggunakan:

- Kelompok kerja Athena untuk manajemen hasil kueri. Untuk informasi selengkapnya tentang grup kerja, lihatMengelola hasil kueri.
- Konfigurasi siklus hidup untuk menyetel periode retensi data. Untuk informasi selengkapnya tentang penyimpanan data, lihatMengatur periode retensi data.

Pertanyaan Athena dalam Data Wrangler

> **Note**
>
> Data Wrangler tidak mendukung kueri federasi.

Jika Anda menggunakan AWS Lake Formation Athena, pastikan izin IAM Lake Formation Anda tidak mengganti izin IAM untuk database. `sagemaker_data_wrangler`

Data Wrangler memberi Anda kemampuan untuk mengimpor seluruh kumpulan data atau mengambil sampel sebagian darinya. Untuk Athena, ini menyediakan opsi pengambilan sampel berikut:

- Tidak ada - Impor seluruh set data.
- K Pertama - Contoh baris K pertama dari kumpulan data, di mana K adalah bilangan bulat yang Anda tentukan.
- Acak - Mengambil sampel acak dari ukuran yang Anda tentukan.
- Bertingkat — Mengambil sampel acak bertingkat. Sampel bertingkat mempertahankan rasio nilai dalam kolom.

Prosedur berikut menunjukkan cara mengimpor dataset dari Athena ke Data Wrangler.

Untuk mengimpor dataset ke Data Wrangler dari Athena

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Pilih Impor data.
9. Di bawah Tersedia, pilih Amazon Athena.
10. Untuk Katalog Data, pilih katalog data.
11. Gunakan daftar dropdown Database untuk memilih database yang ingin Anda kueri. Saat memilih database, Anda dapat melihat pratinjau semua tabel dalam database menggunakan Tabel yang tercantum di bawah Detail.
12. (Opsional) Pilih Konfigurasi lanjutan.
    a. Pilih Workgroup.
    b. Jika grup kerja Anda belum menerapkan lokasi keluaran Amazon S3 atau jika Anda tidak menggunakan grup kerja, tentukan nilai untuk lokasi hasil kueri Amazon S3.
    c. (Opsional) Untuk periode penyimpanan data, pilih kotak centang untuk mengatur periode penyimpanan data dan tentukan jumlah hari untuk menyimpan data sebelum dihapus.
    d. (Opsional) Secara default, Data Wrangler menyimpan koneksi. Anda dapat memilih untuk membatalkan pilihan kotak centang dan tidak menyimpan koneksi.
13. Untuk Sampling, pilih metode pengambilan sampel. Pilih Tidak Ada untuk mematikan pengambilan sampel.
14. Masukkan kueri Anda di editor kueri dan gunakan tombol Jalankan untuk menjalankan kueri. Setelah kueri berhasil, Anda dapat melihat pratinjau hasil Anda di bawah editor.

> **Note**
>
> Data Salesforce menggunakan tipe. `timestamptz` Jika Anda menanyakan kolom stempel waktu yang telah Anda impor ke Athena dari Salesforce, transmisikan data di

kolom ke jenisnya. `timestamp` Kueri berikut melemparkan kolom stempel waktu ke jenis yang benar.

```
# cast column timestamptz_col as timestamp type, and name it as
 timestamp_col
select cast(timestamptz_col as timestamp) as timestamp_col from table
```

15. Untuk mengimpor hasil kueri, pilih Impor.

Setelah Anda menyelesaikan prosedur sebelumnya, kumpulan data yang Anda kueri dan impor akan muncul di alur Data Wrangler.

Secara default, Data Wrangler menyimpan pengaturan koneksi sebagai koneksi baru. Saat Anda mengimpor data, kueri yang telah Anda tentukan muncul sebagai koneksi baru. Koneksi tersimpan menyimpan informasi tentang workgroup Athena dan bucket Amazon S3 yang Anda gunakan. Saat Anda menghubungkan ke sumber data lagi, Anda dapat memilih koneksi yang disimpan.

Mengelola hasil kueri

Data Wrangler mendukung penggunaan workgroup Athena untuk mengelola hasil kueri dalam akun. AWS Anda dapat menentukan lokasi output Amazon S3 untuk setiap workgroup. Anda juga dapat menentukan apakah output kueri dapat masuk ke lokasi Amazon S3 yang berbeda. Untuk informasi selengkapnya, lihat Menggunakan Grup kerja untuk mengontrol akses kueri akses dan biaya.

Workgroup Anda mungkin dikonfigurasi untuk menerapkan lokasi keluaran kueri Amazon S3. Anda tidak dapat mengubah lokasi keluaran hasil kueri untuk kelompok kerja tersebut.

Jika Anda tidak menggunakan grup kerja atau menentukan lokasi keluaran untuk kueri, Data Wrangler menggunakan bucket Amazon S3 default di AWS Wilayah yang sama tempat instance Studio Classic Anda berada untuk menyimpan hasil kueri Athena. Ini membuat tabel sementara dalam database ini untuk memindahkan output kueri ke bucket Amazon S3 ini. Ini menghapus tabel ini setelah data telah diimpor; Namun database,`sagemaker_data_wrangler`, tetap ada. Untuk mempelajari selengkapnya, lihat Penyimpanan Data yang Diimpor.

Untuk menggunakan workgroup Athena, siapkan kebijakan IAM yang memberikan akses ke workgroup. Jika Anda menggunakan a`SageMaker-Execution-Role`, sebaiknya tambahkan kebijakan ke peran tersebut. Untuk informasi selengkapnya tentang kebijakan IAM untuk grup kerja,

lihat kebijakan IAM untuk mengakses grup kerja. Misalnya kebijakan grup kerja, lihat Kebijakan contoh Workgroup.

Mengatur periode retensi data

Data Wrangler secara otomatis menetapkan periode retensi data untuk hasil kueri. Hasilnya akan dihapus setelah periode retensi. Misalnya, periode retensi default adalah lima hari. Hasil kueri dihapus setelah lima hari. Konfigurasi ini dirancang untuk membantu Anda membersihkan data yang tidak lagi Anda gunakan. Membersihkan data Anda mencegah pengguna yang tidak sah mendapatkan akses. Ini juga membantu mengontrol biaya penyimpanan data Anda di Amazon S3.

Jika Anda tidak menetapkan periode retensi, konfigurasi siklus hidup Amazon S3 menentukan durasi penyimpanan objek. Kebijakan penyimpanan data yang telah ditentukan untuk konfigurasi siklus hidup menghapus hasil kueri yang lebih lama dari konfigurasi siklus hidup yang telah ditentukan. Untuk informasi selengkapnya, lihat Mengatur konfigurasi siklus aktif pada bucket.

Data Wrangler menggunakan konfigurasi siklus hidup Amazon S3 untuk mengelola retensi dan kedaluwarsa data. Anda harus memberikan izin peran eksekusi Amazon SageMaker Studio Classic IAM untuk mengelola konfigurasi siklus hidup bucket. Gunakan prosedur berikut untuk memberikan izin.

Untuk memberikan izin untuk mengelola konfigurasi siklus hidup lakukan hal berikut.

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pilih Peran.
3. Di bilah pencarian, tentukan peran SageMaker eksekusi Amazon yang digunakan Amazon SageMaker Studio Classic.
4. Pilih peran.
5. Pilih Tambahkan izin.
6. Pilih Buat kebijakan sebaris.
7. Untuk Layanan, tentukan S3 dan pilih.
8. Di bawah bagian Baca, pilih GetLifecycleConfiguration.
9. Di bawah bagian Tulis, pilih PutLifecycleConfiguration.
10. Untuk Sumber daya, pilih Spesifik.
11. Untuk Tindakan, pilih ikon panah di sebelah Manajemen izin.
12. Pilih PutResourcePolicy.

13. Untuk Sumber daya, pilih Spesifik.
14. Pilih kotak centang di sebelah Apa saja di akun ini.
15. Pilih Tinjau kebijakan.
16. Untuk Nama, tentukan nama.
17. Pilih Buat kebijakan.

Pengimporan data dari Amazon Redshift

Amazon Redshift adalah layanan gudang data dengan skala petabyte yang terkelola penuh di cloud. Langkah pertama untuk membuat gudang data adalah meluncurkan satu set node, yang disebut cluster Amazon Redshift. Setelah menyediakan klaster, Anda dapat mengunggah kumpulan data dan kemudian melakukan kueri analisis data.

Anda dapat terhubung ke dan menanyakan satu atau beberapa klaster Amazon Redshift di Data Wrangler. Untuk menggunakan opsi impor ini, Anda harus membuat setidaknya satu cluster di Amazon Redshift. Untuk mempelajari caranya, lihat Memulai Amazon Redshift.

Anda dapat menampilkan hasil kueri Amazon Redshift Anda di salah satu lokasi berikut:

- Bucket Amazon S3 default
- Lokasi output Amazon S3 yang Anda tentukan

Anda dapat mengimpor seluruh kumpulan data atau mengambil sampel sebagian darinya. Untuk Amazon Redshift, ini menyediakan opsi pengambilan sampel berikut:

- Tidak ada - Impor seluruh set data.
- K Pertama - Contoh baris K pertama dari kumpulan data, di mana K adalah bilangan bulat yang Anda tentukan.
- Acak - Mengambil sampel acak dari ukuran yang Anda tentukan.
- Bertingkat — Mengambil sampel acak bertingkat. Sampel bertingkat mempertahankan rasio nilai dalam kolom.

Bucket Amazon S3 default berada di AWS Wilayah yang sama tempat instans Studio Classic Anda berada untuk menyimpan hasil kueri Amazon Redshift. Untuk informasi selengkapnya, lihat Penyimpanan Data yang Diimpor.

Untuk bucket Amazon S3 default atau bucket yang Anda tentukan, Anda memiliki opsi enkripsi berikut:

- Enkripsi AWS sisi layanan default dengan kunci yang dikelola Amazon S3 (SSE-S3)
- Kunci AWS Key Management Service (AWS KMS) yang Anda tentukan

AWS KMSKunci adalah kunci enkripsi yang Anda buat dan kelola. Untuk informasi lebih lanjut tentang kunci KMS, lihat AWS Key Management Service.

Anda dapat menentukan AWS KMS kunci menggunakan kunci ARN atau ARN akun Anda. AWS

Jika Anda menggunakan kebijakan terkelola IAM`AmazonSageMakerFullAccess`, untuk memberikan izin peran untuk menggunakan Data Wrangler di Studio Classic, nama Pengguna Database Anda harus memiliki awalan. `sagemaker_access`

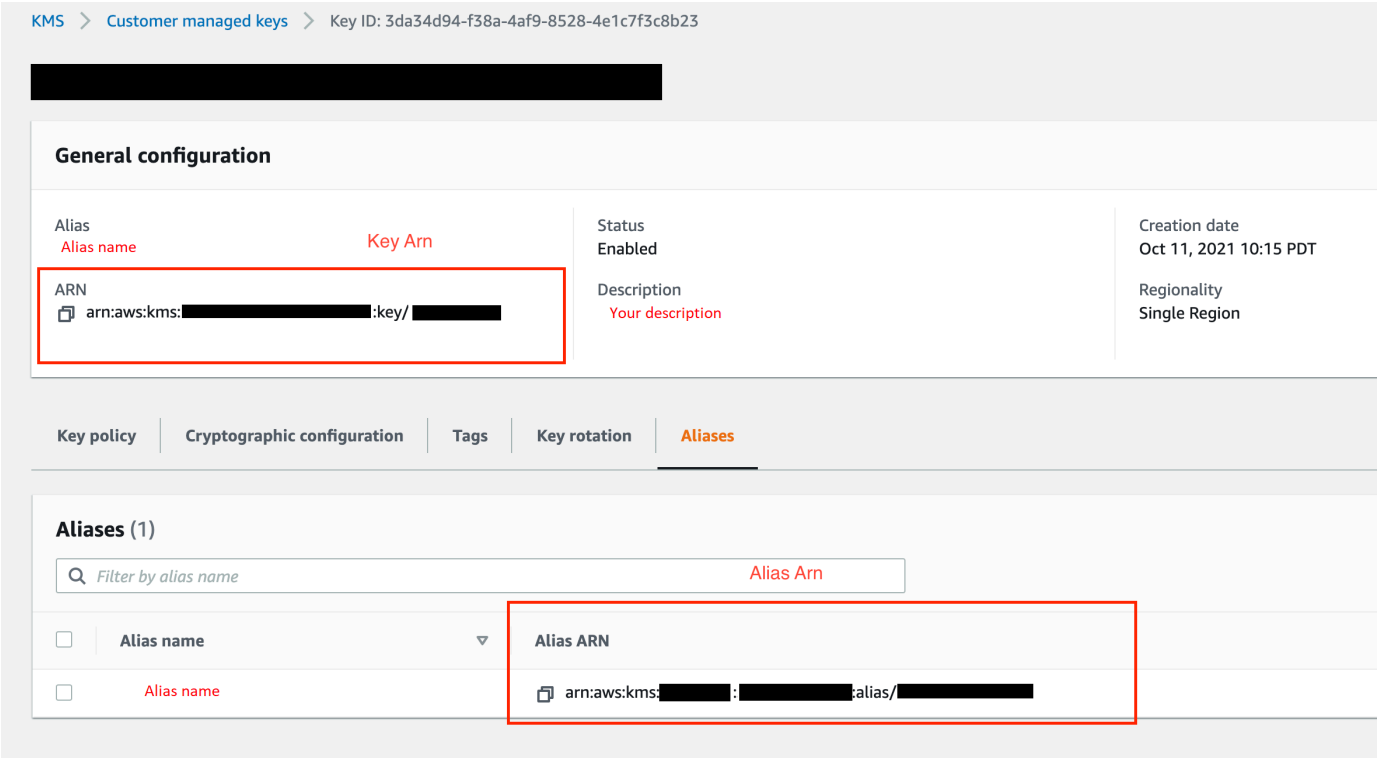
Gunakan prosedur berikut untuk mempelajari cara menambahkan klaster baru.

> **Note**
>
> Data Wrangler menggunakan Amazon Redshift Data API dengan kredensi sementara. Untuk mempelajari lebih lanjut tentang API ini, lihat Menggunakan API Data Amazon Redshift di Panduan Manajemen Pergeseran Merah Amazon.

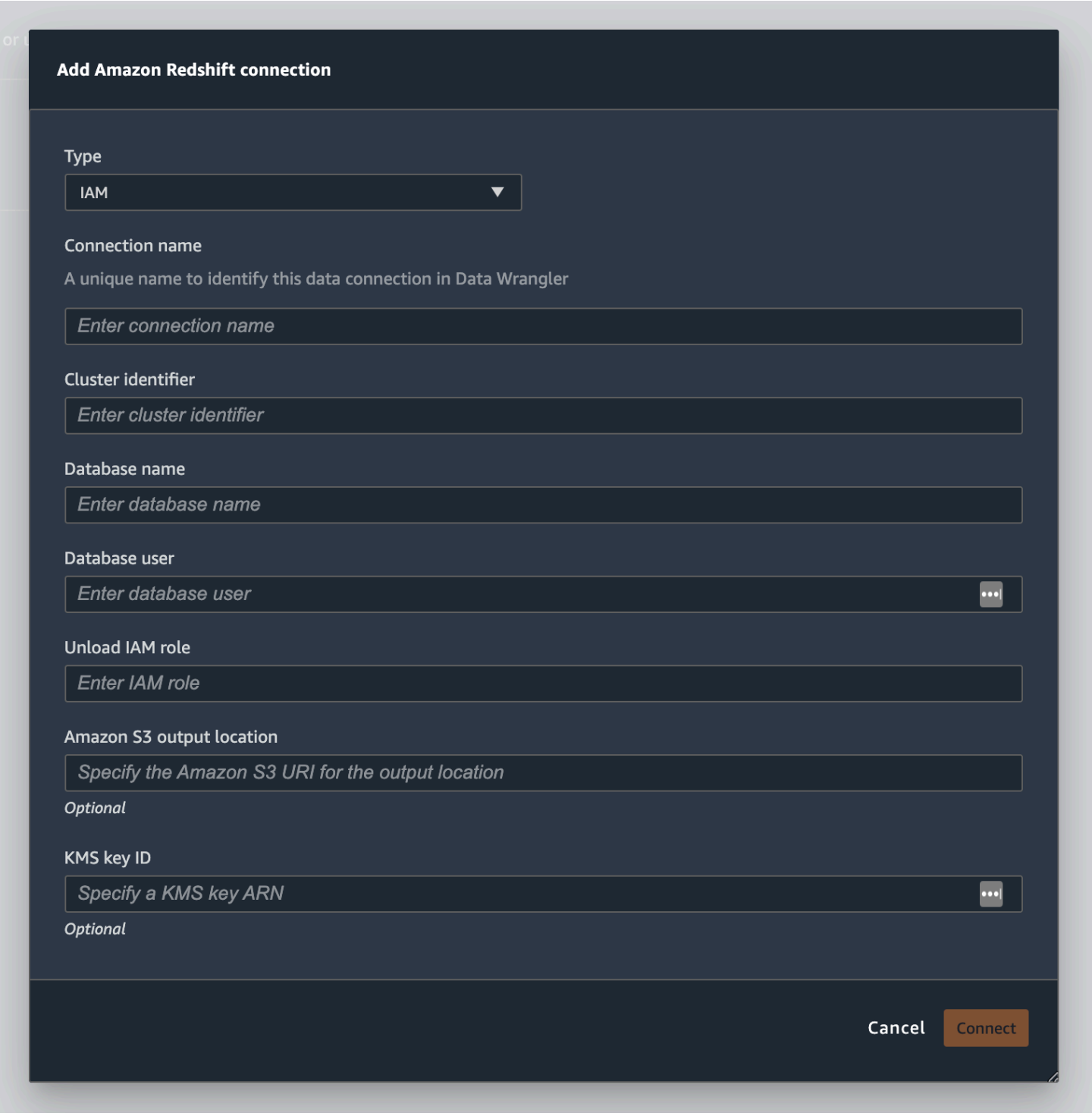
Untuk terhubung ke cluster Amazon Redshift

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Pilih Impor data.
9. Di bawah Tersedia, pilih Amazon Athena.

10. Pilih Amazon Redshift.
11. Pilih Temporary credentials (IAM) untuk Type.
12. Masukkan Nama Koneksi. Ini adalah nama yang digunakan oleh Data Wrangler untuk mengidentifikasi koneksi ini.
13. Masukkan Cluster Identifier untuk menentukan cluster mana yang ingin Anda sambungkan. Catatan: Masukkan hanya pengidentifikasi klaster dan bukan titik akhir penuh klaster Amazon Redshift.
14. Masukkan Nama Basis Data dari basis data yang ingin Anda hubungkan.
15. Masukkan Pengguna Database untuk mengidentifikasi pengguna yang ingin Anda gunakan untuk terhubung ke database.
16. Untuk UNLOAD IAM Role, masukkan ARN peran IAM dari peran yang harus diasumsikan oleh cluster Amazon Redshift untuk memindahkan dan menulis data ke Amazon S3. Untuk informasi selengkapnya tentang peran ini, lihat Mengotorisasi Amazon Redshift untuk mengakses layanan AWS lain atas nama Anda di Panduan Manajemen Amazon Redshift.
17. Pilih Hubungkan.
18. (Opsional) Untuk lokasi keluaran Amazon S3, tentukan URI S3 untuk menyimpan hasil kueri.
19. (Opsional) Untuk ID kunci KMS, tentukan ARN kunci atau AWS KMS alias. Gambar berikut menunjukkan di mana Anda dapat menemukan salah satu kunci diAWS Management Console.



Gambar berikut menunjukkan semua bidang dari prosedur sebelumnya.



Setelah koneksi Anda berhasil dibuat, itu muncul sebagai sumber data di bawah Impor Data. Pilih sumber data ini untuk menanyakan database Anda dan mengimpor data.

Untuk kueri dan impor data dari Amazon Redshift

1. Pilih koneksi yang ingin Anda kueri dari Sumber Data.
2. Pilih Skema. Untuk mempelajari selengkapnya tentang Skema Amazon Redshift, lihat Skema di Panduan Developer Basis Data Amazon Redshift.
3. (Opsional) Di bawah Konfigurasi lanjutan, tentukan metode Sampling yang ingin Anda gunakan.
4. Masukkan kueri Anda di editor kueri dan pilih Jalankan untuk menjalankan kueri. Setelah kueri berhasil, Anda dapat melihat pratinjau hasil Anda di bawah editor.
5. Pilih Impor dataset untuk mengimpor dataset yang telah ditanyakan.
6. Masukkan nama Dataset. Jika Anda menambahkan nama Dataset yang berisi spasi, spasi ini akan diganti dengan garis bawah saat dataset Anda diimpor.
7. Pilih Tambahkan.

Untuk mengedit set data, lakukan hal berikut.

1. Arahkan ke alur Data Wrangler Anda.
2. Pilih + di sebelah Sumber - Sampel.
3. Ubah data yang Anda impor.
4. Pilih Terapkan

Pengimporan data dari Amazon EMR

Anda dapat menggunakan Amazon EMR sebagai sumber data untuk aliran Amazon SageMaker Data Wrangler Anda. Amazon EMR adalah platform cluster terkelola yang dapat Anda gunakan untuk memproses dan menganalisis data dalam jumlah besar. Untuk informasi selengkapnya tentang Amazon EMR, lihat Apa itu Amazon EMR? . Untuk mengimpor dataset dari EMR, Anda menghubungkannya dan menanyakannya.

> **Important**
>
> Anda harus memenuhi prasyarat berikut untuk terhubung ke cluster EMR Amazon:

Prasyarat

- Konfigurasi jaringan
  - Anda memiliki VPC Amazon di Wilayah yang Anda gunakan untuk meluncurkan Amazon SageMaker Studio Classic dan Amazon EMR.
  - Baik Amazon EMR dan Amazon SageMaker Studio Classic harus diluncurkan dalam subnet pribadi. Mereka bisa berada di subnet yang sama atau yang lain.
  - Amazon SageMaker Studio Classic harus dalam mode VPC saja.

    Untuk informasi selengkapnya tentang membuat VPC, lihat Membuat VPC.

    Untuk informasi selengkapnya tentang cara membuat VPC, lihat Connect SageMaker Studio Classic Notebook di VPC ke Sumber Daya Eksternal.
  - Klaster Amazon EMR yang Anda jalankan harus berada dalam VPC Amazon yang sama.
  - Cluster EMR Amazon dan VPC Amazon harus berada di akun yang sama. AWS
  - Cluster EMR Amazon Anda menjalankan Hive atau Presto.
    - Kluster sarang harus mengizinkan lalu lintas masuk dari grup keamanan Studio Classic di port 10000.
    - Cluster Presto harus mengizinkan lalu lintas masuk dari grup keamanan Studio Classic di port 8889.

      > **Note**
      >
      > Nomor port berbeda untuk cluster EMR Amazon yang menggunakan peran IAM. Arahkan ke bagian prasyarat untuk informasi lebih lanjut.
- SageMaker Studio Klasik
  - Amazon SageMaker Studio Classic harus menjalankan Jupyter Lab Versi 3. Untuk informasi tentang memperbarui Versi Lab Jupyter, lihat. Lihat dan perbarui JupyterLab versi aplikasi dari konsol
  - Amazon SageMaker Studio Classic memiliki peran IAM yang mengontrol akses pengguna. Peran IAM default yang Anda gunakan untuk menjalankan Amazon SageMaker Studio Classic tidak memiliki kebijakan yang dapat memberi Anda akses ke kluster EMR Amazon. Anda harus melampirkan kebijakan yang memberikan izin ke

peran IAM. Untuk informasi selengkapnya, lihat Konfigurasikan kemampuan ditemukan kluster EMR Amazon (untuk administrator).

- IAM role juga harus memiliki kebijakan berikut. `secretsmanager:PutResourcePolicy`
- Jika Anda menggunakan domain Studio Classic yang telah Anda buat, pastikan domain tersebut dalam `AppNetworkAccessType` mode khusus VPC. Untuk informasi tentang memperbarui domain agar menggunakan mode khusus VPC, lihat. Matikan dan Perbarui SageMaker Studio Classic

- Kluster Amazon EMR
  - Anda harus menginstal Hive atau Presto di klaster Anda.
  - Amazon Amazon Amazon Amazon Amazon Amazon EMR versi 5.5.0 atau yang lebih baru.

    > **Note**
    >
    > Amazon EMR mendukung penghentian otomatis. Penghentian otomatis menghentikan cluster idle agar tidak berjalan dan mencegah Anda mengeluarkan biaya. Berikut ini adalah rilis yang mendukung penghentian otomatis:
    >
    > - Untuk rilis 6.x, versi 6.1.0 atau yang lebih baru.
    > - Untuk rilis 5.x, versi 5.30.0 atau yang lebih baru.

- Cluster EMR Amazon menggunakan peran runtime IAM
  - Gunakan halaman berikut untuk menyiapkan peran runtime IAM untuk klaster Amazon EMR. Anda harus mengaktifkan enkripsi in-transit saat menggunakan peran runtime:
    - Prasyarat untuk meluncurkan klaster Amazon EMR dengan peran runtime
    - Meluncurkan klaster Amazon EMR dengan kontrol akses berbasis peran
  - Anda harus Lake Formation sebagai alat tata kelola untuk data dalam database Anda. Anda juga harus menggunakan pemfilteran data eksternal untuk kontrol akses.
    - Untuk informasi lebih lanjut tentang Lake Formation, lihat Apa ituAWS Lake Formation?
    - Untuk informasi selengkapnya tentang mengintegrasikan Lake Formation ke Amazon EMR, lihat Mengintegrasikan layanan pihak ketiga dengan Lake Formation.
  - Versi klaster Anda harus berupa 6.9.0 atau yang lebih baru.

- Akses keAWS Secrets Manager. Untuk informasi selengkapnya tentang Secrets Manager lihat Apa ituAWS Secrets Manager?
- Kluster sarang harus mengizinkan lalu lintas masuk dari grup keamanan Studio Classic di port 10000.

Amazon VPC adalah jaringan virtual yang secara logis terisolasi dari jaringan lain di cloud. AWS Amazon SageMaker Studio Classic dan kluster EMR Amazon Anda hanya ada di dalam VPC Amazon.

Gunakan prosedur berikut untuk meluncurkan Amazon SageMaker Studio Classic di Amazon VPC.

Untuk meluncurkan Studio Classic dalam VPC, lakukan hal berikut.

1. Arahkan ke SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Pilih Launch SageMaker Studio Classic.
3. Pilih Pengaturan standar.
4. Untuk peran eksekusi default, pilih peran IAM untuk menyiapkan Studio Classic.
5. Pilih VPC tempat Anda meluncurkan kluster EMR Amazon.
6. Untuk Subnet, pilih subnet pribadi.
7. Untuk grup Keamanan, tentukan grup keamanan yang Anda gunakan untuk mengontrol antara VPC Anda.
8. Pilih VPC Saja.
9. (Opsional) AWS menggunakan kunci enkripsi default. Anda dapat menentukan AWS Key Management Service kunci untuk mengenkripsi data Anda.
10. Pilih Berikutnya.
11. Pada pengaturan Studio, pilih konfigurasi yang paling cocok untuk Anda.
12. Pilih Berikutnya untuk melewati pengaturan SageMaker Canvas.
13. Pilih Berikutnya untuk melewati pengaturan RStudio.

Jika Anda belum menyiapkan klaster Amazon EMR, Anda dapat menggunakan prosedur berikut untuk membuatnya. Untuk informasi selengkapnya tentang Amazon EMR, lihat Apa itu Amazon EMR?

Untuk membuat klaster baru, lakukan hal berikut.

1. Navigasikan keAWS Management Console.
2. Di bilah pencarian, tentukan**`Amazon EMR`**.
3. Pilih Buat klaster.
4. Untuk Nama klaster, tentukan nama klaster Anda.
5. Untuk Rilis, pilih versi rilis cluster.

> **Note**
>
> Amazon EMR mendukung penghentian otomatis untuk rilis berikut:
>
> - Untuk rilis 6.x, rilis 6.1.0 atau yang lebih baru
> - Untuk rilis 5.x, rilis 5.30.0 atau yang lebih baru
>
> Penghentian otomatis menghentikan cluster idle agar tidak berjalan dan mencegah Anda mengeluarkan biaya.

6. (Opsional) Untuk Aplikasi, pilih Presto.
7. Pilih aplikasi yang Anda jalankan di cluster.
8. Di bawah Jaringan, untuk konfigurasi Perangkat Keras, tentukan pengaturan konfigurasi perangkat keras.

> **Important**
>
> Untuk Networking, pilih VPC yang menjalankan Amazon SageMaker Studio Classic dan pilih subnet pribadi.

9. Di bawah Keamanan dan akses, tentukan pengaturan keamanan.
10. Pilih Buat.

Untuk tutorial tentang membuat klaster EMR Amazon, lihat Memulai Amazon EMR. Untuk informasi tentang praktik terbaik untuk mengonfigurasi klaster, lihat Pertimbangan dan praktik terbaik.

> **Note**
>
> Untuk praktik terbaik keamanan, Data Wrangler hanya dapat terhubung ke VPC pada subnet pribadi. Anda tidak dapat terhubung ke node master kecuali Anda menggunakan AWS

Systems Manager untuk instans EMR Amazon Anda. Untuk informasi selengkapnya, lihat Mengamankan akses ke kluster EMR menggunakan. AWS Systems Manager

Saat ini Anda dapat menggunakan metode berikut untuk mengakses klaster Amazon EMR:

- Token Autentikasi
- Protokol Akses Direktori Ringan (LDAP)
- IAM (Peran runtime)

Tidak menggunakan autentikasi atau menggunakan LDAP dapat mengharuskan Anda membuat beberapa cluster dan profil instans Amazon EC2. Jika Anda seorang administrator, Anda mungkin perlu menyediakan grup pengguna dengan tingkat akses yang berbeda ke data. Metode ini dapat menghasilkan overhead administratif yang membuatnya lebih sulit untuk mengelola pengguna Anda.

Sebaiknya gunakan peran runtime IAM yang memberi banyak pengguna kemampuan untuk terhubung ke kluster EMR Amazon yang sama. Peran runtime adalah peran IAM yang dapat Anda tetapkan ke pengguna yang terhubung ke kluster EMR Amazon. Anda dapat mengonfigurasi peran IAM runtime agar memiliki izin yang spesifik untuk setiap grup pengguna.

Gunakan bagian berikut untuk membuat cluster EMR Presto atau Hive Amazon dengan LDAP diaktifkan.

Presto

⚠ Important

Untuk digunakan AWS Glue sebagai metastore untuk tabel Presto, pilih Gunakan metadata tabel Presto untuk menyimpan hasil kueri EMR Amazon Anda dalam AWS Glue katalog data saat meluncurkan klaster EMR. Menyimpan hasil kueri dalam katalog AWS Glue data dapat menyelamatkan Anda dari biaya yang dikenakan.
Untuk menanyakan kumpulan data besar di kluster EMR Amazon, Anda harus menambahkan properti berikut ke file konfigurasi Presto di kluster EMR Amazon Anda:

```
[{"classification":"presto-config","properties":{
"http-server.max-request-header-size":"5MB",
"http-server.max-response-header-size":"5MB"}}]
```

Anda juga dapat mengubah pengaturan konfigurasi saat meluncurkan klaster Amazon EMR.
File konfigurasi untuk kluster EMR Amazon Anda terletak di bawah jalur berikut:. `/etc/presto/conf/config.properties`

Gunakan prosedur berikut untuk membuat klaster Presto dengan LDAP diaktifkan.

Untuk membuat klaster baru, lakukan hal berikut.

1. Navigasikan keAWS Management Console.
2. Di bilah pencarian, tentukan**`Amazon EMR`**.
3. Pilih Buat klaster.
4. Untuk Nama klaster, tentukan nama klaster Anda.
5. Untuk Rilis, pilih versi rilis cluster.

> **Note**
>
> Amazon EMR mendukung penghentian otomatis untuk rilis berikut:
>
> - Untuk rilis 6.x, rilis 6.1.0 atau yang lebih baru
> - Untuk rilis 5.x, rilis 5.30.0 atau yang lebih baru
>
> Penghentian otomatis menghentikan cluster idle agar tidak berjalan dan mencegah Anda mengeluarkan biaya.

6. Pilih aplikasi yang Anda jalankan di cluster.
7. Di bawah Jaringan, untuk konfigurasi Perangkat Keras, tentukan pengaturan konfigurasi perangkat keras.

> **Important**
>
> Untuk Networking, pilih VPC yang menjalankan Amazon SageMaker Studio Classic dan pilih subnet pribadi.

8. Di bawah Keamanan dan akses, tentukan pengaturan keamanan.
9. Pilih Buat.

Hive

> **Important**
>
> Untuk digunakan AWS Glue sebagai metastore untuk tabel Hive, pilih Gunakan metadata tabel Hive untuk menyimpan hasil kueri EMR Amazon Anda dalam AWS Glue katalog data saat meluncurkan klaster EMR. Menyimpan hasil kueri dalam katalog AWS Glue data dapat menyelamatkan Anda dari biaya yang dikenakan.
> Untuk dapat menanyakan kumpulan data besar di kluster EMR Amazon, tambahkan properti berikut ke file konfigurasi Hive di kluster EMR Amazon Anda:
>
> ```
> [{"classification":"hive-site", "properties"
> :{"hive.resultset.use.unique.column.names":"false"}}]
> ```
>
> Anda juga dapat mengubah pengaturan konfigurasi saat meluncurkan klaster Amazon EMR.
> File konfigurasi untuk kluster EMR Amazon Anda terletak di bawah jalur berikut:. `/etc/hive/conf/hive-site.xml` Anda dapat menentukan properti berikut dan memulai kembali klaster:
>
> ```
> <property>
>     <name>hive.resultset.use.unique.column.names</name>
>     <value>false</value>
> </property>
> ```

Gunakan prosedur berikut untuk membuat klaster Hive dengan LDAP diaktifkan.

Untuk membuat cluster Hive dengan LDAP diaktifkan, lakukan hal berikut.

1. Navigasikan keAWS Management Console.

2. Di bilah pencarian, tentukan**`Amazon EMR`**.
3. Pilih Buat klaster.
4. Pilih Pergi ke opsi lanjutan.
5. Untuk Rilis, pilih versi rilis Amazon EMR.
6. Opsi konfigurasi Hive dipilih secara default. Pastikan opsi Hive memiliki kotak centang di sebelahnya.
7. (Opsional) Anda juga dapat memilih Presto sebagai opsi konfigurasi untuk mengaktifkan Hive dan Presto di cluster Anda.
8. (Opsional) Pilih Gunakan untuk metadata tabel Hive untuk menyimpan hasil kueri EMR Amazon Anda dalam katalog data. AWS Glue Menyimpan hasil kueri dalam AWS Glue katalog dapat menyelamatkan Anda dari biaya yang dikenakan. Untuk informasi lebih lanjut, lihat Menggunakan Katalog AWS Glue Data sebagai metastore untuk Hive.

> **Note**
>
> Menyimpan hasil kueri dalam katalog data memerlukan Amazon EMR versi 5.8.0 atau yang lebih baru.

9. Di bawah Enter konfigurasi, tentukan JSON berikut:

```
[
  {
    "classification": "hive-site",
    "properties": {
      "hive.server2.authentication.ldap.baseDN": "dc=example,dc=org",
      "hive.server2.authentication": "LDAP",
      "hive.server2.authentication.ldap.url": "ldap://ldap-server-dns-name:389"
    }
  }
]
```

> **Note**
>
> Sebagai praktik keamanan terbaik, sebaiknya aktifkan SSL HiveServer dengan menambahkan beberapa properti di JSON situs sarang sebelumnya. Untuk informasi lebih lanjut, lihat Aktifkan SSL di HiveServer 2.

10. Tentukan pengaturan cluster yang tersisa dan buat cluster.

Gunakan bagian berikut untuk menggunakan otentikasi LDAP untuk klaster EMR Amazon yang telah Anda buat.

LDAP for Presto

Menggunakan LDAP pada cluster yang menjalankan Presto memerlukan akses ke koordinator Presto melalui HTTPS. Lakukan hal berikut untuk menyediakan akses:

- Aktifkan akses pada port 636
- Aktifkan SSL untuk koordinator Presto

Gunakan template berikut untuk mengkonfigurasi Presto:

```
- Classification: presto-config
    ConfigurationProperties:
       http-server.authentication.type: 'PASSWORD'
       http-server.https.enabled: 'true'
       http-server.https.port: '8889'
       http-server.http.port: '8899'
       node-scheduler.include-coordinator: 'true'
       http-server.https.keystore.path: '/path/to/keystore/path/for/presto'
       http-server.https.keystore.key: 'keystore-key-password'
       discovery.uri: 'http://master-node-dns-name:8899'
- Classification: presto-password-authenticator
    ConfigurationProperties:
       password-authenticator.name: 'ldap'
       ldap.url: !Sub 'ldaps://ldap-server-dns-name:636'
       ldap.user-bind-pattern: "uid=${USER},dc=example,dc=org"
       internal-communication.authentication.ldap.user: "ldap-user-name"
       internal-communication.authentication.ldap.password: "ldap-password"
```

Untuk informasi tentang pengaturan LDAP di Presto, lihat sumber daya berikut ini:

- Otentikasi LDAP
- Menggunakan Otentikasi LDAP untuk Presto di Amazon EMR

> **Note**
>
> Sebagai praktik keamanan terbaik, kami sarankan mengaktifkan SSL untuk Presto. Untuk informasi lebih lanjut, lihat Komunikasi Internal Aman.

LDAP for Hive

Untuk menggunakan LDAP for Hive untuk klaster yang telah Anda buat, gunakan prosedur berikut Mengkonfigurasi ulang grup instans di konsol.

Anda menentukan nama cluster yang Anda hubungkan.

```
[
  {
    "classification": "hive-site",
    "properties": {
      "hive.server2.authentication.ldap.baseDN": "dc=example,dc=org",
      "hive.server2.authentication": "LDAP",
      "hive.server2.authentication.ldap.url": "ldap://ldap-server-dns-name:389"
    }
  }
]
```

Gunakan prosedur berikut untuk mengimpor data dari klaster.

Untuk mengimpor data dari klaster, lakukan hal berikut.

1. Buka alur Data Wrangler.
2. Pilih Buat Koneksi.
3. Pilih Amazon EMR.
4. Lakukan salah satu dari berikut ini.
   - (Opsional) Untuk Rahasia ARN, tentukan Amazon Resource Number (ARN) database dalam klaster. Rahasia memberikan keamanan tambahan. Untuk informasi lebih lanjut tentang rahasia, lihat Apa ituAWS Secrets Manager? Untuk informasi tentang membuat rahasia klaster Anda, lihatMembuat AWS Secrets Manager rahasia untuk cluster Anda.

> **Important**
>
> Anda harus menentukan rahasia jika Anda menggunakan peran runtime IAM untuk otentikasi.

- Dari tabel dropdown, pilih cluster.

5. Pilih Berikutnya.
6. Untuk Pilih titik akhir untuk ***example-cluster-name***cluster, pilih mesin kueri.
7. (Opsional) Pilih Simpan koneksi.
8. Pilih Berikutnya, pilih login dan pilih salah satu dari berikut ini:
   - Token Autentikasi
   - LDAP
   - IAM
9. Untuk Login ke ***example-cluster-name***cluster, tentukan Username dan Password untuk cluster.
10. Pilih Hubungkan.
11. Dalam editor kueri, tentukan kueri SQL.
12. Pilih Jalankan.
13. Pilih Impor.

Membuat AWS Secrets Manager rahasia untuk cluster Anda

Jika Anda menggunakan peran runtime IAM untuk mengakses klaster EMR Amazon Anda, Anda harus menyimpan kredensyal yang Anda gunakan untuk mengakses Amazon EMR sebagai rahasia Secrets Manager. Anda menyimpan semua kredensi yang Anda gunakan untuk mengakses cluster dalam rahasia.

Anda harus menyimpan informasi berikut secara rahasia:

- Titik akhir JDBC — `jdbc:hive2://`
- Nama DNS — Nama DNS klaster Amazon EMR Anda. Ini adalah titik akhir untuk node utama atau nama host.
- Pelabuhan — `8446`

Anda juga dapat menyimpan informasi tambahan berikut di rahasia:

- Peran IAM — Peran IAM yang Anda gunakan untuk mengakses cluster. Data Wrangler menggunakan peran SageMaker eksekusi Anda secara default.
- Jalur Truststore - Secara default, Data Wrangler membuat jalur truststore untuk Anda. Anda juga dapat menggunakan jalur truststore Anda sendiri. Untuk informasi selengkapnya tentang jalur truststore, lihat Enkripsi dalam transit di 2. HiveServer
- Kata sandi Truststore - Secara default, Data Wrangler membuat kata sandi truststore untuk Anda. Anda juga dapat menggunakan jalur truststore Anda sendiri. Untuk informasi selengkapnya tentang jalur truststore, lihat Enkripsi dalam transit di 2. HiveServer

Gunakan prosedur berikut untuk menyimpan kredensi dalam rahasia Secrets Manager.

Untuk menyimpan kredensyal Anda sebagai rahasia, lakukan hal berikut.

1. Navigasikan keAWS Management Console.
2. Di bilah pencarian, tentukan Secrets Manager.
3. Pilih AWS Secrets Manager.
4. Pilih Simpan rahasia baru.
5. Untuk Tipe rahasia, pilih Tipe rahasia lainnya.
6. Di bawah pasangan kunci/nilai, pilih Plaintext.
7. Untuk cluster yang menjalankan Hive, Anda dapat menggunakan template berikut untuk otentikasi IAM.

```
{"jdbcURL": ""
 "iam_auth": {"endpoint": "jdbc:hive2://", #required
               "dns": "ip-xx-x-xxx-xxx.ec2.internal", #required
               "port": "10000", #required
             "cluster_id": "j-xxxxxxxxx", #required
             "iam_role": "arn:aws:iam::xxxxxxxx:role/xxxxxxxxxxxx", #optional
             "truststore_path": "/etc/alternatives/jre/lib/security/cacerts",
 #optional
             "truststore_password": "changeit" #optional

             }}
```

> **Note**
>
> Setelah mengimpor data, Anda menerapkan transformasi ke data tersebut. Anda kemudian mengekspor data yang telah Anda ubah ke lokasi tertentu. Jika Anda menggunakan notebook Jupyter untuk mengekspor data yang diubah ke Amazon S3, Anda harus menggunakan jalur truststore yang ditentukan dalam contoh sebelumnya.

Rahasia Secrets Manager menyimpan URL JDBC dari cluster Amazon EMR sebagai rahasia. Menggunakan rahasia lebih aman daripada langsung memasukkan kredensil Anda.

Gunakan prosedur berikut untuk menyimpan URL JDBC sebagai rahasia.

Untuk menyimpan URL JDBC sebagai rahasia, lakukan hal berikut.

1. Navigasikan keAWS Management Console.
2. Di bilah pencarian, tentukan Secrets Manager.
3. Pilih AWS Secrets Manager.
4. Pilih Simpan rahasia baru.
5. Untuk Tipe rahasia, pilih Tipe rahasia lainnya.
6. Untuk pasangan kunci/nilai, tentukan `jdbcURL` sebagai kunci dan URL JDBC yang valid sebagai nilainya.

   Format URL JDBC yang valid tergantung pada apakah Anda menggunakan otentikasi dan apakah Anda menggunakan Hive atau Presto sebagai mesin kueri. Daftar berikut menunjukkan format URL JBDC yang valid untuk berbagai konfigurasi yang mungkin.

   - Sarang, tidak ada otentikasi - `jdbc:hive2://`*`emr-cluster-master-public`*`-dns:10000/;`
   - Hive, otentikasi LDAP - `jdbc:hive2://`*`emr-cluster-master-public-dns-name`*`:10000/;AuthMech=3;UID=david;PWD=welcome123;`
   - Untuk Hive dengan SSL diaktifkan, format URL JDBC tergantung pada apakah Anda menggunakan File Keystore Java untuk konfigurasi TLS. File Keystore Java membantu memverifikasi identitas simpul utama klaster Amazon EMR. Untuk menggunakan File Keystore Java, buat di cluster EMR dan unggah ke Data Wrangler. Untuk menghasilkan file, gunakan perintah berikut di cluster EMR Amazon,. `keytool -genkey -alias hive -keyalg RSA`

`-keysize 1024 -keystore hive.jks` Untuk informasi tentang menjalankan perintah di klaster EMR Amazon, lihat Mengamankan akses ke kluster EMR menggunakan. AWS Systems Manager Untuk mengunggah file, pilih panah ke atas pada navigasi sebelah kiri UI Data Wrangler.

Berikut ini adalah format URL JDBC yang valid untuk Hive dengan SSL diaktifkan:

- Tanpa File Keystore Java - `jdbc:hive2://`*`emr-cluster-master-public-dns`*`:10000/;AuthMech=3;UID=`*`user-name`*`;PWD=`*`password`*`;SSL=1;AllowSelfSignedCerts=1;`
- Dengan File Keystore Java - `jdbc:hive2://`*`emr-cluster-master-public-dns`*`:10000/;AuthMech=3;UID=`*`user-name`*`;PWD=`*`password`*`;SSL=1;SSLKeyStore=/home/sagemaker-user/data/`*`Java-keystore-file-name`*`;SSLKeyStorePwd=`*`Java-keystore-file-passsword`*`;`

- *`Presto, tidak ada otentikasi - jdbc:presto://-dns:8889/; emr-cluster-master-public`*
- Untuk Presto dengan otentikasi LDAP dan SSL diaktifkan, format URL JDBC tergantung pada apakah Anda menggunakan File Keystore Java untuk konfigurasi TLS. File Keystore Java membantu memverifikasi identitas simpul utama klaster Amazon EMR. Untuk menggunakan File Keystore Java, buat di cluster EMR dan unggah ke Data Wrangler. Untuk mengunggah file, pilih panah ke atas pada navigasi sebelah kiri UI Data Wrangler. Untuk informasi tentang membuat File Keystore Java untuk Presto, lihat File Keystore Java untuk TLS. Untuk informasi tentang menjalankan perintah di klaster EMR Amazon, lihat Mengamankan akses ke kluster EMR menggunakan. AWS Systems Manager
  - Tanpa File Keystore Java - `jdbc:presto://`*`emr-cluster-master-public-dns`*`:8889/;SSL=1;AuthenticationType=LDAP Authentication;UID=`*`user-name`*`;PWD=`*`password`*`;AllowSelfSignedServerCert=1;AllowHostNameCNMismatch=1;`
  - Dengan File Keystore Java - `jdbc:presto://`*`emr-cluster-master-public-dns`*`:8889/;SSL=1;AuthenticationType=LDAP Authentication;SSLTrustStorePath=/home/sagemaker-user/data/`*`Java-keystore-file-name`*`;SSLTrustStorePwd=`*`Java-keystore-file-passsword`*`;UID=`*`user-name`*`;PWD=`*`password`*`;`

Selama proses mengimpor data dari kluster EMR Amazon, Anda mungkin mengalami masalah. Untuk informasi tentang pemecahan masalah, lihat. Pemecahan masalah dengan Amazon EMR

Impor data dari Databricks (JDBC)

Anda dapat menggunakan Databricks sebagai sumber data untuk alur Amazon SageMaker Data Wrangler Anda. Untuk mengimpor dataset dari Databricks, gunakan fungsi impor JDBC (Java Database Connectivity) untuk mengakses database Databricks Anda. Setelah Anda mengakses database, tentukan kueri SQL untuk mendapatkan data dan mengimpornya.

Kami berasumsi bahwa Anda memiliki cluster Databricks yang sedang berjalan dan Anda telah mengonfigurasi driver JDBC Anda untuk itu. Untuk informasi lebih lanjut, lihat halaman dokumentasi Databricks berikut ini:

- Pengemudi JDBC
- Konfigurasi JDBC dan parameter koneksi
- Parameter otentikasi

Data Wrangler menyimpan URL JDBC Anda di. AWS Secrets Manager Anda harus memberikan izin peran eksekusi Amazon SageMaker Studio Classic IAM untuk menggunakan Secrets Manager. Gunakan prosedur berikut untuk memberikan izin.

Untuk memberikan izin kepada Secrets Manager, lakukan hal berikut.

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pilih Peran.
3. Di bilah pencarian, tentukan peran SageMaker eksekusi Amazon yang digunakan Amazon SageMaker Studio Classic.
4. Pilih peran.
5. Pilih Tambahkan izin.
6. Pilih Buat kebijakan sebaris.
7. Untuk Layanan, tentukan Secrets Manager dan pilih.
8. Untuk Tindakan, pilih ikon panah di sebelah Manajemen izin.
9. Pilih PutResourcePolicy.
10. Untuk Sumber daya, pilih Spesifik.
11. Pilih kotak centang di sebelah Apa saja di akun ini.
12. Pilih Tinjau kebijakan.

13. Untuk Nama, tentukan nama.
14. Pilih Buat kebijakan.

Anda dapat menggunakan partisi untuk mengimpor data Anda lebih cepat. Partisi memberikan Data Wrangler kemampuan untuk memproses data secara paralel. Secara default, Data Wrangler menggunakan 2 partisi. Untuk sebagian besar kasus penggunaan, 2 partisi memberi Anda kecepatan pemrosesan data yang hampir optimal.

Jika Anda memilih untuk menentukan lebih dari 2 partisi, Anda juga dapat menentukan kolom untuk mempartisi data. Jenis nilai di kolom harus numerik atau tanggal.

Sebaiknya gunakan partisi hanya jika Anda memahami struktur data dan cara pengolahannya.

Anda dapat mengimpor seluruh kumpulan data atau mengambil sampel sebagian darinya. Untuk database Databricks, ini menyediakan opsi pengambilan sampel berikut:

- Tidak ada - Impor seluruh set data.
- K Pertama - Contoh baris K pertama dari kumpulan data, di mana K adalah bilangan bulat yang Anda tentukan.
- Acak - Mengambil sampel acak dari ukuran yang Anda tentukan.
- Bertingkat — Mengambil sampel acak bertingkat. Sampel bertingkat mempertahankan rasio nilai dalam kolom.

Gunakan prosedur berikut untuk mengimpor data Anda dari basis data Databricks.

Untuk mengimpor data dari Databricks, lakukan hal berikut.

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Dari tab Impor data alur Data Wrangler Anda, pilih Databricks.
6. Tentukan bidang berikut:
    - Nama Dataset — Nama yang ingin Anda gunakan untuk kumpulan data dalam alur Data Wrangler Anda.
    - Pengemudi — com.simba.spark.jdbc.driver.

- URL JDBC — URL dari database Databricks. Pemformatan URL dapat bervariasi antara instance Databricks. Untuk informasi tentang menemukan URL dan menentukan parameter di dalamnya, lihat konfigurasi JDBC dan parameter koneksi. *Berikut ini adalah contoh bagaimana URL dapat diformat: jdbc:spark://aws-sagemaker-datawrangler.cloud.databricks.com:443/default; transportMode = http; ssl=1; httppath=sql/protocolv1/o/3122619508517275/0909-200301-cut318; =3; UID = token; PWD =. AuthMech personal-access-token*

> **Note**
>
> Anda dapat menentukan ARN rahasia yang berisi URL JDBC alih-alih menentukan URL JDBC itu sendiri. Rahasianya harus berisi pasangan kunci-nilai dengan format berikut:. `jdbcURL:`*JDBC-URL* Untuk informasi lebih lanjut, lihat Apa itu Secrets Manager? .

7. Tentukan pernyataan SQL SELECT.

> **Note**
>
> Data Wrangler tidak mendukung Common Table Expressions (CTE) atau tabel sementara dalam kueri.

8. Untuk Sampling, pilih metode pengambilan sampel.
9. Pilih Jalankan.
10. (Opsional) Untuk PREVIEW, pilih roda gigi untuk membuka pengaturan Partisi.
Roda gigi untuk pengaturan tambahan terletak di paling kanan judul PREVIEW.

    - Tentukan jumlah partisi. Anda dapat mempartisi berdasarkan kolom jika Anda menentukan jumlah partisi:
        - Masukkan jumlah partisi - Tentukan nilai yang lebih besar dari 2.
        - (Opsional) Partisi demi kolom - Tentukan bidang berikut. Anda hanya dapat mempartisi dengan kolom jika Anda telah menentukan nilai untuk Masukkan jumlah partisi.
            - Pilih kolom - Pilih kolom yang Anda gunakan untuk partisi data. Jenis data kolom harus berupa angka atau tanggal.
            - Batas atas - Dari nilai di kolom yang telah Anda tentukan, batas atas adalah nilai yang Anda gunakan di partisi. Nilai yang Anda tentukan tidak mengubah data yang Anda

impor. Itu hanya mempengaruhi kecepatan impor. Untuk kinerja terbaik, tentukan batas atas yang mendekati maksimum kolom.

- Batas bawah - Dari nilai di kolom yang telah Anda tentukan, batas bawah adalah nilai yang Anda gunakan di partisi. Nilai yang Anda tentukan tidak mengubah data yang Anda impor. Itu hanya mempengaruhi kecepatan impor. Untuk kinerja terbaik, tentukan batas bawah yang mendekati minimum kolom.

11. Pilih Impor.

Impor data dari Salesforce Data Cloud

Anda dapat menggunakan Salesforce Data Cloud sebagai sumber data di Amazon Data Wrangler untuk menyiapkan SageMaker data di Salesforce Data Cloud Anda untuk pembelajaran mesin.

Dengan Salesforce Data Cloud sebagai sumber data di Data Wrangler, Anda dapat dengan cepat terhubung ke data Salesforce Anda tanpa menulis satu baris kode pun. Anda dapat menggabungkan data Salesforce Anda dengan data dari sumber data lain di Data Wrangler.

Setelah terhubung ke cloud data, Anda dapat melakukan hal berikut ini:

- Visualisasikan data Anda dengan visualisasi bawaan
- Memahami data dan mengidentifikasi potensi kesalahan dan nilai ekstrim
- Transformasi data dengan lebih dari 300 transformasi bawaan
- Ekspor data yang telah Anda ubah

Topik

- Set Alur
- Panduan Ilmuwan Data

Set Alur

> **Important**
>
> Sebelum memulai, pastikan bahwa pengguna Anda menjalankan Amazon SageMaker Studio Classic versi 1.3.0 atau yang lebih baru. Untuk informasi tentang memeriksa versi Studio

Classic dan memperbaruinya, lihatSiapkan Data ML dengan Amazon SageMaker Data Wrangler.

Saat menyiapkan akses ke Salesforce Data Cloud, Anda harus menyelesaikan tugas berikut:

- Mendapatkan URL Domain Salesforce Anda. Salesforce juga mengacu pada URL Domain sebagai URL organisasi Anda.
- Mendapatkan kredensi OAuth dari Salesforce.
- Mendapatkan URL otorisasi dan URL token untuk Domain Salesforce Anda.
- Membuat AWS Secrets Manager rahasia dengan konfigurasi OAuth.
- Membuat konfigurasi siklus hidup yang digunakan Data Wrangler untuk membaca kredensi dari rahasia.
- Memberikan izin Data Wrangler untuk membaca rahasianya.

Setelah Anda melakukan tugas sebelumnya, pengguna Anda dapat masuk ke Salesforce Data Cloud menggunakan OAuth.

> **Note**
>
> Pengguna Anda mungkin mengalami masalah setelah Anda mengatur semuanya. Untuk informasi pemecahan masalah, lihat. Pemecahan Masalah dengan Salesforce

Gunakan prosedur berikut untuk mendapatkan URL Domain.

1. Arahkan ke halaman login Salesforce.
2. Untuk Pencarian cepat, tentukan Domain Saya.
3. Salin nilai URL Domain Saya Saat Ini ke file teks.
4. Tambahkan `https://` ke awal URL.

Setelah Anda mendapatkan URL Domain Salesforce, Anda dapat menggunakan prosedur berikut untuk mendapatkan kredensyal login dari Salesforce dan memungkinkan Data Wrangler untuk mengakses data Salesforce Anda.

Untuk mendapatkan kredensi log in dari Salesforce dan memberikan akses ke Data Wrangler, lakukan hal berikut.

1. Arahkan ke URL Domain Salesforce Anda dan masuk ke akun Anda.
2. Pilih ikon roda gigi.
3. Di bilah pencarian yang muncul, tentukan Manajer Aplikasi.
4. Pilih Aplikasi Terhubung Baru.
5. Tentukan bidang berikut:
   - Nama Aplikasi Terhubung — Anda dapat menentukan nama apa pun, tetapi sebaiknya pilih nama yang menyertakan Data Wrangler. Misalnya, Anda dapat menentukan Integrasi Wrangler Data Cloud Data Salesforce.
   - Nama API - Gunakan Nilai default.
   - Email Kontak - Tentukan alamat email Anda.
   - Di bawah judul API (Aktifkan Pengaturan OAuth), pilih kotak centang untuk mengaktifkan pengaturan OAuth.
   - Untuk URL Callback, tentukan URL Amazon SageMaker Studio Classic. Untuk mendapatkan URL Studio Classic, akses dari AWS Management Console dan salin URL.
6. Di bawah Lingkup OAuth yang Dipilih, pindahkan yang berikut ini dari Cakupan OAuth yang Tersedia ke Lingkup OAuth yang Dipilih:
   - Mengelola data pengguna melalui API (`api`)
   - Lakukan permintaan kapan saja (`refresh_token`,`offline_access`)
   - Lakukan kueri ANSI SQL pada data Salesforce Data Cloud () `cdp_query_api`
   - Mengelola data profil Platform Data Pelanggan Salesforce () `cdp_profile_api`
7. Pilih Simpan. Setelah Anda menyimpan perubahan, Salesforce membuka halaman baru.
8. Pilih Continue (Lanjutkan)
9. Arahkan ke Kunci Konsumen dan Rahasia.
10. Pilih Kelola Detail Konsumen. Salesforce mengarahkan Anda ke halaman baru di mana Anda mungkin harus melewati otentikasi dua faktor.

11.
> **Important**
>
> Salin Kunci Konsumen dan Rahasia Konsumen ke editor teks. Anda memerlukan informasi ini untuk menghubungkan data cloud ke Data Wrangler.

12. Arahkan kembali ke Kelola Aplikasi Terhubung.
13. Arahkan ke Nama Aplikasi Terhubung dan nama aplikasi Anda.
14. Pilih Kelola.
    a. Pilih Edit Kebijakan.
    b. Ubah Relaksasi IP ke Relaksasi pembatasan IP.
    c. Pilih Simpan.

Setelah Anda memberikan akses ke Salesforce Data Cloud, Anda perlu memberikan izin untuk pengguna Anda. Gunakan prosedur berikut untuk memberi mereka izin.

Untuk memberikan izin kepada pengguna Anda, lakukan hal berikut.

1. Arahkan ke halaman beranda pengaturan.
2. Di navigasi sebelah kiri, cari Pengguna dan pilih item menu Pengguna.
3. Pilih hyperlink dengan nama pengguna Anda.
4. Arahkan ke Izin Set Tugas.
5. Pilih Edit Tugas.
6. Tambahkan izin berikut:
   - Admin Platform Data Pelanggan
   - Spesialis Sadar Data Platform Data Pelanggan
7. Pilih Simpan.

Setelah Anda mendapatkan informasi untuk Domain Salesforce Anda, Anda harus mendapatkan URL otorisasi dan URL token untuk AWS Secrets Manager rahasia yang Anda buat.

Gunakan prosedur berikut untuk mendapatkan URL otorisasi dan URL token.

Untuk mendapatkan URL otorisasi dan URL token

1. Arahkan ke URL Domain Salesforce Anda.
2. Gunakan salah satu metode berikut untuk mendapatkan URL. Jika Anda menggunakan distribusi Linux dengan `curl` dan `jq` diinstal, kami sarankan menggunakan metode yang hanya berfungsi di Linux.
   - (Hanya Linux) Tentukan perintah berikut di terminal Anda.

     ```
     curl salesforce-domain-URL/.well-known/openid-configuration | \
     jq '. | { authorization_url: .authorization_endpoint,
      token_url: .token_endpoint }' | \
     jq '.  += { identity_provider: "SALESFORCE", client_id: "example-client-id",
      client_secret: "example-client-secret" }'
     ```

   - a. Arahkan ke *Example-org-URL /.well-known/openid-configuration* di browser Anda.
     b. Salin `authorization_endpoint` dan `token_endpoint` ke editor teks.
     c. Buat objek JSON berikut:

     ```
     {
       "identity_provider": "SALESFORCE",
       "authorization_url": "example-authorization-endpoint",
       "token_url": "example-token-endpoint",
       "client_id": "example-consumer-key",
       "client_secret": "example-consumer-secret"
     }
     ```

Setelah Anda membuat objek konfigurasi OAuth, Anda dapat membuat AWS Secrets Manager rahasia yang menyimpannya. Gunakan prosedur berikut untuk membuat rahasia.

Untuk membuat rahasia, lakukan hal berikut.

1. Navigasikan ke konsol AWS Secrets Manager tersebut.
2. Pilih Simpan rahasia.
3. Pilih Jenis rahasia lainnya.

4. Di bawah pasangan kunci/nilai pilih Plaintext.
5. Ganti JSON kosong dengan pengaturan konfigurasi berikut.

```
{
  "identity_provider": "SALESFORCE",
  "authorization_url": "example-authorization-endpoint",
  "token_url": "example-token-endpoint",
  "client_id": "example-consumer-key",
  "client_secret": "example-consumer-secret"
}
```

6. Pilih Berikutnya.
7. Untuk Nama Rahasia, tentukan nama rahasianya.
8. Di bawah Tag, pilih Tambah.
   - Untuk Kunci, tentukan sagemaker:partner. Untuk Nilai, sebaiknya tentukan nilai yang mungkin berguna untuk kasus penggunaan Anda. Namun, Anda dapat menentukan apa saja.

> **Important**
>
> Anda harus membuat kuncinya. Anda tidak dapat mengimpor data dari Salesforce jika Anda tidak membuatnya.

9. Pilih Berikutnya.
10. Pilih Toko.
11. Pilih rahasia yang Anda buat.
12. Perhatikan bidang berikut:
    - Amazon Resource Number (ARN) rahasia
    - Nama rahasia

Setelah Anda membuat rahasia, Anda harus menambahkan izin untuk Data Wrangler untuk membaca rahasianya. Gunakan prosedur berikut untuk menambahkan izin.

Untuk menambahkan izin baca untuk Data Wrangler, lakukan hal berikut.

1. Arahkan ke SageMaker konsol Amazon.
2. Pilih Domain.
3. Pilih domain yang Anda gunakan untuk mengakses Data Wrangler.
4. Pilih Profil Pengguna Anda.
5. Di bawah Detail, temukan peran Eksekusi. ARN-nya dalam format berikut:. `arn:aws:iam::111122223333:role/example-role` Perhatikan peran SageMaker eksekusi. Di dalam ARN, semuanya setelahnya. `role/`
6. Arahkan ke konsol IAM.
7. Di bilah pencarian IAM Cari, tentukan nama peran SageMaker eksekusi.
8. Pilih peran.
9. Pilih Tambahkan izin.
10. Pilih Buat kebijakan sebaris.
11. Pilih tab JSON.
12. Tentukan kebijakan berikut dalam editor.

```
{
 "Version": "2012-10-17",
 "Statement": [
     {
         "Effect": "Allow",
         "Action": [
             "secretsmanager:GetSecretValue",
             "secretsmanager:PutSecretValue"
         ],
         "Resource": "arn:aws:secretsmanager:*:*:secret:*",
         "Condition": {
             "ForAnyValue:StringLike": {
                 "aws:ResourceTag/sagemaker:partner": "*"
             }
         }
     },
     {
         "Effect": "Allow",
         "Action": [
             "secretsmanager:UpdateSecret"
         ],
         "Resource": "arn:aws:secretsmanager:*:*:secret:AmazonSageMaker-*"
```

```
        }
    ]
}
```

13. Pilih Tinjau Kebijakan.
14. Untuk Nama, tentukan nama.
15. Pilih Buat kebijakan.

Setelah Anda memberikan izin Data Wrangler untuk membaca rahasia, Anda harus menambahkan Konfigurasi Siklus Hidup yang menggunakan rahasia Secrets Manager ke profil pengguna Amazon SageMaker Studio Classic Anda.

Gunakan prosedur berikut ini untuk membuat konfigurasi siklus hidup dan menambahkannya ke profil Studio Classic.

Untuk membuat konfigurasi siklus hidup dan menambahkannya ke profil Studio Classic, lakukan hal berikut.

1. Arahkan ke SageMaker konsol Amazon.
2. Pilih Domain.
3. Pilih domain yang Anda gunakan untuk mengakses Data Wrangler.
4. Pilih Profil Pengguna Anda.
5. Jika Anda melihat aplikasi berikut, hapus:
   - KernelGateway
   - JupyterKernel

   > **Note**
   >
   > Menghapus pembaruan aplikasi Studio Classic. Hal ini dapat memakan waktu beberapa saat untuk pembaruan terjadi.

6. Saat Anda menunggu pembaruan terjadi, pilih Konfigurasi Siklus Hidup.
7. Pastikan halaman yang Anda kunjungi mengatakan konfigurasi Siklus Hidup Studio Classic.
8. Pilih Buat konfigurasi.

9. Pastikan aplikasi server Jupyter telah dipilih.
10. Pilih Berikutnya.
11. Untuk Nama, tentukan nama untuk konfigurasi.
12. Untuk Skrip, tentukan skrip berikut:

```
#!/bin/bash
set -eux

cat > ~/.sfgenie_identity_provider_oauth_config <<EOL
{
    "secret_arn": "secrets-arn-containing-salesforce-credentials"
}
EOL
```

13. Pilih Kirim.
14. Di navigasi sebelah kiri, pilih Domain.
15. Pilih domain.
16. Pilih Lingkungan.
17. Di bawah Konfigurasi Siklus Hidup untuk aplikasi Studio Classic pribadi, pilih Lampirkan.
18. Pilih Konfigurasi yang ada.
19. Di bawah konfigurasi Siklus Hidup Studio Classic pilih konfigurasi siklus hidup yang telah Anda buat.
20. Pilih Lampirkan ke domain.
21. Pilih kotak centang di samping konfigurasi siklus aktif yang telah Anda lampirkan.
22. Pilih Tetapkan sebagai default.

Anda mungkin mengalami masalah saat mengatur konfigurasi siklus. Untuk informasi tentang debugging mereka, lihatDebug konfigurasi siklus hidup.

Panduan Ilmuwan Data

Gunakan yang berikut ini untuk menghubungkan Salesforce Data Cloud dan mengakses data Anda di Data Wrangler.

> **Important**
>
> Administrator Anda perlu menggunakan informasi di bagian sebelumnya untuk menyiapkan Salesforce Data Cloud. Jika Anda mengalami masalah, hubungi mereka untuk bantuan pemecahan masalah.

Untuk membuka Studio Classic dan memeriksa versinya, lihat prosedur berikut.

1. Gunakan langkah-langkah Prasyarat untuk mengakses Data Wrangler melalui Amazon SageMaker Studio Classic.
2. Di samping pengguna yang ingin Anda gunakan untuk meluncurkan Studio Classic, pilih Luncurkan aplikasi.
3. Pilih Studio.

Untuk membuat dataset di Data Wrangler dengan data dari Salesforce Data Cloud

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Pilih Impor data.
9. Di bawah Tersedia, pilih Salesforce Data Cloud.
10. Untuk nama Connection, tentukan nama untuk koneksi Anda ke Salesforce Data Cloud.
11. Untuk URL Org, tentukan URL organisasi di akun Salesforce Anda. Anda bisa mendapatkan URL dari administrator
12. Pilih Hubungkan.
13. Tentukan kredensional Anda untuk masuk ke Salesforce.

Anda dapat mulai membuat kumpulan data menggunakan data dari Salesforce Data Cloud setelah Anda terhubung dengannya.

Setelah Anda memilih tabel, Anda dapat menulis kueri dan menjalankannya. Output kueri Anda ditampilkan di bawah Hasil kueri.

Setelah Anda menyelesaikan output kueri Anda, Anda kemudian dapat mengimpor output kueri Anda ke dalam aliran Data Wrangler untuk melakukan transformasi data.

Setelah membuat kumpulan data, arahkan ke layar Aliran data untuk mulai mengubah data Anda.

Impor data dari Snowflake

Anda dapat menggunakan Snowflake sebagai sumber data di Data Wrangler untuk menyiapkan SageMaker data di Snowflake untuk pembelajaran mesin.

Dengan Snowflake sebagai sumber data di Data Wrangler, Anda dapat dengan cepat terhubung ke Snowflake tanpa menulis satu baris kode pun. Anda dapat menggabungkan data Anda di Snowflake dengan data dari sumber data lain di Data Wrangler.

Setelah terhubung, Anda dapat secara interaktif menanyakan data yang disimpan di Snowflake, mengubah data dengan lebih dari 300 transformasi data yang telah dikonfigurasi sebelumnya, memahami data, dan mengidentifikasi potensi kesalahan dan nilai ekstrem dengan serangkaian templat visualisasi yang telah dikonfigurasi sebelumnya, dengan cepat mengidentifikasi inkonsistensi dalam alur kerja persiapan data Anda, dan mendiagnosis masalah sebelum model digunakan ke dalam produksi. Terakhir, Anda dapat mengekspor alur kerja persiapan data ke Amazon S3 untuk digunakan dengan fitur SageMaker lain seperti Amazon Autopilot, SageMaker Amazon Feature Store, dan SageMaker SageMaker Amazon Model Building Pipelines.

Anda dapat mengenkripsi output kueri Anda menggunakan AWS Key Management Service kunci yang telah Anda buat. Untuk informasi selengkapnya tentang AWS KMS, lihat AWS Key Management Service.

Topik

- Panduan Administrator
- Panduan Ilmuwan Data

Panduan Administrator

> **Important**
>
> Untuk mempelajari lebih lanjut tentang kontrol akses terperinci dan praktik terbaik, lihat Kontrol Akses Keamanan.

Bagian ini untuk administrator Snowflake yang menyiapkan akses ke Snowflake dari dalam Data Wrangler. SageMaker

> **Important**
>
> Anda bertanggung jawab untuk mengelola dan memantau kontrol akses dalam Snowflake. Data Wrangler tidak menambahkan lapisan kontrol akses sehubungan dengan Snowflake. Kontrol akses meliputi:
>
> - Data yang diakses pengguna
> - (Opsional) Integrasi penyimpanan yang menyediakan Snowflake kemampuan untuk menulis hasil kueri ke bucket Amazon S3
> - Kueri yang dapat dijalankan pengguna

(Opsional) Konfigurasikan Izin Impor Data Kepingan Salju

Secara default, Data Wrangler menanyakan data di Snowflake tanpa membuat salinannya di lokasi Amazon S3. Gunakan informasi berikut jika Anda mengonfigurasi integrasi penyimpanan dengan Snowflake. Pengguna Anda dapat menggunakan integrasi penyimpanan untuk menyimpan hasil kueri mereka di lokasi Amazon S3.

Pengguna Anda mungkin memiliki tingkat akses data sensitif yang berbeda. Untuk keamanan data yang optimal, sediakan integrasi penyimpanan masing-masing pengguna. Setiap integrasi penyimpanan harus memiliki kebijakan tata kelola datanya sendiri.

Fitur ini saat ini tidak tersedia di Wilayah opt-in.

Snowflake memerlukan izin berikut pada bucket dan direktori S3 untuk dapat mengakses file di direktori:

- `s3:GetObject`
- `s3:GetObjectVersion`
- `s3:ListBucket`
- `s3:ListObjects`
- `s3:GetBucketLocation`

Membuat kebijakan IAM

Anda harus membuat kebijakan IAM untuk mengonfigurasi izin akses bagi Snowflake untuk memuat dan membongkar data dari bucket Amazon S3.

Berikut ini adalah dokumen kebijakan JSON yang Anda gunakan untuk membuat kebijakan:

```
# Example policy for S3 write access
# This needs to be updated
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
    ],
    "Resource": "arn:aws:s3:::bucket/prefix/*"
  },
  {
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::bucket/",
    "Condition": {
        "StringLike": {
            "s3:prefix": ["prefix/*"]
        }
    }
  }
```

```
 ]
}
```

Untuk informasi dan prosedur tentang membuat kebijakan dengan dokumen kebijakan, lihat Membuat kebijakan IAM.

Untuk dokumentasi yang memberikan ikhtisar penggunaan izin IAM dengan Snowflake, lihat sumber daya berikut:

- Apa itu IAM?
- Buat Peran IAM di AWS
- Buat Integrasi Penyimpanan Cloud di Snowflake
- Ambil Pengguna AWS IAM untuk Akun Snowflake Anda
- Berikan Izin Pengguna IAM untuk Mengakses Bucket.

Untuk memberikan izin penggunaan peran Snowflake ilmuwan data ke integrasi penyimpanan, Anda harus menjalankannya. `GRANT USAGE ON INTEGRATION integration_name TO snowflake_role;`

- `integration_name`adalah nama integrasi penyimpanan Anda.
- `snowflake_role`adalah nama peran Snowflake default yang diberikan kepada pengguna ilmuwan data.

Menyiapkan Akses OAuth Snowflake

Alih-alih meminta pengguna Anda langsung memasukkan kredensialnya ke Data Wrangler, Anda dapat meminta mereka menggunakan penyedia identitas untuk mengakses Snowflake. Berikut ini adalah tautan ke dokumentasi Snowflake untuk penyedia identitas yang didukung Data Wrangler.

- Azure AD
- Okta
- Federasi Ping

Gunakan dokumentasi dari tautan sebelumnya untuk mengatur akses ke penyedia identitas Anda. Informasi dan prosedur di bagian ini membantu Anda memahami cara menggunakan dokumentasi dengan benar untuk mengakses Snowflake dalam Data Wrangler.

Penyedia identitas Anda perlu mengenali Data Wrangler sebagai aplikasi. Gunakan prosedur berikut ini untuk mendaftarkan Data Wrangler sebagai aplikasi dalam penyedia identitas:

1. Pilih konfigurasi yang memulai proses pendaftaran Data Wrangler sebagai aplikasi.
2. Menyediakan pengguna dalam penyedia identitas akses ke Data Wrangler.
3. Aktifkan otentikasi klien OAuth dengan menyimpan kredensyal klien sebagai rahasia. AWS Secrets Manager
4. Tentukan URL pengalihan menggunakan format berikut: https://*domain-ID* .studio. *Wilayah AWS*.sagemaker.aws/jupyter/default/lab

   > **Important**
   >
   > Anda menentukan ID SageMaker Domain Amazon dan Wilayah AWS yang Anda gunakan untuk menjalankan Data Wrangler.

   > **Important**
   >
   > Anda harus mendaftarkan URL untuk setiap SageMaker Domain Amazon dan Wilayah AWS tempat Anda menjalankan Data Wrangler. Pengguna dari Domain dan Wilayah AWS yang tidak memiliki URL pengalihan yang disiapkan untuk mereka tidak akan dapat mengautentikasi dengan penyedia identitas untuk mengakses koneksi Snowflake.
5. Pastikan kode otorisasi dan jenis hibah token refresh diizinkan untuk aplikasi Data Wrangler.

Dalam penyedia identitas Anda, Anda harus menyiapkan server yang mengirimkan token OAuth ke Data Wrangler di tingkat pengguna. Server mengirimkan token dengan Snowflake sebagai penonton.

Snowflake menggunakan konsep peran yang berbeda peran IAM digunakan. AWS Anda harus mengonfigurasi penyedia identitas untuk menggunakan peran apa pun untuk menggunakan peran default yang terkait dengan akun Snowflake. Misalnya, jika pengguna memiliki peran default dalam profil Snowflake mereka, koneksi dari Data Wrangler ke Snowflake digunakan `systems administrator` sebagai peran. `systems administrator`

Gunakan prosedur berikut untuk mengatur server.

Untuk menyiapkan server, lakukan hal berikut. Anda bekerja di dalam Snowflake untuk semua langkah kecuali yang terakhir.

1. Mulai mengatur server atau API.
2. Konfigurasikan server otorisasi untuk menggunakan kode otorisasi dan segarkan jenis hibah token.
3. Tentukan masa pakai token akses.
4. Setel batas waktu idle token refresh. Batas waktu idle adalah waktu token refresh kedaluwarsa jika tidak digunakan.

> **Note**
>
> Jika Anda menjadwalkan pekerjaan di Data Wrangler, kami sarankan untuk membuat waktu tunggu idle lebih besar daripada frekuensi pekerjaan pemrosesan. Jika tidak, beberapa pekerjaan pemrosesan mungkin gagal karena token penyegaran kedaluwarsa sebelum dapat dijalankan. Ketika token penyegaran kedaluwarsa, pengguna harus mengautentikasi ulang dengan mengakses koneksi yang telah mereka buat ke Snowflake melalui Data Wrangler.

5. Tentukan `session:role-any` sebagai ruang lingkup baru.

> **Note**
>
> Untuk Azure AD, salin pengenal unik untuk ruang lingkup. Data Wrangler mengharuskan Anda untuk menyediakannya dengan pengenal.

6.
> **Important**
>
> Dalam Integrasi Keamanan OAuth Eksternal untuk Kepingan Salju, aktifkan. `external_oauth_any_role_mode`

> **Important**
>
> Data Wrangler tidak mendukung token penyegaran yang berputar. Menggunakan token penyegaran yang berputar dapat mengakibatkan kegagalan akses atau pengguna harus sering masuk.

> **Important**
>
> Jika token penyegaran kedaluwarsa, pengguna Anda harus mengautentikasi ulang dengan mengakses koneksi yang telah mereka buat ke Snowflake melalui Data Wrangler.

Setelah menyiapkan penyedia OAuth, Anda memberikan Data Wrangler informasi yang dibutuhkan untuk terhubung ke penyedia. Anda dapat menggunakan dokumentasi dari penyedia identitas Anda untuk mendapatkan nilai untuk bidang berikut:

- URL Token — URL token yang dikirim oleh penyedia identitas ke Data Wrangler.
- URL otorisasi — URL server otorisasi penyedia identitas.
- ID Klien — ID penyedia identitas.
- Rahasia klien — Rahasia yang hanya dikenali oleh server otorisasi atau API.
- (Hanya Azure AD) Kredensional cakupan OAuth yang telah Anda salin.

Anda menyimpan bidang dan nilai dalam AWS Secrets Manager rahasia dan menambahkannya ke konfigurasi siklus hidup Amazon SageMaker Studio Classic yang Anda gunakan untuk Data Wrangler. Konfigurasi Siklus Hidup adalah skrip shell. Gunakan untuk membuat Amazon Resource Name (ARN) dari rahasia dapat diakses oleh Data Wrangler. Untuk informasi tentang membuat rahasia, lihat Memindahkan rahasia hardcode ke. AWS Secrets Manager Untuk informasi tentang menggunakan konfigurasi siklus hidup di Studio Classic, lihat. Menggunakan konfigurasi siklus hidup dengan Amazon Studio Classic SageMaker

> **Important**
>
> Sebelum Anda membuat rahasia Secrets Manager, pastikan bahwa peran SageMaker eksekusi yang Anda gunakan untuk Amazon SageMaker Studio Classic memiliki izin untuk membuat dan memperbarui rahasia di Secrets Manager. Untuk informasi selengkapnya tentang menambahkan izin, lihat Contoh: Izin untuk membuat rahasia.

Untuk Okta dan Ping Federate, berikut ini adalah format rahasianya:

```
{
```

```
    "token_url":"https://identityprovider.com/oauth2/example-portion-of-URL-path/v2/
token",
    "client_id":"example-client-id",
    "client_secret":"example-client-secret",
    "identity_provider":"OKTA"|"PING_FEDERATE",
    "authorization_url":"https://identityprovider.com/oauth2/example-portion-of-URL-
path/v2/authorize"
}
```

Untuk Azure AD, berikut ini adalah format rahasianya:

```
{
    "token_url":"https://identityprovider.com/oauth2/example-portion-of-URL-path/v2/
token",
    "client_id":"example-client-id",
    "client_secret":"example-client-secret",
    "identity_provider":"AZURE_AD",
    "authorization_url":"https://identityprovider.com/oauth2/example-portion-of-URL-
path/v2/authorize",
    "datasource_oauth_scope":"api://appuri/session:role-any)"
}
```

Anda harus memiliki konfigurasi siklus hidup yang menggunakan rahasia Secrets Manager yang telah Anda buat. Anda dapat membuat konfigurasi siklus hidup atau memodifikasi konfigurasi yang telah dibuat. Konfigurasi harus menggunakan skrip berikut.

```
#!/bin/bash

set -eux

## Script Body

cat > ~/.snowflake_identity_provider_oauth_config <<EOL
{
    "secret_arn": "example-secret-arn"
}
EOL
```

Untuk informasi tentang pengaturan siklus aktif, lihat. Membuat dan mengaitkan konfigurasi siklus hidup Saat Anda menjalani proses penyiapan, lakukan hal berikut:

- Atur jenis aplikasi konfigurasi ke`Jupyter Server`.
- Lampirkan konfigurasi ke SageMaker Domain Amazon yang memiliki pengguna Anda.
- Jalankan konfigurasi secara default. Itu harus berjalan setiap kali pengguna login ke Studio Classic. Jika tidak, kredensil yang disimpan dalam konfigurasi tidak akan tersedia untuk pengguna Anda saat mereka menggunakan Data Wrangler.
- Konfigurasi siklus hidup membuat file dengan nama, `snowflake_identity_provider_oauth_config` di folder beranda pengguna. File tersebut berisi rahasia Secrets Manager. Pastikan itu ada di folder beranda pengguna setiap kali instance Jupyter Server diinisialisasi.

Konektivitas Pribadi antara Data Wrangler dan Snowflake via AWS PrivateLink

Bagian ini menjelaskan cara menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara Data Wrangler dan Snowflake. Langkah-langkah tersebut dijelaskan pada bagian berikut.

Buat VPC

Jika Anda tidak memiliki pengaturan VPC, ikuti instruksi Buat VPC baru untuk membuatnya.

Setelah Anda memiliki VPC pilihan yang ingin Anda gunakan untuk membuat koneksi pribadi, berikan kredensyal berikut kepada Administrator Snowflake Anda untuk mengaktifkan: AWS PrivateLink

- ID VPC
- AWSID Akun
- URL akun terkait yang Anda gunakan untuk mengakses Snowflake

> **Important**
>
> Seperti yang dijelaskan dalam dokumentasi Snowflake, mengaktifkan akun Snowflake Anda dapat memakan waktu hingga dua hari kerja.

Mengatur Integrasi Kepingan Salju AWS PrivateLink

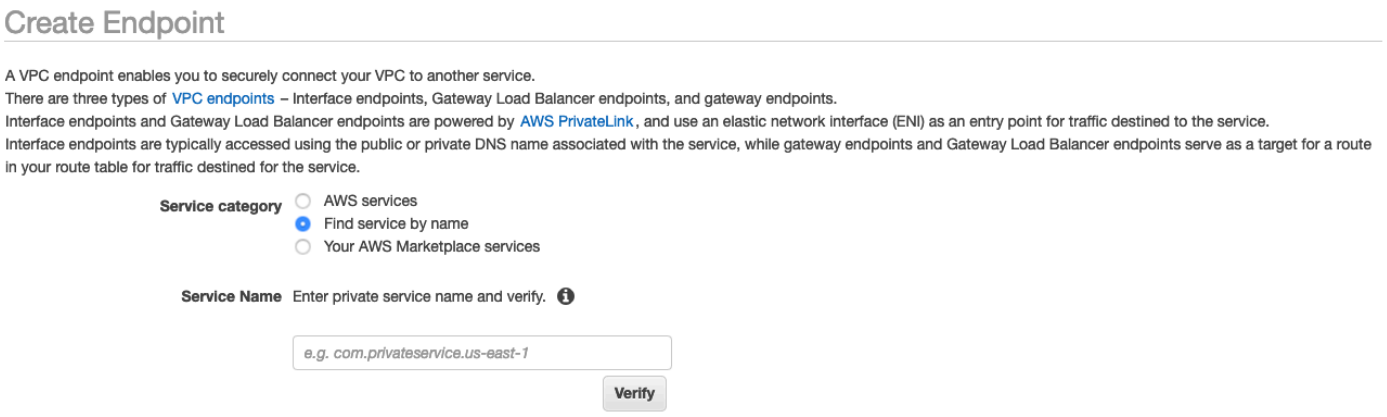
Setelah AWS PrivateLink diaktifkan, ambil AWS PrivateLink konfigurasi untuk Wilayah Anda dengan menjalankan perintah berikut di lembar kerja Snowflake. Masuk ke konsol Snowflake Anda dan masukkan yang berikut ini di bawah Lembar Kerja: `select SYSTEM $GET_PRIVATELINK_CONFIG();`

1. Ambil nilai untuk berikut:`privatelink-account-name`,, `privatelink_ocsp-urlprivatelink-account-url`, dan `privatelink_ocsp-url` dari objek JSON yang dihasilkan. Contoh setiap nilai ditunjukkan pada potongan berikut. Simpan nilai-nilai ini untuk digunakan nanti.

   ```
   privatelink-account-name: xxxxxxxx.region.privatelink
   privatelink-vpce-id: com.amazonaws.vpce.region.vpce-svc-xxxxxxxxxxxxxxxxx
   privatelink-account-url: xxxxxxxx.region.privatelink.snowflakecomputing.com
   privatelink_ocsp-url: ocsp.xxxxxxxx.region.privatelink.snowflakecomputing.com
   ```

2. Beralih ke AWS Konsol Anda dan navigasikan ke menu VPC.
3. Dari panel sisi kiri, pilih tautan Endpoints untuk menavigasi ke pengaturan VPC Endpoints.

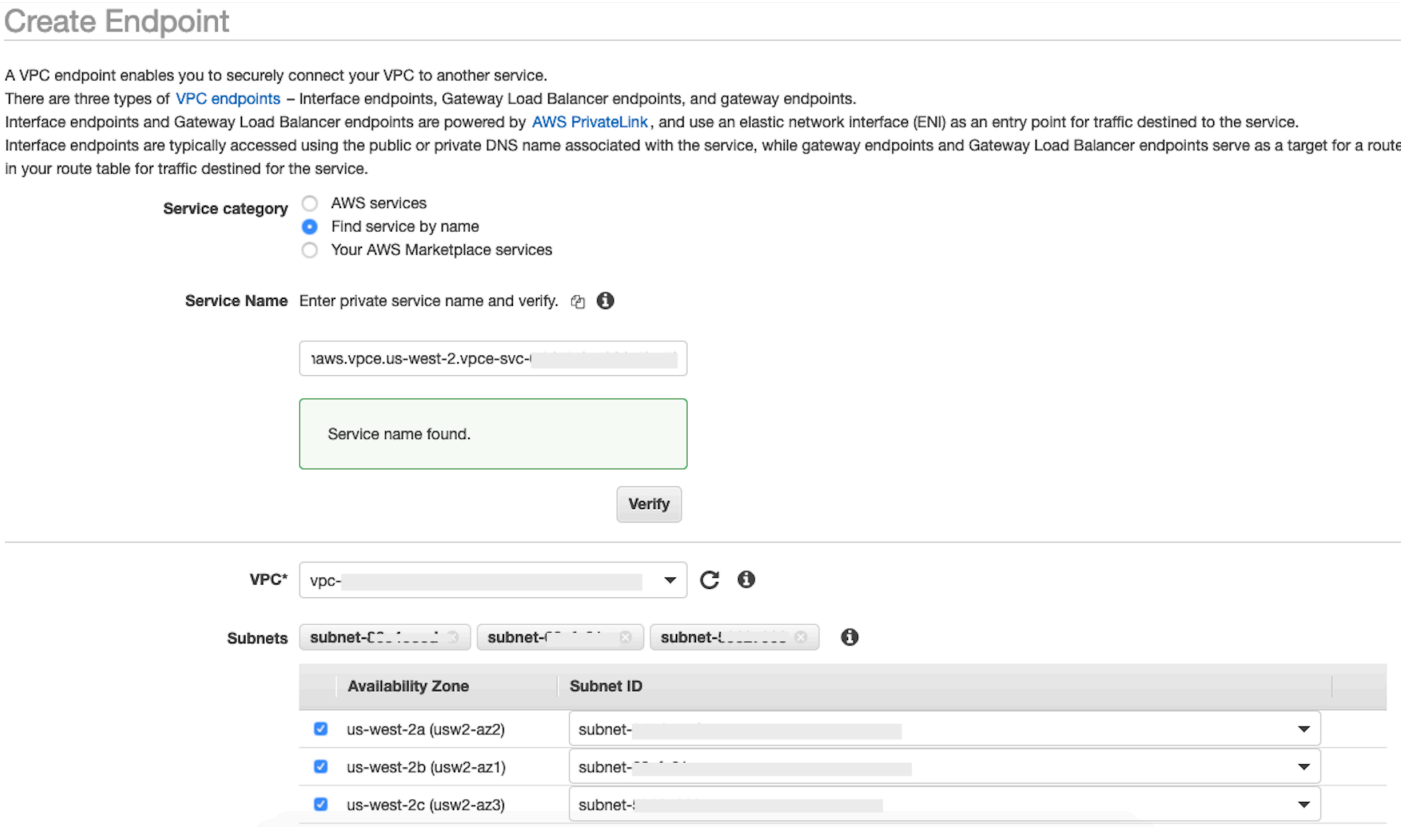
   Sesampai di sana, pilih Create Endpoint.
4. Pilih tombol radio untuk Temukan layanan dengan nama, seperti yang ditunjukkan pada gambar berikut.



5. Di bidang Nama Layanan, tempelkan nilai untuk **`privatelink-vpce-id`** yang Anda ambil di langkah sebelumnya dan pilih Verifikasi.

   Jika koneksi berhasil, peringatan hijau yang mengatakan Nama layanan ditemukan muncul di layar Anda dan opsi VPC dan Subnet secara otomatis meluas, seperti yang ditunjukkan pada gambar

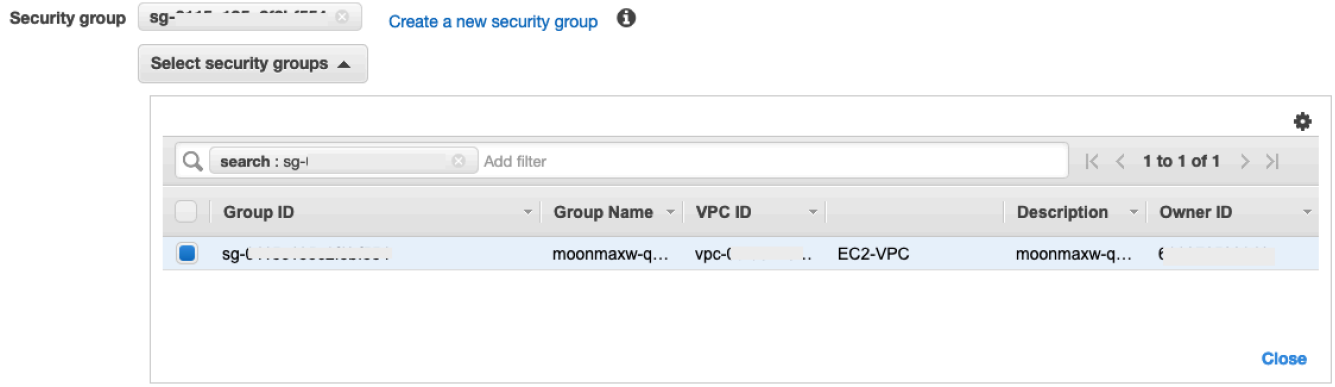
berikut. Bergantung pada Wilayah yang ditargetkan, layar hasil Anda mungkin menampilkan nama AWS Wilayah lain.



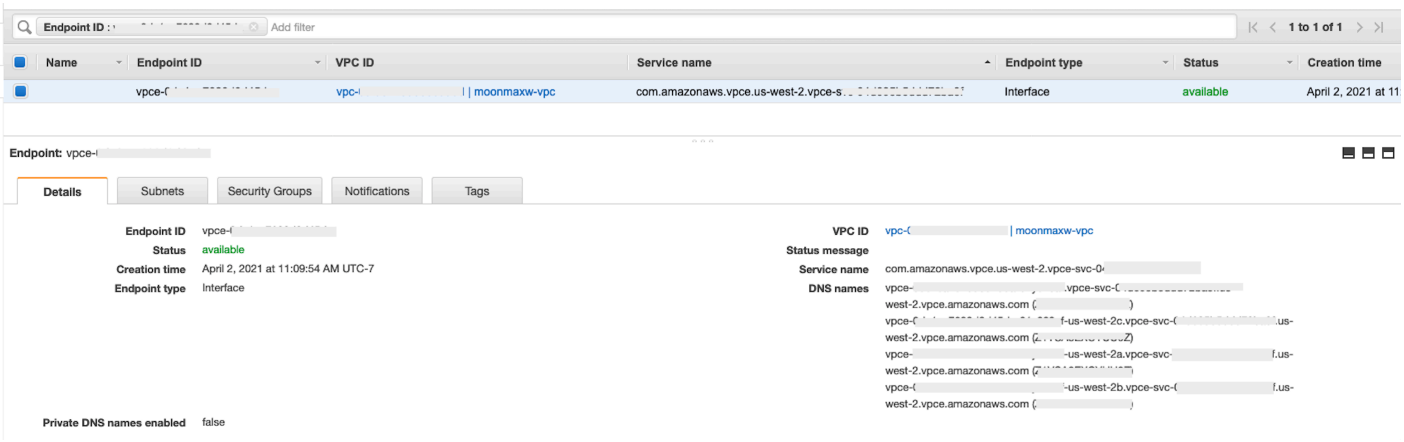
6. Pilih ID VPC yang sama yang Anda kirim ke Snowflake dari daftar dropdown VPC.
7. Jika Anda belum membuat subnet, maka lakukan serangkaian instruksi berikut untuk membuat subnet.
8. Pilih Subnet dari daftar dropdown VPC. Kemudian pilih Buat subnet dan ikuti petunjuk untuk membuat subset di VPC Anda. Pastikan Anda memilih ID VPC yang Anda kirim Snowflake.
9. Di bawah Konfigurasi Grup Keamanan, pilih Buat Grup Keamanan Baru untuk membuka layar Grup Keamanan default di tab baru. Di tab baru ini, pilih t Buat Grup Keamanan.
10. Berikan nama untuk grup keamanan baru (seperti`datawrangler-doc-snowflake-privatelink-connection`) dan deskripsi. Pastikan untuk memilih ID VPC yang telah Anda gunakan pada langkah sebelumnya.
11. Tambahkan dua aturan untuk mengizinkan lalu lintas dari dalam VPC Anda ke titik akhir VPC ini.

    Arahkan ke VPC Anda di bawah VPC Anda di tab terpisah, dan ambil blok CIDR Anda untuk VPC Anda. Kemudian pilih Tambahkan Aturan di bagian Aturan Masuk. Pilih HTTPS jenisnya, biarkan Sumber sebagai Kustom dalam formulir, dan tempel nilai yang diambil dari `describe-vpcs` panggilan sebelumnya (seperti). `10.0.0.0/16`

12. Pilih Buat Grup Keamanan. Ambil ID Grup Keamanan dari grup keamanan yang baru dibuat (seperti`sg-xxxxxxxxxxxxxxxxx`).
13. Di layar konfigurasi VPC Endpoint, hapus grup keamanan default. Tempel di ID grup keamanan di bidang pencarian dan pilih kotak centang.



14. Pilih Buat Titik Akhir.
15. Jika pembuatan titik akhir berhasil, Anda melihat halaman yang memiliki tautan ke konfigurasi titik akhir VPC Anda, yang ditentukan oleh ID VPC. Pilih tautan untuk melihat konfigurasi secara penuh.

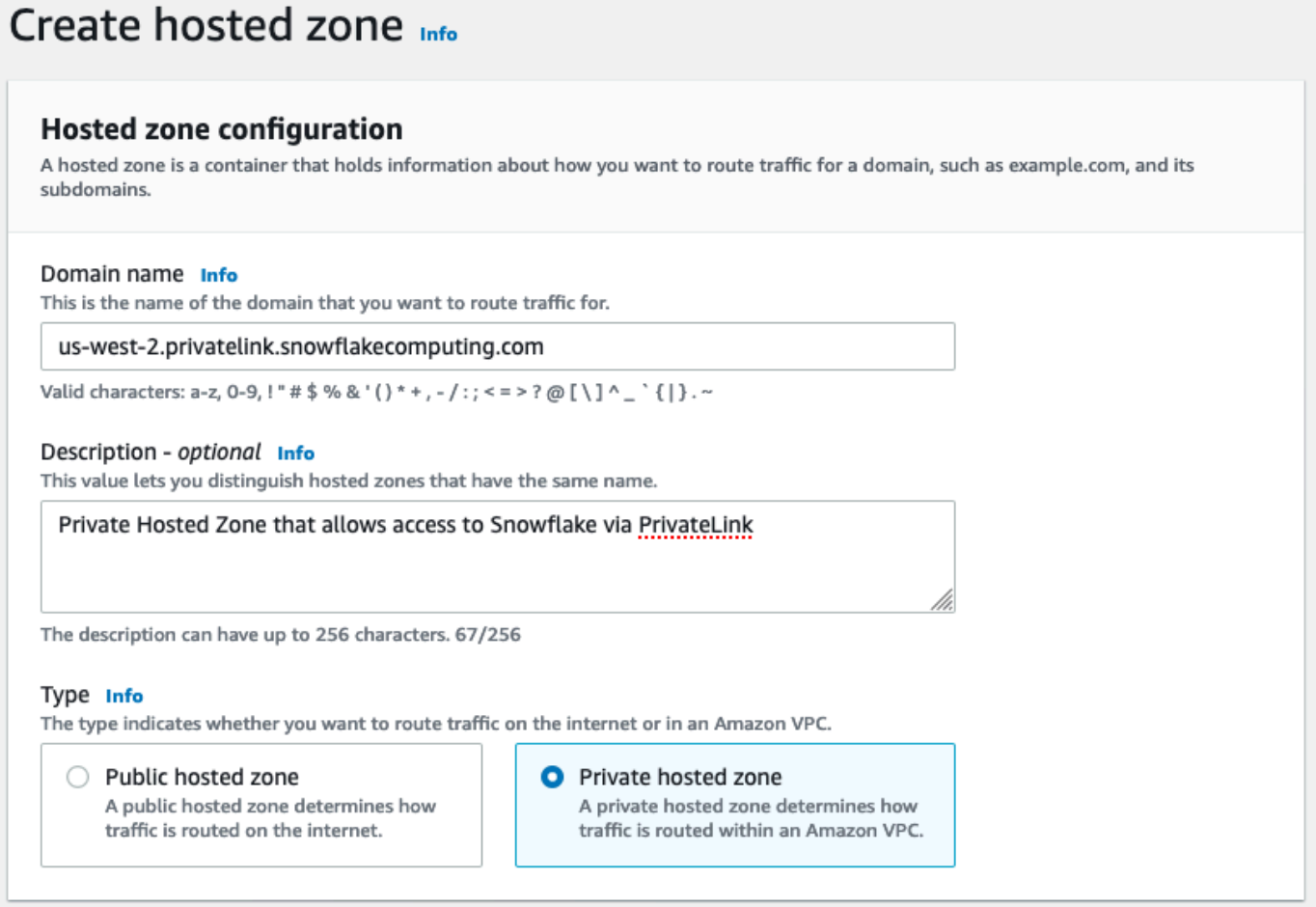


Ambil catatan paling atas dalam daftar nama DNS. Ini dapat dibedakan dari nama DNS lain karena hanya menyertakan nama Wilayah (seperti`us-west-2`), dan tidak ada notasi huruf Availability Zone (seperti). `us-west-2a` Simpan informasi ini untuk digunakan nanti.

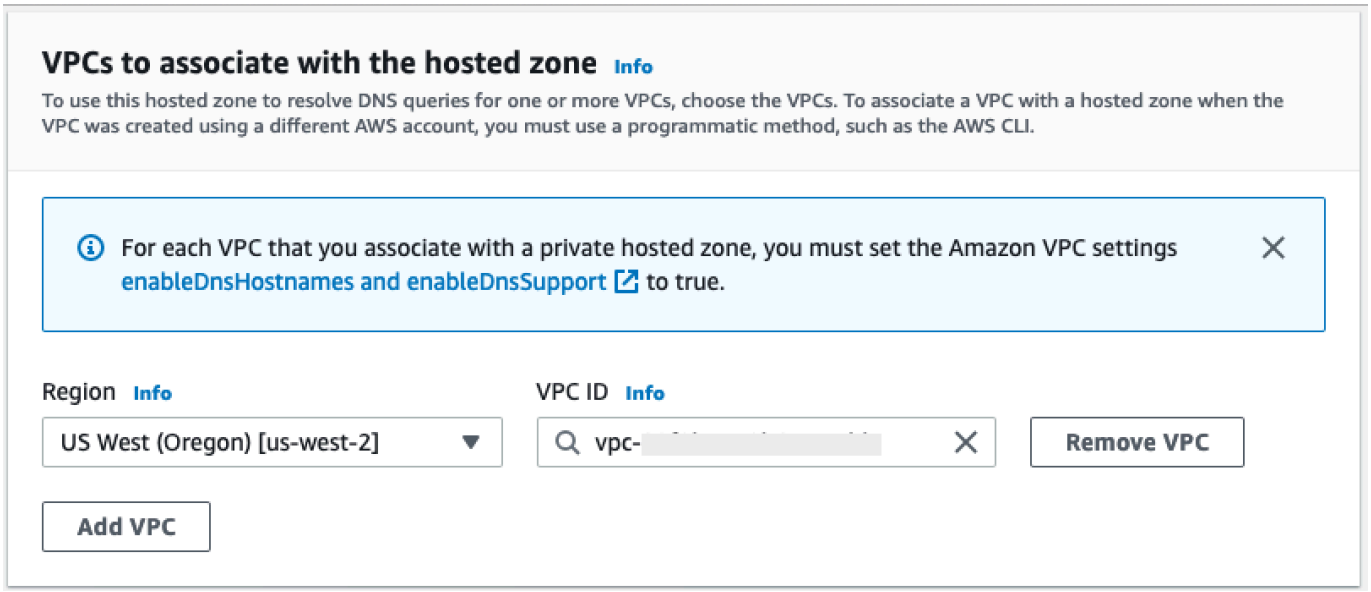
Konfigurasikan DNS untuk Titik Akhir Snowflake di VPC Anda

Bagian ini menjelaskan cara mengonfigurasi DNS untuk titik akhir Snowflake di VPC Anda. Ini memungkinkan VPC Anda menyelesaikan permintaan ke titik akhir SnowflakeAWS PrivateLink.

1. Arahkan ke menu Route 53 di dalam AWS konsol Anda.
2. Pilih opsi Zona yang Dihosting (jika perlu, perluas menu sebelah kiri untuk menemukan opsi ini).
3. Pilih Buat Zona yang Di-hosting.
   a. Di bidang Nama domain, referensi nilai yang disimpan untuk `privatelink-account-url` langkah-langkah sebelumnya. Di bidang ini, ID akun Snowflake Anda dihapus dari nama DNS dan hanya menggunakan nilai yang dimulai dengan pengenal Wilayah. Sebuah Resource Record Set juga dibuat nanti untuk subdomain, seperti,`region.privatelink.snowflakecomputing.com`.
   b. Pilih tombol radio untuk Private Hosted Zone di bagian Type. Kode Wilayah Anda mungkin tidak`us-west-2`. Referensi nama DNS yang dikembalikan kepada Anda oleh Snowflake.



   c. Di bagian VPC untuk dikaitkan dengan zona yang dihosting, pilih Wilayah tempat VPC Anda berada dan ID VPC yang digunakan pada langkah sebelumnya.



d. Pilih Buat zona yang di-hosting.

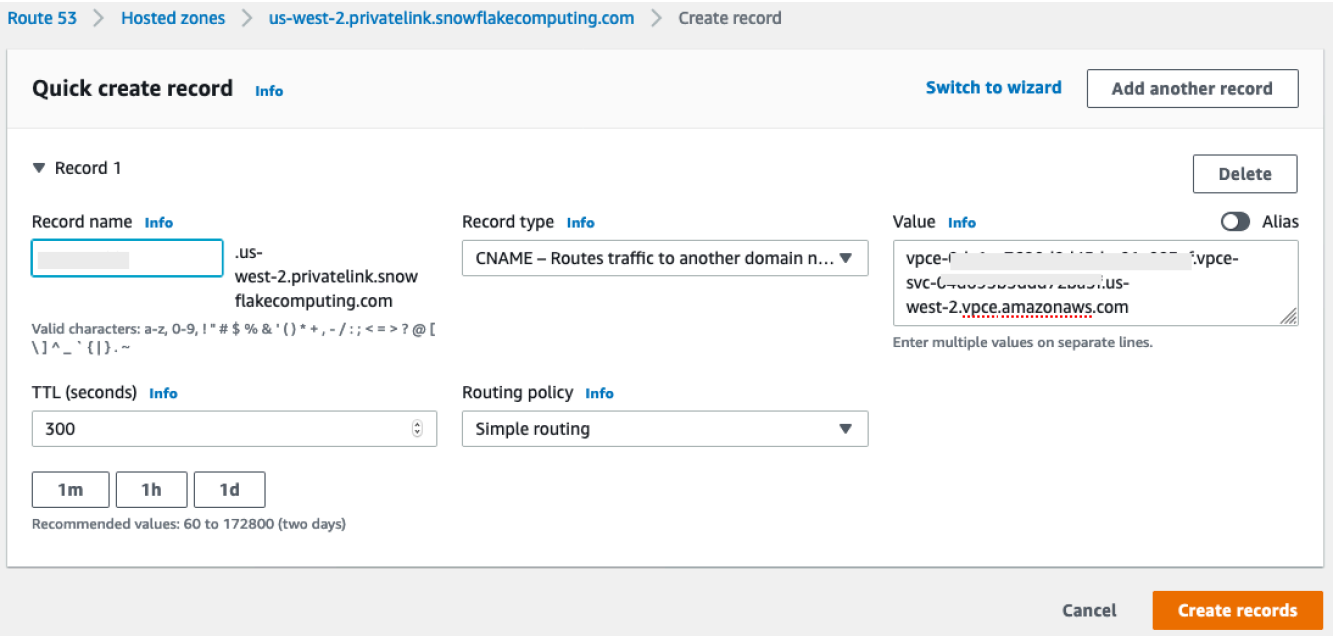
4. Selanjutnya, buat dua catatan, satu untuk `privatelink-account-url` dan satu untuk`privatelink_ocsp-url`.

   - Di menu Zona yang Dihosting, pilih Buat Kumpulan Rekaman.

     a. Di bawah nama Rekam, masukkan ID Akun Snowflake Anda saja (8 karakter pertama di`privatelink-account-url`).

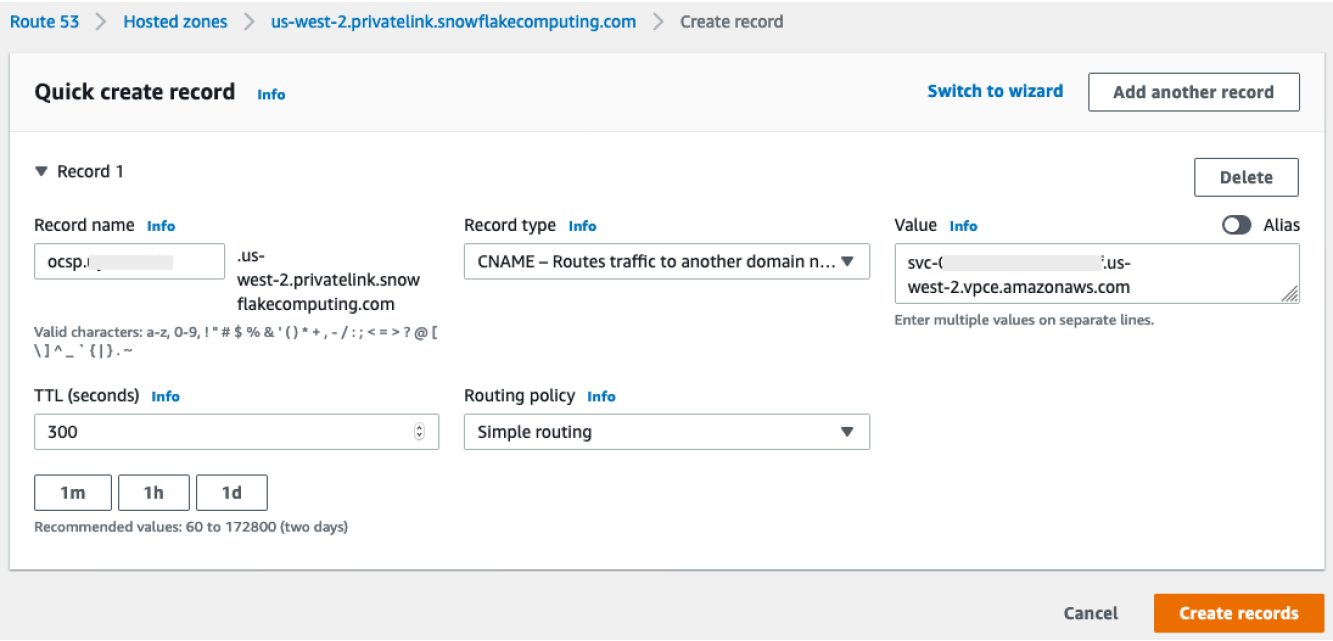
     b. Di bawah Jenis rekaman, pilih CNAME.

     c. Di bawah Nilai, masukkan nama DNS untuk titik akhir VPC regional yang Anda ambil pada langkah terakhir dari bagian Mengatur Integrasi Kepingan Salju. AWS PrivateLink



d. Pilih Create records (Buat catatan).

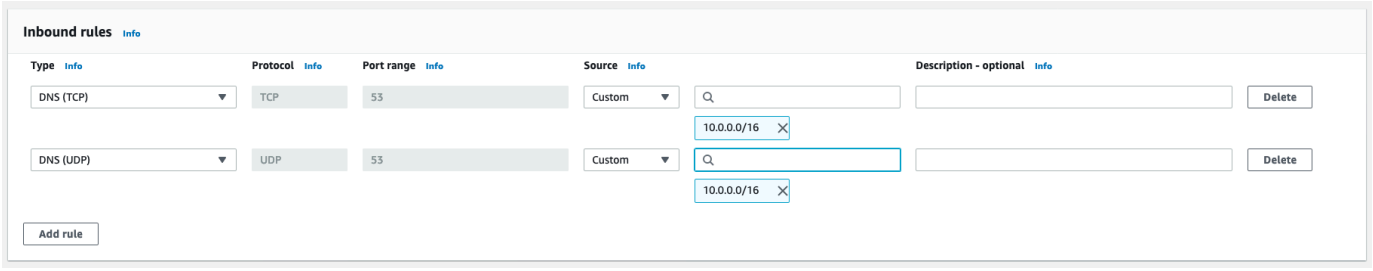
e. Ulangi langkah sebelumnya untuk catatan OCSP yang kami catat`privatelink-ocsp-url`, dimulai dengan `ocsp` melalui ID Snowflake 8 karakter untuk nama rekaman (seperti). `ocsp.xxxxxxxx`



Konfigurasikan Route 53 Resolver Inbound Endpoint untuk VPC Anda

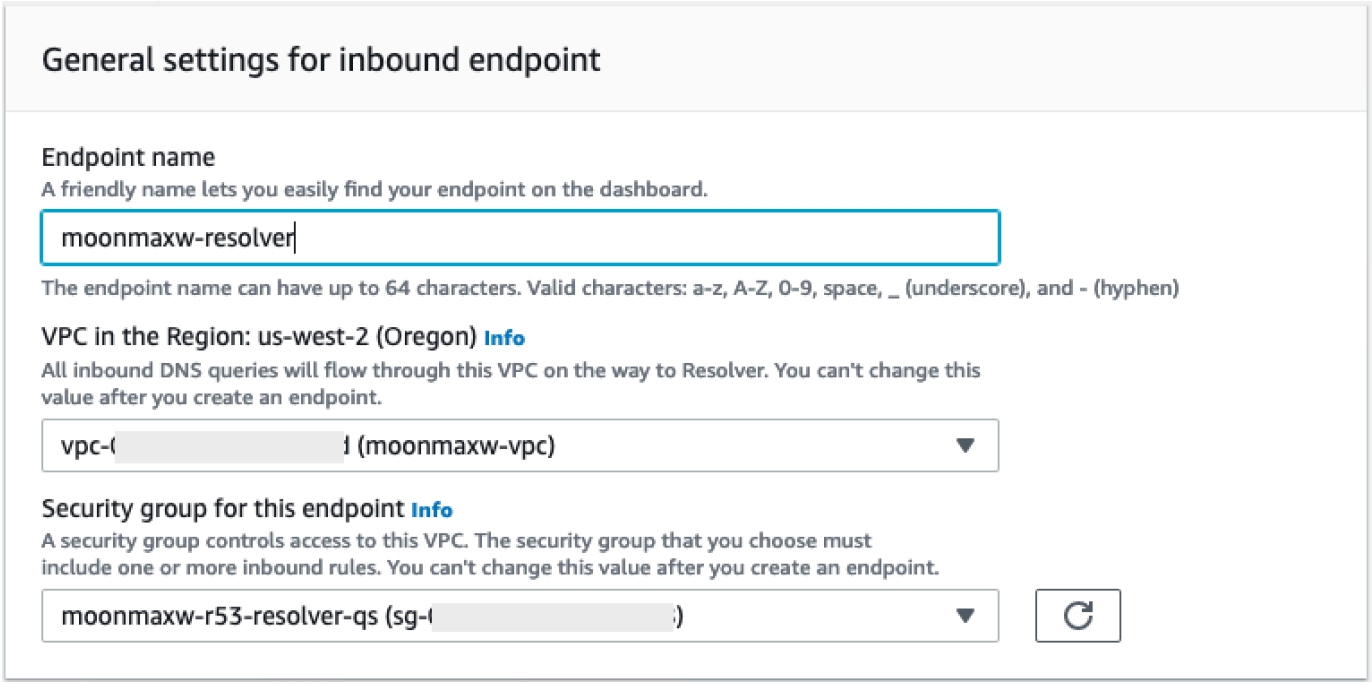
Bagian ini menjelaskan cara mengonfigurasi titik akhir inbound resolver Route 53 untuk VPC Anda.

1. Arahkan ke menu Route 53 di dalam AWS konsol Anda.
   - Di panel sebelah kiri di bagian Keamanan, pilih opsi Grup Keamanan.
2. Pilih Buat Grup Keamanan.
   - Berikan nama untuk grup keamanan Anda (seperti`datawranger-doc-route53-resolver-sg`) dan deskripsinya.
   - Pilih ID VPC yang digunakan pada langkah sebelumnya.
   - Buat aturan yang memungkinkan DNS melalui UDP dan TCP dari dalam blok VPC CIDR.



- Pilih Buat Grup Keamanan. Perhatikan ID Grup Keamanan karena menambahkan aturan untuk mengizinkan lalu lintas ke grup keamanan titik akhir VPC.

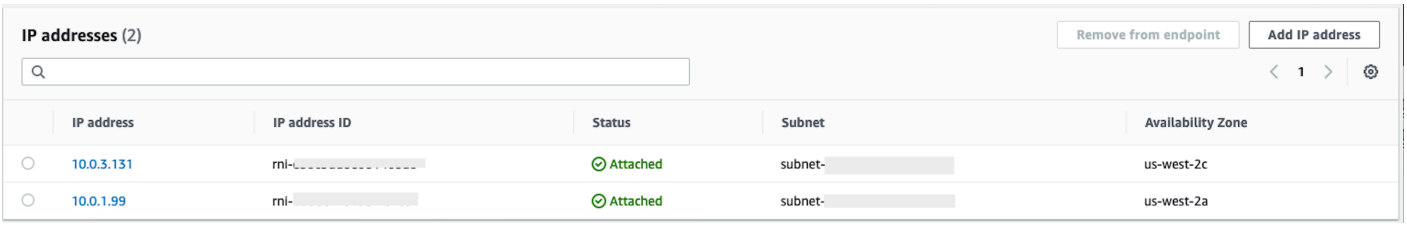
3. Arahkan ke menu Route 53 di dalam AWS konsol Anda.
   - Di bagian Resolver, pilih opsi Inbound Endpoint.
4. Pilih Buat Titik Akhir Masuk.
   - Berikan nama titik akhir.
   - Dari daftar tarik-turun VPC di Wilayah, pilih ID VPC yang telah Anda gunakan di semua langkah sebelumnya.
   - Dalam daftar dropdown grup Keamanan untuk titik akhir ini, pilih ID grup keamanan dari Langkah 2 di bagian ini.



   - Di bagian Alamat IP, pilih Availability Zones, pilih subnet, dan tinggalkan pemilih radio untuk Gunakan alamat IP yang dipilih secara otomatis dipilih untuk setiap alamat IP.

- Pilih Kirim.

5. Pilih titik akhir Inbound setelah dibuat.
6. Setelah titik akhir masuk dibuat, perhatikan dua alamat IP untuk resolver.



SageMaker Titik Akhir VPC

Bagian ini menjelaskan cara membuat titik akhir VPC untuk hal-hal berikut: Amazon SageMaker Studio Classic, SageMaker Notebooks, SageMaker API, Runtime Runtime, dan Amazon SageMaker Feature Store Runtime. SageMaker

Buat grup keamanan yang diterapkan ke semua titik akhir.

1. Arahkan ke menu EC2 di AWS Konsol.
2. Di bagian Jaringan & Keamanan, pilih opsi Grup keamanan.
3. Pilih Buat grup keamanan.
4. Berikan nama dan deskripsi grup keamanan (sepertidatawrangler-doc-sagemaker-vpce-sg). Aturan ditambahkan nanti untuk mengizinkan lalu lintas melalui HTTPS dari SageMaker grup ini.

Membuat titik akhir

1. Arahkan ke menu VPC di konsol. AWS
2. Pilih opsi Endpoints.
3. Pilih Buat Titik Akhir.
4. Cari layanan dengan memasukkan namanya di bidang Pencarian.
5. Dari daftar dropdown VPC, pilih VPC tempat koneksi Snowflake Anda ada. AWS PrivateLink
6. Di bagian Subnet, pilih subnet yang memiliki akses ke koneksi Snowflake PrivateLink .
7. Biarkan kotak centang Aktifkan Nama DNS dipilih.
8. Di bagian Grup Keamanan, pilih grup keamanan yang Anda buat di bagian sebelumnya.
9. Pilih Buat Titik Akhir.

Konfigurasikan Studio Classic dan Data Wrangler

Bagian ini menjelaskan cara mengonfigurasi Studio Classic dan Data Wrangler.

1. Konfigurasikan grup keamanan.
   a. Arahkan ke menu Amazon EC2 di Konsol. AWS
   b. Pilih opsi Grup Keamanan di bagian Jaringan & Keamanan.
   c. Pilih Buat Grup Keamanan.
   d. Berikan nama dan deskripsi untuk grup keamanan Anda (sepertidatawrangler-doc-sagemaker-studio).
   e. Buat aturan inbound berikut ini.
      - Koneksi HTTPS ke grup keamanan yang Anda sediakan untuk PrivateLink koneksi Snowflake yang Anda buat di langkah Atur Integrasi Kepingan Salju. PrivateLink
      - Koneksi HTTP ke grup keamanan yang Anda sediakan untuk PrivateLink koneksi Snowflake yang Anda buat di langkah Atur Integrasi Snowflake. PrivateLink
      - Grup keamanan UDP dan TCP untuk DNS (port 53) ke Route 53 Resolver Inbound Endpoint yang Anda buat di langkah 2 Konfigurasi Route 53 Resolver Inbound Endpoint untuk VPC Anda.
   f. Pilih tombol Create Security Group di pojok kanan bawah.
2. Konfigurasikan Studio Klasik.
   - Arahkan ke SageMaker menu di AWS konsol.
   - Dari konsol sebelah kiri, Pilih opsi SageMakerStudio Classic.
   - Jika Anda tidak memiliki domain yang dikonfigurasi, menu Memulai hadir.
   - Pilih opsi Pengaturan Standar dari menu Memulai.
   - Di bawah metode Authentication, pilih AWSIdentity and Access Management (IAM).
   - Dari menu Izin, Anda dapat membuat peran baru atau menggunakan peran yang sudah ada sebelumnya, tergantung pada kasus penggunaan Anda.
     - Jika Anda memilih Buat peran baru, Anda akan diberikan opsi untuk memberikan nama bucket S3, dan kebijakan dibuat untuk Anda.
     - Jika Anda sudah memiliki peran yang dibuat dengan izin untuk bucket S3 yang Anda perlukan akses, pilih peran dari daftar tarik-turun. Peran ini harus memiliki `AmazonSageMakerFullAccess` kebijakan yang melekat padanya.
   - Pilih daftar tarik-turun Jaringan dan Penyimpanan untuk mengonfigurasi penggunaan VPC, keamanan, dan subnet. SageMaker
     - Di bawah VPC, pilih VPC tempat koneksi Snowflake Anda ada. PrivateLink

    - Di bawah Subnet (s), pilih subnet yang memiliki akses ke koneksi Snowflake PrivateLink.
    - Di bawah Akses Jaringan untuk Studio Classic, pilih VPC Only.
    - Di bawah Grup Keamanan, pilih grup keamanan yang Anda buat di langkah 1.
  - Pilih Kirim.
3. Edit grup SageMaker keamanan.
  - Buat aturan inbound berikut:
    - Port 2049 ke Grup Keamanan NFS masuk dan keluar yang dibuat secara otomatis SageMaker pada langkah 2 (nama grup keamanan berisi ID domain Studio Classic).
    - Akses ke semua port TCP ke dirinya sendiri (diperlukan SageMaker untuk VPC Saja).
4. Edit Grup Keamanan Titik Akhir VPC:
  - Arahkan ke menu Amazon EC2 di konsol. AWS
  - Temukan grup keamanan yang Anda buat pada langkah sebelumnya.
  - Tambahkan aturan masuk yang memungkinkan lalu lintas HTTPS dari grup keamanan yang dibuat pada langkah 1.
5. Buat profil pengguna.
  - Dari Panel Kontrol Klasik SageMaker Studio, pilih Tambah Pengguna.
  - Berikan nama pengguna.
  - Untuk Peran Eksekusi, pilih untuk membuat peran baru atau gunakan peran yang sudah ada sebelumnya.
    - Jika memilih Buat peran baru, Anda akan diberikan opsi untuk memberikan nama bucket Amazon S3, dan kebijakan dibuat untuk Anda.
    - Jika Anda sudah memiliki peran yang dibuat dengan izin ke bucket Amazon S3 yang memerlukan akses, pilih peran dari daftar tarik-turun. Peran ini harus memiliki `AmazonSageMakerFullAccess` kebijakan yang melekat padanya.
  - Pilih Kirim.
6. Buat aliran data (ikuti panduan ilmuwan data yang diuraikan di bagian sebelumnya).
  - Saat menambahkan koneksi Snowflake, masukkan nilai `privatelink-account-name` (dari langkah Set Up Snowflake PrivateLink Integration) ke bidang nama akun Snowflake (alfanumerik), bukan nama akun Snowflake biasa. Segala sesuatu yang lain dibiarkan tidak berubah.

Memberikan informasi kepada ilmuwan data

Berikan ilmuwan data informasi yang mereka butuhkan untuk mengakses Snowflake dari Amazon SageMaker Data Wrangler.

> **Important**
>
> Pengguna Anda harus menjalankan Amazon SageMaker Studio Classic versi 1.3.0 atau yang lebih baru. Untuk informasi tentang memeriksa versi Studio Classic dan memperbaruinya, lihatSiapkan Data ML dengan Amazon SageMaker Data Wrangler.

1. Untuk memungkinkan ilmuwan data Anda mengakses Snowflake dari SageMaker Data Wrangler, berikan mereka salah satu dari berikut ini:

   - Untuk Otentikasi Dasar, nama akun Snowflake, nama pengguna, dan kata sandi.
   - Untuk OAuth, nama pengguna dan kata sandi di penyedia identitas.
   - Untuk ARN, Secrets Manager merahasiakan Amazon Resource Name (ARN).
   - Rahasia yang dibuat dengan AWSSecrets Manager dan ARN of the secret. Gunakan prosedur berikut di bawah ini untuk membuat rahasia Snowflake jika Anda memilih opsi ini.

     > **Important**
     >
     > Jika ilmuwan data Anda menggunakan opsi Snowflake Credentials (Nama pengguna dan Kata Sandi) untuk terhubung ke Snowflake, Anda dapat menggunakan Secrets Manager untuk menyimpan kredensialnya secara rahasia. Secrets Manager memutar rahasia sebagai bagian dari rencana keamanan praktik terbaik. Rahasia yang dibuat di Secrets Manager hanya dapat diakses dengan peran Studio Classic yang dikonfigurasi saat Anda menyiapkan profil pengguna Studio Classic. Ini mengharuskan Anda untuk menambahkan izin ini`secretsmanager:PutResourcePolicy`,, ke kebijakan yang dilampirkan ke peran Studio Classic Anda.
     > Kami sangat menyarankan agar Anda membuat cakupan kebijakan peran untuk menggunakan peran yang berbeda untuk grup pengguna Studio Classic yang berbeda. Anda dapat menambahkan izin berbasis sumber daya tambahan untuk rahasia Secrets Manager. Lihat Mengelola Kebijakan Rahasia untuk kunci kondisi yang dapat Anda gunakan.

Untuk informasi tentang membuat rahasia, lihat Membuat rahasia. Anda dikenakan biaya untuk rahasia yang Anda buat.

2. (Opsional) Berikan nama integrasi penyimpanan kepada ilmuwan data yang Anda buat menggunakan prosedur berikut Buat Integrasi Penyimpanan Cloud di Snowflake. Ini adalah nama integrasi baru dan dipanggil `integration_name` dalam perintah `CREATE INTEGRATION` SQL yang Anda jalankan, yang ditunjukkan dalam cuplikan berikut:

```
CREATE STORAGE INTEGRATION integration_name
TYPE = EXTERNAL_STAGE
STORAGE_PROVIDER = S3
ENABLED = TRUE
STORAGE_AWS_ROLE_ARN = 'iam_role'
[ STORAGE_AWS_OBJECT_ACL = 'bucket-owner-full-control' ]
STORAGE_ALLOWED_LOCATIONS = ('s3://bucket/path/', 's3://bucket/path/')
[ STORAGE_BLOCKED_LOCATIONS = ('s3://bucket/path/', 's3://bucket/path/') ]
```

Panduan Ilmuwan Data

Gunakan yang berikut ini untuk menghubungkan Snowflake dan mengakses data Anda di Data Wrangler.

Important

Administrator Anda perlu menggunakan informasi di bagian sebelumnya untuk mengatur Snowflake. Jika Anda mengalami masalah, hubungi mereka untuk bantuan pemecahan masalah.

Anda dapat terhubung ke Snowflake dengan salah satu cara berikut:

- Menentukan kredenal Snowflake Anda (nama akun, nama pengguna, dan kata sandi) di Data Wrangler.
- Menyediakan Amazon Resource Name (ARN) dari rahasia yang berisi kredensi.
- Menggunakan penyedia open standard for access delegation (OAuth) yang terhubung ke Snowflake. Administrator Anda dapat memberi Anda akses ke salah satu penyedia OAuth berikut:

- Azure AD
- Okta
- Federasi Ping

Bicaralah dengan administrator Anda tentang metode yang perlu Anda gunakan untuk terhubung ke Snowflake.

Bagian berikut memiliki informasi tentang bagaimana Anda dapat terhubung ke Snowflake menggunakan metode sebelumnya.

Specifying your Snowflake Credentials

Untuk mengimpor dataset ke Data Wrangler dari Snowflake menggunakan kredensi Anda

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Pilih Impor data.
9. Di bawah Tersedia, pilih Kepingan Salju.
10. Untuk nama Koneksi, tentukan nama yang secara unik mengidentifikasi koneksi.
11. Untuk metode Authentication, pilih Basic Username-Password.
12. Untuk nama akun Snowflake (alfanumerik), tentukan nama lengkap akun Snowflake.
13. Untuk Nama Pengguna, tentukan nama pengguna yang Anda gunakan untuk mengakses akun Snowflake.
14. Untuk Kata Sandi, tentukan kata sandi yang terkait dengan nama pengguna.
15. (Opsional) Untuk pengaturan lanjutan. tentukan yang berikut ini:

    - Peran — Peran dalam Snowflake. Beberapa peran memiliki akses ke kumpulan data yang berbeda. Jika Anda tidak menentukan peran, Data Wrangler menggunakan peran default di akun Snowflake Anda.

- Integrasi penyimpanan — Saat Anda menentukan dan menjalankan kueri, Data Wrangler membuat salinan sementara hasil kueri dalam memori. Untuk menyimpan salinan permanen hasil kueri, tentukan lokasi Amazon S3 untuk integrasi penyimpanan. Administrator Anda memberi Anda URI S3.
- ID kunci KMS — Kunci KMS yang telah Anda buat. Anda dapat menentukan ARN untuk mengenkripsi output dari kueri Snowflake. Jika tidak, Data Wrangler menggunakan enkripsi default.

16. Pilih Hubungkan.

Providing an Amazon Resource Name (ARN)

Untuk mengimpor dataset ke Data Wrangler dari Snowflake menggunakan ARN

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Pilih Impor data.
9. Di bawah Tersedia, pilih Kepingan Salju.
10. Untuk nama Koneksi, tentukan nama yang secara unik mengidentifikasi koneksi.
11. Untuk metode otentikasi, pilih ARN.
12. Secrets Manager ARN — ARN AWS Secrets Manager rahasia yang digunakan untuk menyimpan kredensyal yang digunakan untuk terhubung ke Snowflake.
13. (Opsional) Untuk pengaturan lanjutan. tentukan yang berikut ini:

    - Peran — Peran dalam Snowflake. Beberapa peran memiliki akses ke kumpulan data yang berbeda. Jika Anda tidak menentukan peran, Data Wrangler menggunakan peran default di akun Snowflake Anda.
    - Integrasi penyimpanan — Saat Anda menentukan dan menjalankan kueri, Data Wrangler membuat salinan sementara hasil kueri dalam memori. Untuk menyimpan

salinan permanen hasil kueri, tentukan lokasi Amazon S3 untuk integrasi penyimpanan. Administrator Anda memberi Anda URI S3.

- ID kunci KMS — Kunci KMS yang telah Anda buat. Anda dapat menentukan ARN untuk mengenkripsi output dari kueri Snowflake. Jika tidak, Data Wrangler menggunakan enkripsi default.

14. Pilih Hubungkan.

Using an OAuth Connection

> **Important**
>
> Administrator Anda menyesuaikan lingkungan Studio Classic Anda untuk menyediakan fungsionalitas yang Anda gunakan untuk menggunakan koneksi OAuth. Anda mungkin perlu memulai ulang aplikasi server Jupyter untuk menggunakan fungsionalitas tersebut. Gunakan prosedur berikut untuk memperbarui aplikasi server Jupyter.
>
> 1. Dalam Studio Classic, pilih File
> 2. Pilih Shut down.
> 3. Pilih Shut down server.
> 4. Tutup tab atau jendela yang Anda gunakan untuk mengakses Studio Classic.
> 5. Dari SageMaker konsol Amazon, buka Studio Classic.

Untuk mengimpor dataset ke Data Wrangler dari Snowflake menggunakan kredensi Anda

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Pilih Impor data.
9. Di bawah Tersedia, pilih Kepingan Salju.

10. Untuk nama Koneksi, tentukan nama yang secara unik mengidentifikasi koneksi.
11. Untuk metode otentikasi, pilih OAuth.
12. (Opsional) Untuk pengaturan lanjutan. tentukan yang berikut ini:
    - Peran — Peran dalam Snowflake. Beberapa peran memiliki akses ke kumpulan data yang berbeda. Jika Anda tidak menentukan peran, Data Wrangler menggunakan peran default di akun Snowflake Anda.
    - Integrasi penyimpanan — Saat Anda menentukan dan menjalankan kueri, Data Wrangler membuat salinan sementara hasil kueri dalam memori. Untuk menyimpan salinan permanen hasil kueri, tentukan lokasi Amazon S3 untuk integrasi penyimpanan. Administrator Anda memberi Anda URI S3.
    - ID kunci KMS — Kunci KMS yang telah Anda buat. Anda dapat menentukan ARN untuk mengenkripsi output dari kueri Snowflake. Jika tidak, Data Wrangler menggunakan enkripsi default.
13. Pilih Hubungkan.

Anda dapat memulai proses mengimpor data Anda dari Snowflake setelah Anda terhubung dengannya.

Dalam Data Wrangler, Anda dapat melihat gudang data, database, dan skema Anda, bersama dengan ikon mata yang dapat digunakan untuk melihat pratinjau tabel Anda. Setelah Anda memilih ikon Tabel Pratinjau, pratinjau skema tabel tersebut dihasilkan. Anda harus memilih gudang sebelum Anda dapat melihat pratinjau tabel.

> **Important**
>
> Jika Anda mengimpor dataset dengan kolom jenis `TIMESTAMP_TZ` atau`TIMESTAMP_LTZ`, tambahkan `::string` ke nama kolom kueri Anda. Untuk informasi selengkapnya, lihat Cara: Membongkar data TIMESTAMP_TZ dan TIMESTAMP_LTZ ke file Parket.

Setelah Anda memilih gudang data, database dan skema, Anda sekarang dapat menulis kueri dan menjalankannya. Output kueri Anda ditampilkan di bawah Hasil kueri.

Setelah Anda menyelesaikan output kueri Anda, Anda kemudian dapat mengimpor output kueri Anda ke dalam aliran Data Wrangler untuk melakukan transformasi data.

Setelah mengimpor data, navigasikan ke alur Data Wrangler Anda dan mulailah menambahkan transformasi ke dalamnya. Untuk daftar transformasi yang tersedia, lihatMemindahkan.

Impor Data Dari Perangkat Lunak sebagai Platform Layanan (SaaS)

Anda dapat menggunakan Data Wrangler untuk mengimpor data dari lebih dari empat puluh platform perangkat lunak sebagai layanan (SaaS). Untuk mengimpor data Anda dari platform SaaS Anda, Anda atau administrator Anda harus menggunakan Amazon AppFlow untuk mentransfer data dari platform ke Amazon S3 atau Amazon Redshift. Untuk informasi selengkapnya tentang Amazon AppFlow, lihat Apa itu Amazon AppFlow? Jika Anda tidak perlu menggunakan Amazon Redshift, kami sarankan untuk mentransfer data ke Amazon S3 untuk proses yang lebih sederhana.

Data Wrangler mendukung transfer data dari platform SaaS berikut:

- Amplitudo
- Asana
- Braintree
- CircleCI
- DocuSign Monitor
- Senang
- Domo
- Datadog
- Dynatrace
- Iklan Facebook
- Wawasan Halaman Facebook
- Iklan Google
- Google Analytics 4
- Google Kalender
- Konsol Penelusuran Google
- GitHub
- GitLab
- Infor Nexus
- Iklan Instagram

- Interkom
- JDBC (Sinkronisasi)
- Awan Jira
- LinkedIn Iklan
- MailChimp
- Marketo
- Microsoft Dynamics 365
- Tim Microsoft
- Mixpanel
- Okta
- Oracle HCM
- Paypal Checkout
- Pendo
- Salesforce
- Cloud Pemasaran Salesforce
- Salesforce Pardot
- SAP OData
- SendGrid
- ServiceNow
- Tunggal
- Kendur
- Smartsheet
- Iklan Snapchat
- Garis
- Tren Mikro
- Jenis huruf
- Veeva
- WooCommerce
- Zendesk
- Obrolan Zendesk

- Jual Zendesk
- Sinar Matahari Zendesk
- Zoho CRM
- Pertemuan Zoom

Daftar sebelumnya memiliki tautan ke informasi lebih lanjut tentang pengaturan sumber data Anda. Anda atau administrator Anda dapat merujuk ke tautan sebelumnya setelah Anda membaca informasi berikut.

Saat Anda menavigasi ke tab Impor aliran Data Wrangler Anda, Anda melihat sumber data di bawah bagian berikut:

- Available
- Siapkan sumber data

Anda dapat terhubung ke sumber data di bawah Tersedia tanpa memerlukan konfigurasi tambahan. Anda dapat memilih sumber data dan mengimpor data Anda.

Sumber data di bawah Mengatur sumber data, mengharuskan Anda atau administrator Anda menggunakan Amazon AppFlow untuk mentransfer data dari platform SaaS ke Amazon S3 atau Amazon Redshift. Untuk informasi tentang melakukan transfer, lihatMenggunakan Amazon AppFlow untuk mentransfer data Anda.

Setelah Anda melakukan transfer data, platform SaaS muncul sebagai sumber data di bawah Tersedia. Anda dapat memilihnya dan mengimpor data yang telah Anda transfer ke Data Wrangler. Data yang Anda transfer muncul sebagai tabel yang dapat Anda kueri.

Menggunakan Amazon AppFlow untuk mentransfer data Anda

Amazon AppFlow adalah platform yang dapat Anda gunakan untuk mentransfer data dari platform SaaS Anda ke Amazon S3 atau Amazon Redshift tanpa harus menulis kode apa pun. Untuk melakukan transfer data, Anda menggunakan fileAWS Management Console.

> **Important**
>
> Anda harus memastikan bahwa Anda telah mengatur izin untuk melakukan transfer data. Untuk informasi selengkapnya, lihat AppFlow Izin Amazon.

Setelah menambahkan izin, Anda dapat mentransfer data. Di Amazon AppFlow, Anda membuat alur untuk mentransfer data. Aliran adalah serangkaian konfigurasi. Anda dapat menggunakannya untuk menentukan apakah Anda menjalankan transfer data sesuai jadwal atau apakah Anda mempartisi data menjadi file terpisah. Setelah mengkonfigurasi alur, Anda menjalankannya untuk mentransfer data.

Untuk informasi tentang membuat alur, lihat Membuat alur di Amazon AppFlow. Untuk informasi tentang menjalankan alur, lihat Mengaktifkan AppFlow aliran Amazon.

Setelah data ditransfer, gunakan prosedur berikut untuk mengakses data di Data Wrangler.

Important

Sebelum Anda mencoba mengakses data Anda, pastikan peran IAM Anda memiliki kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "glue:SearchTables",
            "Resource": [
                "arn:aws:glue:*:*:table/*/*",
                "arn:aws:glue:*:*:database/*",
                "arn:aws:glue:*:*:catalog"
            ]
        }
    ]
}
```

Secara default, peran IAM yang Anda gunakan untuk mengakses Data Wrangler adalah. `SageMakerExecutionRole` Untuk informasi selengkapnya tentang menambahkan kebijakan, lihat Menambahkan izin identitas IAM (konsol).

Untuk terhubung ke sumber data, lakukan hal berikut.

1. Masuk ke SageMakerKonsol Amazon.

2. Pilih Studio.
3. Pilih Luncurkan aplikasi.
4. Dari daftar dropdown, pilih Studio.
5. Pilih ikon Beranda.
6. Pilih Data.
7. Pilih Data Wrangler.
8. Pilih Impor data.
9. Di bawah Tersedia, pilih sumber data.
10. Untuk bidang Nama, tentukan nama koneksi.
11. (Opsional) Pilih Konfigurasi lanjutan.
    a. Pilih Workgroup.
    b. Jika grup kerja Anda belum menerapkan lokasi keluaran Amazon S3 atau jika Anda tidak menggunakan grup kerja, tentukan nilai untuk lokasi hasil kueri Amazon S3.
    c. (Opsional) Untuk periode penyimpanan data, pilih kotak centang untuk mengatur periode penyimpanan data dan tentukan jumlah hari untuk menyimpan data sebelum dihapus.
    d. (Opsional) Secara default, Data Wrangler menyimpan koneksi. Anda dapat memilih untuk membatalkan pilihan kotak centang dan tidak menyimpan koneksi.
12. Pilih Hubungkan.
13. Tentukan kueri.

> **Note**
>
> Untuk membantu Anda menentukan kueri, Anda dapat memilih tabel di panel navigasi sebelah kiri. Data Wrangler menunjukkan nama tabel dan pratinjau tabel. Pilih ikon di samping nama tabel untuk menyalin nama. Anda dapat menggunakan nama tabel di kueri.

14. Pilih Jalankan.
15. Pilih kueri Impor.
16. Untuk nama Dataset, tentukan nama dataset.
17. Pilih Tambahkan.

Saat Anda menavigasi ke layar Impor data, Anda dapat melihat koneksi yang telah Anda buat. Anda dapat menggunakan koneksi untuk mengimpor lebih banyak data.

Penyimpanan Data yang Diimpor

> **Important**
>
> Kami sangat menyarankan agar Anda mengikuti praktik terbaik seputar melindungi bucket Amazon S3 Anda dengan mengikuti praktik terbaik Keamanan.

Saat Anda menanyakan data dari Amazon Athena atau Amazon Redshift, kumpulan data yang ditanyakan akan disimpan secara otomatis di Amazon S3. Data disimpan di bucket SageMaker S3 default untuk AWS Wilayah tempat Anda menggunakan Studio Classic.

Bucket S3 default memiliki konvensi penamaan berikut: `sagemaker-`*`region`*`-`*`account number` Misalnya, jika nomor akun Anda 111122223333 dan Anda menggunakan Studio Classic in, kumpulan data yang diimpor akan disimpan di `us-east-1` 111122223333. `sagemaker-us-east-1-`

Alur Data Wrangler bergantung pada lokasi kumpulan data Amazon S3 ini, jadi Anda tidak boleh memodifikasi kumpulan data ini di Amazon S3 saat Anda menggunakan aliran dependen. Jika Anda memodifikasi lokasi S3 ini, dan Anda ingin terus menggunakan aliran data Anda, Anda harus menghapus semua objek `trained_parameters` dalam file.flow Anda. Untuk melakukan ini, unduh file.flow dari Studio Classic dan untuk setiap instance`trained_parameters`, hapus semua entri. Ketika Anda selesai, `trained_parameters` harus menjadi objek JSON kosong:

```
"trained_parameters": {}
```

Saat Anda mengekspor dan menggunakan aliran data untuk memproses data, file.flow yang Anda ekspor merujuk ke kumpulan data ini di Amazon S3. Gunakan bagian berikut untuk mempelajari lebih banyak.

Penyimpanan Impor Amazon Redshift

Data Wrangler menyimpan kumpulan data yang dihasilkan dari kueri Anda dalam file Parket di bucket S3 default Anda. SageMaker

File ini disimpan di bawah awalan berikut (direktori): redshift/*`uuid/data/, di mana uuid`* adalah pengidentifikasi unik yang dibuat untuk *`setiap kueri`*.

Misalnya, jika bucket default Anda, satu kumpulan data yang ditanyakan dari Amazon Redshift terletak di s3://-1-111122223333/redshift/ uuid /data/. sagemaker-us-east-1-111122223333 sagemaker-us-east

Penyimpanan Impor Amazon Athena

Saat Anda menanyakan database Athena dan mengimpor kumpulan data, Data Wrangler menyimpan kumpulan data, serta subset dari kumpulan data tersebut, atau file pratinjau, di Amazon S3.

Dataset yang Anda impor dengan memilih Impor dataset disimpan dalam format Parket di Amazon S3.

File pratinjau ditulis dalam format CSV saat Anda memilih Jalankan di layar impor Athena, dan berisi hingga 100 baris dari kumpulan data yang Anda kueri.

Dataset yang Anda kueri terletak di bawah awalan (direktori): athena/ uuid /data/, di mana uuid adalah pengidentifikasi unik yang dibuat untuk setiap kueri.

Misalnya, jika bucket default Anda, satu set data yang ditanyakan dari Athena terletak di /athena/ uuid s3://sagemaker-us-east-1-111122223333 / data/ example_dataset.parquet. sagemaker-us-east-1-111122223333

Subset dari kumpulan data yang disimpan untuk melihat pratinjau kerangka data di Data Wrangler disimpan di bawah awalan: athena/.

Membuat dan Menggunakan Data Wrangler Flow

Gunakan alur Amazon SageMaker Data Wrangler, atau aliran data, untuk membuat dan memodifikasi pipeline persiapan data. Aliran data menghubungkan kumpulan data, transformasi, dan analisis, atau langkah, yang Anda buat dan dapat digunakan untuk menentukan pipeline Anda.

Instans

Saat Anda membuat alur Data Wrangler di Amazon SageMaker Studio Classic, Data Wrangler menggunakan instans Amazon EC2 untuk menjalankan analisis dan transformasi dalam alur Anda. Secara default, Data Wrangler menggunakan instance m5.4xlarge. instance m5 adalah instance tujuan umum yang memberikan keseimbangan antara komputasi dan memori. Anda dapat menggunakan instans m5 untuk berbagai beban kerja komputasi.

Data Wrangler juga memberi Anda opsi untuk menggunakan instans r5. Instans r5 dirancang untuk memberikan kinerja cepat yang memproses kumpulan data besar dalam memori.

Kami menyarankan Anda memilih instance yang paling baik dioptimalkan di sekitar beban kerja Anda. Misalnya, r5.8xlarge mungkin memiliki harga yang lebih tinggi daripada m5.4xlarge, tetapi r5.8xlarge mungkin lebih baik dioptimalkan untuk beban kerja Anda. Dengan instans yang dioptimalkan dengan lebih baik, Anda dapat menjalankan aliran data dalam waktu yang lebih singkat dengan biaya lebih rendah.

Tabel berikut menunjukkan instans yang dapat Anda gunakan untuk menjalankan alur Data Wrangler Anda.

Instans	vCPU	Memori
ml.m5.4xlarge	16	64 GiB
ml.m5.8xlarge	32	128 GiB
ml.m5.16xlarge	64	256 GiB
ml.m5.24xlarge	96	384 GiB
r5.4xlarge	16	128 GiB
r5.8xlarge	32	256 GiB
r5.24xlarge	96	768 GiB

Untuk informasi selengkapnya tentang instans r5, lihat Amazon EC2 R5 Instance. Untuk informasi selengkapnya tentang instans m5, lihat Amazon EC2 M5 Instance.

Setiap alur Data Wrangler memiliki instans Amazon EC2 yang terkait dengannya. Anda mungkin memiliki beberapa aliran yang terkait dengan satu instance.

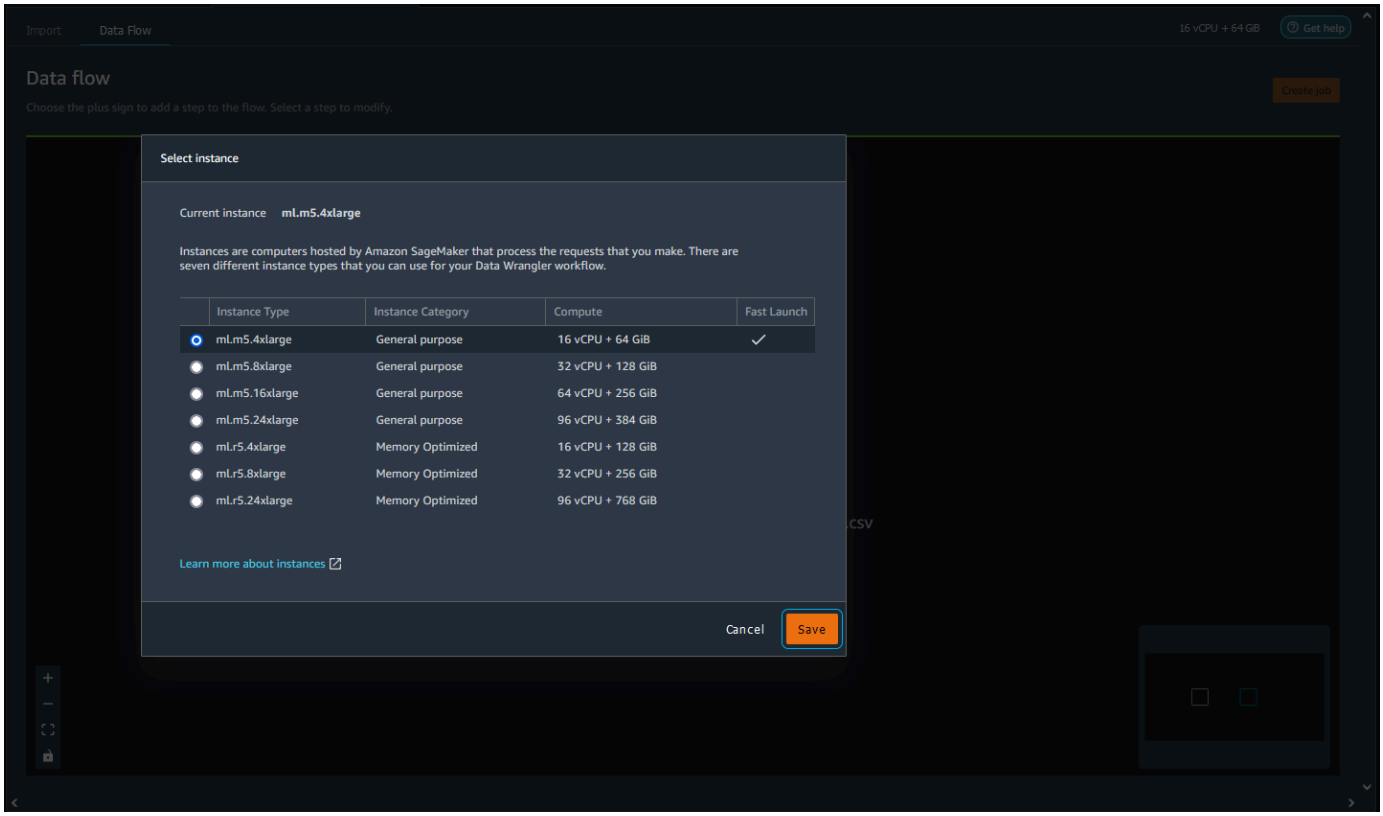
Untuk setiap file aliran, Anda dapat mengganti jenis instans dengan mulus. Jika Anda mengganti jenis instance, instance yang Anda gunakan untuk menjalankan flow terus berjalan.

Untuk mengganti jenis instans aliran Anda, lakukan hal berikut.

1. Pilih ikon

   rumah,

2. Arahkan ke instance yang Anda gunakan dan pilih.
3. Pilih tipe instans yang ingin Anda gunakan.

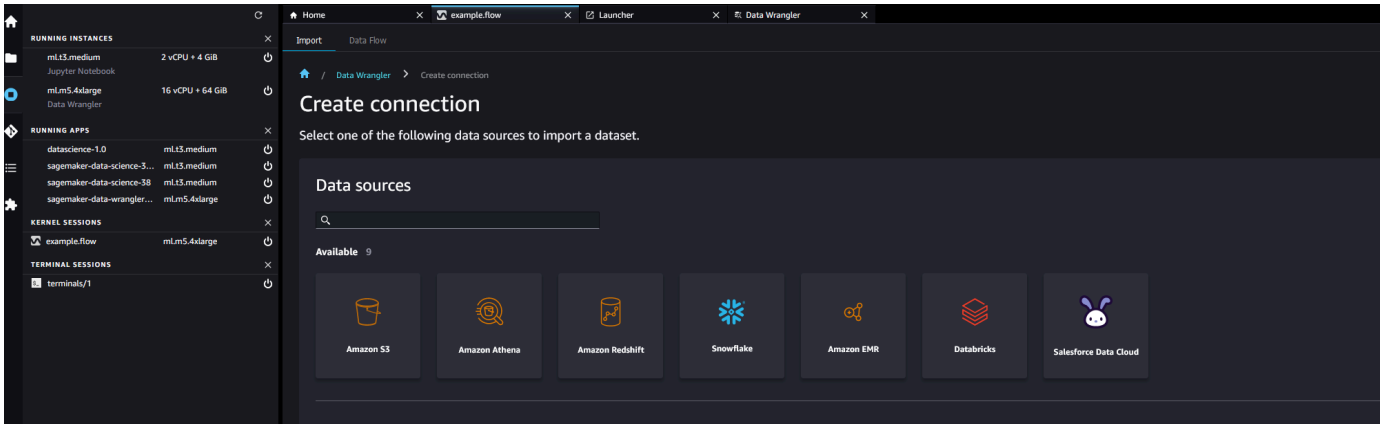


4. Pilih Simpan.

Anda dikenai biaya untuk kedua instans yang berjalan. Untuk menghindari biaya tambahan, matikan instance yang tidak Anda gunakan secara manual. Untuk mematikan instans yang sedang berjalan, gunakan prosedur berikut.

Untuk mematikan instance yang sedang berjalan.

1. Pilih ikon instance. Gambar berikut menunjukkan tempat untuk memilih ikon RUNNING INSTANCES.



2. Pilih Matikan di samping instans yang ingin Anda matikan.

Jika Anda mematikan instance yang digunakan untuk menjalankan aliran, Anda tidak dapat mengakses aliran untuk sementara. Jika Anda mendapatkan kesalahan saat mencoba membuka alur yang menjalankan instance yang sebelumnya Anda matikan, tunggu selama 5 menit dan coba buka lagi.

Saat Anda mengekspor aliran data ke lokasi seperti Amazon Simple Storage Service atau Amazon SageMaker Feature Store, Data Wrangler menjalankan pekerjaan SageMaker pemrosesan Amazon. Anda dapat menggunakan salah satu contoh berikut untuk pekerjaan pemrosesan. Untuk informasi lebih lanjut tentang mengekspor data Anda, lihatEkspor.

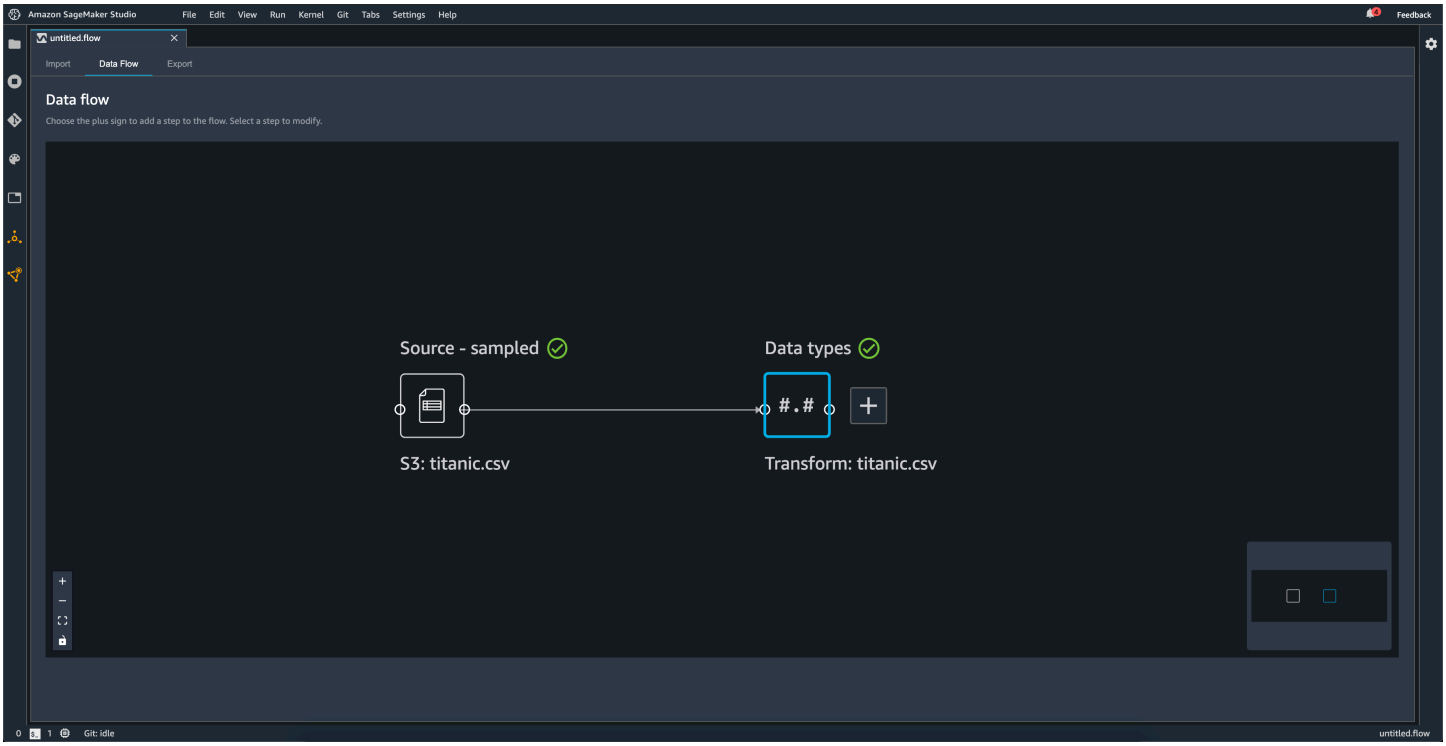
Instans	vCPU	Memori
ml.m5.4xlarge	16	64 GiB
ml.m5.12xlarge	48	192 GiB
ml.m5.24xlarge	96	384 GiB

Untuk informasi selengkapnya tentang biaya per jam untuk menggunakan jenis instans yang tersedia, lihat SageMaker Harga.

Alur

Saat Anda mengimpor dataset, dataset asli muncul di aliran data dan diberi nama Source. Jika Anda mengaktifkan pengambilan sampel saat mengimpor data, kumpulan data ini diberi nama Sumber - sampel. Data Wrangler secara otomatis menyimpulkan jenis setiap kolom dalam kumpulan data

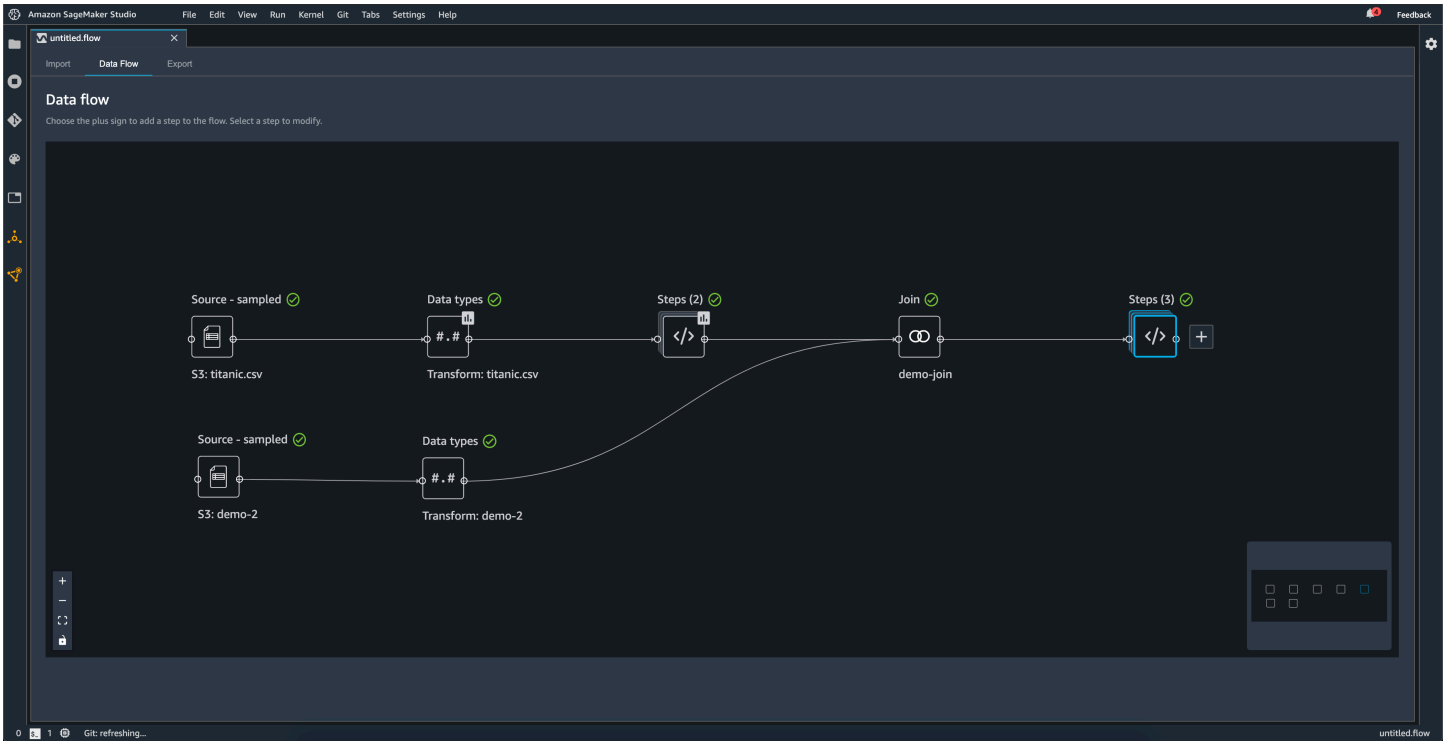
Anda dan membuat kerangka data baru bernama Tipe data. Anda dapat memilih bingkai ini untuk memperbarui tipe data yang disimpulkan. Anda melihat hasil yang mirip dengan yang ditampilkan pada gambar berikut setelah mengunggah satu set data:



Setiap kali Anda menambahkan langkah transformasi, Anda membuat kerangka data baru. Ketika beberapa langkah transformasi (selain Join atau Concatenate) ditambahkan ke kumpulan data yang sama, mereka ditumpuk.

Bergabunglah dan Gabungkan buat langkah mandiri yang berisi kumpulan data baru yang digabungkan atau digabungkan.

Diagram berikut menunjukkan aliran data dengan gabungan antara dua kumpulan data, serta dua tumpukan langkah. Tumpukan pertama (Langkah (2)) menambahkan dua transformasi ke jenis yang disimpulkan dalam kumpulan data tipe Data. Tumpukan hilir, atau tumpukan di sebelah kanan, menambahkan transformasi ke kumpulan data yang dihasilkan dari gabungan bernama demo-join.



Kotak abu-abu kecil di sudut kanan bawah aliran data memberikan gambaran umum tentang jumlah tumpukan dan langkah dalam aliran dan tata letak aliran. Kotak yang lebih terang di dalam kotak abu-abu menunjukkan langkah-langkah yang ada dalam tampilan UI. Anda dapat menggunakan kotak ini untuk melihat bagian aliran data yang berada di luar tampilan UI. Gunakan ikon layar fit
(⛶)
agar sesuai dengan semua langkah dan kumpulan data ke dalam tampilan UI Anda.

Bilah navigasi kiri bawah menyertakan ikon yang dapat Anda gunakan untuk memperbesar
(+)
dan keluar
(—)
aliran data Anda dan mengubah ukuran aliran data agar sesuai dengan layar
(⛶).
Gunakan ikon kunci
(🔓)
untuk mengunci dan membuka kunci lokasi setiap langkah di layar.

Tambahkan Langkah ke Alur Data Anda

Pilih + di sebelah kumpulan data apa pun atau langkah yang ditambahkan sebelumnya, lalu pilih salah satu opsi berikut:

- Mengedit tipe data (Hanya untuk langkah tipe data): Jika Anda belum menambahkan transformasi apa pun ke langkah Jenis data, Anda dapat memilih Edit tipe data untuk memperbarui tipe data yang disimpulkan oleh Wrangler Data saat mengimpor kumpulan data Anda.
- Tambahkan transformasi: Menambahkan langkah transformasi baru. Lihat Memindahkan untuk mempelajari lebih lanjut tentang transformasi data yang dapat Anda tambahkan.
- Tambahkan analisis: Menambahkan analisis. Anda dapat menggunakan opsi ini untuk menganalisis data Anda kapan saja dalam aliran data. Ketika Anda menambahkan satu atau lebih analisis ke langkah, ikon analisis () muncul pada langkah itu. Lihat Analisis dan Memvisualisasikan untuk mempelajari lebih lanjut tentang analisis yang dapat Anda tambahkan.
- Gabung: Bergabung dengan dua kumpulan data dan menambahkan kumpulan data yang dihasilkan ke aliran data. Untuk mempelajari selengkapnya, lihat Set data.
- Menggabungkan: Menggabungkan dua kumpulan data dan menambahkan kumpulan data yang dihasilkan ke aliran data. Untuk mempelajari selengkapnya, lihat Set Data Penggabungan.

Hapus Langkah dari Alur Data Anda

Untuk menghapus langkah, pilih langkah dan pilih Hapus. Jika node adalah node yang memiliki input tunggal, Anda hanya menghapus langkah yang Anda pilih. Menghapus langkah yang memiliki satu input tidak menghapus langkah-langkah yang mengikutinya. Jika Anda menghapus langkah untuk sumber, bergabung, atau menggabungkan node, semua langkah yang mengikutinya juga dihapus.

Untuk menghapus langkah dari tumpukan langkah, pilih tumpukan dan kemudian pilih langkah yang ingin Anda hapus.

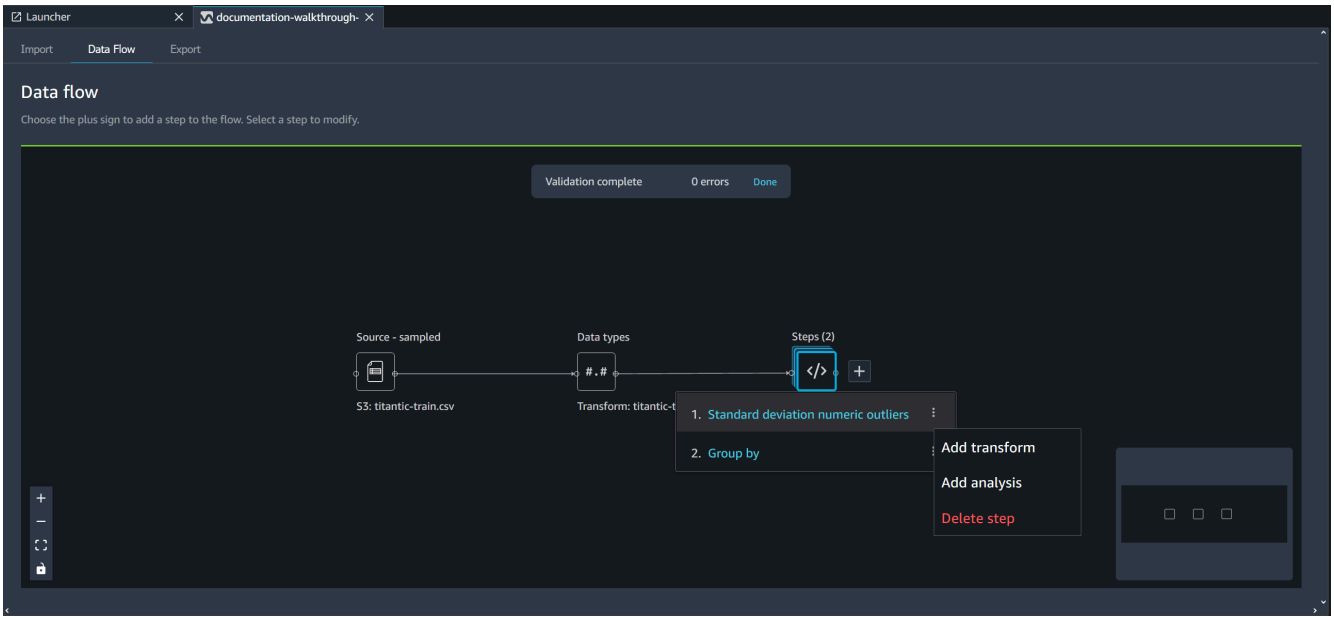
Anda dapat menggunakan salah satu prosedur berikut ini untuk menghapus langkah tanpa menghapus langkah hilir.

Delete a step in the Data Wrangler flow

Anda dapat menghapus langkah individual untuk node dalam aliran data Anda yang memiliki satu input. Anda tidak dapat menghapus langkah individual untuk sumber, bergabung, dan menggabungkan node.

Gunakan prosedur berikut untuk menghapus langkah dalam alur Data Wrangler.

1. Pilih kelompok langkah yang memiliki langkah yang Anda hapus.
2. Pilih ikon di sebelah langkah.
3. Pilih Hapus langkah.

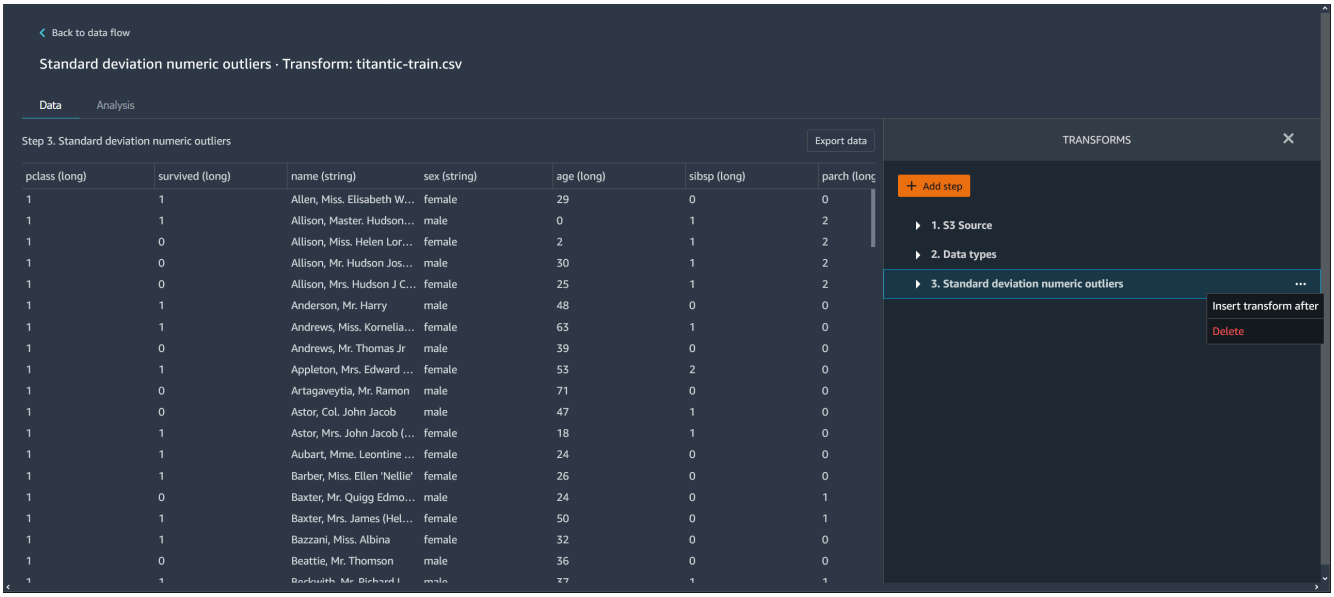


Delete a step in the table view

Gunakan prosedur berikut untuk menghapus langkah dalam tampilan tabel.

Anda dapat menghapus langkah individual untuk node dalam aliran data Anda yang memiliki satu input. Anda tidak dapat menghapus langkah individual untuk sumber, bergabung, dan menggabungkan node.

1. Pilih langkah dan buka tampilan tabel untuk langkah tersebut.
2. Gerakkan kursor Anda ke atas langkah sehingga ikon elipsis muncul.
3. Pilih ikon di sebelah langkah.
4. Pilih Hapus.



Mengedit Langkah dalam Alur Wrangler Data Anda

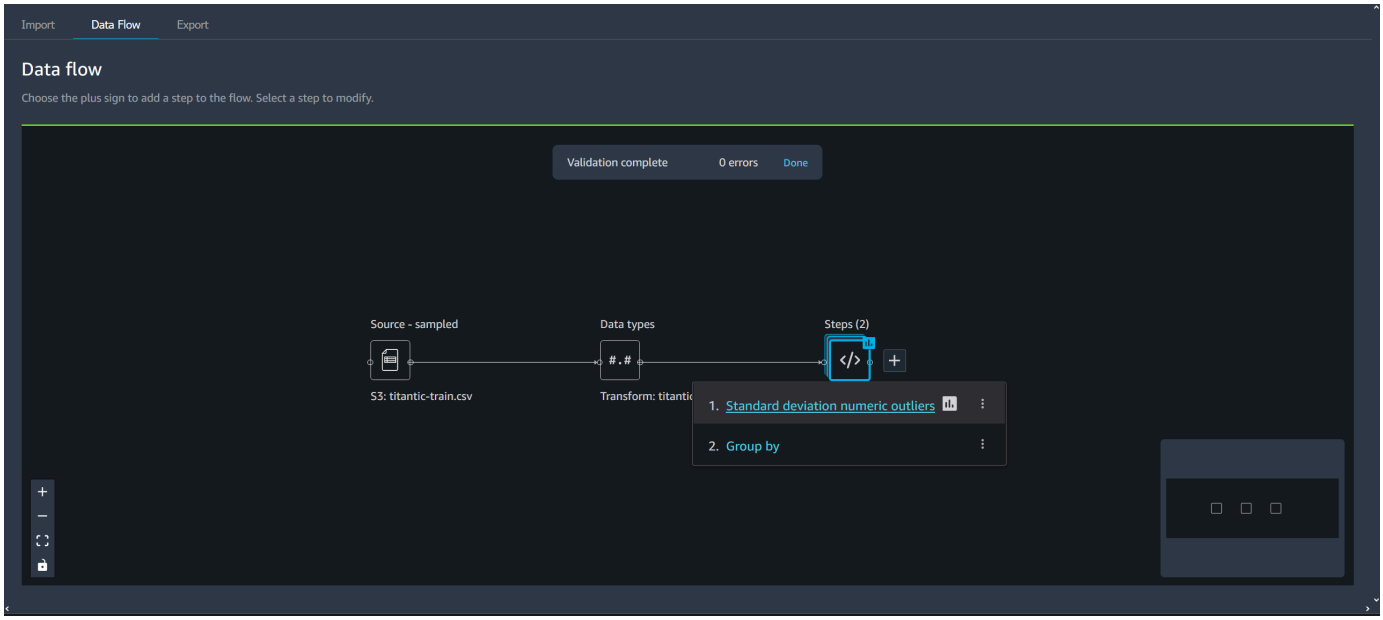
Anda dapat mengedit setiap langkah yang telah Anda tambahkan dalam alur Data Wrangler Anda. Dengan mengedit langkah-langkah, Anda dapat mengubah transformasi atau tipe data kolom. Anda dapat mengedit langkah-langkah untuk membuat perubahan yang dengannya Anda dapat melakukan analisis yang lebih baik.

Ada banyak cara Anda dapat mengedit langkah. Beberapa contoh termasuk mengubah metode imputasi atau mengubah ambang batas untuk mempertimbangkan nilai sebagai outlier.

Gunakan prosedur berikut untuk mengedit langkah.

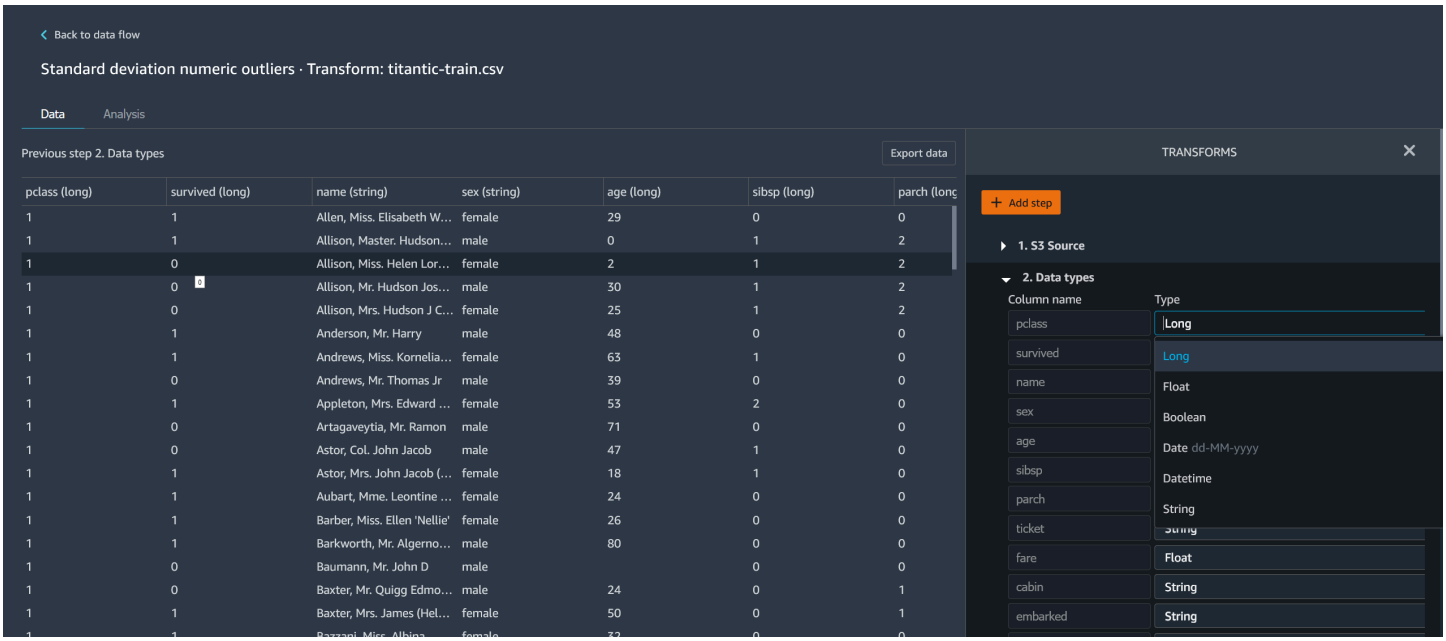
Untuk mengedit langkah, lakukan hal berikut.

1. Pilih langkah dalam alur Data Wrangler untuk membuka tampilan tabel.



2. Pilih langkah dalam aliran data.
3. Edit langkahnya.

Gambar berikut menunjukkan contoh mengedit langkah.



> **Note**
>
> Anda dapat menggunakan spasi bersama dalam SageMaker Domain Amazon untuk bekerja secara kolaboratif pada alur Data Wrangler Anda. Dalam ruang bersama, Anda

dan kolaborator dapat mengedit file aliran secara real-time. Namun, baik Anda maupun kolaborator Anda tidak dapat melihat perubahan secara real-time. Ketika ada yang membuat perubahan pada aliran Data Wrangler, mereka harus segera menyimpannya. Ketika seseorang menyimpan file, kolaborator tidak akan dapat melihatnya kecuali jika menutup file dan membukanya kembali. Setiap perubahan yang tidak disimpan oleh satu orang akan ditimpa oleh orang yang menyimpan perubahan mereka.

Dapatkan Wawasan Tentang Kualitas Data dan Data

Gunakan Laporan Kualitas Data dan Wawasan untuk melakukan analisis data yang telah Anda impor ke Data Wrangler. Kami merekomendasikan bahwa Anda membuat laporan setelah Anda mengimpor dataset Anda. Anda dapat menggunakan laporan untuk membantu Anda membersihkan dan memproses data Anda. Ini memberi Anda informasi seperti jumlah nilai yang hilang dan jumlah outlier. Jika Anda memiliki masalah dengan data Anda, seperti kebocoran target atau ketidakseimbangan, laporan wawasan dapat membawa masalah tersebut ke perhatian Anda.

Gunakan prosedur berikut untuk membuat laporan Kualitas dan Wawasan Data. Ini mengasumsikan bahwa Anda telah mengimpor dataset ke dalam aliran Data Wrangler Anda.

Untuk membuat laporan Kualitas Data dan Wawasan

1. Pilih + di sebelah node dalam alur Data Wrangler Anda.
2. Pilih Dapatkan wawasan data.
3. Untuk nama Analisis, tentukan nama untuk laporan wawasan.
4. (Opsional) Untuk kolom Target, tentukan kolom target.
5. Untuk jenis Masalah, tentukan Regresi atau Klasifikasi.
6. Untuk ukuran Data, tentukan salah satu dari berikut ini:
   - 50 K — Menggunakan 50000 baris pertama dari kumpulan data yang telah Anda impor untuk membuat laporan.
   - Seluruh kumpulan data — Menggunakan seluruh kumpulan data yang telah Anda impor untuk membuat laporan.

> **Note**
>
> Membuat laporan Kualitas Data dan Wawasan di seluruh kumpulan data menggunakan pekerjaan SageMaker pemrosesan Amazon. Pekerjaan SageMaker pemrosesan menyediakan sumber daya komputasi tambahan yang diperlukan untuk mendapatkan wawasan untuk semua data Anda. Untuk informasi selengkapnya tentang SageMaker memproses pekerjaan, lihatMemproses data.

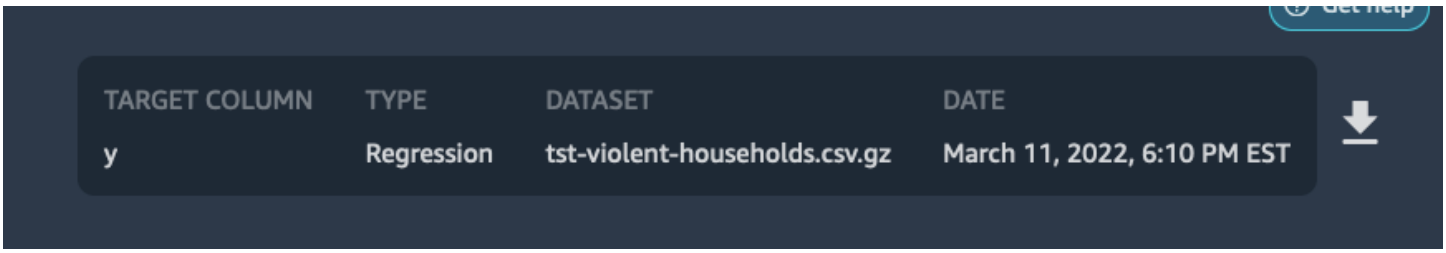
7. Pilih Buat.

Topik berikut menunjukkan bagian laporan:

Topik

- Ringkasan
- Kolom target
- Cara menghapus daftar
- Gambaran Umum
- Sampel
- Ketentuan

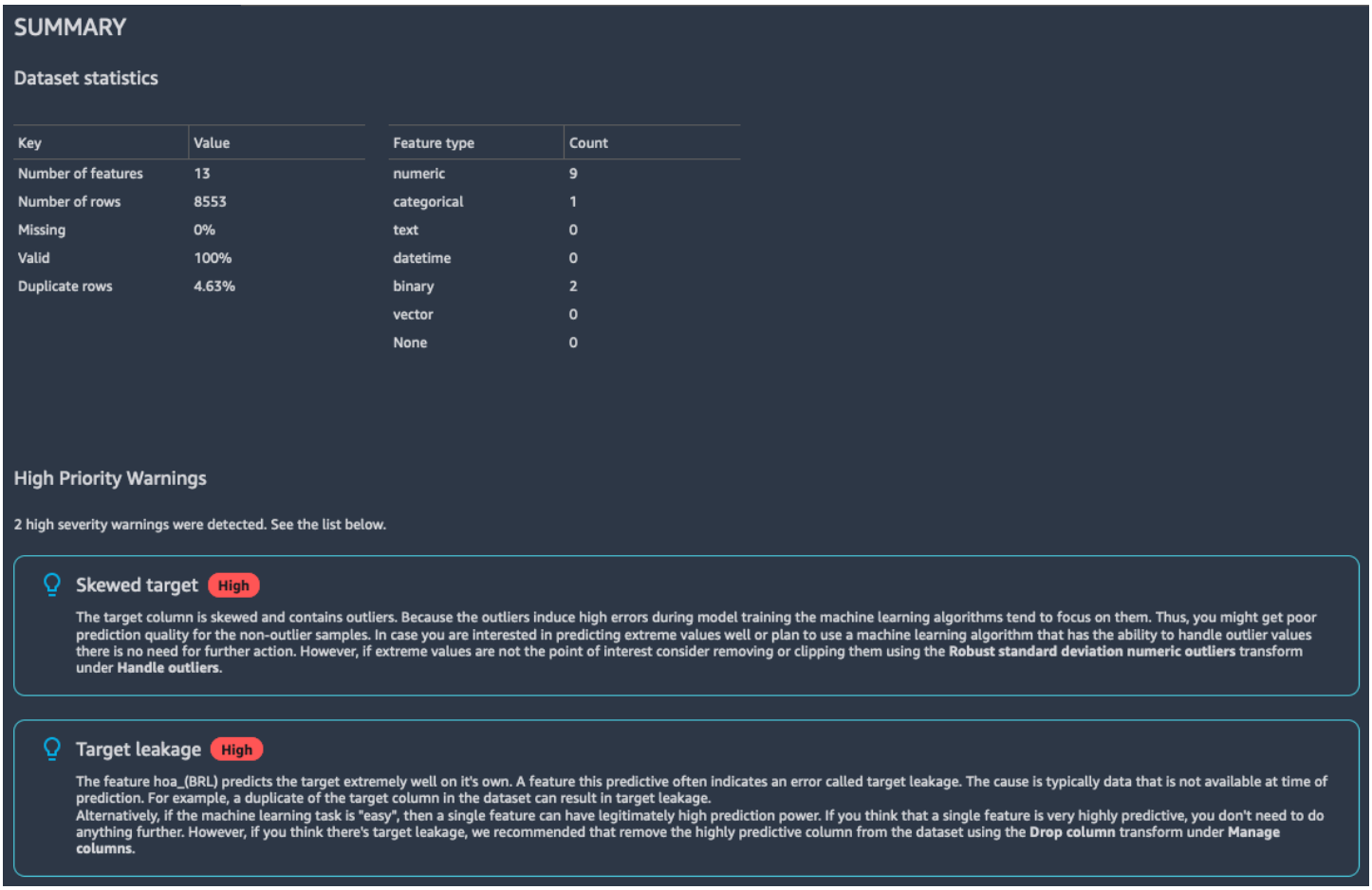
Anda dapat mengunduh laporan atau melihatnya secara online. Untuk mengunduh laporan, pilih tombol unduh di sudut kanan atas layar. Gambar berikut menunjukkan tombol.



Ringkasan

Laporan wawasan memiliki ringkasan singkat dari data yang mencakup informasi umum seperti nilai yang hilang, nilai tidak valid, jenis fitur, jumlah outlier, dan banyak lagi. Ini juga dapat mencakup peringatan tingkat keparahan tinggi yang menunjukkan kemungkinan masalah dengan data. Kami sarankan Anda menyelidiki peringatan.

Berikut ini adalah contoh ringkasan laporan.

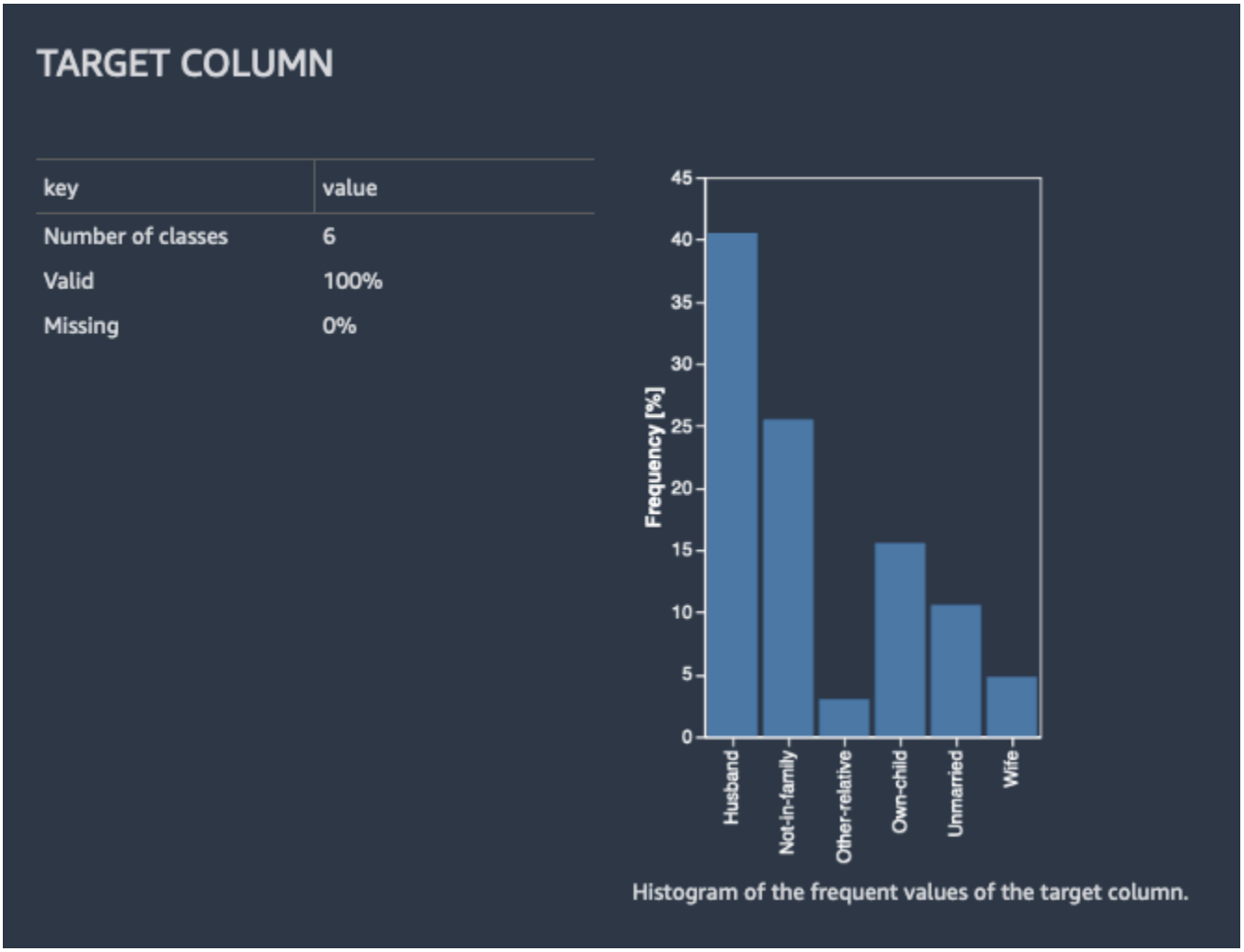


Kolom target

Saat Anda membuat laporan kualitas data dan wawasan, Data Wrangler memberi Anda opsi untuk memilih kolom target. Kolom target adalah kolom yang Anda coba prediksi. Saat Anda memilih kolom target, Data Wrangler secara otomatis membuat analisis kolom target. Ini juga memberi peringkat fitur dalam urutan kekuatan prediksi mereka. Saat memilih kolom target, Anda harus menentukan apakah Anda mencoba memecahkan masalah regresi atau klasifikasi.

Untuk klasifikasi, Data Wrangler menunjukkan tabel dan histogram kelas yang paling umum. Kelas adalah kategori. Ini juga menyajikan pengamatan, atau baris, dengan nilai target yang hilang atau tidak valid.

Gambar berikut menunjukkan contoh analisis kolom target untuk masalah klasifikasi.

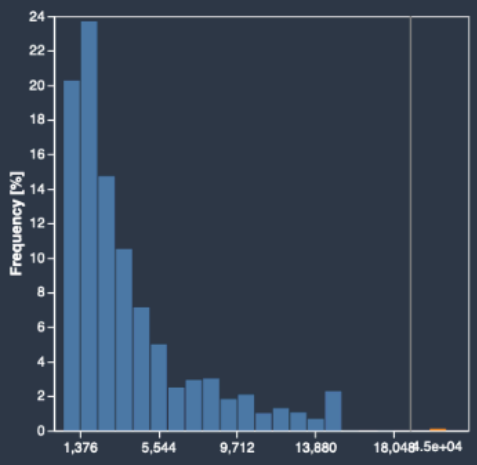


Untuk regresi, Data Wrangler menunjukkan histogram semua nilai di kolom target. Ini juga menyajikan pengamatan, atau baris, dengan nilai target yang hilang, tidak valid, atau outlier.

Gambar berikut menunjukkan contoh analisis kolom target untuk masalah regresi.

TARGET COLUMN

key	value
Valid	100%
Missing	0%
Outliers	0.103%
Min	450
Max	4.5e+04
Mean	3.9e+03
Median	2.66e+03
Skew	1.84
Kurtosis	4.62
Number of unique	1195



Histogram of the target column. The orange bars contain outliers and the value below them is the outliers average.

See below several samples with outlier target values.

city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
São Paulo	700	4	7	8	-	acept	not furnished	0	45000	8750	677	54430
São Paulo	350	3	3	3	-	acept	not furnished	0	30000	560	451	31010
São Paulo	486	8	4	6	-	acept	not furnished	0	25000	2200	376	27580
São Paulo	80	2	1	1	1	acept	not furnished	875	24000	0	305	25180
São Paulo	900	3	4	8	-	acept	not furnished	0	20000	3813	301	24110

Cara menghapus daftar

Model Cepat memberikan perkiraan kualitas prediksi yang diharapkan dari model yang Anda latih pada data Anda.

Data Wrangler membagi data Anda menjadi lipatan pelatihan dan validasi. Ini menggunakan 80% sampel untuk pelatihan dan 20% dari nilai untuk validasi. Untuk klasifikasi, sampel dibagi bertingkat. Untuk pemisahan bertingkat, setiap partisi data memiliki rasio label yang sama. Untuk masalah klasifikasi, penting untuk memiliki rasio label yang sama antara lipatan pelatihan dan klasifikasi. Data Wrangler melatih model XGBoost dengan hyperparameters default. Ini berlaku penghentian awal pada data validasi dan melakukan preprocessing fitur minimal.

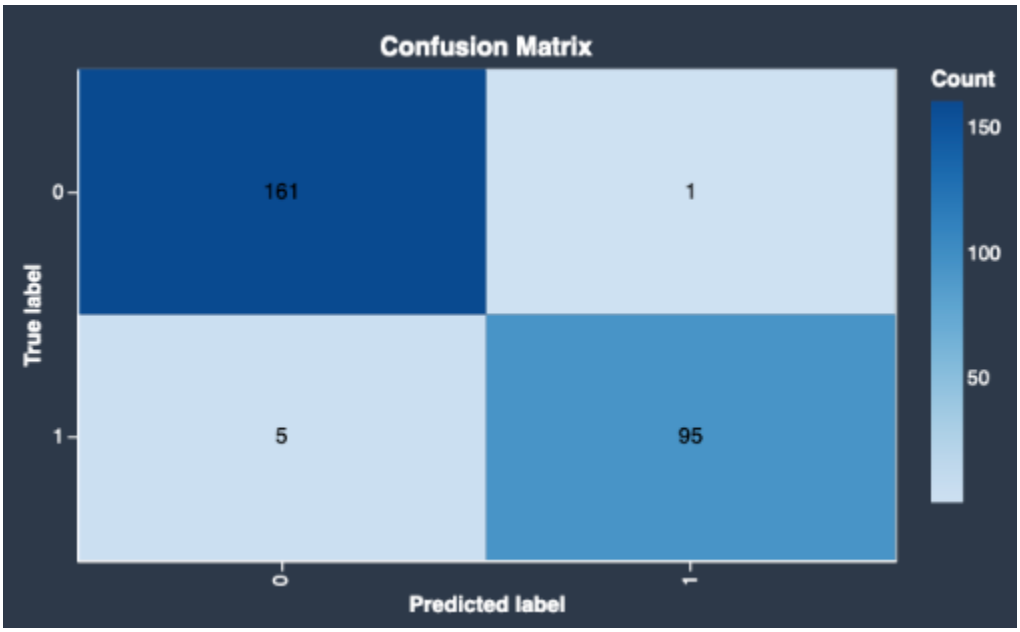
Untuk model klasifikasi, Data Wrangler mengembalikan ringkasan model dan matriks kebingungan.

Berikut ini adalah contoh ringkasan model klasifikasi. Untuk mempelajari lebih lanjut tentang informasi yang dikembalikan, lihatKetentuan.

Metric	Validation scores	Train scores
Accuracy	0.977	0.992
Balanced accuracy	0.972	0.99
ROC-AUC	0.995	1
F1	0.969	0.99
Precision	0.99	0.997
Recall	0.95	0.983

class	precision	recall	f1-score	support
0	0.9698795180722891	0.9938271604938271	0.9817073170731707	162.0
1	0.9895833333333334	0.95	0.9693877551020408	100.0

Berikut ini adalah contoh matriks kebingungan yang dikembalikan oleh model cepat.



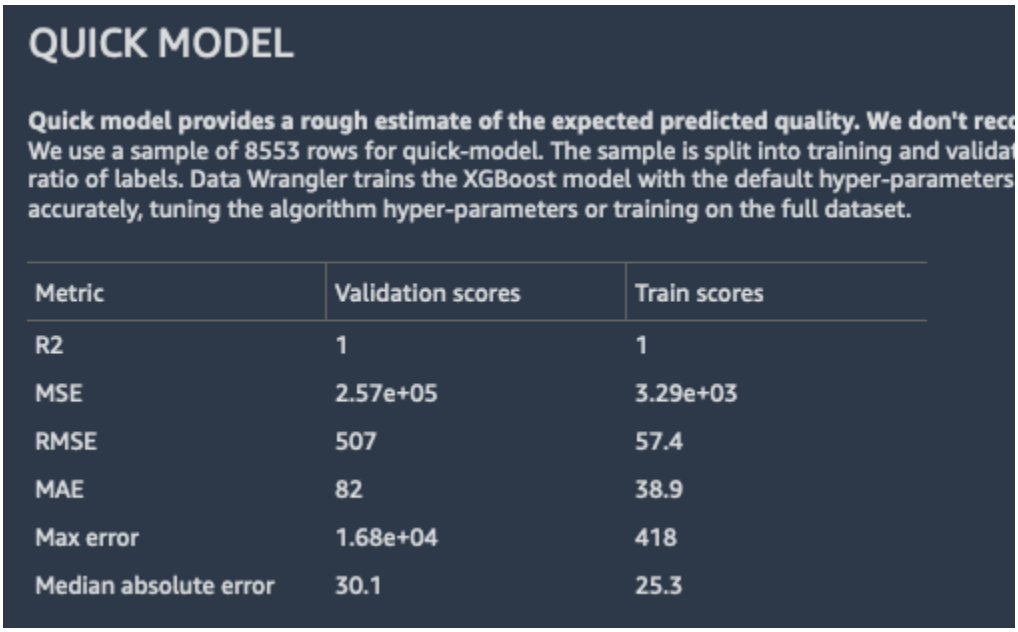
Sebuah matriks kebingungan memberikan informasi berikut untuk Anda:

- Berapa kali label yang diprediksi cocok dengan label sebenarnya.
- Berapa kali label yang diprediksi tidak cocok dengan label sebenarnya.

Label sebenarnya mewakili pengamatan aktual dalam data Anda. Misalnya, jika Anda menggunakan model untuk mendeteksi transaksi penipuan, label sebenarnya mewakili transaksi yang sebenarnya curang atau tidak curang. Label yang diprediksi mewakili label yang ditetapkan model Anda ke data.

Anda dapat menggunakan matriks kebingungan untuk melihat seberapa baik model memprediksi ada atau tidak adanya suatu kondisi. Jika Anda memprediksi transaksi penipuan, Anda dapat menggunakan matriks kebingungan untuk memahami sensitivitas dan kekhususan model. Sensitivitas mengacu pada kemampuan model untuk mendeteksi transaksi penipuan. Kekhususan mengacu pada kemampuan model untuk menghindari mendeteksi transaksi non-penipuan sebagai penipuan.

Berikut ini adalah contoh output model cepat untuk masalah regresi.



QUICK MODEL

Quick model provides a rough estimate of the expected predicted quality. We don't rec
We use a sample of 8553 rows for quick-model. The sample is split into training and validat
ratio of labels. Data Wrangler trains the XGBoost model with the default hyper-parameters
accurately, tuning the algorithm hyper-parameters or training on the full dataset.

Metric	Validation scores	Train scores
R2	1	1
MSE	2.57e+05	3.29e+03
RMSE	507	57.4
MAE	82	38.9
Max error	1.68e+04	418
Median absolute error	30.1	25.3

Gambaran Umum

Saat Anda menentukan kolom target, Data Wrangler memesan fitur berdasarkan kekuatan prediksinya. Kekuatan prediksi diukur pada data setelah dibagi menjadi 80% pelatihan dan 20% lipatan validasi. Data Wrangler cocok dengan model untuk setiap fitur secara terpisah pada lipatan pelatihan. Ini menerapkan preprocessing fitur minimal dan mengukur kinerja prediksi pada data validasi.

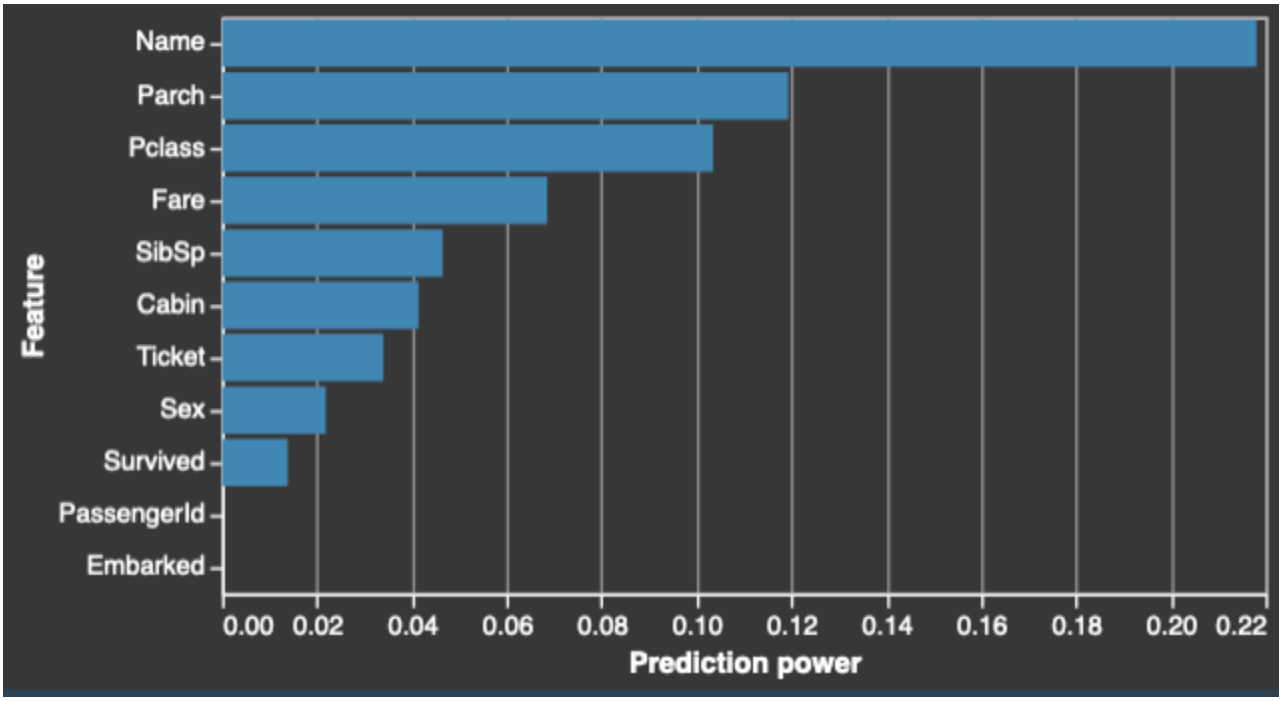
Ini menormalkan skor ke kisaran [0,1]. Skor prediksi yang lebih tinggi menunjukkan kolom yang lebih berguna untuk memprediksi target sendiri. Skor yang lebih rendah menunjuk ke kolom yang tidak memprediksi kolom target.

Ini jarang untuk kolom yang tidak prediktif sendiri untuk menjadi prediktif ketika digunakan bersama-sama dengan kolom lain. Anda dapat dengan yakin menggunakan skor prediksi untuk menentukan apakah fitur dalam kumpulan data Anda bersifat prediktif.

Skor rendah biasanya menunjukkan fitur tersebut berlebihan. Skor 1 menyiratkan kemampuan prediksi sempurna, yang sering menunjukkan kebocoran target. Kebocoran target biasanya terjadi ketika kumpulan data berisi kolom yang tidak tersedia pada waktu prediksi. Misalnya, itu bisa menjadi duplikat dari kolom target.

Berikut ini adalah contoh tabel dan histogram yang menunjukkan nilai prediksi masing-masing fitur.

Feature	Prediction power	Type	Valid	Missing	Outliers	#Warnings
Name	0.274276	text	100.0%	0.0%		0
Pclass	0.154638	numeric	100.0%	0.0%	0.0%	0
SibSp	0.141675	numeric	100.0%	0.0%	3.22%	0
Parch	0.127353	numeric	100.0%	0.0%	1.4%	0
Cabin	0.112283	text	25.91%	74.09%		0
Ticket	0.0869433	numeric	72.97%	0.0%	3.07%	0
Fare	0.0625847	numeric	100.0%	0.0%	2.52%	0
Embarked	0.00600914	categorical	99.72%	0.28%		0
Survived	0.00434197	binary	100.0%	0.0%		0
PassengerId	0	numeric	100.0%	0.0%	0.0%	0
Sex	0	binary	100.0%	0.0%		0



Sampel

Data Wrangler memberikan informasi tentang apakah sampel Anda anomali atau jika ada duplikat dalam kumpulan data Anda.

Data Wrangler mendeteksi sampel anomali menggunakan algoritma hutan isolasi. Hutan isolasi mengaitkan skor anomali dengan setiap sampel (baris) dari kumpulan data. Skor anomali yang rendah menunjukkan sampel anomali. Skor tinggi dikaitkan dengan sampel non-anomali. Sampel dengan skor anomali negatif biasanya dianggap anomali dan sampel dengan skor anomali positif dianggap non-anomali.

Ketika Anda melihat sampel yang mungkin anomali, kami sarankan Anda memperhatikan nilai-nilai yang tidak biasa. Misalnya, Anda mungkin memiliki nilai anomali yang dihasilkan dari kesalahan dalam mengumpulkan dan memproses data. Berikut ini adalah contoh sampel yang paling anomali menurut implementasi data Wrangler dari algoritma hutan isolasi. Sebaiknya gunakan pengetahuan domain dan logika bisnis saat Anda memeriksa sampel anomali.

Data Wrangler mendeteksi baris duplikat dan menghitung rasio baris duplikat dalam data Anda. Beberapa sumber data dapat menyertakan duplikat yang valid. Sumber data lain dapat memiliki duplikat yang menunjukkan masalah dalam pengumpulan data. Sampel duplikat yang dihasilkan dari pengumpulan data yang salah dapat mengganggu proses pembelajaran mesin yang mengandalkan pemisahan data menjadi pelatihan independen dan lipatan validasi.

Berikut ini adalah elemen laporan wawasan yang dapat dipengaruhi oleh sampel duplikat:

- Cara menghapus daftar
- Estimasi daya prediksi
- Penyetelan hyperparameter otomatis

Anda dapat menghapus sampel duplikat dari kumpulan data menggunakan transformasi Drop duplikat di bawah Kelola baris. Data Wrangler menunjukkan baris yang paling sering diduplikasi.

Ketentuan

Berikut ini adalah definisi istilah teknis yang digunakan dalam laporan wawasan data.

Feature types

Berikut ini adalah definisi untuk masing-masing jenis fitur:

- Numerik — Nilai numerik dapat berupa float atau bilangan bulat, seperti usia atau pendapatan. Model pembelajaran mesin mengasumsikan bahwa nilai numerik diurutkan dan jarak ditentukan di atasnya. Misalnya, 3 lebih dekat ke 4 daripada 10 dan 3 < 4 < 10.
- Categorical - Entri kolom milik satu set nilai unik, yang biasanya jauh lebih kecil dari jumlah entri di kolom. Misalnya, kolom dengan panjang 100 dapat berisi nilai unik`Dog`,`Cat`, dan`Mouse`. Nilai bisa berupa numerik, teks, atau kombinasi keduanya. `Horse`,`House`,`8`,`Love`, dan semuanya `3.1` akan menjadi nilai yang valid dan dapat ditemukan di kolom kategoris yang sama. Model pembelajaran mesin tidak mengasumsikan urutan atau jarak pada nilai-nilai fitur kategoris, sebagai lawan dari fitur numerik, bahkan ketika semua nilai adalah angka.
- Biner — Fitur biner adalah jenis fitur kategoris khusus di mana kardinalitas himpunan nilai unik adalah 2.
- Teks - Kolom teks berisi banyak nilai unik non-numerik. Dalam kasus ekstrim, semua elemen kolom itu unik. Dalam kasus ekstrim, tidak ada dua entri yang sama.
- Datetime - Kolom datetime berisi informasi tentang tanggal atau waktu. Ini dapat memiliki informasi tentang tanggal dan waktu.

Feature statistics

Berikut ini adalah definisi untuk masing-masing statistik fitur:

- Kekuatan prediksi — Kekuatan prediksi mengukur seberapa berguna kolom dalam memprediksi target.
- Outlier (dalam kolom numerik) — Data Wrangler mendeteksi outlier menggunakan dua statistik yang kuat untuk outlier: median dan solid standard deviation (RSTD). RSTD diturunkan dengan memotong nilai fitur ke kisaran [5 persentil, 95 persentil] dan menghitung standar deviasi vektor terpotong. Semua nilai yang lebih besar dari median + 5* RSTD atau lebih kecil dari median - 5 * RSTD dianggap outlier.
- Skew (dalam kolom numerik) — Skew mengukur simetri distribusi dan didefinisikan sebagai momen ketiga distribusi dibagi dengan kekuatan ketiga dari standar deviasi. Kemiringan distribusi normal atau distribusi simetris lainnya adalah nol. Nilai positif menyiratkan bahwa ekor kanan distribusi lebih panjang dari ekor kiri. Nilai negatif menyiratkan bahwa ekor kiri distribusi lebih panjang dari ekor kanan. Sebagai aturan praktis, distribusi dianggap miring ketika nilai absolut kemiringan lebih besar dari 3.
- Kurtosis (dalam kolom numerik) — Kurtosis Pearson mengukur beratnya ekor distribusi. Ini didefinisikan sebagai momen keempat dari distribusi dibagi dengan kuadrat dari momen kedua. Kurtosis dari distribusi normal adalah 3. Nilai kurtosis yang lebih rendah dari 3 menyiratkan bahwa distribusi terkonsentrasi di sekitar rata-rata dan ekor lebih ringan dari ekor distribusi normal. Nilai kurtosis lebih tinggi dari 3 menyiratkan ekor atau outlier yang lebih berat.
- Nilai yang hilang - Objek seperti nol, string kosong, dan string yang hanya terdiri dari spasi putih dianggap hilang.
- Nilai yang valid untuk fitur numerik atau target regresi - Semua nilai yang dapat Anda lemparkan ke float terbatas valid. Nilai yang hilang tidak valid.
- Nilai yang valid untuk fitur kategoris, biner, atau teks, atau untuk target klasifikasi - Semua nilai yang tidak hilang valid.
- Fitur Datetime - Semua nilai yang dapat Anda transmisikan ke objek datetime valid. Nilai yang hilang tidak valid.
- Nilai tidak valid - Nilai yang hilang atau Anda tidak dapat mentransmisikan dengan benar. Misalnya, dalam kolom numerik, Anda tidak dapat mentransmisikan string `"six"` atau nilai null.

Quick model metrics for regression

Berikut ini adalah definisi untuk metrik model cepat:

- R2 atau koefisien determinasi) — R2 adalah proporsi variasi target yang diprediksi oleh model. R2 berada dalam kisaran [-infty, 1]. 1 adalah skor model yang memprediksi target dengan sempurna dan 0 adalah skor model sepele yang selalu memprediksi rata-rata target.
- MSE atau kesalahan kuadrat rata-rata — MSE berada dalam kisaran [0, infty]. 0 adalah skor model yang memprediksi target dengan sempurna.
- MAE atau kesalahan absolut rata-rata — MAE berada dalam kisaran [0, infty] di mana 0 adalah skor model yang memprediksi target dengan sempurna.
- RMSE atau kesalahan kuadrat rata-rata akar — RMSE berada dalam kisaran [0, infty] di mana 0 adalah skor model yang memprediksi target dengan sempurna.
- Kesalahan maks - Nilai absolut maksimum kesalahan atas kumpulan data. Kesalahan maks ada dalam kisaran [0, infty]. 0 adalah skor model yang memprediksi target dengan sempurna.
- Kesalahan absolut median — Kesalahan absolut median ada dalam kisaran [0, infty]. 0 adalah skor model yang memprediksi target dengan sempurna.

Quick model metrics for classification

Berikut ini adalah definisi untuk metrik model cepat:

- Akurasi — Akurasi adalah rasio sampel yang diprediksi secara akurat. Akurasi ada dalam kisaran [0, 1]. 0 adalah skor model yang memprediksi semua sampel secara tidak benar dan 1 adalah skor model sempurna.
- Akurasi seimbang — Akurasi seimbang adalah rasio sampel yang diprediksi secara akurat ketika bobot kelas disesuaikan untuk menyeimbangkan data. Semua kelas diberi kepentingan yang sama, terlepas dari frekuensinya. Akurasi seimbang ada dalam kisaran [0, 1]. 0 adalah skor model yang memprediksi semua sampel salah. 1 adalah skor model yang sempurna.
- AUC (klasifikasi biner) — Ini adalah area di bawah kurva karakteristik operasi penerima. AUC berada dalam kisaran [0, 1] di mana model acak mengembalikan skor 0,5 dan model sempurna mengembalikan skor 1.
- AUC (OVR) — Untuk klasifikasi multiclass, ini adalah area di bawah kurva karakteristik operasi penerima yang dihitung secara terpisah untuk setiap label menggunakan satu versus istirahat. Data Wrangler melaporkan rata-rata area. AUC berada dalam kisaran [0, 1] di mana model acak mengembalikan skor 0,5 dan model sempurna mengembalikan skor 1.
- Presisi — Presisi didefinisikan untuk kelas tertentu. Presisi adalah fraksi positif sejati dari semua contoh yang diklasifikasikan model sebagai kelas itu. Presisi ada dalam kisaran [0, 1].

1 adalah skor model yang tidak memiliki positif palsu untuk kelas. Untuk klasifikasi biner, Data Wrangler melaporkan ketepatan kelas positif.

- Ingat — Recall didefinisikan untuk kelas tertentu. Recall adalah fraksi dari instance kelas yang relevan yang berhasil diambil. Ingat ada dalam kisaran [0, 1]. 1 adalah skor model yang mengklasifikasikan semua contoh kelas dengan benar. Untuk klasifikasi biner, Data Wrangler melaporkan penarikan kembali kelas positif.
- F1 — F1 didefinisikan untuk kelas tertentu. Ini adalah rata-rata harmonik dari presisi dan ingatan. F1 berada dalam kisaran [0, 1]. 1 adalah skor model yang sempurna. Untuk klasifikasi biner, Data Wrangler melaporkan F1 untuk kelas dengan nilai positif.

Textual patterns

Pola menggambarkan format tekstual string menggunakan format yang mudah dibaca. Berikut ini adalah contoh pola tekstual:

- “{digits:4-7}” menggambarkan urutan digit yang memiliki panjang antara 4 dan 7.
- “{alnum:5}” menggambarkan string alfa-numerik dengan panjang tepat 5.

Data Wrangler menyimpulkan pola dengan melihat sampel string yang tidak kosong dari data Anda. Ini dapat menggambarkan banyak pola yang umum digunakan. Keyakinan yang dinyatakan sebagai persentase menunjukkan berapa banyak data yang diperkirakan cocok dengan pola. Dengan menggunakan pola tekstual, Anda dapat melihat baris mana dalam data Anda yang perlu Anda koreksi atau jatuhkan.

Berikut ini menjelaskan pola yang dapat dikenali oleh Data Wrangler:

Pola	Format
{alnum}	String alfanumerik
1 = 50	Setiap string karakter kata
{digit}	Urutan digit
{lebih rendah}	Memerukan huruf kecil
{campuran}	Kata kasus campuran

Pola	Format
{nama}	Kata yang dimulai dengan huruf kapital
{atas}	Memerapkan huruf besar
{spasi}	karakter spasi putih

Karakter kata adalah garis bawah atau karakter yang mungkin muncul dalam kata dalam bahasa apa pun. Misalnya, string 'Hello_word' dan 'écoute' keduanya terdiri dari karakter kata. 'H' dan 'é' keduanya merupakan contoh karakter kata.

Secara Otomatis Melatih Model pada Alur Data Anda

Anda dapat menggunakan Amazon SageMaker Autopilot untuk secara otomatis melatih, menyetel, dan menerapkan model pada data yang telah diubah dalam aliran data Anda. Amazon SageMaker Autopilot dapat melalui beberapa algoritma dan menggunakan salah satu yang paling sesuai dengan data Anda. Untuk informasi selengkapnya tentang Amazon SageMaker Autopilot, lihat. SageMaker Autopilot

Saat Anda melatih dan menyetel model, Data Wrangler mengekspor data Anda ke lokasi Amazon S3 tempat SageMaker Amazon Autopilot dapat mengaksesnya.

Anda dapat menyiapkan dan menerapkan model dengan memilih node dalam alur Data Wrangler Anda dan memilih Export and Train di pratinjau data. Anda dapat menggunakan metode ini untuk melihat dataset Anda sebelum Anda memilih untuk melatih model di atasnya.

Anda juga dapat melatih dan menerapkan model langsung dari aliran data Anda.

Prosedur berikut mempersiapkan dan menyebarkan model dari aliran data. Untuk alur Data Wrangler dengan transformasi multi-baris, Anda tidak dapat menggunakan transformasi dari aliran Data Wrangler saat Anda menerapkan model. Anda dapat menggunakan prosedur berikut untuk memproses data sebelum menggunakannya untuk melakukan inferensi.

Untuk melatih dan menerapkan model langsung dari aliran data Anda, lakukan hal berikut.

1. Pilih + di sebelah node yang berisi data pelatihan.
2. Pilih model Kereta.

3. (Opsional) Tentukan AWS KMS kunci atau ID. Untuk informasi selengkapnya tentang membuat dan mengendalikan kunci kriptografi untuk melindungi data Anda, lihat AWS Key Management Service.
4. Pilih Ekspor dan kereta api.
5. Setelah Amazon SageMaker Autopilot melatih model pada data yang diekspor Data Wrangler, tentukan nama untuk nama Eksperimen.
6. Di bawah Input data, pilih Pratinjau untuk memverifikasi bahwa Data Wrangler mengekspor data Anda dengan benar ke Amazon Autopilot. SageMaker
7. Untuk Target, pilih kolom target.
8. (Opsional) Untuk lokasi S3 di bawah Data keluaran, tentukan lokasi Amazon S3 selain lokasi default.
9. Pilih Berikutnya: Metode pelatihan.
10. Pilih metode pelatihan. Untuk informasi selengkapnya, lihat Mode pelatihan.
11. (Opsional) Untuk titik akhir penerapan Otomatis, tentukan nama untuk titik akhir.
12. Untuk opsi Deployment, pilih metode penerapan. Anda dapat memilih untuk menerapkan dengan atau tanpa transformasi yang telah Anda buat pada data Anda.

    > **Important**
    >
    > Anda tidak dapat menerapkan model SageMaker Autopilot Amazon dengan transformasi yang telah Anda buat dalam alur Data Wrangler Anda. Untuk informasi lebih lanjut tentang perubahan tersebut, lihatEkspor ke Endpoint Inferensi.

13. Pilih Selanjutnya: Tinjau dan buat.
14. Pilih Buat percobaan.

Untuk informasi selengkapnya tentang pelatihan dan penerapan model, lihatMembuat tugas regresi atau klasifikasi untuk data tabular menggunakan AutoML API. Autopilot menunjukkan kepada Anda analisis tentang kinerja model terbaik. Untuk informasi lebih lanjut tentang kinerja model, lihatLihat Laporan Kinerja Model Autopilot.

Memindahkan

Amazon SageMaker Data Wrangler menyediakan banyak transformasi data ML untuk merampingkan pembersihan, transformasi, dan fitur data Anda. Ketika Anda menambahkan transformasi, itu

menambahkan langkah ke aliran data. Setiap transformasi yang Anda tambahkan memodifikasi dataset Anda dan menghasilkan kerangka data baru. Semua transformasi selanjutnya berlaku untuk kerangka data yang dihasilkan.

Data Wrangler mencakup transformasi bawaan, yang dapat Anda gunakan untuk mengubah kolom tanpa kode apa pun. Anda juga dapat menambahkan transformasi kustom menggunakan PySpark, Python (User-Defined Function), panda, dan SQL. PySpark Beberapa transformasi beroperasi di tempat, sementara yang lain membuat kolom output baru di dataset Anda.

Anda dapat menerapkan transformasi ke beberapa kolom sekaligus. Misalnya, Anda dapat menghapus beberapa kolom dalam satu langkah.

Anda dapat menerapkan numerik Proses dan Menangani transformasi yang hilang hanya ke satu kolom.

Gunakan halaman ini untuk mempelajari lebih lanjut tentang transformasi bawaan dan kustom ini.

Memindahkan

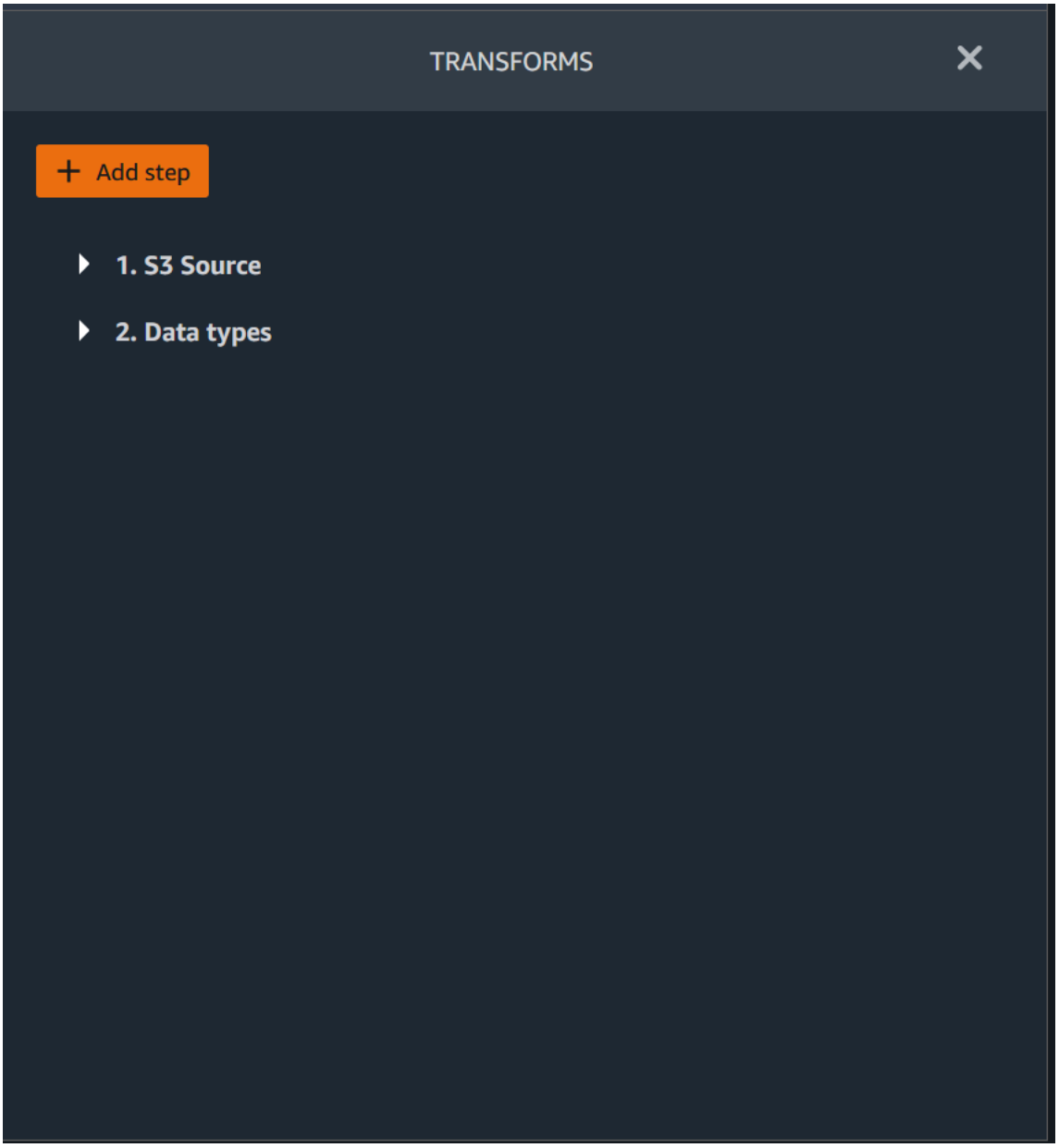
Sebagian besar transformasi bawaan terletak di tab Siapkan UI Data Wrangler. Anda dapat mengakses transformasi join dan concatenate melalui tampilan aliran data. Gunakan tabel berikut untuk melihat pratinjau dua tampilan ini.

Transform

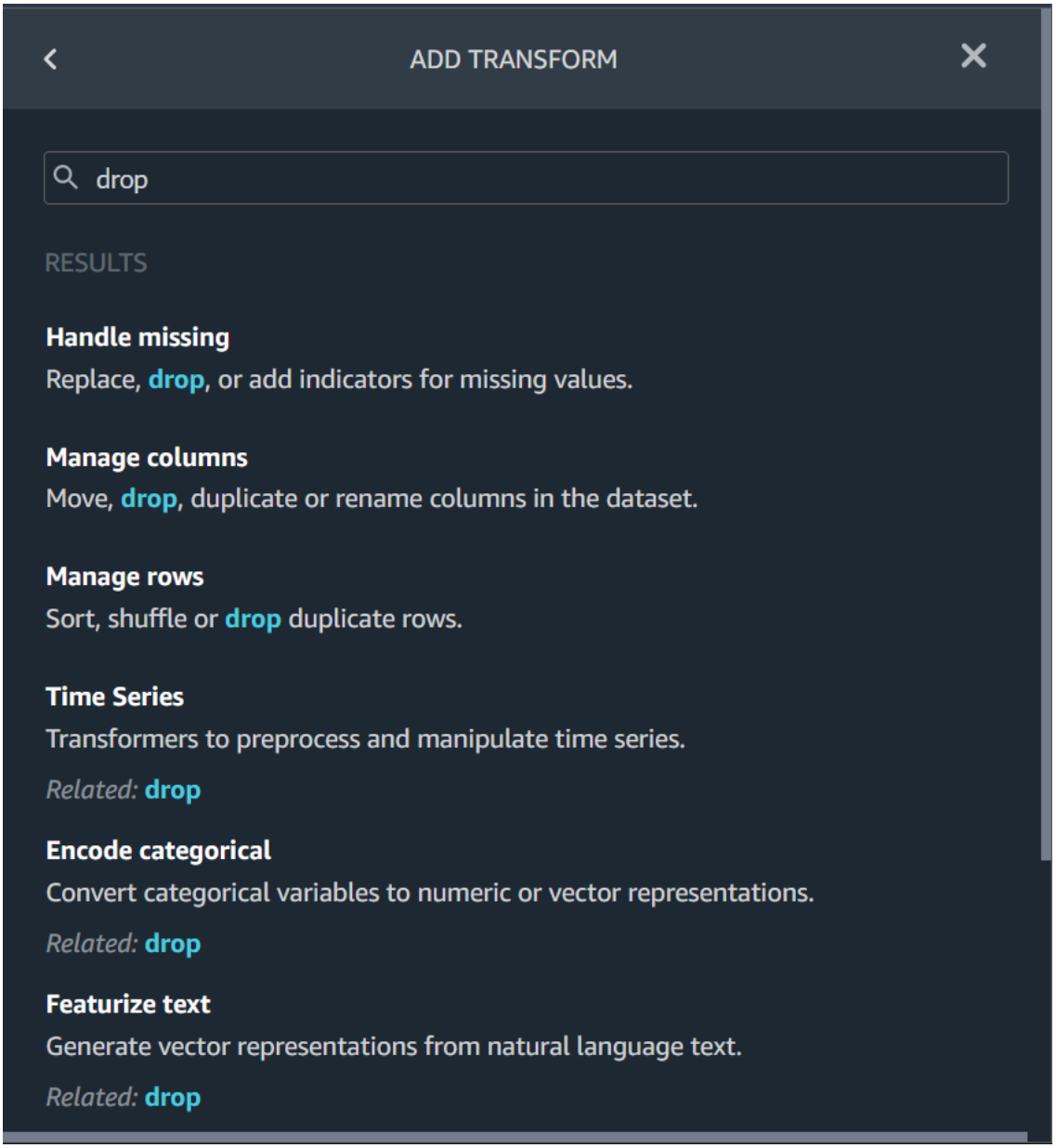
Anda dapat menambahkan transformasi ke langkah apa pun dalam aliran data Anda. Gunakan prosedur berikut untuk menambahkan perubahan ke alur data Anda.

Untuk menambahkan langkah ke aliran data Anda, lakukan hal berikut.

1. Pilih + di sebelah langkah dalam aliran data.
2. Pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.

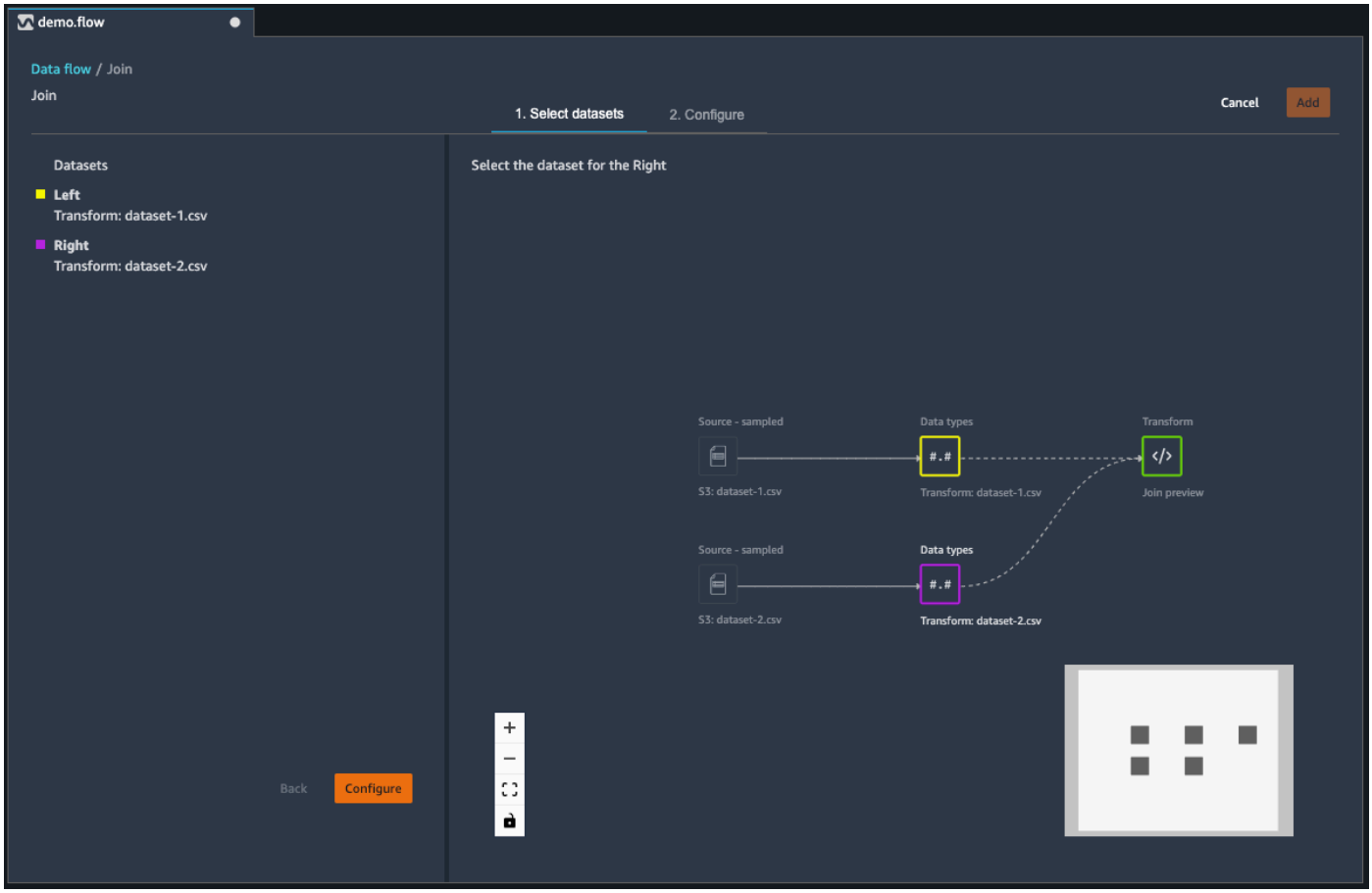


4. Pilih transformasi.
5. (Opsional) Anda dapat mencari perubahan yang ingin Anda gunakan. Data Wrangler menyoroti kueri dalam hasil.

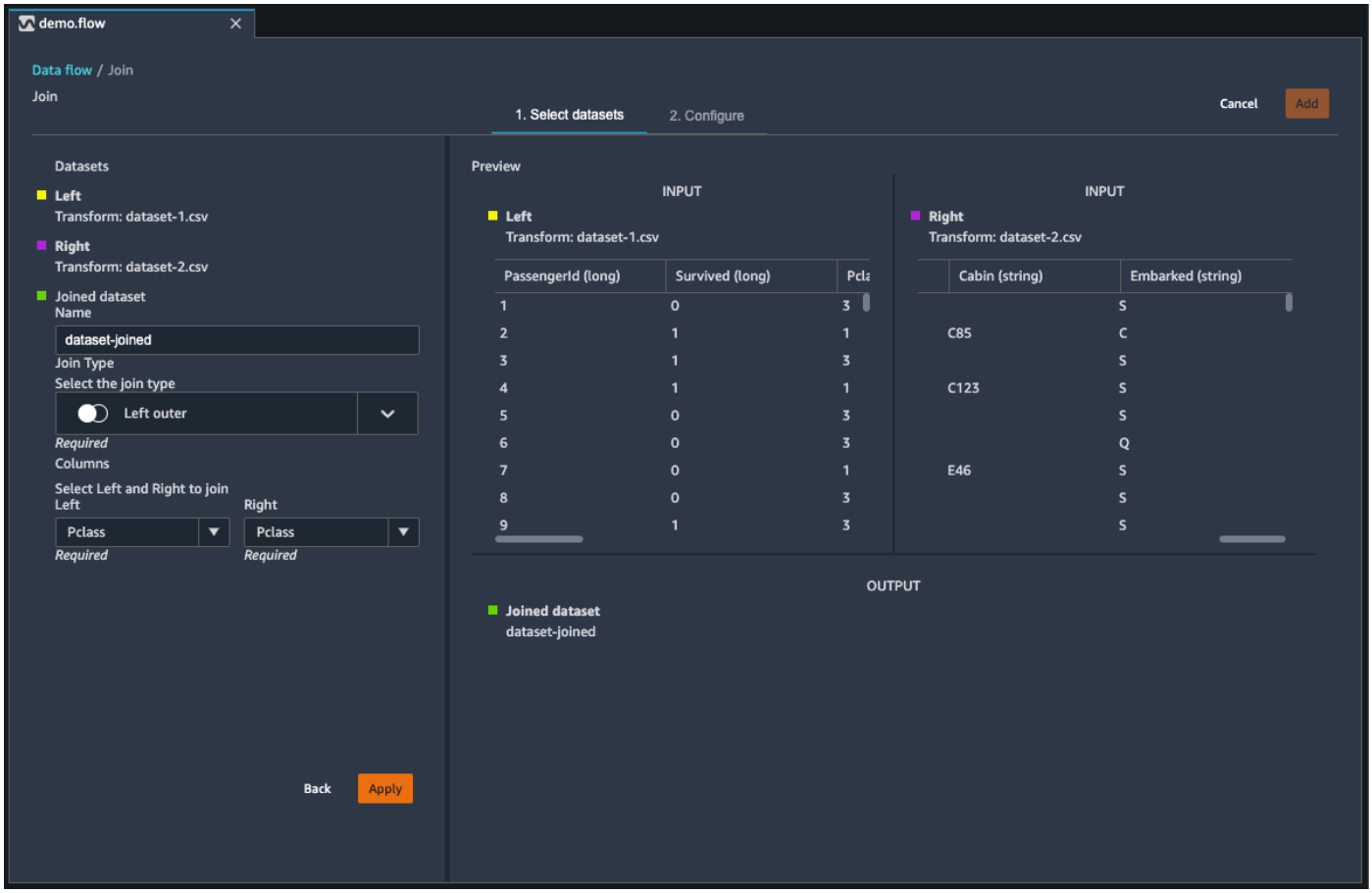


Join View

Untuk menggabungkan dua kumpulan data, pilih kumpulan data pertama dalam aliran data Anda dan pilih Gabung. Saat Anda memilih Gabung, Anda melihat hasil yang mirip dengan yang ditampilkan di gambar berikut. Kumpulan data kiri dan kanan Anda ditampilkan di panel kiri. Panel utama menampilkan aliran data Anda, dengan dataset yang baru bergabung ditambahkan.



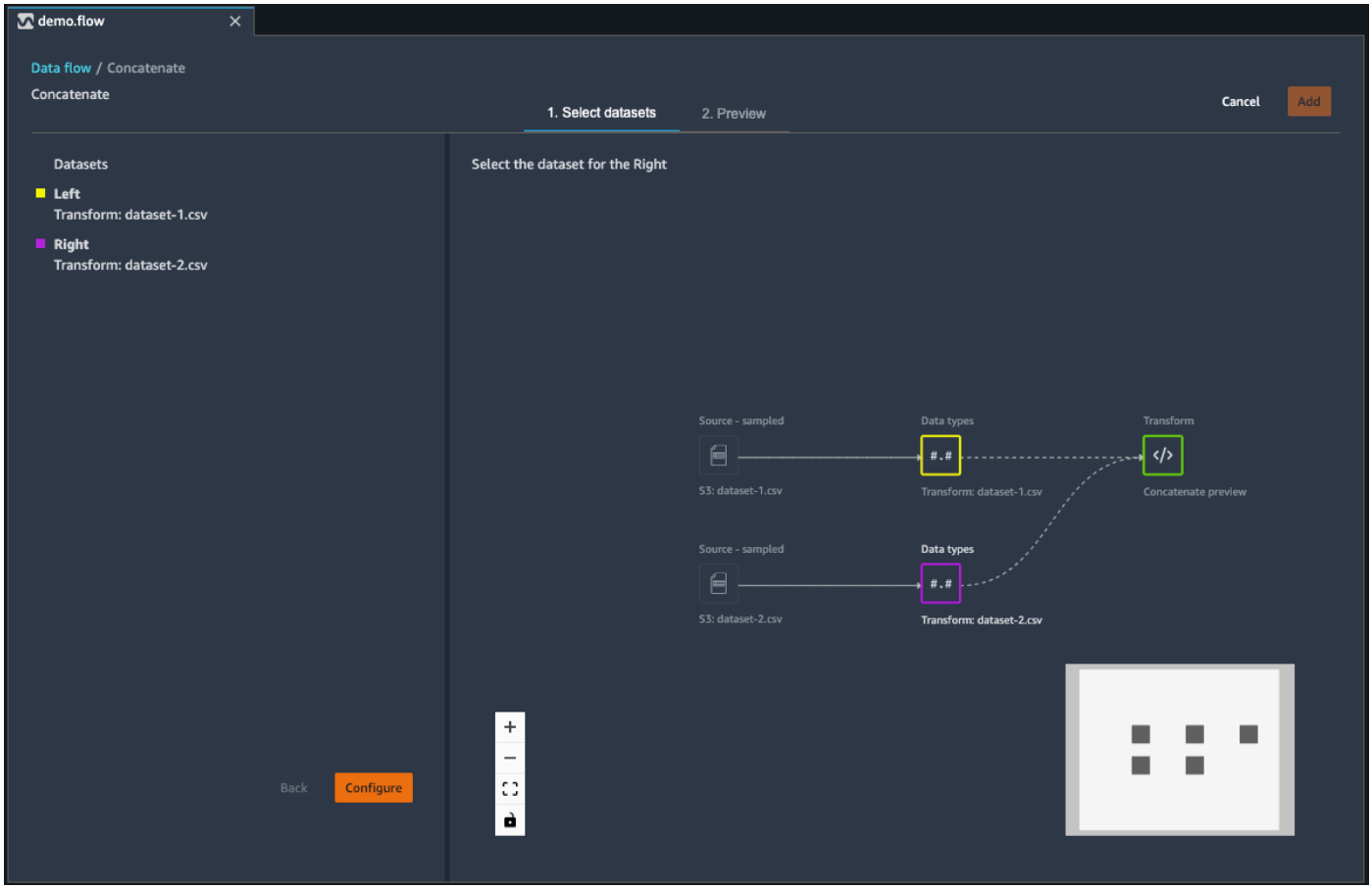
Ketika Anda memilih Konfigurasi untuk mengonfigurasi gabungan Anda, Anda melihat hasil yang mirip dengan yang ditunjukkan pada gambar berikut. Konfigurasi bergabung Anda ditampilkan di panel kiri. Anda dapat menggunakan panel ini untuk memilih nama kumpulan data yang bergabung, jenis gabungan, dan kolom untuk bergabung. Panel utama menampilkan tiga tabel. Dua tabel teratas menampilkan kumpulan data kiri dan kanan masing-masing di kiri dan kanan. Di bawah tabel ini, Anda dapat melihat pratinjau kumpulan data yang digabungkan.



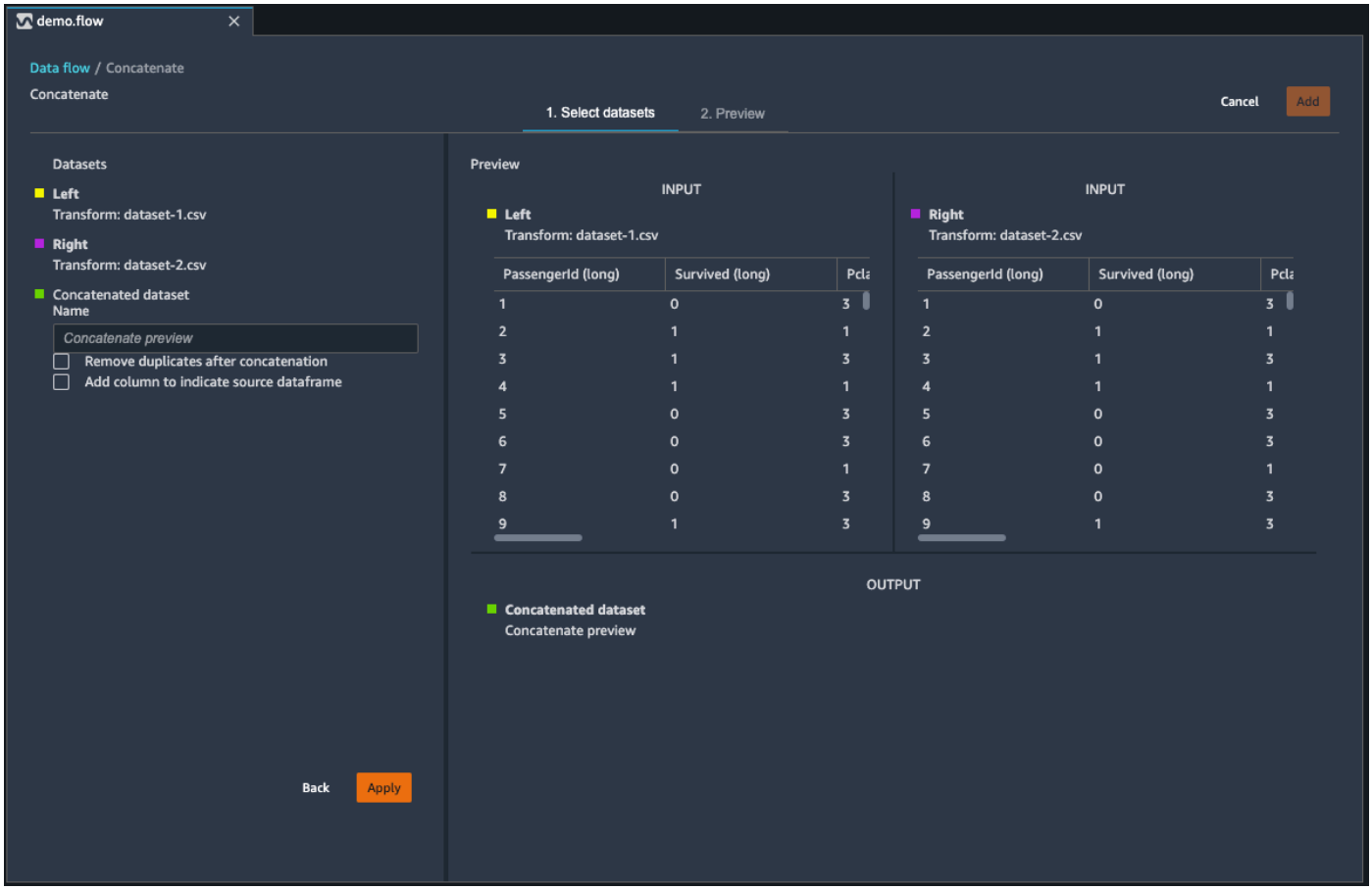
Lihat Set data untuk mempelajari selengkapnya.

Concatenate View

Untuk menggabungkan dua kumpulan data, Anda memilih kumpulan data pertama dalam aliran data Anda dan memilih Concatenate. Saat Anda memilih Concatenate, Anda melihat hasil yang mirip dengan yang ditampilkan di gambar berikut. Kumpulan data kiri dan kanan Anda ditampilkan di panel kiri. Panel utama menampilkan aliran data Anda, dengan dataset yang baru digabungkan ditambahkan.



Ketika Anda memilih Konfigurasi untuk mengonfigurasi rangkaian Anda, Anda melihat hasil yang mirip dengan yang ditunjukkan pada gambar berikut. Konfigurasi gabungan Anda ditampilkan di panel kiri. Anda dapat menggunakan panel ini untuk memilih nama kumpulan data gabungan, dan memilih untuk menghapus duplikat setelah penggabungan dan menambahkan kolom untuk menunjukkan kerangka data sumber. Panel utama menampilkan tiga tabel. Dua tabel teratas menampilkan kumpulan data kiri dan kanan masing-masing di kiri dan kanan. Di bawah tabel ini, Anda dapat melihat pratinjau kumpulan data gabungan.



Lihat Set Data Penggabungan untuk mempelajari selengkapnya.

Set data

Anda bergabung dengan kerangka data secara langsung dalam aliran data Anda. Saat Anda menggabungkan dua kumpulan data, kumpulan data gabungan yang dihasilkan akan muncul di alur Anda. Jenis gabungan berikut didukung oleh Data Wrangler.

- Left Outer - Sertakan semua baris dari tabel kiri. Jika nilai untuk kolom bergabung pada baris tabel kiri tidak cocok dengan nilai baris tabel kanan, baris tersebut berisi nilai nol untuk semua kolom tabel kanan dalam tabel gabungan.
- Anti Kiri - Sertakan baris dari tabel kiri yang tidak mengandung nilai di tabel kanan untuk kolom yang digabungkan.
- Semi kiri - Sertakan satu baris dari tabel kiri untuk semua baris identik yang memenuhi kriteria dalam pernyataan gabungan. Ini tidak termasuk baris duplikat dari tabel kiri yang cocok dengan kriteria gabungan.

- Luar Kanan - Sertakan semua baris dari tabel kanan. Jika nilai untuk kolom bergabung di baris tabel kanan tidak cocok dengan nilai baris tabel kiri, baris tersebut berisi nilai nol untuk semua kolom tabel kiri dalam tabel gabungan.
- Inner - Sertakan baris dari tabel kiri dan kanan yang berisi nilai yang cocok di kolom yang digabungkan.
- Full Outer - Sertakan semua baris dari tabel kiri dan kanan. Jika nilai baris untuk kolom gabungan di salah satu tabel tidak cocok, baris terpisah dibuat dalam tabel gabungan. Jika baris tidak berisi nilai untuk kolom dalam tabel gabungan, null disisipkan untuk kolom itu.
- Cartesian Cross - Sertakan baris yang menggabungkan setiap baris dari tabel pertama dengan setiap baris dari tabel kedua. Ini adalah produk Cartesian dari baris dari tabel di join. Hasil dari produk ini adalah ukuran tabel kiri dikalikan ukuran meja kanan. Oleh karena itu, kami menyarankan agar berhati-hati dalam menggunakan gabungan ini di antara kumpulan data yang sangat besar.

Gunakan prosedur berikut untuk menggabungkan dua kerangka data.

1. Pilih + di sebelah kerangka data kiri yang ingin Anda ikuti. Rangka data pertama yang Anda pilih selalu tabel kiri di gabungan Anda.
2. Pilih Bergabung.
3. Pilih kerangka data yang tepat. Rangka data kedua yang Anda pilih selalu merupakan tabel yang tepat dalam bergabung Anda.
4. Pilih Konfigurasi untuk mengonfigurasi gabungan Anda.
5. Beri nama kumpulan data gabungan Anda menggunakan bidang Nama.
6. Pilih jenis Gabung.
7. Pilih kolom dari tabel kiri dan kanan untuk bergabung.
8. Pilih Terapkan untuk melihat pratinjau kumpulan data yang bergabung di sebelah kanan.
9. Untuk menambahkan tabel gabungan ke alur data Anda, pilih Tambah.

Set Data Penggabungan

Gabungkan dua kumpulan data:

1. Pilih + di sebelah kerangka data kiri yang ingin Anda gabungkan. Rangka data pertama yang Anda pilih selalu tabel kiri dalam rangkaian Anda.

2. Pilih Concatenate.
3. Pilih kerangka data yang tepat. Rangka data kedua yang Anda pilih selalu merupakan tabel yang tepat dalam rangkaian Anda.
4. Pilih Konfigurasi untuk mengonfigurasi rangkaian Anda.
5. Beri nama kumpulan data gabungan Anda menggunakan bidang Nama.
6. (Opsional) Pilih kotak centang di samping Hapus duplikat setelah penggabungan untuk menghapus kolom duplikat.
7. (Opsional) Pilih kotak centang di sebelah Tambahkan kolom untuk menunjukkan kerangka data sumber jika, untuk setiap kolom dalam kumpulan data baru, Anda ingin menambahkan indikator sumber kolom.
8. Pilih Terapkan untuk melihat pratinjau kumpulan data baru.
9. Pilih Tambah untuk menambahkan kumpulan data baru ke alur data Anda.

Data Saldo

Anda dapat menyeimbangkan data untuk kumpulan data dengan kategori yang kurang terwakili. Menyeimbangkan kumpulan data dapat membantu Anda membuat model yang lebih baik untuk klasifikasi biner.

> **Note**
>
> Anda tidak dapat menyeimbangkan kumpulan data yang berisi vektor kolom.

Anda dapat menggunakan operasi data Saldo untuk menyeimbangkan data Anda menggunakan salah satu operator berikut:

- Oversampling acak - Duplikat sampel secara acak dalam kategori minoritas. Misalnya, jika Anda mencoba mendeteksi penipuan, Anda mungkin hanya memiliki kasus penipuan di 10% data Anda. Untuk proporsi yang sama dari kasus penipuan dan non-penipuan, operator ini secara acak menduplikasi kasus penipuan dalam kumpulan data 8 kali.
- Undersampling acak — Kira-kira setara dengan oversampling acak. Secara acak menghapus sampel dari kategori yang terwakili secara berlebihan untuk mendapatkan proporsi sampel yang Anda inginkan.

- Synthetic Minority Oversampling Technique (SMOTE) — Menggunakan sampel dari kategori yang kurang terwakili untuk menginterpolasi sampel minoritas sintetis baru. Untuk informasi lebih lanjut tentang SMOTE, lihat deskripsi berikut.

Anda dapat menggunakan semua transformasi untuk kumpulan data yang berisi fitur numerik dan non-numerik. SMOTE menginterpolasi nilai dengan menggunakan sampel tetangga. Data Wrangler menggunakan jarak R-kuadrat untuk menentukan lingkungan untuk menginterpolasi sampel tambahan. Data Wrangler hanya menggunakan fitur numerik untuk menghitung jarak antara sampel dalam kelompok yang kurang terwakili.

Untuk dua sampel nyata dalam kelompok yang kurang terwakili, Data Wrangler menginterpolasi fitur numerik dengan menggunakan rata-rata tertimbang. Ini secara acak memberikan bobot untuk sampel tersebut dalam kisaran [0, 1]. Untuk fitur numerik, Data Wrangler menginterpolasi sampel menggunakan rata-rata tertimbang sampel. Untuk sampel A dan B, Data Wrangler dapat secara acak menetapkan berat 0,7 hingga A dan 0,3 hingga B. Sampel yang diinterpolasi memiliki nilai 0,7A +0,3B.

Data Wrangler menginterpolasi fitur non-numerik dengan menyalin dari salah satu sampel nyata yang diinterpolasi. Ini menyalin sampel dengan probabilitas bahwa itu secara acak menetapkan untuk setiap sampel. Untuk sampel A dan B, ia dapat menetapkan probabilitas 0,8 ke A dan 0,2 ke B. Untuk probabilitas yang ditetapkan, ia menyalin A 80% dari waktu.

Transformasi

Grup Custom Transforms memungkinkan Anda untuk menggunakan Python (User-Defined Function) PySpark,, pandas, PySpark atau (SQL) untuk menentukan transformasi kustom. Untuk ketiga opsi, Anda menggunakan variabel `df` untuk mengakses kerangka data yang ingin Anda terapkan transformasi. Untuk menerapkan kode kustom Anda ke kerangka data Anda, tetapkan kerangka data dengan transformasi yang telah Anda buat ke variabel. `df` Jika Anda tidak menggunakan Python (User-Defined Function), Anda tidak perlu menyertakan pernyataan pengembalian. Pilih Pratinjau untuk melihat pratinjau hasil transformasi kustom. Pilih Tambah untuk menambahkan transformasi kustom ke daftar langkah Sebelumnya.

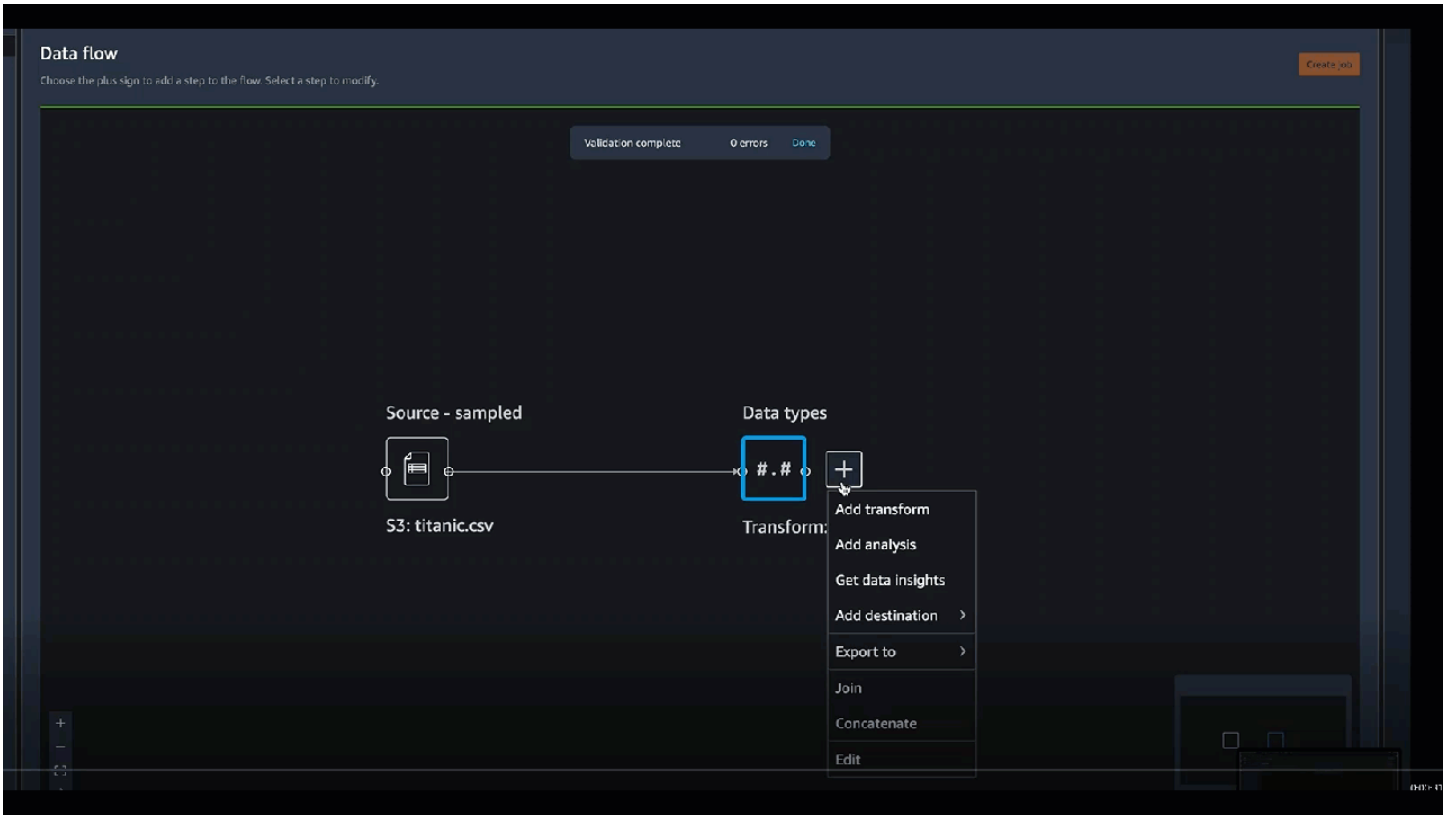
Anda dapat mengimpor pustaka populer dengan `import` pernyataan di blok kode transformasi kustom, seperti berikut ini:

- NumPy versi 1.19.0
- scikit-learn versi 0.23.2

- SciPy versi 1.5.4
- panda versi 1.0.3
- PySpark versi 3.0.0

> **Important**
>
> Transformasi kustom tidak mendukung kolom dengan spasi atau karakter khusus dalam nama. Kami menyarankan Anda menentukan nama kolom yang hanya memiliki karakter alfanumerik dan garis bawah. Anda dapat menggunakan Transformasi kolom Rename di grup Mengelola kolom transformasi untuk menghapus spasi dari nama kolom. Anda juga dapat menambahkan Python (Pandas) Custom transform mirip dengan berikut ini untuk menghapus spasi dari beberapa kolom dalam satu langkah. Contoh ini mengubah kolom bernama `A column` dan `B column` ke `A_column` dan `B_column` masing-masing.
>
> ```
> df.rename(columns={"A column": "A_column", "B column": "B_column"})
> ```

Jika Anda menyertakan pernyataan cetak di blok kode, hasilnya akan muncul saat Anda memilih Pratinjau. Anda dapat mengubah ukuran panel transformator kode khusus. Mengubah ukuran panel menyediakan lebih banyak ruang untuk menulis kode. Gambar berikut menunjukkan mengubah ukuran panel.



Bagian berikut memberikan konteks tambahan dan contoh untuk menulis kode transformasi kustom.

Python (Fungsi yang Ditetapkan Pengguna)

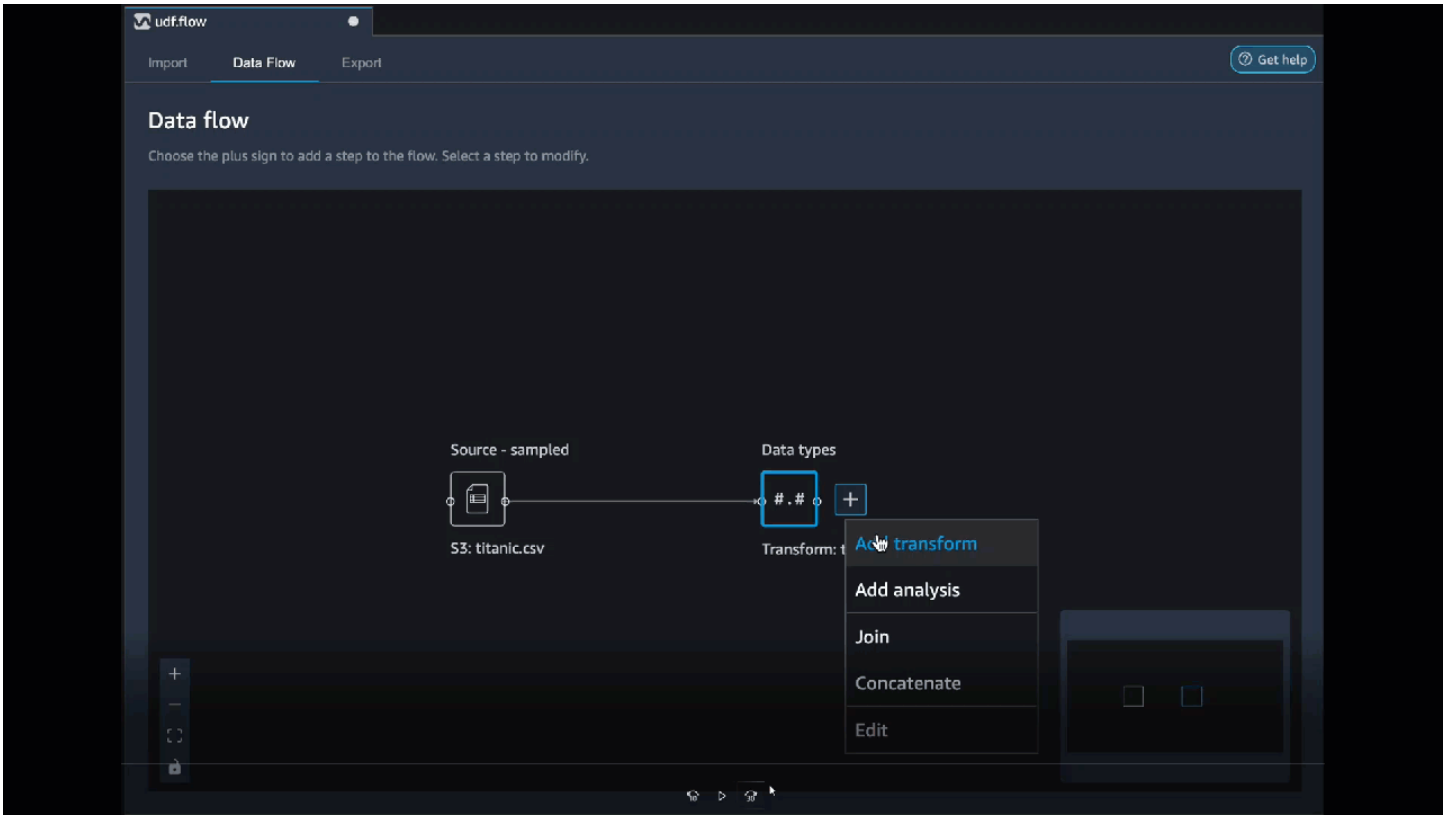
Fungsi Python memberi Anda kemampuan untuk menulis transformasi khusus tanpa perlu mengetahui Apache Spark atau panda. Data Wrangler dioptimalkan untuk menjalankan kode kustom Anda dengan cepat. Anda mendapatkan kinerja serupa menggunakan kode Python khusus dan plugin Apache Spark.

Untuk menggunakan blok kode Python (User-Defined Function), Anda tentukan yang berikut ini:

- Kolom input - Kolom masukan tempat Anda menerapkan transformasi.
- Mode — Mode scripting, baik panda atau Python.
- Jenis pengembalian - Tipe data dari nilai yang Anda kembalikan.

Menggunakan mode panda memberikan kinerja yang lebih baik. Mode Python memudahkan Anda untuk menulis transformasi dengan menggunakan fungsi Python murni.

Video berikut menunjukkan contoh cara menggunakan kode kustom untuk membuat transformasi. Ini menggunakan dataset Titanic untuk membuat kolom dengan salam orang tersebut.



PySpark

Contoh berikut mengekstrak tanggal dan waktu dari stempel waktu.

```
from pyspark.sql.functions import from_unixtime, to_date, date_format
df = df.withColumn('DATE_TIME', from_unixtime('TIMESTAMP'))
df = df.withColumn( 'EVENT_DATE', to_date('DATE_TIME')).withColumn(
'EVENT_TIME', date_format('DATE_TIME', 'HH:mm:ss'))
```

panda

Contoh berikut memberikan ikhtisar kerangka data yang Anda tambahkan transformasi.

```
df.info()
```

PySpark (SQL)

Contoh berikut membuat kerangka data baru dengan empat kolom: name, fare, pclass, survived.

```
SELECT name, fare, pclass, survived FROM df
```

Jika Anda tidak tahu cara menggunakannya PySpark, Anda dapat menggunakan cuplikan kode khusus untuk membantu Anda memulai.

Data Wrangler memiliki kumpulan cuplikan kode yang dapat dicari. Anda dapat menggunakan potongan kode untuk melakukan tugas seperti menjatuhkan kolom, mengelompokkan berdasarkan kolom, atau pemodelan.

Untuk menggunakan cuplikan kode, pilih Cari contoh cuplikan dan tentukan kueri di bilah pencarian. Teks yang Anda tentukan dalam kueri tidak harus sama persis dengan nama cuplikan kode.

Contoh berikut menunjukkan cuplikan kode baris duplikat Jatuhkan yang dapat menghapus baris dengan data serupa di kumpulan data Anda. Anda dapat menemukan cuplikan kode dengan mencari salah satu dari berikut ini:

- Duplikat
- Identik
- Menghapus

Cuplikan berikut memiliki komentar untuk membantu Anda memahami perubahan yang perlu Anda buat. Untuk sebagian besar cuplikan, Anda harus menentukan nama kolom kumpulan data Anda dalam kode.

```
# Specify the subset of columns
# all rows having identical values in these columns will be dropped

subset = ["col1", "col2", "col3"]
df = df.dropDuplicates(subset)

# to drop the full-duplicate rows run
# df = df.dropDuplicates()
```

Untuk menggunakan cuplikan, salin dan tempel kontennya ke bidang Custom transform. Anda dapat menyalin dan menempelkan beberapa cuplikan kode ke bidang transformasi khusus.

Formula

Gunakan rumus Kustom untuk menentukan kolom baru menggunakan ekspresi Spark SQL untuk menanyakan data dalam kerangka data saat ini. Kueri harus menggunakan konvensi ekspresi Spark SQL.

> **Important**
>
> Rumus kustom tidak mendukung kolom dengan spasi atau karakter khusus dalam nama. Kami menyarankan Anda menentukan nama kolom yang hanya memiliki karakter alfanumerik dan garis bawah. Anda dapat menggunakan Transformasi kolom Rename di grup Mengelola kolom transformasi untuk menghapus spasi dari nama kolom. Anda juga dapat menambahkan Python (Pandas) Custom transform mirip dengan berikut ini untuk menghapus spasi dari beberapa kolom dalam satu langkah. Contoh ini mengubah kolom bernama `A column` dan `B column` ke `A_column` dan `B_column` masing-masing.
>
> ```
> df.rename(columns={"A column": "A_column", "B column": "B_column"})
> ```

Anda dapat menggunakan transformasi ini untuk melakukan operasi pada kolom, mereferensikan kolom dengan nama. Misalnya, dengan asumsi kerangka data saat ini berisi kolom bernama col_a dan col_b, Anda dapat menggunakan operasi berikut untuk menghasilkan kolom Output yang merupakan produk dari dua kolom ini dengan kode berikut:

```
col_a * col_b
```

Operasi umum lainnya termasuk yang berikut, dengan asumsi kerangka data berisi dan kolom: `col_a` `col_b`

- Gandungkan dua kolom: `concat(col_a, col_b)`
- Tambahkan dua kolom: `col_a + col_b`
- Kurangi dua kolom: `col_a - col_b`
- Bagilah dua kolom: `col_a / col_b`
- Ambil nilai absolut kolom: `abs(col_a)`

Untuk informasi selengkapnya, lihat dokumentasi percikan tentang pemilihan data.

Mengurangi Dimensionalitas dalam Dataset

Kurangi dimensi dalam data Anda dengan menggunakan Principal Component Analysis (PCA). Dimensi kumpulan data Anda sesuai dengan jumlah fitur. Saat Anda menggunakan pengurangan dimensi di Data Wrangler, Anda mendapatkan serangkaian fitur baru yang disebut komponen. Setiap komponen memperhitungkan beberapa variabilitas dalam data.

Komponen pertama menyumbang jumlah variasi terbesar dalam data. Komponen kedua menyumbang jumlah variasi terbesar kedua dalam data, dan seterusnya.

Anda dapat menggunakan pengurangan dimensi untuk mengurangi ukuran kumpulan data yang Anda gunakan untuk melatih model. Alih-alih menggunakan fitur dalam kumpulan data Anda, Anda dapat menggunakan komponen utama sebagai gantinya.

Untuk melakukan PCA, Data Wrangler membuat sumbu untuk data Anda. Sumbu adalah kombinasi affine kolom dalam kumpulan data Anda. Komponen utama pertama adalah nilai pada sumbu yang memiliki jumlah varians terbesar. Komponen utama kedua adalah nilai pada sumbu yang memiliki jumlah varians terbesar kedua. Komponen utama ke-n adalah nilai pada sumbu yang memiliki jumlah varians terbesar ke-n.

Anda dapat mengonfigurasi jumlah komponen utama yang dikembalikan Data Wrangler. Anda dapat menentukan jumlah komponen utama secara langsung atau Anda dapat menentukan persentase ambang varians. Setiap komponen utama menjelaskan sejumlah varians dalam data. Misalnya, Anda mungkin memiliki komponen utama dengan nilai 0,5. Komponen akan menjelaskan 50% variasi dalam data. Saat Anda menentukan persentase ambang varians, Data Wrangler mengembalikan jumlah komponen terkecil yang memenuhi persentase yang Anda tentukan.

Berikut ini adalah contoh komponen utama dengan jumlah varians yang mereka jelaskan dalam data.

- 1 = 50
- Komponen 2 - 0.45
- Komponen 3 - 0.05

Jika Anda menentukan persentase ambang batas varians dari 94 or95, Data Wrangler mengembalikan Komponen 1 dan Komponen 2. Jika Anda menentukan persentase ambang varians dari96, Data Wrangler mengembalikan ketiga komponen utama.

Anda dapat menggunakan prosedur berikut untuk menjalankan PCA pada dataset Anda.

Untuk menjalankan PCA pada dataset Anda, lakukan hal berikut.

1. Buka aliran data Wrangler Data Anda.
2. Pilih +, dan pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Pengurangan Dimensi.
5. Untuk Kolom Input, pilih fitur yang Anda kurangi menjadi komponen utama.
6. (Opsional) Untuk Jumlah komponen utama, pilih jumlah komponen utama yang dikembalikan Data Wrangler dalam kumpulan data Anda. Jika menentukan nilai untuk bidang, Anda tidak dapat menentukan nilai untuk persentase ambang batas Varians.
7. (Opsional) Untuk persentase ambang batas Varians, tentukan persentase variasi dalam data yang ingin Anda jelaskan oleh komponen utama. Data Wrangler menggunakan nilai default 95 jika Anda tidak menentukan nilai untuk ambang batas varians. Anda tidak dapat menentukan persentase ambang varians jika Anda telah menentukan nilai untuk Jumlah komponen utama.
8. (Opsional) Batalkan pilihan Pusat untuk tidak menggunakan rata-rata kolom sebagai pusat data. Secara default, Data Wrangler memusatkan data dengan mean sebelum penskalaan.
9. (Opsional) Batalkan pilihan Skala untuk tidak menskalakan data dengan standar deviasi unit.
10. (Opsional) Pilih Kolom untuk menampilkan komponen ke kolom terpisah. Pilih Vector untuk menampilkan komponen sebagai vektor tunggal.
11. (Opsional) Untuk kolom Output, tentukan nama untuk kolom keluaran. Jika Anda mengeluarkan komponen ke kolom terpisah, nama yang Anda tentukan adalah awalan. Jika Anda mengeluarkan komponen ke vektor, nama yang Anda tentukan adalah nama kolom vektor.
12. (Opsional) Pilih Simpan kolom input. Kami tidak menyarankan memilih opsi ini jika Anda berencana hanya menggunakan komponen utama untuk melatih model Anda.
13. Pilih Pratinjau.
14. Pilih Tambahkan.

Encode Kategoris

Data kategoris biasanya terdiri dari sejumlah kategori yang terbatas, di mana setiap kategori diwakili dengan string. Misalnya, jika Anda memiliki tabel data pelanggan, kolom yang menunjukkan negara tempat seseorang tinggal adalah kategoris. Kategori-kategorinya adalah Afghanistan, Albania, Aljazair, dan sebagainya. Data kategoris dapat berupa nominal atau ordinal. Kategori ordinal memiliki

urutan yang melekat, dan kategori nominal tidak. Gelar tertinggi yang diperoleh (SMA, Sarjana, Magister, dan sebagainya) adalah contoh kategori ordinal.

Pengkodean data kategoris adalah proses menciptakan representasi numerik untuk kategori. Misalnya, jika kategori Anda adalah Dog dan Cat, Anda dapat menyandikan informasi ini menjadi dua vektor, *[1,0]* untuk mewakili Dog, dan *[0,1]* untuk mewakili Cat.

Saat Anda menyandikan kategori ordinal, Anda mungkin perlu menerjemahkan urutan alami kategori ke dalam pengkodean Anda. Misalnya, Anda dapat mewakili derajat tertinggi yang diperoleh dengan peta berikut:`{"High school": 1, "Bachelors": 2, "Masters":3}`.

Gunakan pengkodean kategoris untuk menyandikan data kategoris yang dalam format string ke dalam array bilangan bulat.

Encoder kategoris Data Wrangler membuat pengkodean untuk semua kategori yang ada di kolom pada saat langkah ditentukan. Jika kategori baru telah ditambahkan ke kolom saat Anda memulai pekerjaan Data Wrangler untuk memproses kumpulan data Anda pada waktu t, dan kolom ini adalah masukan untuk transformasi pengkodean kategoris Data Wrangler pada waktu t -1, kategori baru ini dianggap hilang dalam pekerjaan Data Wrangler. Opsi yang Anda pilih untuk Strategi penanganan tidak valid diterapkan pada nilai yang hilang ini. Contoh kapan ini dapat terjadi adalah:

- Saat Anda menggunakan file.flow untuk membuat pekerjaan Data Wrangler untuk memproses kumpulan data yang diperbarui setelah pembuatan aliran data. Misalnya, Anda dapat menggunakan aliran data untuk memproses data penjualan secara teratur setiap bulan. Jika data penjualan diperbarui setiap minggu, kategori baru dapat dimasukkan ke dalam kolom yang menentukan langkah kategoris encode.
- Ketika Anda memilih Sampling ketika Anda mengimpor dataset Anda, beberapa kategori mungkin ditinggalkan dari sampel.

Dalam situasi ini, kategori baru ini dianggap nilai yang hilang dalam pekerjaan Data Wrangler.

Anda dapat memilih dari dan mengkonfigurasi ordinal dan encode satu-panas. Gunakan bagian berikut untuk mempelajari lebih lanjut tentang opsi ini.

Kedua transformasi membuat kolom baru bernama Output nama kolom. Anda menentukan format output kolom ini dengan gaya Output:

- Pilih Vektor untuk menghasilkan satu kolom dengan vektor jarang.

- Pilih Kolom untuk membuat kolom untuk setiap kategori dengan variabel indikator apakah teks di kolom asli berisi nilai yang sama dengan kategori tersebut.

Pengkodean Ordinal

Pilih Ordinal encode untuk menyandikan kategori menjadi bilangan bulat antara 0 dan jumlah total kategori di kolom Input yang Anda pilih.

Strategi penyerahan tidak valid: Pilih metode untuk menangani nilai yang tidak valid atau hilang.

- Pilih Lewati jika Anda ingin menghilangkan baris dengan nilai yang hilang.
- Pilih Simpan untuk mempertahankan nilai yang hilang sebagai kategori terakhir.
- Pilih Kesalahan jika Anda ingin Data Wrangler melempar kesalahan jika nilai yang hilang ditemukan di kolom Input.
- Pilih Ganti dengan NaN untuk mengganti yang hilang dengan NaN. Opsi ini direkomendasikan jika algoritme ML Anda dapat menangani nilai yang hilang. Jika tidak, tiga opsi pertama dalam daftar ini dapat menghasilkan hasil yang lebih baik.

One-Hot Encode

Pilih One-hot encode untuk Transform untuk menggunakan one-hot encoding. Konfigurasikan transformasi ini menggunakan yang berikut:

- Jatuhkan kategori terakhir: Jika`True`, kategori terakhir tidak memiliki indeks yang sesuai dalam pengkodean satu panas. Ketika nilai yang hilang dimungkinkan, kategori yang hilang selalu menjadi yang terakhir dan `True` menyetelnya berarti bahwa nilai yang hilang menghasilkan vektor nol.
- Strategi penyerahan tidak valid: Pilih metode untuk menangani nilai yang tidak valid atau hilang.
  - Pilih Lewati jika Anda ingin menghilangkan baris dengan nilai yang hilang.
  - Pilih Simpan untuk mempertahankan nilai yang hilang sebagai kategori terakhir.
  - Pilih Kesalahan jika Anda ingin Data Wrangler melempar kesalahan jika nilai yang hilang ditemukan di kolom Input.
- Apakah input ordinal dikodekan: Pilih opsi ini jika vektor input berisi data yang dikodekan ordinal. Opsi ini mengharuskan data input mengandung bilangan bulat non-negatif. Jika Benar, masukan i dikodekan sebagai vektor dengan bukan nol di lokasi ke-i.

Kesamaan menyandikan

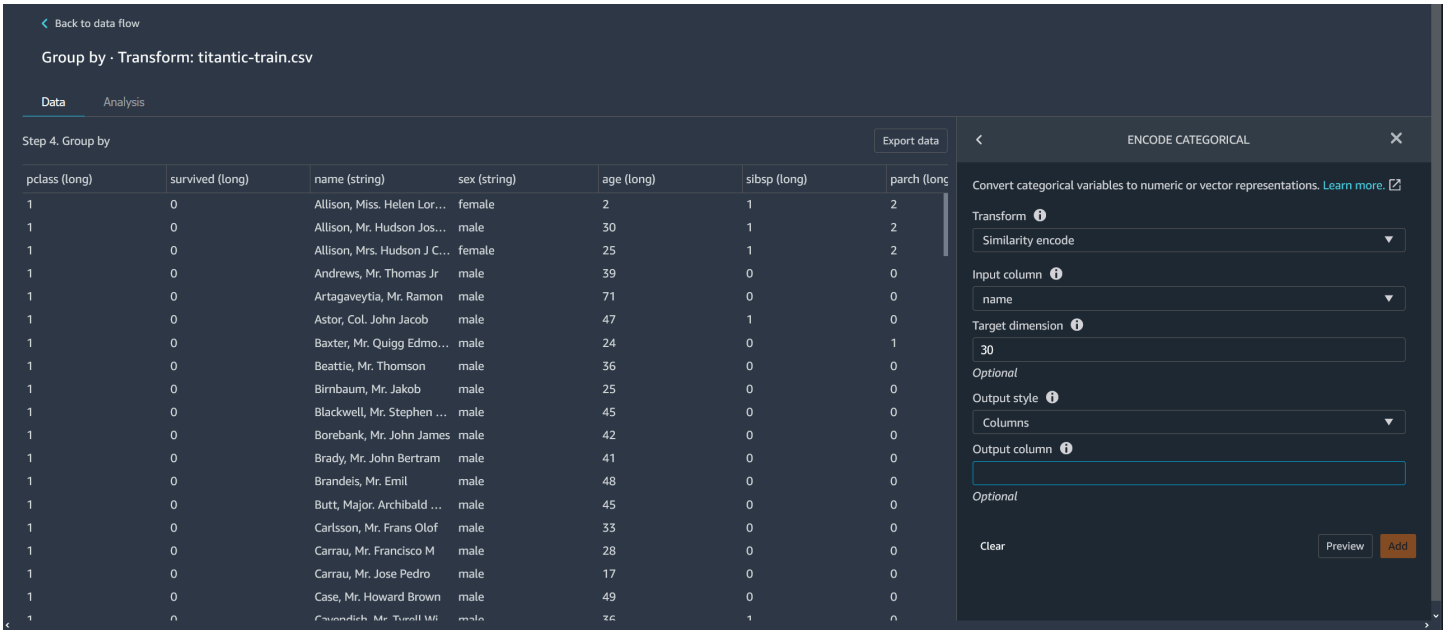
Gunakan pengkodean kesamaan ketika Anda memiliki yang berikut:

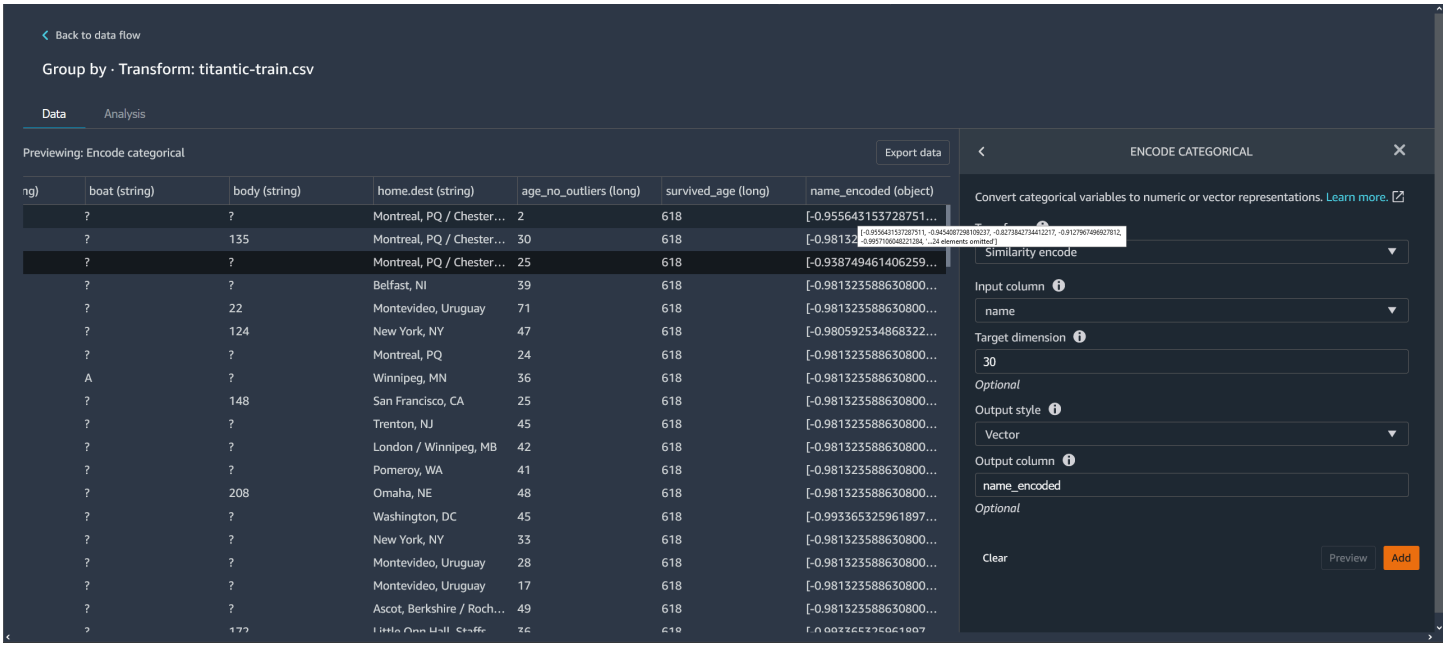
- Sejumlah besar variabel kategoris
- Data berisik

Encoder kesamaan menciptakan embeddings untuk kolom dengan data kategoris. Embedding adalah pemetaan objek diskrit, seperti kata-kata, ke vektor bilangan real. Ini mengkodekan string serupa ke vektor yang mengandung nilai serupa. Misalnya, ia menciptakan pengkodean yang sangat mirip untuk “California” dan “Calfornia”.

Data Wrangler mengonversi setiap kategori dalam kumpulan data Anda menjadi satu set token menggunakan tokenizer 3 gram. Ini mengubah token menjadi embedding menggunakan encoding min-hash.

Contoh berikut menunjukkan cara encoder kesamaan menciptakan vektor dari string.





Pengkodean kesamaan yang dibuat Data Wrangler:

- Memiliki dimensi rendah
- Dapat diskalakan ke sejumlah besar kategori
- Kuat dan tahan terhadap kebisingan

Untuk alasan sebelumnya, pengkodean kesamaan lebih fleksibel daripada pengkodean satu panas.

Untuk menambahkan transformasi pengkodean kesamaan ke kumpulan data Anda, gunakan prosedur berikut.

Untuk menggunakan encoding kesamaan, lakukan hal berikut.

1. Masuk ke SageMakerKonsol Amazon.
2. Pilih Open Studio Classic.
3. Pilih Luncurkan aplikasi.
4. Pilih Studio.
5. Tentukan aliran data Anda.
6. Pilih langkah dengan transformasi.
7. Pilih Tambahkan langkah.
8. Pilih Encode kategoris.

9. Tentukan hal berikut:

    - Transform - Encode kesamaan
    - Kolom input - Kolom yang berisi data kategoris yang Anda enkodekan.
    - Dimensi target — (Opsional) Dimensi vektor embedding kategoris. Nilai default-nya adalah 30. Sebaiknya gunakan dimensi target yang lebih besar jika Anda memiliki kumpulan data besar dengan banyak kategori.
    - Gaya keluaran — Pilih Vektor untuk vektor tunggal dengan semua nilai yang dikodekan. Pilih Kolom untuk memiliki nilai yang dikodekan di kolom terpisah.
    - Kolom keluaran - (Opsional) Nama kolom keluaran untuk output yang dikodekan vektor. Untuk output yang dikodekan kolom, ini adalah awalan dari nama kolom diikuti dengan nomor yang terdaftar.

Ikhtisar

Gunakan grup transformasi Teks Featurize untuk memeriksa kolom yang diketik string dan gunakan penyematan teks untuk menyesuaikan kolom ini.

Grup fitur ini berisi dua fitur, statistik Karakter dan Vektor. Gunakan bagian berikut untuk mempelajari lebih lanjut tentang perubahan ini. Untuk kedua opsi, kolom Input harus berisi data teks (tipe string).

Statistik Karakter

Gunakan statistik Karakter untuk menghasilkan statistik untuk setiap baris dalam kolom yang berisi data teks.

Transformasi ini menghitung rasio dan hitungan berikut untuk setiap baris, dan membuat kolom baru untuk melaporkan hasilnya. Kolom baru diberi nama menggunakan nama kolom input sebagai awalan dan akhiran yang spesifik untuk rasio atau hitungan.

- Jumlah kata: Jumlah kata dalam baris itu. Sufiks untuk kolom keluaran ini adalah`-stats_word_count`.
- Jumlah karakter: Jumlah total karakter di baris itu. Sufiks untuk kolom keluaran ini adalah`-stats_char_count`.
- Rasio atas: Jumlah karakter huruf besar, dari A hingga Z, dibagi dengan semua karakter di kolom. Sufiks untuk kolom keluaran ini adalah`-stats_capital_ratio`.
- Rasio yang lebih rendah: Jumlah karakter huruf kecil, dari a hingga z, dibagi dengan semua karakter di kolom. Sufiks untuk kolom keluaran ini adalah`-stats_lower_ratio`.

- Rasio digit: Rasio digit dalam satu baris di atas jumlah digit di kolom input. Sufiks untuk kolom keluaran ini adalah`-stats_digit_ratio`.
- Rasio karakter khusus: Rasio karakter non-alfanumerik (seperti # $&%: @) terhadap jumlah semua karakter di kolom input. Sufiks untuk kolom keluaran ini adalah`-stats_special_ratio`.

Vektorisasi

Penyematan teks melibatkan pemetaan kata atau frasa dari kosakata ke vektor bilangan real. Gunakan transformasi penyematan teks Data Wrangler untuk memberi token dan memvektorisasi data teks menjadi vektor frekuensi terminal-inverse document frequency (TF-IDF).

Ketika TF-IDF dihitung untuk kolom data teks, setiap kata dalam setiap kalimat diubah menjadi bilangan real yang mewakili kepentingan semantiknya. Angka yang lebih tinggi dikaitkan dengan kata-kata yang lebih jarang, yang cenderung lebih bermakna.

Saat Anda menentukan langkah transformasi Vektor, Data Wrangler menggunakan data dalam kumpulan data Anda untuk menentukan metode count vectorizer dan TF-IDF. Menjalankan pekerjaan Data Wrangler menggunakan metode yang sama.

Anda mengonfigurasi transformasi ini menggunakan yang berikut:

- Nama kolom keluaran: Transformasi ini membuat kolom baru dengan penyematan teks. Gunakan bidang ini untuk menentukan nama untuk kolom keluaran ini.
- Tokenizer: Tokenizer mengubah kalimat menjadi daftar kata, atau token.

  Pilih Standar untuk menggunakan tokenizer yang dibagi dengan spasi putih dan mengubah setiap kata menjadi huruf kecil. Misalnya, `"Good dog"` diberi token ke. `["good","dog"]`

  Pilih Custom untuk menggunakan tokenizer yang disesuaikan. Jika Anda memilih Custom, Anda dapat menggunakan bidang berikut untuk mengkonfigurasi tokenizer:
  - Panjang token minimum: Panjang minimum, dalam karakter, agar token valid. Default ke `1`. Misalnya, jika Anda menentukan `3` panjang token minimum, kata-kata seperti `a, at, in` dijatuhkan dari kalimat tokenized.
  - Haruskah regex terbelah pada celah: Jika dipilih, regex terbagi pada celah. Jika tidak, itu cocok dengan token. Default ke `True`.
  - Pola Regex: Pola Regex yang mendefinisikan proses tokenisasi. Default ke `' \\ s+'`.
  - Untuk huruf kecil: Jika dipilih, Data Wrangler mengonversi semua karakter menjadi huruf kecil sebelum tokenisasi. Default ke `True`.

Untuk mempelajari lebih lanjut, lihat dokumentasi Spark di Tokenizer.

- Vectorizer: Vectorizer mengubah daftar token menjadi vektor numerik jarang. Setiap token sesuai dengan indeks dalam vektor dan bukan nol menunjukkan keberadaan token dalam kalimat input. Anda dapat memilih dari dua opsi vectorizer, Count dan Hashing.
  - Count vectorize memungkinkan penyesuaian yang memfilter token yang jarang atau terlalu umum. Parameter vektorisasi hitung meliputi yang berikut:
    - Frekuensi istilah minimum: Di setiap baris, istilah (token) dengan frekuensi yang lebih kecil disaring. Jika Anda menentukan bilangan bulat, ini adalah ambang absolut (inklusif). Jika Anda menentukan pecahan antara 0 (inklusif) dan 1, ambang batas relatif terhadap jumlah suku total. Default ke `1`.
    - Frekuensi dokumen minimum: Jumlah baris minimum di mana istilah (token) harus muncul untuk disertakan. Jika Anda menentukan bilangan bulat, ini adalah ambang absolut (inklusif). Jika Anda menentukan pecahan antara 0 (inklusif) dan 1, ambang batas relatif terhadap jumlah suku total. Default ke `1`.
    - Frekuensi dokumen maksimum: Jumlah maksimum dokumen (baris) di mana istilah (token) dapat muncul untuk dimasukkan. Jika Anda menentukan bilangan bulat, ini adalah ambang absolut (inklusif). Jika Anda menentukan pecahan antara 0 (inklusif) dan 1, ambang batas relatif terhadap jumlah suku total. Default ke `0.999`.
    - Ukuran kosakata maksimum: Ukuran maksimum kosakata. Kosakata terdiri dari semua istilah (token) di semua baris kolom. Default ke `262144`.
    - Output biner: Jika dipilih, output vektor tidak termasuk jumlah penampilan suatu istilah dalam dokumen, melainkan merupakan indikator biner dari penampilannya. Default ke `False`.

    Untuk mempelajari lebih lanjut tentang opsi ini, lihat dokumentasi Spark di CountVectorizer.
  - Hashing secara komputasi lebih cepat. Parameter vektor hash meliputi yang berikut:
    - Jumlah fitur selama hashing: Sebuah hash vectorizer memetakan token ke indeks vektor sesuai dengan nilai hash mereka. Fitur ini menentukan jumlah nilai hash yang mungkin. Nilai yang besar menghasilkan lebih sedikit tabrakan antara nilai hash tetapi vektor keluaran dimensi yang lebih tinggi.

    Untuk mempelajari lebih lanjut tentang opsi ini, lihat dokumentasi Spark di FeatureHasher
- Terapkan IDF menerapkan transformasi IDF, yang mengalikan frekuensi istilah dengan frekuensi dokumen terbalik standar yang digunakan untuk penyematan TF-IDF. Parameter IDF meliputi:

  - Frekuensi dokumen minimum: Jumlah minimum dokumen (baris) di mana istilah (token) harus muncul untuk disertakan. Jika count_vectorize adalah vectorizer yang dipilih, kami sarankan Anda menyimpan nilai default dan hanya memodifikasi bidang min_doc_freq dalam parameter Count vectorize. Default ke 5.
- Format output: Format output dari setiap baris.
  - Pilih Vektor untuk menghasilkan satu kolom dengan vektor jarang.
  - Pilih Flattened untuk membuat kolom untuk setiap kategori dengan variabel indikator apakah teks di kolom asli berisi nilai yang sama dengan kategori tersebut. Anda hanya dapat memilih diratakan ketika Vectorizer ditetapkan sebagai Count vectorizer.

Mengubah Seri Waktu

Di Data Wrangler, Anda dapat mengubah data deret waktu. Nilai dalam dataset deret waktu diindeks ke waktu tertentu. Misalnya, kumpulan data yang menunjukkan jumlah pelanggan di toko untuk setiap jam dalam sehari adalah kumpulan data deret waktu. Tabel berikut menunjukkan contoh dataset deret waktu.

Jumlah pelanggan per jam di toko

Jumlah tindakan	Waktu (jam)
4	09:00
10	10:00
14	11:00
25	12:00
20	13:00
18	14:00

Untuk tabel sebelumnya, kolom Jumlah Pelanggan berisi data deret waktu. Data deret waktu diindeks pada data per jam di kolom Waktu (jam).

Anda mungkin perlu melakukan serangkaian transformasi pada data Anda untuk mendapatkannya dalam format yang dapat Anda gunakan untuk analisis Anda. Gunakan grup transformasi deret waktu

untuk mengubah data deret waktu Anda. Untuk informasi selengkapnya tentang transformasi yang dapat Anda lakukan, lihat bagian berikut.

Topik

Kelompokkan berdasarkan Time Series

Anda dapat menggunakan grup berdasarkan operasi untuk mengelompokkan data deret waktu untuk nilai tertentu dalam kolom.

Misalnya, Anda memiliki tabel berikut yang melacak rata-rata penggunaan listrik harian dalam rumah tangga.

Rata-rata penggunaan listrik rumah tangga harian

ID Rumah Tangga	Default-nya	Penggunaan listrik (kWh)	Jumlah Penghuni Rumah Tangga
rumah tangga_0	1/1/2020	30	2
rumah tangga_0	1 = 50	40	2
rumah tangga_0	1/4/2020	35	3
rumah tangga_1	1 = 50	45	3
rumah tangga_1	1/3/2020	55	4

Jika Anda memilih untuk mengelompokkan berdasarkan ID, Anda mendapatkan tabel berikut.

Penggunaan listrik dikelompokkan berdasarkan ID rumah tangga

ID Rumah Tangga	Seri penggunaan listrik (kWh)	Jumlah seri penghuni rumah tangga
rumah tangga_0	[30, 40, 35]	[2, 2, 3]
rumah tangga_1	[45, 55]	[3, 4]

Setiap entri dalam urutan deret waktu diurutkan oleh stempel waktu yang sesuai. Elemen pertama dari urutan sesuai dengan stempel waktu pertama dari seri. Untuk`household_0`, `30` adalah nilai pertama dari Seri Penggunaan Listrik. Nilai `30` sesuai dengan stempel waktu pertama. `1/1/2020`

Anda dapat menyertakan stempel waktu awal dan stempel waktu akhir. Tabel berikut menunjukkan bagaimana informasi tersebut muncul.

Penggunaan listrik dikelompokkan berdasarkan ID rumah tangga

ID Rumah Tangga	Seri penggunaan listrik (kWh)	Jumlah seri penghuni rumah tangga	Mulai_waktu	Akhir waktu
rumah tangga_0	[30, 40, 35]	[2, 2, 3]	1/1/2020	1/4/2020
rumah tangga_1	[45, 55]	[3, 4]	1 = 50	1/3/2020

Anda dapat menggunakan prosedur berikut untuk mengelompokkan berdasarkan kolom deret waktu.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
4. Pilih Tambahkan langkah.
5. Pilih Time Series.
6. Di bawah Transform, pilih Group by.
7. Tentukan kolom di Grup menurut kolom ini.

8. Untuk Terapkan ke kolom, tentukan nilai.
9. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
10. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Sampel Ulang Data Seri Waktu

Data deret waktu biasanya memiliki pengamatan yang tidak diambil secara berkala. Misalnya, kumpulan data dapat memiliki beberapa pengamatan yang direkam setiap jam dan pengamatan lain yang dicatat setiap dua jam.

Banyak analisis, seperti algoritma peramalan, memerlukan pengamatan yang harus dilakukan secara berkala. Resampling memberi Anda kemampuan untuk menetapkan interval reguler untuk pengamatan dalam kumpulan data Anda.

Anda dapat melakukan upsample atau downsample deret waktu. Downsampling meningkatkan interval antara pengamatan dalam dataset. Misalnya, jika Anda menurunkan sampel pengamatan yang diambil setiap jam atau setiap dua jam, setiap pengamatan dalam kumpulan data Anda dilakukan setiap dua jam. Pengamatan per jam dikumpulkan menjadi satu nilai menggunakan metode agregasi seperti mean atau median.

Upsampling mengurangi interval antara pengamatan dalam dataset. Misalnya, jika Anda mengambil sampel pengamatan yang dilakukan setiap dua jam ke dalam pengamatan per jam, Anda dapat menggunakan metode interpolasi untuk menyimpulkan pengamatan per jam dari pengamatan yang dilakukan setiap dua jam. Untuk informasi tentang metode interpolasi, lihat panda. DataFrame.interpolasi.

Anda dapat mengambil sampel ulang data numerik dan non-numerik.

Gunakan operasi Sampel Ulang untuk mengambil sampel ulang data deret waktu Anda. Jika Anda memiliki beberapa deret waktu dalam kumpulan data Anda, Data Wrangler menstandarisasi interval waktu untuk setiap deret waktu.

Tabel berikut menunjukkan contoh data deret waktu downsampling dengan menggunakan mean sebagai metode agregasi. Data di-downsample dari setiap dua jam menjadi setiap jam.

Pembacaan suhu per jam selama sehari sebelum downsampling

Stempel Waktu	Suhu (Celcius)
12:00	30

Stempel Waktu	Suhu (Celcius)
1:00	32
2:00	35
3:00	32
4:00	30

Pembacaan suhu diturunkan sampelnya menjadi setiap dua jam

Stempel Waktu	Suhu (Celcius)
12:00	30
2:00	33.5
4:00	35

Anda dapat menggunakan prosedur berikut untuk mengambil data deret waktu.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
4. Pilih Tambahkan langkah.
5. Pilih Sampel Ulang.
6. Untuk Timestamp, pilih kolom timestamp.
7. Untuk unit Frekuensi, tentukan frekuensi yang Anda resampling.
8. (Opsional) Tentukan nilai untuk kuantitas Frekuensi.
9. Konfigurasi transformasi dengan menentukan bidang yang tersisa.
10. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
11. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Tangani Data Seri Waktu yang Hilang

Jika Anda memiliki nilai yang hilang dalam kumpulan data Anda, Anda dapat melakukan salah satu hal berikut:

- Untuk kumpulan data yang memiliki beberapa deret waktu, lepaskan deret waktu yang memiliki nilai hilang yang lebih besar dari ambang batas yang Anda tentukan.
- Imputasi nilai yang hilang dalam deret waktu dengan menggunakan nilai lain dalam deret waktu.

Mengimplikasikan nilai yang hilang melibatkan penggantian data dengan menentukan nilai atau dengan menggunakan metode inferensial. Berikut ini adalah metode yang dapat Anda gunakan untuk imputasi:

- Nilai konstan — Ganti semua data yang hilang dalam dataset Anda dengan nilai yang Anda tentukan.
- Nilai paling umum — Ganti semua data yang hilang dengan nilai yang memiliki frekuensi tertinggi dalam kumpulan data.
- Forward fill — Gunakan forward fill untuk mengganti nilai yang hilang dengan nilai yang tidak hilang yang mendahului nilai yang hilang. Untuk urutan: [2, 4, 7, NaN, NaN, NaN, 8], semua nilai yang hilang diganti dengan 7. Urutan yang dihasilkan dari penggunaan isian maju adalah [2, 4, 7, 7, 7, 7, 8].
- Isi mundur - Gunakan pengisian mundur untuk mengganti nilai yang hilang dengan nilai yang tidak hilang yang mengikuti nilai yang hilang. Untuk urutan: [2, 4, 7, NaN, NaN, NaN, 8], semua nilai yang hilang diganti dengan 8. Urutan yang dihasilkan dari penggunaan pengisian mundur adalah [2, 4, 7, 8, 8, 8, 8].
- Interpolasi — Menggunakan fungsi interpolasi untuk menghitung nilai yang hilang. Untuk informasi selengkapnya tentang fungsi yang dapat Anda gunakan untuk interpolasi, lihat panda. DataFrame.interpolasi.

Beberapa metode imputasi mungkin tidak dapat memperhitungkan semua nilai yang hilang dalam kumpulan data Anda. Misalnya, Forward fill tidak dapat menyiratkan nilai yang hilang yang muncul di awal deret waktu. Anda dapat mengimputasi nilai dengan menggunakan isian maju atau pengisian mundur.

Anda dapat memasukkan nilai yang hilang di dalam sel atau di dalam kolom.

Contoh berikut menunjukkan bagaimana nilai diperhitungkan dalam sel.

Penggunaan listrik dengan nilai yang hilang

ID Rumah Tangga	Seri penggunaan listrik (kWh)
rumah tangga_0	[30, 40, 35, NaN, NaN]
rumah tangga_1	[45, NaN, 55]

Penggunaan listrik dengan nilai yang diperhitungkan menggunakan pengisian ke depan

ID Rumah Tangga	Seri penggunaan listrik (kWh)
rumah tangga_0	[30, 40, 35, 35, 35]
rumah tangga_1	[45, 45, 55]

Contoh berikut menunjukkan bagaimana nilai diperhitungkan dalam kolom.

Rata-rata penggunaan listrik rumah tangga harian dengan nilai yang hilang

ID Rumah Tangga	Penggunaan listrik (kWh)
rumah tangga_0	30
rumah tangga_0	40
rumah tangga_0	NaN
rumah tangga_1	NaN
rumah tangga_1	NaN

Rata-rata penggunaan listrik rumah tangga harian dengan nilai yang diperhitungkan menggunakan pengisian ke depan

ID Rumah Tangga	Penggunaan listrik (kWh)
rumah tangga_0	30

ID Rumah Tangga	Penggunaan listrik (kWh)
rumah tangga_0	40
rumah tangga_0	40
rumah tangga_1	40
rumah tangga_1	40

Anda dapat menggunakan prosedur berikut untuk menangani nilai yang hilang.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
4. Pilih Tambahkan langkah.
5. Pilih Handle hilang.
6. Untuk jenis input deret waktu, pilih apakah Anda ingin menangani nilai yang hilang di dalam sel atau di sepanjang kolom.
7. Untuk Impute nilai yang hilang untuk kolom ini, tentukan kolom yang memiliki nilai yang hilang.
8. Untuk Metode untuk menghitung nilai, pilih metode.
9. Konfigurasi transformasi dengan menentukan bidang yang tersisa.
10. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
11. Jika Anda memiliki nilai yang hilang, Anda dapat menentukan metode untuk mengimplikasikan mereka di bawah Metode untuk memasukkan nilai.
12. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Validasi Stempel Waktu Data Deret Waktu Anda

Anda mungkin memiliki data stempel waktu yang tidak valid. Anda dapat menggunakan fungsi Validasi stempel waktu untuk menentukan apakah stempel waktu dalam kumpulan data Anda valid. Stempel waktu Anda mungkin tidak valid karena satu atau beberapa alasan berikut:

- Kolom stempel waktu Anda memiliki nilai yang hilang.
- Nilai di kolom stempel waktu Anda tidak diformat dengan benar.

Jika Anda memiliki stempel waktu yang tidak valid dalam kumpulan data, Anda tidak dapat melakukan analisis dengan sukses. Anda dapat menggunakan Data Wrangler untuk mengidentifikasi stempel waktu yang tidak valid dan memahami di mana Anda perlu membersihkan data Anda.

Validasi deret waktu bekerja dalam salah satu dari dua cara:

Anda dapat mengonfigurasi Data Wrangler untuk melakukan salah satu hal berikut jika menemukan nilai yang hilang di kumpulan data Anda:

- Jatuhkan baris yang memiliki nilai hilang atau tidak valid.
- Identifikasi baris yang memiliki nilai hilang atau tidak valid.
- Lempar kesalahan jika menemukan nilai yang hilang atau tidak valid di kumpulan data Anda.

Anda dapat memvalidasi stempel waktu pada kolom yang memiliki `timestamp` tipe atau jenisnya. `string` Jika kolom memiliki `string` tipe, Data Wrangler mengubah jenis kolom ke `timestamp` dan melakukan validasi.

Anda dapat menggunakan prosedur berikut untuk memvalidasi stempel waktu dalam kumpulan data Anda.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
4. Pilih Tambahkan langkah.
5. Pilih Validasi stempel waktu.
6. Untuk Timestamp Column, pilih kolom timestamp.
7. Untuk Kebijakan, pilih apakah Anda ingin menangani stempel waktu yang hilang.
8. (Opsional) Untuk kolom Output, tentukan nama untuk kolom output.
9. Jika kolom waktu tanggal diformat untuk jenis string, pilih Cast to datetime.
10. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
11. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Standardisasi Panjang Deret Waktu

Jika Anda memiliki data deret waktu yang disimpan sebagai array, Anda dapat menstandarisasi setiap deret waktu dengan panjang yang sama. Standarisasi panjang array deret waktu mungkin memudahkan Anda untuk melakukan analisis pada data.

Anda dapat menstandarisasi deret waktu Anda untuk transformasi data yang memerlukan panjang data Anda untuk diperbaiki.

Banyak algoritma ML mengharuskan Anda untuk meratakan data deret waktu Anda sebelum Anda menggunakannya. Meratakan data deret waktu memisahkan setiap nilai deret waktu menjadi kolomnya sendiri dalam kumpulan data. Jumlah kolom dalam kumpulan data tidak dapat berubah, sehingga panjang deret waktu perlu distandarisasi antara Anda meratakan setiap array menjadi satu set fitur.

Setiap deret waktu diatur ke panjang yang Anda tentukan sebagai kuantil atau persentil dari rangkaian deret waktu. Misalnya, Anda dapat memiliki tiga urutan yang memiliki panjang berikut:

- 3
- 4
- 5

Anda dapat mengatur panjang semua urutan sebagai panjang urutan yang memiliki panjang persentil ke-50.

Array deret waktu yang lebih pendek dari panjang yang Anda tentukan memiliki nilai yang hilang ditambahkan. Berikut ini adalah contoh format standarisasi deret waktu ke panjang yang lebih panjang: [2, 4, 5, NaN, NaN, NaN].

Anda dapat menggunakan pendekatan yang berbeda untuk menangani nilai yang hilang. Untuk informasi tentang pendekatan tersebut, lihatTangani Data Seri Waktu yang Hilang.

Array deret waktu yang lebih panjang dari panjang yang Anda tentukan terpotong.

Anda dapat menggunakan prosedur berikut untuk menstandarisasi panjang deret waktu.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.

4. Pilih Tambahkan langkah.
5. Pilih Standarisasi panjang.
6. Untuk Standarisasi panjang deret waktu untuk kolom, pilih kolom.
7. (Opsional) Untuk kolom Output, tentukan nama untuk kolom output. Jika Anda tidak menentukan nama, perubahan dilakukan di tempat.
8. Jika kolom datetime diformat untuk jenis string, pilih Cast to datetime.
9. Pilih Cutoff quantile dan tentukan kuantil untuk mengatur panjang urutan.
10. Pilih Ratakan output untuk menampilkan nilai deret waktu ke dalam kolom terpisah.
11. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
12. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Ekstrak Fitur dari Data Seri Waktu Anda

Jika Anda menjalankan klasifikasi atau algoritma regresi pada data deret waktu Anda, sebaiknya ekstrak fitur dari deret waktu sebelum menjalankan algoritme. Mengekstrak fitur dapat meningkatkan kinerja algoritme Anda.

Gunakan opsi berikut untuk memilih bagaimana Anda ingin mengekstrak fitur dari data Anda:

- Gunakan subset Minimal untuk menentukan ekstraksi 8 fitur yang Anda tahu berguna dalam analisis hilir. Anda dapat menggunakan subset minimal saat Anda perlu melakukan perhitungan dengan cepat. Anda juga dapat menggunakannya ketika algoritme ML Anda memiliki risiko overfitting yang tinggi dan Anda ingin menyediakannya dengan lebih sedikit fitur.
- Gunakan subset Efisien untuk menentukan penggalian fitur sebanyak mungkin tanpa mengekstraksi fitur yang intensif secara komputasi dalam analisis Anda.
- Gunakan Semua fitur untuk menentukan ekstraksi semua fitur dari seri lagu.
- Gunakan subset Manual untuk memilih daftar fitur yang menurut Anda menjelaskan variasi data Anda dengan baik.

Gunakan prosedur berikut ini untuk mengekstrak fitur dari data deret waktu Anda.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.

4. Pilih Tambahkan langkah.
5. Pilih fitur Ekstrak.
6. Untuk fitur Ekstrak untuk kolom ini, pilih kolom.
7. (Opsional) Pilih Ratakan untuk menampilkan fitur ke dalam kolom terpisah.
8. Untuk Strategi, pilih strategi untuk mengekstrak fitur.
9. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
10. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Gunakan Fitur Lagged dari Data Time Series Anda

Untuk banyak kasus penggunaan, cara terbaik untuk memprediksi perilaku future dari time series Anda adalah dengan menggunakan perilaku terbarunya.

Penggunaan paling umum dari fitur lagged adalah sebagai berikut:

- Mengumpulkan beberapa nilai masa lalu. Misalnya, untuk waktu, t + 1, Anda mengumpulkan t, t - 1, t - 2, dan t - 3.
- Mengumpulkan nilai-nilai yang sesuai dengan perilaku musiman dalam data. Misalnya, untuk memprediksi hunian di restoran pada pukul 13:00, Anda mungkin ingin menggunakan fitur mulai pukul 13.00 pada hari sebelumnya. Menggunakan fitur dari 12:00 PM atau 11:00 AM pada hari yang sama mungkin tidak prediktif seperti menggunakan fitur dari hari-hari sebelumnya.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
4. Pilih Tambahkan langkah.
5. Pilih fitur Lag.
6. Untuk Menghasilkan fitur lag untuk kolom ini, pilih kolom.
7. Untuk Timestamp Column, pilih kolom yang berisi stempel waktu.
8. Untuk Lag, tentukan durasi lag.
9. (Opsional) Konfigurasi output menggunakan salah satu opsi berikut:
    - Sertakan seluruh jendela lag
    - Ratakan output

- Jatuhkan baris tanpa riwayat

10. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
11. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Buat Rentang Datetime Dalam Seri Waktu Anda

Anda mungkin memiliki data deret waktu yang tidak memiliki stempel waktu. Jika Anda tahu bahwa pengamatan dilakukan secara berkala, Anda dapat menghasilkan stempel waktu untuk deret waktu di kolom terpisah. Untuk menghasilkan stempel waktu, Anda menentukan nilai untuk stempel waktu awal dan frekuensi stempel waktu.

Misalnya, Anda mungkin memiliki data deret waktu berikut untuk jumlah pelanggan di restoran.

Data deret waktu tentang jumlah pelanggan di restoran

Jumlah tindakan
10
14
24
40
30
20

Jika Anda tahu bahwa restoran dibuka pada pukul 17:00 dan pengamatan dilakukan setiap jam, Anda dapat menambahkan kolom stempel waktu yang sesuai dengan data deret waktu. Anda dapat melihat kolom stempel waktu pada tabel berikut.

Data deret waktu tentang jumlah pelanggan di restoran

Jumlah tindakan	Stempel Waktu
10	1:00PM
14	14:00 SORE

Jumlah tindakan	Stempel Waktu
24	15:00 SORE
40	16:00 SORE
30	17:00 SORE
20	6:00 SORE

Gunakan prosedur berikut untuk menambahkan rentang datetime ke data Anda.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.
4. Pilih Tambahkan langkah.
5. Pilih rentang Datetime.
6. Untuk tipe Frekuensi, pilih unit yang digunakan untuk mengukur frekuensi stempel waktu.
7. Untuk Memulai stempel waktu, tentukan stempel waktu mulai.
8. Untuk kolom Output, tentukan nama untuk kolom output.
9. (Opsional) Konfigurasikan output menggunakan bidang yang tersisa.
10. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
11. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Gunakan Jendela Bergulir Dalam Seri Waktu Anda

Anda dapat mengekstrak fitur selama periode waktu tertentu. Misalnya, untuk waktu, t, dan panjang jendela waktu 3, dan untuk baris yang menunjukkan stempel waktu t th, kami menambahkan fitur yang diekstraksi dari deret waktu pada waktu t - 3, t -2, dan t - 1. Untuk informasi tentang mengekstraksi fitur, lihatEkstrak Fitur dari Data Seri Waktu Anda.

Anda dapat menggunakan prosedur berikut untuk mengekstrak fitur selama periode waktu tertentu.

1. Buka aliran data Wrangler Data Anda.
2. Jika Anda belum mengimpor dataset Anda, impor di bawah tab Impor data.
3. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan transformasi.

4. Pilih Tambahkan langkah.
5. Pilih fitur jendela bergulir.
6. Untuk Menghasilkan fitur jendela bergulir untuk kolom ini, pilih kolom.
7. Untuk Timestamp Column, pilih kolom yang berisi stempel waktu.
8. (Opsional) Untuk Kolom Keluaran, tentukan nama kolom output.
9. Untuk ukuran jendela, tentukan ukuran jendela.
10. Untuk Strategi, pilih strategi ekstraksi.
11. Pilih Pratinjau untuk menghasilkan pratinjau transformasi.
12. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Duplikat

Gunakan Featurize tanggal/waktu untuk membuat embedding vektor yang mewakili bidang datetime. Untuk menggunakan transformasi ini, data datetime Anda harus berada dalam salah satu format berikut:

- String yang menjelaskan datetime: Misalnya,. `"January 1st, 2020, 12:44pm"`
- Stempel waktu Unix: Stempel waktu Unix menggambarkan jumlah detik, milidetik, mikrodetik, atau nanodetik dari 1/1/1970.

Anda dapat memilih untuk Menyimpulkan format datetime dan menyediakan format Datetime. Jika Anda menyediakan format datetime, Anda harus menggunakan kode yang dijelaskan dalam dokumentasi Python. Opsi yang Anda pilih untuk dua konfigurasi ini memiliki implikasi untuk kecepatan operasi dan hasil akhir.

- Opsi paling manual dan tercepat secara komputasi adalah menentukan format Datetime dan pilih No for Infer datetime format.
- Untuk mengurangi tenaga kerja manual, Anda dapat memilih format Infer datetime dan tidak menentukan format datetime. Ini juga merupakan operasi komputasi cepat; Namun, format datetime pertama yang ditemui di kolom input diasumsikan sebagai format untuk seluruh kolom. Jika ada format lain di kolom, nilai-nilai ini adalah NaN di output akhir. Menyimpulkan format datetime dapat memberi Anda string yang tidak diurai.
- Jika Anda tidak menentukan format dan memilih No for Index datetime format, Anda mendapatkan hasil yang paling kuat. Semua string datetime yang valid diuraikan. Namun, operasi ini bisa menjadi urutan besarnya lebih lambat dari dua opsi pertama dalam daftar ini.

Bila Anda menggunakan transformasi ini, Anda menentukan kolom Input yang berisi data datetime dalam salah satu format yang tercantum di atas. Transformasi menciptakan kolom output bernama Output nama kolom. Format kolom output tergantung pada konfigurasi Anda menggunakan yang berikut ini:

- Vektor: Mengeluarkan satu kolom sebagai vektor.
- Kolom: Membuat kolom baru untuk setiap fitur. Misalnya, jika output berisi tahun, bulan, dan hari, tiga kolom terpisah dibuat untuk tahun, bulan, dan hari.

Selain itu, Anda harus memilih mode Embedding. Untuk model linier dan jaringan dalam, kami sarankan memilih siklik. Untuk algoritma berbasis pohon, kami sarankan memilih ordinal.

Format String

Transformasi string Format berisi operasi pemformatan string standar. Misalnya, Anda dapat menggunakan operasi ini untuk menghapus karakter khusus, menormalkan panjang string, dan memperbarui casing string.

Grup fitur ini berisi transformasi berikut. Semua transformasi mengembalikan salinan string di kolom Input dan menambahkan hasilnya ke kolom keluaran baru.

Nama	Fungsi
Pad kiri	Left-pad string dengan karakter Fill yang diberikan ke lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.
Pad kanan	Right-pad string dengan karakter Fill yang diberikan ke lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.
Tengah (pad di kedua sisi)	Tengah-pad string (tambahkan padding di kedua sisi string) dengan karakter Fill yang diberikan ke lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.

Nama	Fungsi
Menutup nol	Isi kiri string numerik dengan nol, hingga lebar yang diberikan. Jika string lebih panjang dari lebar, nilai kembali disingkat menjadi karakter lebar.
Strip kiri dan kanan	Mengembalikan salinan string dengan karakter utama dan trailing dihapus.
Strip karakter dari kiri	Mengembalikan salinan string dengan karakter utama dihapus.
Strip karakter dari kanan	Mengembalikan salinan string dengan karakter trailing dihapus.
Cetak huruf besar	Ubah semua huruf dalam teks menjadi huruf kecil.
Kasus besar	Ubah semua huruf dalam teks menjadi huruf besar.
Kapitalisasi	Kapitalisasi huruf pertama di setiap kalimat.
Mengkonversi ke domain	Mengkonversi semua karakter huruf besar ke huruf kecil dan semua karakter huruf kecil untuk karakter huruf besar dari string yang diberikan, dan mengembalikannya.
Tambahkan awalan atau akhiran	Menambahkan awalan dan akhiran kolom string. Anda harus menentukan setidaknya satu dari Awalan dan Akhiran.
Hapus simbol	Menghapus simbol yang diberikan dari string. Semua karakter yang terdaftar dihapus. Default-nya adalah spasi putih.

Tangani Outlier

Model pembelajaran mesin sensitif terhadap distribusi dan jangkauan nilai fitur Anda. Pencilan, atau nilai langka, dapat berdampak negatif pada akurasi model dan menyebabkan waktu pelatihan yang lebih lama. Gunakan grup fitur ini untuk mendeteksi dan memperbarui outlier dalam kumpulan data Anda.

Saat Anda menentukan langkah transformasi Handle outlier, statistik yang digunakan untuk mendeteksi outlier dihasilkan pada data yang tersedia di Data Wrangler saat mendefinisikan langkah ini. Statistik yang sama ini digunakan saat menjalankan pekerjaan Data Wrangler.

Gunakan bagian berikut untuk mempelajari selengkapnya tentang transformasi yang dikandung grup ini. Anda menentukan nama Output dan masing-masing transformasi ini menghasilkan kolom output dengan data yang dihasilkan.

Outlier numerik deviasi standar yang kuat

Transformasi ini mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan statistik yang kuat untuk outlier.

Anda harus menentukan kuantil Atas dan kuantil Bawah untuk statistik yang digunakan untuk menghitung outlier. Anda juga harus menentukan jumlah Standar deviasi dari mana nilai harus bervariasi dari rata-rata untuk dianggap sebagai outlier. Misalnya, jika Anda menentukan 3 untuk Standar deviasi, nilai harus jatuh lebih dari 3 standar deviasi dari rata-rata untuk dianggap sebagai outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.
- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Pencilan Numerik Deviasi Standar

Transformasi ini mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan mean dan standar deviasi.

Anda menentukan jumlah Standar deviasi suatu nilai harus bervariasi dari rata-rata untuk dianggap sebagai outlier. Misalnya, jika Anda menentukan 3 untuk Standar deviasi, nilai harus jatuh lebih dari 3 standar deviasi dari rata-rata untuk dianggap sebagai outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.
- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Pencilan Numerik Kuantil

Gunakan transformasi ini untuk mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan kuantil. Anda dapat menentukan kuantil Atas dan kuantil Bawah. Semua nilai yang berada di atas kuantil atas atau di bawah kuantil bawah dianggap outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.
- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Pencilan Numerik Min-Max

Transformasi ini mendeteksi dan memperbaiki outlier dalam fitur numerik menggunakan ambang batas atas dan bawah. Gunakan metode ini jika Anda mengetahui nilai ambang batas yang mendemark outlier.

Anda menentukan ambang atas dan ambang bawah, dan jika nilai jatuh di atas atau di bawah ambang tersebut masing-masing, mereka dianggap outlier.

Metode Fix adalah metode yang digunakan untuk menangani outlier ketika terdeteksi. Anda dapat memilih dari opsi berikut:

- Klip: Gunakan opsi ini untuk memotong outlier ke terikat deteksi outlier yang sesuai.
- Hapus: Gunakan opsi ini untuk menghapus baris dengan outlier dari kerangka data.

- Tidak valid: Gunakan opsi ini untuk mengganti outlier dengan nilai yang tidak valid.

Ganti Rare

Saat Anda menggunakan Ganti transformasi langka, Anda menentukan ambang batas dan Data Wrangler menemukan semua nilai yang memenuhi ambang batas tersebut dan menggantinya dengan string yang Anda tentukan. Misalnya, Anda mungkin ingin menggunakan transformasi ini untuk mengkategorikan semua outlier dalam kolom ke dalam kategori “Lainnya”.

- String pengganti: String yang digunakan untuk mengganti outlier.
- Ambang batas absolut: Kategori jarang terjadi jika jumlah instance kurang dari atau sama dengan ambang absolut ini.
- Ambang pecahan: Kategori jarang terjadi jika jumlah instance kurang dari atau sama dengan ambang fraksi ini dikalikan dengan jumlah baris.
- Kategori umum maksimum: Kategori maksimum yang tidak langka yang tersisa setelah operasi. Jika ambang batas tidak menyaring kategori yang cukup, mereka yang memiliki jumlah penampilan teratas diklasifikasikan sebagai tidak jarang. Jika disetel ke 0 (default), tidak ada batasan keras untuk jumlah kategori.

Menangani Nilai yang Hilang

Nilai yang hilang adalah kejadian umum dalam kumpulan data pembelajaran mesin. Dalam beberapa situasi, adalah tepat untuk menghitung data yang hilang dengan nilai yang dihitung, seperti nilai rata-rata atau kategoris umum. Anda dapat memproses nilai yang hilang menggunakan grup transformasi nilai Handle yang hilang. Grup ini berisi transformasi berikut.

Mengidentifikasi

Gunakan Fill missing transform untuk mengganti nilai yang hilang dengan nilai Fill yang Anda tentukan.

Mengidentifikasi

Gunakan transformasi yang hilang Impute untuk membuat kolom baru yang berisi nilai yang diperhitungkan di mana nilai yang hilang ditemukan dalam data kategoris dan numerik input. Konfigurasi tergantung pada tipe data Anda.

Untuk data numerik, pilih strategi imputing, strategi yang digunakan untuk menentukan nilai baru yang akan diperhitungkan. Anda dapat memilih untuk menghitung mean atau median atas nilai yang

ada dalam kumpulan data Anda. Data Wrangler menggunakan nilai yang dihitung untuk menghitung nilai yang hilang.

Untuk data kategoris, Data Wrangler menyiratkan nilai yang hilang menggunakan nilai yang paling sering di kolom. Untuk memasukkan string kustom, gunakan Fill missing transform sebagai gantinya.

Tambahkan Indikator untuk Hilang

Gunakan indikator Tambah untuk transformasi yang hilang untuk membuat kolom indikator baru, yang berisi Boolean `"false"` jika baris berisi nilai, dan `"true"` jika baris berisi nilai yang hilang.

Kirim

Gunakan opsi Drop missing untuk menjatuhkan baris yang berisi nilai yang hilang dari kolom Input.

Kelola Kolom

Anda dapat menggunakan transformasi berikut untuk memperbarui dan mengelola kolom dengan cepat di kumpulan data Anda:

Nama	Fungsi
Jatuhkan Kolom	Hapus kolom
Duplikat	Duplikat
Ubah Nama Kolom	Ubah Nama Kolom
Memindahkan	Pindahkan lokasi kolom dalam kumpulan data. Pilih untuk memindahkan kolom Anda ke awal atau akhir kumpulan data, sebelum atau sesudah kolom referensi, atau ke indeks tertentu.

Mempersiapkan nama

Gunakan grup transformasi ini untuk dengan cepat melakukan operasi pengurutan dan pengocokan pada baris. Grup ini berisi hal berikut ini:

- Urutkan: Urutkan seluruh kerangka data dengan kolom tertentu. Pilih kotak centang di sebelah Urutan naik untuk opsi ini; jika tidak, batalkan centang kotak dan urutan menurun digunakan untuk pengurutan.
- Shuffle: Aduk semua baris dalam kumpulan data secara acak.

Kelola Vektor

Gunakan grup transformasi ini untuk menggabungkan atau meratakan kolom vektor. Grup ini berisi transformasi berikut.

- Merakit: Gunakan transformasi ini untuk menggabungkan vektor Spark dan data numerik menjadi satu kolom. Misalnya, Anda dapat menggabungkan tiga kolom: dua berisi data numerik dan satu berisi vektor. Tambahkan semua kolom yang ingin Anda gabungkan Kolom input dan tentukan nama kolom Output untuk data gabungan.
- Flatten: Gunakan transformasi ini untuk meratakan satu kolom yang berisi data vektor. Kolom input harus berisi PySpark vektor atau objek seperti array. Anda dapat mengontrol jumlah kolom yang dibuat dengan menentukan Metode untuk mendeteksi jumlah output. Misalnya, jika Anda memilih Panjang vektor pertama, jumlah elemen dalam vektor atau larik valid pertama yang ditemukan di kolom menentukan jumlah kolom keluaran yang dibuat. Semua vektor input lainnya dengan terlalu banyak item terpotong. Masukan dengan terlalu sedikit item diisi dengan NaNs.

  Anda juga menentukan awalan Output, yang digunakan sebagai awalan untuk setiap kolom output.

Mempersiapkan Data

Gunakan grup fitur Process Numeric untuk memproses data numerik. Setiap skalar dalam grup ini didefinisikan menggunakan perpustakaan Spark. Skalar berikut didukung:

- Standard Scaler: Standarisasi kolom input dengan mengurangi rata-rata dari setiap nilai dan penskalaan ke varians unit. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk StandardScaler.
- Robust Scaler: Skala kolom input menggunakan statistik yang kuat untuk outlier. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk RobustScaler.
- Min Max Scaler: Ubah kolom input dengan menskalakan setiap fitur ke rentang tertentu. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk MinMaxScaler.

- Max Absolute Scaler: Skala kolom input dengan membagi setiap nilai dengan nilai absolut maksimum. Untuk mempelajari lebih lanjut, lihat dokumentasi Spark untuk MaxAbsScaler.

Pengambilan sampel

Setelah mengimpor data, Anda dapat menggunakan transformator Sampling untuk mengambil satu atau lebih sampelnya. Saat Anda menggunakan transformator sampling, Data Wrangler mengambil sampel kumpulan data asli Anda.

Anda dapat memilih salah satu metode sampel berikut:

- Batas: Sampel kumpulan data mulai dari baris pertama hingga batas yang Anda tentukan.
- Acak: Mengambil sampel acak dari ukuran yang Anda tentukan.
- Bertingkat: Mengambil sampel acak bertingkat.

Anda dapat membuat stratifikasi sampel acak untuk memastikan bahwa sampel tersebut mewakili distribusi asli kumpulan data.

Anda mungkin melakukan persiapan data untuk beberapa kasus penggunaan. Untuk setiap kasus penggunaan, Anda dapat mengambil sampel yang berbeda dan menerapkan serangkaian transformasi yang berbeda.

Prosedur berikut menjelaskan proses pembuatan sampel acak.

Untuk mengambil sampel acak dari data Anda.

1. Pilih + di sebelah kanan kumpulan data yang telah Anda impor. Nama dataset Anda terletak di bawah +.
2. Pilih Tambahkan transformasi.
3. Pilih Pengambilan sampel.
4. Untuk metode Sampling, pilih metode sampling.
5. Untuk Perkiraan ukuran sampel, pilih perkiraan jumlah pengamatan yang Anda inginkan dalam sampel Anda.
6. (Opsional) Tentukan bilangan bulat untuk benih Acak untuk membuat sampel yang dapat direproduksi.

Prosedur berikut menjelaskan proses pembuatan sampel bertingkat.

Untuk mengambil sampel bertingkat dari data Anda.

1. Pilih + di sebelah kanan kumpulan data yang telah Anda impor. Nama dataset Anda terletak di bawah +.
2. Pilih Tambahkan transformasi.
3. Pilih Pengambilan sampel.
4. Untuk metode Sampling, pilih metode sampling.
5. Untuk Perkiraan ukuran sampel, pilih perkiraan jumlah pengamatan yang Anda inginkan dalam sampel Anda.
6. Untuk kolom Stratify, tentukan nama kolom yang ingin Anda stratifikasi.
7. (Opsional) Tentukan bilangan bulat untuk benih Acak untuk membuat sampel yang dapat direproduksi.

Penyempurnaan Data

Gunakan bagian ini untuk mencari dan mengedit pola tertentu dalam string. Misalnya, Anda dapat menemukan dan memperbarui string dalam kalimat atau dokumen, membagi string dengan pembatas, dan menemukan kemunculan string tertentu.

Transformasi berikut didukung di bawah Cari dan edit. Semua transformasi mengembalikan salinan string di kolom Input dan menambahkan hasilnya ke kolom output baru.

Nama	Fungsi
Memeriksa satu spasi	Mengembalikan indeks kejadian pertama dari Substring yang Anda cari, Anda dapat memulai dan mengakhiri pencarian di Mulai dan Akhir masing-masing.
Temukan substring (dari kanan)	Mengembalikan indeks kejadian terakhir dari Substring yang Anda cari. Anda dapat memulai dan mengakhiri pencarian di Mulai dan Akhir masing-masing.
Penyerapan prefiks	Mengembalikan nilai Boolean jika string berisi Pola yang diberikan. Sebuah pola dapat berupa urutan karakter atau ekspresi reguler. Secara

Nama	Fungsi
	opsional, Anda dapat membuat pola peka huruf besar/huruf besar.
Temukan semua kejadian	Mengembalikan array dengan semua kejadian dari pola yang diberikan. Sebuah pola dapat berupa urutan karakter atau ekspresi reguler.
Ekstrak menggunakan regex	Mengembalikan string yang cocok dengan pola Regex tertentu.
Ekstrak antara pembatas	Mengembalikan string dengan semua karakter ditemukan antara pembatas Kiri dan pembatas Kanan.
Ekstrak dari posisi	Mengembalikan string, mulai dari posisi Mulai dalam string input, yang berisi semua karakter hingga posisi awal ditambah Panjang.
Temukan dan mengganti substring	Mengembalikan string dengan semua kecocokan dari Pola tertentu (ekspresi reguler) digantikan oleh string Penggantian.
Ganti antara pembatas	Mengembalikan string dengan substring ditemukan antara penampilan pertama pembatas Kiri dan penampilan terakhir dari pembatas Kanan digantikan oleh string Penggantian. Jika tidak ada kecocokan ditemukan, tidak ada yang diganti.
Ganti dari posisi	Mengembalikan string dengan substring antara posisi Mulai dan posisi Mulai ditambah Panjang diganti dengan string Penggantian. Jika posisi Mulai ditambah Panjang lebih besar dari panjang string pengganti, outputnya berisi....

Nama	Fungsi
Konversi regex menjadi hilang	Mengkonversi string ke None jika tidak valid dan mengembalikan hasilnya. Validitas didefinisikan dengan ekspresi reguler dalam Pola.
Pisahkan string dengan pembatas	Mengembalikan array string dari string input, dibagi dengan Delimiter, dengan sampai jumlah Max split (opsional). Delimiter default ke spasi putih.

Pengenal data per set data

Gunakan transformasi data Split untuk membagi kumpulan data Anda menjadi dua atau tiga kumpulan data. Misalnya, Anda dapat membagi kumpulan data menjadi kumpulan data yang digunakan untuk melatih model dan kumpulan data yang digunakan untuk mengujinya. Anda dapat menentukan proporsi dataset yang masuk ke setiap split. Misalnya, jika Anda membagi satu kumpulan data menjadi dua kumpulan data, kumpulan data pelatihan dapat memiliki 80% data sementara kumpulan data pengujian memiliki 20%.

Memisahkan data Anda menjadi tiga kumpulan data memberi Anda kemampuan untuk membuat kumpulan data pelatihan, validasi, dan pengujian. Anda dapat melihat seberapa baik kinerja model pada kumpulan data pengujian dengan menjatuhkan kolom target.

Kasus penggunaan Anda menentukan berapa banyak kumpulan data asli yang didapat masing-masing kumpulan data Anda dan metode yang Anda gunakan untuk membagi data. Misalnya, Anda mungkin ingin menggunakan pemisahan bertingkat untuk memastikan bahwa distribusi pengamatan di kolom target sama di seluruh kumpulan data. Anda dapat menggunakan transformasi split berikut:

- Pemisahan acak - Setiap pemisahan adalah sampel acak dan tidak tumpang tindih dari kumpulan data asli. Untuk kumpulan data yang lebih besar, menggunakan pemisahan acak mungkin mahal secara komputasi dan membutuhkan waktu lebih lama daripada pemisahan yang dipesan.
- Pemisahan berurutan — Membagi kumpulan data berdasarkan urutan pengamatan yang berurutan. Misalnya, untuk pemisahan uji kereta 80/20, pengamatan pertama yang membentuk 80% dari kumpulan data masuk ke kumpulan data pelatihan. 20% terakhir dari pengamatan pergi

ke dataset pengujian. Pemisahan yang dipesan efektif dalam menjaga urutan data yang ada di antara pemisahan.

- Pemisahan bertingkat — Membagi kumpulan data untuk memastikan bahwa jumlah pengamatan di kolom input memiliki representasi proporsional. Untuk kolom input yang memiliki pengamatan 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, pemisahan 80/20 pada kolom berarti bahwa sekitar 80% dari 1s, 80% dari 2s, dan 80% dari 3s pergi ke set pelatihan. Sekitar 20% dari setiap jenis pengamatan pergi ke set pengujian.
- Split by key — Menghindari data dengan kunci yang sama terjadi di lebih dari satu split. Misalnya, jika Anda memiliki kumpulan data dengan kolom 'customer_id' dan Anda menggunakannya sebagai kunci, tidak ada id pelanggan di lebih dari satu split.

Setelah Anda membagi data, Anda dapat menerapkan transformasi tambahan ke setiap kumpulan data. Untuk sebagian besar kasus penggunaan, mereka tidak diperlukan.

Data Wrangler menghitung proporsi perpecahan untuk kinerja. Anda dapat memilih ambang kesalahan untuk mengatur keakuratan pemisahan. Ambang kesalahan yang lebih rendah lebih akurat mencerminkan proporsi yang Anda tentukan untuk pemisahan. Jika Anda menetapkan ambang kesalahan yang lebih tinggi, Anda mendapatkan kinerja yang lebih baik, tetapi akurasi yang lebih rendah.

Untuk membagi data dengan sempurna, atur ambang kesalahan ke 0. Anda dapat menentukan ambang batas antara 0 dan 1 untuk kinerja yang lebih baik. Jika Anda menentukan nilai yang lebih besar dari 1, Data Wrangler menafsirkan nilai itu sebagai 1.

Jika Anda memiliki 10000 baris dalam kumpulan data Anda dan Anda menentukan pemisahan 80/20 dengan kesalahan 0,001, Anda akan mendapatkan pengamatan yang mendekati salah satu hasil berikut:

- 8010 pengamatan di set pelatihan dan 1990 di set pengujian
- 7990 pengamatan di set pelatihan dan 2010 di set pengujian

Jumlah pengamatan untuk pengujian yang ditetapkan dalam contoh sebelumnya adalah dalam interval antara 8010 dan 7990.

Secara default, Data Wrangler menggunakan seed acak untuk membuat split dapat direproduksi. Anda dapat menentukan nilai yang berbeda untuk benih untuk membuat pemisahan yang dapat direproduksi yang berbeda.

Randomized split

Gunakan prosedur berikut ini untuk melakukan pemisahan acak pada kumpulan data Anda.

Untuk membagi kumpulan data Anda secara acak, lakukan hal berikut

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.
3. Pilih Pisahkan data.
4. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.
5. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua perpecahan. Proporsi harus berjumlah 1.
6. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
7. (Opsional) Tentukan nilai untuk benih acak.
8. Pilih Pratinjau.
9. Pilih Tambahkan.

Ordered split

Gunakan prosedur berikut ini untuk melakukan pemisahan berurutan pada kumpulan data Anda.

Untuk membuat pemisahan berurutan dalam kumpulan data Anda, lakukan hal berikut.

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.
3. Untuk Transform, pilih Ordered split.
4. Pilih Pisahkan data.
5. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.
6. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua perpecahan. Proporsi harus berjumlah 1.
7. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
8. (Opsional) Untuk kolom Input, tentukan kolom dengan nilai numerik. Menggunakan nilai kolom untuk menyimpulkan catatan mana yang ada di setiap split. Nilai yang lebih kecil berada dalam satu split dengan nilai yang lebih besar di split lainnya.

9. (Opsional) Pilih Tangani duplikat untuk menambahkan noise ke nilai duplikat dan buat kumpulan data dengan nilai yang sepenuhnya unik.
10. (Opsional) Tentukan nilai untuk benih acak.
11. Pilih Pratinjau.
12. Pilih Tambahkan.

Stratified split

Gunakan prosedur berikut ini untuk melakukan pemisahan bertingkat pada kumpulan data Anda.

Untuk membuat pemisahan bertingkat dalam kumpulan data Anda, lakukan hal berikut.

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.
3. Pilih Pisahkan data.
4. Untuk Transform, pilih Stratified split.
5. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.
6. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua perpecahan. Proporsi harus berjumlah 1.
7. Untuk kolom Input, tentukan kolom dengan hingga 100 nilai unik. Data Wrangler tidak dapat membuat stratifikasi kolom dengan lebih dari 100 nilai unik.
8. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
9. (Opsional) Tentukan nilai untuk benih acak untuk menentukan benih yang berbeda.
10. Pilih Pratinjau.
11. Pilih Tambahkan.

Split by column keys

Gunakan prosedur berikut untuk membagi dengan kunci kolom dalam dataset Anda.

Untuk membagi dengan kunci kolom dalam kumpulan data Anda, lakukan hal berikut.

1. Pilih + di sebelah node yang berisi kumpulan data yang Anda pisahkan.
2. Pilih Tambahkan transformasi.

3. Pilih Pisahkan data.
4. Untuk Transform, pilih Split by key.
5. (Opsional) Untuk Splits, tentukan nama dan proporsi setiap split. Proporsi harus berjumlah 1.
6. (Opsional) Pilih + untuk membuat split tambahan.
    - Tentukan nama dan proporsi semua perpecahan. Proporsi harus berjumlah 1.
7. Untuk kolom Kunci, tentukan kolom dengan nilai yang tidak ingin Anda tampilkan di kedua kumpulan data.
8. (Opsional) Tentukan nilai untuk ambang kesalahan selain nilai default.
9. Pilih Pratinjau.
10. Pilih Tambahkan.

Parse Nilai sebagai Tipe

Gunakan transformasi ini untuk mentransmisikan kolom ke tipe baru. Tipe data Data Wrangler yang didukung adalah:

- Panjang
- Desimal
- Boolean
- Tanggal, dalam format DD-MM-YYYY, masing-masing mewakili hari, bulan, dan tahun.
- String

Validasi String

Gunakan transformasi string Validasi untuk membuat kolom baru yang menunjukkan bahwa baris data teks memenuhi kondisi tertentu. Misalnya, Anda dapat menggunakan transformasi string Validasi untuk memverifikasi bahwa string hanya berisi karakter huruf kecil. Transformasi berikut didukung di bawah Validasi string.

Transformasi berikut termasuk dalam grup transformasi ini. Jika transformasi menghasilkan nilai Boolean, `True` diwakili dengan a `1` dan `False` diwakili dengan a. `0`

Nama	Fungsi
Pengenal metadata	Mengembalikan `True` jika panjang string sama dengan panjang tertentu. Jika tidak, mengembalikan `False`.
Starts with	Mengembalikan `True` jika string dimulai akan awalan tertentu. Jika tidak, mengembalikan `False`.
Ends with	Mengembalikan `True` jika panjang string sama dengan panjang tertentu. Jika tidak, mengembalikan `False`.
Apakah alfanumerik	Mengembalikan `True` jika string hanya berisi angka dan huruf. Jika tidak, mengembalikan `False`.
Apakah alpha (huruf)	Mengembalikan `True` jika string hanya berisi huruf. Jika tidak, mengembalikan `False`.
Digit dan lebih rendah	Mengembalikan `True` jika string hanya berisi digit. Jika tidak, mengembalikan `False`.
Menghapus satu spasi	Mengembalikan `True` jika string hanya berisi angka dan huruf. Jika tidak, mengembalikan `False`.
Adalah judul	Mengembalikan `True` jika string berisi spasi putih. Jika tidak, mengembalikan `False`.
Pengerapan huruf besar	Mengembalikan `True` jika string hanya berisi huruf kecil. Jika tidak, mengembalikan `False`.
Pengerapan huruf besar	Mengembalikan `True` jika string hanya berisi huruf besar. Jika tidak, mengembalikan `False`.
Adalah numerik	Mengembalikan `True` jika string hanya berisi angka. Jika tidak, mengembalikan `False`.

Nama	Fungsi
Adalah desimal	Mengembalikan `True` jika string hanya berisi angka desimal. Jika tidak, mengembalikan `False`.

Fungsi JSON

Jika Anda memiliki file.csv, Anda mungkin memiliki nilai dalam kumpulan data Anda yang merupakan string JSON. Demikian pula, Anda mungkin memiliki data bersarang di kolom file Parket atau dokumen JSON.

Gunakan operator terstruktur Flatten untuk memisahkan kunci tingkat pertama menjadi kolom terpisah. Kunci tingkat pertama adalah kunci yang tidak bersarang dalam nilai.

Misalnya, Anda mungkin memiliki kumpulan data yang memiliki kolom orang dengan informasi demografis pada setiap orang yang disimpan sebagai string JSON. String JSON mungkin terlihat seperti berikut ini.

```
"{"seq": 1,"name": {"first": "Nathaniel","last": "Ferguson"},"age": 59,"city":
"Posbotno","state": "WV"}"
```

Operator terstruktur Flatten mengonversi kunci tingkat pertama berikut menjadi kolom tambahan dalam kumpulan data Anda:

- seq
- name
- usia
- kota
- status

Data Wrangler menempatkan nilai-nilai kunci sebagai nilai di bawah kolom. Berikut ini menunjukkan nama kolom dan nilai-nilai JSON.

```
seq, name,                                    age, city, state
1, {"first": "Nathaniel","last": "Ferguson"}, 59, Posbotno, WV
```

Untuk setiap nilai dalam kumpulan data Anda yang berisi JSON, operator terstruktur Flatten membuat kolom untuk kunci tingkat pertama. Untuk membuat kolom untuk kunci bersarang, panggil operator lagi. Untuk contoh sebelumnya, memanggil operator membuat kolom:

- name_first
- name_last

Contoh berikut menunjukkan kumpulan data yang dihasilkan dari pemanggilan operasi lagi.

```
seq, name,                                    age, city, state, name_first, name_last
1, {"first": "Nathaniel","last": "Ferguson"}, 59, Posbotno, WV, Nathaniel, Ferguson
```

Pilih Kunci untuk diratakan untuk menentukan kunci tingkat pertama yang ingin diekstrak sebagai kolom terpisah. Jika Anda tidak menentukan kunci apa pun, Data Wrangler mengekstrak semua kunci secara default.

Penyempurnaan Data

Gunakan Explode array untuk memperluas nilai array menjadi baris output terpisah. Misalnya, operasi dapat mengambil setiap nilai dalam array, [[1, 2, 3,], [4, 5, 6], [7, 8, 9]] dan membuat kolom baru dengan baris berikut:

```
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
```

Data Wrangler menamai kolom baru, input_column_name_flatten.

Anda dapat memanggil operasi array Explode beberapa kali untuk mendapatkan nilai bersarang dari array ke dalam kolom output terpisah. Contoh berikut menunjukkan hasil pemanggilan operasi beberapa kali pada dataset dengan array bersarang.

Menempatkan nilai-nilai array bersarang ke dalam kolom terpisah

id	array	id	array_item	id	array_items_items
1	[[kucing, anjingnya], [kelelawar, katak]]	1	[kucing, anjingnya]	1	kucing
2	[[mawar, petunia], [lily, daisy]]	1	[kelelawar, katak]	1	anjingnya
		2	[mawar, petunia]	1	kelelawar
		2	[bunga bakung, daisy]	1	katak
			2	2	mawar
			2	2	petunia
			2	2	bunga bakung
			2	2	bunga aster

Memeriksa metadata

Gunakan Data Wrangler untuk mengimpor dan mengubah gambar yang Anda gunakan untuk pipeline machine learning (ML) Anda. Setelah menyiapkan data gambar Anda, Anda dapat mengekspornya dari alur Data Wrangler Anda ke pipeline MM Anda.

Anda dapat menggunakan informasi yang disediakan di sini untuk membiasakan diri dengan mengimpor dan mengubah data gambar di Data Wrangler. Data Wrangler menggunakan OpenCV

untuk mengimpor gambar. Untuk informasi selengkapnya tentang format gambar yang didukung, lihat Membaca dan menulis file gambar.

Setelah Anda membiasakan diri dengan konsep mengubah data gambar Anda, ikuti tutorial berikut, Siapkan data gambar dengan Amazon SageMaker Data Wrangler.

Industri dan kasus penggunaan berikut adalah contoh di mana menerapkan pembelajaran mesin ke data gambar yang diubah dapat berguna:

- Manufaktur - Mengidentifikasi cacat pada item dari jalur perakitan
- Makanan — Mengidentifikasi makanan busuk atau busuk
- Kedokteran — Mengidentifikasi lesi pada jaringan

Saat Anda bekerja dengan data gambar di Data Wrangler, Anda melalui proses berikut:

1. Impor — Pilih gambar dengan memilih direktori yang berisi mereka di bucket Amazon S3.
2. Transform - Gunakan transformasi bawaan untuk menyiapkan gambar untuk pipeline pembelajaran mesin Anda.
3. Ekspor - Ekspor gambar yang telah Anda ubah ke lokasi yang dapat diakses dari pipeline.

Gunakan prosedur berikut untuk mengimpor data gambar Anda.

Untuk mengimpor data gambar Anda

1. Arahkan ke halaman Buat koneksi.
2. Pilih Amazon S3.
3. Tentukan jalur file Amazon S3 yang berisi data gambar.
4. Untuk jenis File, pilih Gambar.
5. (Opsional) Pilih Impor direktori bersarang untuk mengimpor gambar dari beberapa jalur Amazon S3.
6. Pilih Impor.

Data Wrangler menggunakan pustaka imgaug sumber terbuka untuk transformasi gambar bawaannya. Anda dapat menggunakan transformasi bawaan berikut:

- ResizeImage

- EnhanceImage
- CorruptImage
- SplitImage
- DropCorruptedImages
- DropImageDuplicates
- Kecerahan
- ColorChannels
- Skala Abu-abu
- Putar

Gunakan prosedur berikut ini untuk mengubah gambar Anda tanpa menulis kode.

Untuk mengubah data gambar tanpa menulis kode

1. Dari alur Data Wrangler Anda, pilih + di sebelah node yang mewakili gambar yang telah Anda impor.
2. Pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih transformasi dan konfigurasikan.
5. Pilih Pratinjau.
6. Pilih Tambahkan.

Selain menggunakan transformasi yang disediakan Data Wrangler, Anda juga dapat menggunakan cuplikan kode kustom Anda sendiri. Untuk informasi selengkapnya tentang menggunakan cuplikan kode kustom lihat. Transformasi Anda dapat mengimpor pustaka OpenCV dan imgaug dalam cuplikan kode Anda dan menggunakan transformasi yang terkait dengannya. Berikut ini adalah contoh cuplikan kode yang mendeteksi tepi dalam gambar.

```
# A table with your image data is stored in the `df` variable
import cv2
import numpy as np
from pyspark.sql.functions import column
```

```
from sagemaker_dataprep.compute.operators.transforms.image.constants import
 DEFAULT_IMAGE_COLUMN, IMAGE_COLUMN_TYPE
from sagemaker_dataprep.compute.operators.transforms.image.decorators import
 BasicImageOperationDecorator, PandasUDFOperationDecorator


@BasicImageOperationDecorator
def my_transform(image: np.ndarray) -> np.ndarray:
  # To use the code snippet on your image data, modify the following lines within the
 function
    HYST_THRLD_1, HYST_THRLD_2 = 100, 200
    edges = cv2.Canny(image,HYST_THRLD_1,HYST_THRLD_2)
    return edges


@PandasUDFOperationDecorator(IMAGE_COLUMN_TYPE)
def custom_image_udf(image_row):
    return my_transform(image_row)


df = df.withColumn(DEFAULT_IMAGE_COLUMN,
 custom_image_udf(column(DEFAULT_IMAGE_COLUMN)))
```

Saat menerapkan transformasi dalam alur Data Wrangler Anda, Data Wrangler hanya menerapkannya pada sampel gambar dalam kumpulan data Anda. Untuk mengoptimalkan pengalaman Anda dengan aplikasi, Data Wrangler tidak menerapkan transformasi ke semua gambar Anda.

Untuk menerapkan transformasi ke semua gambar Anda, ekspor alur Data Wrangler Anda ke lokasi Amazon S3. Anda dapat menggunakan gambar yang telah Anda ekspor dalam jalur pelatihan atau inferensi Anda. Gunakan node tujuan atau Notebook Jupyter untuk mengekspor data Anda. Anda dapat mengakses salah satu metode untuk mengekspor data Anda dari aliran Data Wrangler. Untuk informasi tentang cara menggunakan metode ini, lihatEkspor ke Amazon S3.

Penyempurnaan Data

Gunakan Data Wrangler untuk memfilter data di kolom Anda. Saat Anda memfilter data dalam kolom, Anda menentukan bidang berikut:

- Nama kolom — Nama kolom yang Anda gunakan untuk memfilter data.
- Kondisi - Jenis filter yang Anda terapkan pada nilai di kolom.

- Nilai - Nilai atau kategori di kolom tempat Anda menerapkan filter.

Anda dapat memfilter pada kondisi berikut:

- = — Mengembalikan nilai yang cocok dengan nilai atau kategori yang Anda tentukan.
- ! = — Mengembalikan nilai yang tidak cocok dengan nilai atau kategori yang Anda tentukan.
- >= — Untuk data Long atau Float, filter untuk nilai yang lebih besar dari atau sama dengan nilai yang Anda tentukan.
- <= — Untuk data Long atau Float, filter untuk nilai yang kurang dari atau sama dengan nilai yang Anda tentukan.
- > — Untuk data Long atau Float, filter untuk nilai yang lebih besar dari nilai yang Anda tentukan.
- < — Untuk data Long atau Float, filter untuk nilai yang kurang dari nilai yang Anda tentukan.

Untuk kolom yang memiliki kategori, `male` dan`female`, Anda dapat memfilter semua `male` nilai. Anda juga dapat memfilter untuk semua `female` nilai. Karena hanya ada `male` dan `female` nilai di kolom, filter mengembalikan kolom yang hanya memiliki `female` nilai.

Anda juga dapat menambahkan beberapa filter. Filter dapat diterapkan di beberapa kolom atau kolom yang sama. Misalnya, jika Anda membuat kolom yang hanya memiliki nilai dalam rentang tertentu, Anda menambahkan dua filter berbeda. Satu filter menentukan bahwa kolom harus memiliki nilai yang lebih besar dari nilai yang Anda berikan. Filter lain menentukan bahwa kolom harus memiliki nilai kurang dari nilai yang Anda berikan.

Gunakan prosedur berikut untuk menambahkan transformasi filter ke data Anda.

Untuk memfilter data Anda

1. Dari alur Data Wrangler Anda, pilih + di sebelah node dengan data yang Anda filter.
2. Pilih Tambahkan transformasi.
3. Pilih Tambahkan langkah.
4. Pilih Filter data.
5. Tentukan bidang berikut:
    - Nama kolom - Kolom yang Anda filter.
    - Kondisi — Kondisi filter.
    - Nilai - Nilai atau kategori di kolom tempat Anda menerapkan filter.

6. (Opsional) Pilih + mengikuti filter yang telah Anda buat.
7. Konfigurasikan filter.
8. Pilih Pratinjau.
9. Pilih Tambahkan.

Kolom Peta untuk Amazon Personalisasi

Data Wrangler terintegrasi dengan Amazon Personalize, layanan pembelajaran mesin terkelola penuh yang menghasilkan rekomendasi item dan segmen pengguna. Anda dapat menggunakan kolom Peta untuk transformasi Amazon Personalize untuk memasukkan data Anda ke dalam format yang dapat ditafsirkan oleh Amazon Personalize. Untuk informasi selengkapnya tentang transformasi khusus untuk Amazon Personalize, lihat Mengimpor data menggunakan Amazon SageMaker Data Wrangler. Untuk informasi selengkapnya tentang Amazon Personalize, lihat Apa yang Dimaksudkannya dengan Amazon Personalize?

Analisis dan Memvisualisasikan

Amazon SageMaker Data Wrangler menyertakan analisis bawaan yang membantu Anda menghasilkan visualisasi dan analisis data dalam beberapa klik. Anda juga dapat membuat analisis kustom menggunakan kode Anda sendiri.

Anda menambahkan analisis ke kerangka data dengan memilih langkah dalam aliran data Anda, lalu memilih Tambah analisis. Untuk mengakses analisis yang telah Anda buat, pilih langkah yang berisi analisis, dan pilih analisisnya.

Semua analisis dihasilkan menggunakan 100.000 baris dataset Anda.

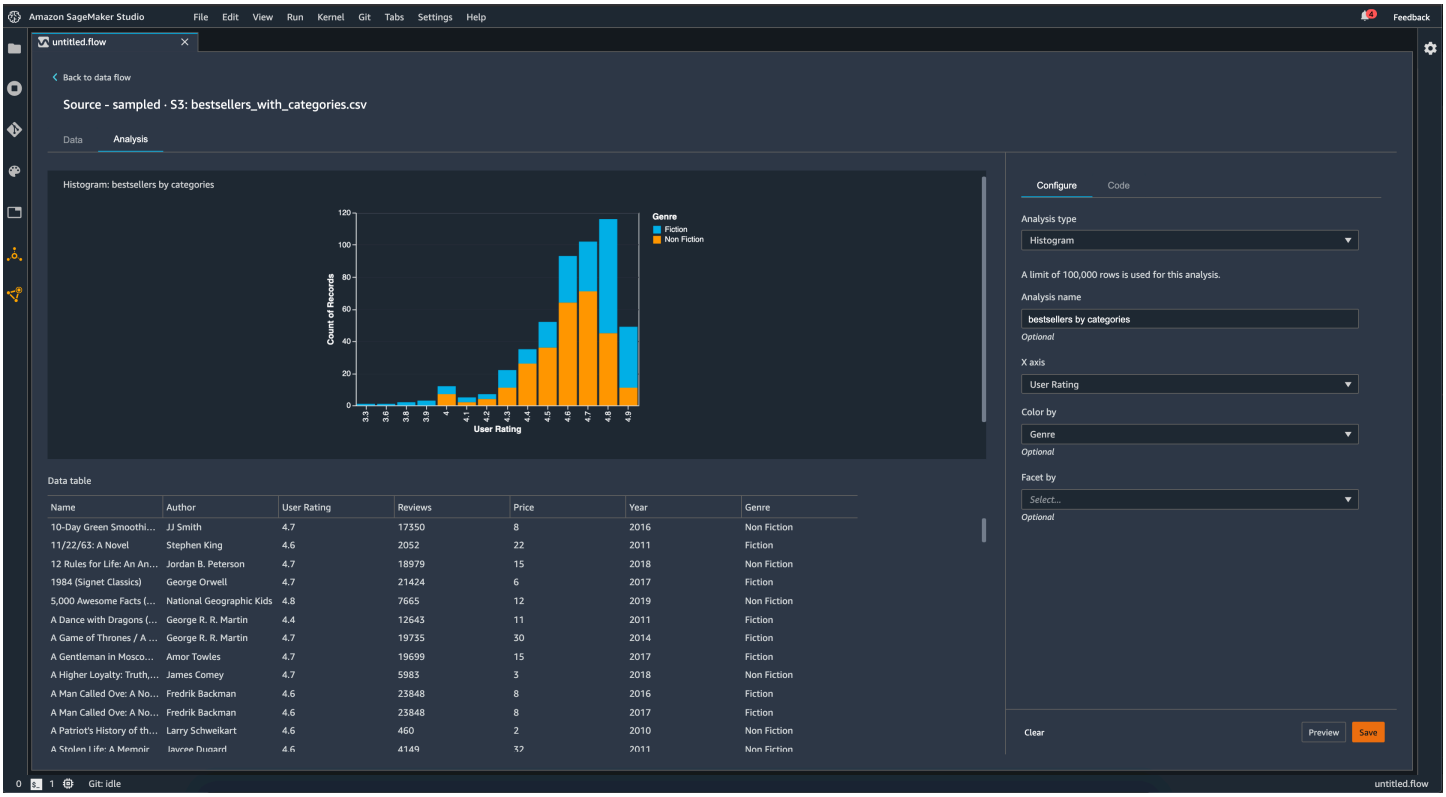
Anda dapat menambahkan analisis berikut ke kerangka data:

- Visualisasi data, termasuk histogram dan plot pencar.
- Ringkasan singkat kumpulan data Anda, termasuk jumlah entri, nilai minimum dan maksimum (untuk data numerik), dan kategori yang paling sering dan paling jarang (untuk data kategoris).
- Model cepat kumpulan data, yang dapat digunakan untuk menghasilkan skor penting untuk setiap fitur.
- Laporan kebocoran target, yang dapat Anda gunakan untuk menentukan apakah satu atau lebih fitur berkorelasi kuat dengan fitur target Anda.
- Visualisasi khusus menggunakan kode Anda sendiri.

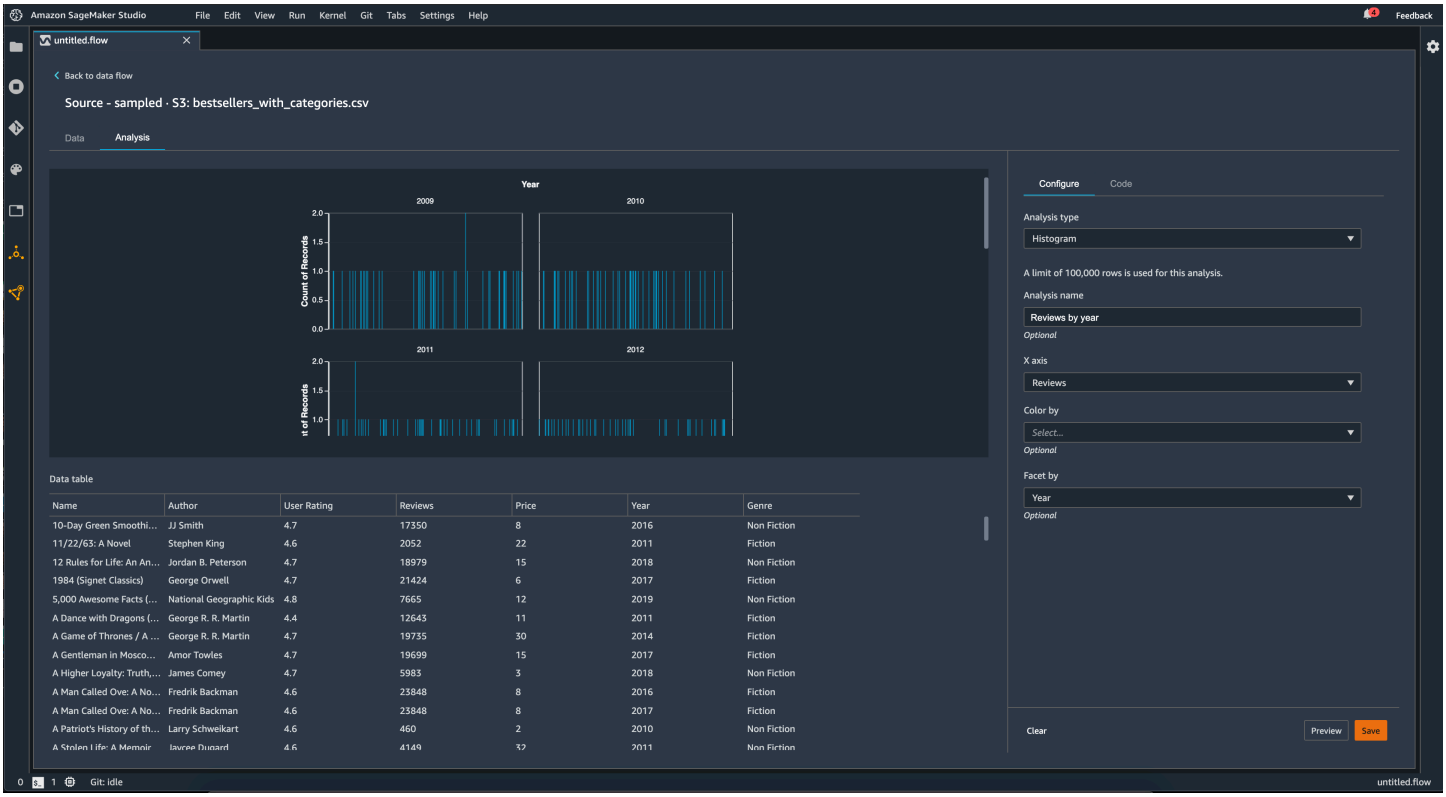
Gunakan bagian berikut untuk mempelajari lebih lanjut tentang opsi ini.

Histogram

Gunakan histogram untuk melihat jumlah nilai fitur untuk fitur tertentu. Anda dapat memeriksa hubungan antar fitur menggunakan opsi Color by. Misalnya, histogram berikut memetakan distribusi peringkat pengguna buku terlaris di Amazon dari 2009-2019, diwarnai berdasarkan genre.



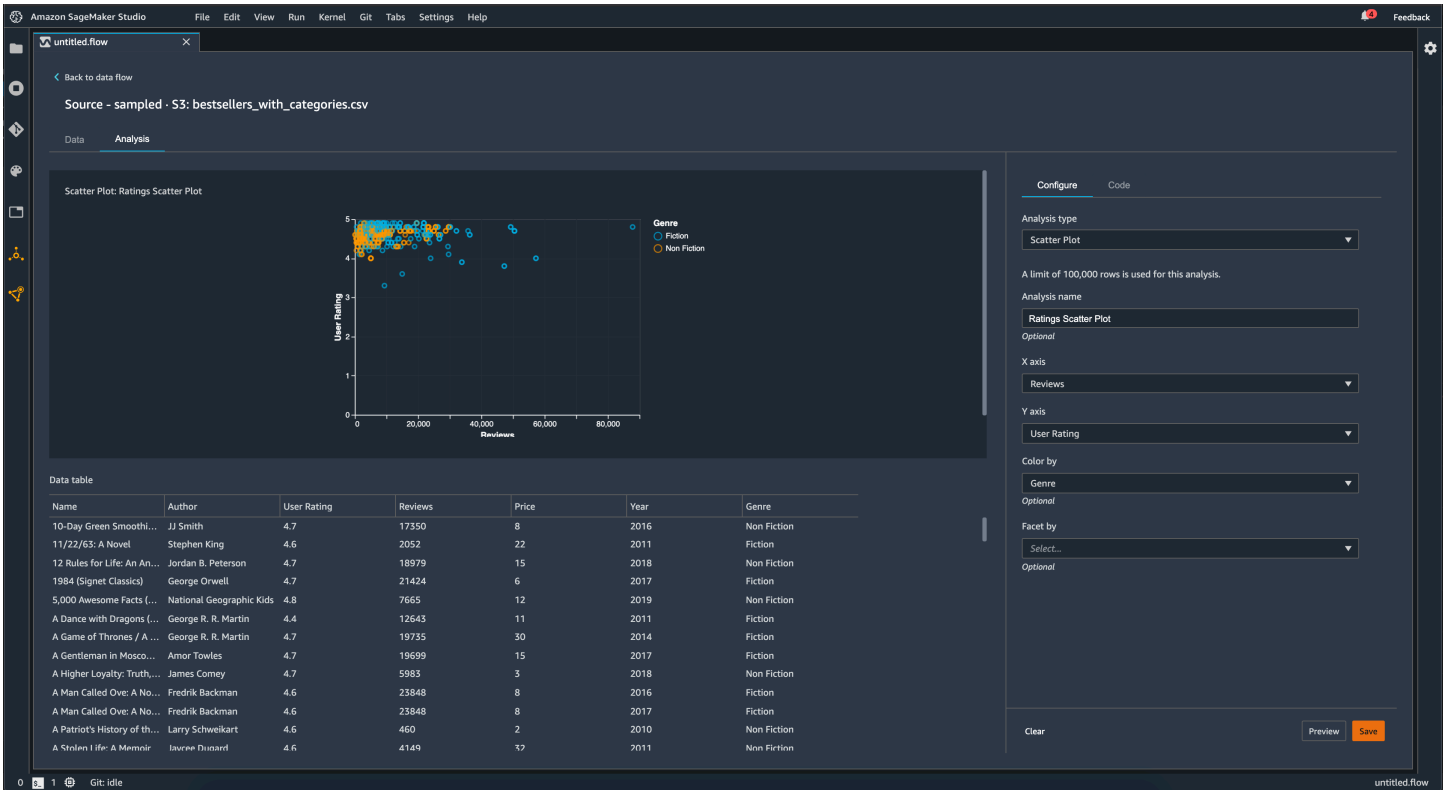
Anda dapat menggunakan fitur Facet by untuk membuat histogram dari satu kolom, untuk setiap nilai di kolom lain. Misalnya, diagram berikut menunjukkan histogram ulasan pengguna buku terlaris di Amazon jika dilihat berdasarkan tahun.



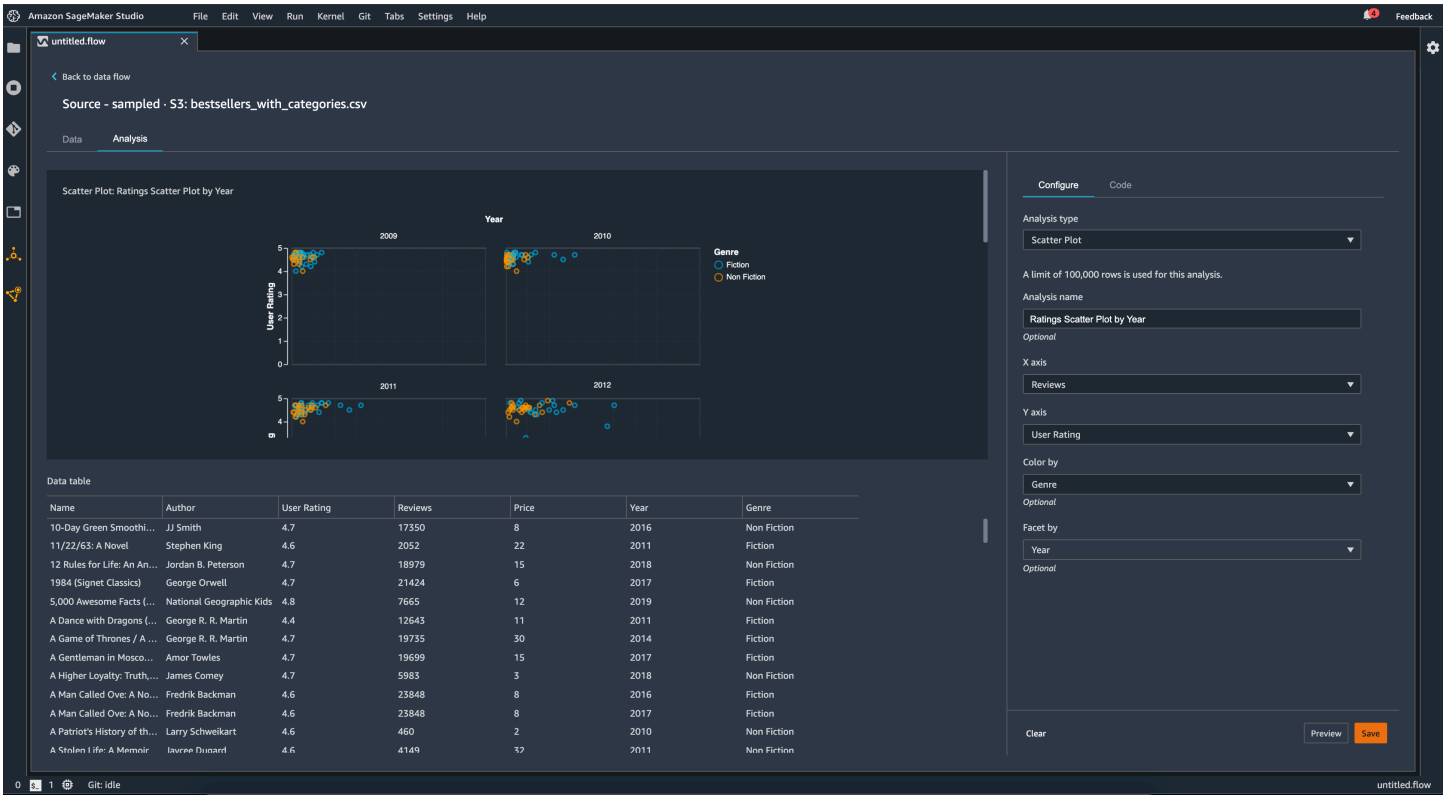
Plot Pencar

Gunakan fitur Scatter Plot untuk memeriksa hubungan antar fitur. Untuk membuat plot pencar, pilih fitur untuk diplot pada sumbu X dan sumbu Y. Kedua kolom ini harus berupa kolom yang diketik numerik.

Anda dapat mewarnai plot pencar dengan kolom tambahan. Misalnya, contoh berikut menunjukkan plot pencar yang membandingkan jumlah ulasan terhadap peringkat pengguna buku terlaris di Amazon antara 2009 dan 2019. Plot pencar diwarnai oleh genre buku.



Selain itu, Anda dapat membagi plot pencar berdasarkan fitur. Misalnya, gambar berikut menunjukkan contoh plot pencar ulasan yang sama versus peringkat pengguna, berdasarkan tahun.



Ringkasan Tabel

Gunakan analisis Ringkasan Tabel untuk meringkas data Anda dengan cepat.

Untuk kolom dengan data numerik, termasuk data log dan float, ringkasan tabel melaporkan jumlah entri (hitungan), minimum (min), maksimum (maks), rata-rata, dan standar deviasi (stddev) untuk setiap kolom.

Untuk kolom dengan data non-numerik, termasuk kolom dengan string, Boolean, atau data tanggal/waktu, ringkasan tabel melaporkan jumlah entri (hitungan), nilai paling sering (min), dan nilai paling sering (maks).

Fungsi SQL baru

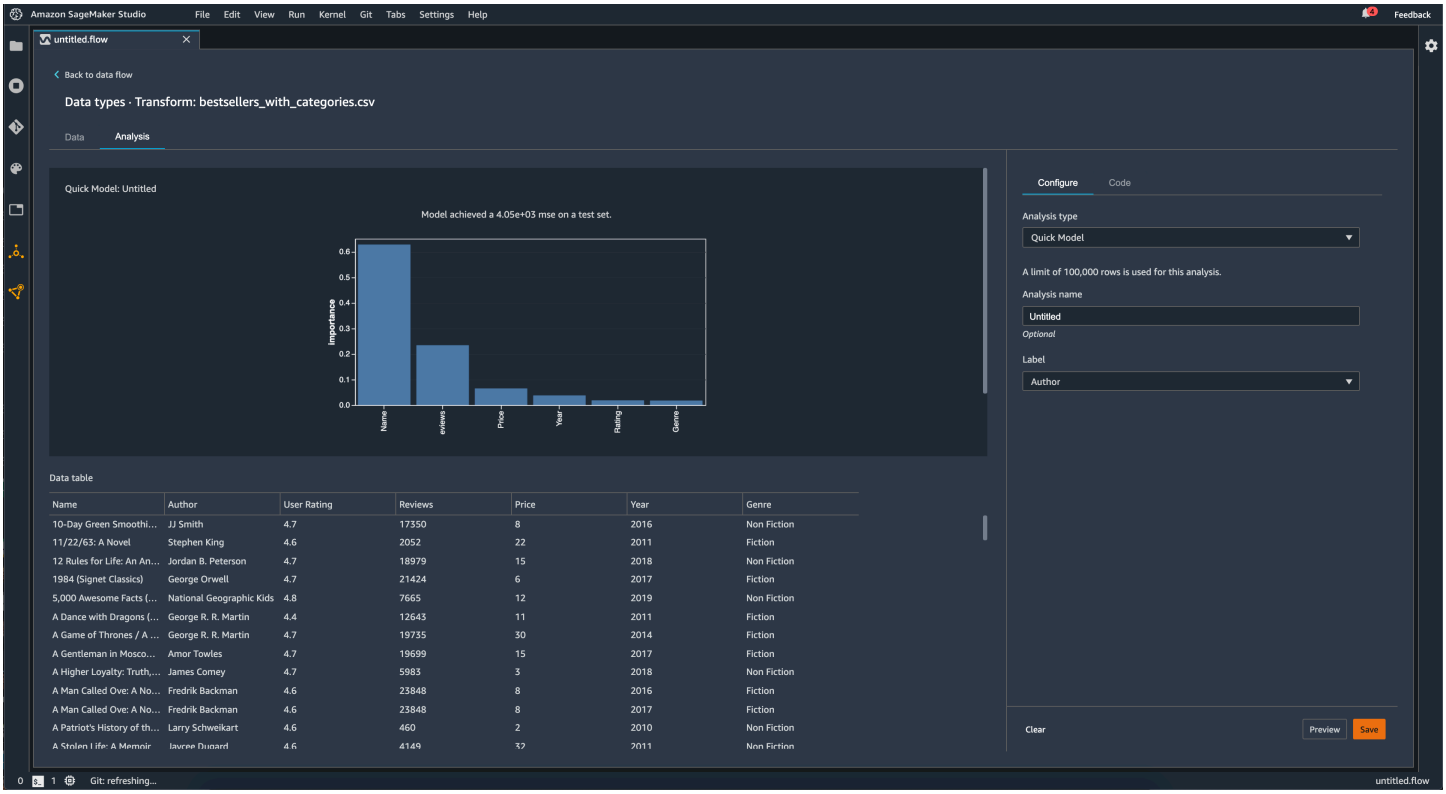
Gunakan visualisasi Model Cepat untuk mengevaluasi data Anda dengan cepat dan menghasilkan skor penting untuk setiap fitur. Skor nilai kepentingan fitur menunjukkan seberapa berguna fitur dalam memprediksi label target. Skor kepentingan fitur adalah antara [0, 1] dan angka yang lebih tinggi menunjukkan bahwa fitur tersebut lebih penting untuk seluruh kumpulan data. Di bagian atas bagan model cepat, ada skor model. Masalah klasifikasi menunjukkan skor F1. Masalah regresi memiliki skor mean squared error (MSE).

Saat Anda membuat bagan model cepat, Anda memilih kumpulan data yang ingin dievaluasi, dan label target yang ingin Anda bandingkan dengan kepentingan fitur. Data Wrangler melakukan hal berikut:

- Menyimpulkan tipe data untuk label target dan setiap fitur dalam kumpulan data yang dipilih.
- Menentukan jenis masalah. Berdasarkan jumlah nilai yang berbeda di kolom label, Data Wrangler menentukan apakah ini adalah jenis masalah regresi atau klasifikasi. Data Wrangler menetapkan ambang kategoris ke 100. Jika ada lebih dari 100 nilai yang berbeda di kolom label, Data Wrangler mengklasifikasikannya sebagai masalah regresi; jika tidak, itu diklasifikasikan sebagai masalah klasifikasi.
- Fitur pra-proses dan data label untuk pelatihan. Algoritma yang digunakan membutuhkan fitur pengkodean untuk jenis vektor dan label pengkodean untuk tipe ganda.
- Melatih algoritma hutan acak dengan 70% data. Spark RandomForestRegressordigunakan untuk melatih model untuk masalah regresi. RandomForestClassifierIni digunakan untuk melatih model untuk masalah klasifikasi.
- Mengevaluasi model hutan acak dengan sisa 30% data. Data Wrangler mengevaluasi model klasifikasi menggunakan skor F1 dan mengevaluasi model regresi menggunakan skor MSE.

- Menghitung pentingnya fitur untuk setiap fitur menggunakan metode kepentingan Gini.

Gambar berikut menampilkan antarmuka pengguna untuk fitur model cepat.



Pengenal pasti

Kebocoran target terjadi ketika ada data dalam kumpulan data pelatihan pembelajaran mesin yang sangat berkorelasi dengan label target, tetapi tidak tersedia dalam data dunia nyata. Misalnya, Anda mungkin memiliki kolom dalam kumpulan data yang berfungsi sebagai proxy untuk kolom yang ingin Anda prediksi dengan model Anda.

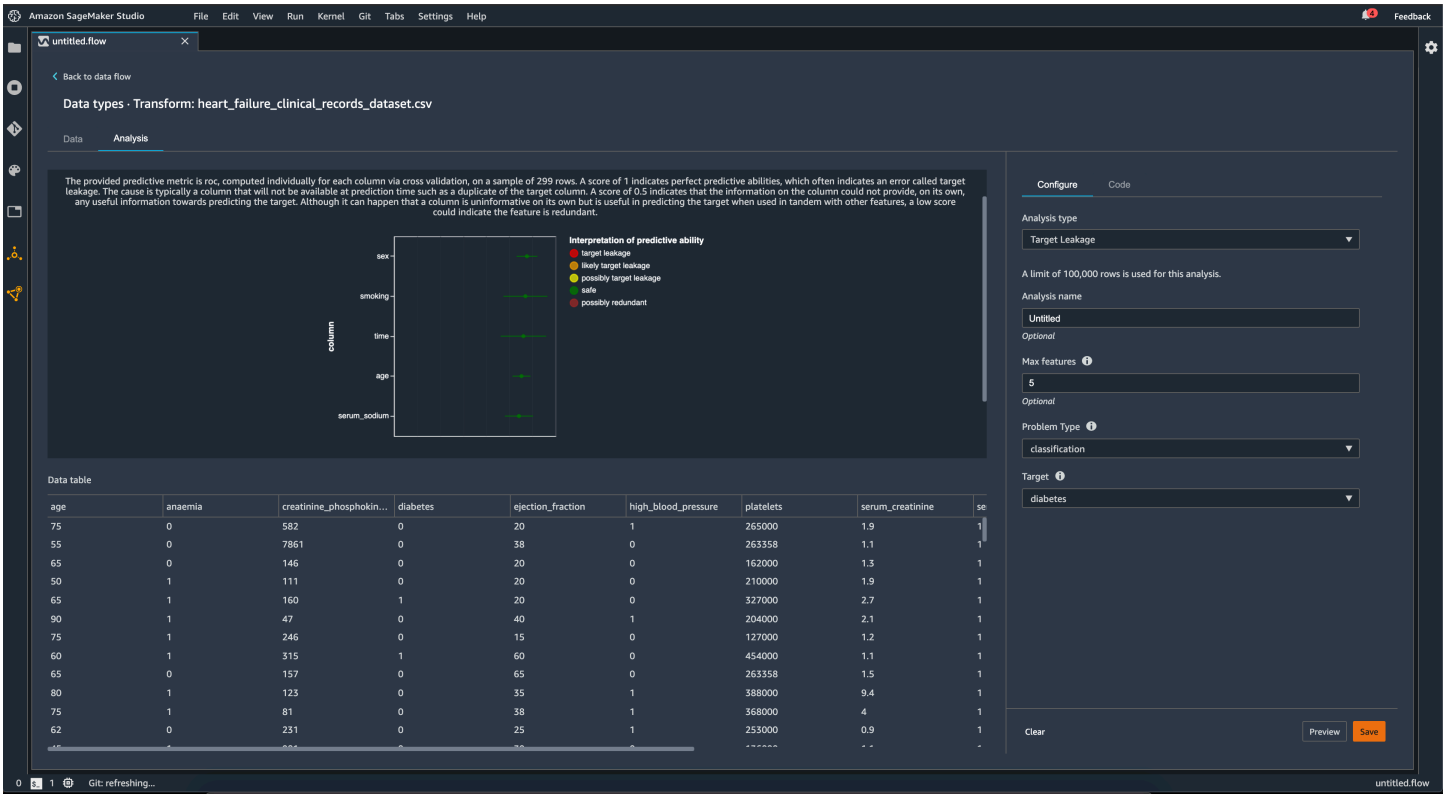
Saat Anda menggunakan analisis Kebocoran Target, Anda menentukan yang berikut:

- Target: Ini adalah fitur yang Anda inginkan agar model ML Anda dapat membuat prediksi.
- Jenis masalah: Ini adalah jenis masalah ML tempat Anda bekerja. Jenis masalah dapat berupa klasifikasi atau regresi.
- (Opsional) Fitur maks: Ini adalah jumlah maksimum fitur untuk hadir dalam visualisasi, yang menunjukkan fitur yang diberi peringkat berdasarkan risiko kebocoran target.

Untuk klasifikasi, analisis kebocoran target menggunakan area di bawah karakteristik operasi penerima, atau kurva AUC - ROC untuk setiap kolom, hingga fitur Max. Untuk regresi, ia menggunakan koefisien determinasi, atau metrik R2.

Kurva AUC - ROC menyediakan metrik prediktif, dihitung secara individual untuk setiap kolom menggunakan validasi silang, pada sampel hingga sekitar 1000 baris. Skor 1 menunjukkan kemampuan prediksi sempurna, yang sering menunjukkan kebocoran target. Skor 0,5 atau lebih rendah menunjukkan bahwa informasi pada kolom tidak dapat memberikan, dengan sendirinya, informasi yang berguna untuk memprediksi target. Meskipun dapat terjadi bahwa kolom tidak informatif dengan sendirinya tetapi berguna dalam memprediksi target ketika digunakan bersama-sama dengan fitur lain, skor rendah dapat menunjukkan fitur tersebut berlebihan.

Misalnya, gambar berikut menunjukkan laporan kebocoran target untuk masalah klasifikasi diabetes, yaitu memprediksi apakah seseorang menderita diabetes atau tidak. Kurva AUC - ROC digunakan untuk menghitung kemampuan prediksi dari lima fitur, dan semuanya ditentukan untuk aman dari kebocoran target.



Multikolinieritas

Multikolinearitas adalah keadaan di mana dua atau lebih variabel prediktor terkait satu sama lain. Variabel prediktor adalah fitur dalam kumpulan data Anda yang Anda gunakan untuk memprediksi

variabel target. Ketika Anda memiliki multikolinieritas, variabel prediktor tidak hanya memprediksi variabel target, tetapi juga prediktif satu sama lain.

Anda dapat menggunakan Variance Inflation Factor (VIF), Principal Component Analysis (PCA), atau pemilihan fitur Lasso sebagai ukuran multikolinearitas dalam data Anda. Untuk informasi selengkapnya, lihat hal berikut.

Variance Inflation Factor (VIF)

Faktor Inflasi Varians (VIF) adalah ukuran kolinearitas di antara pasangan variabel. Data Wrangler mengembalikan skor VIF sebagai ukuran seberapa dekat variabel terkait satu sama lain. Nilai VIF adalah angka positif yang lebih besar dari atau sama dengan 1.

Skor 1 berarti bahwa variabel tidak berkorelasi dengan variabel lainnya. Skor lebih besar dari 1 menunjukkan korelasi yang lebih tinggi.

Secara teoritis, Anda dapat memiliki skor VIF dengan nilai tak terhingga. Data Wrangler klip skor tinggi menjadi 50. Jika Anda memiliki skor VIF lebih besar dari 50, Data Wrangler menetapkan skor menjadi 50.

Anda dapat menggunakan pedoman berikut untuk menafsirkan skor VIF Anda:

- Skor VIF kurang dari atau sama dengan 5 menunjukkan bahwa variabel cukup berkorelasi dengan variabel lainnya.
- Skor VIF lebih besar dari atau sama dengan 5 menunjukkan bahwa variabel sangat berkorelasi dengan variabel lainnya.

Principle Component Analysis (PCA)

Principal Component Analysis (PCA) mengukur varians data di sepanjang arah yang berbeda di ruang fitur. Ruang fitur terdiri dari semua variabel prediktor yang Anda gunakan untuk memprediksi variabel target dalam kumpulan data Anda.

Misalnya, jika Anda mencoba memprediksi siapa yang selamat di RMS Titanic setelah menabrak gunung es, ruang fitur Anda dapat mencakup usia penumpang, jenis kelamin, dan tarif yang mereka bayar.

Dari ruang fitur, PCA menghasilkan daftar varians yang diurutkan. Varians ini juga dikenal sebagai nilai tunggal. Nilai dalam daftar varians lebih besar dari atau sama dengan 0. Kita dapat menggunakannya untuk menentukan berapa banyak multikolinearitas yang ada dalam data kita.

Ketika angka-angkanya kira-kira seragam, data memiliki sangat sedikit contoh multikolinieritas. Ketika ada banyak variabilitas di antara nilai-nilai, kami memiliki banyak contoh multikolinieritas. Sebelum melakukan PCA, Data Wrangler menormalkan setiap fitur untuk memiliki rata-rata 0 dan standar deviasi 1.

> **Note**
>
> PCA dalam keadaan ini juga dapat disebut sebagai Singular Value Decomposition (SVD).

Lasso feature selection

Pemilihan fitur laso menggunakan teknik regularisasi L1 untuk hanya menyertakan fitur paling prediktif dalam kumpulan data Anda.

Untuk klasifikasi dan regresi, teknik regularisasi menghasilkan koefisien untuk setiap fitur. Nilai absolut koefisien memberikan skor penting untuk fitur tersebut. Skor kepentingan yang lebih tinggi menunjukkan bahwa itu lebih prediktif dari variabel target. Metode pemilihan fitur yang umum adalah dengan menggunakan semua fitur yang memiliki koefisien laso bukan nol.

Deteksi Anomali Dalam Data Seri Waktu

Anda dapat menggunakan visualisasi deteksi anomali untuk melihat outlier dalam data deret waktu Anda. Untuk memahami apa yang menentukan anomali, Anda perlu memahami bahwa kami menguraikan deret waktu menjadi istilah yang diprediksi dan istilah kesalahan. Kami memperlakukan musiman dan tren deret waktu sebagai istilah yang diprediksi. Kami memperlakukan residu sebagai istilah kesalahan.

Untuk istilah kesalahan, Anda menentukan ambang batas sebagai jumlah standar deviasi, residu dapat jauh dari rata-rata agar dianggap sebagai anomali. Misalnya, Anda dapat menentukan ambang batas sebagai 3 standar deviasi. Setiap residu yang lebih besar dari 3 standar deviasi dari mean adalah anomali.

Anda dapat menggunakan prosedur berikut untuk melakukan analisis deteksi anomali.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan analisis.
3. Untuk jenis Analisis, pilih Time Series.
4. Untuk Visualisasi, pilih Deteksi anomali.

5. Untuk ambang anomali, pilih ambang batas bahwa nilai dianggap anomali.
6. Pilih Pratinjau untuk menghasilkan pratinjau analisis.
7. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Dekomposisi Tren Musiman Dalam Data Deret Waktu

Anda dapat menentukan apakah ada musiman dalam data deret waktu Anda dengan menggunakan visualisasi Seasonal Trend Decomposition. Kami menggunakan metode STL (Seasonal Trend decomposition using LOESS) untuk melakukan dekomposisi. Kami menguraikan deret waktu menjadi komponen musiman, tren, dan sisa. Tren ini mencerminkan perkembangan jangka panjang dari seri ini. Komponen musiman adalah sinyal yang berulang dalam periode waktu. Setelah menghapus tren dan komponen musiman dari deret waktu, Anda memiliki residu.

Anda dapat menggunakan prosedur berikut untuk melakukan analisis dekomposisi Seasonal-Trend.

1. Buka aliran data Wrangler Data Anda.
2. Dalam aliran data Anda, di bawah Tipe data, pilih +, dan pilih Tambahkan analisis.
3. Untuk jenis Analisis, pilih Time Series.
4. Untuk Visualisasi, pilih dekomposisi Seasonal-Trend.
5. Untuk ambang anomali, pilih ambang batas bahwa nilai dianggap anomali.
6. Pilih Pratinjau untuk menghasilkan pratinjau analisis.
7. Pilih Tambah untuk menambahkan transformasi ke aliran data Data Wrangler.

Fungsi SQL baru

Anda dapat menggunakan laporan bias di Data Wrangler untuk mengungkap potensi bias dalam data Anda. Untuk menghasilkan laporan bias, Anda harus menentukan kolom target, atau Label, yang ingin Anda prediksi dan Facet, atau kolom yang ingin Anda periksa untuk bias.

Label: Fitur yang Anda inginkan model untuk membuat prediksi. Misalnya, jika Anda memprediksi konversi pelanggan, Anda dapat memilih kolom yang berisi data tentang apakah pelanggan telah melakukan pemesanan atau tidak. Anda juga harus menentukan apakah fitur ini adalah label atau ambang batas. Jika Anda menentukan label, Anda harus menentukan seperti apa hasil positif dalam data Anda. Dalam contoh konversi pelanggan, hasil positif mungkin 1 di kolom pesanan, mewakili hasil positif dari pelanggan yang melakukan pemesanan dalam tiga bulan terakhir. Jika Anda menentukan ambang batas, Anda harus menentukan batas bawah yang menentukan hasil

positif. Misalnya, jika kolom pesanan pelanggan Anda berisi jumlah pesanan yang ditempatkan pada tahun lalu, Anda mungkin ingin menentukan 1.

Facet: Kolom yang ingin Anda periksa untuk bias. Misalnya, jika Anda mencoba memprediksi konversi pelanggan, aspek Anda mungkin adalah usia pelanggan. Anda dapat memilih aspek ini karena Anda percaya bahwa data Anda bias terhadap kelompok usia tertentu. Anda harus mengidentifikasi apakah faset diukur sebagai nilai atau ambang batas. Misalnya, jika Anda ingin memeriksa satu atau lebih usia tertentu, Anda memilih Nilai dan menentukan usia tersebut. Jika Anda ingin melihat kelompok usia, pilih Ambang batas dan tentukan ambang usia yang ingin Anda periksa.

Setelah Anda memilih fitur dan label, Anda memilih jenis metrik bias yang ingin Anda hitung.

Untuk mempelajari lebih lanjut, lihat Menghasilkan laporan untuk bias dalam data pra-pelatihan.

Buat Visualisasi Kustom

Anda dapat menambahkan analisis ke alur Data Wrangler Anda untuk membuat visualisasi kustom. Dataset Anda, dengan semua transformasi yang Anda terapkan, tersedia sebagai Panda. DataFrame Data Wrangler menggunakan `df` variabel untuk menyimpan kerangka data. Anda mengakses kerangka data dengan memanggil variabel.

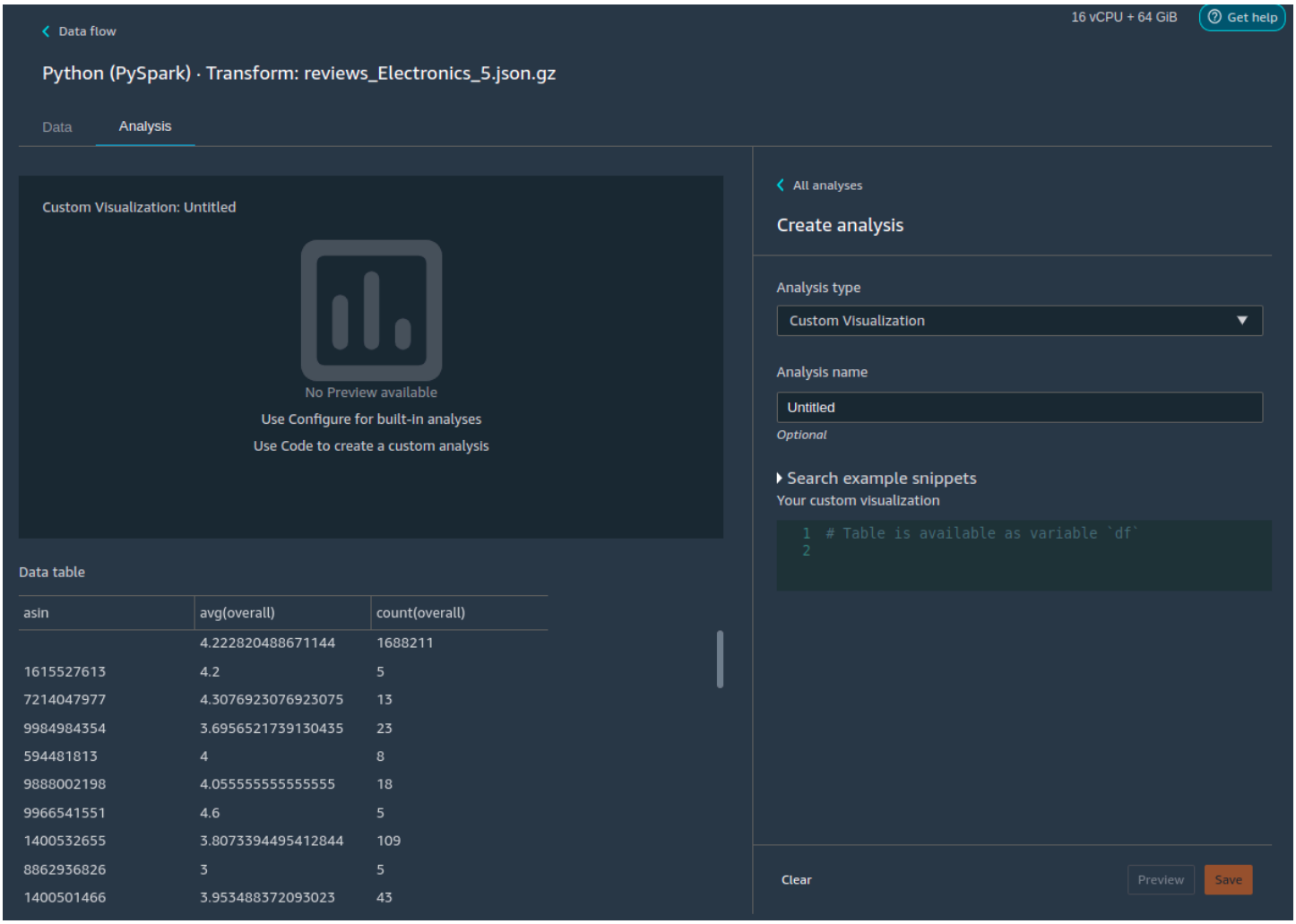
Anda harus memberikan variabel output,`chart`, untuk menyimpan bagan output Altair. Misalnya, Anda dapat menggunakan blok kode berikut untuk membuat histogram khusus menggunakan dataset Titanic.

```
import altair as alt
df = df.iloc[:30]
df = df.rename(columns={"Age": "value"})
df = df.assign(count=df.groupby('value').value.transform('count'))
df = df[["value", "count"]]
base = alt.Chart(df)
bar = base.mark_bar().encode(x=alt.X('value', bin=True, axis=None), y=alt.Y('count'))
rule = base.mark_rule(color='red').encode(
    x='mean(value):Q',
    size=alt.value(5))
chart = bar + rule
```

Untuk membuat visualisasi kustom:

1. Di samping node yang berisi transformasi yang ingin Anda visualisasikan, pilih +.
2. Pilih Tambahkan analisis.

3. Untuk jenis Analisis, pilih Visualisasi Kustom.
4. Untuk nama Analisis, tentukan nama.
5. Masukkan kode Anda di kotak kode.
6. Pilih Pratinjau untuk melihat visualisasi Anda.
7. Pilih Simpan untuk menambahkan visualisasi Anda.



Jika Anda tidak tahu cara menggunakan paket visualisasi Altair dengan Python, Anda dapat menggunakan cuplikan kode khusus untuk membantu Anda memulai.

Data Wrangler memiliki koleksi cuplikan visualisasi yang dapat dicari. Untuk menggunakan cuplikan visualisasi, pilih Cari contoh cuplikan dan tentukan kueri di bilah pencarian.

Contoh berikut menggunakan cuplikan kode scatterplot Binned. Ini memplot histogram untuk 2 dimensi.

Cuplikan memiliki komentar untuk membantu Anda memahami perubahan yang perlu Anda buat pada kode. Anda biasanya perlu menentukan nama kolom dataset Anda dalam kode.

```
import altair as alt

# Specify the number of top rows for plotting
rows_number = 1000
df = df.head(rows_number)
# You can also choose bottom rows or randomly sampled rows
# df = df.tail(rows_number)
# df = df.sample(rows_number)


chart = (
    alt.Chart(df)
    .mark_circle()
    .encode(
        # Specify the column names for binning and number of bins for X and Y axis
        x=alt.X("col1:Q", bin=alt.Bin(maxbins=20)),
        y=alt.Y("col2:Q", bin=alt.Bin(maxbins=20)),
        size="count()",
    )
)

# :Q specifies that label column has quantitative type.
# For more details on Altair typing refer to
# https://altair-viz.github.io/user_guide/encoding.html#encoding-data-types
```

Menggunakan Kembali Alur Data untuk Kumpulan Data yang Berbeda

Untuk sumber data Amazon Simple Storage Service (Amazon S3), Anda dapat membuat dan menggunakan parameter. Parameter adalah variabel yang telah Anda simpan dalam aliran Data Wrangler Anda. Nilainya dapat berupa bagian mana pun dari jalur Amazon S3 sumber data. Gunakan parameter untuk dengan cepat mengubah data yang Anda impor ke dalam aliran Data Wrangler atau mengekspor ke pekerjaan pemrosesan. Anda juga dapat menggunakan parameter untuk memilih dan mengimpor subset tertentu dari data Anda.

Setelah Anda membuat alur Data Wrangler, Anda mungkin telah melatih model pada data yang telah Anda ubah. Untuk kumpulan data yang memiliki skema yang sama, Anda dapat menggunakan parameter untuk menerapkan transformasi yang sama pada kumpulan data yang berbeda dan

melatih model yang berbeda. Anda dapat menggunakan kumpulan data baru untuk melakukan inferensi dengan model Anda atau Anda dapat menggunakannya untuk melatih kembali model Anda.

Secara umum, parameter memiliki atribut berikut:

- Nama - Nama yang Anda tentukan untuk parameter
- Jenis - Jenis nilai yang diwakili oleh parameter
- Nilai default - Nilai parameter saat Anda tidak menentukan nilai baru

> **Note**
>
> Parameter Datetime memiliki atribut rentang waktu yang mereka gunakan sebagai nilai default.

Data Wrangler menggunakan kurawal kurawal,`{{}}`, untuk menunjukkan bahwa parameter sedang digunakan di jalur Amazon S3. Misalnya, Anda dapat memiliki URL seperti`s3://DOC-EXAMPLE-BUCKET1/{{example_parameter_name}}/example-dataset.csv`.

Anda membuat parameter saat mengedit sumber data Amazon S3 yang telah Anda impor. Anda dapat mengatur setiap bagian dari path file ke nilai parameter. Anda dapat mengatur nilai parameter ke nilai atau pola. Berikut ini adalah tipe nilai parameter yang tersedia dalam aliran Data Wrangler:

- Angka
- String
- Pola
- Datetime

> **Note**
>
> Anda tidak dapat membuat parameter pola atau parameter datetime untuk nama bucket di jalur Amazon S3.

Anda harus menetapkan angka sebagai nilai default dari parameter angka. Anda dapat mengubah nilai parameter ke nomor yang berbeda saat mengedit parameter atau saat Anda meluncurkan

pekerjaan pemrosesan. Misalnya, di jalur S3`s3://DOC-EXAMPLE-BUCKET/example-prefix/example-file-1.csv`, Anda dapat membuat parameter angka bernama `number_parameter` di tempat. 1 Jalur S3 Anda sekarang muncul sebagai`s3://DOC-EXAMPLE-BUCKET/example-prefix/example-file-{{number_parameter}}.csv`. Jalur terus menunjuk ke `example-file-1.csv` kumpulan data hingga Anda mengubah nilai parameter. Jika Anda mengubah nilai `number_parameter` 2 ke jalur sekarang`s3://DOC-EXAMPLE-BUCKET/example-prefix/example-file-2.csv`. Anda dapat mengimpor `example-file-2.csv` ke Data Wrangler jika Anda telah mengunggah file ke lokasi Amazon S3 tersebut.

Parameter string menyimpan string sebagai nilai defaultnya. Misalnya, di jalur S3`s3://DOC-EXAMPLE-BUCKET/example-prefix/example-file-1.csv`, Anda dapat membuat parameter string bernama `string_parameter` di tempat nama file,. `example-file-1.csv` Jalan sekarang muncul sebagai`s3://DOC-EXAMPLE-BUCKET/example-prefix/{{string_parameter}}`. Ini terus cocok`s3://DOC-EXAMPLE-BUCKET/example-prefix/example-file-1.csv`, sampai Anda mengubah nilai parameter.

Alih-alih menentukan nama file sebagai parameter string, Anda dapat membuat parameter string menggunakan seluruh jalur Amazon S3. Anda dapat menentukan kumpulan data dari lokasi Amazon S3 mana pun di parameter string.

Parameter pola menyimpan string ekspresi reguler (Python REGEX) sebagai nilai defaultnya. Anda dapat menggunakan parameter pola untuk mengimpor beberapa file data sekaligus. Untuk mengimpor lebih dari satu objek sekaligus, tentukan nilai parameter yang cocok dengan objek Amazon S3 yang Anda impor.

Anda juga dapat membuat parameter pola untuk kumpulan data berikut:

- *s3://DOC-CONTOH-BUCKET1 /example-prefix/example-file-1.csv*
- *s3://DOC-CONTOH-BUCKET1 /example-prefix/example-file-2.csv*
- *s3://DOC-CONTOH-BUCKET1 /example-prefix/example-file-10.csv*
- *s3://DOC-EXAMPLE-BUCKET /example-prefix/example-file-0123.csv*

Untuk`s3://DOC-EXAMPLE-BUCKET1/example-prefix/example-file-1.csv`, Anda dapat membuat parameter pola di tempat`1`, dan mengatur nilai default parameter ke`\d+`. String `\d+` REGEX cocok dengan satu atau lebih digit desimal. Jika Anda membuat parameter pola bernama`pattern_parameter`, jalur S3 Anda akan muncul sebagai`s3://DOC-EXAMPLE-BUCKET1/example-prefix/example-file-{{pattern_parameter}}.csv`.

Anda juga dapat menggunakan parameter pola untuk mencocokkan semua objek CSV dalam bucket Anda. Untuk mencocokkan semua objek dalam bucket, buat parameter pola dengan nilai default `.*` dan atur path kes`s3://DOC-EXAMPLE-BUCKET/{{pattern_parameter}}.csv`. `.*`Karakter cocok dengan karakter string apa pun di jalur.

`s3://DOC-EXAMPLE-BUCKET/{{pattern_parameter}}.csv`Path dapat cocok dengan dataset berikut.

- `example-file-1.csv`
- `other-example-file.csv`
- `example-file-a.csv`

Parameter datetime menyimpan format dengan informasi berikut:

- Format untuk mengurai string di dalam jalur Amazon S3.
- Rentang waktu relatif untuk membatasi nilai datetime yang cocok

Misalnya, di jalur file Amazon S3,`s3://DOC-EXAMPLE-BUCKET/2020/01/01/example-dataset.csv`, 2020/01/01 mewakili datetime dalam format file. `year/month/day` Anda dapat mengatur rentang waktu parameter ke interval seperti `1 years` atau`24 hours`. Interval `1 years` kecocokan semua jalur S3 dengan waktu tanggal yang berada di antara waktu saat ini dan waktu tepat setahun sebelum waktu saat ini. Waktu saat ini adalah waktu ketika Anda mulai mengekspor transformasi yang telah Anda buat ke data. Untuk informasi lebih lanjut tentang mengekspor data, lihatEkspor. Jika tanggal saat ini adalah 2022/01/01 dan rentang waktunya`1 years`, jalur S3 cocok dengan kumpulan data seperti berikut:

- *`s3://DOC-EXAMPLE-BUCKET /2021/01/01/example-dataset.csv`*
- *`s3://DOC-EXAMPLE-BUCKET /2021/06/30/example-dataset.csv`*
- *`s3://DOC-EXAMPLE-BUCKET /2021/12/31/example-dataset.csv`*

Nilai datetime dalam rentang waktu relatif berubah seiring berjalannya waktu. Jalur S3 yang termasuk dalam rentang waktu relatif mungkin juga berbeda.

Untuk jalur file Amazon S3,`s3://DOC-EXAMPLE-BUCKET1/20200101/example-dataset.csv`, `20220101` adalah contoh jalur yang dapat menjadi parameter datetime.

Untuk melihat tabel semua parameter yang telah Anda buat di alur Data Wrangler, pilih `{{}}` di sebelah kanan kotak teks yang berisi jalur Amazon S3. Jika Anda tidak lagi memerlukan parameter yang telah Anda buat, Anda dapat mengedit atau menghapus. Untuk mengedit atau menghapus parameter, pilih ikon di sebelah kanan parameter.

> **Important**
>
> Sebelum Anda menghapus parameter, pastikan Anda belum menggunakannya di mana pun dalam alur Data Wrangler Anda. Parameter yang dihapus yang masih dalam aliran menyebabkan kesalahan.

Anda dapat membuat parameter untuk setiap langkah aliran Data Wrangler Anda. Anda dapat menyunting atau menghapus parameter apa pun yang Anda buat. Jika Anda menerapkan transformasi ke data yang tidak lagi relevan dengan kasus penggunaan Anda, Anda dapat memodifikasi nilai parameter. Memodifikasi nilai parameter mengubah data yang Anda impor.

Bagian berikut memberikan contoh tambahan dan panduan umum tentang penggunaan parameter. Anda dapat menggunakan bagian untuk memahami parameter yang paling sesuai untuk Anda.

> **Note**
>
> Bagian berikut berisi prosedur yang menggunakan antarmuka Data Wrangler untuk mengganti parameter dan membuat pekerjaan pemrosesan.
> Anda juga dapat mengganti parameter dengan menggunakan prosedur berikut.
>
> Untuk mengekspor aliran Data Wrangler Anda dan mengganti nilai parameter, lakukan hal berikut.
>
> 1. Pilih + di sebelah simpul yang ingin Anda ekspor.
> 2. Pilih Ekspor ke.
> 3. Pilih lokasi tempat Anda mengekspor data.
> 4. Di bawah`parameter_overrides`, tentukan nilai yang berbeda untuk parameter yang telah Anda buat.
> 5. Jalankan Notebook Jupyter.

Menerapkan aliran Data Wrangler ke file menggunakan pola

Anda dapat menggunakan parameter untuk menerapkan transformasi dalam alur Data Wrangler ke file berbeda yang cocok dengan pola di jalur URI Amazon S3. Ini membantu Anda menentukan file di bucket S3 yang ingin Anda ubah dengan spesifisitas tinggi. Misalnya, Anda mungkin memiliki kumpulan data dengan jalurnya`s3://DOC-EXAMPLE-BUCKET1/example-prefix-0/example-prefix-1/example-prefix-2/example-dataset.csv`. Kumpulan data yang berbeda bernama `example-dataset.csv` disimpan di bawah banyak contoh awalan yang berbeda. Awalan mungkin juga diberi nomor secara berurutan. Anda dapat membuat pola untuk angka di Amazon S3 URI. Parameter pola menggunakan REGEX untuk memilih sejumlah file yang cocok dengan pola ekspresi. Berikut ini adalah pola REGEX yang mungkin berguna:

- `.*`— Cocokkan nol atau lebih karakter apa pun, kecuali karakter baris baru
- `.+`— Cocokkan satu atau lebih karakter apa pun, tidak termasuk karakter baris baru
- `\d+`— Cocokkan satu atau lebih digit desimal
- `\w+`— Cocokkan satu atau lebih karakter alfanumerik
- `[abc-_]{2,4}`— Cocokkan string dua, tiga, atau empat karakter yang terdiri dari kumpulan karakter yang disediakan dalam satu set tanda kurung
- `abc|def`— Cocokkan satu string atau lainnya. Misalnya, operasi cocok dengan salah satu `abc` atau `def`

Anda dapat mengganti setiap nomor di jalur berikut dengan satu parameter yang memiliki nilai`\d+`.

- `s3://DOC-EXAMPLE-BUCKET1/example-prefix-3/example-prefix-4/example-prefix-5/example-dataset.csv`
- `s3://DOC-EXAMPLE-BUCKET1/example-prefix-8/example-prefix-12/example-prefix-13/example-dataset.csv`
- `s3://DOC-EXAMPLE-BUCKET1/example-prefix-4/example-prefix-9/example-prefix-137/example-dataset.csv`

Prosedur berikut membuat parameter pola untuk dataset dengan paths`s3://DOC-EXAMPLE-BUCKET1/example-prefix-0/example-prefix-1/example-prefix-2/example-dataset.csv`.

Untuk membuat parameter pola, lakukan hal berikut.

1. Di samping dataset yang telah Anda impor, pilih Edit dataset.
2. Sorot `0` masuk`example-prefix-0`.
3. Tentukan nilai untuk bidang berikut:
   - Nama — Nama untuk parameter
   - Jenis - Pola
   - Nilai —\ d+ekspresi reguler yang sesuai dengan satu atau lebih digit
4. Pilih Buat.
5. Ganti 1 dan 2 di jalur URI S3 dengan parameter. Jalur harus memiliki format berikut: `s3://DOC-EXAMPLE-BUCKET1/example-prefix-{{example_parameter_name}}/example-prefix-{{example_parameter_name}}/example-prefix-{{example_parameter_name}}/example-dataset.csv`

Berikut ini adalah prosedur umum untuk membuat parameter pola.

1. Arahkan ke alur Data Wrangler Anda.
2. Di samping dataset yang telah Anda impor, pilih Edit dataset.
3. Sorot bagian URI yang Anda gunakan sebagai nilai parameter pola.
4. Pilih Buat parameter kustom.
5. Tentukan nilai untuk bidang berikut:
   - Nama — Nama untuk parameter
   - Jenis - Pola
   - Nilai - Ekspresi reguler yang berisi pola yang ingin Anda simpan.
6. Pilih Buat.

Menerapkan aliran Data Wrangler ke file menggunakan nilai numerik

Anda dapat menggunakan parameter untuk menerapkan transformasi dalam aliran Data Wrangler Anda ke file berbeda yang memiliki jalur serupa. Misalnya, Anda mungkin memiliki kumpulan data dengan jalurnya`s3://DOC-EXAMPLE-BUCKET1/example-prefix-0/example-prefix-1/example-prefix-2/example-dataset.csv`.

Anda mungkin memiliki transformasi dari alur Data Wrangler yang telah Anda terapkan ke kumpulan data di bawahnya. `example-prefix-1` Anda mungkin ingin menerapkan transformasi yang sama

dengan `example-dataset.csv` yang termasuk di bawah `example-prefix-10` atau`example-prefix-20`.

Anda dapat membuat parameter yang menyimpan nilai`1`. Jika Anda ingin menerapkan transformasi ke kumpulan data yang berbeda, Anda dapat membuat pekerjaan pemrosesan yang menggantikan nilai parameter dengan nilai yang berbeda. Parameter bertindak sebagai pengganti bagi Anda untuk mengubah ketika Anda ingin menerapkan transformasi dari aliran Data Wrangler Anda ke data baru. Anda dapat mengganti nilai parameter saat membuat pekerjaan pemrosesan Data Wrangler untuk menerapkan transformasi dalam aliran Data Wrangler Anda ke kumpulan data yang berbeda.

Gunakan prosedur berikut ini untuk membuat parameter numerik untuk`s3://DOC-EXAMPLE-BUCKET1/example-prefix-0/example-prefix-1/example-prefix-2/example-dataset.csv`.

Untuk membuat parameter untuk jalur URI S3 sebelumnya, lakukan hal berikut.

1. Arahkan ke alur Data Wrangler Anda.
2. Di samping dataset yang telah Anda impor, pilih Edit dataset.
3. Sorot nomor dalam contoh awalan. `example-prefix-number`
4. Pilih Buat parameter kustom.
5. Untuk Nama, tentukan nama untuk parameter.
6. Untuk Type, pilih Integer.
7. Untuk Nilai, tentukan nomornya.
8. Buat parameter untuk angka yang tersisa dengan mengulangi prosedur.

Setelah Anda membuat parameter, terapkan transformasi ke dataset Anda dan buat node tujuan untuk mereka. Untuk informasi selengkapnya tentang simpul tujuan, lihatEkspor.

Gunakan prosedur berikut untuk menerapkan transformasi dari aliran Data Wrangler Anda ke rentang waktu yang berbeda. Ini mengasumsikan bahwa Anda telah membuat node tujuan untuk transformasi dalam alur Anda.

Untuk mengubah nilai parameter numerik dalam pekerjaan pemrosesan Data Wrangler, lakukan hal berikut.

1. Dari alur Data Wrangler Anda, pilih Buat pekerjaan
2. Pilih hanya node tujuan yang berisi transformasi ke dataset yang berisi parameter datetime.

3. Pilih Konfigurasi pekerjaan.
4. Pilih Parameter.
5. Pilih nama parameter yang telah Anda buat.
6. Ubah nilai dari parameter.
7. Ulangi prosedur untuk parameter lainnya.
8. Pilih Jalankan.

Menerapkan aliran Data Wrangler ke file menggunakan string

Anda dapat menggunakan parameter untuk menerapkan transformasi dalam aliran Data Wrangler Anda ke file berbeda yang memiliki jalur serupa. Misalnya, Anda mungkin memiliki kumpulan data dengan jalurnya`s3://DOC-EXAMPLE-BUCKET1/example-prefix/example-dataset.csv`.

Anda mungkin memiliki transformasi dari alur Data Wrangler yang telah Anda terapkan ke kumpulan data di bawahnya. `example-prefix` Anda mungkin ingin menerapkan transformasi yang sama ke `example-dataset.csv` bawah `another-example-prefix` atau`example-prefix-20`.

Anda dapat membuat parameter yang menyimpan nilai`example-prefix`. Jika Anda ingin menerapkan transformasi ke kumpulan data yang berbeda, Anda dapat membuat pekerjaan pemrosesan yang menggantikan nilai parameter dengan nilai yang berbeda. Parameter bertindak sebagai pengganti bagi Anda untuk mengubah ketika Anda ingin menerapkan transformasi dari aliran Data Wrangler Anda ke data baru. Anda dapat mengganti nilai parameter saat membuat pekerjaan pemrosesan Data Wrangler untuk menerapkan transformasi dalam aliran Data Wrangler Anda ke kumpulan data yang berbeda.

Gunakan prosedur berikut ini untuk membuat sebuah parameter string untuk`s3://DOC-EXAMPLE-BUCKET1/example-prefix/example-dataset.csv`.

Untuk membuat parameter untuk jalur URI S3 sebelumnya, lakukan hal berikut.

1. Arahkan ke alur Data Wrangler Anda.
2. Di samping dataset yang telah Anda impor, pilih Edit dataset.
3. Sorot contoh awalan,`example-prefix`.
4. Pilih Buat parameter kustom.
5. Untuk Nama, tentukan nama untuk parameter.
6. Untuk Type, pilih String.

7. Untuk Nilai, tentukan awalan.

Setelah Anda membuat parameter, terapkan transformasi ke dataset Anda dan buat node tujuan untuk mereka. Untuk informasi selengkapnya tentang simpul tujuan, lihatEkspor.

Gunakan prosedur berikut untuk menerapkan transformasi dari aliran Data Wrangler Anda ke rentang waktu yang berbeda. Ini mengasumsikan bahwa Anda telah membuat node tujuan untuk transformasi dalam alur Anda.

Untuk mengubah nilai parameter numerik dalam pekerjaan pemrosesan Data Wrangler, lakukan hal berikut:

1. Dari alur Data Wrangler Anda, pilih Buat pekerjaan
2. Pilih hanya node tujuan yang berisi transformasi ke dataset yang berisi parameter datetime.
3. Pilih Konfigurasi pekerjaan.
4. Pilih Parameter.
5. Pilih nama parameter yang telah Anda buat.
6. Ubah nilai dari parameter.
7. Ulangi prosedur untuk parameter lainnya.
8. Pilih Jalankan.

Menerapkan aliran Data Wrangler ke rentang waktu tanggal yang berbeda

Gunakan parameter datetime untuk menerapkan transformasi dalam alur Data Wrangler Anda ke rentang waktu yang berbeda. Sorot bagian URI Amazon S3 yang memiliki stempel waktu dan buat parameter untuknya. Saat Anda membuat parameter, Anda menentukan rentang waktu dari waktu saat ini ke waktu di masa lalu. Misalnya, Anda mungkin memiliki URI Amazon S3 yang terlihat seperti berikut ini:. `s3://DOC-EXAMPLE-BUCKET1/example-prefix/2022/05/15/example-dataset.csv` Anda dapat menyimpan `2022/05/15` sebagai parameter datetime. Jika Anda menentukan tahun sebagai rentang waktu, rentang waktu mencakup saat Anda menjalankan pekerjaan pemrosesan yang berisi parameter datetime dan waktu tepat satu tahun yang lalu. Jika saat Anda menjalankan pekerjaan pemrosesan adalah 6 September 2022 atau`2022/09/06`, rentang waktu dapat mencakup yang berikut:

- `s3://DOC-EXAMPLE-BUCKET1/example-prefix/2022/03/15/example-dataset.csv`
- `s3://DOC-EXAMPLE-BUCKET1/example-prefix/2022/01/08/example-dataset.csv`

- `s3://DOC-EXAMPLE-BUCKET1/example-prefix/2022/07/31/example-dataset.csv`
- `s3://DOC-EXAMPLE-BUCKET1/example-prefix/2021/09/07/example-dataset.csv`

Transformasi dalam aliran Data Wrangler berlaku untuk semua awalan sebelumnya. Mengubah nilai parameter dalam pekerjaan pemrosesan tidak mengubah nilai parameter dalam aliran Data Wrangler. Untuk menerapkan transformasi ke kumpulan data dalam rentang waktu yang berbeda, lakukan hal berikut:

1. Buat simpul tujuan yang berisi semua transformasi yang ingin Anda gunakan.
2. Buat pekerjaan Data Wrangler.
3. Konfigurasikan pekerjaan untuk menggunakan rentang waktu yang berbeda untuk parameter. Mengubah nilai parameter dalam pekerjaan pemrosesan tidak mengubah nilai parameter dalam aliran Data Wrangler.

Untuk informasi selengkapnya tentang node tujuan dan pekerjaan Data Wrangler, lihat. Ekspor

Prosedur berikut membuat parameter datetime untuk jalur Amazon S3:. `s3://DOC-EXAMPLE-BUCKET1/example-prefix/2022/05/15/example-dataset.csv`

Untuk membuat parameter datetime untuk jalur URI S3 sebelumnya, lakukan hal berikut.

1. Arahkan ke alur Data Wrangler Anda.
2. Di samping dataset yang telah Anda impor, pilih Edit dataset.
3. Sorot bagian URI yang Anda gunakan sebagai nilai parameter datetime.
4. Pilih Buat parameter kustom.
5. Untuk Nama, tentukan nama untuk parameter.
6. Untuk Type, pilih Datetime.

> **Note**
>
> Secara default, Data Wrangler memilih Predefined, yang menyediakan menu dropdown bagi Anda untuk memilih format tanggal. Namun, format stempel waktu yang Anda gunakan mungkin tidak tersedia. Alih-alih menggunakan Predefined sebagai opsi default, Anda dapat memilih Custom dan menentukan format stempel waktu secara manual.

7. Untuk format Tanggal, buka menu dropdown berikut Predefined dan pilih YYYY/mm/dd. Formatnya, yyyy/mm/dd, sesuai dengan tahun/bulan/hari stempel waktu.
8. Untuk Timezone, pilih zona waktu.

> **Note**
>
> Data yang Anda analisis mungkin memiliki stempel waktu yang diambil di zona waktu yang berbeda dari zona waktu Anda. Pastikan zona waktu yang Anda pilih cocok dengan zona waktu data.

9. Untuk Rentang waktu, tentukan rentang waktu untuk parameter tersebut.
10. (Opsional) Masukkan deskripsi untuk menjelaskan bagaimana Anda menggunakan parameter.
11. Pilih Buat.

Setelah Anda membuat parameter datetime, terapkan transformasi ke dataset Anda dan buat node tujuan untuk mereka. Untuk informasi selengkapnya tentang simpul tujuan, lihatEkspor.

Gunakan prosedur berikut untuk menerapkan transformasi dari aliran Data Wrangler Anda ke rentang waktu yang berbeda. Ini mengasumsikan bahwa Anda telah membuat node tujuan untuk transformasi dalam alur Anda.

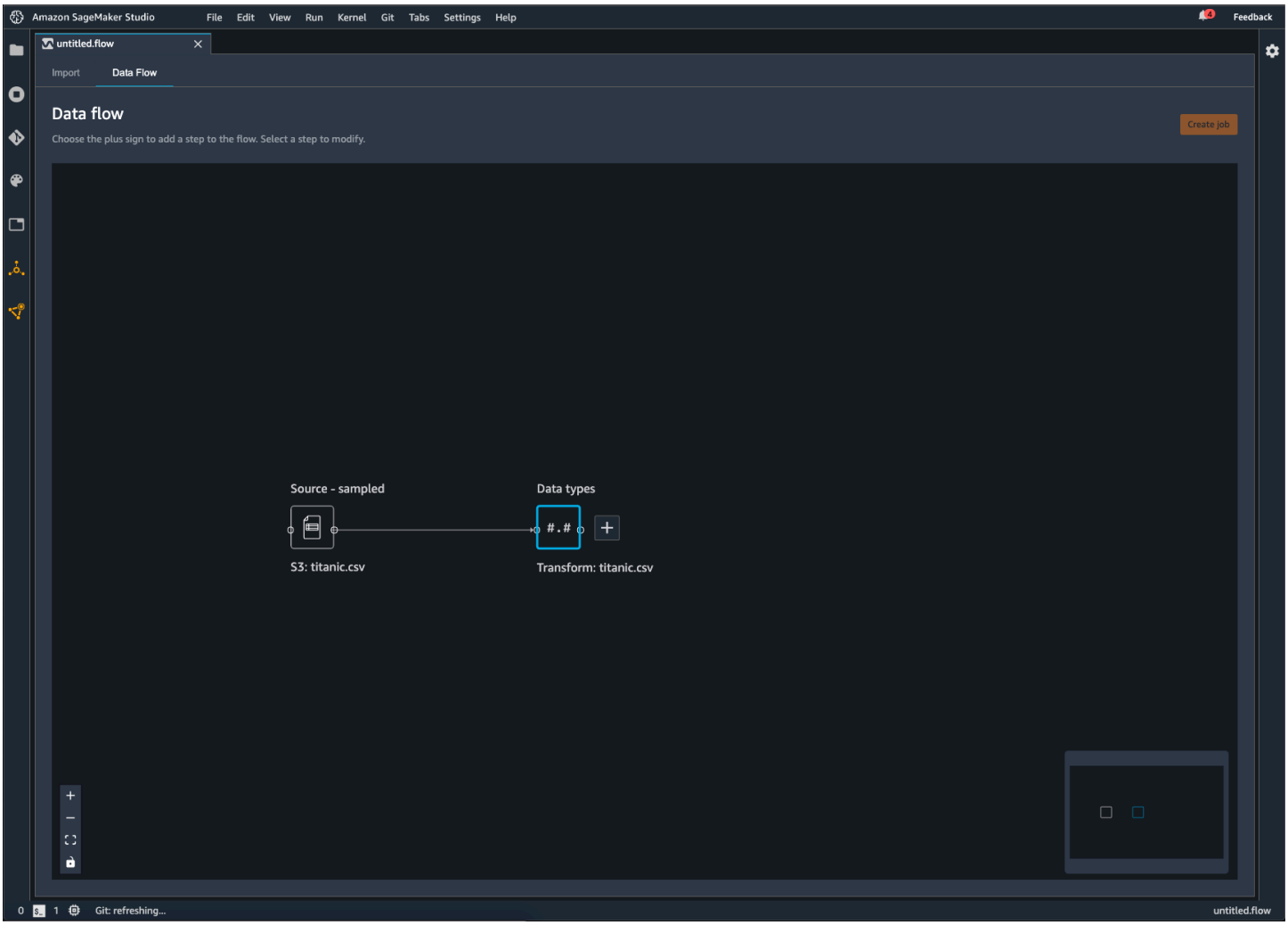
Untuk mengubah nilai parameter datetime dalam pekerjaan pemrosesan Data Wrangler, lakukan hal berikut:

1. Dari alur Data Wrangler Anda, pilih Buat pekerjaan
2. Pilih hanya node tujuan yang berisi transformasi ke dataset yang berisi parameter datetime.
3. Pilih Konfigurasi pekerjaan.
4. Pilih Parameter.
5. Pilih nama parameter datetime yang telah Anda buat.
6. Untuk rentang waktu, ubah rentang waktu untuk kumpulan data.
7. Pilih Jalankan.

Ekspor

Dalam alur Data Wrangler, Anda dapat mengekspor beberapa atau semua transformasi yang telah Anda buat ke pipeline pemrosesan data Anda.

Alur Data Wrangler adalah serangkaian langkah persiapan data yang telah Anda lakukan pada data Anda. Dalam persiapan data Anda, Anda melakukan satu atau lebih transformasi ke data Anda. Setiap transformasi dilakukan dengan menggunakan langkah transformasi. Aliran memiliki serangkaian node yang mewakili impor data Anda dan transformasi yang telah Anda lakukan. Untuk contoh simpul, lihat gambar berikut.



Gambar sebelumnya menunjukkan aliran Data Wrangler dengan dua node. Node Source - sampel menunjukkan sumber data dari mana Anda telah mengimpor data Anda. Node tipe Data menunjukkan bahwa Data Wrangler telah melakukan transformasi untuk mengubah kumpulan data menjadi format yang dapat digunakan.

Setiap transformasi yang Anda tambahkan ke aliran Data Wrangler muncul sebagai node tambahan. Untuk informasi tentang transformasi yang dapat Anda tambahkan, lihatMemindahkan. Gambar berikut menunjukkan aliran Data Wrangler yang memiliki node Rename-column untuk mengubah nama kolom dalam dataset.

Anda dapat mengekspor transformasi data Anda ke yang berikut:

- Amazon S3
- SageMaker Pipa
- Toko SageMaker Fitur Amazon
- Kode Python

> **Important**
>
> Kami menyarankan Anda menggunakan kebijakan `AmazonSageMakerFullAccess` terkelola IAM untuk memberikan AWS izin menggunakan Data Wrangler. Jika Anda tidak menggunakan kebijakan terkelola, Anda dapat menggunakan kebijakan IAM yang memberikan akses Data Wrangler ke bucket Amazon S3. Untuk informasi selengkapnya tentang kebijakan ini, lihatKeamanan dan Izin.

Saat mengekspor aliran data, Anda dikenakan biaya untuk AWS sumber daya yang Anda gunakan. Anda dapat menggunakan tag alokasi biaya untuk mengatur dan mengelola biaya sumber daya tersebut. Anda membuat tag ini untuk profil pengguna Anda dan Data Wrangler secara otomatis menerapkannya ke sumber daya yang digunakan untuk mengekspor aliran data. Untuk informasi selengkapnya, lihat Menggunakan Tag Alokasi Biaya.

Ekspor ke Amazon S3

Data Wrangler memberi Anda kemampuan untuk mengekspor data ke lokasi dalam bucket Amazon S3. Anda dapat menentukan lokasi menggunakan salah satu metode berikut:

- Node tujuan — Dimana Data Wrangler menyimpan data setelah memprosesnya.
- Ekspor ke - Mengekspor data yang dihasilkan dari transformasi ke Amazon S3.
- Ekspor data — Untuk kumpulan data kecil, dapat dengan cepat mengekspor data yang telah Anda ubah.

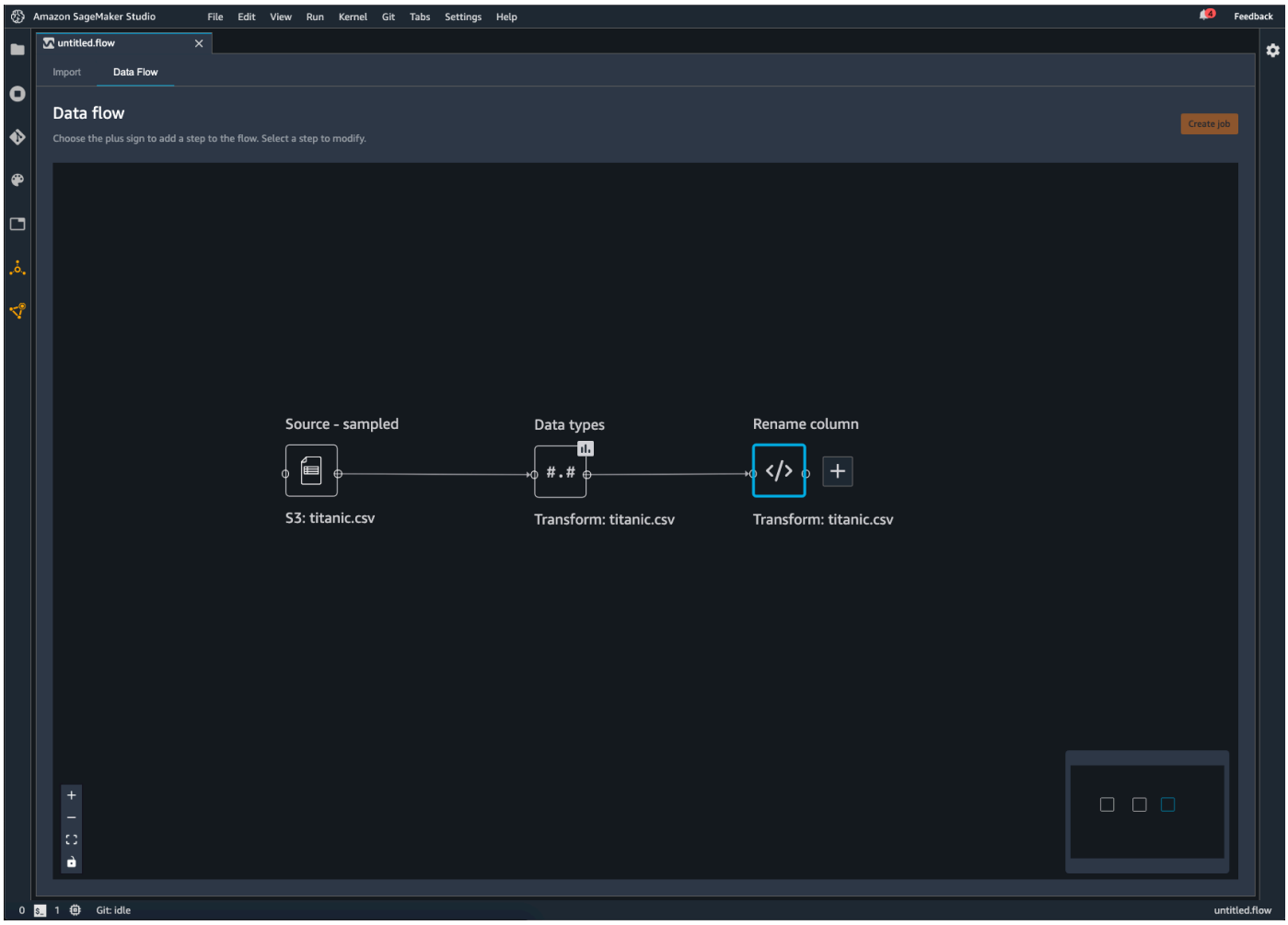
Gunakan bagian berikut untuk mempelajari selengkapnya tentang masing-masing metode ini.

Destination Node

Jika Anda ingin menampilkan serangkaian langkah pemrosesan data yang telah Anda lakukan ke Amazon S3, Anda membuat node tujuan. Node tujuan memberi tahu Data Wrangler tempat menyimpan data setelah Anda memprosesnya. Setelah Anda membuat node tujuan, Anda membuat pekerjaan pemrosesan untuk menampilkan data. Pekerjaan pemrosesan adalah pekerjaan SageMaker pemrosesan Amazon. Saat Anda menggunakan node tujuan, ia menjalankan sumber daya komputasi yang diperlukan untuk menampilkan data yang telah Anda ubah ke Amazon S3.

Anda dapat menggunakan node tujuan untuk mengekspor beberapa transformasi atau semua transformasi yang telah Anda buat dalam alur Data Wrangler Anda.

Anda dapat menggunakan beberapa node tujuan untuk mengekspor transformasi atau set transformasi yang berbeda. Contoh berikut menunjukkan dua node tujuan dalam aliran Data Wrangler tunggal.



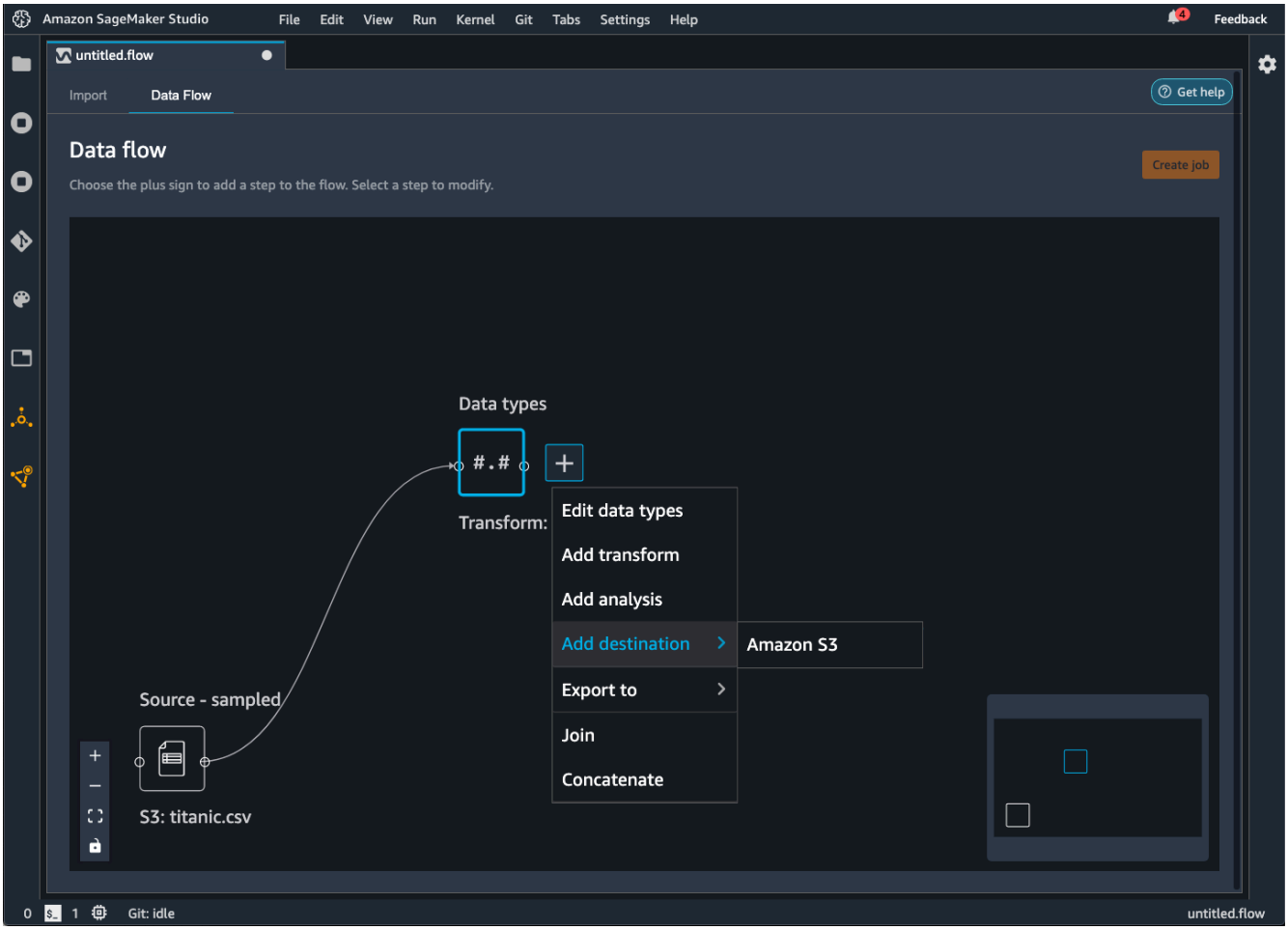
Anda dapat menggunakan prosedur berikut untuk membuat simpul tujuan dan mengekspornya ke bucket Amazon S3.

Untuk mengekspor aliran data Anda, Anda membuat node tujuan dan pekerjaan Data Wrangler untuk mengekspor data. Membuat pekerjaan Data Wrangler memulai pekerjaan SageMaker pemrosesan untuk mengekspor alur Anda. Anda dapat memilih node tujuan yang ingin Anda ekspor setelah Anda membuatnya.

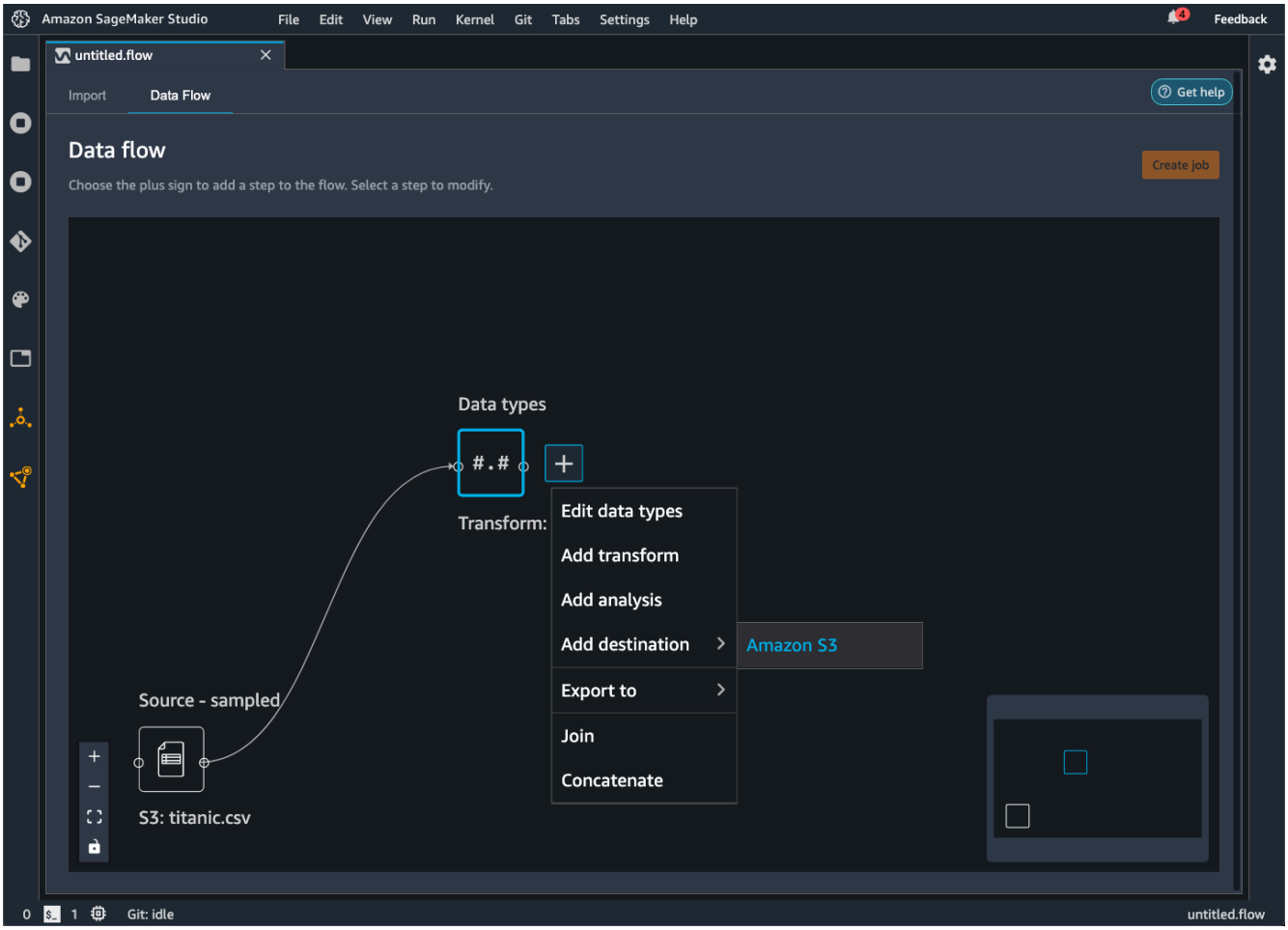
> **Note**
>
> Anda dapat memilih Buat pekerjaan di alur Data Wrangler untuk melihat instruksi untuk menggunakan pekerjaan pemrosesan.

Gunakan prosedur berikut untuk membuat simpul tujuan.

1. Pilih + di sebelah node yang mewakili transformasi yang ingin Anda ekspor.
2. Pilih Tambahkan tujuan.



3. Pilih Amazon S3.



4. Tentukan bidang berikut.

   - Nama Dataset — Nama yang Anda tentukan untuk dataset yang Anda ekspor.
   - Jenis file — Format file yang Anda ekspor.
   - Delimiter (file CSV dan Parket saja) — Nilai yang digunakan untuk memisahkan nilai lainnya.
   - Kompresi (file CSV dan Parket saja) — Metode kompresi yang digunakan untuk mengurangi ukuran file. Anda dapat menggunakan metode kompresi berikut:
     - bzip2
     - mengempiskan
     - gzip
   - (Opsional) Lokasi Amazon S3 — Lokasi S3 yang Anda gunakan untuk menampilkan file.
   - (Opsional) Jumlah partisi — Jumlah kumpulan data yang Anda tulis sebagai output dari pekerjaan pemrosesan.

- (Opsional) Partisi demi kolom - Menulis semua data dengan nilai unik yang sama dari kolom.
- (Opsional) Parameter Inferensi — Memilih Hasilkan artefak inferensi menerapkan semua transformasi yang Anda gunakan dalam aliran Data Wrangler ke data yang masuk ke pipeline inferensi Anda. Model dalam pipeline Anda membuat prediksi pada data yang diubah.

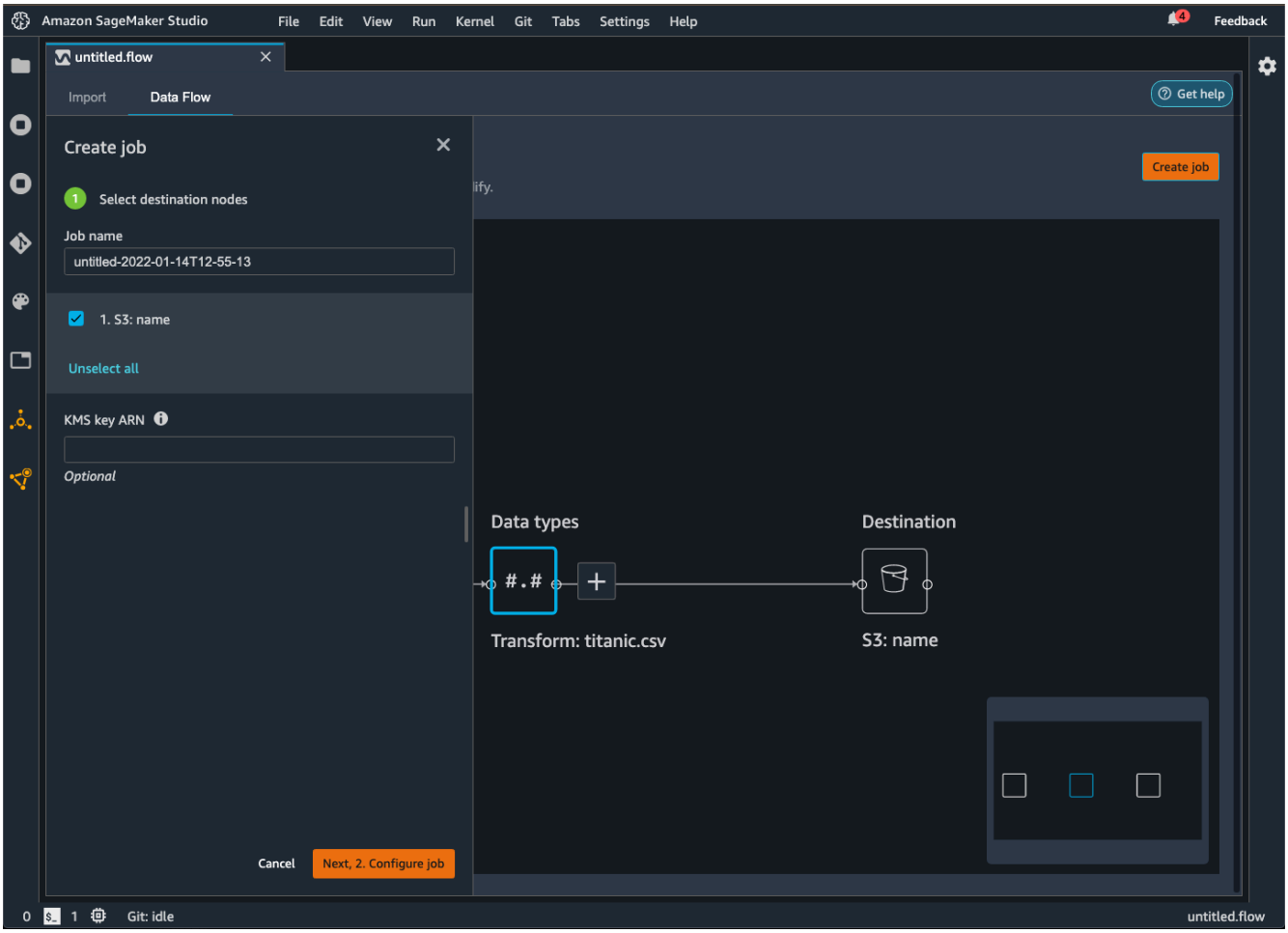
5. Pilih Tambahkan tujuan.

Gunakan prosedur berikut untuk membuat tugas pemrosesan.

Buat pekerjaan dari halaman aliran Data dan pilih node tujuan yang ingin Anda ekspor.

> **Note**
>
> Anda dapat memilih Buat pekerjaan di alur Data Wrangler untuk melihat instruksi untuk membuat pekerjaan pemrosesan.

1. Pilih Buat tugas. Gambar berikut menunjukkan panel yang muncul setelah Anda memilih Buat pekerjaan.



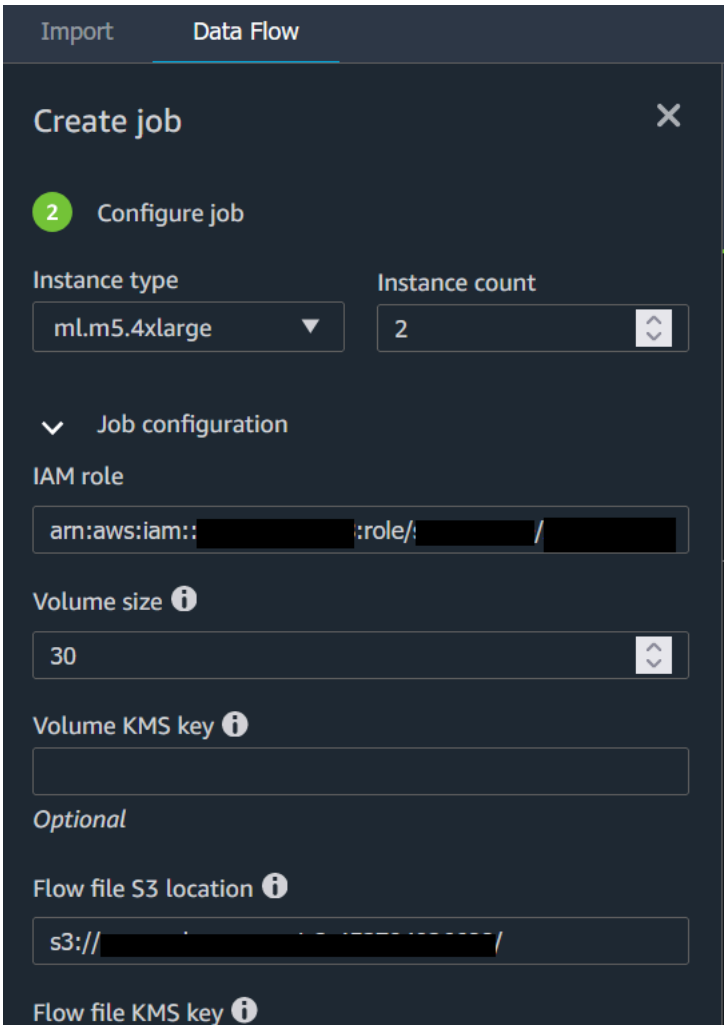
2. Untuk nama Job, tentukan nama pekerjaan ekspor.
3. Pilih simpul tujuan yang ingin diekspor.
4. (Opsional) Tentukan AWS KMS ARN kunci. AWS KMSKunci adalah kunci kriptografi yang dapat Anda gunakan untuk melindungi data Anda. Untuk informasi selengkapnya tentang AWS KMS kunci, lihat AWS Key Management Service.
5. (Opsional) Di bawah parameter Terlatih. pilih Reparasi jika Anda telah melakukan hal berikut:

   - Sumber data per set data
   - Menerapkan transformasi yang menggunakan data Anda untuk membuat kolom baru dalam kumpulan data

   Untuk informasi selengkapnya tentang memperbaiki transformasi yang telah Anda buat ke seluruh kumpulan data, lihat. Reparasi Transformasi ke Seluruh Dataset dan Ekspor Mereka

> **Note**
>
> Untuk data gambar, Data Wrangler mengekspor transformasi yang telah Anda buat ke semua gambar. Memperbaiki transformasi tidak berlaku untuk kasus penggunaan Anda.

6. Pilih Konfigurasi pekerjaan. Gambar berikut menunjukkan halaman tugas Configure.



7. (Opsional) Konfigurasikan pekerjaan Data Wrangler. Anda dapat membuat konfigurasi berikut:

   - Konfigurasi Job
   - Konfigurasi memori percikan
   - Konfigurasi jaringan
   - Tanda

- Parameter
- Jadwal Asosiasi

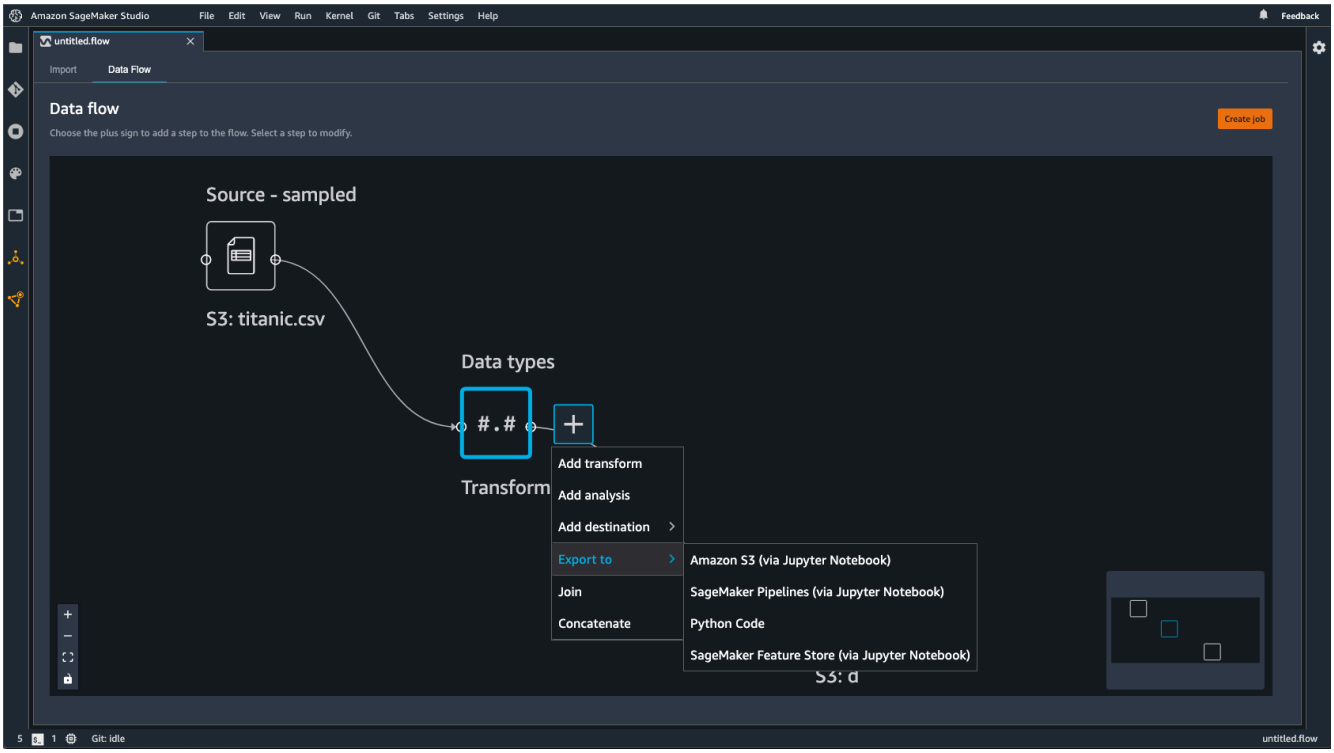
8. Pilih Jalankan.

Export to

Sebagai alternatif untuk menggunakan node tujuan, Anda dapat menggunakan opsi Ekspor ke untuk mengekspor aliran Data Wrangler Anda ke Amazon S3 menggunakan notebook Jupyter. Anda dapat memilih node data apa pun dalam aliran Data Wrangler Anda dan mengekspornya. Mengekspor node data mengekspor transformasi yang diwakili oleh node dan transformasi yang mendahuluinya.

Gunakan prosedur berikut untuk membuat notebook Jupyter dan menjalankannya untuk mengekspor aliran Data Wrangler Anda ke Amazon S3.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor ke.
3. Pilih Amazon S3 (melalui Jupyter Notebook).
4. Jalankan notebook Jupyter.



Saat Anda menjalankan buku catatan, buku catatan akan mengekspor aliran data Anda (file.flow) Wilayah AWS sama dengan alur Data Wrangler.

Notebook menyediakan opsi yang dapat Anda gunakan untuk mengonfigurasi pekerjaan pemrosesan dan data yang dikeluarkan.

> **Important**
>
> Kami memberi Anda konfigurasi pekerjaan untuk mengonfigurasi output data Anda. Untuk opsi partisi dan memori driver, kami sangat menyarankan agar Anda tidak menentukan konfigurasi kecuali Anda sudah memiliki pengetahuan tentang mereka.

Di bawah Job Configurations, Anda dapat mengonfigurasi hal berikut:

- `output_content_type`— Jenis konten dari file output. Digunakan `CSV` sebagai format default, tetapi Anda dapat menentukan`Parquet`.
- `delimiter`— Karakter yang digunakan untuk memisahkan nilai dalam dataset saat menulis ke file CSV.
- `compression`— Jika diatur, kompres file output. Menggunakan gzip sebagai format kompresi default.
- `num_partitions`— Jumlah partisi atau file yang ditulis Data Wrangler sebagai output.
- `partition_by`— Nama-nama kolom yang Anda gunakan untuk mempartisi output.

Untuk mengubah format file output dari CSV ke Parquet, ubah nilainya dari `"CSV"` ke. `"Parquet"` Untuk sisa bidang sebelumnya, batalkan komentar pada baris yang berisi bidang yang ingin Anda tentukan.

Di bawah (Opsional) Konfigurasikan Memori Driver Cluster Spark Anda dapat mengonfigurasi properti Spark untuk pekerjaan itu, seperti memori driver Spark, di kamus. `config`

Berikut ini menunjukkan `config` kamus.

```
config = json.dumps({
    "Classification": "spark-defaults",
    "Properties": {
```

```
        "spark.driver.memory": f"{driver_memory_in_mb}m",
    }
})
```

Untuk menerapkan konfigurasi ke pekerjaan pemrosesan, hapus komentar pada baris berikut:

```
# data_sources.append(ProcessingInput(
#     source=config_s3_uri,
#     destination="/opt/ml/processing/input/conf",
#     input_name="spark-config",
#     s3_data_type="S3Prefix",
#     s3_input_mode="File",
#     s3_data_distribution_type="FullyReplicated"
# ))
```

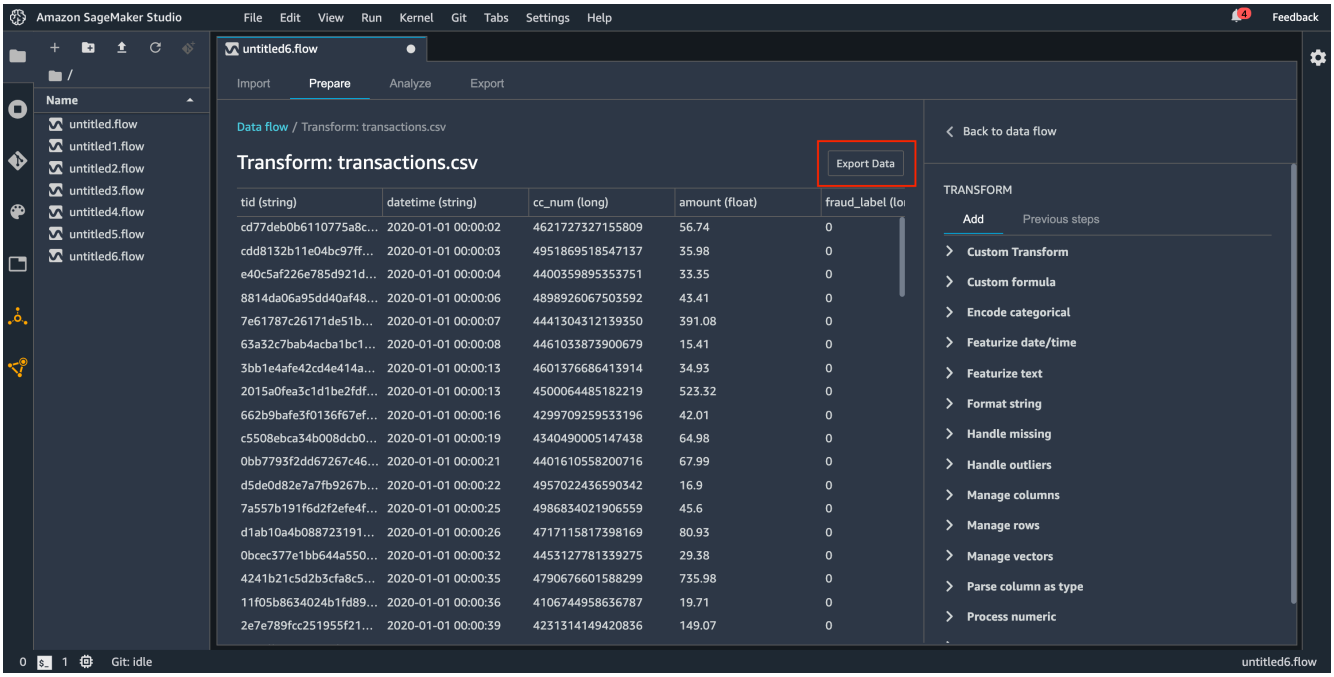
Export data

Jika Anda memiliki transformasi pada kumpulan data kecil yang ingin Anda ekspor dengan cepat, Anda dapat menggunakan metode Ekspor data. Saat Anda mulai memilih Ekspor data, Data Wrangler bekerja secara sinkron untuk mengekspor data yang telah Anda ubah ke Amazon S3. Anda tidak dapat menggunakan Data Wrangler sampai selesai mengekspor data Anda atau membatalkan operasi.

Untuk informasi tentang penggunaan metode Ekspor data dalam alur Data Wrangler Anda, lihat prosedur berikut.

Untuk menggunakan metode data Ekspor:

1. Pilih node dalam aliran Data Wrangler Anda dengan membuka (mengklik dua kali) itu.



2. Konfigurasikan bagaimana Anda ingin mengekspor data.
3. Pilih Ekspor data.

Saat Anda mengekspor aliran data ke bucket Amazon S3, Data Wrangler menyimpan salinan file alur di bucket S3. Ini menyimpan file aliran di bawah awalan data_wrangler_flows. Jika Anda menggunakan bucket Amazon S3 default untuk menyimpan file flow, bucket ini menggunakan konvensi penamaan berikut:. `sagemaker-region-account number` Misalnya, jika nomor akun Anda adalah 111122223333 dan Anda menggunakan Studio Classic di us-east-1, kumpulan data yang Anda impor akan disimpan. `sagemaker-us-east-1-111122223333` Dalam contoh ini, file.flow yang dibuat di us-east-1 disimpan. `s3://sagemaker-region-account number/data_wrangler_flows/`

Ekspor ke SageMaker Pipa

Saat ingin membangun dan menerapkan alur kerja machine learning (ML) skala besar, Anda dapat menggunakan SageMaker Pipelines untuk membuat alur kerja yang mengelola dan menerapkan pekerjaan. SageMaker Dengan SageMaker Pipelines, Anda dapat membangun alur kerja yang mengelola persiapan SageMaker data, pelatihan model, dan memodelkan pekerjaan penerapan. Anda dapat menggunakan algoritma pihak pertama yang SageMaker menawarkan dengan menggunakan SageMaker Pipelines. Untuk informasi lebih lanjut tentang SageMaker Pipelines, lihat SageMaker Pipelines.

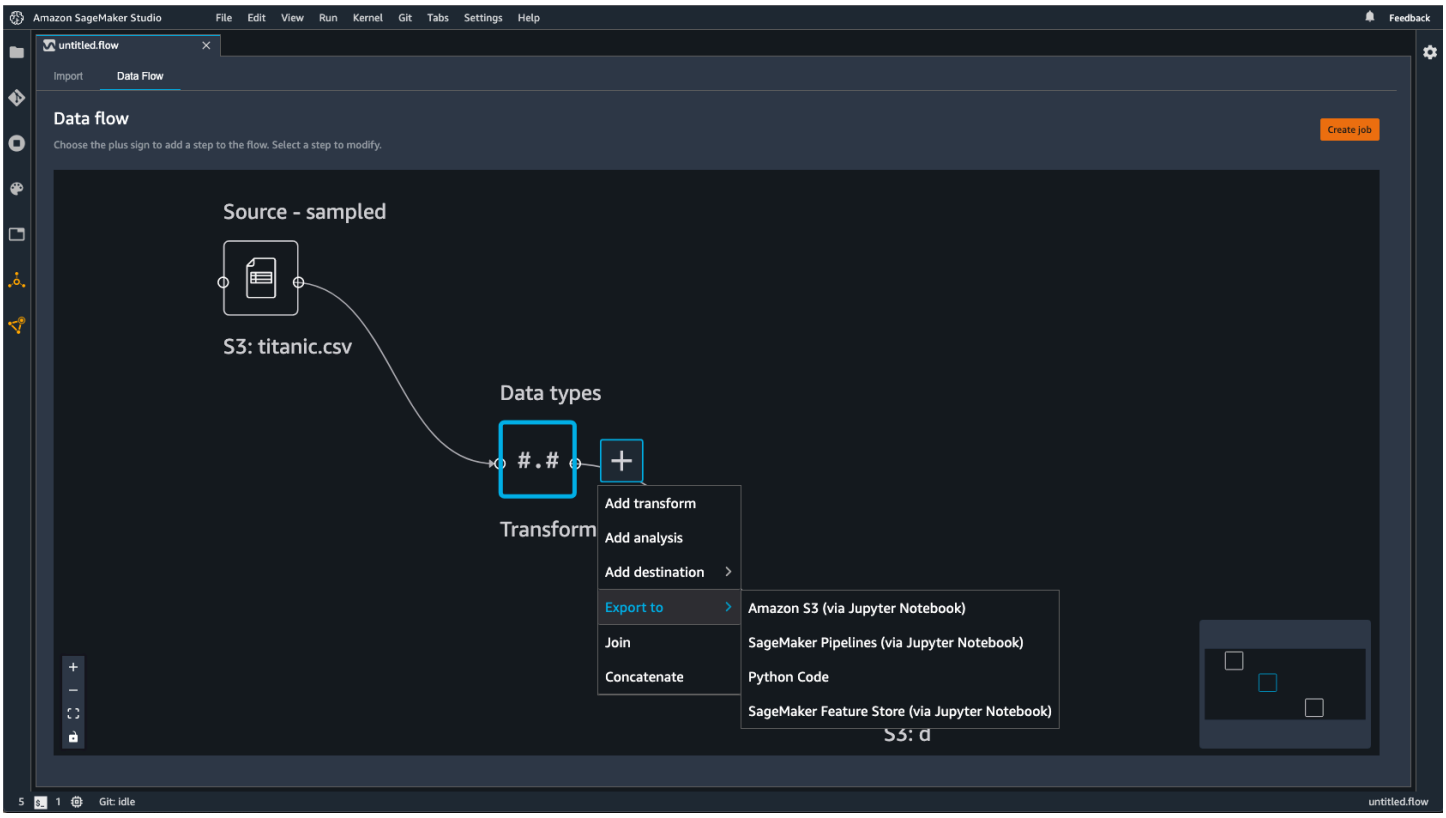
Saat Anda mengekspor satu atau beberapa langkah dari aliran data ke SageMaker Pipelines, Data Wrangler akan membuat buku catatan Jupyter yang dapat Anda gunakan untuk menentukan, membuat instance, menjalankan, dan mengelola pipeline.

Menggunakan Notebook Jupyter untuk Membuat Pipeline

Gunakan prosedur berikut untuk membuat buku catatan Jupyter untuk mengekspor aliran Data Wrangler Anda ke Pipelines. SageMaker

Gunakan prosedur berikut untuk membuat notebook Jupyter dan menjalankannya untuk mengekspor aliran Data Wrangler Anda ke Pipelines. SageMaker

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor ke.
3. Pilih SageMaker Pipelines (melalui Jupyter Notebook).
4. Jalankan notebook Jupyter.



Anda dapat menggunakan notebook Jupyter yang dihasilkan Data Wrangler untuk menentukan pipeline. Pipeline mencakup langkah-langkah pemrosesan data yang ditentukan oleh alur Data Wrangler Anda.

Anda dapat menambahkan langkah tambahan ke pipeline dengan menambahkan langkah-langkah ke `steps` daftar dalam kode berikut di buku catatan:

```
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[instance_type, instance_count],
    steps=[step_process], #Add more steps to this list to run in your Pipeline
)
```

Untuk informasi selengkapnya tentang mendefinisikan pipeline, lihat Mendefinisikan SageMaker Pipeline.

Ekspor ke Endpoint Inferensi

Gunakan alur Data Wrangler Anda untuk memproses data pada saat inferensi dengan membuat pipeline inferensi SageMaker serial dari alur Data Wrangler Anda. Pipa inferensi adalah serangkaian langkah yang menghasilkan model terlatih yang membuat prediksi pada data baru. Pipa inferensi serial dalam Data Wrangler mengubah data mentah dan menyediakannya ke model pembelajaran mesin untuk prediksi. Anda membuat, menjalankan, dan mengelola pipeline inferensi dari notebook Jupyter dalam Studio Classic. Untuk informasi selengkapnya tentang mengakses buku catatan, lihatMenggunakan Notebook Jupyter untuk membuat titik akhir inferensi.

Di dalam buku catatan, Anda dapat melatih model pembelajaran mesin atau menentukan model yang sudah Anda latih. Anda dapat menggunakan Amazon SageMaker Autopilot atau XGBoost untuk melatih model menggunakan data yang telah Anda ubah dalam alur Data Wrangler Anda.

Pipeline menyediakan kemampuan untuk melakukan inferensi batch atau real-time. Anda juga dapat menambahkan aliran Data Wrangler ke SageMaker Model Registry. Untuk informasi selengkapnya tentang model hosting, lihatHost beberapa model dalam satu wadah di belakang satu titik akhir.

> **Important**
>
> Anda tidak dapat mengekspor aliran Data Wrangler ke titik akhir inferensi jika memiliki transformasi berikut:
>
> - Join

- Rangkuman
- Grup oleh

Jika Anda harus menggunakan transformasi sebelumnya untuk menyiapkan data Anda, gunakan prosedur berikut.

Untuk mempersiapkan data Anda untuk inferensi dengan transformasi yang tidak didukung

1. Buat alur Data Wrangler.
2. Menerapkan transformasi sebelumnya yang tidak didukung.
3. Ekspor data ke bucket Amazon S3.
4. Buat alur Data Wrangler terpisah.
5. Impor data yang telah Anda ekspor dari alur sebelumnya.
6. Terapkan transformasi yang tersisa.
7. Buat pipeline inferensi serial menggunakan notebook Jupyter yang kami sediakan.

Untuk informasi tentang mengekspor data ke bucket Amazon S3, lihat. Ekspor ke Amazon S3 Untuk informasi tentang membuka notebook Jupyter yang digunakan untuk membuat pipeline inferensi serial, lihat. Menggunakan Notebook Jupyter untuk membuat titik akhir inferensi

Data Wrangler mengabaikan transformasi yang menghapus data pada saat inferensi. Misalnya, Data Wrangler mengabaikan Menangani Nilai yang Hilang transformasi jika Anda menggunakan konfigurasi Drop missing.

Jika Anda telah mereparasi transformasi ke seluruh kumpulan data Anda, transformasi terbawa ke saluran inferensi Anda. Misalnya, jika Anda menggunakan nilai median untuk mengimputasi nilai yang hilang, nilai median dari refitting transformasi diterapkan ke permintaan inferensi Anda. Anda dapat mereparasi transformasi dari alur Data Wrangler saat menggunakan notebook Jupyter atau saat mengekspor data ke pipeline inferensi. Untuk informasi tentang memperbaiki transformasi, lihat. Reparasi Transformasi ke Seluruh Dataset dan Ekspor Mereka

Pipa inferensi serial mendukung tipe data berikut untuk string input dan output. Setiap tipe data memiliki seperangkat persyaratan.

Tipe data yang didukung

- `text/csv`— tipe data untuk string CSV
  - String tidak dapat memiliki header.
  - Fitur yang digunakan untuk pipa inferensi harus dalam urutan yang sama dengan fitur dalam kumpulan data pelatihan.
  - Harus ada pembatas koma di antara fitur.
  - Catatan harus dibatasi oleh karakter baris baru.

  Berikut ini adalah contoh string CSV yang diformat secara valid yang dapat Anda berikan dalam permintaan inferensi.

```
abc,0.0,"Doe, John",12345\ndef,1.1,"Doe, Jane",67890
```

- `application/json`— tipe data untuk string JSON
  - Fitur yang digunakan dalam kumpulan data untuk pipa inferensi harus dalam urutan yang sama dengan fitur dalam kumpulan data pelatihan.
  - Data harus memiliki skema tertentu. Anda mendefinisikan skema sebagai `instances` objek tunggal yang memiliki satu set. `features` Setiap `features` objek mewakili pengamatan.

  Berikut ini adalah contoh string JSON yang diformat secara valid yang dapat Anda berikan dalam permintaan inferensi.

```
{
    "instances": [
        {
            "features": ["abc", 0.0, "Doe, John", 12345]
        },
        {
            "features": ["def", 1.1, "Doe, Jane", 67890]
        }
    ]
}
```

Menggunakan Notebook Jupyter untuk membuat titik akhir inferensi

Gunakan prosedur berikut untuk mengekspor alur Data Wrangler Anda untuk membuat pipeline inferensi.

Untuk membuat pipeline inferensi menggunakan notebook Jupyter, lakukan hal berikut.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor ke.
3. Pilih SageMaker Inference Pipeline (melalui Jupyter Notebook).
4. Jalankan notebook Jupyter.

Saat Anda menjalankan notebook Jupyter, itu menciptakan artefak aliran inferensi. Artefak aliran inferensi adalah file aliran Data Wrangler dengan metadata tambahan yang digunakan untuk membuat pipeline inferensi serial. Node yang Anda ekspor mencakup semua transformasi dari node sebelumnya.

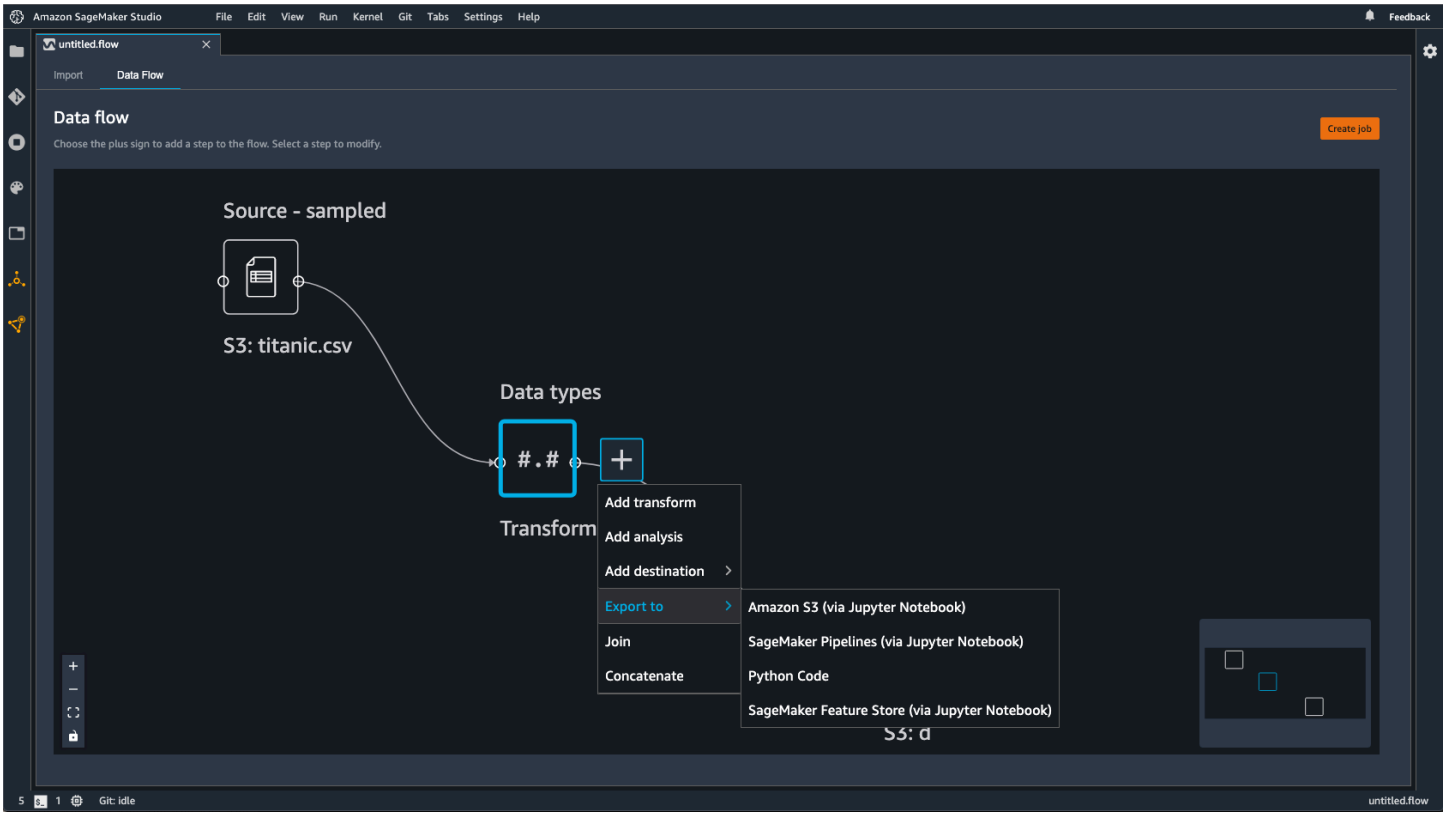
> **Important**
>
> Data Wrangler membutuhkan artefak aliran inferensi untuk menjalankan pipa inferensi. Anda tidak dapat menggunakan file aliran Anda sendiri sebagai artefak. Anda harus membuatnya dengan menggunakan prosedur sebelumnya.

Ekspor ke Kode Python

Untuk mengekspor semua langkah dalam aliran data Anda ke file Python yang dapat Anda integrasikan secara manual ke dalam alur kerja pemrosesan data apa pun, gunakan prosedur berikut.

Gunakan prosedur berikut untuk menghasilkan notebook Jupyter dan menjalankannya untuk mengekspor aliran Data Wrangler Anda ke Kode Python.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor ke.
3. Pilih Kode Python.
4. Jalankan notebook Jupyter.



Anda mungkin perlu mengonfigurasi skrip Python untuk membuatnya berjalan di pipeline Anda. Misalnya, jika Anda menjalankan lingkungan Spark, pastikan Anda menjalankan skrip dari lingkungan yang memiliki izin untuk mengakses AWS sumber daya.

Ekspor ke Toko SageMaker Fitur Amazon

Anda dapat menggunakan Data Wrangler untuk mengekspor fitur yang telah Anda buat ke Amazon SageMaker Feature Store. Fitur adalah kolom dalam dataset Anda. Feature Store adalah toko terpusat untuk fitur dan metadata terkait. Anda dapat menggunakan Feature Store untuk membuat, berbagi, dan mengelola data yang dikurasi untuk pengembangan machine learning (ML). Toko terpusat membuat data Anda lebih mudah ditemukan dan dapat digunakan kembali. Untuk informasi selengkapnya tentang Toko Fitur, lihat Toko SageMaker Fitur Amazon.

Konsep inti di Feature Store adalah grup fitur. Grup fitur adalah kumpulan fitur, catatan mereka (pengamatan), dan metadata terkait. Ini mirip dengan tabel dalam database.

Anda dapat menggunakan Data Wrangler untuk melakukan salah satu hal berikut:

- Perbarui grup fitur yang ada dengan catatan baru. Catatan adalah pengamatan dalam dataset.

- Buat grup fitur baru dari node dalam alur Data Wrangler Anda. Data Wrangler menambahkan pengamatan dari kumpulan data Anda sebagai catatan dalam grup fitur Anda.

Jika Anda memperbarui grup fitur yang ada, skema kumpulan data Anda harus cocok dengan skema grup fitur. Semua catatan dalam grup fitur diganti dengan pengamatan di kumpulan data Anda.

Anda dapat menggunakan notebook Jupyter atau node tujuan untuk memperbarui grup fitur Anda dengan pengamatan dalam kumpulan data.

Jika grup fitur Anda dengan format tabel Iceberg memiliki kunci enkripsi toko offline khusus, pastikan Anda memberikan IAM yang Anda gunakan untuk izin pekerjaan Amazon SageMaker Processing untuk menggunakannya. Minimal, Anda harus memberikan izin untuk mengenkripsi data yang Anda tulis ke Amazon S3. Untuk memberikan izin, berikan peran IAM kemampuan untuk menggunakan. GenerateDataKey Untuk informasi selengkapnya tentang pemberian izin IAM role untuk menggunakan kunci, lihat AWS KMS https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html

Destination Node

Jika Anda ingin menampilkan serangkaian langkah pemrosesan data yang telah Anda lakukan ke grup fitur, Anda dapat membuat simpul tujuan. Saat Anda membuat dan menjalankan node tujuan, Data Wrangler memperbarui grup fitur dengan data Anda. Anda juga dapat membuat grup fitur baru dari UI simpul tujuan. Setelah Anda membuat node tujuan, Anda membuat pekerjaan pemrosesan untuk menampilkan data. Pekerjaan pemrosesan adalah pekerjaan SageMaker pemrosesan Amazon. Saat Anda menggunakan node tujuan, ia menjalankan sumber daya komputasi yang diperlukan untuk menampilkan data yang telah Anda ubah ke grup fitur.

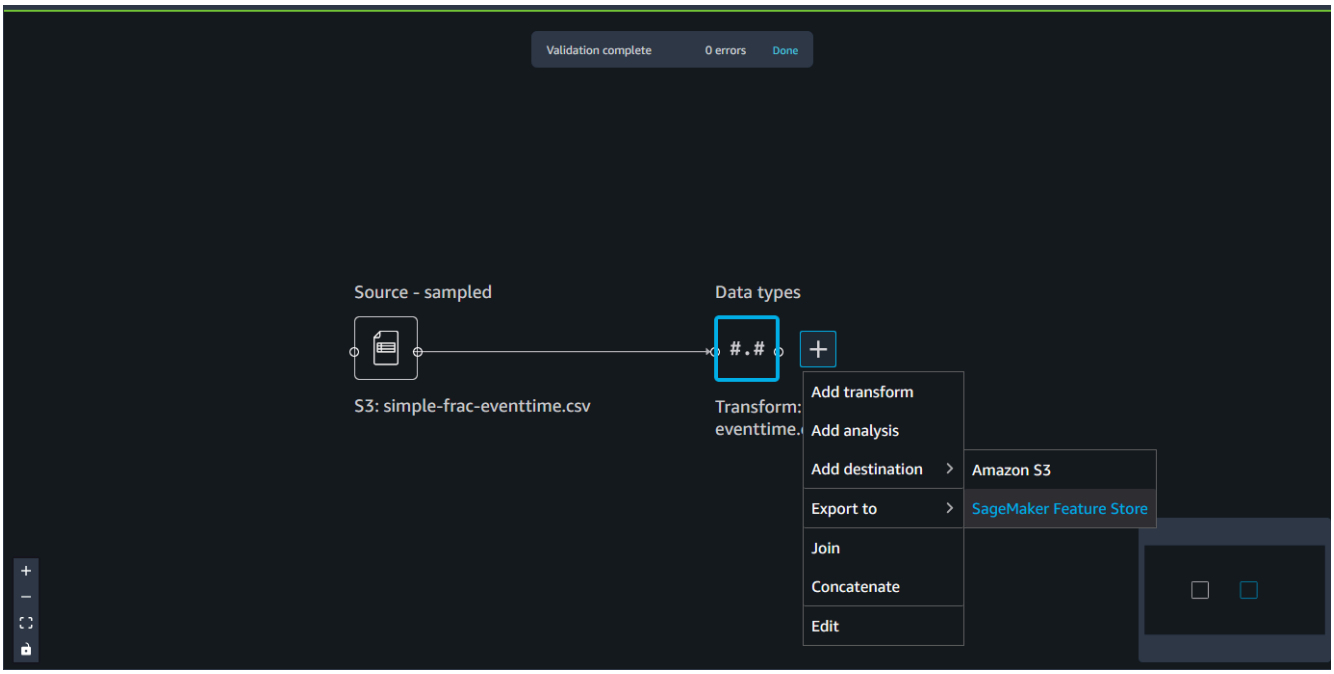
Anda dapat menggunakan node tujuan untuk mengekspor beberapa transformasi atau semua transformasi yang telah Anda buat dalam alur Data Wrangler Anda.

Gunakan prosedur berikut untuk membuat node tujuan untuk memperbarui grup fitur dengan pengamatan dari kumpulan data Anda.

Untuk memperbarui grup fitur menggunakan node tujuan, lakukan hal berikut.

> **Note**
>
> Anda dapat memilih Buat pekerjaan di alur Data Wrangler untuk melihat petunjuk penggunaan pekerjaan pemrosesan untuk memperbarui grup fitur.

1. Pilih simbol + di sebelah node yang berisi kumpulan data yang ingin Anda ekspor.
2. Di bawah Tambahkan tujuan, pilih Toko SageMaker Fitur.



3. Pilih (klik dua kali) grup fitur. Data Wrangler memeriksa apakah skema grup fitur cocok dengan skema data yang Anda gunakan untuk memperbarui grup fitur.
4. (Opsional) Pilih Ekspor ke toko offline hanya untuk grup fitur yang memiliki toko online dan toko offline. Opsi ini hanya memperbarui toko offline dengan pengamatan dari kumpulan data Anda.
5. Setelah Data Wrangler memvalidasi skema kumpulan data Anda, pilih Tambah.

Gunakan prosedur berikut untuk membuat grup fitur baru dengan data dari kumpulan data Anda.

Anda dapat menyimpan grup fitur Anda dengan salah satu cara berikut:

- Online — Latensi rendah, cache ketersediaan tinggi untuk grup fitur yang menyediakan pencarian catatan secara real-time. Toko online memungkinkan akses cepat ke nilai terbaru untuk catatan dalam grup fitur.
- Offline — Menyimpan data untuk grup fitur Anda di bucket Amazon S3. Anda dapat menyimpan data secara offline saat Anda tidak memerlukan pembacaan latensi rendah (sub-detik). Anda dapat menggunakan toko offline untuk fitur yang digunakan dalam eksplorasi data, pelatihan model, dan inferensi batch.
- Baik online maupun offline — Menyimpan data Anda di toko online dan toko offline.

Untuk membuat grup fitur menggunakan node tujuan, lakukan hal berikut.

1. Pilih simbol + di sebelah node yang berisi kumpulan data yang ingin Anda ekspor.
2. Di bawah Tambahkan tujuan, pilih Toko SageMaker Fitur.
3. Pilih Buat Grup Fitur.
4. Di kotak dialog berikut, jika kumpulan data Anda tidak memiliki kolom waktu acara, pilih Buat kolom “EventTime”.
5. Pilih Berikutnya.
6. Pilih Salin Skema JSON. Saat Anda membuat grup fitur, Anda menempelkan skema ke dalam definisi fitur.
7. Pilih Buat.
8. Untuk nama grup Fitur, tentukan nama untuk grup fitur Anda.
9. Untuk Deskripsi (opsional), tentukan deskripsi untuk membuat grup fitur Anda lebih mudah ditemukan.
10. Untuk membuat grup fitur untuk toko online, lakukan hal berikut.

    a. Pilih Aktifkan penyimpanan online.

    b. Untuk kunci enkripsi toko online, tentukan kunci enkripsi AWS terkelola atau kunci enkripsi Anda sendiri.
11. Untuk membuat grup fitur untuk toko offline, lakukan hal berikut.

    a. Pilih Aktifkan penyimpanan offline. Tentukan nilai untuk bidang berikut:

    - Nama bucket S3 — Nama bucket Amazon S3 yang menyimpan grup fitur.
    - (Opsional) Nama direktori Dataset — Awalan Amazon S3 yang Anda gunakan untuk menyimpan grup fitur.
    - IAM Role ARN — Peran IAM yang memiliki akses ke Feature Store.
    - Format Tabel - Format tabel toko offline Anda. Anda dapat menentukan Glue atau Iceberg. Glue adalah format default.
    - Kunci enkripsi toko offline — Secara default, Toko Fitur menggunakan kunci AWS Key Management Service terkelola, tetapi Anda dapat menggunakan bidang untuk menentukan kunci Anda sendiri.

    b. Tentukan nilai untuk bidang berikut:

    - Nama bucket S3 — Nama bucket yang menyimpan grup fitur.

- (Opsional) Nama direktori Dataset — Awalan Amazon S3 yang Anda gunakan untuk menyimpan grup fitur.
- IAM Role ARN — Peran IAM yang memiliki akses ke feature store.
- Kunci enkripsi toko offline — Secara default, Toko Fitur menggunakan kunci AWS terkelola, tetapi Anda dapat menggunakan bidang untuk menentukan kunci Anda sendiri.

12. Pilih Lanjutkan.
13. PilihJSON.
14. Lepaskan tanda kurung placeholder di jendela.
15. Tempel teks JSON dari Langkah 6.
16. Pilih Lanjutkan.
17. Untuk RECORD IDENTIFIER FEATURE NAME, pilih kolom di dataset Anda yang memiliki pengidentifikasi unik untuk setiap record dalam dataset Anda.
18. Untuk NAMA FITUR WAKTU ACARA, pilih kolom dengan nilai stempel waktu.
19. Pilih Lanjutkan.
20. (Opsional) Tambahkan tag untuk membuat grup fitur Anda lebih mudah ditemukan.
21. Pilih Lanjutkan.
22. Pilih Buat grup fitur.
23. Arahkan kembali ke alur Data Wrangler Anda dan pilih ikon penyegaran di sebelah bilah pencarian Grup Fitur.

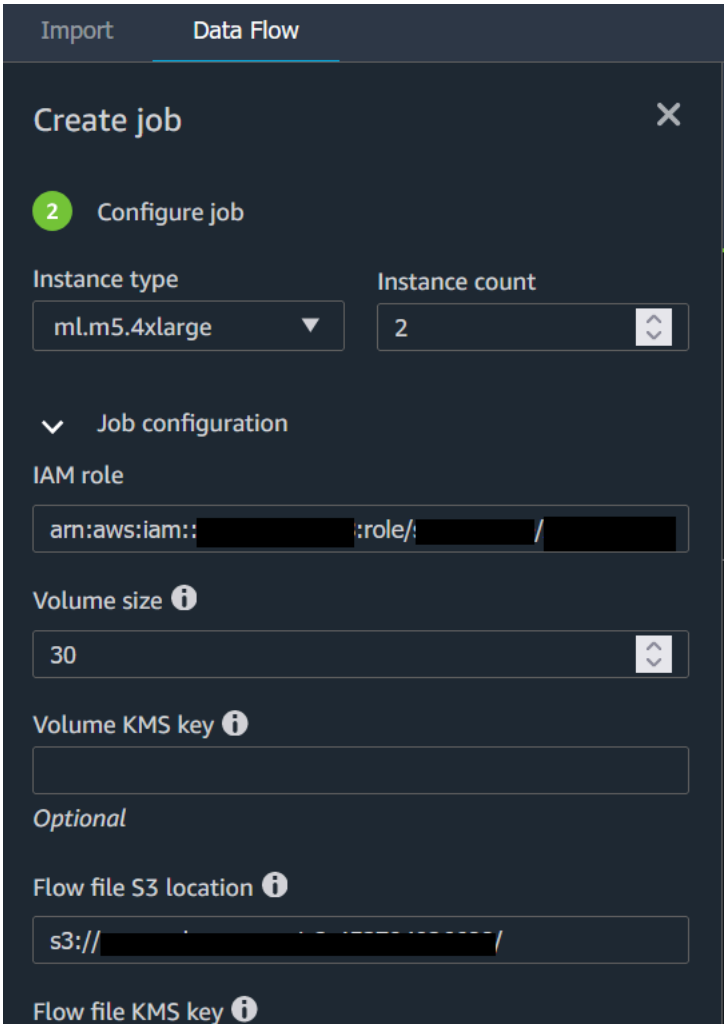
> **Note**
>
> Jika Anda telah membuat node tujuan untuk grup fitur dalam alur, Anda tidak dapat membuat node tujuan lain untuk grup fitur yang sama. Jika Anda ingin membuat node tujuan lain untuk grup fitur yang sama, Anda harus membuat file aliran lain.

Gunakan prosedur berikut untuk membuat pekerjaan Data Wrangler.

Buat pekerjaan dari halaman aliran Data dan pilih node tujuan yang ingin Anda ekspor.

1. Pilih Buat tugas. Gambar berikut menunjukkan panel yang muncul setelah Anda memilih Buat pekerjaan.

2. Untuk nama Job, tentukan nama pekerjaan ekspor.
3. Pilih simpul tujuan yang ingin diekspor.
4. (Opsional) Untuk Output KMS Key, tentukan ARN, ID, atau alias kunci. AWS KMS Kunci KMS adalah kunci kriptografi. Anda dapat menggunakan kunci untuk mengenkripsi data output dari pekerjaan. Untuk informasi selengkapnya tentang AWS KMS kunci, lihat AWS Key Management Service.
5. Gambar berikut menunjukkan halaman Configure job dengan tab konfigurasi Job terbuka.



(Opsional) Di bawah parameter Terlatih. pilih Reparasi jika Anda telah melakukan hal berikut:

- Sumber data per set data
- Menerapkan transformasi yang menggunakan data Anda untuk membuat kolom baru dalam kumpulan data

Untuk informasi selengkapnya tentang memperbaiki transformasi yang telah Anda buat ke seluruh kumpulan data, lihat. Reparasi Transformasi ke Seluruh Dataset dan Ekspor Mereka

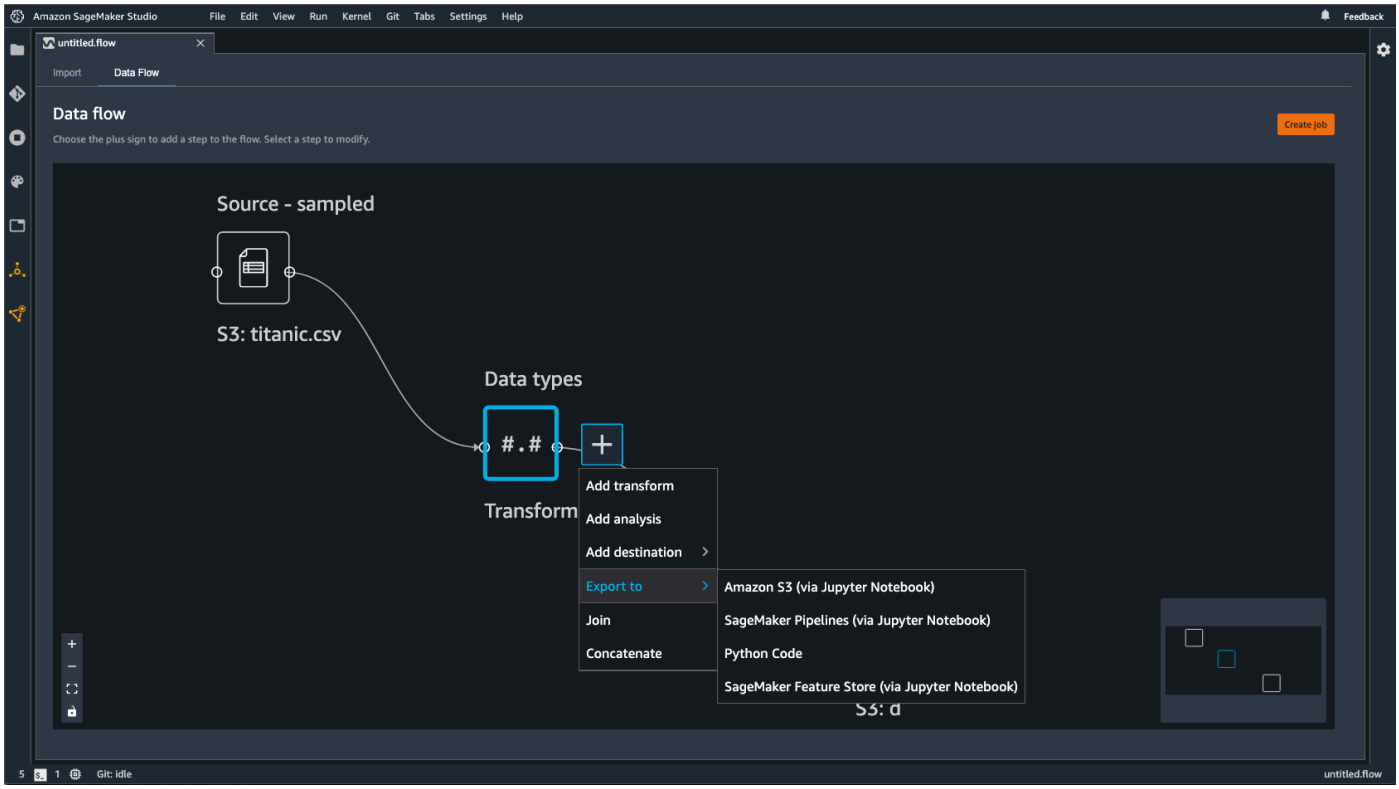
6. Pilih Konfigurasi pekerjaan.
7. (Opsional) Konfigurasikan pekerjaan Data Wrangler. Anda dapat membuat konfigurasi berikut:
   - Konfigurasi Job
   - Konfigurasi memori percikan
   - Konfigurasi jaringan
   - Tanda
   - Parameter
   - Jadwal Asosiasi
8. Pilih Jalankan.

Jupyter notebook

Gunakan prosedur berikut ke notebook Jupyter untuk mengekspor ke Amazon SageMaker Feature Store.

Gunakan prosedur berikut untuk membuat notebook Jupyter dan menjalankannya untuk mengekspor aliran Data Wrangler Anda ke Feature Store.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor ke.
3. Pilih Amazon SageMaker Feature Store (melalui Jupyter Notebook).
4. Jalankan notebook Jupyter.



Menjalankan notebook Jupyter menjalankan pekerjaan Data Wrangler. Menjalankan pekerjaan Data Wrangler memulai pekerjaan SageMaker pemrosesan. Pekerjaan pemrosesan menyerap aliran ke feature store online dan offline.

> **Important**
>
> Peran IAM yang Anda gunakan untuk menjalankan buku catatan ini harus memiliki kebijakan AWS terkelola berikut yang dilampirkan: `AmazonSageMakerFullAccess` dan`AmazonSageMakerFeatureStoreAccess`.

Anda hanya perlu mengaktifkan satu feature store online atau offline saat membuat grup fitur. Anda juga dapat mengaktifkan keduanya. Untuk menonaktifkan pembuatan toko online, atur `EnableOnlineStore` ke`False`:

```
# Online Store Configuration
online_store_config = {
    "EnableOnlineStore": False
}
```

Notebook menggunakan nama kolom dan jenis kerangka data yang Anda ekspor untuk membuat skema grup fitur, yang digunakan untuk membuat grup fitur. Grup fitur adalah sekelompok fitur yang ditentukan di feature store untuk mendeskripsikan rekaman. Grup fitur mendefinisikan skema dan fitur yang terkandung dalam grup fitur. Definisi grup fitur terdiri dari daftar fitur, nama fitur pengenal catatan, nama fitur waktu acara, dan konfigurasi untuk toko online dan toko offline.

Setiap fitur dalam grup fitur dapat memiliki salah satu dari jenis berikut: String, Fractional, atau Integral. Jika kolom dalam kerangka data yang diekspor Anda bukan salah satu dari jenis ini, itu defaultnya. `String`

Berikut ini adalah contoh skema grup fitur.

```
column_schema = [
    {
        "name": "Height",
        "type": "long"
    },
    {
        "name": "Input",
        "type": "string"
    },
    {
        "name": "Output",
        "type": "string"
    },
    {
        "name": "Sum",
        "type": "string"
    },
    {
        "name": "Time",
        "type": "string"
    }
]
```

Selain itu, Anda harus menentukan nama pengenal catatan dan nama fitur waktu acara:

- Nama pengenal rekaman adalah nama fitur yang nilainya secara unik mengidentifikasi catatan yang ditentukan di feature store. Hanya catatan terbaru per nilai pengenal yang disimpan di toko online. Nama fitur pengenal catatan harus menjadi salah satu nama definisi fitur.

- Nama fitur waktu acara adalah nama fitur yang `EventTime` menyimpan catatan dalam grup fitur. An `EventTime` adalah titik waktu ketika peristiwa baru terjadi yang sesuai dengan pembuatan atau pembaruan catatan dalam suatu fitur. Semua catatan dalam grup fitur harus memiliki yang sesuai`EventTime`.

  Notebook menggunakan konfigurasi ini untuk membuat grup fitur, memproses data Anda dalam skala besar, dan kemudian memasukkan data yang diproses ke toko fitur online dan offline Anda. Untuk mempelajari lebih lanjut, lihat Sumber Data dan Penyerapan.

Notebook menggunakan konfigurasi ini untuk membuat grup fitur, memproses data Anda dalam skala besar, dan kemudian memasukkan data yang diproses ke toko fitur online dan offline Anda. Untuk mempelajari lebih lanjut, lihat Sumber Data dan Penyerapan.

Reparasi Transformasi ke Seluruh Dataset dan Ekspor Mereka

Saat Anda mengimpor data, Data Wrangler menggunakan sampel data untuk menerapkan pengkodean. Secara default, Data Wrangler menggunakan 50.000 baris pertama sebagai sampel, tetapi Anda dapat mengimpor seluruh kumpulan data atau menggunakan metode pengambilan sampel yang berbeda. Untuk informasi selengkapnya, lihat Impor.

Transformasi berikut menggunakan data Anda untuk membuat kolom dalam kumpulan data:

- Encode Kategoris
- Ikhtisar
- Tangani Outlier
- Menangani Nilai yang Hilang

Jika Anda menggunakan sampling untuk mengimpor data Anda, transformasi sebelumnya hanya menggunakan data dari sampel untuk membuat kolom. Transformasi mungkin tidak menggunakan semua data yang relevan. Misalnya, jika Anda menggunakan transformasi Encode Categorical, mungkin ada kategori di seluruh kumpulan data yang tidak ada dalam sampel.

Anda dapat menggunakan node tujuan atau notebook Jupyter untuk mereparasi transformasi ke seluruh kumpulan data. Ketika Data Wrangler mengekspor transformasi dalam aliran, itu menciptakan pekerjaan pemrosesan. SageMaker Saat pekerjaan pemrosesan selesai, Data Wrangler menyimpan file berikut di lokasi Amazon S3 default atau lokasi S3 yang Anda tentukan:

- File aliran Data Wrangler yang menentukan transformasi yang direparasi ke kumpulan data
- Dataset dengan transformasi reparasi diterapkan padanya

Anda dapat membuka file aliran Data Wrangler dalam Data Wrangler dan menerapkan transformasi ke kumpulan data yang berbeda. Misalnya, jika Anda telah menerapkan transformasi ke kumpulan data pelatihan, Anda dapat membuka dan menggunakan file aliran Data Wrangler untuk menerapkan transformasi ke kumpulan data yang digunakan untuk inferensi.

Untuk informasi tentang penggunaan node tujuan untuk mereparasi transformasi dan ekspor, lihat halaman berikut:

- Ekspor ke Amazon S3
- Ekspor ke Toko SageMaker Fitur Amazon

Gunakan prosedur berikut untuk menjalankan notebook Jupyter untuk mereparasi transformasi dan mengekspor data.

Untuk menjalankan notebook Jupyter dan untuk mereparasi transformasi dan mengekspor alur Data Wrangler Anda, lakukan hal berikut.

1. Pilih + di sebelah simpul yang ingin Anda ekspor.
2. Pilih Ekspor ke.
3. Pilih lokasi tempat Anda mengekspor data.
4. Untuk `refit_trained_params` objek, atur `refit` ke`True`.
5. Untuk `output_flow` bidang, tentukan nama file aliran output dengan transformasi reparasi.
6. Jalankan notebook Jupyter.

Buat Jadwal untuk Memproses Data Baru Secara Otomatis

Jika Anda memproses data secara berkala, Anda dapat membuat jadwal untuk menjalankan pekerjaan pemrosesan secara otomatis. Misalnya, Anda dapat membuat jadwal yang menjalankan pekerjaan pemrosesan secara otomatis saat Anda mendapatkan data baru. Untuk informasi selengkapnya tentang memproses pekerjaan, lihat Ekspor ke Amazon S3 danEkspor ke Toko SageMaker Fitur Amazon.

Saat Anda membuat pekerjaan, Anda harus menentukan peran IAM yang memiliki izin untuk membuat pekerjaan. Secara default, peran IAM yang Anda gunakan untuk mengakses Data Wrangler adalah. `SageMakerExecutionRole`

Izin berikut memungkinkan Data Wrangler mengakses EventBridge dan memungkinkan EventBridge untuk menjalankan pekerjaan pemrosesan:

- Tambahkan kebijakan AWS Terkelola berikut ke peran eksekusi Amazon SageMaker Studio Classic yang memberikan izin kepada Data Wrangler untuk digunakan: EventBridge

  ```
  arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess
  ```

  Untuk informasi selengkapnya tentang kebijakan tersebut, lihat kebijakan AWS terkelola EventBridge.

- Tambahkan kebijakan berikut ke IAM role yang Anda tentukan saat Anda membuat tugas di Data Wrangler:

  ```
  {
      "Version": "2012-10-17",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": "sagemaker:StartPipelineExecution",
              "Resource": "arn:aws:sagemaker:Region:AWS-account-id:pipeline/data-
  wrangler-*"
          }
      ]
  }
  ```

  Jika Anda menggunakan peran IAM default, Anda menambahkan kebijakan sebelumnya ke peran eksekusi Amazon SageMaker Studio Classic.

  Tambahkan kebijakan kepercayaan berikut ke peran untuk memungkinkan untuk EventBridge mengasumsikannya.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
```

Important

Saat Anda membuat jadwal, Data Wrangler membuat `eventRule` in. EventBridge Anda dikenakan biaya untuk aturan acara yang Anda buat dan instance yang digunakan untuk menjalankan pekerjaan pemrosesan.
Untuk informasi tentang EventBridge harga, lihat EventBridge harga Amazon. Untuk informasi tentang memproses harga lowongan kerja, lihat SageMaker Harga Amazon.

Anda dapat mengatur jadwal menggunakan salah satu metode berikut:

- Ekspresi CRON

  Note

  Data Wrangler tidak mendukung ekspresi berikut:

  - LW#
  - Singkatan untuk hari
  - Singkatan selama berbulan-bulan

- Ekspresi RATE
- Berulang — Tetapkan interval per jam atau harian untuk menjalankan pekerjaan.
- Waktu spesifik - Tetapkan hari dan waktu tertentu untuk menjalankan pekerjaan.

Bagian berikut memberikan prosedur tentang menciptakan lapangan kerja.

CRON

Gunakan prosedur berikut ini untuk membuat jadwal dengan ekspresi CRON.

Untuk menentukan jadwal dengan ekspresi CRON, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya, 2. Konfigurasikan pekerjaan.
5. Pilih Jadwal Rekanan.
6. Pilih Membuat sebuah jadwal baru.
7. Untuk Nama Jadwal, tentukan nama jadwal.
8. Untuk Run Frequency, pilih CRON.
9. Tentukan ekspresi CRON yang valid.
10. Pilih Buat.
11. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

    > **Note**
    >
    > Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

12. Pilih salah satu cara berikut:

    - Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
    - Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.

13. Pilih Jalankan

RATE

Gunakan prosedur berikut ini untuk membuat jadwal dengan ekspresi RATE.

Untuk menentukan jadwal dengan ekspresi RATE, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya, 2. Konfigurasikan pekerjaan.
5. Pilih Jadwal Rekanan.
6. Pilih Membuat sebuah jadwal baru.
7. Untuk Nama Jadwal, tentukan nama jadwal.
8. Untuk Run Frequency, pilih Rate.
9. Untuk Nilai, tentukan bilangan bulat.
10. Untuk Unit, pilih salah satu dari berikut ini:
    - Menit
    - Jam
    - Hari
11. Pilih Buat.
12. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

    > **Note**
    >
    > Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

13. Pilih salah satu cara berikut:
    - Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
    - Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.
14. Pilih Jalankan

Recurring

Gunakan prosedur berikut untuk membuat jadwal yang menjalankan pekerjaan secara berulang.

Untuk menentukan jadwal dengan ekspresi CRON, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya, 2. Konfigurasikan pekerjaan.
5. Pilih Jadwal Rekanan.
6. Pilih Membuat sebuah jadwal baru.
7. Untuk Nama Jadwal, tentukan nama jadwal.
8. Untuk Run Frequency, pastikan Recurring dipilih secara default.
9. Untuk Setiap x jam, tentukan frekuensi per jam yang dijalankan pekerjaan pada siang hari. Nilai yang valid adalah bilangan bulat dalam rentang inklusif dan**1**. **23**
10. Untuk Pada hari, pilih salah satu opsi berikut:

    - Setiap hari
    - Akhir pekan
    - Hari kerja
    - Pilih Hari

    - (Opsional) Jika Anda telah memilih Pilih Hari, pilih hari dalam seminggu untuk menjalankan pekerjaan.

    > **Note**
    >
    > Jadwal diatur ulang setiap hari. Jika Anda menjadwalkan pekerjaan untuk dijalankan setiap lima jam, itu berjalan pada waktu-waktu berikut di siang hari:
    >
    > - 00:00
    > - 05:00
    > - 10:00
    > - 15:00
    > - 20:00

11. Pilih Buat.

12. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

> **Note**
>
> Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

13. Pilih salah satu cara berikut:

    - Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
    - Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.

14. Pilih Jalankan

Specific time

Gunakan prosedur berikut untuk membuat jadwal yang menjalankan tugas pada waktu tertentu.

Untuk menentukan jadwal dengan ekspresi CRON, lakukan hal berikut.

1. Buka alur Data Wrangler Anda.
2. Pilih Buat tugas.
3. (Opsional) Untuk tombol Output KMS, tentukan AWS KMS kunci untuk mengkonfigurasi output pekerjaan.
4. Pilih Berikutnya, 2. Konfigurasikan pekerjaan.
5. Pilih Jadwal Rekanan.
6. Pilih Membuat sebuah jadwal baru.
7. Untuk Nama Jadwal, tentukan nama jadwal.
8. Pilih Buat.
9. (Opsional) Pilih Tambahkan jadwal lain untuk menjalankan pekerjaan pada jadwal tambahan.

> **Note**
>
> Anda dapat mengaitkan maksimal dua jadwal. Jadwal independen dan tidak mempengaruhi satu sama lain kecuali waktu tumpang tindih.

10. Pilih salah satu cara berikut:

- Jadwalkan dan jalankan sekarang — Data Wrangler pekerjaan berjalan segera dan kemudian berjalan sesuai jadwal.
- Jadwal saja — Data Wrangler pekerjaan hanya berjalan pada jadwal yang Anda tentukan.

11. Pilih Jalankan

Anda dapat menggunakan Amazon SageMaker Studio Classic melihat pekerjaan yang dijadwalkan untuk dijalankan. Pekerjaan pemrosesan Anda berjalan di dalam SageMaker Pipelines. Setiap pekerjaan pemrosesan memiliki pipa sendiri. Ini berjalan sebagai langkah pemrosesan di dalam pipa. Anda dapat melihat jadwal yang telah Anda buat dalam pipeline. Untuk informasi tentang melihat pipeline, lihatMembuat Alur.

Gunakan prosedur berikut untuk melihat pekerjaan yang telah Anda jadwalkan.

Untuk melihat pekerjaan yang telah Anda jadwalkan, lakukan hal berikut.

1. Buka Amazon SageMaker Studio Classic.
2. Buka SageMaker Pipa
3. Lihat alur untuk tugas yang telah Anda buat.

   Pipeline yang menjalankan pekerjaan menggunakan nama pekerjaan sebagai awalan. Misalnya, jika Anda telah membuat pekerjaan bernama`housing-data-feature-enginnering`, nama pipeline adalah`data-wrangler-housing-data-feature-engineering`.
4. Pilih pipeline yang berisi pekerjaan Anda.
5. Lihat status jaringan pipa. Pipelines dengan Status Sukses telah menjalankan pekerjaan pemrosesan dengan sukses.

Untuk menghentikan pekerjaan pemrosesan agar tidak berjalan, lakukan hal berikut:

Untuk menghentikan pekerjaan pemrosesan agar tidak berjalan, hapus aturan acara yang menentukan jadwal. Menghapus aturan acara menghentikan semua pekerjaan yang terkait dengan jadwal berjalan. Untuk informasi tentang menghapus aturan, lihat Menonaktifkan atau menghapus aturan Amazon. EventBridge

Anda dapat menghentikan dan menghapus saluran pipa yang terkait dengan jadwal juga. Untuk informasi tentang menghentikan pipa, lihat StopPipelineExecution. Untuk informasi tentang menghapus pipeline, lihat DeletePipeline.

Menggunakan Widget Persiapan Data Interaktif di Notebook Amazon SageMaker Studio Classic untuk Mendapatkan Wawasan Data

Gunakan widget persiapan data Wrangler Data untuk berinteraksi dengan data Anda, mendapatkan visualisasi, menjelajahi wawasan yang dapat ditindaklanjuti, dan memperbaiki masalah kualitas data.

Anda dapat mengakses widget persiapan data dari notebook Amazon SageMaker Studio Classic. Untuk setiap kolom, widget membuat visualisasi yang membantu Anda lebih memahami distribusinya. Jika kolom memiliki masalah kualitas data, peringatan muncul di tajuknya.

Untuk melihat masalah kualitas data, pilih header kolom yang menunjukkan peringatan. Anda dapat menggunakan informasi yang Anda dapatkan dari wawasan dan visualisasi untuk menerapkan transformasi bawaan widget untuk membantu Anda memperbaiki masalah.

Misalnya, widget mungkin mendeteksi bahwa Anda memiliki kolom yang hanya memiliki satu nilai unik dan menunjukkan peringatan kepada Anda. Peringatan memberikan opsi untuk menjatuhkan kolom dari kumpulan data.

Memulai dengan menjalankan widget

Gunakan informasi berikut ini untuk membantu Anda memulai menjalankan buku catatan.

Buka buku catatan di Amazon SageMaker Studio Classic. Untuk informasi tentang membuka buku catatan, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic.

Important

Untuk menjalankan widget, notebook harus menggunakan salah satu gambar berikut:

- Python 3 (Ilmu Data) dengan Python 3.7
- Python 3 (Ilmu Data 2.0) dengan Python 3.8
- Python 3 (Ilmu Data 3.0) dengan Python 3.10
- SparkAnalytics Sumber data yang lebih tinggi
- SparkAnalytics 2.0

Untuk informasi selengkapnya tentang gambar, lihat SageMaker Gambar Amazon yang Tersedia.

Gunakan kode berikut untuk mengimpor widget persiapan data dan panda. Widget menggunakan kerangka data panda untuk menganalisis data Anda.

```
import pandas as pd
import sagemaker_datawrangler
```

Contoh kode berikut memuat file ke dalam kerangka data yang disebut. `df`

```
df = pd.read_csv("example-dataset.csv")
```

Anda dapat menggunakan kumpulan data dalam format apa pun yang dapat Anda muat sebagai objek kerangka data panda. Untuk informasi selengkapnya tentang format panda, lihat Alat IO (teks, CSV, HDF5,...).

Sel berikut menjalankan `df` variabel untuk memulai widget.

```
df
```

Bagian atas kerangka data memiliki opsi berikut:

- Lihat tabel Pandas - Beralih antara visualisasi interaktif dan tabel panda.
- Gunakan semua baris dalam kumpulan data Anda untuk menghitung wawasan. Menggunakan seluruh kumpulan data dapat meningkatkan waktu yang diperlukan untuk menghasilkan wawasan. — Jika Anda tidak memilih opsi, Data Wrangler menghitung wawasan untuk 10.000 baris pertama kumpulan data.

Rangka data menunjukkan 1000 baris pertama dari kumpulan data. Setiap header kolom memiliki bagan batang bertumpuk yang menunjukkan karakteristik kolom. Ini menunjukkan proporsi nilai yang valid, nilai yang tidak valid, dan nilai yang hilang. Anda dapat mengarahkan kursor ke berbagai bagian bagan batang bertumpuk untuk mendapatkan persentase yang dihitung.

Setiap kolom memiliki visualisasi di header. Berikut ini menunjukkan jenis visualisasi yang dapat dimiliki kolom:

- Kategoris - Bagan batang
- Pengerapan versi yang lebih tinggi
- Datetime - Bagan batang

- Teks - Bagan batang

Untuk setiap visualisasi, widget persiapan data menyoroti outlier berwarna oranye.

Ketika Anda memilih kolom, itu membuka panel samping. Panel samping menunjukkan tab Wawasan. Panel menyediakan hitungan untuk jenis nilai berikut:

- Nilai tidak valid - Nilai yang tipenya tidak cocok dengan tipe kolom.
- Nilai yang hilang — Nilai yang hilang, seperti `NaN` atau`None`.
- Nilai yang valid - Nilai yang tidak hilang atau tidak valid.

Untuk kolom numerik, tab Wawasan menampilkan statistik ringkasan berikut:

- Minimum — Nilai terkecil.
- Maksimum — Nilai terbesar.
- Mean — Mean dari nilai-nilai.
- Mode — Nilai yang paling sering muncul.
- Standar deviasi — Standar deviasi dari nilai-nilai.

Untuk kolom kategoris, tab Wawasan menampilkan statistik ringkasan berikut:

- Nilai unik — Jumlah nilai unik di kolom.
- Top — Nilai yang paling sering muncul.

Kolom yang memiliki ikon peringatan di header mereka memiliki masalah kualitas data. Memilih kolom membuka tab Kualitas data yang dapat Anda gunakan untuk menemukan transformasi untuk membantu Anda memperbaiki masalah. Peringatan memiliki salah satu tingkat keparahan berikut:

- Rendah — Masalah yang mungkin tidak memengaruhi analisis Anda, tetapi dapat berguna untuk diperbaiki.
- Medium — Masalah yang mungkin memengaruhi analisis Anda, tetapi kemungkinan tidak penting untuk diperbaiki.
- Tinggi - Masalah berat yang sangat kami sarankan untuk diperbaiki.

> **Note**
>
> Widget mengurutkan kolom untuk menunjukkan nilai yang memiliki masalah kualitas data di bagian atas kerangka data. Ini juga menyoroti nilai-nilai yang menyebabkan masalah. Warna penyorotan sesuai dengan tingkat keparahan.

Di bawah TRANSFORMASI YANG DISARANKAN, Anda dapat memilih transformasi untuk memperbaiki masalah kualitas data. Widget dapat menawarkan beberapa transformasi yang dapat memperbaiki masalah. Ini dapat menawarkan rekomendasi untuk transformasi yang paling cocok untuk masalah. Anda dapat memindahkan kursor Anda ke atas transformasi untuk mendapatkan informasi lebih lanjut.

Untuk menerapkan transformasi ke kumpulan data, pilih Terapkan dan ekspor kode. Transformasi memodifikasi kumpulan data dan memperbarui visualisasi dengan nilai yang dimodifikasi. Kode untuk transformasi muncul di sel notebook berikut. Jika Anda menerapkan transformasi tambahan ke kumpulan data, widget menambahkan transformasi ke sel. Anda dapat menggunakan kode yang dihasilkan widget untuk melakukan hal berikut:

- Sesuaikan agar lebih sesuai dengan kebutuhan Anda.
- Gunakan dalam alur kerja Anda sendiri.

Anda dapat mereproduksi semua transformasi yang telah Anda buat dengan menjalankan ulang semua sel di buku catatan.

Widget dapat memberikan wawasan dan peringatan untuk kolom target. Kolom target adalah kolom yang Anda coba prediksi. Gunakan prosedur berikut untuk mendapatkan wawasan kolom target.

Untuk mendapatkan wawasan kolom target, lakukan hal berikut.

1. Pilih kolom yang Anda gunakan sebagai kolom target.
2. Pilih Pilih sebagai kolom target.
3. Pilih jenis masalah. Wawasan dan peringatan widget disesuaikan dengan jenis masalah. Berikut ini adalah jenis masalahnya:
   - Klasifikasi — Kolom target memiliki data kategoris.
   - Regresi — Kolom target memiliki data numerik.
4. Pilih Jalankan.

5. (Opsional) Di bawah Wawasan Kolom Target, pilih salah satu transformasi yang disarankan.

Referensi untuk wawasan dan transformasi di widget

Untuk kolom fitur (kolom yang bukan kolom target), Anda bisa mendapatkan wawasan berikut untuk memperingatkan Anda tentang masalah dengan kumpulan data Anda.

- Nilai hilang - Kolom memiliki nilai yang hilang sepertiNone, NaN (bukan angka), atau NaT (bukan stempel waktu). Banyak algoritma pembelajaran mesin tidak mendukung nilai yang hilang dalam data input. Oleh karena itu, mengisi atau menjatuhkan baris dengan data yang hilang merupakan langkah persiapan data yang penting. Jika Anda melihat peringatan nilai yang hilang, Anda dapat menggunakan salah satu transformasi berikut untuk memperbaiki masalah.
  - Jatuhkan hilang - Menjatuhkan baris dengan nilai yang hilang. Sebaiknya jatuhkan baris saat persentase baris dengan data yang hilang kecil dan memasukkan nilai yang hilang tidak sesuai.
  - Ganti dengan nilai baru - Mengganti nilai tekstual yang hilang dengan. Other Anda dapat mengubah Other ke nilai yang berbeda dalam kode output. Mengganti nilai numerik yang hilang dengan 0.
  - Ganti dengan mean - Mengganti nilai yang hilang dengan rata-rata kolom.
  - Ganti dengan median - Mengganti nilai yang hilang dengan median kolom.
  - Jatuhkan kolom - Jatuhkan kolom dengan nilai yang hilang dari kumpulan data. Sebaiknya jatuhkan seluruh kolom ketika ada persentase baris yang tinggi dengan data yang hilang.
- Nilai hilang yang disamarkan - Kolom telah menyamarkan nilai yang hilang. Nilai hilang yang disamarkan adalah nilai yang tidak secara eksplisit dikodekan sebagai nilai yang hilang. Misalnya, alih-alih menggunakan a NaN untuk menunjukkan nilai yang hilang, nilainya bisa jadiPlaceholder. Anda dapat menggunakan salah satu transformasi berikut ini untuk menangani nilai yang hilang:
  - Jatuhkan hilang - Menjatuhkan baris dengan nilai yang hilang
  - Ganti dengan nilai baru - Mengganti nilai tekstual yang hilang dengan. Other Anda dapat mengubah Other ke nilai yang berbeda dalam kode output. Mengganti nilai numerik yang hilang dengan 0.
- Kolom konstan - Kolom hanya memiliki satu nilai. Oleh karena itu tidak memiliki kekuatan prediksi. Kami sangat menyarankan menggunakan transformasi kolom Drop untuk menjatuhkan kolom dari kumpulan data.
- Kolom ID - Kolom tidak memiliki nilai berulang. Semua nilai dalam kolom adalah unik. Mereka mungkin ID atau kunci database. Tanpa informasi tambahan, kolom tidak memiliki kekuatan

prediksi. Kami sangat menyarankan menggunakan transformasi kolom Drop untuk menjatuhkan kolom dari kumpulan data.

- Kardinalitas tinggi - Kolom memiliki persentase nilai unik yang tinggi. Kardinalitas tinggi membatasi kekuatan prediksi kolom kategoris. Periksa pentingnya kolom dalam analisis Anda dan pertimbangkan untuk menggunakan transformasi kolom Drop untuk menjatuhkannya.

Untuk kolom target, Anda bisa mendapatkan wawasan berikut untuk memperingatkan Anda tentang masalah dengan kumpulan data Anda. Anda dapat menggunakan transformasi yang disarankan yang disertakan dengan peringatan untuk memperbaiki masalah.

- Tipe data campuran dalam target (Regresi) - Ada beberapa nilai non-numerik di kolom target. Mungkin ada kesalahan entri data. Sebaiknya hapus baris yang memiliki nilai yang tidak dapat dikonversi.
- Label yang sering — Nilai tertentu di kolom target muncul lebih sering daripada yang normal dalam konteks regresi. Mungkin ada kesalahan dalam pengumpulan atau pemrosesan data. Kategori yang sering muncul mungkin menunjukkan bahwa nilai tersebut digunakan sebagai nilai default atau bahwa itu adalah placeholder untuk nilai yang hilang. Sebaiknya gunakan Ganti dengan transformasi nilai baru untuk mengganti nilai yang hilang dengan`Other`.
- Terlalu sedikit contoh per kelas - Kolom target memiliki kategori yang jarang muncul. Beberapa kategori tidak memiliki cukup baris agar kolom target berguna. Anda dapat menggunakan salah satu transformasi berikut:
  - Jatuhkan target langka — Menjatuhkan nilai unik dengan kurang dari sepuluh pengamatan. Misalnya, turunkan nilainya `cat` jika muncul sembilan kali di kolom.
  - Ganti target langka — Mengganti kategori yang jarang muncul di kumpulan data dengan nilainya. `Other`
- Kelas terlalu tidak seimbang (klasifikasi multi-kelas) — Ada kategori dalam kumpulan data yang muncul jauh lebih sering daripada kategori lainnya. Ketidakseimbangan kelas dapat mempengaruhi akurasi prediksi. Untuk prediksi yang paling akurat, kami sarankan memperbarui kumpulan data dengan baris yang memiliki kategori yang saat ini lebih jarang muncul.
- Sejumlah besar kelas/terlalu banyak kelas — Ada sejumlah besar kelas di kolom target. Memiliki banyak kelas dapat menghasilkan waktu pelatihan yang lebih lama atau kualitas prediksi yang buruk. Sebaiknya lakukan salah satu dari langkah berikut:
  - Mengelompokkan beberapa kategori ke dalam kategori mereka sendiri. Misalnya, jika enam kategori terkait erat, kami sarankan menggunakan satu kategori untuk mereka.
  - Menggunakan algoritma ML yang tahan terhadap beberapa kategori.

Keamanan dan Izin

Saat Anda melakukan kueri data dari Athena atau Amazon Redshift, kumpulan data yang ditanyakan akan disimpan secara otomatis di bucket S3 SageMaker default untuk Wilayah tempat Anda menggunakan Studio AWS Classic. Selain itu, saat Anda mengekspor Notebook Jupyter dari Amazon SageMaker Data Wrangler dan menjalankannya, aliran data, atau file.flow, disimpan ke bucket default yang sama, di bawah awalan data_wrangler_flows.

Untuk kebutuhan keamanan tingkat tinggi, Anda dapat mengonfigurasi kebijakan bucket yang membatasi AWS peran yang memiliki akses ke bucket SageMaker S3 default ini. Gunakan bagian berikut untuk menambahkan jenis kebijakan ini ke bucket S3. Untuk mengikuti petunjuk di halaman ini, gunakan AWS Command Line Interface (AWS CLI). Untuk mempelajari caranya, lihat Mengkonfigurasi AWS CLI di Panduan Pengguna IAM.

Selain itu, Anda perlu memberikan setiap peran IAM yang menggunakan izin Data Wrangler untuk mengakses sumber daya yang diperlukan. Jika Anda tidak memerlukan izin terperinci untuk peran IAM yang Anda gunakan untuk mengakses Data Wrangler, Anda dapat menambahkan kebijakan terkelola IAM, ke peran IAM yang Anda gunakan untuk membuat pengguna Studio Classic Anda. `AmazonSageMakerFullAccess` Kebijakan ini memberi Anda izin penuh untuk menggunakan Data Wrangler. Jika Anda memerlukan izin yang lebih terperinci, lihat bagian,. Berikan Izin Peran IAM untuk Menggunakan Data Wrangler

Menambahkan Kebijakan Bucket Untuk Membatasi Akses ke Kumpulan Data yang Diimpor ke Data Wrangler

Anda dapat menambahkan kebijakan ke bucket S3 yang berisi resource Data Wrangler menggunakan kebijakan bucket Amazon S3. Sumber daya yang diunggah Data Wrangler ke bucket SageMaker S3 default Anda di AWS Wilayah yang Anda gunakan Studio Classic termasuk yang berikut:

- Menanyakan hasil Amazon Redshift. Ini disimpan di bawah awalan redshift/.
- Menanyakan hasil Athena. Ini disimpan di bawah awalan athena/.
- File.flow yang diunggah ke Amazon S3 saat Anda menjalankan Jupyter Notebook Data Wrangler yang diekspor dihasilkan. Ini disimpan di bawah awalan data_wrangler_flows/.

Gunakan prosedur berikut untuk membuat kebijakan bucket S3 yang dapat Anda tambahkan untuk membatasi akses peran IAM ke bucket tersebut. Untuk mempelajari cara menambahkan kebijakan ke bucket S3, lihat Bagaimana cara menambahkan kebijakan S3 Bucket? .

Untuk menyiapkan kebijakan bucket pada bucket S3 yang menyimpan resource Data Wrangler Anda:

1. Konfigurasikan satu atau beberapa peran IAM yang Anda inginkan untuk dapat mengakses Data Wrangler.
2. Buka command prompt atau shell. Untuk setiap peran yang Anda buat, ganti *nama peran* dengan nama peran dan jalankan yang berikut ini:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Dalam tanggapan, Anda melihat `RoleId` string yang dimulai dengan`AROA`. Fungsi SQL baru
3. Tambahkan kebijakan berikut ke bucket SageMaker default di AWS Wilayah tempat Anda menggunakan Data Wrangler. Ganti *wilayah* dengan AWS Wilayah tempat bucket berada, dan id *akun dengan ID* AWS akun Anda. Ganti `userId` s dimulai dengan *AROAEXAMPLEID* dengan ID AWS peran yang ingin Anda berikan izin untuk menggunakan Data Wrangler.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Deny",
         "Principal": "*",
         "Action": "s3:*",
         "Resource": [
           "arn:aws:s3:::sagemaker-region-account-id/data_wrangler_flows/",
           "arn:aws:s3:::sagemaker-region-account-id/data_wrangler_flows/*",
           "arn:aws:s3:::sagemaker-region-account-id/athena",
           "arn:aws:s3:::sagemaker-region-account-id/athena/*",
           "arn:aws:s3:::sagemaker-region-account-id/redshift",
           "arn:aws:s3:::sagemaker-region-account-id/redshift/*"

         ],
         "Condition": {
           "StringNotLike": {
             "aws:userId": [
               "AROAEXAMPLEID_1:*",
               "AROAEXAMPLEID_2:*"
             ]
           }
         }
       }
     ]
   ```

```
}
```

Buat Daftar Izinkan untuk Data Wrangler

Setiap kali pengguna mulai menjalankan Data Wrangler dari antarmuka pengguna Amazon SageMaker Studio Classic, mereka membuat panggilan ke antarmuka pemrograman SageMaker aplikasi (API) untuk membuat aplikasi Data Wrangler.

Organisasi Anda mungkin tidak memberikan izin untuk melakukan panggilan API tersebut secara default. Untuk memberikan izin, Anda harus membuat dan melampirkan kebijakan ke peran IAM pengguna menggunakan templat kebijakan berikut: Contoh Daftar Izinkan Data Wrangler.

> **Note**
>
> Contoh kebijakan sebelumnya hanya memberi pengguna Anda akses ke aplikasi Data Wrangler.

Untuk informasi tentang membuat kebijakan, lihat Membuat kebijakan di tab JSON. Saat Anda membuat kebijakan, salin dan tempel kebijakan JSON dari Contoh Daftar Izinkan Data Wrangler di tab JSON.

> **Important**
>
> Hapus kebijakan IAM apa pun yang mencegah pengguna menjalankan operasi berikut:
>
> - CreateApp
> - DescribeApp
>
> Jika Anda tidak menghapus kebijakan, pengguna Anda masih dapat terpengaruh olehnya.

Setelah membuat kebijakan menggunakan templat, lampirkan ke peran IAM pengguna Anda. Untuk informasi tentang melampirkan kebijakan, lihat Menambahkan izin identitas IAM (konsol).

Berikan Izin Peran IAM untuk Menggunakan Data Wrangler

Anda dapat memberikan izin peran IAM untuk menggunakan Data Wrangler dengan kebijakan umum yang dikelola IAM,. `AmazonSageMakerFullAccess` Ini adalah kebijakan umum yang mencakup izin yang diperlukan untuk menggunakan semua SageMaker layanan. Kebijakan ini memberikan peran IAM akses penuh ke Data Wrangler. Anda harus mengetahui hal berikut saat menggunakan `AmazonSageMakerFullAccess` untuk memberikan akses ke Data Wrangler:

- Jika Anda mengimpor data dari Amazon Redshift, nama Pengguna Database harus memiliki awalan. `sagemaker_access`
- Kebijakan terkelola ini hanya memberikan izin untuk mengakses bucket dengan salah satu dari berikut ini dalam nama:`SageMaker`,, `SageMakersagemaker`, atau. `aws-glue` Jika ingin menggunakan Data Wrangler untuk mengimpor dari bucket S3 tanpa frasa ini dalam nama, lihat bagian terakhir di halaman ini untuk mempelajari cara memberikan izin kepada entitas IAM untuk mengakses bucket S3 Anda.

Jika Anda memiliki kebutuhan keamanan tinggi, Anda dapat melampirkan kebijakan di bagian ini ke entitas IAM untuk memberikan izin yang diperlukan untuk menggunakan Data Wrangler.

Jika Anda memiliki kumpulan data di Amazon Redshift atau Athena yang perlu diimpor oleh peran IAM dari Data Wrangler, Anda harus menambahkan kebijakan ke entitas tersebut untuk mengakses sumber daya ini. Kebijakan berikut adalah kebijakan paling ketat yang dapat Anda gunakan untuk memberikan izin peran IAM untuk mengimpor data dari Amazon Redshift dan Athena.

Untuk mempelajari cara melampirkan kebijakan kustom ke peran IAM, lihat Mengelola kebijakan IAM di Panduan Pengguna IAM.

Contoh kebijakan untuk memberikan akses ke impor dataset Athena

Kebijakan berikut mengasumsikan bahwa peran IAM memiliki izin untuk mengakses bucket S3 yang mendasari tempat data disimpan melalui kebijakan IAM terpisah.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "athena:ListDataCatalogs",
                "athena:ListDatabases",
```

```
                "athena:ListTableMetadata",
                "athena:GetQueryExecution",
                "athena:GetQueryResults",
                "athena:StartQueryExecution",
                "athena:StopQueryExecution"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateTable"
            ],
            "Resource": [
                "arn:aws:glue:*:*:table/*/sagemaker_tmp_*",
                "arn:aws:glue:*:*:table/sagemaker_featurestore/*",
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:DeleteTable"
            ],
            "Resource": [
                "arn:aws:glue:*:*:table/*/sagemaker_tmp_*",
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabases",
                "glue:GetTable",
                "glue:GetTables"
            ],
            "Resource": [
                "arn:aws:glue:*:*:table/*",
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:GetDatabase"
            ],
            "Resource": [
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/sagemaker_featurestore",
                "arn:aws:glue:*:*:database/sagemaker_processing",
                "arn:aws:glue:*:*:database/default",
                "arn:aws:glue:*:*:database/sagemaker_data_wrangler"
            ]
        }
    ]
}
```

Contoh kebijakan untuk memberikan akses ke impor dataset Amazon Redshift

Kebijakan berikut memberikan izin untuk menyiapkan sambungan Amazon Redshift ke Data Wrangler menggunakan pengguna database yang memiliki awalan dalam `sagemaker_access` nama. Untuk memberikan izin untuk terhubung menggunakan pengguna database tambahan, tambahkan entri tambahan `"Resources"` di bawah kebijakan berikut. Kebijakan berikut mengasumsikan bahwa peran IAM memiliki izin untuk mengakses bucket S3 yang mendasari tempat data disimpan melalui kebijakan IAM terpisah, jika berlaku.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift-data:ExecuteStatement",
                "redshift-data:DescribeStatement",
                "redshift-data:CancelStatement",
                "redshift-data:GetStatementResult",
                "redshift-data:ListSchemas",
                "redshift-data:ListTables"
            ],
            "Resource": [
                "*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "redshift:GetClusterCredentials"
            ],
            "Resource": [
                "arn:aws:redshift:*:*:dbuser:*/sagemaker_access*",
                "arn:aws:redshift:*:*:dbname:*"
            ]
        }
    ]
}
```

Kebijakan untuk memberikan akses ke bucket S3

Jika kumpulan data disimpan di Amazon S3, Anda dapat memberikan izin peran IAM untuk mengakses bucket ini dengan kebijakan yang serupa dengan berikut ini. *Contoh ini memberikan akses baca-tulis terprogram ke bucket bernama test.*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::test"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject"
      ],
      "Resource": ["arn:aws:s3:::test/*"]
    }
  ]
}
```

Untuk mengimpor data dari Athena dan Amazon Redshift, Anda harus memberikan izin peran IAM untuk mengakses awalan berikut di bawah bucket Amazon S3 default di Region Data Wrangler yang

AWS digunakan:,. `athena/` `redshift/` Jika bucket Amazon S3 default belum ada di AWS Wilayah, Anda juga harus memberikan izin peran IAM untuk membuat bucket di wilayah ini.

Selain itu, jika Anda ingin peran IAM dapat menggunakan opsi ekspor pekerjaan Amazon SageMaker Feature Store, SageMaker Pipelines, dan Data Wrangler, Anda harus memberikan akses ke awalan `data_wrangler_flows/` di bucket ini.

Data Wrangler menggunakan `redshift/` awalan `athena/` dan untuk menyimpan file pratinjau dan dataset yang diimpor. Untuk mempelajari selengkapnya, lihat Penyimpanan Data yang Diimpor.

Data Wrangler menggunakan `data_wrangler_flows/` awalan untuk menyimpan file.flow saat Anda menjalankan Notebook Jupyter yang diekspor dari Data Wrangler. Untuk mempelajari selengkapnya, lihat Ekspor.

Gunakan kebijakan yang serupa dengan berikut ini untuk memberikan izin yang dijelaskan dalam paragraf sebelumnya.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::sagemaker-region-account-id/data_wrangler_flows/",
                "arn:aws:s3:::sagemaker-region-account-id/data_wrangler_flows/*",
                "arn:aws:s3:::sagemaker-region-account-id/athena",
                "arn:aws:s3:::sagemaker-region-account-id/athena/*",
                "arn:aws:s3:::sagemaker-region-account-id/redshift",
                "arn:aws:s3:::sagemaker-region-account-id/redshift/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::sagemaker-region-account-id"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        }
    ]
}
```

Anda juga dapat mengakses data di bucket Amazon S3 dari AWS akun lain dengan menentukan URI bucket Amazon S3. Untuk melakukannya, kebijakan IAM yang memberikan akses ke bucket Amazon S3 di akun lain harus menggunakan kebijakan yang mirip dengan contoh berikut, `BucketFolder` di mana direktori spesifik di bucket pengguna. `UserBucket` Kebijakan ini harus ditambahkan ke pengguna yang memberikan akses ke bucket mereka untuk pengguna lain.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::UserBucket/BucketFolder/*"
            },
                {
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket"
                ],
                "Resource": "arn:aws:s3:::UserBucket",
                "Condition": {
                "StringLike": {
                    "s3:prefix": [
                    "BucketFolder/*"
                    ]
                }
            }
        }
```

```
    ]
}
```

Pengguna yang mengakses bucket (bukan pemilik bucket) harus menambahkan kebijakan yang mirip dengan contoh berikut kepada penggunanya. Perhatikan bahwa `AccountX` dan `TestUser` di bawah ini mengacu pada pemilik bucket dan penggunanya masing-masing.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountX:user/TestUser"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::UserBucket/BucketFolder/*"
            ]
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountX:user/TestUser"
            },
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::UserBucket"
            ]
        }
    ]
}
```

Contoh kebijakan untuk memberikan akses untuk menggunakan SageMaker Studio

Gunakan kebijakan seperti berikut ini untuk membuat peran eksekusi IAM yang dapat digunakan untuk menyiapkan instance Studio Classic.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedDomainUrl",
                "sagemaker:DescribeDomain",
                "sagemaker:ListDomains",
                "sagemaker:DescribeUserProfile",
                "sagemaker:ListUserProfiles",
                "sagemaker:*App",
                "sagemaker:ListApps"
            ],
            "Resource": "*"
        }
    ]
}
```

Kepingan Salju dan Data Wrangler

Semua izin untuk AWS sumber daya dikelola melalui peran IAM yang dilampirkan ke instans Studio Classic Anda. Administrator Snowflake mengelola izin khusus Snowflake, karena mereka dapat memberikan izin dan hak istimewa terperinci kepada setiap pengguna Snowflake. Ini termasuk database, skema, tabel, gudang, dan objek integrasi penyimpanan. Anda harus memastikan bahwa izin yang benar diatur di luar Data Wrangler.

Perhatikan bahwa `COPY INTO Amazon S3` perintah Snowflake memindahkan data dari Snowflake ke Amazon S3 melalui internet publik secara default, tetapi data dalam perjalanan diamankan menggunakan SSL. Data saat istirahat di Amazon S3 dienkripsi dengan SSE-KMS menggunakan default. AWS KMS key

Sehubungan dengan penyimpanan kredensil Snowflake, Data Wrangler tidak menyimpan kredensil pelanggan. Data Wrangler menggunakan Secrets Manager untuk menyimpan kredensialnya secara rahasia dan memutar rahasia sebagai bagian dari rencana keamanan praktik terbaik. Administrator Snowflake atau Studio Classic perlu memastikan bahwa peran eksekusi Studio Classic ilmuwan data diberikan izin untuk melakukan `GetSecretValue` rahasia yang menyimpan kredensyal. Jika sudah dilampirkan ke peran eksekusi Studio Classic, `AmazonSageMakerFullAccess` kebijakan memiliki izin yang diperlukan untuk membaca rahasia yang dibuat oleh Data Wrangler dan rahasia yang dibuat dengan mengikuti konvensi penamaan dan penandaan dalam instruksi di atas. Rahasia

yang tidak mengikuti konvensi harus diberikan akses secara terpisah. Sebaiknya gunakan Secrets Manager untuk mencegah berbagi kredensyal melalui saluran yang tidak aman; namun, perhatikan bahwa pengguna yang masuk dapat mengambil kata sandi teks biasa dengan meluncurkan terminal atau notebook Python di Studio Classic dan kemudian menjalankan panggilan API dari Secrets Manager API.

Enkripsi Data dengan AWS KMS

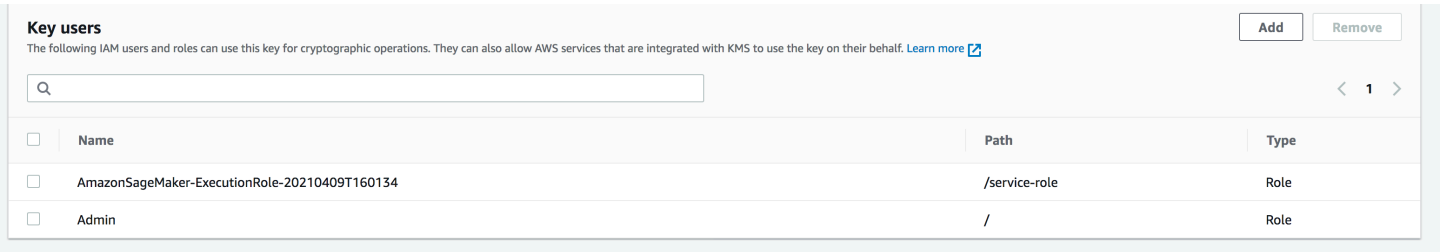
Dalam Data Wrangler, Anda dapat mendekripsi file terenkripsi dan menambahkannya ke aliran Data Wrangler Anda. Anda juga dapat mengenkripsi output transformasi menggunakan AWS KMS kunci default atau yang Anda berikan.

Anda dapat mengimpor file jika memiliki yang berikut:

- enkripsi sisi server
- SSE-KMS sebagai jenis enkripsi

Untuk mendekripsi file dan mengimpor ke alur Data Wrangler, Anda harus menambahkan pengguna SageMaker Studio Classic yang Anda gunakan sebagai pengguna kunci.

Tangkapan layar berikut menunjukkan peran pengguna Studio Classic yang ditambahkan sebagai pengguna utama. Lihat Peran IAM untuk mengakses pengguna di bawah panel kiri untuk membuat perubahan ini.



Penyiapan kunci terkelola pelanggan Amazon S3 untuk penyimpanan data impor Data Wrangler

Secara default, Data Wrangler menggunakan bucket Amazon S3 yang memiliki konvensi penamaan berikut:. `sagemaker-region-account number` Misalnya, jika nomor akun Anda `111122223333` dan Anda menggunakan Studio Classic di us-east-1, kumpulan data yang diimpor disimpan dengan konvensi penamaan berikut:. `sagemaker-us-east-1-111122223333`

Petunjuk berikut menjelaskan cara menyiapkan kunci terkelola pelanggan untuk bucket Amazon S3 default Anda.

1. Untuk mengaktifkan enkripsi sisi server dan menyiapkan kunci terkelola pelanggan untuk bucket S3 default Anda, lihat Menggunakan Enkripsi KMS.
2. Setelah mengikuti langkah 1, arahkan ke AWS KMS dalamAWS Management Console. Temukan kunci terkelola pelanggan yang Anda pilih di langkah 1 dari langkah sebelumnya dan tambahkan peran Studio Classic sebagai pengguna kunci. Untuk melakukannya, ikuti petunjuk di Memungkinkan pengguna kunci menggunakan kunci yang dikelola pelanggan.

Mengenkripsi Data yang Anda Ekspor

Anda dapat mengenkripsi data yang Anda ekspor menggunakan salah satu metode berikut:

- Menentukan bahwa bucket Amazon S3 memiliki objek menggunakan enkripsi SSE-KMS.
- Menentukan AWS KMS kunci untuk mengenkripsi data yang Anda ekspor dari Data Wrangler.

Pada halaman Ekspor data, tentukan nilai untuk ID AWS KMS kunci atau ARN.

Untuk informasi selengkapnya tentang menggunakan AWS KMS kunci, lihat Melindungi Data Menggunakan Enkripsi Sisi Server dengan AWS KMS kunci yang Disimpan di AWSAWS Key Management Service (SSE-KMS).

AppFlow Izin Amazon

Saat melakukan transfer, Anda harus menentukan peran IAM yang memiliki izin untuk melakukan transfer. Anda dapat menggunakan peran IAM yang sama yang memiliki izin untuk menggunakan Data Wrangler. Secara default, peran IAM yang Anda gunakan untuk mengakses Data Wrangler adalah. `SageMakerExecutionRole`

Peran IAM harus memiliki izin berikut:

- Izin ke Amazon AppFlow
- Izin ke Katalog AWS Glue Data
- Izin AWS Glue untuk menemukan sumber data yang tersedia

Saat Anda menjalankan transfer, Amazon AppFlow menyimpan metadata dari transfer di Katalog AWS Glue Data. Data Wrangler menggunakan metadata dari katalog untuk menentukan apakah tersedia untuk Anda kueri dan impor.

Untuk menambahkan izin ke Amazon AppFlow, tambahkan kebijakan `AmazonAppFlowFullAccess` AWS terkelola ke peran IAM. Untuk informasi selengkapnya tentang menambahkan kebijakan, lihat Menambahkan atau menghapus izin identitas IAM.

Jika Anda mentransfer data ke Amazon S3, Anda juga harus melampirkan kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketTagging",
        "s3:ListBucketVersions",
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:GetBucketPolicy",
        "s3:PutEncryptionConfiguration",
        "s3:GetEncryptionConfiguration",
        "s3:PutBucketTagging",
        "s3:GetObjectTagging",
        "s3:GetBucketOwnershipControls",
        "s3:PutObjectTagging",
        "s3:DeleteObject",
        "s3:DeleteBucket",
        "s3:DeleteObjectTagging",
        "s3:GetBucketPublicAccessBlock",
        "s3:GetBucketPolicyStatus",
        "s3:PutBucketPublicAccessBlock",
        "s3:PutAccountPublicAccessBlock",
        "s3:ListAccessPoints",
        "s3:PutBucketOwnershipControls",
        "s3:PutObjectVersionTagging",
        "s3:DeleteObjectVersionTagging",
        "s3:GetBucketVersioning",
        "s3:GetBucketAcl",
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetAccountPublicAccessBlock",
        "s3:ListAllMyBuckets",
        "s3:GetAnalyticsConfiguration",
```

```
        "s3:GetBucketLocation"
      ],
      "Resource": "*"
    }
  ]
}
```

Untuk menambahkan AWS Glue izin, tambahkan kebijakan `AWSGlueConsoleFullAccess` terkelola ke peran IAM. Untuk informasi selengkapnya tentang AWS Glue izin dengan Amazon AppFlow, lihat [link-to-appflow-page].

Amazon AppFlow perlu mengakses AWS Glue dan Data Wrangler agar Anda dapat mengimpor data yang telah Anda transfer. Untuk memberikan AppFlow akses Amazon, tambahkan kebijakan kepercayaan berikut ke peran IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root",
                "Service": [
                    "appflow.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Untuk menampilkan AppFlow data Amazon di Data Wrangler, tambahkan kebijakan berikut ke peran IAM:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": "glue:SearchTables",
            "Resource": [
                "arn:aws:glue:*:*:table/*/*",
                "arn:aws:glue:*:*:database/*",
                "arn:aws:glue:*:*:catalog"
            ]
        }
    ]
}
```

Menggunakan Konfigurasi Siklus Hidup di Data Wrangler

Anda mungkin memiliki instans Amazon EC2 yang dikonfigurasi untuk menjalankan aplikasi Kernel Gateway, tetapi bukan aplikasi Data Wrangler. Aplikasi Kernel Gateway menyediakan akses ke lingkungan dan kernel yang Anda gunakan untuk menjalankan notebook dan terminal Studio Classic. Aplikasi Data Wrangler adalah aplikasi UI yang menjalankan Data Wrangler. Instans Amazon EC2 yang bukan instans Data Wrangler memerlukan modifikasi pada konfigurasi siklus hidupnya untuk menjalankan Data Wrangler. Konfigurasi siklus hidup adalah skrip shell yang mengotomatiskan penyesuaian lingkungan Amazon Studio Classic Anda. SageMaker

Untuk informasi selengkapnya tentang konfigurasi siklus hidup lihat. Menggunakan konfigurasi siklus hidup dengan Amazon Studio Classic SageMaker

Konfigurasi siklus hidup default untuk instans Anda tidak mendukung penggunaan Data Wrangler. Anda dapat membuat modifikasi berikut pada konfigurasi default untuk menggunakan Data Wrangler dengan instans Anda.

```
#!/bin/bash
set -eux
STATUS=$(
python3 -c "import sagemaker_dataprep"
echo $?
)
if [ "$STATUS" -eq 0 ]; then
echo 'Instance is of Type Data Wrangler'
else
echo 'Instance is not of Type Data Wrangler'

# Replace this with the URL of your git repository
export REPOSITORY_URL="https://github.com/aws-samples/sagemaker-studio-lifecycle-
config-examples.git"
```

```
git -C /root clone $REPOSTIORY_URL

fi
```

Anda dapat menyimpan skrip sebagai`lifecycle_configuration.sh`.

Anda melampirkan konfigurasi siklus hidup ke domain Studio Classic atau profil pengguna. Untuk informasi selengkapnya tentang membuat dan melampirkan konfigurasi siklus hidup, lihat. Membuat dan mengaitkan konfigurasi siklus hidup

Petunjuk berikut menunjukkan cara melampirkan konfigurasi siklus hidup ke domain Studio Classic atau profil pengguna.

Anda mungkin mengalami kesalahan saat membuat atau melampirkan konfigurasi siklus hidup. Untuk informasi tentang kesalahan konfigurasi siklus hidup debugging,. KernelGateway kegagalan aplikasi

Catatan rilis

Data Wrangler diperbarui secara berkala dengan fitur dan perbaikan bug baru. Untuk memutakhirkan versi Data Wrangler yang Anda gunakan di Studio Classic, ikuti petunjuk di. Matikan dan Perbarui Aplikasi Studio Classic

Catatan rilis
8/31/2023 Fungsi JSON Sekarang Anda dapat membuat laporan Kualitas Data dan Wawasan di seluruh kumpulan data Anda. Untuk informasi selengkapnya, lihat Dapatkan Wawasan Tentang Kualitas Data dan Data.
5/20/2023 Fungsi JSON Anda sekarang dapat mengimpor data Anda dari Salesforce Data Cloud. Untuk informasi selengkapnya, lihat Impor data dari Salesforce Data Cloud.
4/18/2023

Catatan rilis

Fungsi JSON

Anda sekarang bisa mendapatkan data Anda dalam format yang dapat ditafsirkan oleh Amazon Personalize. Untuk informasi selengkapnya, lihat Kolom Peta untuk Amazon Personalisasi.

3/1/2023

Fungsi JSON

Anda sekarang dapat menggunakan Hive untuk mengimpor data Anda dari Amazon EMR. Untuk informasi selengkapnya, lihat Pengimporan data dari Amazon EMR.

12/10/2022

Fungsi JSON

Anda sekarang dapat mengekspor aliran Data Wrangler Anda ke titik akhir inferensi. Untuk informasi selengkapnya, lihat Ekspor ke Endpoint Inferensi.

Fungsi JSON

Anda sekarang dapat menggunakan widget notebook interaktif untuk persiapan data. Untuk informasi selengkapnya, lihat Menggunakan Widget Persiapan Data Interaktif di Notebook Amazon SageMaker Studio Classic untuk Mendapatkan Wawasan Data.

Fungsi JSON

Anda sekarang dapat mengimpor data dari platform SaaS. Untuk informasi selengkapnya, lihat Impor Data Dari Perangkat Lunak sebagai Platform Layanan (SaaS).

10/12/2022

Fungsi JSON

Anda sekarang dapat menggunakan kembali aliran data untuk kumpulan data yang berbeda. Untuk informasi selengkapnya, lihat Menggunakan Kembali Alur Data untuk Kumpulan Data yang Berbeda.

10/05/2022

Catatan rilis

Fungsi JSON

Anda sekarang dapat menggunakan Principal Component Analysis (PCA) sebagai transformasi. Untuk informasi selengkapnya, lihat Mengurangi Dimensionalitas dalam Dataset.

10/05/2022

Fungsi JSON

Anda sekarang dapat mereparasi parameter dalam alur Data Wrangler Anda. Untuk informasi selengkapnya, lihat Ekspor.

10/03/2022

Fungsi JSON

Anda sekarang dapat menerapkan model dari alur Data Wrangler Anda. Untuk informasi selengkapnya, lihat Secara Otomatis Melatih Model pada Alur Data Anda.

9/20/2022

Fungsi JSON

Anda sekarang dapat mengatur periode retensi data di Athena. Untuk informasi selengkapnya, lihat Impor data dari Athena.

6/9/2022

Fungsi JSON

Anda sekarang dapat menggunakan Amazon SageMaker Autopilot untuk melatih model langsung dari aliran Data Wrangler Anda. Untuk informasi selengkapnya, lihat Secara Otomatis Melatih Model pada Alur Data Anda.

5/6/2022

Fungsi JSON

Anda sekarang dapat menggunakan instance m5 dan r5 tambahan. Untuk informasi selengkap nya, lihat Instans.

Catatan rilis

27/04/2022

Fungsi JSON

- Anda sekarang bisa mendapatkan laporan kualitas data. Lihat informasi yang lebih lengkap di Dapatkan Wawasan Tentang Kualitas Data dan Data
- Anda sekarang dapat melakukan pengambilan sampel acak dan pengambilan sampel bertingka t. Untuk informasi selengkapnya, lihat Pengambilan sampel.

4/1/2022

Fungsi JSON

Anda sekarang dapat menggunakan Databricks sebagai sumber data. Untuk informasi selengkap nya, lihat Impor data dari Databricks (JDBC).

2/2/2022

Fungsi JSON

- Anda sekarang dapat mengekspor menggunakan node tujuan. Lihat informasi yang lebih lengkap di Ekspor
- Anda dapat mengimpor file ORC dan JSON. Untuk informasi selengkapnya tentang tipe file, lihatImpor.
- Data Wrangler sekarang mendukung penggunaan transformasi SMOTE. Untuk informasi selengkapnya, lihat Data Saldo.
- Data Wrangler sekarang mendukung pengkodean kesamaan untuk data kategoris. Untuk informasi selengkapnya, lihat Kesamaan menyandikan.
- Data Wrangler sekarang mendukung unnesting data JSON. Untuk informasi selengkapnya, lihat Fungsi JSON.
- Data Wrangler sekarang mendukung perluasan nilai-nilai array ke kolom terpisah. Untuk informasi selengkapnya, lihat Penyempurnaan Data.
- Data Wrangler sekarang mendukung menjangkau tim layanan saat Anda mengalami masalah. Untuk informasi selengkapnya, lihat Pemecahan Masalah.

Catatan rilis

- Data Wrangler mendukung langkah pengeditan dan penghapusan dalam aliran data Anda. Lihat informasi yang lebih lengkap di Hapus Langkah dari Alur Data Anda dan Mengedit Langkah dalam Alur Wrangler Data Anda.
- Anda sekarang dapat melakukan transformasi pada beberapa kolom. Untuk informasi selengkap nya, lihat Memindahkan.
- Data Wrangler sekarang mendukung tag alokasi biaya. Untuk informasi selengkapnya, lihat Menggunakan Tag Alokasi Biaya.

10/16/2021

Fungsi JSON

Data Wrangler sekarang mendukung kelompok kerja Athena. Untuk informasi selengkapnya, lihat Impor data dari Athena.

10/6/2021

Fungsi JSON

Data Wrangler sekarang mendukung transformasi data deret waktu. Untuk informasi selengkap nya, lihat Mengubah Seri Waktu.

7/15/2021

Fungsi JSON

- Kepingan Salju dan Data Wranglersekarang didukung. Anda dapat menggunakan Snowflake sebagai sumber data di Data Wrangler.
- Menambahkan dukungan untuk pembatas bidang kustom di CSV. Sekarang koma, titik dua, titik koma, pipa (|) dan Tab didukung.
- Sekarang Anda dapat mengekspor hasil langsung ke Amazon S3.
- Menambahkan beberapa analisa multikolinearitas baru: Faktor Inflasi Varians, Analisis Komponen Utama dan pemilihan fitur Lasso.

Penyempurnaan Fungsi JSON

Catatan rilis

- Grafik analisis tidak bisa lagi dikemas dengan label yang tumpang tindih.

Perbaikan Bug:

- Encoder satu panas menangani string kosong dengan anggun.
- Memperbaiki crash yang terjadi ketika nama kolom kerangka data berisi titik-titik.

26/04/2021

Penyempurnaan Fungsi JSON

- Menambahkan dukungan untuk Pekerjaan pemrosesan terdistribusi. Anda dapat menggunakan beberapa instance saat menjalankan pekerjaan pemrosesan.
- Pekerjaan Pemrosesan Data Wrangler sekarang secara otomatis menggabungkan output kecil ketika perkiraan ukuran hasil kurang dari 1 gigabyte.
- Feature Store Notebook: Peningkatan kinerja konsumsi feature store
- Pekerjaan Pemrosesan Data Wrangler sekarang menggunakan 1.x sebagai tag wadah otoritatif untuk rilis masa depan.

Perbaikan Bug:

- Memperbaiki masalah rendering untuk histogram faset.
- Memperbaiki Ekspor ke Processing Job untuk mendukung kolom tipe vektor.
- `Extract using regex`Operator tetap untuk mengembalikan grup yang ditangkap pertama jika satu atau lebih ada dalam ekspresi reguler atau regex.

2/8/2021

Fungsi JSON

- Data Wrangler Flows mendukung beberapa instance.
- Diperbarui Ekspor ke Data Wrangler Job Notebook untuk menggunakan SageMaker SDK 2.20.0.
- Diperbarui Ekspor ke Notebook Pipeline untuk menggunakan SageMaker SDK 2.20.0.

Catatan rilis

- Diperbarui Ekspor ke Pipeline Notebook untuk menambahkan contoh pelatihan XGBoost sebagai langkah opsional.

Penyempurnaan Fungsi JSON

- Untuk meningkatkan kinerja, mengimpor file CSV yang berisi beberapa baris dalam satu bidang tidak lagi didukung.

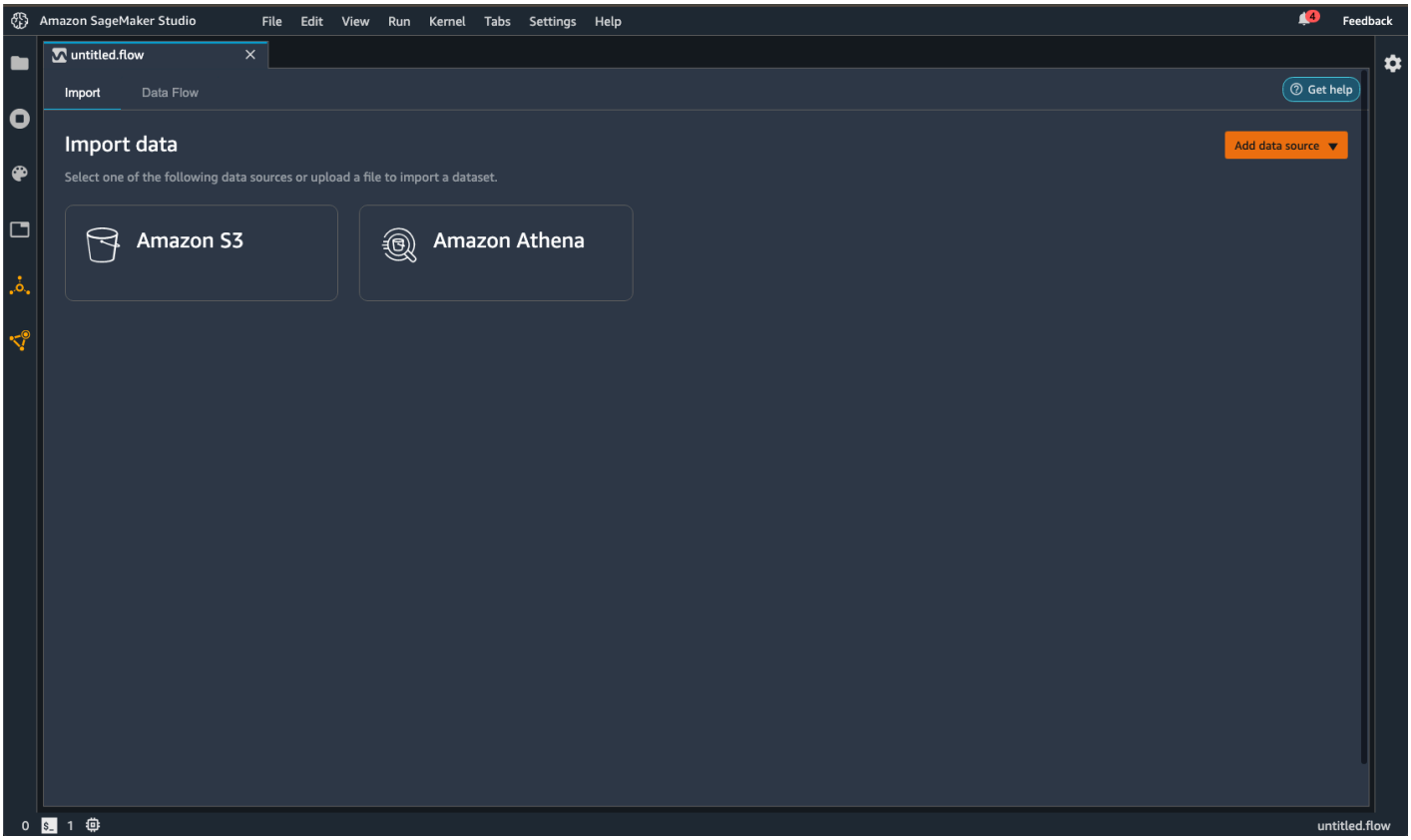
Perbaikan Bug:

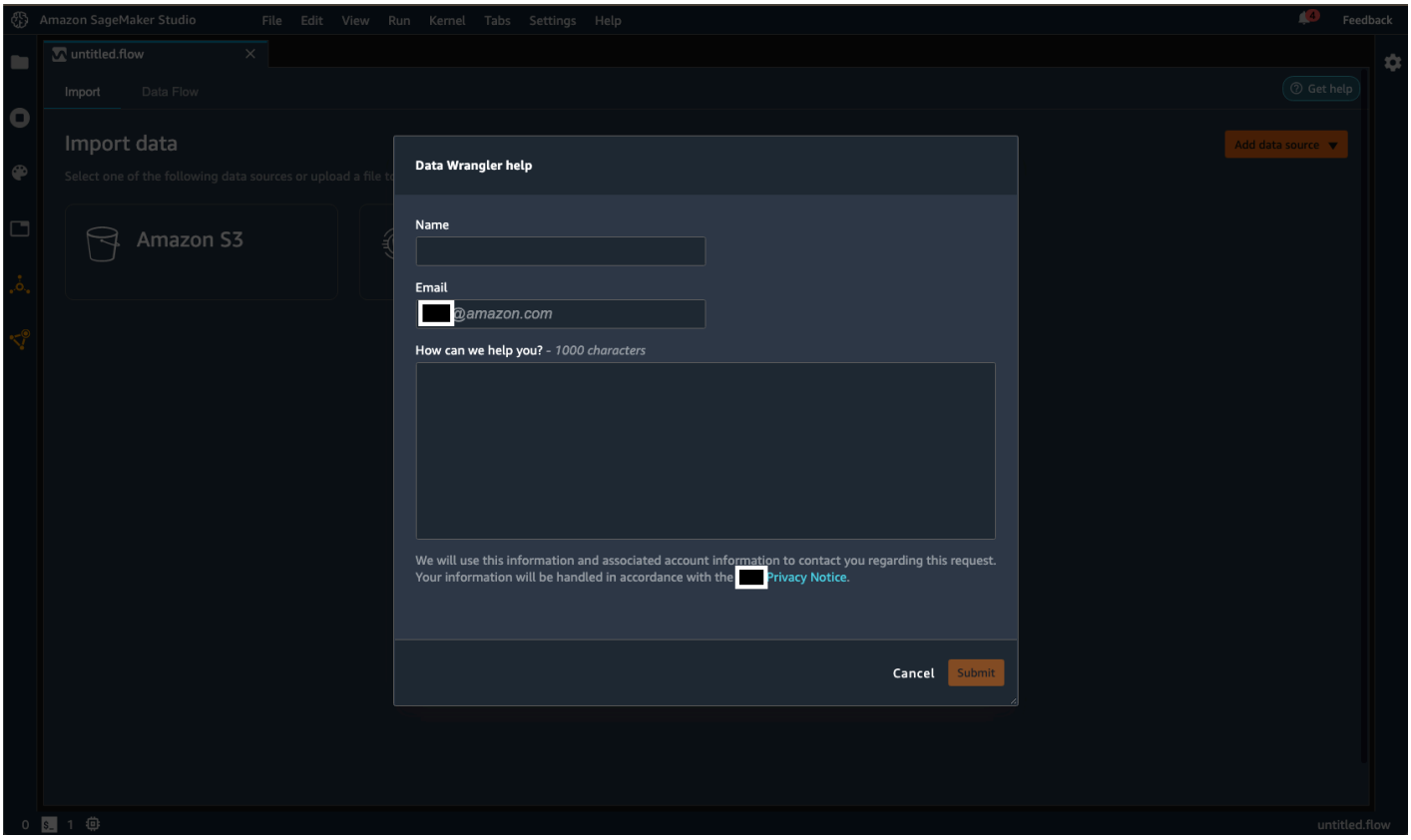
- Memperbaiki masalah inferensi tipe dalam model Cepat.
- Memperbaiki bug metrik bias dalam laporan bias.
- Memperbaiki transformasi teks Featurize agar berfungsi dengan kolom dengan nilai yang hilang.
- Visualisasi bawaan plot Histogram dan Scatter tetap untuk bekerja dengan kumpulan data yang berisi kolom seperti array.
- Kueri Athena sekarang berjalan kembali jika ID eksekusi kueri telah kedaluwarsa.

Pemecahan Masalah

Jika masalah muncul saat menggunakan Amazon SageMaker Data Wrangler, kami sarankan Anda melakukan hal berikut:

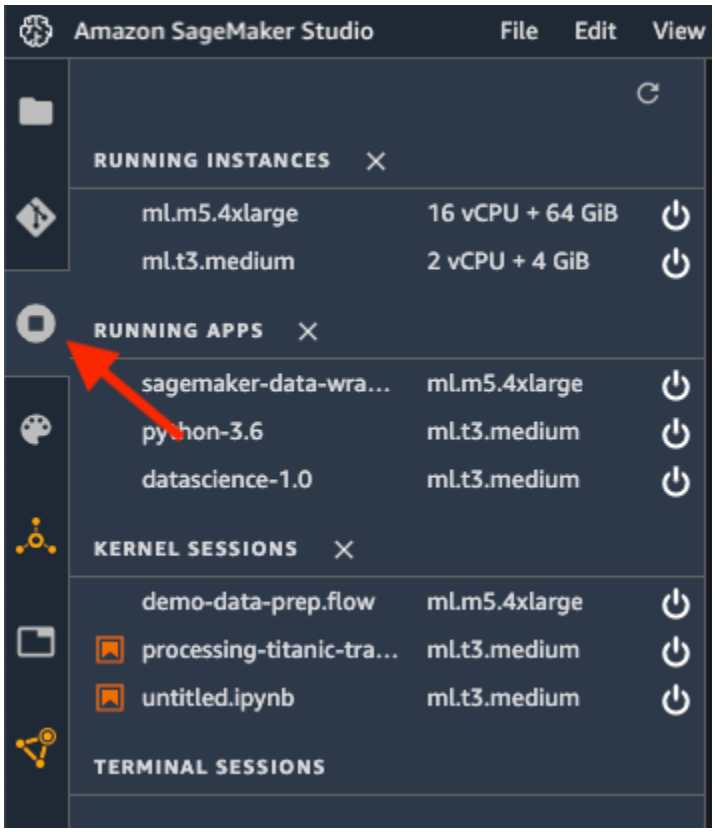
- Jika pesan kesalahan disediakan, baca pesan dan selesaikan masalah yang dilaporkan jika memungkinkan.
- Pastikan peran IAM pengguna Studio Classic Anda memiliki izin yang diperlukan untuk melakukan tindakan. Untuk informasi selengkapnya, lihat Keamanan dan Izin.
- Jika masalah terjadi saat Anda mencoba mengimpor dari AWS layanan lain, seperti Amazon Redshift atau Athena, pastikan Anda telah mengonfigurasi izin dan sumber daya yang diperlukan untuk melakukan impor data. Untuk informasi selengkapnya, lihat Impor.
- Jika Anda masih mengalami masalah, pilih Dapatkan bantuan di kanan atas layar untuk menghubungi tim Data Wrangler. Untuk informasi selengkapnya, lihat gambar berikut.


Amazon SageMaker Studio
File Edit View Run Kernel Tabs Settings Help
Feedback
untitled.flow
Import
Data Flow
Get help
Import data
Add data source
Select one of the following data sources or upload a file to import a dataset.
Amazon S3
Amazon Athena
0 1
untitled.flow

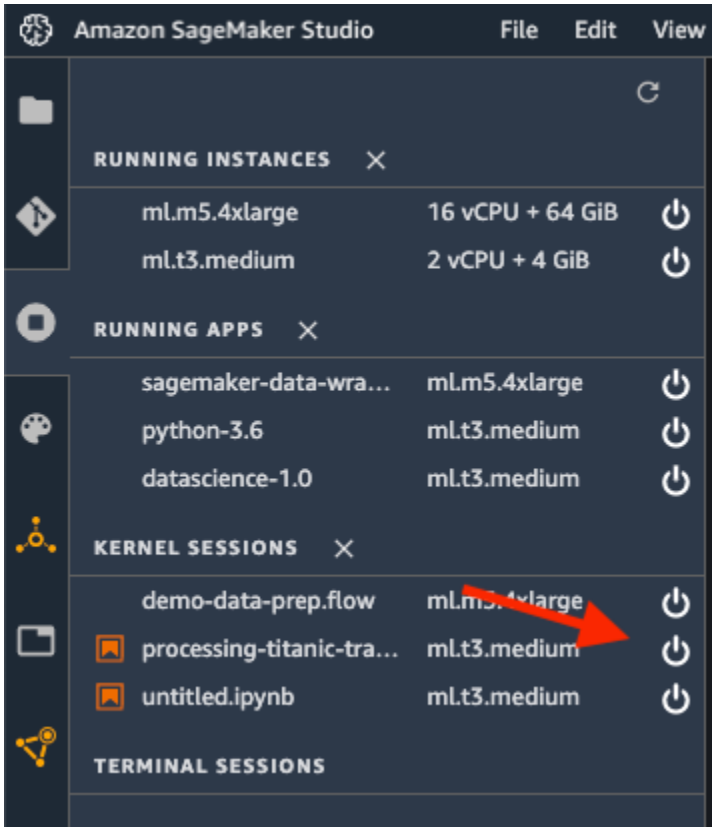

Amazon SageMaker Studio
File Edit View Run Kernel Tabs Settings Help
Feedback
untitled.flow
Import
Data Flow
Get help
Import data
Add data source
Amazon S3
Data Wrangler help
Name
Email
@amazon.com
How can we help you? - 1000 characters
We will use this information and associated account information to contact you regarding this request.
Your information will be handled in accordance with the Privacy Notice.
Cancel
Submit
0 1
untitled.flow

Sebagai upaya terakhir, Anda dapat mencoba memulai ulang kernel tempat Data Wrangler berjalan.

1. Simpan dan keluar dari file.flow yang ingin Anda mulai ulang kernelnya.
2. Pilih ikon Running Terminal dan Kernels, seperti yang ditunjukkan pada gambar berikut.



3. Pilih ikon Stop di sebelah kanan file.flow yang ingin Anda akhiri kernelnya, seperti yang ditunjukkan pada gambar berikut.



4. Segarkan peramban.
5. Buka kembali file.flow tempat Anda bekerja.

Pemecahan masalah dengan Amazon EMR

Gunakan informasi berikut ini untuk membantu Anda memecahkan masalah yang mungkin muncul saat Anda menggunakan Amazon EMR.

- Kegagalan koneksi - Jika koneksi gagal dengan pesan berikut`The IP address of the EMR cluster isn't private error message`, kluster EMR Amazon Anda mungkin tidak diluncurkan di subnet pribadi. Sebagai praktik terbaik keamanan, Data Wrangler hanya mendukung koneksi ke kluster EMR Amazon pribadi. Pilih subnet EC2 pribadi yang Anda luncurkan kluster EMR.
- Koneksi hang dan timing out — Masalah ini kemungkinan besar disebabkan oleh masalah konektivitas jaringan. Setelah Anda mulai menghubungkan ke cluster, layar tidak menyegarkan. Setelah sekitar 2 menit, Anda mungkin melihat kesalahan berikut`JdbcAddConnectionError: An error occurred when trying to connect to presto: xxx: Connect to xxx`

`failed: Connection timed out (Connection timed out) will display on top of the screen..`

Kesalahan mungkin memiliki dua akar penyebab:

- Amazon EMR dan Amazon SageMaker Studio Classic ada di VPC yang berbeda. Kami merekomendasikan meluncurkan Amazon EMR dan Studio Classic di VPC yang sama. Anda juga dapat menggunakan VPC peering. Untuk informasi selengkapnya, lihat Apa yang itu peering VPC? .
- Grup keamanan master EMR Amazon tidak memiliki aturan lalu lintas masuk untuk grup keamanan Amazon SageMaker Studio Classic di port yang digunakan untuk Presto. Untuk mengatasi masalah ini, izinkan lalu lintas masuk di port 8889.

- Koneksi gagal karena jenis koneksi yang salah dikonfigurasi - Anda mungkin melihat pesan galat berikut: `Data Wrangler couldn't create a connection to {connection_source} successfully. Try connecting to {connection_source} again. For more information, see Troubleshoot. If you're still experiencing issues, contact support.`

  Periksa metode autentikasi. Metode otentikasi yang telah Anda tentukan di Data Wrangler harus cocok dengan metode otentikasi yang Anda gunakan di cluster.

- Anda tidak memiliki izin HDFS untuk otentikasi LDAP — Gunakan panduan berikut untuk mengatasi masalah Mengatur Izin HDFS menggunakan Kredensyal Linux. Anda dapat masuk ke klaster menggunakan perintah berikut:

  ```
  hdfs dfs -mkdir /user/USERNAME
  hdfs dfs -chown USERNAME:USERNAME /user/USERNAME
  ```

- Otentikasi LDAP hilang kesalahan kunci koneksi — Anda mungkin melihat pesan galat berikut:. `Data Wrangler couldn't connect to EMR hive successfully. JDBC connection is missing required connection key(s): PWD`

  Untuk otentikasi LDAP, Anda harus menentukan nama pengguna dan kata sandi. URL JDBC yang disimpan di Secrets Manager tidak memiliki properti. `PWD`

- Saat Anda memecahkan masalah konfigurasi LDAP: Sebaiknya pastikan autentikator LDAP (server LDAP) dikonfigurasi dengan benar untuk terhubung ke kluster EMR Amazon. Gunakan

`ldapwhoami` perintah untuk membantu Anda menyelesaikan masalah konfigurasi. Berikut ini adalah contoh perintah yang dapat Anda jalankan:

- Untuk LDAPS — `ldapwhoami -x -H ldaps://ldap-server`
- Untuk LDAP — `ldapwhoami -x -H ldap://ldap-server`

Perintah mana pun harus kembali `Anonymous` jika Anda telah mengkonfigurasi autentikator dengan sukses.

Pemecahan Masalah dengan Salesforce

Kesalahan konfigurasi siklus hidup

Saat pengguna Anda membuka Studio Classic untuk pertama kalinya, mereka mungkin mendapatkan kesalahan yang mengatakan bahwa ada yang salah dengan konfigurasi siklus hidupnya. Gunakan Amazon CloudWatch untuk mengakses log yang ditulis oleh skrip konfigurasi siklus hidup Anda. Untuk informasi selengkapnya tentang konfigurasi siklus hidup men-debug, lihat. Debug konfigurasi siklus hidup

Jika Anda tidak dapat men-debug kesalahan, Anda dapat membuat file konfigurasi secara manual. Anda harus membuat file setiap kali Anda menghapus atau memulai ulang server Jupyter. Gunakan prosedur berikut untuk membuat file secara manual.

Untuk membuat file konfigurasi

1. Arahkan ke Studio Classic.
2. Pilih File, lalu Baru, lalu Terminal.
3. Buat `.sfgenie_identity_provider_oauth_config`.
4. Buka file di editor teks.
5. Tambahkan objek JSON yang berisi Amazon Resource Name (ARN) rahasia Secrets Manager ke file. Anda dapat menggunakan template berikut untuk membuat objek.

```
{
  "secret_arn": "example-secret-ARN"
}
```

6. Simpan perubahan Anda ke file .

Tidak dapat mengakses Salesforce Data Cloud dari alur Data Wrangler

Setelah pengguna Anda memilih Salesforce Data Cloud dari alur Data Wrangler Anda, mereka mungkin mendapatkan kesalahan yang menunjukkan prasyarat untuk menyiapkan koneksi belum terpenuhi. Ini mungkin disebabkan oleh kesalahan berikut:

- Rahasia Salesforce di Secrets Manager belum dibuat.
- Rahasia Salesforce di Secrets Manager telah dibuat, tetapi tidak ada tag Salesforce.
- Rahasia Salesforce di Secrets Manager telah dibuat dengan cara yang salah. Wilayah AWS Misalnya, pengguna Anda tidak akan dapat mengakses Salesforce Data Cloud `ca-central-1` karena Anda telah membuat rahasia di. `us-east-1` Anda dapat mereplikasi rahasia ke `ca-central-1` atau membuat rahasia baru dengan kredensi yang sama di. `ca-central-1` Untuk informasi tentang mereplikasi rahasia, lihat Mereplikasi AWS Secrets Manager rahasia ke yang lain. Wilayah AWS
- Kebijakan yang digunakan pengguna untuk mengakses Amazon SageMaker Studio Classic tidak memiliki izin AWS Secrets Manager
- Ada kesalahan ketik di Secrets Manager ARN dari objek JSON yang telah Anda tentukan melalui konfigurasi siklus hidup Anda.
- Ada kesalahan ketik dalam rahasia Secrets Manager yang berisi konfigurasi OAuth Salesforce Anda

Halaman kosong menampilkan **`redirect_uri_mismatch`**

Setelah pengguna Anda memilih Simpan dan Connect, mereka mungkin akan diarahkan ke halaman yang ditampilkan`redirect_uri_mismatch`. URI callback yang telah Anda daftarkan di setelan Salesforce Connected App hilang atau salah.

Gunakan URL berikut untuk memeriksa apakah URL Studio Classic Anda terdaftar dengan benar di setelan Aplikasi Terhubung organisasi Salesforce Anda:. `https://EXAMPLE_SALESFORCE_ORG/lightning/setup/NavigationMenus/home/` Untuk informasi selengkapnya tentang menggunakan pengaturan aplikasi tersambung, kunjungi URL berikut:`https://EXAMPLE_SALESFORCE_ORG/lightning/setup/NavigationMenus/home/`.

> **Note**
>
> Dibutuhkan sekitar sepuluh menit untuk menyebarkan URI dalam sistem Salesforce.

Fungsi SQL baru

Ruang bersama saat ini tidak berfungsi dengan integrasi Salesforce Data Cloud. Anda dapat menghapus spasi bersama di SageMaker Domain Amazon yang ingin Anda gunakan, atau Anda dapat menggunakan Domain lain yang tidak memiliki spasi bersama yang disiapkan.

Kesalahan Pengalihan OAuth

Pengguna Anda harus dapat mengimpor data mereka dari Salesforce Data Cloud setelah mereka memilih Connect. Jika mereka mengalami kesalahan, kami sarankan meminta mereka untuk melakukan hal berikut:

- Beri tahu mereka untuk bersabar — Ketika mereka diarahkan kembali ke Amazon SageMaker Studio Classic, diperlukan waktu hingga satu menit untuk menyelesaikan proses otentikasi. Sementara mereka diarahkan, kami sarankan untuk memberitahu mereka untuk menghindari berinteraksi dengan browser. Misalnya, mereka tidak boleh menutup tab browser, beralih ke tab lain, atau berinteraksi dengan aliran Data Wrangler. Berinteraksi dengan browser mungkin menghapus kode otorisasi yang diperlukan untuk terhubung ke cloud data.
- Minta pengguna Anda terhubung kembali ke cloud data — Ada masalah sementara yang dapat menyebabkan koneksi ke Salesforce Data Cloud gagal. Mintalah pengguna Anda membuat alur Data Wrangler baru dan coba sambungkan ke Salesforce Data Cloud lagi.
- Pastikan pengguna Anda menutup semua tab lain dengan Amazon SageMaker Studio Classic — Memiliki Studio Classic terbuka di beberapa tab dapat menyebabkan koneksi Salesforce Data Cloud gagal. Pastikan pengguna Anda hanya memiliki satu tab Studio Classic yang terbuka.
- Beberapa pengguna yang mengakses Studio Classic secara bersamaan — Hanya satu pengguna yang dapat mengakses SageMaker Domain Amazon pada satu waktu. Jika beberapa pengguna mengakses Domain yang sama, koneksi yang pengguna coba buat ke Salesforce Data Cloud mungkin gagal.

Memperbarui Data Wrangler dan Studio Classic juga dapat memperbaiki kesalahan mereka. Untuk informasi tentang memperbarui Data Wrangler, lihat. Perbarui Data Wrangler Untuk informasi tentang memperbarui Studio Classic, lihatMatikan dan Perbarui SageMaker Studio Classic.

Jika tidak ada langkah pemecahan masalah sebelumnya yang berhasil, Anda mungkin menemukan pesan kesalahan dari Salesforce dengan deskripsi terkait yang disematkan di URL Studio Classic. Berikut ini adalah contoh dari sebuah pesan yang dapat Anda temukan: `error=invalid_client_id&error_description=client%20identifier%20invalid`

Anda dapat melihat pesan kesalahan di URL dan mencoba mengatasi masalah yang ditimbulkannya. Jika pesan kesalahan atau deskripsi tidak jelas, sebaiknya cari Basis Pengetahuan Salesforce. Jika mencari basis pengetahuan tidak berhasil, Anda dapat menghubungi help desk Salesforce untuk bantuan lebih lanjut.

Data Wrangler membutuhkan waktu lama untuk dimuat

Ketika pengguna Anda diarahkan kembali ke Data Wrangler dari Salesforce Data Cloud, mereka mungkin mengalami waktu muat yang lama.

Jika ini adalah pertama kalinya pengguna menggunakan Data Wrangler atau mereka telah menghapus kernel, mungkin diperlukan waktu sekitar 5 menit untuk menyediakan instans Amazon EC2 baru untuk menggunakan Data Wrangler.

Jika ini bukan pertama kalinya pengguna menggunakan Data Wrangler dan mereka belum menghapus kernel, Anda dapat meminta mereka untuk menyegarkan halaman atau menutup tab browser sebanyak mungkin.

Jika tidak ada intervensi sebelumnya yang berfungsi, minta mereka mengatur koneksi baru ke Salesforce Data Cloud.

Pengguna gagal mengekspor data mereka dengan **`Invalid batch Id`** kesalahan

Saat pengguna Anda mengekspor transformasi yang mereka buat ke data Salesforce mereka, pekerjaan SageMaker pemrosesan yang digunakan Data Wrangler di backend mungkin gagal. Salesforce Data Cloud mungkin sementara tidak tersedia atau mungkin ada masalah caching.

Untuk mengatasi masalah ini, kami sarankan agar pengguna Anda kembali ke langkah di mana mereka mengimpor data dan mengubah urutan kolom yang mereka kueri. Misalnya, mereka dapat mengubah kueri berikut:

```
SELECT col_A, col_B FROM table
```

Untuk kueri berikut:

```
SELECT col_B, col_A FROM table
```

Setelah mereka mengubah urutan kolom dan memastikan bahwa transformasi berikutnya yang mereka buat masih valid, mereka dapat mulai mengekspor data mereka lagi.

Pengguna tidak dapat mengekspor kumpulan data yang sangat besar

Jika pengguna Anda mengimpor dataset yang sangat besar dari Salesforce Data Cloud, mereka mungkin tidak dapat mengekspor transformasi yang telah mereka buat. Dataset besar mungkin memiliki terlalu banyak baris, atau dapat dihasilkan dari kueri yang kompleks.

Kami menyarankan agar pengguna Anda melakukan tindakan berikut:

- Menyederhanakan kueri SQL mereka
- Mengambil sampel data mereka

Berikut ini adalah beberapa strategi yang dapat mereka gunakan untuk menyederhanakan pertanyaan mereka:

- Tentukan nama kolom alih-alih menggunakan * operator
- Menemukan subset data yang ingin mereka impor alih-alih menggunakan subset yang lebih besar
- Meminimalkan gabungan antara kumpulan data yang sangat besar

Mereka dapat menggunakan sampling untuk mengurangi jumlah baris dalam dataset mereka. Untuk informasi tentang metode pengambilan sampel, pengguna Anda dapat merujuk kePengambilan sampel.

Pengguna tidak dapat mengekspor data karena token penyegaran tidak valid

Data Wrangler menggunakan driver JDBC untuk berintegrasi dengan Salesforce Data Cloud. Metode untuk otentikasi adalah OAuth. Untuk OAuth, token penyegaran dan token akses adalah dua bagian data berbeda yang digunakan untuk mengotorisasi akses ke sumber daya dalam Cloud Data Salesforce Anda.

Token akses, atau token inti, adalah apa yang memungkinkan Anda mengakses data Salesforce Anda dan menjalankan kueri secara langsung melalui Data Wrangler. Ini berumur pendek dan dirancang untuk kedaluwarsa dengan cepat. Untuk mempertahankan akses ke data Salesforce Anda, Data Wrangler menggunakan token penyegaran untuk mendapatkan token akses baru dari Salesforce.

Anda mungkin telah mengatur penyegaran agar kedaluwarsa terlalu cepat untuk mendapatkan token akses baru bagi pengguna Anda. Anda mungkin harus meninjau kembali kebijakan token penyegaran Anda untuk memastikan bahwa itu dapat mengakomodasi kueri yang membutuhkan waktu lama untuk dijalankan bagi pengguna Anda. Untuk informasi tentang mengonfigurasi kebijakan token penyegaran, lihat`https://EXAMPLE_SALESFORCE_ORG_URL/lightning/setup/ConnectedApplication/home/`.

Kueri gagal atau tabel tidak dimuat

Salesforce mengalami pemadaman layanan. Bahkan jika Anda telah mengonfigurasi semuanya dengan benar, pengguna Anda mungkin tidak dapat mengimpor data mereka untuk jangka waktu tertentu.

Pemadaman layanan dapat terjadi karena alasan pemeliharaan. Kami merekomendasikan untuk memeriksa di hari berikutnya untuk melihat apakah masalah telah diselesaikan.

Jika Anda mengalami masalah selama lebih dari satu hari, kami sarankan untuk menghubungi help desk Salesforce untuk bantuan lebih lanjut. Untuk informasi tentang menghubungi Salesforce, lihat Bagaimana Anda ingin menghubungi Salesforce?

`OAUTH_APP_BLOCKED`selama pengalihan Studio Classic

Ketika pengguna Anda diarahkan kembali ke Amazon SageMaker Studio Classic, mereka mungkin melihat parameter kueri `error=OAUTH_APP_BLOCKED` dalam URL. Mereka mungkin mengalami masalah sementara yang harus diselesaikan sendiri dalam sehari.

Ada kemungkinan bahwa Anda telah memblokir akses mereka ke Aplikasi Terhubung juga. Untuk informasi tentang menyelesaikan masalah, lihat`https://EXAMPLE_SALESFORCE_ORG_URL/lightning/setup/ConnectedApplication/home/`.

`OAUTH_APP_DENIED`selama pengalihan Studio Classic

Ketika pengguna Anda diarahkan kembali ke Amazon SageMaker Studio Classic, mereka mungkin melihat parameter kueri `error=OAUTH_APP_ACCESS_DENIED` dalam URL. Anda belum memberikan izin jenis profil mereka untuk mengakses yang `Connected App` terkait dengan Data Wrangler.

Untuk mengatasi masalah akses mereka, navigasikan ke `https://EXAMPLE_SALESFORCE_ORG_URL/lightning/setup/ManageUsers/home/` dan periksa apakah pengguna ditetapkan ke profil yang benar.

Meningkatkan Batas Instans Amazon EC2

Anda mungkin melihat pesan galat berikut saat menggunakan Data Wrangler: `The following instance type is not available: ml.m5.4xlarge. Try selecting a different instance below.`

Pesan dapat menunjukkan bahwa Anda perlu memilih jenis instans yang berbeda, tetapi juga dapat menunjukkan bahwa Anda tidak memiliki cukup instans Amazon EC2 untuk berhasil menjalankan Data Wrangler pada alur kerja Anda. Anda dapat meningkatkan jumlah instans dengan menggunakan prosedur berikut.

Untuk menambah jumlah instans, lakukan hal berikut.

1. Buka AWS Management Console.
2. Di bilah pencarian, tentukan**`Services Quotas`**.
3. Pilih Service Quotas.
4. Pilih AWSlayanan.
5. Di bilah pencarian, tentukan**`Amazon SageMaker`**.
6. Pilih Amazon SageMaker.
7. Di bawah Kuota layanan, tentukan**`Studio KernelGateway Apps running on ml.m5.4xlarge instance`**.

   > **Note**
   >
   > ml.m5.4xlarge adalah tipe instance default untuk Data Wrangler. Anda dapat menggunakan jenis instans lain dan meminta peningkatan kuota untuk mereka. Untuk informasi selengkapnya, lihat Instans.

8. Pilih KernelGateway Aplikasi Studio yang berjalan pada instance ***`ml.m5.4xlarge.`***
9. Pilih Ajukan peningkatan kuota.
10. Untuk Ubah nilai kuota, tentukan nilai yang lebih besar dari nilai kuota Terapan.
11. Pilih Minta.

Jika permintaan Anda disetujui, AWS kirimkan notifikasi ke alamat email yang terkait dengan akun Anda. Anda juga dapat memeriksa status permintaan Anda dengan memilih riwayat permintaan Kuota pada halaman Service Quotas. Permintaan yang diproses memiliki Status Tertutup.

Perbarui Data Wrangler

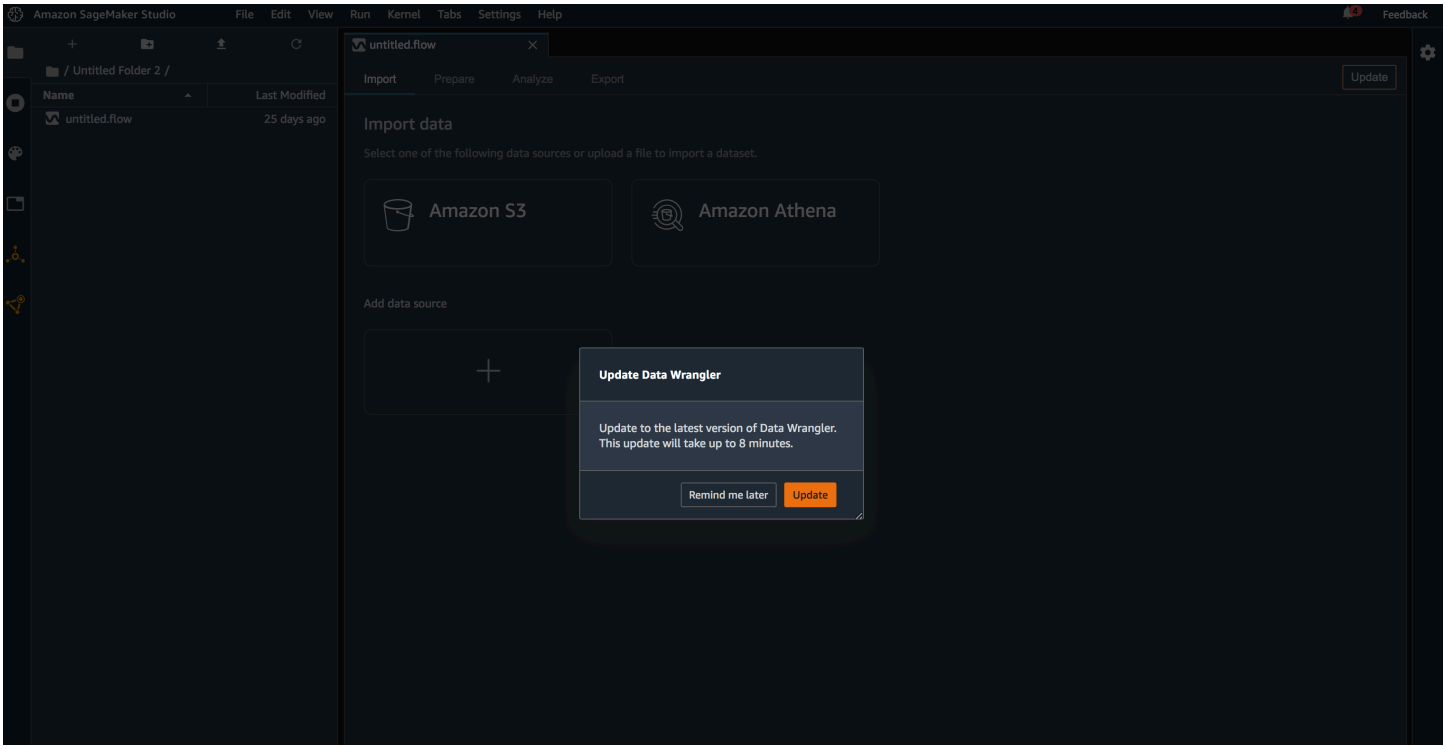
Untuk memperbarui Data Wrangler ke rilis terbaru, pertama-tama matikan KernelGateway aplikasi yang sesuai dari panel kontrol Amazon SageMaker Studio Classic. Setelah KernelGateway aplikasi dimatikan, restart dengan membuka aliran Data Wrangler baru atau yang sudah ada di Studio Classic. Saat Anda membuka aliran Data Wrangler baru atau yang sudah ada, kernel yang dimulai berisi versi terbaru Data Wrangler.

Memperbarui instans Studio Classic dan Data Wrangler

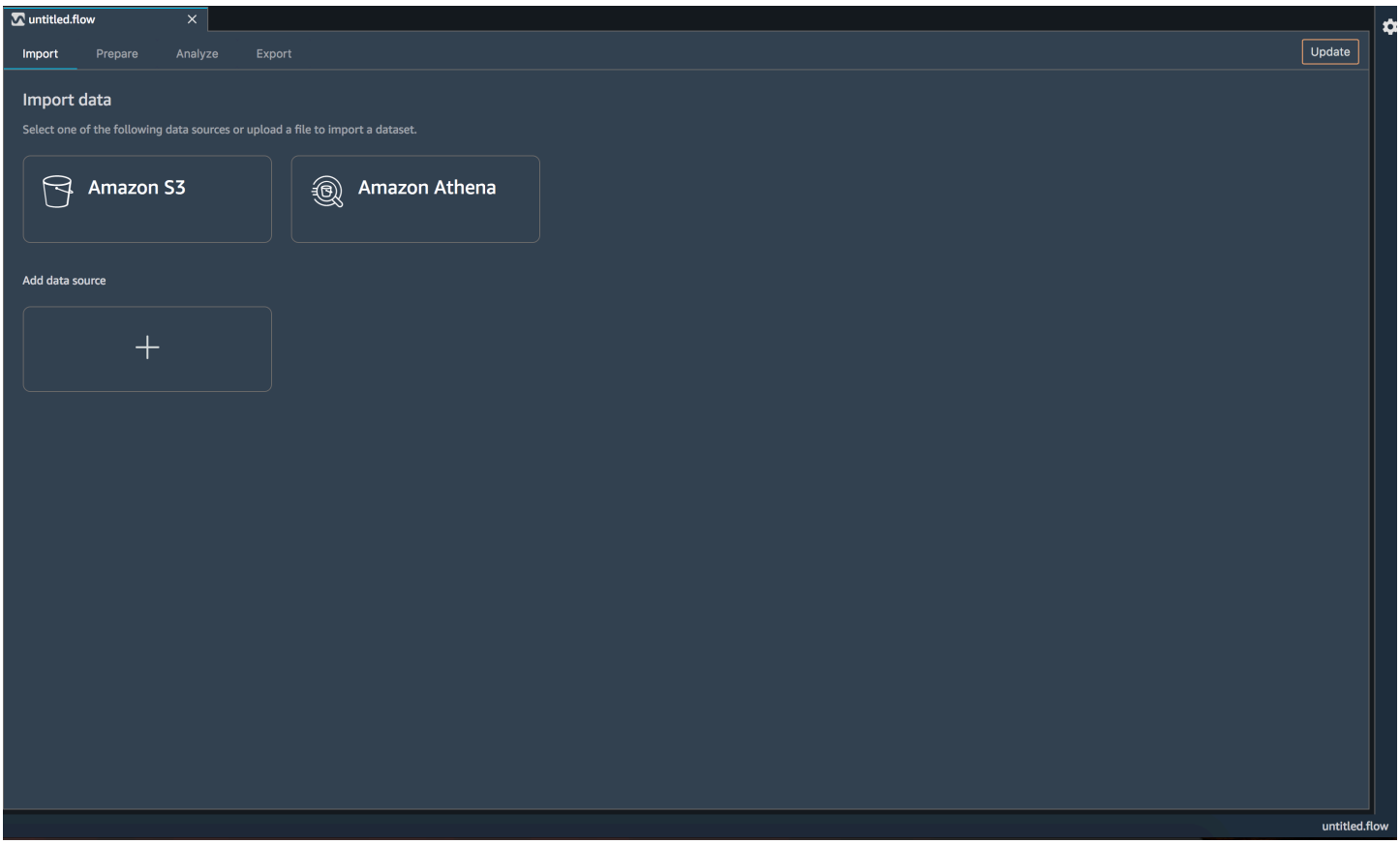
1. Arahkan ke SageMakerKonsol Anda.
2. Pilih SageMaker dan kemudian Studio Classic.
3. Pilih nama pengguna Anda.
4. Di bawah Aplikasi, di baris yang menampilkan nama Aplikasi, pilih Hapus aplikasi untuk aplikasi yang dimulai`sagemaker-data-wrang`, dan untuk JupyterServer aplikasi.
5. Pilih Ya, hapus aplikasi.
6. `delete`Ketik kotak konfirmasi.
7. Pilih Hapus.
8. Buka kembali instans Studio Classic Anda. Saat Anda mulai membuat alur Data Wrangler, instans Anda sekarang menggunakan versi terbaru Data Wrangler.

Atau, jika Anda menggunakan versi aplikasi Data Wrangler yang bukan versi terbaru, dan Anda memiliki aliran Data Wrangler yang ada terbuka, Anda diminta untuk memperbarui versi aplikasi Data Wrangler Anda di UI Studio Classic. Tangkapan layar berikut menunjukkan prompt ini.

> **Important**
>
> Ini hanya memperbarui aplikasi gateway kernel Data Wrangler. Anda masih perlu mematikan JupyterServer aplikasi di akun pengguna Anda. Untuk melakukannya, ikuti langkah-langkah sebelumnya.



Anda juga dapat memilih Ingatkan saya nanti, dalam hal ini tombol Pembaruan muncul di sudut kanan atas layar.



Matikan Data Wrangler

Saat Anda tidak menggunakan Data Wrangler, penting untuk mematikan instance yang dijalankannya untuk menghindari biaya tambahan.

Untuk menghindari kehilangan pekerjaan, simpan aliran data Anda sebelum mematikan Data Wrangler. Untuk menyimpan aliran data Anda di Studio Classic, pilih File dan kemudian pilih Save Data Wrangler Flow. Data Wrangler secara otomatis menyimpan aliran data Anda setiap 60 detik.

Untuk mematikan instance Data Wrangler di Studio Classic

1. Di Studio Classic, pilih ikon Running Instances dan Kernels
   ()  .
2. Di bawah RUNNING APPS adalah aplikasi sagemaker-data-wrangler-1.0. Pilih ikon shutdown di sebelah aplikasi ini
   ().

   Data Wrangler berjalan pada instance ml.m5.4xlarge. Instance ini menghilang dari RUNNING INSTANCES saat Anda mematikan aplikasi Data Wrangler.

Setelah Anda mematikan aplikasi Data Wrangler, aplikasi ini harus dimulai ulang saat berikutnya Anda membuka file aliran Data Wrangler. Hal ini dapat menghabiskan waktu beberapa menit.

Persiapan data skala menggunakan Apache Spark, Hive, atau Presto di Amazon EMR atau AWS Glue dari notebook Amazon Studio Classic SageMaker

Amazon SageMaker Studio Classic menyediakan alat bagi ilmuwan data, insinyur pembelajaran mesin, dan dokter umum untuk melakukan analisis data dan persiapan data dalam skala besar. Menganalisis, mengubah, dan menyiapkan data dalam jumlah besar adalah langkah dasar dari setiap ilmu data dan alur kerja ML. SageMaker Studio Classic dilengkapi dengan integrasi built-in Amazon EMR dan Sesi AWS Glue Interaktif untuk menangani persiapan data interaktif skala besar dan alur kerja pembelajaran mesin, semuanya dalam notebook Studio Classic Anda.

Amazon EMR adalah platform data besar terkelola dengan sumber daya untuk membantu Anda menjalankan pekerjaan pemrosesan data terdistribusi skala petabyte menggunakan kerangka kerja analitik sumber terbuka AWS seperti Apache Spark, Apache Hive, Presto, HBase, Flink, dan Hudi.

Insinyur data dan ilmuwan data menggunakan Amazon EMR untuk berbagai kasus penggunaan, termasuk analitik data besar, analisis bagaimana-jika, analitik real-time, dan persiapan data untuk pembelajaran mesin. Dengan integrasi Studio Classic dengan Amazon EMR, Anda dapat membuat, menelusuri, menemukan, dan terhubung ke kluster EMR Amazon tanpa meninggalkan notebook Studio Classic Anda. Anda juga dapat memantau dan men-debug beban kerja Spark Anda dengan akses sekali klik ke UI Spark dari dalam notebook. Anda harus mempertimbangkan Amazon EMR untuk beban kerja persiapan data Anda jika Anda menginginkan kontrol maksimum atas versi perangkat keras dan perangkat lunak, wadah, dan aplikasi pemrosesan data besar.

AWS GlueSesi Interaktif adalah layanan tanpa server yang dapat Anda daftarkan untuk mengumpulkan, mengubah, membersihkan, dan menyiapkan data untuk penyimpanan di danau data dan jalur data Anda. AWS Glue Sesi Interaktif menyediakan lingkungan runtime Apache Spark tanpa server sesuai permintaan yang dapat Anda inisialisasi dalam hitungan detik pada Unit Pemrosesan Data (DPU) khusus tanpa harus khawatir tentang penyediaan dan pengelolaan infrastruktur cluster komputasi yang kompleks. Setelah inisialisasi, Anda dapat dengan cepat menelusuri katalog AWS Glue data, menjalankan kueri besar, mengakses data yang diatur olehAWS Lake Formation, dan menganalisis serta menyiapkan data secara interaktif menggunakan Spark, langsung di notebook Studio Classic Anda. Anda kemudian dapat menggunakan data yang disiapkan untuk melatih, menyetel, dan menerapkan model menggunakan alat ML yang dibuat khusus dalam Studio Classic. SageMaker Anda harus mempertimbangkan Sesi AWS Glue Interaktif untuk beban kerja persiapan data Anda ketika Anda menginginkan layanan Spark tanpa server dengan kontrol konfigurasi dan fleksibilitas yang moderat.

Daftar isi

Siapkan data menggunakan Amazon EMR

Amazon SageMaker Studio Classic hadir dengan integrasi built-in Amazon EMR, yang dengannya ilmuwan data dan insinyur data dapat melakukan persiapan data interaktif skala petabyte dan pembelajaran mesin (ML) langsung dari notebook Studio Classic mereka. Dalam notebook, mereka dapat menemukan dan terhubung ke cluster EMR Amazon yang ada, kemudian secara interaktif mengeksplorasi, memvisualisasikan, dan menyiapkan data skala besar untuk pembelajaran mesin menggunakan Apache Spark, Apache Hive, Presto. Selain itu, pengguna dapat mengakses Spark UI dengan satu klik untuk memantau pekerjaan Spark mereka dari notebook Studio Classic mereka.

Administrator dapat menggunakan AWS Service Cataloguntuk menentukan AWS CloudFormationtemplat kluster EMR Amazon yang dapat diakses oleh pengguna Studio Classic. Ilmuwan data kemudian dapat memilih templat yang telah ditentukan untuk menyediakan sendiri cluster EMR Amazon langsung dari notebook Amazon SageMaker Studio Classic. Administrator dapat membuat parameter template lebih lanjut agar pengguna dapat memilih aspek klaster agar sesuai dengan beban kerja mereka dalam nilai yang telah ditentukan sebelumnya. Misalnya, ilmuwan data atau insinyur data mungkin ingin menentukan jumlah node inti cluster hingga nilai maksimum yang telah ditentukan, atau pilih jenis instance node dari menu tarik-turun.

- Jika Anda seorang administrator, pastikan Anda telah mengaktifkan komunikasi antara notebook Amazon SageMaker Studio Classic dan kluster Amazon EMR. Untuk instruksi, lihat Konfigurasikan jaringan (untuk administrator) bagian. Setelah komunikasi ini diaktifkan, Anda memiliki opsi untuk:
  - Tentukan template cluster AWS Service Catalog dan pastikan ketersediaan template ini melalui notebook Studio Classic:Konfigurasikan templat EMR Amazon di AWS Service Catalog (untuk administrator).
  - Konfigurasikan kemampuan ditemukan kluster EMR Amazon yang ada langsung dari notebook Studio Classic:. Konfigurasikan kemampuan ditemukan kluster EMR Amazon (untuk administrator)
- Jika Anda seorang ilmuwan data atau insinyur data yang ingin menyediakan sendiri kluster EMR Amazon, lihat. Meluncurkan klaster Amazon EMR dari Studio Classic
- Jika Anda seorang ilmuwan data atau insinyur data yang ingin menemukan dan terhubung ke kluster EMR Amazon yang ada dari Studio Classic, lihat. Gunakan kluster EMR Amazon dari notebook Studio Classic

Daftar topik

Konfigurasikan jaringan (untuk administrator)

Bagian ini memberikan informasi tentang bagaimana administrator dapat mengonfigurasi jaringan mereka untuk memungkinkan komunikasi antara notebook Amazon SageMaker Studio Classic dan klaster EMR Amazon.

Instruksi jaringan bervariasi berdasarkan apakah SageMaker Studio Classic dan Amazon EMR digunakan dalam Amazon Virtual Private Cloud (VPC) pribadi atau berkomunikasi melalui internet.

Secara default, SageMaker Studio Classic berjalan di VPC AWS terkelola dengan akses internet. Saat menggunakan koneksi internet, Studio Classic mengakses AWS sumber daya, seperti bucket Amazon S3, melalui internet. Namun, jika Anda memiliki persyaratan keamanan untuk mengontrol akses ke data dan wadah pekerjaan, sebaiknya Anda mengonfigurasi SageMaker Studio Classic dan Amazon EMR agar data dan container Anda tidak dapat diakses melalui internet. Untuk mengontrol akses ke sumber daya Anda atau menjalankan SageMaker Studio Classic tanpa akses internet publik, Anda dapat menentukan jenis akses `VPC only` jaringan saat Anda onboard ke SageMaker Domain Amazon. Dalam skenario ini, SageMaker Studio Classic membuat koneksi dengan AWS layanan lain melalui titik akhir VPC pribadi. Untuk informasi tentang mengonfigurasi SageMaker Studio Classic dalam `VPC only` mode, lihat Connect SageMaker Studio Classic notebook di VPC ke sumber daya eksternal. .

Dua bagian pertama menjelaskan cara memastikan komunikasi antara SageMaker Studio Classic dan cluster EMR Amazon di VPC tanpa akses internet publik. Bagian terakhir mencakup cara memastikan komunikasi antara SageMaker Studio Classic dan Amazon EMR menggunakan koneksi internet. Sebelum menghubungkan SageMaker Studio Classic dan Amazon EMR tanpa akses internet, pastikan untuk membuat titik akhir untuk Amazon Simple Storage Service (penyimpanan data), Amazon CloudWatch (logging dan monitoring), dan Amazon SageMaker Runtime (kontrol akses berbasis peran halus (RBAC)).

- Jika klaster Amazon SageMaker Studio Classic dan Amazon EMR Anda disiapkan di VPC yang berbeda di AWS akun yang sama atau di akun yang berbeda, lihat. Studio Classic dan Amazon EMR digunakan di VPC terpisah
- Jika klaster Amazon SageMaker Studio Classic dan Amazon EMR Anda diatur dalam VPC yang sama, lihat. Amazon SageMaker Studio Classic dan Amazon EMR berada di VPC yang sama
- Jika Anda memilih untuk menghubungkan Amazon SageMaker Studio Classic dan Amazon EMR cluster melalui internet publik, lihat. Amazon SageMaker Studio Classic dan Amazon EMR berkomunikasi melalui internet publik

Studio Classic dan Amazon EMR digunakan di VPC terpisah

Untuk memungkinkan komunikasi antara SageMaker Studio Classic dan klaster EMR Amazon saat digunakan di VPC yang berbeda:

1. Mulailah dengan menghubungkan VPC Anda melalui koneksi peering VPC.
2. Perbarui tabel perutean Anda di setiap VPC untuk merutekan lalu lintas jaringan antara subnet Studio Classic dan subnet Amazon EMR dengan dua arah.
3. Konfigurasikan grup keamanan Anda untuk mengizinkan lalu lintas masuk dan keluar.

Langkah-langkahnya serupa, terlepas dari apakah Amazon SageMaker Studio Classic dan kluster EMR Amazon digunakan dalam AWS akun yang sama (Kasus penggunaan akun tunggal) atau akun yang berbeda AWS (Kasus penggunaan lintas akun).

1. Peering VPC

   Buat koneksi peering VPC untuk memfasilitasi jaringan antara dua VPC (Studio SageMaker Classic dan Amazon EMR).

   a. Dari akun SageMaker Studio Classic Anda, di dasbor VPC, pilih Koneksi peering, lalu Buat koneksi peering.

   b. Buat permintaan Anda untuk mengintip VPC Studio Classic dalam VPC EMR Amazon. Saat meminta mengintip di AWS akun lain, pilih Akun lain di Pilih VPC lain untuk diajak mengintip.

      Untuk mengintip lintas akun, administrator harus menerima permintaan dari akun EMR Amazon.

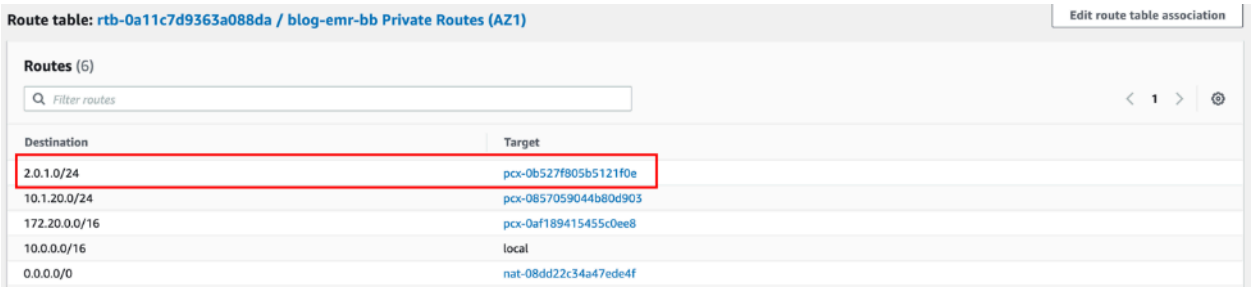
      Saat mengintip subnet pribadi, Anda harus mengaktifkan resolusi DNS IP pribadi pada tingkat koneksi peering VPC.

2. Tabel rute per set

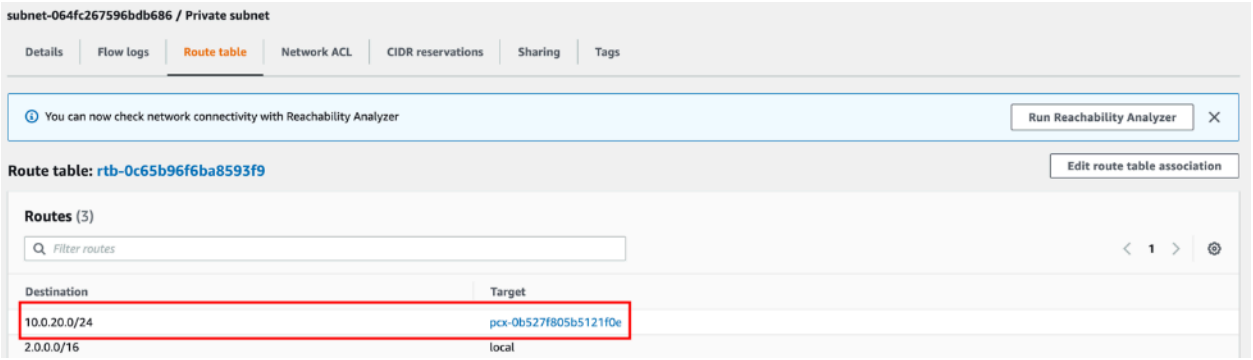
   Kirim lalu lintas jaringan antara subnet SageMaker Studio Classic dan subnet Amazon EMR dua arah.

   Setelah Anda membuat koneksi peering, administrator (pada setiap akun untuk akses lintas akun) dapat menambahkan rute ke tabel rute subnet pribadi untuk merutekan lalu lintas antara notebook dan subnet cluster. Anda dapat menentukan rute tersebut dengan membuka bagian Tabel Rute dari setiap VPC di dasbor VPC.

Ilustrasi berikut dari tabel rute Studio Classic VPCSubnet menunjukkan contoh rute keluar dari akun Studio Classic ke rentang IP VPC EMR Amazon (di sini) melalui koneksi peering. `2.0.1.0/24`



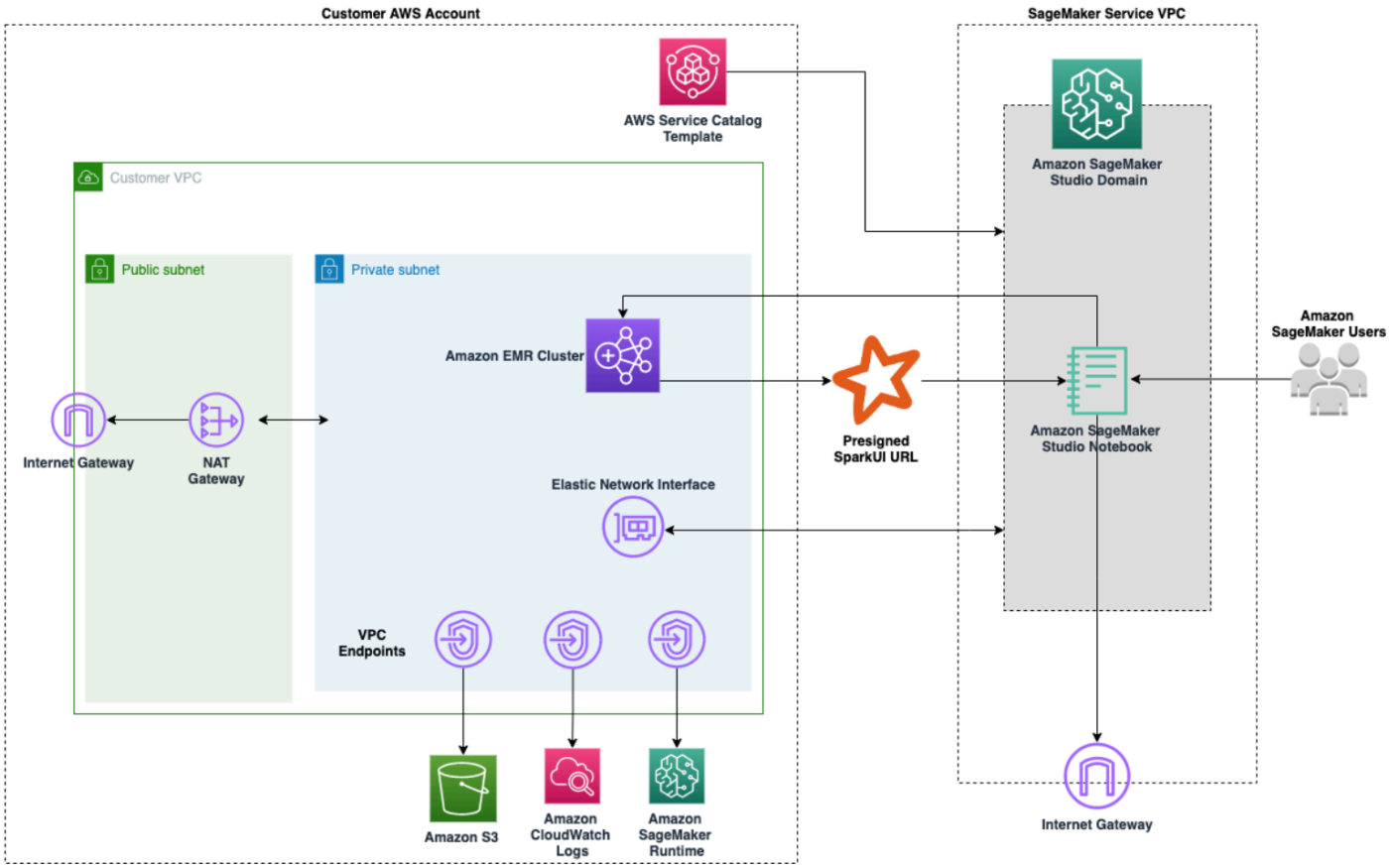
Ilustrasi berikut dari tabel rute subnet Amazon EMR VPC menunjukkan contoh rute kembali dari Amazon EMR VPC ke Studio Classic VPC IP range (di sini) melalui koneksi peering. `10.0.20.0/24`



3. Grup keamanan

   Terakhir, grup keamanan domain Studio Classic Anda harus mengizinkan lalu lintas keluar, dan grup keamanan node utama EMR Amazon harus mengizinkan lalu lintas masuk pada port Apache Livy, Hive, atau Presto TCP (masing-masing8998,, dan) dari grup keamanan instans Studio Classic. `10000` 8889 Apache Livy adalah sebuah layanan yang memungkinkan interaksi dengan klaster Amazon EMR melalui antarmuka REST.

Gambar berikut menunjukkan contoh penyiapan VPC Amazon yang memungkinkan notebook SageMaker Studio Classic menyediakan kluster EMR Amazon dari AWS CloudFormation templat dan kemudian terhubung ke kluster EMR Amazon dalam akun yang sama. AWS Diagram memberikan ilustrasi tambahan tentang titik akhir yang diperlukan untuk koneksi langsung ke berbagai AWS layanan, seperti Amazon S3 atau CloudWatch Amazon, ketika VPC tidak memiliki akses internet. Atau, gateway NAT harus digunakan untuk memungkinkan instance di subnet pribadi beberapa VPC untuk berbagi satu alamat IP publik yang disediakan oleh gateway internet saat mengakses internet.



Amazon SageMaker Studio Classic dan Amazon EMR berada di VPC yang sama

Jika Amazon SageMaker Studio Classic dan klaster berada dalam subnet yang berbeda, tambahkan rute ke setiap tabel rute subnet pribadi untuk merutekan lalu lintas antara notebook dan subnet cluster. Anda dapat menentukan rute tersebut dengan membuka bagian Tabel Rute dari setiap VPC di dasbor VPC. Jika Anda menggunakan Amazon SageMaker Studio Classic dan kluster EMR Amazon di VPC yang sama dan subnet yang sama, Anda tidak perlu merutekan lalu lintas antara notebook dan cluster.

Apakah Anda perlu memperbarui tabel perutean atau tidak, grup keamanan domain Studio Classic Anda harus mengizinkan lalu lintas keluar, dan grup keamanan node utama EMR Amazon harus mengizinkan lalu lintas masuk pada port Apache Livy, Hive, atau Presto TCP (masing-masing8998,, dan) dari grup keamanan instans Studio Classic. `10000` `8889` Apache Livy adalah sebuah layanan yang memungkinkan interaksi dengan klaster Amazon EMR melalui antarmuka REST.

Amazon SageMaker Studio Classic dan Amazon EMR berkomunikasi melalui internet publik

Secara default, SageMaker Studio Classic menyediakan antarmuka jaringan yang memungkinkan komunikasi dengan internet melalui gateway internet di VPC yang terkait dengan Domain. SageMaker Jika Anda memilih untuk terhubung ke Amazon EMR melalui internet publik, kluster EMR Amazon Anda harus menerima lalu lintas masuk pada port Apache Livy, Hive, atau Presto TCP (masing-masing,, dan) dari gateway internetnya. `8998` `10000` `8889` Apache Livy adalah layanan yang memungkinkan interaksi dengan cluster EMR Amazon melalui antarmuka REST.

Perlu diingat bahwa setiap port tempat Anda mengizinkan lalu lintas masuk merupakan potensi kelemahan keamanan. Cermatlah dalam meninjau grup keamanan kustom untuk memastikan bahwa Anda meminimalisir kelemahan. Untuk informasi selengkapnya, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

Atau, lihat Panduan dan whitepaper panduan terperinci tentang cara mengaktifkan Kerberos di Amazon EMR, mengatur cluster di subnet pribadi, dan mengakses cluster menggunakan Network Load Balancer (NLB) untuk mengekspos hanya port tertentu, yang dikendalikan akses melalui grup keamanan.

> **Note**
>
> Saat menghubungkan ke titik akhir Apache Livy Anda melalui internet publik, kami menyarankan agar Anda mengamankan komunikasi antara Amazon SageMaker Studio Classic dan klaster EMR Amazon Anda menggunakan TLS.
> Untuk informasi tentang pengaturan HTTPS dengan Apache Livy, lihat Mengaktifkan HTTPS dengan Apache Livy. Untuk informasi tentang menyetel klaster EMR Amazon dengan enkripsi transit diaktifkan, lihat Menyediakan sertifikat untuk mengenkripsi data saat transit dengan enkripsi Amazon EMR. Selain itu, Anda perlu mengonfigurasi Studio Classic untuk mengakses kunci sertifikat Anda seperti yang ditentukan dalamConnect ke klaster Amazon EMR melalui HTTPS.

Buat klaster EMR Amazon dari notebook Studio Classic

Administrator dapat menggunakan AWS Service Cataloguntuk menentukan AWS CloudFormationtemplate klaster EMR Amazon sebagai produk portofolio, kemudian membuatnya tersedia untuk pengguna yang dipilih. Menggunakan Service Catalog, administrator dapat sepenuhnya mengontrol pengaturan organisasi, keamanan, dan jaringan kluster Amazon EMR. Ilmuwan data dan insinyur data kemudian dapat melihat, memilih, dan menyesuaikan templat

tersebut untuk beban kerja spesifik mereka guna membuat klaster EMR Amazon sesuai permintaan langsung dari notebook Studio Classic mereka. SageMaker Ini dapat dilakukan tanpa mengatur konfigurasi yang rumit secara manual. Pengguna juga dapat menghentikan cluster EMR Amazon dari notebook Studio Classic setelah digunakan.

- Jika Anda seorang administrator yang ingin mengonfigurasi AWS CloudFormation template sebagai AWS Service Catalog produk sehingga pengguna dapat membuat kluster EMR Amazon dari Studio Classic, lihat. Konfigurasikan templat EMR Amazon di AWS Service Catalog (untuk administrator)
- Jika Anda seorang ilmuwan data atau insinyur data yang ingin menyediakan sendiri klaster EMR Amazon untuk memproses data dalam skala besar menggunakan kerangka kerja sumber terbuka seperti Apache Spark, Apache Hive, atau Presto, lihat. Meluncurkan klaster Amazon EMR dari Studio Classic
- Jika Anda ingin menemukan dan terhubung ke kluster EMR Amazon yang ada dari Studio Classic, lihat. Gunakan kluster EMR Amazon dari notebook Studio Classic

Topik

Konfigurasikan templat EMR Amazon di AWS Service Catalog (untuk administrator)

Bagian ini memberikan detail tentang cara administrator mengonfigurasi AWS Service Catalogproduk sehingga pengguna dapat menyediakan sendiri klaster EMR Amazon dari notebook Amazon Studio Classic secara mandiri. SageMaker Selain itu, administrator dapat mengonfigurasi templat klaster EMR Amazon dengan cara sehingga pengguna akhir dapat menyesuaikan berbagai aspek cluster agar sesuai dengan kebutuhan spesifik mereka. Misalnya, administrator dapat menentukan daftar jenis instance yang diizinkan dari mana pengguna dapat memilih saat membuat cluster.

Topik ini mengasumsikan bahwa Anda terbiasa dengan pembuatan portofolio dan produk di AWS Service Catalog serta Amazon EMR, dan. AWS CloudFormation

> **Note**
>
> Anda dapat merujuk ke AWS CloudFormation template di aws-samples/ sagemaker-studio-emr GitHub repositori sebagai contoh CloudFormation tumpukan untuk menerapkan peran IAM, Amazon VPC, Sandbox Studio Classic Domain, profil pengguna, serta template untuk

meluncurkan klaster EMR Amazon. CloudFormation Beberapa opsi tersedia tergantung pada metode otentikasi Anda antara Studio Classic dan klaster EMR Amazon. Dalam contoh ini, CloudFormation template induk meneruskan parameter ID SageMaker VPC, grup keamanan, dan subnet ID ke CloudFormation template kluster EMR Amazon.
Anda dapat mengakses berbagai contoh templat EMR CloudFormation Amazon di repositori bersarang sagemaker-studio-emr/cloudformation/emr_servicecatalog_templates dan selanjutnya memilih dari satu penerapan akun ke lintas akun.
Untuk informasi selengkapnya tentang metode otentikasi yang tersedia saat menyambungkan ke klaster EMR Amazon, lihat. Gunakan kluster EMR Amazon dari notebook Studio Classic

Untuk menyederhanakan pembuatan cluster EMR Amazon, administrator dapat mendaftarkan CloudFormation template cluster EMR Amazon sebagai produk dalam portofolio. AWS Service Catalog Kemudian mereka mengaitkan portofolio Service Catalog dengan peran eksekusi Studio Classic untuk memastikan ketersediaan template di Studio Classic. Selain itu, untuk memastikan bahwa ilmuwan data dapat menemukan template tersebut, menyediakan kluster EMR Amazon, dan terhubung ke kluster EMR Amazon dari notebook Studio Classic mereka, administrator perlu menetapkan izin akses yang tepat.

Daftar berikut menyediakan pengaturan tambahan yang perlu diterapkan administrator ke CloudFormation tumpukan dasar untuk mengaktifkan Studio Classic mengakses produk Service Catalog dan menyediakan kluster Amazon EMR. Pengaturan tersebut harus diterapkan di berbagai tingkatan:

- Dalam portofolio Service Catalog
- Dalam produk Service Catalog
- Dalam template CloudFormation Amazon EMR dinyatakan sebagai produk Service Catalog

Terakhir, administrator harus menetapkan izin yang diperlukan untuk peran eksekusi Studio Classic yang mengakses cluster dan akun tempat Amazon EMR digunakan, berdasarkan apakah Studio Classic dan Amazon EMR berada di akun yang sama atau berbeda. AWS

- Prasyarat: Persyaratan jaringan dan otentikasi

  Sebagai prasyarat, pastikan bahwa Anda telah meninjau persyaratan jaringan dan keamanan Konfigurasikan jaringan (untuk administrator) dan bahwa Anda telah membuat CloudFormation

tumpukan dasar yang mendukung metode otentikasi pilihan Anda. Anda dapat menemukan contoh CloudFormation template di sagemaker-studio-emraws-samples/.

- Dalam portofolio Service Catalog Anda:

  Tambahkan bagian berikut ke CloudFormation template portofolio Anda (lihat contoh dalam format YAMAL) untuk mengaitkan portofolio Anda dengan peran eksekusi Studio Classic yang mengakses klaster Anda.

```
SageMakerStudioEMRProductPortfolioPrincipalAssociation:
    Type: AWS::ServiceCatalog::PortfolioPrincipalAssociation
    Properties:
      PrincipalARN: SageMakerExecutionRole.Arn
      PortfolioId: SageMakerStudioEMRProductPortfolio ID
      PrincipalType: IAM
```

- Dalam produk Service Catalog Anda:

  Tambahkan kunci tag berikut `"sagemaker:studio-visibility:emr"` dan atur ke nilai `"true"` (di sini di YAMAL) ke produk Service Catalog yang mereferensikan sumber daya template Amazon EMR. Ini memastikan visibilitas template di Studio Classic.

```
SMStudioEMRNoAuthProduct:
    Type: AWS::ServiceCatalog::CloudFormationProduct
    Properties:
      Owner: AWS
      Name: SageMaker Studio Domain No Auth EMR
      ProvisioningArtifactParameters:
        - Name: SageMaker Studio Domain No Auth EMR
          Description: Provisions a SageMaker domain and No Auth EMR Cluster
          Info:
            LoadTemplateFromURL: Link to your CloudFormation template. For example,
 https://aws-ml-blog.s3.amazonaws.com/artifacts/astra-m4-sagemaker/end-to-end/CFN-
EMR-NoStudioNoAuthTemplate-v3.yaml
      Tags:
        - Key: "sagemaker:studio-visibility:emr"
          Value: "true"
```

- Dalam CloudFormation template cluster EMR Amazon dalam produk Service Catalog Anda:

Tambahkan parameter tumpukan wajib berikut sebagai placeholder. Bagian ini diisi dengan nama proyek Studio Classic dan pengenal yang digunakan oleh pengguna saat menyediakan klaster dari Studio Classic.

```
SageMakerProjectName:
Type: String
Description: Name of the project

SageMakerProjectId:
Type: String
Description: Service generated Id of the project.
```

Administrator dapat menentukan `Default` dan `AllowedValues` menyertakan pilihan di bagian parameter template, membiarkan pengguna memasukkan atau memilih nilai kustom saat membuat klaster. Contoh berikut menggambarkan parameter input tambahan yang administrator dapat mengatur saat membuat template Amazon EMR.

```
"Parameters": {
    "EmrClusterName": {
      "Type": "String",
      "Description": "EMR cluster Name."
    },
    "MasterInstanceType": {
      "Type": "String",
      "Description": "Instance type of the EMR master node.",
      "Default": "m5.xlarge",
      "AllowedValues": [
        "m5.xlarge",
        "m5.2xlarge",
        "m5.4xlarge"
      ]
    },
    "CoreInstanceType": {
      "Type": "String",
      "Description": "Instance type of the EMR core nodes.",
      "Default": "m5.xlarge",
      "AllowedValues": [
        "m5.xlarge",
        "m5.2xlarge",
        "m5.4xlarge",
        "m3.medium",
```

```
        "m3.large",
        "m3.xlarge",
        "m3.2xlarge"
      ]
    },
    "CoreInstanceCount": {
      "Type": "String",
      "Description": "Number of core instances in the EMR cluster.",
      "Default": "2",
      "AllowedValues": [
        "2",
        "5",
        "10"
      ]
    },
    "EmrReleaseVersion": {
      "Type": "String",
      "Description": "The release version of EMR to launch.",
      "Default": "emr-5.33.1",
      "AllowedValues": [
        "emr-5.33.1",
        "emr-6.4.0"
      ]
    }
  }
```

- Terakhir, lampirkan kebijakan IAM yang diperlukan untuk mengaktifkan visibilitas template CloudFormation EMR Amazon dan penyediaan sendiri kluster EMR Amazon dari notebook Studio Classic. Peran yang harus Anda tambahkan kebijakan tersebut bergantung pada apakah Studio Classic dan Amazon EMR digunakan di akun yang sama (akun tunggal) atau di akun yang berbeda (lintas akun).
  - Jika klaster EMR Amazon Anda digunakan di AWS akun yang sama dengan akun Studio Classic, lihat tab Akun Tunggal.
  - Jika klaster EMR Amazon Anda digunakan di akun yang berbeda dari AWS akun Studio Classic, lihat tab Lintas Akun.

  Untuk informasi selengkapnya tentang akses lintas akun menggunakan peran, lihat Akses sumber daya lintas akun dalam logika evaluasi kebijakan IAM atau lintas akun.

  Single account

  Lampirkan izin berikut ke peran eksekusi Studio Classic yang mengakses klaster Anda.

Daftar berikut memberikan rincian izin yang diperlukan.

- `AllowEMRTemplateDiscovery`memungkinkan penemuan untuk template EMR Amazon.
- `AllowSagemakerProjectManagement`memungkinkan terciptanya SageMaker proyek. Di Studio Classic, akses ke AWS Service Catalog diberikan melalui Proyek.
- `AllowClusterDetailsDiscovery`dan `AllowClusterDiscovery` memungkinkan penemuan dan koneksi ke cluster EMR Amazon.
- `AllowPresignedUrl`memungkinkan pembuatan URL yang telah ditandatangani sebelumnya untuk mengakses Spark UI.

Berikut ini adalah JSON komprehensif yang mencakup izin ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:CreatePersistentAppUI",
                "elasticmapreduce:DescribePersistentAppUI",
                "elasticmapreduce:GetPersistentAppUIPresignedURL",
                "elasticmapreduce:GetOnClusterAppUIPresignedURL"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:studio-region:studio-account:cluster/*"
            ]
        },
        {
            "Sid": "AllowClusterDetailsDiscovery",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:DescribeSecurityConfiguration"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:studio-region:studio-account:cluster/*"
```

```
            ]
        },
        {
            "Sid": "AllowClusterDiscovery",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowEMRTemplateDiscovery",
            "Effect": "Allow",
            "Action": [
              "servicecatalog:SearchProducts"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSagemakerProjectManagement",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateProject",
                "sagemaker:DeleteProject"
            ],
            "Resource": "arn:aws:sagemaker:studio-region:studio-account:project/*"
        },
    ]
}
```

Cross accounts

Jika klaster EMR Amazon dan Studio Classic digunakan di AWS akun terpisah, Anda mengonfigurasi izin dalam beberapa langkah.

- Pada akun kepercayaan (akun di mana Amazon EMR digunakan), buat peran IAM khusus (disebut `ASSUMABLE-ROLE` sebagai di halaman ini) dengan izin dan hubungan kepercayaan berikut.

  Untuk selengkapnya tentang membuat peran di AWS akun, lihat Membuat peran IAM (konsol).

  1. Tambahkan kebijakan IAM yang menentukan izin berikut.

- `AllowClusterDetailsDiscovery`dan `AllowClusterDiscovery` untuk memungkinkan penemuan dan koneksi ke cluster EMR Amazon.
- `AllowPresignedUrl`untuk memungkinkan pembuatan URL yang telah ditandatangani sebelumnya untuk mengakses Spark UI.

Berikut ini adalah JSON komprehensif yang mencakup izin ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:CreatePersistentAppUI",
                "elasticmapreduce:DescribePersistentAppUI",
                "elasticmapreduce:GetPersistentAppUIPresignedURL",
                "elasticmapreduce:GetOnClusterAppUIPresignedURL"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:emr-region:emr-account:cluster/*"
            ]
        },
        {
            "Sid": "AllowClusterDetailsDiscovery",
            "Effect": "Allow",
            "Action": [
               "elasticmapreduce:DescribeCluster",
               "elasticmapreduce:ListInstances",
               "elasticmapreduce:ListInstanceGroups",
               "elasticmapreduce:DescribeSecurityConfiguration"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:emr-region:emr-account:cluster/*"
            ]
        },
        {
            "Sid": "AllowClusterDiscovery",
            "Effect": "Allow",
            "Action": [
```

```
                "elasticmapreduce:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}
```

2. Untuk memberikan akun tepercaya (akun tempat Studio Classic digunakan) izin untuk mengambil peran dalam akun kepercayaan, sertakan hubungan kepercayaan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::studio-account:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Di akun tepercaya (akun tempat Studio Classic digunakan), tambahkan izin dan hubungan kepercayaan berikut ke peran eksekusi Studio Classic.

  1. Tambahkan kebijakan IAM yang menentukan izin berikut.

     - `AllowSagemakerProjectManagement`untuk memungkinkan terciptanya SageMaker proyek. Di Studio Classic, akses ke AWS Service Catalog diberikan melalui Proyek.
     - `AllowEMRTemplateDiscovery`untuk memungkinkan penemuan template EMR Amazon.

     Berikut ini adalah JSON komprehensif yang mencakup izin ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
            "Sid": "AllowSagemakerProjectManagement",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateProject",
```

```
                "sagemaker:DeleteProject"
            ],
            "Resource": "arn:aws:sagemaker:::project/*"
    },
    {
            "Sid": "AllowEMRTemplateDiscovery",
            "Effect": "Allow",
            "Action": [
              "servicecatalog:SearchProducts"
            ],
            "Resource": "*"
    }
     ]
}
```

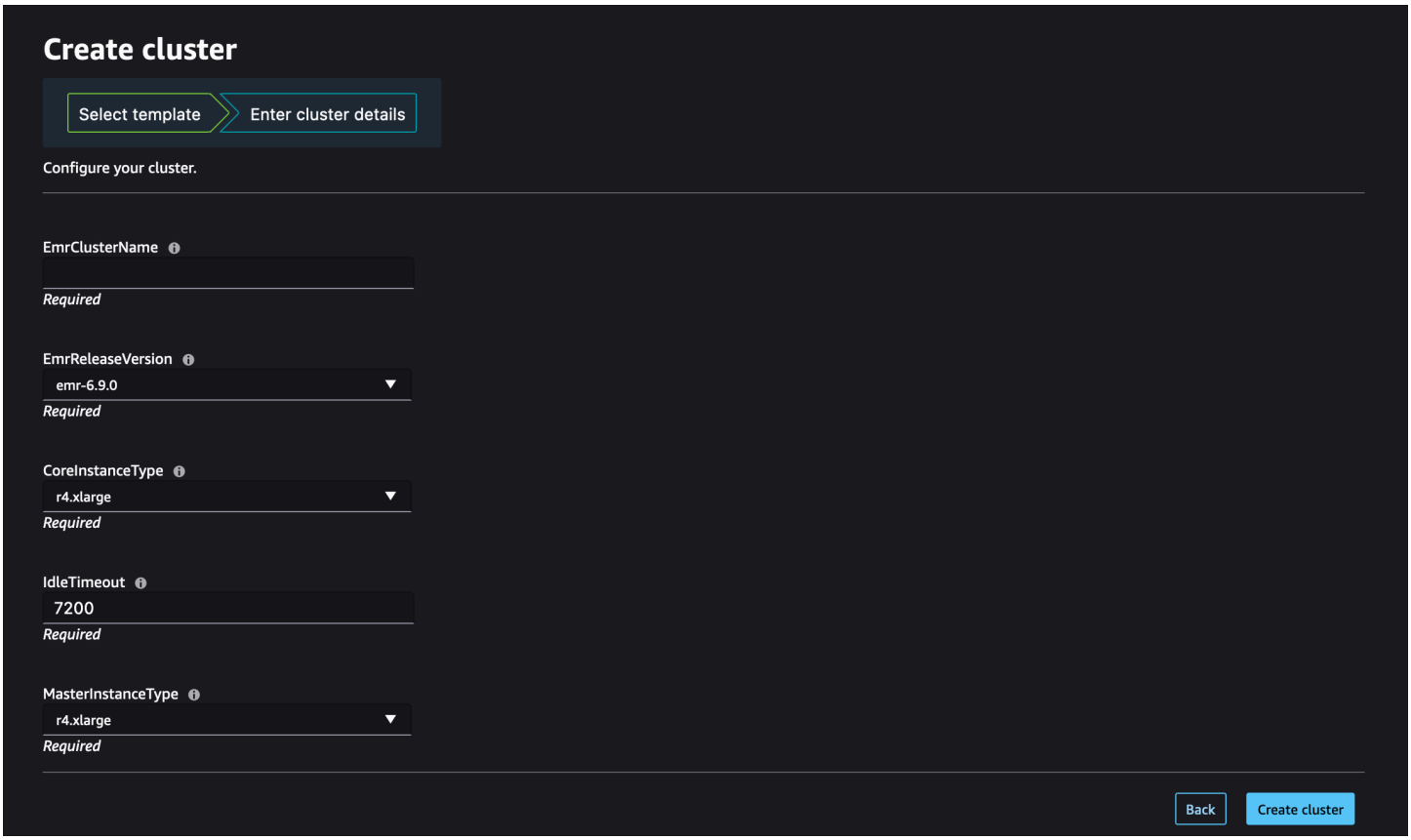
2. Untuk memberikan izin kepada peran eksekusi Studio Classic untuk mengambil alih akun kepercayaan, sertakan hubungan kepercayaan berikut. `ASSUMABLE-ROLE`

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
            "Sid": "AllowRoleAssumptionForCrossAccountDiscovery",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource":  ["arn:aws:iam::emr-account:role/ASSUMABLE-ROLE" ]
  }]
  }
```

- Terakhir, lihat Konfigurasi tambahan untuk kasus penggunaan lintas akun (untuk administrator) untuk mempelajari cara menyediakan ARN peran eksekusi `ASSUMABLE-ROLE` ke Studio Classic. ARN dimuat oleh server Studio Classic Jupyter saat diluncurkan. Peran eksekusi Studio Classic mengasumsikan peran lintas akun tersebut untuk menemukan dan terhubung ke klaster EMR Amazon di akun kepercayaan.

Setelah CloudFormation template tersedia di Amazon SageMaker Studio Classic, ilmuwan data dapat menggunakannya untuk menyediakan sendiri klaster EMR Amazon. Masing-masing yang `"Parameters"` ditentukan dalam template menjadi kotak input dalam bentuk pembuatan cluster Studio Classic, dengan yang sesuai `"AllowedValues"` muncul di menu tarik-turun.

Ilustrasi berikut menunjukkan bentuk dinamis yang dirakit dari template EMR CloudFormation Amazon untuk membuat cluster EMR Amazon di Studio Classic. SageMaker



Kunjungi Meluncurkan klaster Amazon EMR dari Studio Classic untuk mempelajari cara meluncurkan klaster dari Studio Classic menggunakan templat EMR Amazon tersebut.

Meluncurkan klaster Amazon EMR dari Studio Classic

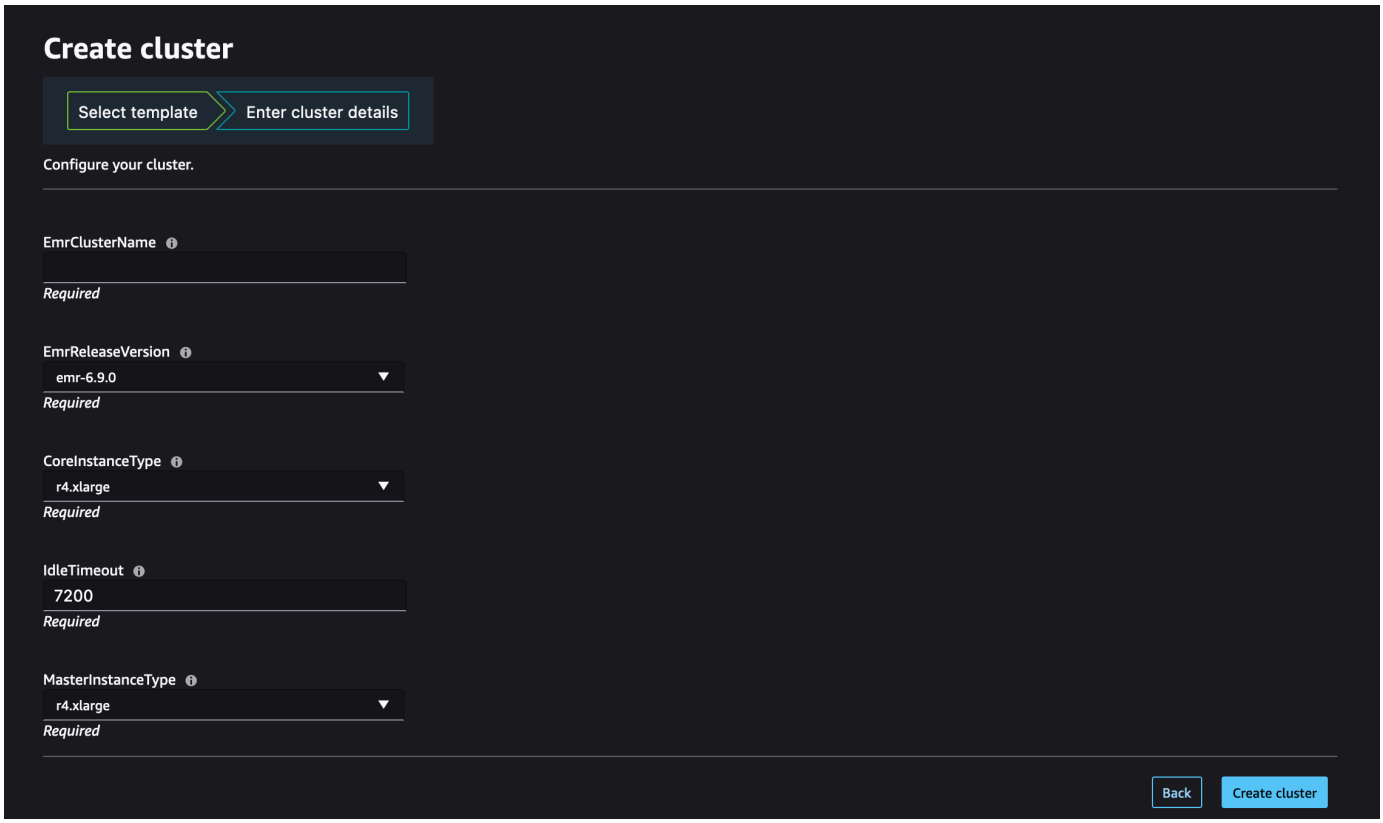
Ilmuwan data dan insinyur data dapat menyediakan sendiri klaster EMR Amazon dari Studio Classic AWS CloudFormation menggunakan templat yang dikonfigurasi oleh administrator mereka. Jika Anda seorang administrator yang ingin mengonfigurasi CloudFormation template sebagai AWS Service Catalog produk sehingga pengguna dapat membuat kluster EMR Amazon dari Studio Classic, lihat. Konfigurasikan templat EMR Amazon di AWS Service Catalog (untuk administrator)

Untuk menyediakan kluster EMR Amazon baru dari Studio Classic:

1. Pilih ikon Home
   ( )
   di panel sisi kiri Studio Classic UI, lalu pilih node Data di menu navigasi. Arahkan ke node

Clusters. Ini membuka halaman yang mencantumkan kluster EMR Amazon yang dapat Anda akses dari SageMaker Studio Classic.

2. Pilih Buat klaster. Ini membuka halaman, di area kerja utama, mencantumkan template cluster yang tersedia untuk Anda.
3. Pilih template konfigurasi cluster dengan memilih nama template. Pemilihan template mengaktifkan tombol Select template. Pilih Pilih template. Ini membuka formulir pembuatan cluster.
4. Masukkan detail klaster, seperti nama cluster dan parameter tertentu yang dapat dikonfigurasi yang ditetapkan oleh administrator Anda, lalu pilih Buat klaster. Pembuatan klaster mungkin memakan waktu beberapa menit.



Setelah klaster disediakan, UI Studio Classic menampilkan pesan Kluster telah berhasil dibuat.

Untuk terhubung ke cluster Anda, lihat Gunakan kluster EMR Amazon dari notebook Studio Classic

Gunakan kluster EMR Amazon dari notebook Studio Classic

Di bagian ini, Anda mempelajari cara menemukan, menghubungkan, atau menghentikan klaster EMR Amazon SageMaker dari notebook Studio Classic.

- Jika Anda seorang administrator, lihat Konfigurasikan kemampuan ditemukan kluster EMR Amazon (untuk administrator) untuk mengonfigurasi kemampuan untuk menemukan klaster EMR Amazon dari notebook Studio Classic. SageMaker
- Jika Anda seorang ilmuwan data atau insinyur data yang ingin menemukan kluster EMR Amazon dari notebook Studio Classic Anda, lihat. Temukan kluster EMR Amazon dari Studio Classic SageMaker
- Jika Anda seorang ilmuwan data atau insinyur data yang ingin terhubung ke kluster EMR Amazon yang ada dari notebook Studio Classic Anda, lihat. Connect ke klaster EMR Amazon dari Studio Classic SageMaker

Saat menyambungkan ke klaster EMR Amazon dari SageMaker Studio Classic, Anda dapat melakukan autentikasi ke klaster dengan Kerberos, Lightweight Directory Access Protocol (LDAP), atau menggunakan autentikasi peran IAM runtime. Metode otentikasi Anda bergantung pada konfigurasi klaster Anda. Anda dapat merujuk ke contoh ini Akses Apache Livy menggunakan Network Load Balancer pada kluster Amazon EMR berkemampuan KerberOS untuk menyiapkan kluster EMR Amazon yang menggunakan Kerberos. Atau, Anda dapat melihat CloudFormation contoh template menggunakan Kerberos atau LDAP di aws-samples/repositori. sagemaker-studio-emr GitHub

Temukan daftar perintah koneksi yang tersedia ke kluster EMR Amazon per metode otentikasi Masukkan perintah koneksi ke klaster Amazon EMR EMR secara manual untuk terhubung ke kluster EMR Amazon Anda.

Gambar dan kernel yang didukung untuk terhubung ke kluster SageMaker EMR Amazon dari Studio Classic

SageMaker Studio Classic menyediakan dukungan bawaan untuk terhubung ke kluster EMR Amazon dalam gambar dan kernel berikut:

- DataScience — Kernel Python 3
- DataScience 2.0 — Kernel Python 3
- DataScience 3.0 — Kernel Python 3
- SparkAnalytics 1.0 — SparkMagic dan PySpark kernel
- SparkAnalytics 2.0 — SparkMagic dan PySpark kernel
- SparkMagic — SparkMagic dan PySpark kernel
- PyTorch 1.8 - Python 3 kernel

- TensorFlow 2.6 — Kernel Python 3
- TensorFlow 2.11 — Kernel Python 3

Gambar dan kernel tersebut disertakan sagemaker-studio-analytics-extension, ekstensi notebook yang memungkinkan koneksi ke cluster Spark jarak jauh (Amazon EMR) melalui SparkMagicperpustakaan menggunakan Apache Livy.

Untuk terhubung ke kluster EMR Amazon menggunakan gambar bawaan lain atau gambar Anda sendiri, ikuti instruksi di. Bring Your Own Image

Bring Your Own Image

Untuk membawa gambar Anda sendiri di SageMaker Studio Classic dan memungkinkan notebook Anda terhubung ke kluster EMR Amazon, instal ekstensi sagemaker-studio-analytics-extensionberikut ke kernel Anda. Ini mendukung menghubungkan notebook SageMaker Studio Classic ke cluster Spark (Amazon EMR) melalui perpustakaan. SparkMagic

```
pip install sparkmagic
pip install sagemaker-studio-sparkmagic-lib
pip install sagemaker-studio-analytics-extension
```

Selain itu, untuk terhubung ke Amazon EMR dengan otentikasi Kerberos, Anda harus menginstal klien kinit. Tergantung pada OS Anda, perintah untuk menginstal klien kinit dapat bervariasi. Untuk membawa gambar Ubuntu (berbasis Debian), gunakan `apt-get install -y -qq krb5-user` perintah.

Untuk informasi selengkapnya tentang membawa gambar Anda sendiri di SageMaker Studio Classic, lihat Membawa SageMaker gambar Anda sendiri.

Konfigurasikan kemampuan ditemukan kluster EMR Amazon (untuk administrator)

Bagian ini memberikan detail tentang bagaimana administrator dapat mengonfigurasi kemampuan ditemukan klaster EMR Amazon yang ada dari Studio Classic. SageMaker Cluster dapat digunakan di AWS akun yang sama dengan Studio Classic (tab Akun Tunggal) atau di akun terpisah (tab Lintas Akun).

Single Account

Lampirkan izin berikut ke peran eksekusi SageMaker Studio Classic yang mengakses klaster Anda.

Daftar berikut memberikan rincian izin yang diperlukan.

- `AllowSagemakerProjectManagement`memungkinkan terciptanya SageMakerproyek. Di Studio Classic, akses ke AWS Service Catalog diberikan melalui Proyek.
- `AllowClusterDetailsDiscovery`dan `AllowClusterDiscovery` memungkinkan penemuan dan koneksi ke cluster EMR Amazon.
- `AllowPresignedUrl`memungkinkan pembuatan URL yang telah ditandatangani sebelumnya untuk mengakses Spark UI.

Berikut ini adalah JSON komprehensif yang mencakup izin ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:CreatePersistentAppUI",
                "elasticmapreduce:DescribePersistentAppUI",
                "elasticmapreduce:GetPersistentAppUIPresignedURL",
                "elasticmapreduce:GetOnClusterAppUIPresignedURL"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:region:account-id:cluster/*"
            ]
        },
        {
            "Sid": "AllowClusterDetailsDiscovery",
            "Effect": "Allow",
            "Action": [
               "elasticmapreduce:DescribeCluster",
               "elasticmapreduce:ListInstances",
               "elasticmapreduce:ListInstanceGroups",
               "elasticmapreduce:DescribeSecurityConfiguration"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:region:account-id:cluster/*"
            ]
```

```
        },
        {
            "Sid": "AllowClusterDiscovery",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSagemakerProjectManagement",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateProject",
                "sagemaker:DeleteProject"
            ],
            "Resource": "arn:aws:sagemaker:region:account-id:project/*"
        }
    ]
}
```

Cross Accounts

Jika klaster EMR Amazon dan SageMaker Studio Classic digunakan di AWS akun terpisah, Anda mengonfigurasi izin dalam beberapa langkah.

- Pada akun kepercayaan (akun di mana Amazon EMR digunakan), buat peran IAM khusus (disebut `ASSUMABLE-ROLE` sebagai di halaman ini) dengan izin dan hubungan kepercayaan berikut.

  Untuk selengkapnya tentang membuat peran di AWS akun, lihat Membuat peran IAM (konsol).

  1. Tambahkan kebijakan yang menentukan izin berikut.
     - `AllowClusterDetailsDiscovery`dan `AllowClusterDiscovery` untuk memungkinkan penemuan dan koneksi ke cluster EMR Amazon.
     - `AllowPresignedUrl`untuk memungkinkan pembuatan URL yang telah ditandatangani sebelumnya untuk mengakses Spark UI.

     Berikut ini adalah JSON komprehensif yang mencakup izin ini.

     ```
     {
         "Version": "2012-10-17",
     ```

```
    "Statement": [
        {
            "Sid": "AllowPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:CreatePersistentAppUI",
                "elasticmapreduce:DescribePersistentAppUI",
                "elasticmapreduce:GetPersistentAppUIPresignedURL",
                "elasticmapreduce:GetOnClusterAppUIPresignedURL"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:emr-region:emr-account:cluster/*"
            ]
        },
        {
            "Sid": "AllowClusterDetailsDiscovery",
            "Effect": "Allow",
            "Action": [
               "elasticmapreduce:DescribeCluster",
               "elasticmapreduce:ListInstances",
               "elasticmapreduce:ListInstanceGroups",
               "elasticmapreduce:DescribeSecurityConfiguration"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:emr-region:emr-account:cluster/*"
            ]
        },
        {
            "Sid": "AllowClusterDiscovery",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}
```

2. Untuk memberikan akun tepercaya (akun tempat akun SageMaker Studio Classic digunakan) izin untuk mengambil peran dalam akun kepercayaan, tambahkan hubungan kepercayaan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::studio-account:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Di akun tepercaya (akun tempat SageMaker Studio Classic digunakan), tambahkan hubungan kepercayaan berikut ke peran eksekusi Studio Classic.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
            "Sid": "AllowRoleAssumptionForCrossAccountDiscovery",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource":  ["arn:aws:iam::emr-account:role/ASSUMABLE-ROLE" ]
  }]
}
```

- Terakhir, lihat Konfigurasi tambahan untuk kasus penggunaan lintas akun (untuk administrator) untuk mempelajari cara menyediakan ARN peran eksekusi `ASSUMABLE-ROLE` ke Studio Classic. ARN dimuat oleh server Studio Classic Jupyter saat diluncurkan. Peran eksekusi Studio Classic mengasumsikan peran lintas akun tersebut untuk menemukan dan terhubung ke klaster EMR Amazon di akun kepercayaan.

Kunjungi Temukan kluster EMR Amazon dari Studio Classic SageMaker untuk mempelajari cara menemukan dan terhubung ke kluster EMR Amazon dari notebook Studio Classic.

Temukan kluster EMR Amazon dari Studio Classic SageMaker

Ilmuwan data dan insinyur data dapat menemukan, terhubung, dan mengelola klaster EMR Amazon dari Amazon SageMaker Studio Classic. Cluster EMR Amazon mungkin berada di akun yang AWS sama dengan Amazon SageMaker Studio Classic atau di akun yang berbeda. AWS

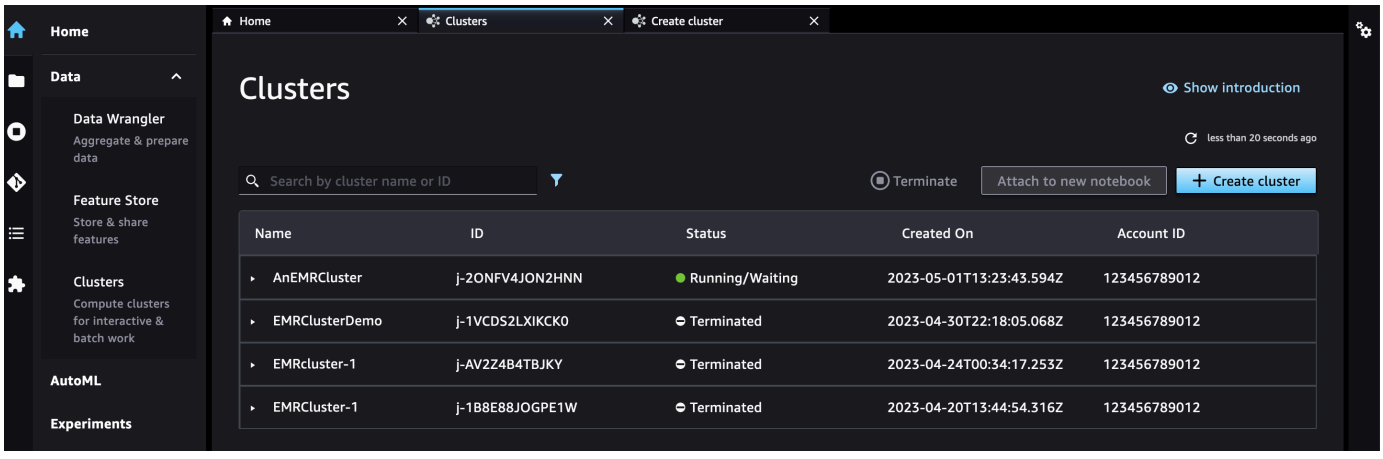
Jika administrator mengonfigurasi penemuan lintas akun klaster EMR Amazon, Anda dapat melihat daftar klaster terkonsolidasi di akun yang digunakan oleh SageMaker Studio Classic serta di AWS akun jarak jauh.

Jika Anda seorang administrator yang ingin mengatur kemampuan untuk menemukan klaster EMR Amazon dari SageMaker Studio Classic, lihat. Konfigurasikan kemampuan ditemukan kluster EMR Amazon (untuk administrator)

Untuk melihat daftar kluster EMR Amazon yang tersedia dari SageMaker Studio Classic:

1. Pilih ikon Home
   ( )
   di panel sisi kiri Studio Classic UI, lalu pilih node Data di menu navigasi.
2. Arahkan ke node Clusters. Ini membuka halaman yang mencantumkan kluster EMR Amazon yang dapat Anda akses dari SageMaker Studio Classic.

   Daftar menampilkan status masing-masing klaster. Status cluster dapat Mulai, Bootstrapping, Running/Walking, Terminating, Terminated, dan Terminated dengan kesalahan. Anda dapat memfilter cluster berdasarkan status dengan memilih ikon filter. Gambar berikut menunjukkan contoh daftar klaster.



3. Untuk terhubung ke cluster Running/Walking tertentu, lihat. Connect ke klaster EMR Amazon dari Studio Classic SageMaker

Connect ke klaster EMR Amazon dari Studio Classic SageMaker

Bagian ini menjelaskan bagaimana Anda dapat terhubung ke kluster EMR Amazon dari notebook Studio Classic saat Anda menggunakan kernel yang didukung.

Connect ke klaster Amazon EMR secara otomatis

Untuk menyambung ke klaster menggunakan UI Studio Classic, Anda dapat memulai koneksi dari daftar kluster yang diaksesTemukan kluster EMR Amazon dari Studio Classic SageMaker, atau dari buku catatan di SageMaker Studio Classic.

Untuk terhubung ke cluster tertentu dari daftar cluster

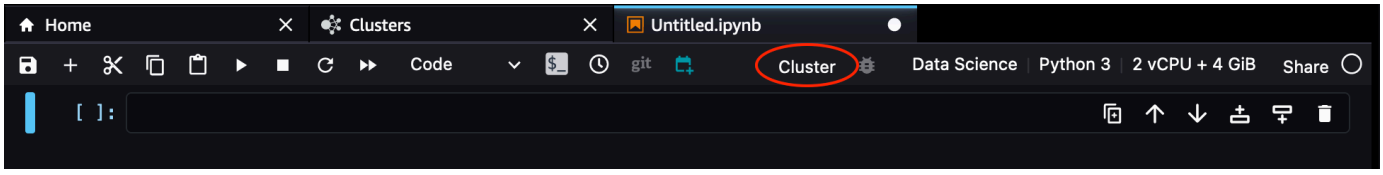
1. Pilih nama klaster dalam daftar Anda. Ini mengaktifkan tombol Lampirkan ke notebook baru.
2. Pilih Lampirkan ke buku catatan baru. Ini membuka kotak pilihan gambar dan kernel.
3. Pilih gambar dan kernel Anda, lalu pilih Pilih. Untuk daftar gambar yang didukung, lihat Gambar dan kernel yang didukung untuk terhubung ke kluster SageMaker EMR Amazon dari Studio Classic atau lihatBring Your Own Image.
4. Jika klaster yang Anda pilih tidak menggunakan otentikasi peran Kerberos, LDAP, atau runtime, Studio Classic meminta Anda untuk memilih jenis kredensialnya. Pilih dari otentikasi dasar Http atau No credentials, lalu masukkan kredensialnya, jika berlaku. Perintah koneksi mengisi sel pertama notebook Anda dan memulai koneksi dengan cluster EMR Amazon.

   Setelah koneksi berhasil, sebuah pesan mengkonfirmasi koneksi dan dimulainya aplikasi Spark.

Atau, Anda dapat terhubung ke klaster dari notebook.

1. Pilih Cluster di bagian atas buku catatan Anda.

   Cluster hanya terlihat ketika Anda menggunakan kernel dari Gambar dan kernel yang didukung untuk terhubung ke kluster SageMaker EMR Amazon dari Studio Classic atau dariBring Your Own Image. Jika Anda tidak dapat melihat Cluster di bagian atas buku catatan Anda, pastikan administrator Anda telah mengonfigurasi kemampuan untuk dapat ditemukan klaster Anda dan beralih ke kernel yang didukung.



Ini membuka daftar klaster yang tersedia.

2. Pilih klaster yang ingin Anda hubungkan, lalu pilih Connect.
3. Jika Anda mengonfigurasi kluster EMR Amazon untuk mendukung peran IAM runtime dan administrator memuat peran Anda sebelumnya dalam konfigurasi peran eksekusi JSON, Anda dapat memilih peran akses EMR Amazon dari menu tarik-turun peran eksekusi EMR Amazon. Jika peran Anda tidak dimuat sebelumnya, Studio Classic akan menggunakan peran eksekusi Studio Classic secara default. Untuk informasi tentang menggunakan peran runtime dengan Amazon EMR, lihat. Connect ke klaster Amazon EMR dari Studio Classic menggunakan peran IAM runtime Saat Anda terhubung ke cluster, Studio Classic menambahkan blok kode ke sel aktif untuk membuat koneksi.

   Jika tidak, jika cluster yang Anda pilih tidak menggunakan otentikasi peran Kerberos, LDAP, atau runtime, Studio Classic meminta Anda untuk memilih jenis kredensialnya. Anda dapat memilih otentikasi dasar HTTP atau No credential.
4. Sel aktif mengisi dan berjalan. Sel ini berisi perintah koneksi untuk terhubung ke klaster Amazon EMR Anda.

   Setelah koneksi berhasil, sebuah pesan mengkonfirmasi koneksi dan dimulainya aplikasi Spark.

Masukkan perintah koneksi ke klaster Amazon EMR EMR secara manual

Anda dapat terhubung secara manual ke klaster EMR Amazon dari notebook Studio Classic baik aplikasi dan klaster Studio Classic berada di akun yang sama atau tidak. AWS

Untuk setiap jenis autentikasi berikut, gunakan perintah yang ditentukan untuk menyambung secara manual ke klaster Anda dari notebook Studio Classic Anda.

- Kerberos

  Tambahkan `--assumable-role-arn` argumen jika Anda memerlukan akses EMR Amazon lintas akun. Tambahkan `--verify-certificate` argumen jika Anda terhubung ke cluster Anda dengan HTTPS.

```
%load_ext sagemaker_studio_analytics_extension.magics
%sm_analytics emr connect --cluster-id cluster_id \
--auth-type Kerberos --language python
[--assumable-role-arn EMR_access_role_ARN ]
```

```
[--verify-certificate /home/user/certificateKey.pem]
```

- LDAP

  Tambahkan `--assumable-role-arn` argumen jika Anda memerlukan akses EMR Amazon lintas akun. Tambahkan `--verify-certificate` argumen jika Anda terhubung ke cluster Anda dengan HTTPS.

```
%load_ext sagemaker_studio_analytics_extension.magics
%sm_analytics emr connect --cluster-id cluster_id \
--auth-type Basic_Access --language python
[--assumable-role-arn EMR_access_role_ARN ]
[--verify-certificate /home/user/certificateKey.pem]
```

- NoAuth

  Tambahkan `--assumable-role-arn` argumen jika Anda memerlukan akses EMR Amazon lintas akun. Tambahkan `--verify-certificate` argumen jika Anda terhubung ke cluster Anda dengan HTTPS.

```
%load_ext sagemaker_studio_analytics_extension.magics
%sm_analytics emr connect --cluster-id cluster_id \
--auth-type None --language python
[--assumable-role-arn EMR_access_role_ARN ]
[--verify-certificate /home/user/certificateKey.pem]
```

- Peran IAM runtime

  Tambahkan `--assumable-role-arn` argumen jika Anda memerlukan akses EMR Amazon lintas akun. Tambahkan `--verify-certificate` argumen jika Anda terhubung ke cluster Anda dengan HTTPS.

  Untuk informasi selengkapnya tentang menghubungkan ke klaster EMR Amazon menggunakan peran IAM runtime, lihat. Connect ke klaster Amazon EMR dari Studio Classic menggunakan peran IAM runtime

```
%load_ext sagemaker_studio_analytics_extension.magics
%sm_analytics emr connect --cluster-id cluster_id \
--auth-type Basic_Access \
--emr-execution-role-arn arn:aws:iam::studio_account_id:role/emr-execution-role-name
[--assumable-role-arn EMR_access_role_ARN]
```

```
[--verify-certificate /home/user/certificateKey.pem]
```

Connect ke klaster Amazon EMR melalui HTTPS

Jika Anda telah mengonfigurasi klaster EMR Amazon Anda dengan enkripsi transit diaktifkan dan server Apache Livy untuk HTTPS dan ingin Studio Classic berkomunikasi dengan Amazon EMR menggunakan HTTPS, Anda perlu mengonfigurasi Studio Classic untuk mengakses kunci sertifikat Anda.

Untuk sertifikat yang ditandatangani sendiri atau ditandatangani oleh Otoritas Sertifikat lokal (CA), Anda dapat melakukannya dalam dua langkah:

1. Unduh file PEM sertifikat Anda ke sistem file lokal Anda menggunakan salah satu opsi berikut:
   - Fungsi unggah file bawaan Jupyter.
   - Sebuah sel notebook.
   - Skrip konfigurasi siklus hidup (LCC).

     Untuk informasi tentang cara menggunakan skrip LCC, lihat Menyesuaikan Instans Notebook Menggunakan Skrip Konfigurasi Siklus Hidup
2. Aktifkan validasi sertifikat dengan memberikan jalur ke sertifikat Anda dalam `--verify-certificate` argumen perintah koneksi Anda.

```
%sm_analytics emr connect --cluster-id cluster_id \
    --verify-certificate /home/user/certificateKey.pem ...
```

Untuk sertifikat yang diterbitkan CA publik, tetapkan validasi sertifikat dengan menetapkan `--verify-certificate` parameter sebagai`true`.

Atau, Anda dapat menonaktifkan validasi sertifikat dengan menetapkan `--verify-certificate` parameter sebagai`false`.

Anda dapat menemukan daftar perintah koneksi yang tersedia ke cluster EMR Amazon di. Masukkan perintah koneksi ke klaster Amazon EMR EMR secara manual

Connect ke klaster Amazon EMR dari Studio Classic menggunakan peran IAM runtime

Saat tersambung ke klaster EMR Amazon dari notebook Amazon SageMaker Studio Classic, Anda dapat menelusuri daftar peran IAM secara visual, yang dikenal sebagai peran runtime, dan

memilihnya dengan cepat. Selanjutnya, semua pekerjaan Apache Spark, Apache Hive, atau Presto yang dibuat dari notebook Studio Classic hanya mengakses data dan sumber daya yang diizinkan oleh kebijakan yang dilampirkan pada peran runtime. Selain itu, saat data diakses dari data lake yang dikelolaAWS Lake Formation, Anda dapat menerapkan akses tingkat tabel dan tingkat kolom menggunakan kebijakan yang dilampirkan pada peran runtime.

Dengan kemampuan ini, Anda dan rekan tim Anda dapat terhubung ke cluster yang sama, masing-masing menggunakan peran runtime yang dicakup dengan izin yang sesuai dengan tingkat akses individual Anda ke data. Sesi Anda juga terisolasi satu sama lain di cluster bersama. Dengan kemampuan ini untuk mengontrol akses halus ke data pada cluster bersama yang sama, Anda dapat menyederhanakan penyediaan kluster EMR Amazon, mengurangi biaya operasional, dan menghemat biaya.

Untuk mencoba fitur baru ini, lihat Menerapkan kontrol akses data berbutir halus dengan dan AWS Lake Formation Amazon EMR dari Amazon Studio Classic. SageMaker Posting blog ini membantu Anda mengatur lingkungan demo tempat Anda dapat mencoba menggunakan peran runtime yang telah dikonfigurasi sebelumnya untuk terhubung ke kluster EMR Amazon.

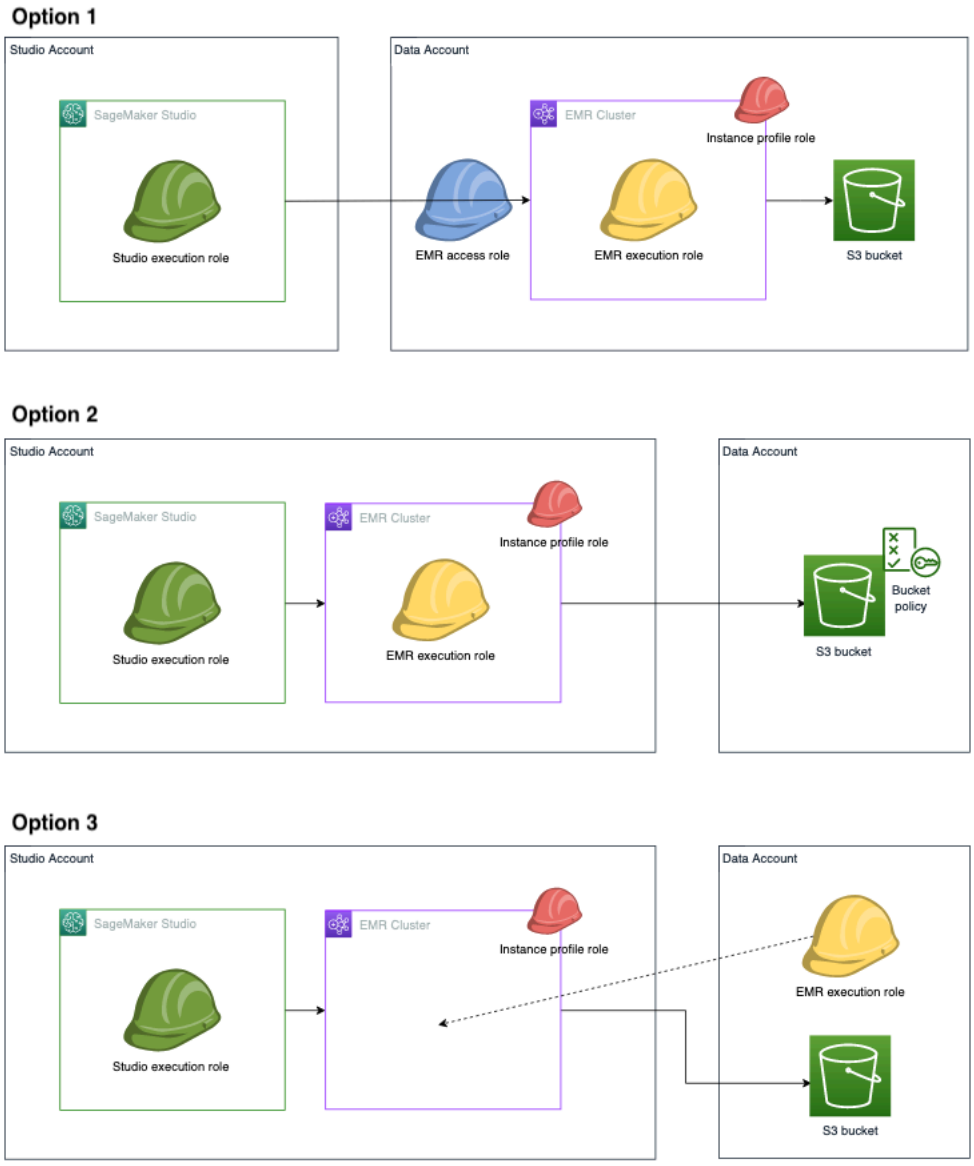
Prasyarat

Sebelum memulai, pastikan Anda memenuhi persyaratan berikut:

- Gunakan Amazon EMR versi 6.9 atau lebih tinggi.
- Gunakan JupyterLab versi 3 dalam konfigurasi aplikasi server Studio Classic Jupyter. Versi ini mendukung koneksi Studio Classic ke cluster EMR Amazon menggunakan peran runtime.
- Izinkan penggunaan peran runtime dalam konfigurasi keamanan klaster Anda. Untuk informasi selengkapnya, lihat Peran runtime untuk langkah-langkah Amazon EMR.
- Buat buku catatan dengan salah satu kernel yang tercantum diGunakan kluster EMR Amazon dari notebook Studio Classic.
- Pastikan Anda meninjau instruksi Siapkan Studio Classic untuk menggunakan peran IAM runtime untuk mengonfigurasi peran runtime dengan Studio Classic.

Skenario koneksi lintas akun

Autentikasi peran runtime mendukung berbagai skenario koneksi lintas akun saat data berada di luar akun Studio Classic. Gambar berikut menunjukkan tiga cara berbeda untuk menetapkan klaster EMR Amazon, data, dan bahkan peran eksekusi EMR Amazon antara Studio Classic dan akun data:



Di opsi 1, klaster EMR Amazon dan peran eksekusi EMR Amazon berada di akun data terpisah dari akun Studio Classic Anda. Anda menentukan kebijakan izin peran akses EMR Amazon terpisah yang memberikan izin ke peran eksekusi Studio Classic Anda untuk mengambil peran akses Amazon EMR. Peran akses EMR Amazon kemudian memanggil API EMR Amazon `GetClusterSessionCredentials` atas nama peran eksekusi Studio Classic Anda, memberi Anda akses ke klaster.

Di opsi 2, klaster EMR Amazon dan peran eksekusi EMR Amazon ada di akun Studio Classic Anda. Peran eksekusi Studio Classic Anda memiliki izin untuk menggunakan Amazon EMR API `GetClusterSessionCredentials` untuk mendapatkan akses ke klaster Anda. Untuk mengakses

bucket Amazon S3, berikan izin akses bucket Amazon S3 lintas akun peran eksekusi EMR Amazon S3 lintas akun — Anda memberikan izin ini dalam kebijakan bucket Amazon S3 Anda.

Di opsi 3, kluster EMR Amazon Anda ada di akun Studio Classic Anda, dan peran eksekusi EMR Amazon ada di akun data. Peran eksekusi Studio Classic Anda memiliki izin untuk menggunakan Amazon EMR API `GetClusterSessionCredentials` untuk mendapatkan akses ke klaster Anda. Tambahkan peran eksekusi EMR Amazon ke dalam konfigurasi peran eksekusi JSON. Kemudian Anda dapat memilih peran di UI saat memilih klaster. Untuk detail tentang cara menyiapkan file JSON konfigurasi peran eksekusi, lihatMemuat peran eksekusi Anda ke Studio Classic.

Siapkan Studio Classic untuk menggunakan peran IAM runtime

Untuk membuat otentikasi peran runtime untuk kluster EMR Amazon Anda, konfigurasikan kebijakan IAM, jaringan, dan peningkatan kegunaan yang diperlukan. Penyiapan Anda bergantung pada apakah Anda menangani pengaturan lintas akun jika klaster EMR Amazon, peran eksekusi EMR Amazon, atau keduanya, berada di luar akun Amazon Studio Classic Anda. SageMaker Diskusi berikut memandu Anda melalui kebijakan untuk menginstal, cara mengonfigurasi jaringan untuk mengizinkan lalu lintas antar akun, dan file konfigurasi lokal yang akan disiapkan untuk mengotomatiskan koneksi EMR Amazon Anda.

Konfigurasikan autentikasi peran runtime saat klaster EMR Amazon dan Studio Classic berada di akun yang sama

Jika klaster EMR Amazon Anda berada di akun Studio Classic, tambahkan kebijakan dasar untuk menyambung ke klaster EMR Amazon Anda dan setel izin untuk memanggil Amazon EMR `GetClusterSessionCredentials` API, yang memberi Anda akses ke klaster. Selesaikan langkah-langkah berikut untuk menambahkan izin yang diperlukan ke kebijakan eksekusi Studio Classic Anda:

1. Tambahkan kebijakan IAM yang diperlukan untuk terhubung ke kluster EMR Amazon. Untuk detailnya, lihat Temukan kluster EMR Amazon dari Studio Classic SageMaker.
2. Berikan izin untuk memanggil API EMR Amazon `GetClusterSessionCredentials` saat Anda meneruskan satu atau beberapa peran eksekusi EMR Amazon yang diizinkan yang ditentukan dalam kebijakan.
3. (Opsional) Berikan izin untuk meneruskan peran IAM yang mengikuti konvensi penamaan yang ditentukan pengguna.
4. (Opsional) Berikan izin untuk mengakses kluster EMR Amazon yang ditandai dengan string yang ditentukan pengguna tertentu.

5. Jika Anda tidak ingin memanggil perintah koneksi Amazon EMR secara manual, instal file SageMaker konfigurasi di Amazon EFS lokal Anda dan pilih peran yang akan digunakan saat Anda memilih kluster Amazon EMR Anda. Untuk detail tentang cara memuat peran IAM Anda, lihat. Memuat peran eksekusi Anda ke Studio Classic

Contoh kebijakan berikut memungkinkan peran eksekusi EMR Amazon milik kelompok pemodelan dan pelatihan untuk dipanggil. `GetClusterSessionCredentials` Selain itu, pemegang polis dapat mengakses kluster EMR Amazon yang ditandai dengan string atau. `modeling` `training`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "elasticmapreduce:GetClusterSessionCredentials",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "elasticmapreduce:ExecutionRoleArn": [
                        "arn:aws:iam::123456780910:role/emr-execution-role-ml-
modeling*",
                        "arn:aws:iam::123456780910:role/emr-execution-role-ml-
training*"
                    ],
                    "elasticmapreduce:ResourceTag/group": [
                        "*modeling*",
                        "*training*"
                    ]
                }
            }
        }
    ]
}
```

Konfigurasikan autentikasi peran runtime saat klaster dan Studio Classic Anda berada di akun yang berbeda

Jika klaster EMR Amazon Anda tidak ada di akun Studio Classic, izinkan peran eksekusi Studio Classic Anda untuk mengambil peran akses Amazon EMR lintas akun sehingga Anda dapat

terhubung ke klaster. Selesaikan langkah-langkah berikut untuk menyiapkan konfigurasi lintas akun Anda:

1. Buat kebijakan izin peran eksekusi Studio Classic agar peran eksekusi dapat mengambil peran akses Amazon EMR. Kebijakan berikut adalah contoh:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAssumeCrossAccountEMRAccessRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::emr_account_id:role/emr-access-role-name"
        }
    ]
}
```

2. Buat kebijakan kepercayaan untuk menentukan ID akun Studio Classic mana yang dipercaya untuk mengambil peran akses EMR Amazon. Kebijakan berikut adalah contoh:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountSageMakerExecutionRoleToAssumeThisRole",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::studio_account_id:role/studio_execution_role"
      },
      "Action": "sts:AssumeRole"
    }
}
```

3. Buat kebijakan izin akses Amazon EMR, yang memberikan izin yang diperlukan untuk melaksanakan peran eksekusi Amazon EMR untuk melaksanakan tugas yang diinginkan di klaster. Konfigurasikan peran akses EMR Amazon untuk memanggil API `GetClusterSessionCredentials` dengan peran eksekusi EMR Amazon yang ditentukan dalam kebijakan izin peran akses. Kebijakan berikut adalah contoh:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "AllowCallingEmrGetClusterSessionCredentialsAPI",
            "Effect": "Allow",
            "Action": "elasticmapreduce:GetClusterSessionCredentials",
            "Resource": "",
            "Condition": {
                "StringLike": {
                    "elasticmapreduce:ExecutionRoleArn": [
                        "arn:aws:iam::emr_account_id:role/emr-execution-role-name"
                    ]
                }
            }
        }
    ]
}
```

4. Siapkan jaringan lintas akun sehingga lalu lintas dapat bergerak bolak-balik antar akun Anda. Untuk instruksi terpandu, lihat Mengatur jaringan di posting blog Membuat dan mengelola Cluster EMR Amazon dari SageMaker Studio Classic untuk menjalankan beban kerja Spark dan ML interaktif — Bagian 2. Langkah-langkah dalam posting blog membantu Anda menyelesaikan tugas berikut:

   a. VPC-peer akun Studio Classic Anda dan akun EMR Amazon Anda untuk membuat koneksi.

   b. Tambahkan rute secara manual ke tabel rute subnet pribadi di kedua akun. Ini memungkinkan pembuatan dan koneksi cluster EMR Amazon dari akun Studio Classic ke subnet pribadi akun jarak jauh.

   c. Siapkan grup keamanan yang dilampirkan ke domain Studio Classic Anda untuk mengizinkan lalu lintas keluar dan grup keamanan simpul utama EMR Amazon untuk mengizinkan lalu lintas TCP masuk dari grup keamanan instans Studio Classic.

5. Jika Anda tidak ingin memanggil perintah koneksi Amazon EMR secara manual, instal file SageMaker konfigurasi di Amazon EFS lokal Anda sehingga Anda dapat memilih peran yang akan digunakan saat memilih cluster EMR Amazon Anda. Untuk detail tentang cara memuat peran IAM Anda, lihat. Memuat peran eksekusi Anda ke Studio Classic

Konfigurasikan akses Lake Formation

Saat mengakses data dari data lake yang dikelola olehAWS Lake Formation, Anda dapat menerapkan akses tingkat tabel dan tingkat kolom menggunakan kebijakan yang dilampirkan pada

peran runtime Anda. Untuk mengonfigurasi izin akses Lake Formation, lihat Mengintegrasikan Amazon EMR dengan. AWS Lake Formation

Memuat peran eksekusi Anda ke Studio Classic

Jika Anda tidak ingin memanggil perintah koneksi Amazon EMR secara manual, Anda dapat menginstal file SageMaker konfigurasi di Amazon EFS lokal Anda sehingga Anda dapat memilih peran eksekusi yang akan digunakan saat memilih cluster EMR Amazon Anda.

Untuk menulis file konfigurasi untuk peran eksekusi EMR Amazon, kaitkan Menggunakan konfigurasi siklus hidup dengan Amazon Studio Classic SageMaker (LCC) ke aplikasi server Jupyter. Atau, Anda dapat menulis atau memperbarui file konfigurasi dan memulai ulang server Jupyter dengan perintah:. `restart-jupyter-server`

Cuplikan berikut adalah contoh skrip bash LCC yang dapat Anda terapkan jika aplikasi dan klaster Studio Classic Anda berada di akun yang sama:

```
#!/bin/bash

set -eux

FILE_DIRECTORY="/home/sagemaker-user/.sagemaker-analytics-configuration-DO_NOT_DELETE"
FILE_NAME="emr-configurations-DO_NOT_DELETE.json"
FILE="$FILE_DIRECTORY/$FILE_NAME"

mkdir -p $FILE_DIRECTORY

cat << 'EOF' > "$FILE"
{
    "emr-execution-role-arns":
    {
      "123456789012": [
          "arn:aws:iam::123456789012:role/emr-execution-role-1",
          "arn:aws:iam::123456789012:role/emr-execution-role-2"
      ]
    }
}
EOF
```

Jika aplikasi dan klaster Studio Classic Anda berada di akun yang berbeda, tentukan peran akses EMR Amazon yang dapat menggunakan klaster. Dalam contoh kebijakan berikut, 123456789012

adalah ARN untuk akun klaster EMR Amazon, dan 212121212121 dan 434343434343 adalah ARN untuk peran akses EMR Amazon yang diizinkan.

```
#!/bin/bash

set -eux

FILE_DIRECTORY="/home/sagemaker-user/.sagemaker-analytics-configuration-DO_NOT_DELETE"
FILE_NAME="emr-configurations-DO_NOT_DELETE.json"
FILE="$FILE_DIRECTORY/$FILE_NAME"

mkdir -p $FILE_DIRECTORY

cat << 'EOF' > "$FILE"
{
    "emr-execution-role-arns":
    {
      "123456789012": [
          "arn:aws:iam::212121212121:role/emr-execution-role-1",
          "arn:aws:iam::434343434343:role/emr-execution-role-2"
      ]
    }
}
EOF

# add your cross-account EMR access role
FILE_DIRECTORY="/home/sagemaker-user/.cross-account-configuration-DO_NOT_DELETE"
FILE_NAME="emr-discovery-iam-role-arns-DO_NOT_DELETE.json"
FILE="$FILE_DIRECTORY/$FILE_NAME"

mkdir -p $FILE_DIRECTORY

cat << 'EOF' > "$FILE"
{
    "123456789012": "arn:aws:iam::123456789012:role/cross-account-emr-access-role"
}
EOF
```

Mengakhiri klaster EMR Amazon dari Studio Classic

Prosedur berikut menunjukkan cara menghentikan klaster Amazon EMR dari notebook Studio Classic.

Untuk mengakhiri klaster dalam **`Running`** status, navigasikan ke daftar kluster EMR Amazon yang tersedia.

1. Di SageMaker Studio Classic, pilih ikon Home
   ( )
   di panel sisi kiri Studio Classic UI, lalu pilih simpul Data di menu navigasi.
2. Arahkan ke node Clusters. Ini membuka halaman yang mencantumkan kluster EMR Amazon yang dapat Anda akses dari SageMaker Studio Classic.
3. Pilih nama cluster yang ingin Anda akhiri, lalu pilih Terminate.
4. Ini membuka jendela konfirmasi yang memberi tahu Anda bahwa pekerjaan atau data yang tertunda di klaster Anda akan hilang secara permanen setelah penghentian. Konfirmasikan dengan memilih Hentikan lagi.

Akses Spark UI dari Studio Classic

Bagian berikut memberikan instruksi untuk mengakses UI Spark dari notebook SageMaker Studio Classic. UI Spark memungkinkan Anda memantau dan men-debug Pekerjaan Spark yang dikirimkan untuk berjalan di Amazon EMR dari notebook Studio Classic. Tunneling SSH dan URL presigned adalah dua cara untuk mengakses UI Spark.

Siapkan tunneling SSH untuk akses Spark UI

Untuk mengatur tunneling SSH untuk mengakses UI Spark, ikuti salah satu dari dua opsi di bagian ini.

Opsi untuk mengatur tunneling SSH:

- Opsi 1: Buat terowongan SSH ke simpul utama menggunakan penerusan port lokal
- Opsi 2, bagian 1: Buat terowongan SSH ke simpul utama menggunakan penerusan port dinamis

  Opsi 2, bagian 2: Konfigurasi pengaturan proksi untuk melihat situs web yang di-host di simpul utama

Untuk informasi tentang melihat antarmuka web yang di-host di klaster Amazon EMR, lihat Lihat antarmuka Web yang di-host di Klaster Amazon EMR. Anda juga dapat mengunjungi konsol EMR Amazon Anda untuk mendapatkan akses ke UI Spark.

> **Note**
>
> Anda dapat mengatur terowongan SSH meskipun URL yang telah ditetapkan sebelumnya tidak tersedia untuk Anda.

URL yang ditandatangani sebelumnya

Untuk membuat URL sekali klik yang dapat mengakses Spark UI di Amazon EMR dari notebook SageMaker Studio Classic, Anda harus mengaktifkan izin IAM berikut. Pilih opsi yang berlaku untuk Anda:

- Untuk klaster EMR Amazon yang berada di akun yang sama dengan notebook SageMaker Studio Classic: Tambahkan izin berikut ke peran eksekusi IAM SageMaker Studio Classic.
- Untuk klaster EMR Amazon yang berada di akun berbeda (bukan buku catatan SageMaker Studio Classic): Tambahkan izin berikut ke peran lintas akun yang Anda buat. Temukan kluster EMR Amazon dari Studio Classic SageMaker

> **Note**
>
> Anda dapat mengakses URL yang telah ditetapkan sebelumnya dari konsol di wilayah berikut:
>
> - Wilayah AS Timur (Virginia Utara)
> - Wilayah AS Barat (California Utara)
> - Wilayah Kanada (Pusat)
> - Wilayah Eropa (Frankfurt)
> - Wilayah Eropa (Stockholm)
> - Wilayah Eropa (Irlandia)
> - Wilayah Eropa (London)
> - Wilayah Eropa (Paris)
> - Wilayah Asia Pacific (Tokyo)
> - Wilayah Asia Pasifik (Seoul)
> - Wilayah Asia Pasifik (Sydney)
> - Wilayah Asia Pasifik (Mumbai)

- Wilayah Asia Pasifik (Singapura)
- Amerika Selatan (Sao Paulo)

Kebijakan berikut memberikan akses ke URL yang telah ditetapkan sebelumnya untuk peran eksekusi Anda.

```
{
            "Sid": "AllowPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:CreatePersistentAppUI",
                "elasticmapreduce:DescribePersistentAppUI",
                "elasticmapreduce:GetPersistentAppUIPresignedURL",
                "elasticmapreduce:GetOnClusterAppUIPresignedURL"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:region:account-id:cluster/*"
            ]
}
```

Panduan dan whitepaper

Blog berikut menggunakan studi kasus prediksi sentimen untuk tinjauan film untuk menggambarkan proses pelaksanaan alur kerja pembelajaran mesin yang lengkap. Ini termasuk persiapan data, pemantauan pekerjaan Spark, dan pelatihan serta penerapan model ML untuk mendapatkan prediksi langsung dari notebook Studio Classic Anda.

- Buat dan kelola kluster EMR Amazon dari SageMaker Studio Classic untuk menjalankan beban kerja Spark dan ML interaktif.
- Untuk memperluas kasus penggunaan ke konfigurasi lintas akun tempat SageMaker Studio Classic dan klaster EMR Amazon Anda digunakan di akun AWS terpisah, lihat Membuat dan mengelola klaster EMR Amazon SageMaker dari Studio Classic untuk menjalankan beban kerja Spark dan ML interaktif - Bagian 2.

Lihat juga:

- Panduan konfigurasi Access Apache Livy menggunakan Network Load Balancer pada kluster Amazon EMR berkemampuan KerberOS.
- AWSwhitepaper untuk praktik terbaik SageMaker Studio Classic.

Konfigurasi tambahan untuk kasus penggunaan lintas akun (untuk administrator)

Untuk mengaktifkan penemuan klaster di seluruh akun, administrator perlu menyediakan ARN peran IAM lintas akun ke peran eksekusi Studio Classic. SageMaker SageMaker Peran eksekusi Studio Classic mengasumsikan peran jarak jauh untuk menemukan dan terhubung ke klaster EMR Amazon di akun kepercayaan. ARN dari peran ini dimuat oleh server Jupyter Studio Classic saat diluncurkan.

Anda dapat menentukan informasi ini dengan dua cara.

- Tulis peran jarak jauh ini dalam file bernama yang `emr-discovery-iam-role-arns-DO_NOT_DELETE.json` ditempatkan di direktori `.cross-account-configuration-DO_NOT_DELETE` di direktori home Anda yang terletak di volume penyimpanan Amazon EFS yang digunakan oleh SageMaker Studio Classic.
- Atau, Anda dapat mengotomatiskan proses ini dengan menggunakan skrip Lifecycle Configuration (LCC). Anda dapat melampirkan LCC ke Domain Anda atau profil pengguna tertentu. Skrip LCC yang Anda gunakan harus berupa JupyterServer konfigurasi. Untuk informasi selengkapnya tentang cara membuat skrip LCC, lihat Menggunakan Konfigurasi Siklus Hidup dengan Studio Classic.

Berikut ini adalah contoh skrip LCC. Untuk memodifikasi skrip, ganti `ASSUMABLE-ROLE` dan `emr-account` dengan nama peran dan ID akun jarak jauh Anda, masing-masing. Jumlah akun silang dibatasi hingga lima.

```
# This script creates the file that informs SageMaker Studio Classic that the role
 "arn:aws:iam::emr-account:role/ASSUMABLE-ROLE" in remote account "emr-account" must be
 assumed to list and describe Amazon EMR clusters in the remote account.

#!/bin/bash

set -eux

FILE_DIRECTORY="/home/sagemaker-user/.cross-account-configuration-DO_NOT_DELETE"
FILE_NAME="emr-discovery-iam-role-arns-DO_NOT_DELETE.json"
FILE="$FILE_DIRECTORY/$FILE_NAME"
```

```
mkdir -p $FILE_DIRECTORY

cat > "$FILE" <<- "EOF"
{
  emr-cross-account1: "arn:aws:iam::emr-cross-account1:role/ASSUMABLE-ROLE",
  emr-cross-account2: "arn:aws:iam::emr-cross-account2:role/ASSUMABLE-ROLE"
}
EOF
```

Setelah LCC berjalan dan file ditulis, server membaca file `/home/sagemaker-user/.cross-account-configuration-DO_NOT_DELETE/emr-discovery-iam-role-arns-DO_NOT_DELETE.json` dan menyimpan ARN lintas akun.

Memecahkan masalah

Berikut ini adalah kesalahan umum yang mungkin terjadi saat menghubungkan atau menggunakan kluster EMR Amazon dari notebook Studio Classic.

Memecahkan masalah koneksi Livy yang macet atau gagal

Berikut ini adalah masalah konektivitas Livy yang mungkin terjadi saat menggunakan kluster EMR Amazon dari notebook Studio Classic.

- Kluster EMR Amazon Anda mengalami kesalahan out-of-memory.

  Alasan yang mungkin untuk koneksi Livy melalui `sparkmagic` hang atau kegagalan adalah jika cluster EMR Amazon Anda mengalami kesalahan. out-of-memory

  Secara default, parameter konfigurasi Java dari driver Apache Spark,`spark.driver.defaultJavaOptions`, diatur ke. `-XX:OnOutOfMemoryError='kill -9 %p'` Ini berarti bahwa tindakan default yang diambil ketika program driver bertemu `OutOfMemoryError` adalah untuk menghentikan program driver dengan mengirimkan sinyal SIGKILL. Ketika driver Apache Spark dihentikan, koneksi Livy apa pun melalui `sparkmagic` itu tergantung pada driver tersebut hang atau gagal. Ini karena driver Spark bertanggung jawab untuk mengelola sumber daya aplikasi Spark, termasuk penjadwalan tugas dan eksekusi. Tanpa driver, aplikasi Spark tidak dapat berfungsi, dan setiap upaya untuk berinteraksi dengannya gagal.

  Jika Anda mencurigai klaster Spark Anda mengalami masalah memori, Anda dapat memeriksa log EMR Amazon. Kontainer mati karena out-of-memory kesalahan biasanya keluar dengan kode`137`. Dalam kasus seperti itu, Anda perlu me-restart aplikasi Spark dan membuat koneksi Livy baru untuk melanjutkan interaksi dengan cluster Spark.

Anda dapat merujuk ke artikel basis pengetahuan Bagaimana cara mengatasi kesalahan “Wadah yang dibunuh oleh YARN karena melebihi batas memori” di Spark on Amazon EMR? AWS re:Postuntuk mempelajari berbagai strategi dan parameter yang dapat digunakan untuk mengatasi suatu out-of-memory masalah.

Sebaiknya tinjau Panduan Praktik Terbaik EMR Amazon untuk praktik terbaik dan panduan penyetelan dalam menjalankan beban kerja Apache Spark di kluster EMR Amazon Anda.

- Waktu sesi Livy Anda habis saat menghubungkan ke klaster Amazon EMR untuk pertama kalinya.

  Saat Anda awalnya terhubung ke kluster EMR Amazon menggunakan sagemaker-studio-analytics-extension, yang memungkinkan koneksi ke cluster Spark (Amazon EMR) jarak jauh melalui SparkMagicpustaka menggunakan Apache Livy, Anda mungkin mengalami kesalahan batas waktu koneksi:

  ```
  An error was encountered: Session 0 did not start up in 60 seconds.
  ```

  Jika klaster EMR Amazon Anda memerlukan inisialisasi aplikasi Spark saat membuat koneksi, ada kemungkinan lebih besar untuk melihat kesalahan batas waktu koneksi.

  Untuk mengurangi kemungkinan mendapatkan batas waktu saat menghubungkan ke kluster EMR Amazon menggunakan Livy melalui ekstensi analitik`sagemaker-studio-analytics-extension`, `0.0.19` versi dan yang lebih baru mengganti batas waktu sesi server default `120` ke detik, bukan `sparkmagic` default detik. `60`

  Sebaiknya upgrade ekstensi Anda `0.0.18` dan lebih cepat dengan menjalankan perintah upgrade berikut.

  ```
  pip install --upgrade sagemaker-studio-analytics-extension
  ```

  Perhatikan bahwa saat menyediakan konfigurasi batas waktu khusus di`sparkmagic`, `sagemaker-studio-analytics-extension` menghormati penggantian ini. Namun, menyetel batas waktu sesi ke `60` detik secara otomatis memicu batas waktu sesi server default dalam hitungan detik. `120` `sagemaker-studio-analytics-extension`

Siapkan data menggunakan Sesi AWS Glue Interaktif

AWS GlueSesi Interaktif adalah lingkungan runtime Apache Spark sesuai permintaan, tanpa server yang dapat digunakan oleh ilmuwan dan insinyur data untuk membangun, menguji, dan menjalankan persiapan data dan aplikasi analitik dengan cepat.

Anda dapat memulai sesi AWS Glue interaktif dengan memulai notebook SageMaker Studio Classic. Saat membuat notebook Studio Classic, pilih built-in `Glue PySpark` atau `Glue Spark` kernel. Ini secara otomatis memulai sesi Spark interaktif tanpa server. Anda tidak perlu menyediakan atau mengelola klaster atau infrastruktur komputasi apa pun. Setelah inisialisasi, Anda dapat menjelajahi, mengeksekusi kueri kompleksAWS Glue Data Catalog, dan menganalisis dan menyiapkan data secara interaktif menggunakan Spark dalam notebook Studio Classic Anda. Anda kemudian dapat menggunakan data yang disiapkan untuk membuat, melatih, menyetel, dan menerapkan model menggunakan alat ML yang dibuat khusus dalam Studio Classic. SageMaker

Sebelum memulai sesi AWS Glue interaktif di SageMaker Studio Classic, Anda perlu menetapkan peran dan kebijakan yang sesuai. Selain itu, Anda mungkin perlu memberikan akses ke sumber daya tambahan, seperti bucket Amazon S3, yang mungkin memerlukan kebijakan tambahan. Untuk informasi selengkapnya tentang kebijakan IAM yang diperlukan dan tambahan, lihatIzin untuk Sesi AWS Glue Interaktif di SageMaker Studio Classic.

SageMaker Studio Classic menyediakan konfigurasi default untuk sesi AWS Glue interaktif Anda, namun, Anda dapat menggunakan AWS Glue katalog lengkap perintah ajaib Jupyter untuk lebih menyesuaikan lingkungan Anda. Untuk informasi tentang sihir Jupyter default dan tambahan yang dapat Anda gunakan dalam sesi AWS Glue interaktif Anda, lihat. Konfigurasikan sesi AWS Glue interaktif Anda di SageMaker Studio Classic

Gambar dan kernel yang didukung untuk menghubungkan ke sesi AWS Glue interaktif adalah sebagai berikut:

- Gambar: SparkAnalytics 1.0, SparkAnalytics 2.0
- Kernel: Glue Python [PySpark dan Ray] dan Glue Spark

Prasyarat:

SparkAnalytics Gambar yang Anda pilih untuk meluncurkan AWS Glue sesi Anda di Studio Classic adalah kombinasi dari dua kerangka kerja - kerangka SparkMagic kerja (digunakan dengan Amazon EMR), dan. AWS Glue Untuk alasan ini, prasyarat untuk kedua kerangka kerja berlaku. Namun, Anda tidak perlu mengatur cluster EMR Amazon jika Anda hanya berencana untuk menggunakan

Sesi AWS Glue Interaktif. Sebelum Anda memulai sesi AWS Glue interaktif pertama Anda di Studio Classic, selesaikan yang berikut ini:

- Lengkapi prasyarat yang diperlukan untuk menggunakan gambar. SparkMagic Untuk daftar prasyarat, lihat bagian Prasyarat di Siapkan Data pada Skala dengan Notebook Studio Klasik.
- Buat peran eksekusi dengan izin untuk keduanya AWS Glue dan SageMaker Studio Classic. Tambahkan kebijakan terkelola`AwsGlueSessionUserRestrictedServiceRole`, dan buat kebijakan khusus yang menyertakan izin `sts:GetCallerIdentityiam:GetRole`, dan`IAM:Passrole`. Untuk instruksi tentang cara membuat izin yang diperlukan, lihatIzin untuk Sesi AWS Glue Interaktif di SageMaker Studio Classic.
- Buat SageMaker Domain dengan peran eksekusi yang Anda buat. Untuk petunjuk tentang cara membuat Domain, lihatOrientasi khusus menggunakan IAM.

Memulai Sesi AWS Glue Interaktif

Dalam panduan ini, Anda mempelajari cara memulai sesi AWS Glue interaktif di SageMaker Studio Classic, dan mengelola lingkungan Anda dengan sihir Jupyter.

Izin untuk Sesi AWS Glue Interaktif di SageMaker Studio Classic

Bagian ini mencantumkan kebijakan yang diperlukan untuk menjalankan sesi AWS Glue interaktif di Studio Classic dan menjelaskan cara mengaturnya. Secara khusus, ini merinci cara:

- Lampirkan kebijakan `AwsGlueSessionUserRestrictedServiceRole` terkelola ke peran SageMaker eksekusi Anda.
- Buat kebijakan kustom inline pada peran SageMaker eksekusi Anda.
- Ubah hubungan kepercayaan dari peran SageMaker eksekusi Anda.

Untuk melampirkan kebijakan **`AwsGlueSessionUserRestrictedServiceRole`** terkelola ke peran eksekusi Anda

1. Buka konsol IAM.
2. Pilih Peran di panel sisi kiri.
3. Temukan peran eksekusi Studio Classic Anda. Pilih nama peran untuk mengakses halaman ringkasan peran.
4. Di bawah tab Izin, pilih Lampirkan kebijakan dari menu tarik-turun Tambahkan Izin.

5. Pilih kotak centang di samping kebijakan `AwsGlueSessionUserRestrictedServiceRole` terkelola.
6. Pilih Lampirkan kebijakan.

   Halaman ringkasan menampilkan kebijakan terkelola yang baru ditambahkan.

Untuk membuat kebijakan kustom inline pada peran eksekusi Anda

1. Pilih Buat kebijakan sebaris di menu tarik-turun Tambahkan Izin.
2. Pilih tab JSON.
3. Salin dan tempel dalam kebijakan berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "unique_statement_id",

            "Effect": "Allow",
            "Action": [
  "iam:GetRole",
                "iam:PassRole",
                "sts:GetCallerIdentity"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Pilih Tinjau kebijakan.
5. Masukkan Nama dan pilih Buat kebijakan.

   Halaman ringkasan menunjukkan kebijakan kustom Anda yang baru ditambahkan.

Untuk mengubah hubungan kepercayaan dari peran eksekusi Anda

1. Pilih tab Hubungan kepercayaan.
2. Pilih Edit kebijakan kepercayaan.

3. Salin dan tempel dalam kebijakan berikut.

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "glue.amazonaws.com",
                "sagemaker.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
}
```

4. Pilih Perbarui kebijakan.

Anda dapat menambahkan peran dan kebijakan tambahan jika Anda perlu mengakses AWS sumber daya lain. Untuk deskripsi peran dan kebijakan tambahan yang dapat Anda sertakan, lihat Sesi interaktif dengan IAM dalam AWS Glue dokumentasi.

Memeriksa metadata

Tag biasanya digunakan untuk melacak dan mengalokasikan biaya, mengontrol akses ke sesi Anda, mengisolasi sumber daya Anda, dan banyak lagi. Untuk mempelajari tentang menambahkan metadata ke AWS sumber daya Anda menggunakan penandaan, atau untuk detail tentang kasus penggunaan umum, lihat. Informasi tambahan

Anda dapat mengaktifkan propagasi otomatis AWS tag ke sesi AWS Glue interaktif baru yang dibuat dari dalam UI Studio Classic. Ketika sesi AWS Glue interaktif dibuat dari SageMaker Studio Classic, setiap tag yang ditentukan pengguna yang dilampirkan ke profil pengguna atau ruang bersama dibawa ke sesi AWS Glue interaktif baru. Selain itu, SageMaker Studio Classic secara otomatis menambahkan dua tag internal yang AWS dihasilkan ((`sagemaker:user-profile-arn`dan`sagemaker:domain-arn`) atau (`sagemaker:shared-space-arn`dan`sagemaker:domain-arn`)) ke sesi AWS Glue interaktif baru yang dibuat dari UI Studio Classic. Anda dapat menggunakan tag ini untuk mengumpulkan biaya di setiap Domain, profil pengguna, atau spasi.

Aktifkan propagasi tanda

Untuk mengaktifkan propagasi otomatis tag ke sesi AWS Glue interaktif baru, tetapkan izin berikut untuk peran SageMaker eksekusi Anda dan peran IAM yang terkait dengan sesi Anda: AWS Glue

> **Note**
>
> Secara default, peran yang terkait dengan sesi AWS Glue interaktif sama dengan peran SageMaker eksekusi. Anda dapat menentukan peran eksekusi yang berbeda untuk sesi AWS Glue interaktif dengan menggunakan perintah `%iam_role` ajaib. Untuk informasi tentang perintah ajaib Jupyter yang tersedia untuk mengonfigurasi sesi AWS Glue interaktif, lihat. Konfigurasikan sesi AWS Glue interaktif Anda di SageMaker Studio Classic

- Pada peran SageMaker eksekusi Anda: Buat kebijakan inline baru, dan tempel file JSON berikut. Kebijakan memberikan izin peran eksekusi untuk mendeskripsikan (`DescribeUserProfile`,`DescribeSpace`,`DescribeDomain`) dan mencantumkan tag (`ListTag`) yang ditetapkan pada profil pengguna, spasi bersama, dan SageMaker Domain.

```
{
    "Effect": "Allow",
    "Action": [
        "sagemaker:ListTags"
    ],
    "Resource": [
        "arn:aws:sagemaker:*:*:user-profile/*",
        "arn:aws:sagemaker:*:*:space/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "sagemaker:DescribeUserProfile"
    ],
    "Resource": [
        "arn:aws:sagemaker:*:*:user-profile/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
```

```
            "sagemaker:DescribeSpace"
        ],
        "Resource": [
            "arn:aws:sagemaker:*:*:space/*"
        ]
    }
    {
        "Effect": "Allow",
        "Action": [
            "sagemaker:DescribeDomain"
        ],
        "Resource": [
            "arn:aws:sagemaker:*:*:domain/*"
        ]
    }
```

- Pada peran IAM AWS Glue sesi Anda: Buat kebijakan inline baru, dan tempel file JSON berikut. Kebijakan memberikan izin peran Anda untuk melampirkan tag (`TagResource`) ke sesi Anda, atau mengambil daftar tag ()`GetTags`.

```
{
    "Effect": "Allow",
    "Action": [
        "glue:TagResource",
        "glue:GetTags"
    ],
    "Resource": [
        "arn:aws:glue:*:*:session/*"
    ]
}
```

> **Note**
>
> - Kegagalan yang terjadi saat menerapkan izin tersebut tidak mencegah pembuatan sesi AWS Glue interaktif. Anda dapat menemukan detail tentang alasan kegagalan di CloudWatchlog SageMaker Studio Classic.
> - Anda harus memulai ulang kernel sesi AWS Glue interaktif Anda untuk menyebarkan pembaruan nilai tag.

Penting untuk dicatat poin-poin berikut:

- Setelah tag dilampirkan ke sesi, tag tidak dapat dihapus dengan propagasi.

  Anda dapat menghapus tag dari sesi AWS Glue interaktif langsung melaluiAWS CLI, AWS Glue API, atau https://console.aws.amazon.com/sagemaker/. Misalnya, dengan menggunakanAWS CLI, Anda dapat menghapus tag dengan memberikan ARN sesi dan kunci tag yang ingin Anda hapus sebagai berikut:

  ```
  aws glue untag-resource \
  --resource-arn arn:aws:glue:region:account-id:session:session-name \
  --tags-to-remove tag-key1,tag-key2
  ```

- SageMaker Studio Classic menambahkan dua tag internal AWS yang dihasilkan ((`sagemaker:user-profile-arn`dan`sagemaker:domain-arn`) atau (`sagemaker:shared-space-arn`dan`sagemaker:domain-arn`)) ke sesi AWS Glue interaktif baru yang dibuat dari UI Studio Classic. Tag tersebut dihitung terhadap batas 50 tag yang ditetapkan pada semua AWS sumber daya. Keduanya `sagemaker:user-profile-arn` dan `sagemaker:shared-space-arn` berisi ID Domain tempat mereka berada.
- Tombol tag dimulai dengan `aws:AWS:`, atau kombinasi huruf besar dan kecil sebagai awalan untuk kunci tidak disebarkan dan dicadangkan untuk digunakan. AWS

Informasi tambahan

Untuk informasi selengkapnya tentang penandaan, lihat sumber daya berikut.

- Untuk mempelajari cara menambahkan metadata ke AWS sumber daya Anda dengan penandaan, lihat Menandai sumber daya. AWS
- Untuk informasi tentang melacak biaya menggunakan tag, lihat Analisis biaya di Praktik Terbaik Administrasi SageMaker Studio Classic.
- Untuk informasi tentang mengontrol akses AWS Glue berdasarkan kunci tag, lihat ABAC dengan AWS Glue.

Luncurkan sesi AWS Glue interaktif Anda di SageMaker Studio Classic

Setelah membuat peran, kebijakan, dan SageMaker Domain, Anda dapat meluncurkan sesi AWS Glue interaktif di SageMaker Studio Classic.

Untuk diluncurkan AWS Glue di SageMaker Studio Classic

1. Buat SageMaker Domain. Untuk petunjuk tentang cara membuat Domain baru, lihatIkhtisar SageMaker Domain Amazon.
2. Masuk ke SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
3. Pilih Control Panel di panel sisi kiri.
4. Di menu tarik-turun Luncurkan Aplikasi di sebelah nama pengguna, pilih Studio.
5. Di tampilan Jupyter, pilih File, lalu New, lalu Notebook.
6. Di menu dropdown Gambar, pilih SparkAnalytics 1.0 atau SparkAnalytics 2.0. Di menu dropdown kernel, pilih Glue Spark atau Glue Python [dan Ray]. PySpark Pilih Pilih.
7. (opsional) Gunakan sihir Jupyter untuk menyesuaikan lingkungan Anda. Untuk informasi selengkapnya tentang sihir Jupyter, lihat. Konfigurasikan sesi AWS Glue interaktif Anda di SageMaker Studio Classic
8. Mulai menulis skrip pemrosesan data Spark Anda.

Konfigurasikan sesi AWS Glue interaktif Anda di SageMaker Studio Classic

> **Note**
>
> Semua konfigurasi ajaib dibawa ke sesi berikutnya selama masa pakai AWS Glue kernel.

Anda dapat menggunakan sihir Jupyter dalam sesi AWS Glue interaktif Anda untuk memodifikasi parameter sesi dan konfigurasi Anda. Sihir adalah perintah pendek yang diawali dengan % di awal sel Jupyter yang menyediakan cara cepat dan mudah untuk membantu Anda mengontrol lingkungan Anda. Dalam sesi AWS Glue interaktif Anda, sihir berikut diatur untuk Anda secara default:

Sihir	Nilai default
`%glue_version`	Fitur SQL baru
`%iam_role`	*`peran eksekusi yang melekat pada SageMaker Domain Anda`*
`%region`	Fungsi SQL baru

Anda dapat menggunakan sihir untuk lebih menyesuaikan lingkungan Anda. Misalnya, jika Anda ingin mengubah jumlah pekerja yang dialokasikan ke pekerjaan Anda dari default lima menjadi 10, Anda dapat menentukan`%number_of_workers 10`. Jika Anda ingin mengonfigurasi sesi Anda untuk berhenti setelah 10 menit waktu idle, bukan 2880 default, Anda dapat menentukan. `%idle_timeout 10`

Semua sihir Jupyter yang saat ini tersedia juga AWS Glue tersedia di SageMaker Studio Classic. Untuk daftar lengkap AWS Glue sihir yang tersedia, lihat Mengonfigurasi sesi AWS Glue interaktif untuk notebook Jupyter dan AWS Glue Studio Classic.

AWS GlueHarga Sesi Interaktif

Saat Anda menggunakan Sesi AWS Glue Interaktif di notebook SageMaker Studio Classic, Anda akan dikenakan biaya secara terpisah untuk penggunaan sumber daya AWS Glue dan notebook Studio Classic.

AWSbiaya untuk Sesi AWS Glue Interaktif berdasarkan berapa lama sesi aktif dan jumlah Unit Pemrosesan Data (DPU) yang digunakan. Anda akan ditagih tarif per jam untuk jumlah DPU yang digunakan untuk menjalankan beban kerja Anda, ditagih dengan penambahan satu detik. AWS Glue Sesi Interaktif menetapkan default lima DPU dan membutuhkan minimal dua DPU. Ada juga durasi penagihan minimum satu menit untuk setiap sesi interaktif. Untuk melihat contoh AWS Glue harga dan harga, atau untuk memperkirakan biaya Anda menggunakan Kalkulator AWS Harga, lihat AWS Glueharga.

Notebook SageMaker Studio Classic berjalan pada instans Amazon EC2 dan Anda dikenakan biaya untuk jenis instans yang Anda pilih, berdasarkan durasi penggunaan. Studio Classic memberi Anda jenis instans EC2 default `ml-t3-medium` saat Anda memilih `SparkAnalytics` gambar dan kernel terkait. Anda dapat mengubah jenis instans untuk notebook Studio Classic agar sesuai dengan beban kerja Anda. Untuk informasi tentang harga SageMaker Studio Classic, lihat SageMakerHarga Amazon.

Memproses data

Untuk menganalisis data dan mengevaluasi model pembelajaran mesin di Amazon SageMaker, gunakan Amazon SageMaker Processing. Dengan Processing, Anda dapat menggunakan pengalaman yang disederhanakan dan dikelola SageMaker untuk menjalankan beban kerja pemrosesan data Anda, seperti rekayasa fitur, validasi data, evaluasi model, dan interpretasi model. Anda juga dapat menggunakan Amazon SageMaker Processing API selama fase eksperimen dan setelah kode diterapkan dalam produksi untuk mengevaluasi kinerja.

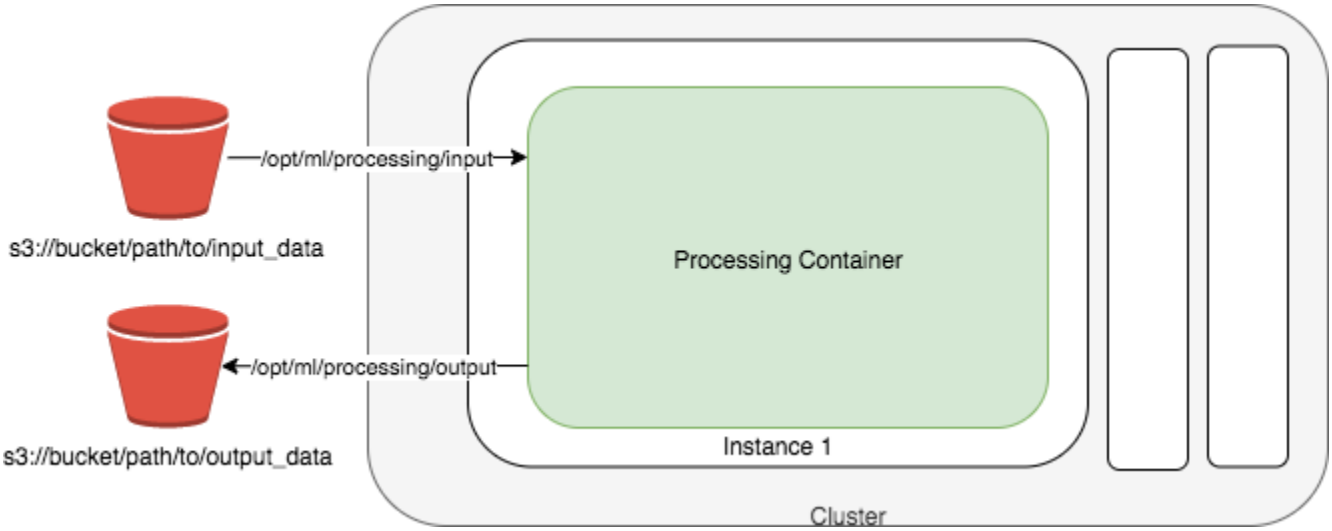


Diagram sebelumnya menunjukkan bagaimana Amazon SageMaker memutar pekerjaan Pemrosesan. Amazon SageMaker mengambil skrip Anda, menyalin data Anda dari Amazon Simple Storage Service (Amazon S3), dan kemudian menarik wadah pemrosesan. Gambar kontainer pemrosesan dapat berupa gambar SageMaker bawaan Amazon atau gambar khusus yang Anda berikan. Infrastruktur dasar untuk pekerjaan Pemrosesan sepenuhnya dikelola oleh Amazon SageMaker. Sumber daya cluster disediakan selama durasi pekerjaan Anda, dan dibersihkan saat pekerjaan selesai. Output dari pekerjaan Pemrosesan disimpan di bucket Amazon S3 yang Anda tentukan.

> **Note**
>
> Data input Anda harus disimpan dalam bucket Amazon S3. Atau, Anda dapat menggunakan Amazon Athena atau Amazon Redshift sebagai sumber input.

> **Tip**
>
> Untuk mempelajari praktik terbaik untuk komputasi terdistribusi pelatihan pembelajaran mesin (ML) dan pekerjaan pemrosesan secara umum, lihatKomputasi terdistribusi dengan praktik SageMaker terbaik.

Gunakan Notebook Sampel SageMaker Pemrosesan Amazon

Kami menyediakan dua contoh notebook Jupyter yang menunjukkan cara melakukan pra-pemrosesan data, evaluasi model, atau keduanya.

Untuk contoh buku catatan yang menunjukkan cara menjalankan skrip scikit-learn untuk melakukan prapemrosesan data serta pelatihan serta evaluasi model dengan SageMaker Python SDK for Processing, lihat scikit-learn Processing. Notebook ini juga menunjukkan cara menggunakan container kustom Anda sendiri untuk menjalankan beban kerja pemrosesan dengan pustaka Python Anda dan dependensi spesifik lainnya.

Untuk contoh buku catatan yang menunjukkan cara menggunakan Amazon SageMaker Processing untuk melakukan pra-pemrosesan data terdistribusi dengan Spark, lihat Pemrosesan Terdistribusi (Spark). Notebook ini juga menunjukkan cara melatih model regresi menggunakan XGBoost pada dataset yang telah diproses sebelumnya.

Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan sampel ini, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMaker Contoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Pantau Pekerjaan SageMaker Pemrosesan Amazon dengan CloudWatch Log dan Metrik

Amazon SageMaker Processing menyediakan CloudWatch log dan metrik Amazon untuk memantau pekerjaan pemrosesan. CloudWatch menyediakan CPU, GPU, memori, memori GPU, dan metrik disk, dan pencatatan peristiwa. Lihat informasi yang lebih lengkap di Pantau Amazon SageMaker dengan Amazon CloudWatch dan Log SageMaker Acara Amazon dengan Amazon CloudWatch.

Pemrosesan Data dengan Apache Spark

Apache Spark adalah mesin analisis terpadu untuk pemrosesan data skala besar. Amazon SageMakermenyediakan image Docker bawaan yang mencakup Apache Spark dan dependensi lain yang diperlukan untuk menjalankan pekerjaan pemrosesan data terdistribusi. DenganAmazon SageMaker Python SDK, Anda dapat dengan mudah menerapkan transformasi data dan mengekstrak fitur (rekayasa fitur) menggunakan kerangka Spark. Untuk informasi tentang menggunakan SageMaker Python SDK untuk menjalankan pekerjaan pemrosesan Spark, lihatPemrosesan Data dengan Sparkdi dalamAmazon SageMaker Python SDK.

Repositori kode yang berisi kode sumber dan Dockerfiles untuk gambar Spark tersedia diGitHub.

Menjalankan Job Pemrosesan Spark

Anda dapat menggunakan`sagemaker.spark.PySparkProcessor`atau`sagemaker.spark.SparkJarProcessor`ke untuk menjalankan aplikasi Spark Anda di dalam pekerjaan pemrosesan. Catatan Anda dapat mengatur MaxRuntimeInSeconds ke batas runtime maksimum 5 hari. Sehubungan dengan waktu eksekusi, dan jumlah instance yang digunakan, beban kerja percikan sederhana melihat hubungan linier dekat antara jumlah instance vs waktu hingga penyelesaian.

Contoh kode berikut menunjukkan cara menjalankan pekerjaan pemrosesan yang memanggil Anda PySpark skrip`preprocess.py`.

```
from sagemaker.spark.processing import PySparkProcessor

spark_processor = PySparkProcessor(
    base_job_name="spark-preprocessor",
    framework_version="2.4",
    role=role,
    instance_count=2,
    instance_type="ml.m5.xlarge",
    max_runtime_in_seconds=1200,
)

spark_processor.run(
    submit_app="preprocess.py",
    arguments=['s3_input_bucket', bucket,
               's3_input_key_prefix', input_prefix,
               's3_output_bucket', bucket,
```

```
                's3_output_key_prefix', output_prefix]
)
```

Untuk tampilan yang mendalam, lihat Pemrosesan Data Terdistribusi dengan Apache Spark dan SageMaker PemrosesanNotebook contoh.

Jika Anda tidak menggunakanAmazon SageMakerPython SDKdan salah satu kelas Prosesor untuk mengambil gambar pra-dibangun, Anda dapat mengambil gambar-gambar ini sendiri. Klaster SageMaker image Docker bawaan disimpan di Amazon Elastic Container Registry (Amazon ECR). Untuk daftar lengkap image Docker yang tersedia, lihatimage yang tersediadokumen.

Untuk mempelajari selengkapnya tentang menggunakan SageMaker Python SDK dengan Container Processing, lihatAmazon SageMaker Python SDK.

Pemrosesan Data dengan scikit-learn

Untuk contoh notebook yang menunjukkan cara menjalankan skrip scikit-learn menggunakan image Docker yang disediakan dan dikelola oleh SageMaker untuk memproses data dan mengevaluasi model, lihat scikit-learn Processing. Untuk menggunakan notebook ini, Anda perlu menginstal SageMaker Python SDK untuk Processing.

Notebook ini menjalankan pekerjaan pemrosesan menggunakan `SKLearnProcessor` kelas dari SageMaker Python SDK untuk menjalankan skrip scikit-learn yang Anda berikan. Script preprocesses data, melatih model menggunakan pekerjaan SageMaker pelatihan, dan kemudian menjalankan pekerjaan pemrosesan untuk mengevaluasi model terlatih. Pekerjaan pemrosesan memperkirakan bagaimana model diharapkan untuk tampil dalam produksi.

Untuk mempelajari lebih lanjut tentang menggunakan SageMaker Python SDK dengan Container Processing, lihat SageMakerPython SDK. Untuk daftar lengkap image Docker bawaan yang tersedia untuk memproses pekerjaan, lihat Jalur Registri Docker dan Kode Contoh.

Contoh kode berikut menunjukkan bagaimana notebook menggunakan `SKLearnProcessor` untuk menjalankan skrip scikit-learn Anda sendiri menggunakan gambar Docker yang disediakan dan dikelola olehSageMaker, bukan gambar Docker Anda sendiri.

```
from sagemaker.sklearn.processing import SKLearnProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput

sklearn_processor = SKLearnProcessor(framework_version='0.20.0',
                                     role=role,
```

```
                                    instance_type='ml.m5.xlarge',
                                    instance_count=1)

sklearn_processor.run(code='preprocessing.py',
                      inputs=[ProcessingInput(
                        source='s3://path/to/my/input-data.csv',
                        destination='/opt/ml/processing/input')],
                      outputs=[ProcessingOutput(source='/opt/ml/processing/output/
train'),
                               ProcessingOutput(source='/opt/ml/processing/output/
validation'),
                               ProcessingOutput(source='/opt/ml/processing/output/
test')]
                     )
```

Untuk memproses data secara parallel menggunakan Scikit-Learn on Amazon SageMaker Processing, Anda dapat memecahkan objek input dengan kunci S3 dengan menyetel `s3_data_distribution_type='ShardedByS3Key'` di dalam sebuah `ProcessingInput` sehingga setiap instans menerima jumlah objek input yang hampir sama.

Pemrosesan Data dengan Prosesor Kerangka

A `FrameworkProcessor` dapat menjalankan tugas Pemrosesan dengan kerangka kerja pembelajaran mesin tertentu, memberi Anda wadah yang SageMaker dikelola Amazon untuk kerangka kerja pembelajaran mesin mana pun yang Anda pilih. `FrameworkProcessor`menyediakan wadah premade untuk kerangka kerja pembelajaran mesin berikut: Hugging Face, MXNet,, dan XGBoost. PyTorch TensorFlow

`FrameworkProcessor`Kelas ini juga memberi Anda kustomisasi atas konfigurasi kontainer. `FrameworkProcessor`Kelas mendukung menentukan direktori sumber `source_dir` untuk skrip pemrosesan dan dependensi Anda. Dengan kemampuan ini, Anda dapat memberikan akses prosesor ke beberapa skrip dalam direktori alih-alih hanya menentukan satu skrip. `FrameworkProcessor`juga mendukung termasuk `requirements.txt` file di `source_dir` untuk menyesuaikan pustaka Python untuk diinstal dalam wadah.

Untuk informasi selengkapnya tentang `FrameworkProcessor` kelas dan metode serta parameternya, lihat FrameworkProcessordi Amazon SageMaker Python SDK.

Untuk melihat contoh penggunaan `FrameworkProcessor` untuk setiap kerangka kerja pembelajaran mesin yang didukung, lihat topik berikut.

Topik

Prosesor Kerangka Kerja Hugging Face

Hugging Face adalah penyedia open source model natural language processing (NLP). SDK SageMaker Python `HuggingFaceProcessor` di Amazon memberi Anda kemampuan untuk menjalankan pekerjaan pemrosesan dengan skrip Hugging Face. Saat Anda menggunakannya`HuggingFaceProcessor`, Anda dapat memanfaatkan wadah Docker buatan Amazon dengan lingkungan Hugging Face yang dikelola sehingga Anda tidak perlu membawa wadah sendiri.

Contoh kode berikut menunjukkan bagaimana Anda dapat menggunakan `HuggingFaceProcessor` untuk menjalankan pekerjaan Processing Anda menggunakan image Docker yang disediakan dan dikelola oleh SageMaker. Perhatikan bahwa ketika Anda menjalankan pekerjaan, Anda dapat menentukan direktori yang berisi skrip dan dependensi Anda dalam `source_dir` argumen, dan Anda dapat memiliki `requirements.txt` file yang terletak di dalam `source_dir` direktori Anda yang menentukan dependensi untuk skrip pemrosesan Anda. SageMaker Pemrosesan menginstal dependensi di `requirements.txt` dalam wadah untuk Anda.

```
from sagemaker.huggingface import HuggingFaceProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker import get_execution_role

#Initialize the HuggingFaceProcessor
hfp = HuggingFaceProcessor(
    role=get_execution_role(),
    instance_count=1,
    instance_type='ml.g4dn.xlarge',
    transformers_version='4.4.2',
    pytorch_version='1.6.0',
    base_job_name='frameworkprocessor-hf'
)
```

```
#Run the processing job
hfp.run(
    code='processing-script.py',
    source_dir='scripts',
    inputs=[
        ProcessingInput(
            input_name='data',
            source=f's3://{BUCKET}/{S3_INPUT_PATH}',
            destination='/opt/ml/processing/input/data/'
        )
    ],
    outputs=[
        ProcessingOutput(output_name='train', source='/opt/ml/processing/output/
train/', destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'),
        ProcessingOutput(output_name='test', source='/opt/ml/processing/output/test/',
 destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'),
        ProcessingOutput(output_name='val', source='/opt/ml/processing/output/val/',
 destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}')
    ]
)
```

Jika Anda memiliki `requirements.txt` file, itu harus berupa daftar pustaka yang ingin Anda instal di wadah. Jalur untuk `source_dir` dapat berupa jalur URI relatif, absolut, atau Amazon S3. Namun, jika Anda menggunakan URI Amazon S3, maka itu harus mengarah ke file tar.gz. Anda dapat memiliki beberapa skrip di direktori yang Anda tentukan`source_dir`. Untuk mempelajari selengkapnya tentang `HuggingFaceProcessor` kelas, lihat Hugging Face Estimator di Amazon SageMaker Python SDK.

Prosesor MxNet Framework

Apache MXNet adalah kerangka pembelajaran mendalam open-source yang biasa digunakan untuk melatih dan menyebarkan jaringan saraf. SDK SageMaker Python `MXNetProcessor` di Amazon memberi Anda kemampuan untuk menjalankan pekerjaan pemrosesan dengan skrip MXNet. Saat Anda menggunakan`MXNetProcessor`, Anda dapat memanfaatkan wadah Docker buatan Amazon dengan lingkungan MXNet terkelola sehingga Anda tidak perlu membawa wadah Anda sendiri.

Contoh kode berikut menunjukkan bagaimana Anda dapat menggunakan `MXNetProcessor` untuk menjalankan pekerjaan Processing Anda menggunakan image Docker yang disediakan dan dikelola oleh SageMaker. Perhatikan bahwa ketika Anda menjalankan pekerjaan, Anda dapat menentukan direktori yang berisi skrip dan dependensi Anda dalam `source_dir` argumen, dan Anda dapat memiliki `requirements.txt` file yang terletak di dalam `source_dir` direktori Anda

yang menentukan dependensi untuk skrip pemrosesan Anda. SageMaker Pemrosesan menginstal dependensi di `requirements.txt` dalam wadah untuk Anda.

```
from sagemaker.mxnet import MXNetProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker import get_execution_role

#Initialize the MXNetProcessor
mxp = MXNetProcessor(
    framework_version='1.8.0',
    py_version='py37',
    role=get_execution_role(),
    instance_count=1,
    instance_type='ml.c5.xlarge',
    base_job_name='frameworkprocessor-mxnet'
)

#Run the processing job
mxp.run(
    code='processing-script.py',
    source_dir='scripts',
    inputs=[
        ProcessingInput(
            input_name='data',
            source=f's3://{BUCKET}/{S3_INPUT_PATH}',
            destination='/opt/ml/processing/input/data/'
        )
    ],
    outputs=[
        ProcessingOutput(
            output_name='processed_data',
            source='/opt/ml/processing/output/',
            destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'
        )
    ]
)
```

Jika Anda memiliki `requirements.txt` file, itu harus berupa daftar pustaka yang ingin Anda instal di wadah. Jalur untuk `source_dir` dapat berupa jalur URI relatif, absolut, atau Amazon S3. Namun, jika Anda menggunakan URI Amazon S3, maka itu harus mengarah ke file tar.gz. Anda dapat memiliki beberapa skrip di direktori yang Anda tentukan`source_dir`. Untuk mempelajari lebih lanjut tentang `MXNetProcessor` kelas, lihat MXNet Estimator di Amazon SageMaker Python SDK.

PyTorch Prosesor Kerangka

PyTorch adalah kerangka pembelajaran mesin sumber terbuka. SDK SageMaker Python Amazon `PyTorchProcessor` di Amazon memberi Anda kemampuan untuk menjalankan pekerjaan pemrosesan dengan skrip. PyTorch Saat Anda menggunakan`PyTorchProcessor`, Anda dapat memanfaatkan wadah Docker buatan Amazon dengan PyTorch lingkungan terkelola sehingga Anda tidak perlu membawa wadah sendiri.

Contoh kode berikut menunjukkan bagaimana Anda dapat menggunakan `PyTorchProcessor` untuk menjalankan pekerjaan Processing Anda menggunakan image Docker yang disediakan dan dikelola oleh SageMaker. Perhatikan bahwa ketika Anda menjalankan pekerjaan, Anda dapat menentukan direktori yang berisi skrip dan dependensi Anda dalam `source_dir` argumen, dan Anda dapat memiliki `requirements.txt` file yang terletak di dalam `source_dir` direktori Anda yang menentukan dependensi untuk skrip pemrosesan Anda. SageMaker Pemrosesan menginstal dependensi di `requirements.txt` dalam wadah untuk Anda.

Untuk PyTorch versi yang didukung oleh SageMaker, lihat gambar Deep Learning Container yang tersedia.

```
from sagemaker.pytorch.processing import PyTorchProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker import get_execution_role

#Initialize the PyTorchProcessor
pytorch_processor = PyTorchProcessor(
    framework_version='1.8',
    role=get_execution_role(),
    instance_type='ml.m5.xlarge',
    instance_count=1,
    base_job_name='frameworkprocessor-PT'
)

#Run the processing job
pytorch_processor.run(
    code='processing-script.py',
    source_dir='scripts',
    inputs=[
        ProcessingInput(
            input_name='data',
            source=f's3://{BUCKET}/{S3_INPUT_PATH}',
            destination='/opt/ml/processing/input'
```

```
        )
    ],
    outputs=[
        ProcessingOutput(output_name='data_structured', source='/opt/ml/processing/tmp/
data_structured', destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'),
        ProcessingOutput(output_name='train', source='/opt/ml/processing/output/train',
 destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'),
        ProcessingOutput(output_name='validation', source='/opt/ml/processing/output/
val', destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'),
        ProcessingOutput(output_name='test', source='/opt/ml/processing/output/test',
 destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'),
        ProcessingOutput(output_name='logs', source='/opt/ml/processing/logs',
 destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}')
    ]
)
```

Jika Anda memiliki `requirements.txt` file, itu harus berupa daftar pustaka yang ingin Anda instal di wadah. Jalur untuk `source_dir` dapat berupa jalur URI relatif, absolut, atau Amazon S3. Namun, jika Anda menggunakan URI Amazon S3, maka itu harus mengarah ke file tar.gz. Anda dapat memiliki beberapa skrip di direktori yang Anda tentukan`source_dir`. Untuk mempelajari lebih lanjut tentang `PyTorchProcessor` kelas, lihat PyTorch Estimator di Amazon SageMaker Python SDK.

TensorFlow Prosesor Kerangka

TensorFlow adalah pembelajaran mesin sumber terbuka dan perpustakaan kecerdasan buatan. SDK SageMaker Python Amazon `TensorFlowProcessor` di Amazon memberi Anda kemampuan untuk menjalankan pekerjaan pemrosesan dengan skrip. TensorFlow Saat Anda menggunakan`TensorFlowProcessor`, Anda dapat memanfaatkan wadah Docker buatan Amazon dengan TensorFlow lingkungan terkelola sehingga Anda tidak perlu membawa wadah sendiri.

Contoh kode berikut menunjukkan bagaimana Anda dapat menggunakan `TensorFlowProcessor` untuk menjalankan pekerjaan Processing Anda menggunakan image Docker yang disediakan dan dikelola oleh SageMaker. Perhatikan bahwa ketika Anda menjalankan pekerjaan, Anda dapat menentukan direktori yang berisi skrip dan dependensi Anda dalam `source_dir` argumen, dan Anda dapat memiliki `requirements.txt` file yang terletak di dalam `source_dir` direktori Anda yang menentukan dependensi untuk skrip pemrosesan Anda. SageMaker Pemrosesan menginstal dependensi di `requirements.txt` dalam wadah untuk Anda.

```
from sagemaker.tensorflow import TensorFlowProcessor
```

```
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker import get_execution_role

#Initialize the TensorFlowProcessor
tp = TensorFlowProcessor(
    framework_version='2.3',
    role=get_execution_role(),
    instance_type='ml.m5.xlarge',
    instance_count=1,
    base_job_name='frameworkprocessor-TF',
    py_version='py37'
)

#Run the processing job
tp.run(
    code='processing-script.py',
    source_dir='scripts',
    inputs=[
        ProcessingInput(
            input_name='data',
            source=f's3://{BUCKET}/{S3_INPUT_PATH}',
            destination='/opt/ml/processing/input/data'
        ),
        ProcessingInput(
            input_name='model',
            source=f's3://{BUCKET}/{S3_PATH_TO_MODEL}',
            destination='/opt/ml/processing/input/model'
        )
    ],
    outputs=[
        ProcessingOutput(
            output_name='predictions',
            source='/opt/ml/processing/output',
            destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'
        )
    ]
)
```

Jika Anda memiliki `requirements.txt` file, itu harus berupa daftar pustaka yang ingin Anda instal di wadah. Jalur untuk `source_dir` dapat berupa jalur URI relatif, absolut, atau Amazon S3. Namun, jika Anda menggunakan URI Amazon S3, maka itu harus mengarah ke file tar.gz. Anda dapat memiliki beberapa skrip di direktori yang Anda tentukan`source_dir`. Untuk mempelajari lebih

lanjut tentang `TensorFlowProcessor` kelas, lihat TensorFlow Estimator di Amazon SageMaker Python SDK.

Prosesor Kerangka XGBoost

XGBoost adalah kerangka pembelajaran mesin open-source. SDK SageMaker Python Amazon `XGBoostProcessor` di Amazon memberi Anda kemampuan untuk menjalankan pekerjaan pemrosesan dengan skrip XGBoost. Saat Anda menggunakan XGBoostProcessor, Anda dapat memanfaatkan wadah Docker buatan Amazon dengan lingkungan XGBoost terkelola sehingga Anda tidak perlu membawa wadah Anda sendiri.

Contoh kode berikut menunjukkan bagaimana Anda dapat menggunakan `XGBoostProcessor` untuk menjalankan pekerjaan Processing Anda menggunakan image Docker yang disediakan dan dikelola oleh SageMaker. Perhatikan bahwa ketika Anda menjalankan pekerjaan, Anda dapat menentukan direktori yang berisi skrip dan dependensi Anda dalam `source_dir` argumen, dan Anda dapat memiliki `requirements.txt` file yang terletak di dalam `source_dir` direktori Anda yang menentukan dependensi untuk skrip pemrosesan Anda. SageMaker Pemrosesan menginstal dependensi di `requirements.txt` dalam wadah untuk Anda.

```
from sagemaker.xgboost import XGBoostProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker import get_execution_role

#Initialize the XGBoostProcessor
xgb = XGBoostProcessor(
    framework_version='1.2-2',
    role=get_execution_role(),
    instance_type='ml.m5.xlarge',
    instance_count=1,
    base_job_name='frameworkprocessor-XGB',
)

#Run the processing job
xgb.run(
    code='processing-script.py',
    source_dir='scripts',
    inputs=[
        ProcessingInput(
            input_name='data',
            source=f's3://{BUCKET}/{S3_INPUT_PATH}',
            destination='/opt/ml/processing/input/data'
```

```
        )
    ],
    outputs=[
        ProcessingOutput(
            output_name='processed_data',
            source='/opt/ml/processing/output/',
            destination=f's3://{BUCKET}/{S3_OUTPUT_PATH}'
        )
    ]
)
```

Jika Anda memiliki `requirements.txt` file, itu harus berupa daftar pustaka yang ingin Anda instal di wadah. Jalur untuk `source_dir` dapat berupa jalur URI relatif, absolut, atau Amazon S3. Namun, jika Anda menggunakan URI Amazon S3, maka itu harus mengarah ke file tar.gz. Anda dapat memiliki beberapa skrip di direktori yang Anda tentukan`source_dir`. Untuk mempelajari lebih lanjut tentang `XGBoostProcessor` kelas, lihat XGBoost Estimator di Amazon Python SageMaker SDK.

Gunakan Kode Pemrosesan Anda Sendiri

Anda dapat menginstal pustaka untuk menjalankan skrip di wadah pemrosesan Anda sendiri atau, dalam skenario yang lebih canggih, Anda dapat membuat wadah pemrosesan sendiri yang memenuhi kontrak untuk dijalankan di Amazon. SageMaker Untuk informasi lebih lanjut tentang kontainer diSageMaker, lihatGunakan kontainer Docker untuk membuat model. Untuk spesifikasi formal yang mendefinisikan kontrak untuk container Amazon SageMaker Processing, lihatBangun Kontainer Pemrosesan Anda Sendiri (Skenario Lanjutan).

Topik

- Jalankan Skrip dengan Kontainer Pemrosesan Anda Sendiri
- Bangun Kontainer Pemrosesan Anda Sendiri (Skenario Lanjutan)

Jalankan Skrip dengan Kontainer Pemrosesan Anda Sendiri

Anda dapat menggunakan skrip scikit-learn untuk memproses data dan mengevaluasi model Anda. Untuk melihat cara menjalankan skrip scikit-learn untuk melakukan tugas-tugas ini, lihat notebook sampel Pemrosesan scikit-learn. Notebook ini menggunakan `ScriptProcessor` class dari Amazon SageMaker Python SDK for Processing.

Contoh berikut menunjukkan alur kerja umum untuk menggunakan `ScriptProcessor` kelas dengan wadah pengolahan Anda sendiri. Alur kerja menunjukkan cara membuat gambar Anda sendiri, membangun wadah Anda, dan menggunakan `ScriptProcessor` kelas untuk menjalankan skrip preprocessing Python dengan wadah. Pekerjaan pemrosesan memproses data input Anda dan menyimpan data yang diproses di Amazon Simple Storage Service (Amazon S3).

Sebelum menggunakan contoh berikut, Anda harus memiliki data input Anda sendiri dan skrip Python yang disiapkan untuk memproses data Anda. Untuk contoh terpandu dari proses ini secara end-to-end, lihat kembali ke buku catatan sampel Pemrosesan scikit-learn.

1. Buat direktori Docker dan tambahkan Dockerfile yang digunakan untuk membuat wadah pemrosesan. Instal panda dan scikit-pelajari ke dalamnya. (Anda juga dapat menginstal dependensi Anda sendiri dengan RUN perintah serupa.)

```
mkdir docker

%%writefile docker/Dockerfile

FROM python:3.7-slim-buster

RUN pip3 install pandas==0.25.3 scikit-learn==0.21.3
ENV PYTHONUNBUFFERED=TRUE

ENTRYPOINT ["python3"]
```

2. Buat container menggunakan perintah docker, buat repositori Amazon Elastic Container Registry (Amazon ECR), dan dorong gambar ke Amazon ECR.

```
import boto3

account_id = boto3.client('sts').get_caller_identity().get('Account')
region = boto3.Session().region_name
ecr_repository = 'sagemaker-processing-container'
tag = ':latest'
processing_repository_uri = '{}.dkr.ecr.{}.amazonaws.com/{}'.format(account_id,
 region, ecr_repository + tag)

# Create ECR repository and push docker image
!docker build -t $ecr_repository docker
!aws ecr get-login-password --region {region} | docker login --username AWS --
password-stdin {account_id}.dkr.ecr.{region}.amazonaws.com
!aws ecr create-repository --repository-name $ecr_repository
```

```
!docker tag {ecr_repository + tag} $processing_repository_uri
!docker push $processing_repository_uri
```

3. Mengatur `ScriptProcessor` dari SageMaker Python SDK untuk menjalankan script. Ganti *`image_uri dengan URI`* untuk gambar yang Anda buat, dan ganti *`role_arn dengan ARN`* untuk AWS Identity and Access Management peran yang memiliki akses ke bucket Amazon S3 target Anda.

```
from sagemaker.processing import ScriptProcessor, ProcessingInput, ProcessingOutput

script_processor = ScriptProcessor(command=['python3'],
                image_uri='image_uri',
                role='role_arn',
                instance_count=1,
                instance_type='ml.m5.xlarge')
```

4. Jalankan penulisan. Ganti *`preprocessing.py`* dengan nama skrip pemrosesan Python Anda sendiri, dan ganti *`s3://path/to/my/input-data.csv`* dengan jalur Amazon S3 ke data input Anda.

```
script_processor.run(code='preprocessing.py',
                     inputs=[ProcessingInput(
                        source='s3://path/to/my/input-data.csv',
                        destination='/opt/ml/processing/input')],
                     outputs=[ProcessingOutput(source='/opt/ml/processing/output/
train'),
                              ProcessingOutput(source='/opt/ml/processing/output/
validation'),
                              ProcessingOutput(source='/opt/ml/processing/output/
test')])
```

Anda dapat menggunakan prosedur yang sama dengan perpustakaan atau dependensi sistem lainnya. Anda juga dapat menggunakan gambar Docker yang ada. Ini termasuk gambar yang kamu jalankan pada platform lain seperti Kubernetes.

Bangun Kontainer Pemrosesan Anda Sendiri (Skenario Lanjutan)

Anda dapat menyediakan Amazon SageMaker Processing dengan image Docker yang memiliki kode dan dependensi Anda sendiri untuk menjalankan beban kerja pemrosesan data, rekayasa fitur, dan evaluasi model.

Contoh berikut dari Dockerfile membangun wadah dengan pustaka Python scikit-learn dan panda, yang dapat Anda jalankan sebagai pekerjaan pemrosesan.

```
FROM python:3.7-slim-buster

# Install scikit-learn and pandas
RUN pip3 install pandas==0.25.3 scikit-learn==0.21.3

# Add a Python script and configure Docker to run it
ADD processing_script.py /
ENTRYPOINT ["python3", "/processing_script.py"]
```

Untuk contoh skrip pemrosesan, lihat Memulai dengan SageMaker Processing.

Buat dan dorong image Docker ini ke repositori Amazon Elastic Container Registry (Amazon ECR) dan pastikan peran SageMaker IAM Anda dapat menarik gambar dari Amazon ECR. Kemudian Anda dapat menjalankan gambar ini di Amazon SageMaker Processing.

Cara Amazon SageMaker Processing Menjalankan Gambar Kontainer Pemrosesan Anda

Amazon SageMaker Processing menjalankan image container pemrosesan Anda dengan cara yang sama seperti perintah berikut, di `AppSpecification.ImageUri` mana URI image Amazon ECR yang Anda tentukan dalam `CreateProcessingJob` operasi.

```
docker run [AppSpecification.ImageUri]
```

Perintah ini menjalankan `ENTRYPOINT` perintah yang dikonfigurasi dalam image Docker Anda.

Anda juga dapat mengganti perintah entrypoint dalam gambar atau memberikan argumen baris perintah ke perintah entrypoint Anda menggunakan parameter dan dalam permintaan Anda. `AppSpecification.ContainerEntrypoint` `AppSpecification.ContainerArgument` `CreateProcessingJob` Menentukan parameter ini mengonfigurasi Amazon SageMaker Processing untuk menjalankan kontainer yang mirip dengan cara perintah berikut.

```
docker run --entry-point [AppSpecification.ContainerEntrypoint]
[AppSpecification.ImageUri] [AppSpecification.ContainerArguments]
```

Misalnya, jika Anda menentukan yang `ContainerEntrypoint` akan ada `[python3, -v, /processing_script.py]` dalam `CreateProcessingJob` permintaan Anda, dan

`ContainerArguments` menjadi`[data-format, csv]`, Amazon SageMaker Processing menjalankan kontainer Anda dengan perintah berikut.

```
python3 -v /processing_script.py data-format csv
```

Saat membangun wadah pemrosesan Anda, pertimbangkan detail berikut:

- Amazon SageMaker Processing memutuskan apakah pekerjaan selesai atau gagal tergantung pada kode keluar dari perintah yang dijalankan. Pekerjaan pemrosesan selesai jika semua kontainer pemrosesan berhasil keluar dengan kode keluar 0, dan gagal jika salah satu kontainer keluar dengan kode keluar bukan nol.
- Amazon SageMaker Processing memungkinkan Anda mengganti entrypoint kontainer pemrosesan dan menetapkan argumen baris perintah seperti yang Anda bisa dengan API Docker. Gambar buruh pelabuhan juga dapat mengkonfigurasi entrypoint dan argumen baris perintah menggunakan instruksi dan CMD. `ENTRYPOINT` Cara `CreateProcessingJob` `ContainerEntrypoint` dan `ContainerArgument` parameter mengkonfigurasi entrypoint dan argumen gambar Docker mencerminkan bagaimana Docker menimpa entrypoint dan argumen melalui API Docker:
  - Jika tidak `ContainerEntrypoint` juga `ContainerArguments` disediakan, Processing menggunakan default `ENTRYPOINT` atau CMD dalam gambar.
  - Jika `ContainerEntrypoint` disediakan, tetapi tidak`ContainerArguments`, Processing menjalankan gambar dengan entrypoint yang diberikan, dan mengabaikan `ENTRYPOINT` dan CMD dalam gambar.
  - Jika `ContainerArguments` disediakan, tetapi tidak`ContainerEntrypoint`, Processing menjalankan gambar dengan default `ENTRYPOINT` dalam gambar dan dengan argumen yang disediakan.
  - Jika keduanya `ContainerEntrypoint` dan `ContainerArguments` disediakan, Processing menjalankan gambar dengan entrypoint dan argumen yang diberikan, dan mengabaikan `ENTRYPOINT` dan CMD dalam gambar.
- Anda harus menggunakan formulir exec `ENTRYPOINT` instruksi di Dockerfile Anda (`ENTRYPOINT["executable", "param1", "param2"]`)bukan bentuk shell (). `ENTRYPOINT command param1 param2` Hal ini memungkinkan kontainer pemrosesan Anda menerima `SIGINT` dan `SIGKILL` memberi sinyal, yang digunakan Pemrosesan untuk menghentikan pemrosesan pekerjaan dengan `StopProcessingJob` API.
- `/opt/ml`dan semua subdirektorinya dicadangkan oleh. SageMaker Saat membuat image Processing Docker Anda, jangan letakkan data apa pun yang diperlukan oleh container pemrosesan Anda di direktori ini.

- Jika Anda berencana untuk menggunakan perangkat GPU, pastikan kontainer Anda kompatibel dengan nvidia-docker. Sertakan hanya toolkit CUDA dalam wadah. Jangan bundel driver NVIDIA dengan gambar. Untuk informasi lebih lanjut tentang nvidia-docker, lihat NVIDIA/NVIDIA-Docker.

Cara Amazon SageMaker Processing Mengonfigurasi Input dan Output Untuk Container Pemrosesan Anda

Saat Anda membuat pekerjaan pemrosesan menggunakan `CreateProcessingJob` operasi, Anda dapat menentukan beberapa `ProcessingInput` dan`ProcessingOutput`. nilai.

Anda menggunakan `ProcessingInput` parameter untuk menentukan URI Amazon Simple Storage Service (Amazon S3) untuk mengunduh data dari, dan jalur dalam wadah pemrosesan Anda untuk mengunduh data. `ProcessingOutput`Parameter mengonfigurasi jalur dalam wadah pemrosesan untuk mengunggah data, dan tempat di Amazon S3 untuk mengunggah data tersebut. Untuk keduanya `ProcessingInput` dan`ProcessingOutput`, jalur dalam wadah pemrosesan harus dimulai dengan`/opt/ml/processing/` .

Misalnya, Anda dapat membuat pekerjaan pemrosesan dengan satu `ProcessingInput` parameter yang mengunduh data dari `s3://your-data-bucket/path/to/input/csv/data` `/opt/ml/processing/csv` dalam wadah pemrosesan Anda, dan `ProcessingOutput` parameter yang mengunggah data dari `/opt/ml/processing/processed_csv` kes`3://your-data-bucket/path/to/output/csv/data`. Pekerjaan pemrosesan Anda akan membaca data input, dan menulis data keluaran ke`/opt/ml/processing/processed_csv`. Kemudian mengunggah data yang ditulis ke jalur ini ke lokasi keluaran Amazon S3 yang ditentukan.

> **Important**
>
> Tautan simbolik (symlink) tidak dapat digunakan untuk mengunggah data keluaran ke Amazon S3. Symlink tidak diikuti saat mengunggah data keluaran.

Cara Amazon SageMaker Processing Menyediakan Log dan Metrik untuk Kontainer Pemrosesan Anda

Saat wadah pemrosesan Anda menulis ke `stdout` ataustderr, Amazon SageMaker Processing menyimpan output dari setiap wadah pemrosesan dan menempatkannya di CloudWatch log Amazon. Untuk informasi tentang logging, lihatLog SageMaker Acara Amazon dengan Amazon CloudWatch.

Amazon SageMaker Processing juga menyediakan CloudWatch metrik untuk setiap instans yang menjalankan wadah pemrosesan Anda. Untuk informasi tentang metrik, lihatPantau Amazon SageMaker dengan Amazon CloudWatch.

Cara Amazon SageMaker Processing Mengkonfigurasi Container Pemrosesan Anda

Amazon SageMaker Processing menyediakan informasi konfigurasi ke wadah pemrosesan Anda melalui variabel lingkungan dan dua file JSON— `/opt/ml/config/processingjobconfig.json` dan `/opt/ml/config/resourceconfig.json` — di lokasi yang telah ditentukan dalam wadah.

Ketika pekerjaan pemrosesan dimulai, ia menggunakan variabel lingkungan yang Anda tentukan dengan `Environment` peta dalam `CreateProcessingJob` permintaan. `/opt/ml/config/processingjobconfig.json`File berisi informasi tentang nama host wadah pemrosesan Anda, dan juga ditentukan dalam `CreateProcessingJob` permintaan.

Contoh berikut menunjukkan format `/opt/ml/config/processingjobconfig.json` file.

```
{
    "ProcessingJobArn": "<processing_job_arn>",
    "ProcessingJobName": "<processing_job_name>",
    "AppSpecification": {
        "ImageUri": "<image_uri>",
        "ContainerEntrypoint": null,
        "ContainerArguments": null
    },
    "Environment": {
        "KEY": "VALUE"
    },
    "ProcessingInputs": [
        {
            "InputName": "input-1",
            "S3Input": {
                "LocalPath": "/opt/ml/processing/input/dataset",
                "S3Uri": "<s3_uri>",
                "S3DataDistributionType": "FullyReplicated",
                "S3DataType": "S3Prefix",
                "S3InputMode": "File",
                "S3CompressionType": "None",
                "S3DownloadMode": "StartOfJob"
            }
        }
```

```
    ],
    "ProcessingOutputConfig": {
        "Outputs": [
            {
                "OutputName": "output-1",
                "S3Output": {
                    "LocalPath": "/opt/ml/processing/output/dataset",
                    "S3Uri": "<s3_uri>",
                    "S3UploadMode": "EndOfJob"
                }
            }
        ],
        "KmsKeyId": null
    },
    "ProcessingResources": {
        "ClusterConfig": {
            "InstanceCount": 1,
            "InstanceType": "ml.m5.xlarge",
            "VolumeSizeInGB": 30,
            "VolumeKmsKeyId": null
        }
    },
    "RoleArn": "<IAM role>",
    "StoppingCondition": {
        "MaxRuntimeInSeconds": 86400
    }
}
```

`/opt/ml/config/resourceconfig.json`File berisi informasi tentang nama host wadah pemrosesan Anda. Gunakan nama host berikut saat membuat atau menjalankan kode pemrosesan terdistribusi.

```
{
  "current_host": "algo-1",
  "hosts": ["algo-1","algo-2","algo-3"]
}
```

Jangan gunakan informasi tentang nama host yang terdapat di dalamnya `/etc/hostname` atau `/etc/hosts` karena mungkin tidak akurat.

Informasi nama host mungkin tidak segera tersedia untuk wadah pemrosesan. Sebaiknya tambahkan kebijakan coba lagi pada operasi resolusi nama host saat node tersedia di klaster.

Simpan dan Akses Informasi Metadata Tentang Pekerjaan Pemrosesan Anda

Untuk menyimpan metadata dari wadah pemrosesan setelah keluar, kontainer dapat menulis teks yang dikodekan UTF-8 ke file. `/opt/ml/output/message` Setelah pekerjaan pemrosesan memasuki status terminal apa pun (" `Completed` "," `Stopped` ", atau" `Failed` "), bidang`ExitMessage`" "di `DescribeProcessingJob`berisi 1 KB pertama dari file ini. Akses bagian awal file dengan panggilan ke `DescribeProcessingJob`, yang mengembalikannya melalui `ExitMessage` parameter. Untuk pekerjaan pemrosesan yang gagal, Anda dapat menggunakan bidang ini untuk mengkomunikasikan informasi tentang mengapa wadah pemrosesan gagal.

> **Important**
>
> Jangan menulis data sensitif ke `/opt/ml/output/message` file.

Jika data dalam file ini tidak UTF-8 dikodekan, pekerjaan gagal dan mengembalikan file. `ClientError` Jika beberapa kontainer keluar dengan `ExitMessage`, konten `ExitMessage` dari setiap wadah pemrosesan digabungkan, maka dipotong menjadi 1 KB.

Jalankan Container Pemrosesan Anda Menggunakan SageMaker Python SDK

Anda dapat menggunakan SageMaker Python SDK untuk menjalankan gambar pemrosesan Anda sendiri dengan menggunakan kelas. `Processor` Contoh berikut menunjukkan cara menjalankan wadah pemrosesan Anda sendiri dengan satu input dari Amazon Simple Storage Service (Amazon S3) dan satu output ke Amazon S3.

```
from sagemaker.processing import Processor, ProcessingInput, ProcessingOutput

processor = Processor(image_uri='<your_ecr_image_uri>',
                      role=role,
                      instance_count=1,
                      instance_type="ml.m5.xlarge")

processor.run(inputs=[ProcessingInput(
                         source='<s3_uri or local path>',
                         destination='/opt/ml/processing/input_data')],
                      outputs=[ProcessingOutput(
                         source='/opt/ml/processing/processed_data',
                         destination='<s3_uri>')],
                      )
```

Alih-alih membangun kode pemrosesan Anda ke dalam gambar pemrosesan Anda, Anda dapat memberikan gambar Anda dan perintah yang ingin Anda jalankan, bersama dengan kode yang ingin Anda jalankan di dalam wadah itu. `ScriptProcessor` Sebagai contoh, lihat Jalankan Skrip dengan Kontainer Pemrosesan Anda Sendiri.

Anda juga dapat menggunakan gambar scikit-learn yang disediakan Amazon SageMaker Processing `SKLearnProcessor` untuk menjalankan skrip scikit-learn. Sebagai contoh, lihat Pemrosesan Data dengan scikit-learn.

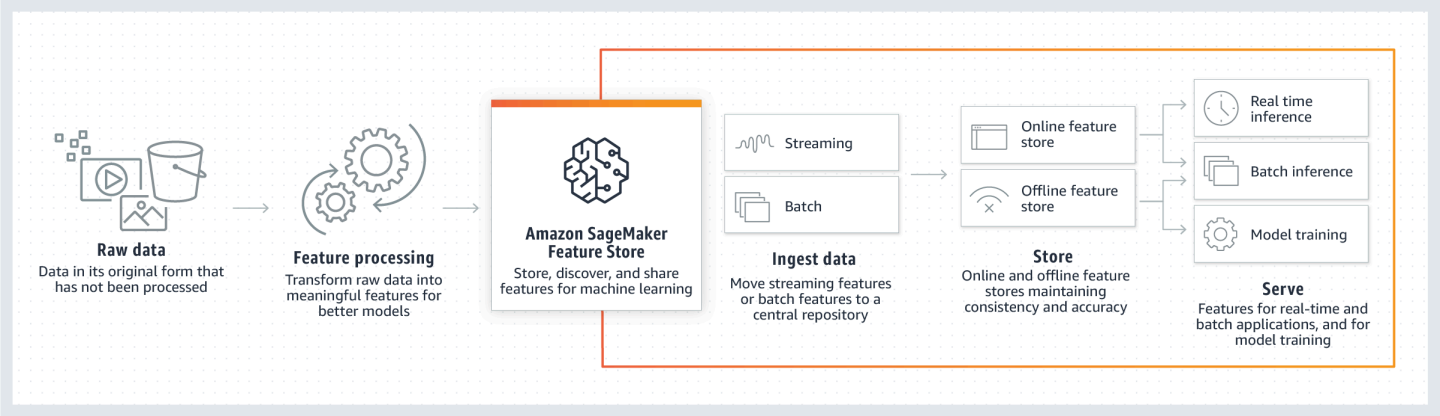
Buat, simpan, dan bagikan fitur dengan Amazon SageMaker Feature Store

Proses pengembangan machine learning (ML) sering dimulai dengan mengekstraksi sinyal data yang juga dikenal sebagai fitur dari data untuk melatih model ML. Amazon SageMaker Feature Store memudahkan ilmuwan data, insinyur pembelajaran mesin, dan dokter umum untuk membuat, berbagi, dan mengelola fitur untuk pengembangan ML. Feature Store mempercepat proses ini dengan mengurangi pemrosesan data berulang dan pekerjaan kurasi yang diperlukan untuk mengonversi data mentah menjadi fitur untuk melatih algoritma ML.

Selanjutnya, logika pemrosesan untuk data Anda ditulis hanya sekali, dan fitur yang dihasilkan digunakan untuk pelatihan dan inferensi, mengurangi kemiringan penyajian pelatihan. Feature Store adalah toko terpusat untuk fitur dan metadata terkait sehingga fitur dapat dengan mudah ditemukan dan digunakan kembali. Anda dapat membuat toko online atau offline. Toko online digunakan untuk kasus penggunaan inferensi real-time latensi rendah, dan toko offline digunakan untuk pelatihan dan inferensi batch.

Diagram berikut menunjukkan bagaimana Anda dapat menggunakan Feature Store sebagai bagian dari pipeline pembelajaran mesin Anda. Pertama, Anda membaca data mentah Anda dan memprosesnya. Anda dapat menelan data melalui streaming ke toko online dan offline, atau dalam batch langsung ke toko offline. Anda pertama kali membuat `FeatureGroup` dan mengkonfigurasinya ke toko online atau offline, atau keduanya. Kemudian, Anda dapat menelan data ke dalam Anda `FeatureGroup` dan menyimpannya di toko Anda. A `FeatureGroup` adalah sekelompok fitur yang didefinisikan melalui skema di Feature Store untuk menggambarkan catatan.

Toko online terutama dirancang untuk mendukung prediksi real-time yang membutuhkan pembacaan latensi milidetik rendah dan penulisan throughput tinggi. Toko offline terutama ditujukan untuk prediksi batch dan pelatihan model. Toko offline adalah toko tambahan saja dan dapat digunakan untuk menyimpan dan mengakses data fitur historis. Toko offline dapat membantu Anda menyimpan dan melayani fitur untuk eksplorasi dan pelatihan model. Toko online hanya menyimpan data fitur terbaru. Grup Fitur dapat berubah dan dapat mengembangkan skema mereka setelah pembuatan.



Cara kerja Feature Store

Di Feature Store, fitur disimpan dalam koleksi yang disebut grup fitur. Anda dapat memvisualisasikan grup fitur sebagai tabel di mana setiap kolom adalah fitur, dengan pengenal unik untuk setiap baris. Pada prinsipnya, grup fitur terdiri dari fitur dan nilai khusus untuk setiap fitur. A `Record` adalah kumpulan nilai untuk fitur yang sesuai dengan yang unik`RecordIdentifier`. Secara keseluruhan, a `FeatureGroup` adalah sekelompok fitur yang didefinisikan dalam Anda `FeatureStore` untuk menggambarkan a`Record`.

Anda dapat menggunakan Toko Fitur dalam mode berikut:

- Online — Dalam mode online, fitur dibaca dengan pembacaan latensi rendah (milidetik) dan digunakan untuk prediksi throughput tinggi. Mode ini membutuhkan grup fitur untuk disimpan di toko online.
- Offline — Dalam mode offline, aliran data yang besar diumpankan ke toko offline, yang dapat digunakan untuk pelatihan dan inferensi batch. Mode ini membutuhkan grup fitur untuk disimpan di toko offline. Toko offline menggunakan bucket S3 Anda untuk penyimpanan dan juga dapat mengambil data menggunakan kueri Athena.
- Online dan Offline — Ini termasuk mode online dan offline.

Anda dapat memasukkan data ke dalam grup fitur di Feature Store dengan dua cara: streaming atau dalam batch. Saat Anda menyerap data melalui streaming, kumpulan catatan akan didorong ke Feature Store dengan memanggil panggilan `PutRecord` API sinkron. API ini memungkinkan Anda mempertahankan nilai fitur terbaru di Feature Store dan mendorong nilai fitur baru segera setelah pembaruan terdeteksi.

Atau, Feature Store dapat memproses dan menelan data dalam batch. Anda dapat membuat fitur menggunakan Amazon SageMaker Data Wrangler, membuat grup fitur di Feature Store, dan menyerap fitur dalam batch menggunakan pekerjaan SageMaker Processing dengan notebook yang diekspor dari Data Wrangler. Mode ini memungkinkan konsumsi batch ke toko offline. Ini juga mendukung konsumsi ke toko online jika grup fitur dikonfigurasi untuk penggunaan online dan offline.

Buat grup fitur

Untuk memasukkan fitur ke dalam Feature Store, Anda harus terlebih dahulu menentukan grup fitur dan definisi fitur (nama fitur dan tipe data) untuk semua fitur yang termasuk dalam grup fitur. Setelah dibuat, grup fitur dapat berubah dan dapat mengembangkan skema mereka. Nama grup fitur unik dalam Wilayah AWS danAkun AWS. Saat membuat grup fitur, Anda juga dapat membuat metadata untuk grup fitur, seperti deskripsi singkat, konfigurasi penyimpanan, fitur untuk mengidentifikasi setiap catatan, dan waktu acara, serta tag untuk menyimpan informasi seperti penulis, sumber data, versi, dan lainnya.

> **Important**
>
> `FeatureGroup`nama atau metadata terkait seperti deskripsi atau tag tidak boleh mengandung informasi identitas pribadi (PII) atau informasi rahasia apa pun.

Temukan, temukan, dan bagikan fitur

Setelah Anda membuat grup fitur di Feature Store, pengguna resmi lainnya dari feature store dapat berbagi dan menemukannya. Pengguna dapat menelusuri daftar semua grup fitur di Toko Fitur atau menemukan grup fitur yang ada dengan mencari berdasarkan nama grup fitur, deskripsi, nama pengenal catatan, tanggal pembuatan, dan tag.

Inferensi waktu nyata untuk fitur yang disimpan di toko online

Dengan Feature Store, Anda dapat memperkaya fitur yang disimpan di toko online secara real time dengan data dari sumber streaming (data aliran bersih dari aplikasi lain) dan menyajikan fitur dengan latensi milidetik rendah untuk inferensi waktu nyata.

Anda juga dapat melakukan gabungan di berbagai `FeatureGroups` untuk inferensi real-time dengan menanyakan dua yang berbeda `FeatureGroups` dalam aplikasi klien.

Toko offline untuk pelatihan model dan inferensi batch

Feature Store menyediakan penyimpanan offline untuk nilai fitur di bucket S3 Anda. Data Anda disimpan dalam bucket S3 menggunakan skema awalan berdasarkan waktu acara. Toko offline adalah toko khusus tambahan, memungkinkan Toko Fitur untuk menyimpan catatan historis semua nilai fitur. Data disimpan di toko offline dalam format Parket untuk penyimpanan dan akses kueri yang dioptimalkan.

Anda dapat melakukan kueri, menjelajahi, dan memvisualisasikan fitur menggunakan Data Wrangler dari konsol. Feature Store mendukung penggabungan data untuk menghasilkan, melatih, memvalidasi, dan menguji kumpulan data, dan memungkinkan Anda mengekstrak data pada titik waktu yang berbeda.

Konsumsi data fitur

Jaringan pipa generasi fitur dapat dibuat untuk memproses batch besar (1 juta baris data atau lebih) atau batch kecil, dan untuk menulis data fitur ke toko offline atau online. Sumber streaming seperti Amazon Managed Streaming for Apache Kafka atau Amazon Kinesis juga dapat digunakan sebagai sumber data dari mana fitur diekstraksi dan langsung diumpankan ke toko online untuk pelatihan, inferensi, atau pembuatan fitur.

Anda dapat mendorong catatan ke Feature Store dengan memanggil panggilan `PutRecord` API sinkron. Karena ini adalah panggilan API sinkron, ini memungkinkan sejumlah kecil pembaruan didorong dalam satu panggilan API. Ini memungkinkan Anda untuk mempertahankan kesegaran nilai fitur yang tinggi dan mempublikasikan nilai segera setelah pembaruan terdeteksi. Ini juga disebut fitur streaming.

Ketika data fitur dicerna dan diperbarui, Feature Store menyimpan data historis untuk semua fitur di toko offline. Untuk batch ingest, Anda dapat menarik nilai fitur dari bucket S3 Anda atau menggunakan Athena untuk melakukan kueri. Anda juga dapat menggunakan Data Wrangler untuk memproses dan merekayasa fitur baru yang kemudian dapat diekspor ke bucket S3 yang dipilih untuk diakses oleh Feature Store. Untuk batch ingestion, Anda dapat mengonfigurasi tugas pemrosesan untuk mengumpulkan data secara batch ke Feature Store, atau Anda dapat menarik nilai fitur dari bucket S3 menggunakan Athena.

Untuk menghapus a `Record` dari toko online Anda, gunakan panggilan `DeleteRecord`API. Ini juga akan menambahkan catatan yang dihapus ke toko offline.

Ketahanan di Feature Store

Toko Fitur didistribusikan di beberapa Availability Zone (AZ). AZ adalah lokasi yang terisolasi dalam fileWilayah AWS. Jika beberapa AZ gagal, Feature Store dapat menggunakan AZ lain. Untuk informasi selengkapnya tentang AZ, lihatKetahanan di Amazon SageMaker.

Memulai dengan Amazon SageMaker Feature Store

Topik berikut memberikan informasi tentang penggunaan Amazon SageMaker Feature Store. Pertama pelajari konsep Feature Store, lalu cara mengelola izin untuk menggunakan Feature Store, cara membuat dan menggunakan grup fitur menggunakan Studio Classic, Jupyter atau JupyterLab notebook, cara menggunakan Feature Store menggunakan User Interface melalui konsol, dan cara menghapus grup fitur menggunakan konsol dan. AWS SDK for Python (Boto3)

Petunjuk tentang penggunaan Feature Store melalui konsol bergantung pada apakah Anda telah mengaktifkan Studio atau Studio Classic sebagai pengalaman default Anda. Untuk informasi tentang mengakses Studio Classic, lihatLuncurkan Studio Classic Menggunakan SageMaker Konsol Amazon.

Topik

Konsep Feature Store

Kami mencantumkan istilah umum yang digunakan di Amazon SageMaker Feature Store, diikuti dengan contoh diagram untuk memvisualisasikan beberapa konsep:

- Feature Store: Lapisan penyimpanan dan manajemen data untuk fitur pembelajaran mesin (ML). Berfungsi sebagai satu-satunya sumber kebenaran untuk menyimpan, mengambil, menghapus, melacak, berbagi, menemukan, dan mengontrol akses ke fitur. Dalam diagram contoh berikut, Feature Store adalah toko untuk grup fitur Anda, yang berisi data ML Anda, dan menyediakan layanan tambahan.

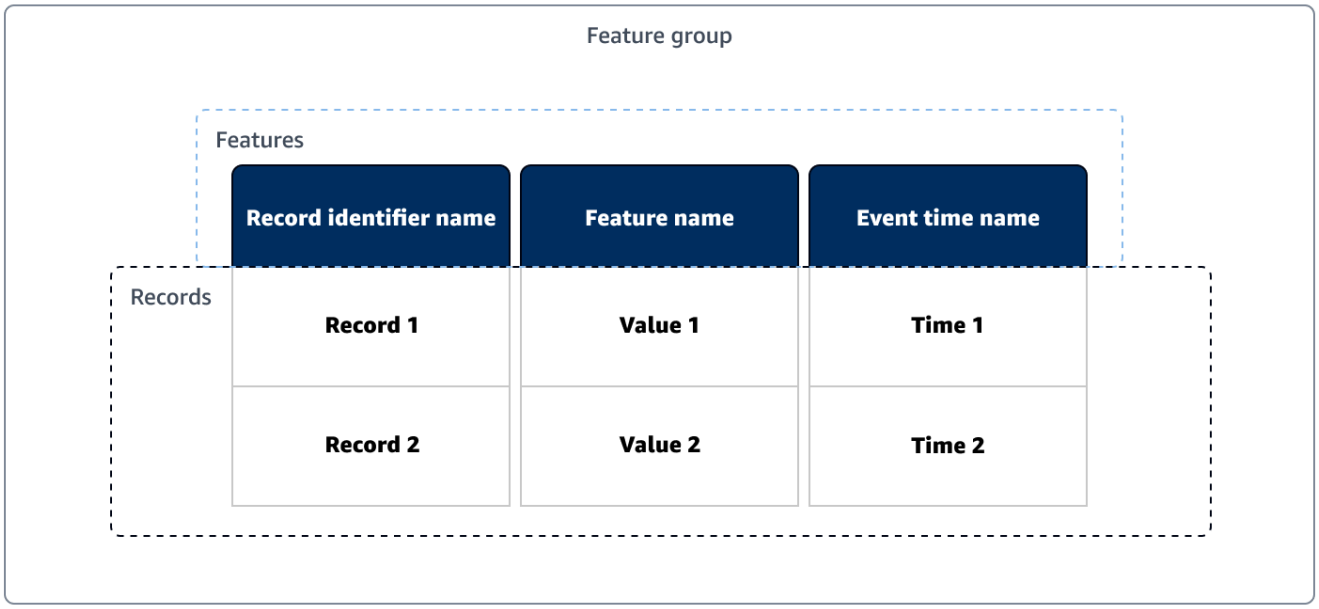
- Toko online: Latensi rendah, toko ketersediaan tinggi untuk grup fitur yang memungkinkan pencarian catatan secara real-time. Toko online memungkinkan akses cepat ke catatan terbaru melalui `GetRecord` API.
- Toko offline: Menyimpan data historis di bucket Amazon S3 Anda. Toko offline digunakan ketika pembacaan latensi rendah (sub-detik) tidak diperlukan. Misalnya, toko offline dapat digunakan saat Anda ingin menyimpan dan menyajikan fitur untuk eksplorasi, pelatihan model, dan inferensi batch.
- Grup fitur: Sumber daya utama Feature Store yang berisi data dan metadata yang digunakan untuk pelatihan atau prediksi dengan model ML. Grup fitur adalah pengelompokan logis fitur yang digunakan untuk menggambarkan catatan. Dalam contoh diagram berikut, grup fitur berisi data ML Anda.
- Fitur: Properti yang digunakan sebagai salah satu input untuk melatih atau memprediksi menggunakan model ML Anda. Di Feature Store API, fitur adalah atribut catatan. Dalam contoh diagram berikut, fitur menjelaskan kolom dalam tabel data ML Anda.
- Definisi fitur: Terdiri dari nama dan salah satu tipe data: integral, string atau fraksional. Grup fitur berisi daftar definisi fitur. Untuk informasi selengkapnya tentang tipe data Feature Store, lihatTipe Data.
- Rekam: Kumpulan nilai untuk fitur untuk pengenal rekaman tunggal. Kombinasi pengenal rekaman dan nilai waktu peristiwa secara unik mengidentifikasi catatan dalam grup fitur. Dalam contoh diagram berikut, catatan adalah baris dalam tabel data ML Anda.
- Nama pengenal rekaman: Nama pengenal catatan adalah nama fitur yang mengidentifikasi catatan. Ini harus merujuk ke salah satu nama fitur yang didefinisikan dalam definisi fitur grup fitur. Setiap grup fitur didefinisikan dengan nama pengenal catatan.
- Waktu acara: Stempel waktu yang Anda berikan sesuai dengan saat peristiwa rekaman terjadi. Semua catatan dalam grup fitur harus memiliki waktu acara yang sesuai. Toko online hanya berisi catatan yang sesuai dengan waktu acara terbaru, sedangkan toko offline berisi semua catatan sejarah. Untuk informasi selengkapnya tentang format waktu acara, lihatTipe Data.
- Ingestion: Menambahkan catatan baru ke grup fitur. Tertelan biasanya dicapai melalui API. `PutRecord`

Topik

- Diagram ikhtisar konsep
- Penyempurnaan satu spasi

Diagram ikhtisar konsep

Contoh diagram berikut mengkonseptualisasikan beberapa konsep Feature Store:



Toko Fitur berisi grup fitur Anda dan grup fitur berisi data ML Anda. Dalam diagram contoh, grup fitur asli berisi tabel data yang memiliki tiga fitur (masing-masing menggambarkan kolom) dan dua catatan (baris).

- Definisi fitur menjelaskan nama fitur dan tipe data dari nilai fitur yang terkait dengan catatan.
- Catatan berisi nilai fitur dan diidentifikasi secara unik oleh pengidentifikasi catatannya dan harus menyertakan waktu acara.

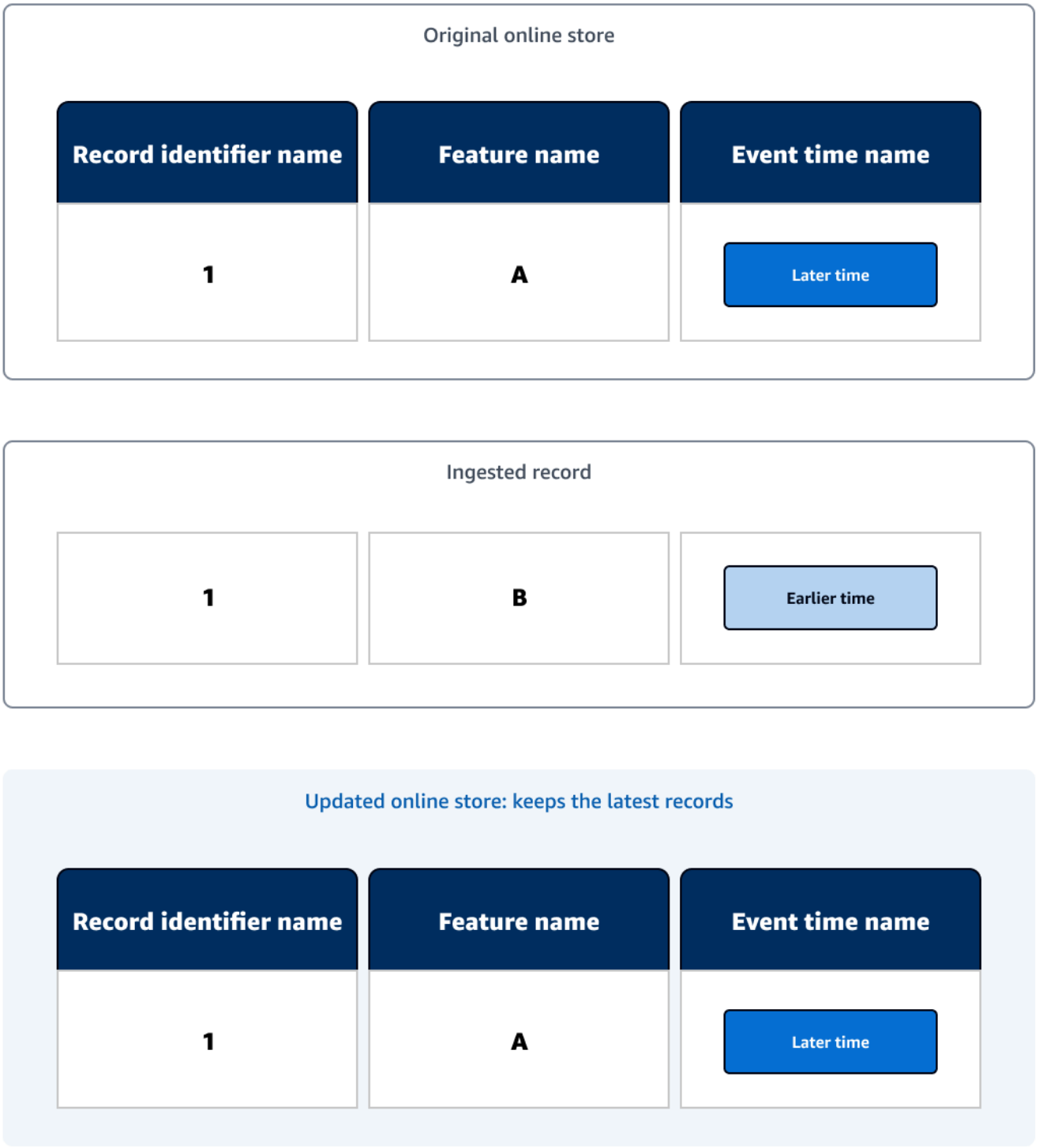
Penyempurnaan satu spasi

Ingestion adalah tindakan menambahkan catatan atau catatan ke grup fitur yang ada. Toko online dan offline diperbarui secara berbeda untuk kasus penggunaan penyimpanan yang berbeda.

Tertelan ke contoh toko online

Toko online bertindak sebagai pencarian catatan waktu nyata dan hanya menyimpan sebagian besar up-to-date catatan. Setelah catatan dicerna ke toko online yang ada, toko online yang diperbarui hanya akan menyimpan catatan dengan waktu acara terbaru.

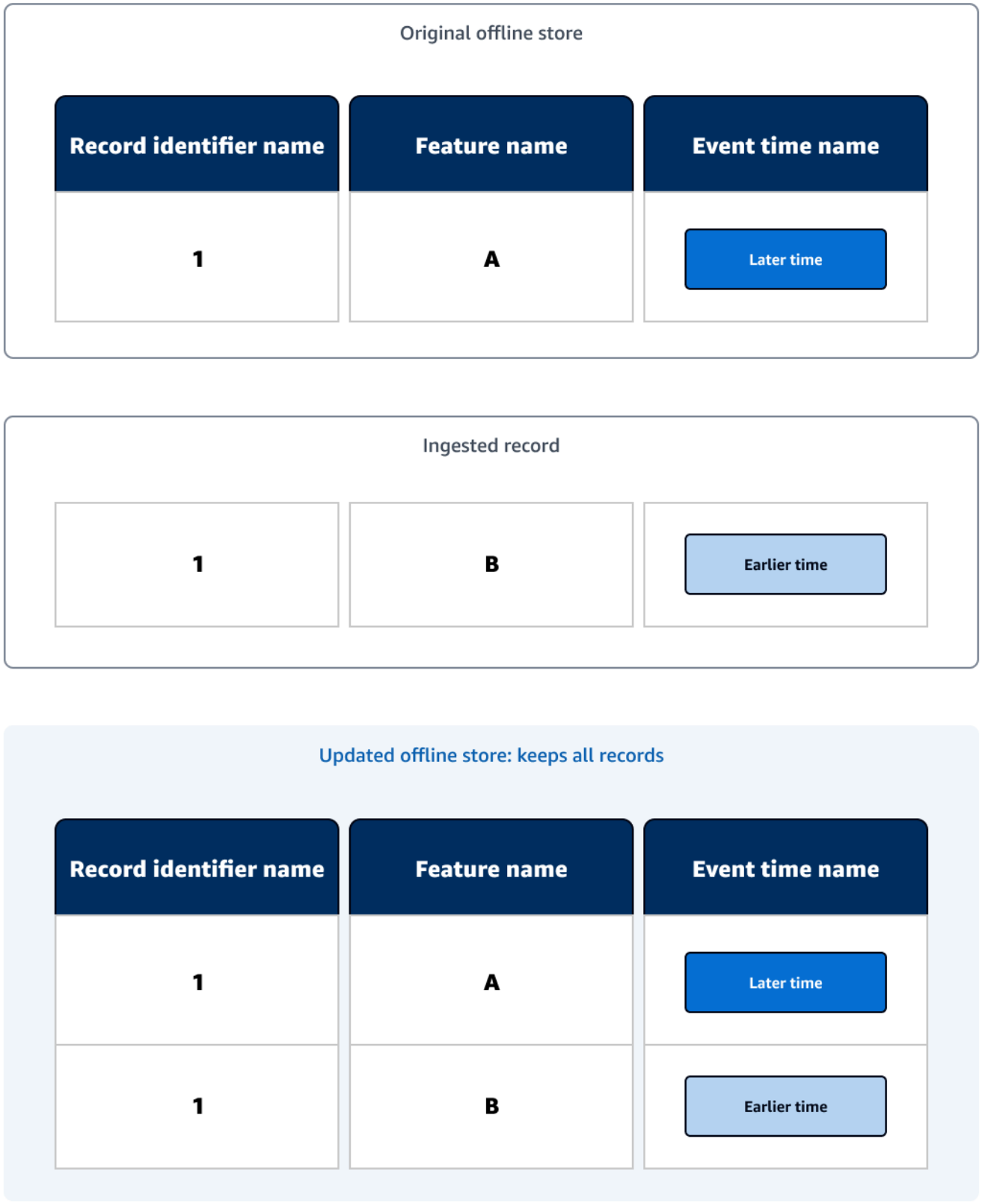
Dalam contoh diagram berikut, toko online asli berisi tabel data ML dengan satu catatan. Rekaman dicerna dengan nama pengenal rekaman yang sama dengan catatan asli, dan catatan yang dicerna memiliki waktu peristiwa yang lebih awal dari catatan aslinya. Karena toko online yang diperbarui hanya menyimpan catatan dengan waktu acara terbaru, toko online yang diperbarui berisi catatan asli.



Tertelan ke contoh toko offline

Toko offline bertindak sebagai pencarian sejarah catatan dan menyimpan semua catatan. Setelah catatan baru dicerna ke toko offline yang ada, toko offline yang diperbarui akan menyimpan rekor baru.

Dalam contoh diagram berikut, toko offline asli berisi tabel data ML dengan satu catatan. Rekaman dicerna dengan nama pengenal rekaman yang sama dengan catatan asli, dan catatan yang dicerna memiliki waktu peristiwa lebih awal dari catatan aslinya. Karena toko offline yang diperbarui menyimpan semua catatan, toko offline yang diperbarui berisi kedua catatan.


Original offline store
Record identifier name
Feature name
Event time name
1
A
Later time
Ingested record
1
B
Earlier time
Updated offline store: keeps all records
Record identifier name
Feature name
Event time name
1
A
Later time
1
B
Earlier time

Menambahkan kebijakan ke peran IAM Anda

Untuk memulai Amazon SageMaker Feature Store, Anda harus memiliki peran dan menambahkan kebijakan yang diperlukan ke peran Anda`AmazonSageMakerFeatureStoreAccess`. Berikut ini adalah panduan tentang cara melihat kebijakan yang dilampirkan pada peran dan cara menambahkan kebijakan ke peran Anda. Untuk informasi tentang cara membuat peran, lihatSageMaker Peran. Untuk informasi tentang cara mendapatkan peran eksekusi Anda, lihatDapatkan peran eksekusi.

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi di sebelah kiri konsol IAM, pilih Peran.
3. Di bilah pencarian masukkan peran yang Anda gunakan untuk Amazon SageMaker Feature Store.

   Untuk contoh tentang cara menemukan peran eksekusi ARN untuk notebook di dalamnya SageMaker, lihat. Dapatkan peran eksekusi Peran tersebut berada di akhir peran eksekusi ARN.
4. Setelah Anda memasukkan peran di bilah pencarian, pilih peran.

   Di bawah Kebijakan Izin, Anda dapat melihat kebijakan yang dilampirkan pada peran tersebut.
5. Setelah Anda memilih peran, pilih Tambahkan izin, lalu pilih Lampirkan kebijakan.
6. Di bilah pencarian di bawah Kebijakan izin lainnya masuk `AmazonSageMakerFeatureStoreAccess` dan tekan enter. Jika kebijakan tidak ditampilkan, Anda mungkin sudah memiliki kebijakan yang terlampir, yang tercantum di bawah kebijakan izin saat ini.
7. Setelah Anda menekan enter, pilih kotak centang di sebelah kebijakan, lalu pilih Tambahkan izin.
8. Setelah Anda melampirkan kebijakan ke peran Anda, kebijakan akan muncul di bawah Kebijakan Izin untuk peran IAM Anda.

Gunakan Feature Store dengan SDK for Python (Boto3)

Grup fitur adalah sumber daya Toko Fitur utama yang berisi data machine learning (ML) dan metadata yang disimpan di Amazon SageMaker Feature Store. Grup fitur adalah pengelompokan logis fitur dan catatan. Definisi grup fitur terdiri dari konfigurasi untuk toko online dan offline dan daftar definisi fitur yang digunakan untuk menggambarkan nilai catatan Anda. Definisi fitur harus menyertakan nama pengenal catatan dan nama waktu acara. Untuk informasi selengkapnya tentang konsep feature store, lihatKonsep Feature Store.

Sebelum menggunakan feature store, Anda biasanya memuat kumpulan data, menjalankan transformasi, dan menyiapkan fitur untuk dikonsumsi. Proses ini memiliki banyak variasi dan sangat tergantung pada data Anda. Kode contoh dalam topik berikut mengacu pada contoh buku catatan Pengantar Toko Fitur dan Deteksi Penipuan dengan Amazon SageMaker Feature Store. Keduanya menggunakanAWS SDK for Python (Boto3). Untuk contoh dan sumber daya Toko Fitur lainnya, lihatSumber daya Toko SageMaker Fitur Amazon.

Feature Store mendukung jenis fitur berikut:`String`, `Fractional` (nilai floating point IEEE 64-bit), dan `Integral` (nilai integral bertanda Int64 - 64 bit). Jenis default diatur ke`String`. Ini berarti bahwa, jika kolom dalam kumpulan data Anda bukan dari tipe `float` atau `long` fitur, itu default di `String` feature store Anda.

Anda dapat menggunakan skema untuk mendeskripsikan kolom dan tipe data data Anda. Anda meneruskan skema ini ke dalam`FeatureDefinitions`, parameter yang diperlukan untuk a`FeatureGroup`. Anda dapat menggunakan SDK for Python (Boto3), yang memiliki deteksi tipe data otomatis saat Anda menggunakan fungsi tersebut. `load_feature_definitions`

Perilaku default saat catatan fitur baru ditambahkan dengan ID rekaman yang sudah ada adalah sebagai berikut. Di toko offline, catatan baru akan ditambahkan. Di toko online, jika waktu acara rekaman baru kurang dari waktu acara yang ada maka tidak akan terjadi apa-apa, tetapi jika waktu peristiwa dari catatan baru lebih besar dari atau sama dengan waktu acara yang ada, catatan akan ditimpa.

Saat Anda membuat grup fitur baru, Anda dapat memilih salah satu format tabel berikut:

- AWS Glue(Default)
- Fungsi SQL yang lebih besar

Menelan data, terutama saat streaming, dapat mengakibatkan sejumlah besar file kecil disimpan ke toko offline. Ini dapat berdampak negatif pada kinerja kueri karena jumlah operasi file yang lebih tinggi yang diperlukan. Untuk menghindari potensi masalah kinerja, gunakan format tabel Apache Iceberg saat membuat grup fitur baru. Dengan Iceberg Anda dapat memadatkan file data kecil menjadi lebih sedikit file besar di partisi, menghasilkan kueri yang jauh lebih cepat. Operasi pemadatan ini bersamaan dan tidak memengaruhi operasi baca dan tulis yang sedang berlangsung pada grup fitur. Jika Anda memilih opsi Iceberg saat membuat grup fitur baru, Amazon SageMaker Feature Store akan membuat tabel Iceberg menggunakan format file Parket, dan mendaftarkan tabel dengan. AWS Glue Data Catalog

> **Important**
>
> Perhatikan bahwa untuk grup fitur dalam format tabel Iceberg, Anda harus menentukan `String` sebagai nilai untuk waktu acara. Jika Anda menentukan jenis lainnya, Anda tidak dapat membuat grup fitur dengan sukses.

Berikut ini kami mencantumkan beberapa sumber daya terkelola Feature Store yang tersedia.

Topik

- Pengantar buku catatan contoh Toko Fitur
- Deteksi penipuan dengan notebook contoh Feature Store

Pengantar buku catatan contoh Toko Fitur

Kode contoh pada halaman ini mengacu pada contoh buku catatan Pengantar Toko Fitur. Kami menyarankan Anda menjalankan notebook ini di Studio Classic, instance notebook, atau JupyterLab karena kode dalam panduan ini konseptual dan tidak berfungsi penuh jika disalin.

Gunakan yang berikut ini untuk mengkloning aws/ amazon-sagemaker-examples GitHub repositori, yang berisi contoh notebook:

- Untuk Studio Klasik

  Luncurkan Studio Klasik. Anda dapat membuka Studio Classic jika Studio atau Studio Classic diaktifkan sebagai pengalaman default Anda. Untuk instruksi tentang cara membuka Studio Classic, lihatLuncurkan Studio Classic Menggunakan SageMaker Konsol Amazon.

  Kloning aws/ amazon-sagemaker-examples GitHub repositori ke Studio Classic dengan mengikuti langkah-langkah di. Mengkloning Repositori Git di Studio Classic SageMaker

- Untuk instance SageMaker notebook Amazon

  Luncurkan instance SageMaker notebook dengan mengikuti instruksi diAkses Instans Notebook.

  Periksa apakah contoh sudah ada di buku catatan Anda dengan mengikuti instruksi diContoh Notebook. Jika tidak, ikuti instruksi diTambahkan Repositori Git ke Akun Amazon Anda SageMaker .

Sekarang setelah Anda memiliki SageMaker contoh notebook, navigasikan ke `amazon-sagemaker-examples/sagemaker-featurestore` direktori dan buka buku catatan contoh Pengantar ke Feature Store.

Langkah 1: Atur SageMaker sesi Anda

Untuk mulai menggunakan Feature Store, buat SageMaker sesi. Kemudian, siapkan bucket Amazon Simple Storage Service (Amazon S3) yang ingin Anda gunakan untuk fitur Anda. Bucket Amazon S3 adalah toko offline Anda. Kode berikut menggunakan bucket SageMaker default dan menambahkan awalan kustom ke dalamnya.

> **Note**
>
> Peran yang Anda gunakan untuk menjalankan buku catatan harus memiliki kebijakan terkelola berikut yang dilampirkan padanya: `AmazonS3FullAccess` dan`AmazonSageMakerFeatureStoreAccess`. Untuk informasi tentang menambahkan kebijakan ke peran IAM Anda, lihatMenambahkan kebijakan ke peran IAM Anda.

```
# SageMaker Python SDK version 2.x is required
import sagemaker
import sys
```

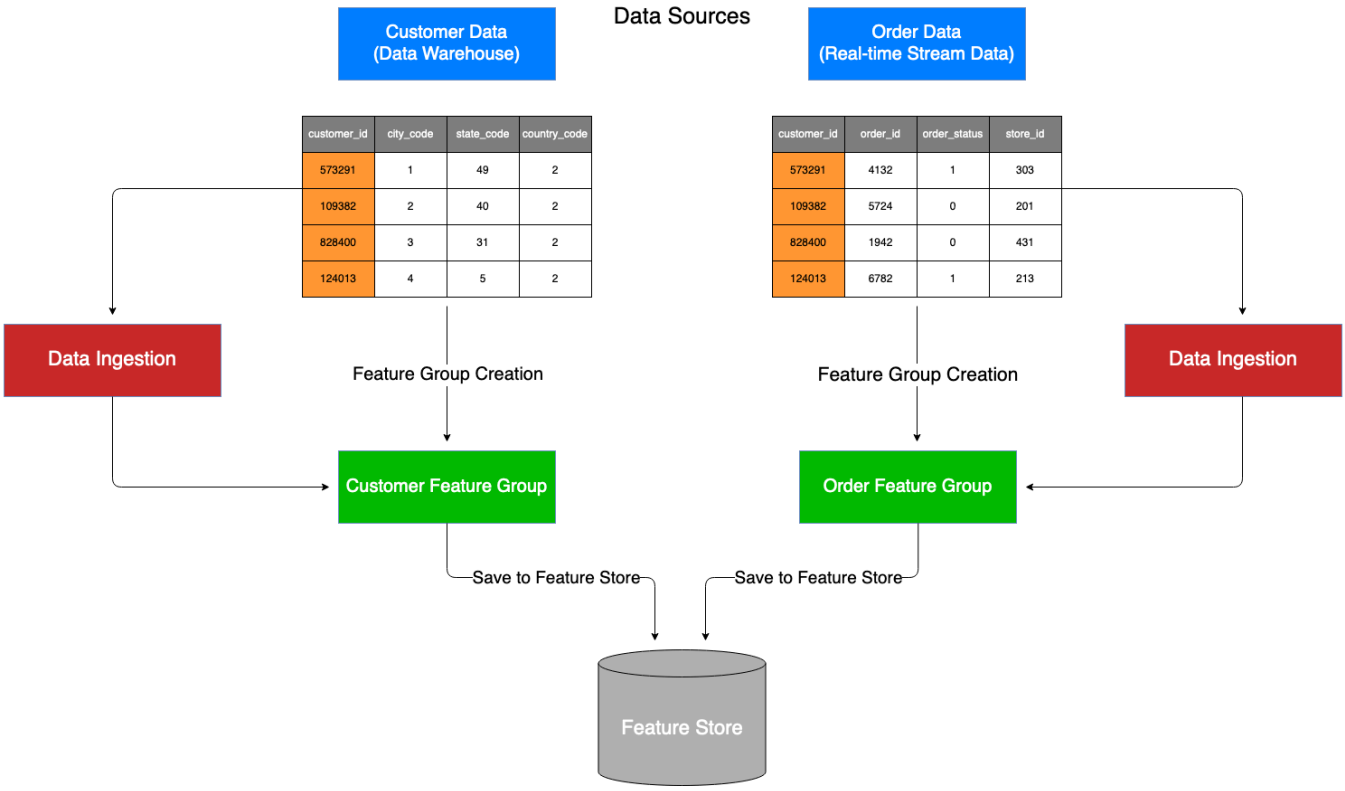
```
import boto3
import pandas as pd
import numpy as np
import io
from sagemaker.session import Session
from sagemaker import get_execution_role

prefix = 'sagemaker-featurestore-introduction'
role = get_execution_role()

sagemaker_session = sagemaker.Session()
region = sagemaker_session.boto_region_name
s3_bucket_name = sagemaker_session.default_bucket()
```

Langkah 2: Periksa data Anda

Dalam contoh notebook ini, kami menelan data sintetis dari GitHub repositori yang menampung notebook lengkap.

```
customer_data = pd.read_csv("data/feature_store_introduction_customer.csv")
orders_data = pd.read_csv("data/feature_store_introduction_orders.csv")

print(customer_data.head())
print(orders_data.head())
```

Diagram berikut mengilustrasikan langkah-langkah yang dilalui data sebelum Feature Store menelannya. Dalam buku catatan ini, kami mengilustrasikan kasus penggunaan di mana Anda memiliki data dari berbagai sumber dan ingin menyimpannya secara independen di Toko Fitur. Contoh kami mempertimbangkan data dari gudang data (data pelanggan), dan data dari layanan streaming real-time (data pesanan).



Langkah 3: Membuat grup fitur

Pertama-tama kita mulai dengan membuat nama grup fitur untuk customer_data dan orders_data. Setelah ini, kami membuat dua grup fitur, satu untuk `customer_data` dan satu lagi untuk`orders_data`:

```
import time
from time import strftime, gmtime
customers_feature_group_name = 'customers-feature-group-' + strftime('%d-%H-%M-%S',
 gmtime())
orders_feature_group_name = 'orders-feature-group-' + strftime('%d-%H-%M-%S', gmtime())
```

Buat instance `FeatureGroup` objek untuk `customers_data` dan: `orders_data`

```
from sagemaker.feature_store.feature_group import FeatureGroup

customers_feature_group = FeatureGroup(
    name=customers_feature_group_name, sagemaker_session=sagemaker_session
)
orders_feature_group = FeatureGroup(
    name=orders_feature_group_name, sagemaker_session=sagemaker_session
)
```

```
import time
current_time_sec = int(round(time.time()))
record_identifier_feature_name = "customer_id"
```

Tambahkan `EventTime` fitur ke bingkai data Anda. Parameter ini diperlukan, dan stempel waktu setiap titik data:

```
customer_data["EventTime"] = pd.Series([current_time_sec]*len(customer_data),
 dtype="float64")
orders_data["EventTime"] = pd.Series([current_time_sec]*len(orders_data),
 dtype="float64")
```

Muat definisi fitur ke grup fitur Anda:

```
customers_feature_group.load_feature_definitions(data_frame=customer_data)
orders_feature_group.load_feature_definitions(data_frame=orders_data)
```

Panggilan berikut `create` untuk membuat dua grup fitur, `customers_feature_group` dan`orders_feature_group`, masing-masing:

```
customers_feature_group.create(
    s3_uri=f"s3://{s3_bucket_name}/{prefix}",
    record_identifier_name=record_identifier_feature_name,
    event_time_feature_name="EventTime",
    role_arn=role,
    enable_online_store=True
)

orders_feature_group.create(
    s3_uri=f"s3://{s3_bucket_name}/{prefix}",
    record_identifier_name=record_identifier_feature_name,
    event_time_feature_name="EventTime",
    role_arn=role,
    enable_online_store=True
)
```

Untuk mengonfirmasi bahwa grup fitur Anda telah dibuat, kami menampilkannya dengan menggunakan `DescribeFeatureGroup` dan `ListFeatureGroups` API:

```
customers_feature_group.describe()
```

```
orders_feature_group.describe()
```

```
sagemaker_session.boto_session.client('sagemaker',
 region_name=region).list_feature_groups() # We use the boto client to list
 FeatureGroups
```

Langkah 4: Menyerap data ke dalam grup fitur

Setelah grup fitur dibuat, kita dapat memasukkan data ke dalamnya. Jika Anda menggunakan SageMakerAWS SDK for Python (Boto3), gunakan panggilan `ingest` API. Jika Anda menggunakan SDK for Python (Boto3), gunakan API. `PutRecord` Diperlukan waktu kurang dari 1 menit untuk menelan data kedua opsi ini. Contoh ini menggunakan SageMaker SDK for Python (Boto3), sehingga menggunakan panggilan API: `ingest`

```
def check_feature_group_status(feature_group):
    status = feature_group.describe().get("FeatureGroupStatus")
```

```
    while status == "Creating":
        print("Waiting for Feature Group to be Created")
        time.sleep(5)
        status = feature_group.describe().get("FeatureGroupStatus")
    print(f"FeatureGroup {feature_group.name} successfully created.")

check_feature_group_status(customers_feature_group)
check_feature_group_status(orders_feature_group)
```

```
customers_feature_group.ingest(
    data_frame=customer_data, max_workers=3, wait=True
)
```

```
orders_feature_group.ingest(
    data_frame=orders_data, max_workers=3, wait=True
)
```

Menggunakan id catatan pelanggan arbitrer, 573291 kami gunakan `get_record` untuk memeriksa apakah data telah dicerna ke dalam grup fitur.

```
customer_id = 573291
sample_record = sagemaker_session.boto_session.client('sagemaker-featurestore-runtime',
 region_name=region).get_record(FeatureGroupName=customers_feature_group_name,
 RecordIdentifierValueAsString=str(customer_id))
```

```
print(sample_record)
```

Berikut ini menunjukkan bagaimana menggunakan `batch_get_record` untuk mendapatkan batch catatan.

```
all_records = sagemaker_session.boto_session.client(
    "sagemaker-featurestore-runtime", region_name=region
).batch_get_record(
    Identifiers=[
        {
            "FeatureGroupName": customers_feature_group_name,
            "RecordIdentifiersValueAsString": ["573291", "109382", "828400", "124013"],
        },
        {
            "FeatureGroupName": orders_feature_group_name,
```

```
            "RecordIdentifiersValueAsString": ["573291", "109382", "828400", "124013"],
        },
    ]
)
```

```
print(all_records)
```

Langkah 5: Bersihkan

Di sini kita menghapus Grup Fitur yang kita buat.

```
customers_feature_group.delete()
orders_feature_group.delete()
```

Langkah 6: Langkah selanjutnya

Di buku catatan contoh ini, Anda mempelajari cara memulai dengan Feature Store, membuat grup fitur, dan memasukkan data ke dalamnya.

Untuk contoh lanjutan tentang cara menggunakan Toko Fitur untuk kasus penggunaan deteksi penipuan, lihat Deteksi Penipuan dengan Toko Fitur.

Langkah 7: Contoh kode untuk programmer

Di notebook ini kami menggunakan berbagai panggilan API yang berbeda. Sebagian besar dari mereka dapat diakses melalui SageMaker Python SDK, namun beberapa hanya ada dalam Boto3. Anda dapat memanggil panggilan API SDK SageMaker Python langsung pada objek Feature Store Anda, sedangkan untuk memanggil panggilan API yang ada dalam Boto3, Anda harus terlebih dahulu mengakses klien Boto3 melalui Boto3 dan sesi Anda: misalnya,. SageMaker `sagemaker_session.boto_session.client()`

Berikut ini adalah daftar panggilan API untuk notebook ini. Panggilan ini ada di dalam SDK for Python dan ada di Boto3, untuk referensi Anda:

Panggilan API SDK for Python (Boto3)

```
describe()
ingest()
delete()
create()
load_feature_definitions()
```

Panggilan API Boto3

```
list_feature_groups()
get_record()
```

Deteksi penipuan dengan notebook contoh Feature Store

Kode contoh pada halaman ini mengacu pada contoh buku catatan: Deteksi Penipuan dengan Amazon SageMaker Feature Store. Kami menyarankan Anda menjalankan notebook ini di Studio Classic, instance notebook, atau Jupyter Lab karena kode dalam panduan ini bersifat konseptual dan tidak berfungsi penuh jika disalin.

Gunakan yang berikut ini untuk mengkloning aws/ amazon-sagemaker-examples GitHub repositori, yang berisi contoh notebook.

- Untuk Studio Klasik

  Peluncuran pertama Studio Classic. Anda dapat membuka Studio Classic jika Studio atau Studio Classic diaktifkan sebagai pengalaman default Anda. Untuk membuka Studio Classic, lihatLuncurkan Studio Classic Menggunakan SageMaker Konsol Amazon.

  Kloning aws/ amazon-sagemaker-examples GitHub repositori ke Studio Classic dengan mengikuti langkah-langkah di. Mengkloning Repositori Git di Studio Classic SageMaker
- Untuk instance SageMaker notebook Amazon

  Pertama luncurkan contoh SageMaker notebook dengan mengikuti instruksi diAkses Instans Notebook.

  Periksa apakah contoh sudah ada di buku catatan Anda dengan mengikuti instruksi diContoh Notebook. Jika tidak, ikuti instruksi diTambahkan Repositori Git ke Akun Amazon Anda SageMaker .

Sekarang setelah Anda memiliki SageMaker contoh notebook, navigasikan ke `amazon-sagemaker-examples/sagemaker-featurestore` direktori dan buka contoh notebook Deteksi Penipuan dengan Amazon SageMaker Feature Store.

Langkah 1: Atur sesi Feature Store

Untuk mulai menggunakan Feature Store, buat SageMaker sesi, sesi Boto3, dan sesi Feature Store. Selain itu, siapkan bucket Amazon S3 yang ingin Anda gunakan untuk fitur Anda. Ini adalah toko

offline Anda. Kode berikut menggunakan bucket SageMaker default dan menambahkan awalan kustom ke dalamnya.

> **Note**
>
> Peran yang Anda gunakan untuk menjalankan buku catatan harus memiliki kebijakan terkelola berikut yang dilampirkan padanya: `AmazonSageMakerFullAccess` dan`AmazonSageMakerFeatureStoreAccess`. Untuk informasi tentang menambahkan kebijakan ke peran IAM Anda, lihat[Menambahkan kebijakan ke peran IAM Anda](#).

```
import boto3
import sagemaker
from sagemaker.session import Session

sagemaker_session = sagemaker.Session()
region = sagemaker_session.boto_region_name
boto_session = boto3.Session(region_name=region)
role = sagemaker.get_execution_role()
default_bucket = sagemaker_session.default_bucket()
prefix = 'sagemaker-featurestore'
offline_feature_store_bucket = 's3://{}/{}'.format(default_bucket, prefix)

sagemaker_client = boto_session.client(service_name='sagemaker', region_name=region)
featurestore_runtime = boto_session.client(service_name='sagemaker-featurestore-
runtime', region_name=region)

feature_store_session = Session(
    boto_session=boto_session,
    sagemaker_client=sagemaker_client,
    sagemaker_featurestore_runtime_client=featurestore_runtime
)
```

Langkah 2: Muat dataset dan partisi data ke dalam grup fitur

Muat data Anda ke dalam bingkai data untuk setiap fitur Anda. Anda menggunakan bingkai data ini setelah mengatur grup fitur. Dalam contoh deteksi penipuan, Anda dapat melihat langkah-langkah ini dalam kode berikut.

```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import io

s3_client = boto3.client(service_name='s3', region_name=region)

fraud_detection_bucket_name = 'sagemaker-featurestore-fraud-detection'
identity_file_key = 'sampled_identity.csv'
transaction_file_key = 'sampled_transactions.csv'

identity_data_object = s3_client.get_object(Bucket=fraud_detection_bucket_name,
 Key=identity_file_key)
transaction_data_object = s3_client.get_object(Bucket=fraud_detection_bucket_name,
 Key=transaction_file_key)

identity_data = pd.read_csv(io.BytesIO(identity_data_object['Body'].read()))
transaction_data = pd.read_csv(io.BytesIO(transaction_data_object['Body'].read()))

identity_data = identity_data.round(5)
transaction_data = transaction_data.round(5)

identity_data = identity_data.fillna(0)
transaction_data = transaction_data.fillna(0)

# Feature transformations for this dataset are applied before ingestion into
 FeatureStore.
# One hot encode card4, card6
encoded_card_bank = pd.get_dummies(transaction_data['card4'], prefix = 'card_bank')
encoded_card_type = pd.get_dummies(transaction_data['card6'], prefix = 'card_type')

transformed_transaction_data = pd.concat([transaction_data, encoded_card_type,
 encoded_card_bank], axis=1)
transformed_transaction_data =
 transformed_transaction_data.rename(columns={"card_bank_american express":
 "card_bank_american_express"})
```

Langkah 3: Mengatur grup fitur

Saat menyiapkan grup fitur, Anda perlu menyesuaikan nama fitur dengan nama unik dan mengatur setiap grup fitur dengan `FeatureGroup` kelas.

```
from sagemaker.feature_store.feature_group import FeatureGroup
feature_group_name = "some string for a name"
```

```
feature_group = FeatureGroup(name=feature_group_name,
 sagemaker_session=feature_store_session)
```

Misalnya, dalam contoh deteksi penipuan, dua grup fitur adalah `identity` dan`transaction`. Dalam kode berikut, Anda dapat melihat bagaimana nama disesuaikan dengan stempel waktu, dan kemudian setiap grup diatur dengan meneruskan nama dan sesi.

```
import time
from time import gmtime, strftime, sleep
from sagemaker.feature_store.feature_group import FeatureGroup

identity_feature_group_name = 'identity-feature-group-' + strftime('%d-%H-%M-%S',
 gmtime())
transaction_feature_group_name = 'transaction-feature-group-' + strftime('%d-%H-%M-%S',
 gmtime())

identity_feature_group = FeatureGroup(name=identity_feature_group_name,
 sagemaker_session=feature_store_session)
transaction_feature_group = FeatureGroup(name=transaction_feature_group_name,
 sagemaker_session=feature_store_session)
```

Langkah 4: Mengatur pengenal catatan dan fitur waktu acara

Pada langkah ini, Anda menentukan nama pengenal catatan dan nama fitur waktu acara. Nama ini dipetakan ke kolom fitur yang sesuai dalam data Anda. Misalnya, dalam contoh deteksi penipuan, kolom yang diminati adalah`TransactionID`. `EventTime`dapat ditambahkan ke data Anda ketika tidak ada stempel waktu yang tersedia. Dalam kode berikut, Anda dapat melihat bagaimana variabel-variabel ini diatur, dan kemudian `EventTime` ditambahkan ke data kedua fitur ini.

```
record_identifier_name = "TransactionID"
event_time_feature_name = "EventTime"
current_time_sec = int(round(time.time()))
identity_data[event_time_feature_name] =
 pd.Series([current_time_sec]*len(identity_data), dtype="float64")
transformed_transaction_data[event_time_feature_name] =
 pd.Series([current_time_sec]*len(transaction_data), dtype="float64")
```

Langkah 5: Muat definisi fitur

Anda sekarang dapat memuat definisi fitur dengan melewatkan bingkai data yang berisi data fitur. Dalam kode berikut untuk contoh deteksi penipuan, fitur identitas dan fitur transaksi masing-

masing dimuat dengan menggunakan`load_feature_definitions`, dan fungsi ini secara otomatis mendeteksi tipe data dari setiap kolom data. Untuk pengembang yang menggunakan skema daripada deteksi otomatis, lihat contoh Ekspor Grup Fitur dari Data Wrangler untuk kode yang menunjukkan cara memuat skema, memetakannya, dan menambahkannya sebagai kode `FeatureDefinition` yang dapat Anda gunakan untuk membuat skema. `FeatureGroup` Contoh ini juga mencakup AWS SDK for Python (Boto3) implementasi, yang dapat Anda gunakan alih-alih SageMaker Python SDK.

```
identity_feature_group.load_feature_definitions(data_frame=identity_data); # output is
 suppressed
transaction_feature_group.load_feature_definitions(data_frame=transformed_transaction_data);
 # output is suppressed
```

Langkah 6: Membuat grup fitur

Pada langkah ini, Anda menggunakan `create` fungsi untuk membuat grup fitur. Kode berikut menunjukkan semua parameter yang tersedia. Toko online tidak dibuat secara default, jadi Anda harus mengatur ini `True` seolah-olah Anda ingin mengaktifkannya. `s3_uri`Ini adalah lokasi bucket S3 dari toko offline Anda.

```
# create a FeatureGroup
feature_group.create(
    description = "Some info about the feature group",
    feature_group_name = feature_group_name,
    record_identifier_name = record_identifier_name,
    event_time_feature_name = event_time_feature_name,
    feature_definitions = feature_definitions,
    role_arn = role,
    s3_uri = offline_feature_store_bucket,
    enable_online_store = True,
    online_store_kms_key_id = None,
    offline_store_kms_key_id = None,
    disable_glue_table_creation = False,
    data_catalog_config = None,
    tags = ["tag1","tag2"])
```

Kode berikut dari contoh deteksi penipuan menunjukkan `create` panggilan minimal untuk masing-masing dari dua grup fitur yang sedang dibuat.

```
identity_feature_group.create(
```

```
    s3_uri=offline_feature_store_bucket,
    record_identifier_name=record_identifier_name,
    event_time_feature_name=event_time_feature_name,
    role_arn=role,
    enable_online_store=True
)

transaction_feature_group.create(
    s3_uri=offline_feature_store_bucket,
    record_identifier_name=record_identifier_name,
    event_time_feature_name=event_time_feature_name,
    role_arn=role,
    enable_online_store=True
)
```

Saat Anda membuat grup fitur, dibutuhkan waktu untuk memuat data, dan Anda harus menunggu hingga grup fitur dibuat sebelum Anda dapat menggunakannya. Anda dapat memeriksa status menggunakan metode berikut.

```
status = feature_group.describe().get("FeatureGroupStatus")
```

Saat grup fitur sedang dibuat, Anda menerima `Creating` sebagai tanggapan. Ketika langkah ini telah selesai dengan sukses, jawabannya adalah`Created`. Status lain yang mungkin adalah`CreateFailed`,`Deleting`, atau`DeleteFailed`.

Langkah 7: Bekerja dengan grup fitur

Setelah menyiapkan grup fitur, Anda dapat melakukan manapun dari tugas berikut:

Topik

- Jelaskan grup fitur
- Fungsi SQL baru
- Menempatkan catatan dalam grup fitur
- Mendapatkan catatan dari grup fitur
- Hasilkan perintah sarang DDL
- Membangun dataset pelatihan
- Menulis dan mengeksekusi kueri Athena
- Menghapus grup fitur

Jelaskan grup fitur

Anda dapat mengambil informasi tentang grup fitur Anda dengan `describe` fungsi tersebut.

```
feature_group.describe()
```

Fungsi SQL baru

Anda dapat mencantumkan semua grup fitur Anda dengan `list_feature_groups` fungsi tersebut.

```
sagemaker_client.list_feature_groups()
```

Menempatkan catatan dalam grup fitur

Anda dapat menggunakan `ingest` fungsi ini untuk memuat data fitur Anda. Anda meneruskan bingkai data data fitur, mengatur jumlah pekerja, dan memilih untuk menunggu kembali atau tidak. Contoh berikut menunjukkan penggunaan `ingest` fungsi.

```
feature_group.ingest(
    data_frame=feature_data, max_workers=3, wait=True
)
```

Untuk setiap grup fitur yang Anda miliki, jalankan `ingest` fungsi pada data fitur yang ingin Anda muat.

Mendapatkan catatan dari grup fitur

Anda dapat menggunakan `get_record` fungsi untuk mengambil data untuk fitur tertentu dengan pengenal catatannya. Contoh berikut menggunakan contoh identifier untuk mengambil catatan.

```
record_identifier_value = str(2990130)
featurestore_runtime.get_record(FeatureGroupName=transaction_feature_group_name,
 RecordIdentifierValueAsString=record_identifier_value)
```

Contoh respons dari contoh deteksi penipuan:

```
...
'Record': [{'FeatureName': 'TransactionID', 'ValueAsString': '2990130'},
```

```
  {'FeatureName': 'isFraud', 'ValueAsString': '0'},
  {'FeatureName': 'TransactionDT', 'ValueAsString': '152647'},
  {'FeatureName': 'TransactionAmt', 'ValueAsString': '75.0'},
  {'FeatureName': 'ProductCD', 'ValueAsString': 'H'},
  {'FeatureName': 'card1', 'ValueAsString': '4577'},
...
```

Hasilkan perintah sarang DDL

`FeatureStore`Kelas SageMaker Python SDK juga menyediakan fungsionalitas untuk menghasilkan perintah Hive DDL. Skema tabel dihasilkan berdasarkan definisi fitur. Kolom dinamai setelah nama fitur dan tipe data disimpulkan berdasarkan jenis fitur.

```
print(feature_group.as_hive_ddl())
```

Contoh output:

```
CREATE EXTERNAL TABLE IF NOT EXISTS sagemaker_featurestore.identity-feature-
group-27-19-33-00 (
  TransactionID INT
  id_01 FLOAT
  id_02 FLOAT
  id_03 FLOAT
  id_04 FLOAT
 ...
```

Membangun dataset pelatihan

Feature Store secara otomatis membuat katalog AWS Glue data saat Anda membuat grup fitur dan Anda dapat menonaktifkannya jika diinginkan. Berikut ini menjelaskan cara membuat kumpulan data pelatihan tunggal dengan nilai fitur dari grup fitur identitas dan transaksi yang dibuat sebelumnya dalam topik ini. Selain itu, berikut ini menjelaskan cara menjalankan kueri Amazon Athena untuk menggabungkan data yang disimpan di toko offline dari grup fitur identitas dan transaksi.

Untuk memulai, buat kueri Athena menggunakan `athena_query()` grup fitur identitas dan transaksi. `table_name` adalah AWS Glue tabel yang dibuat secara otomatis oleh Feature Store.

```
identity_query = identity_feature_group.athena_query()
transaction_query = transaction_feature_group.athena_query()
```

```
identity_table = identity_query.table_name
transaction_table = transaction_query.table_name
```

Menulis dan mengeksekusi kueri Athena

Anda menulis kueri menggunakan SQL pada grup fitur ini, lalu jalankan kueri dengan `.run()` perintah dan tentukan lokasi bucket Amazon S3 agar kumpulan data disimpan di sana.

```
# Athena query
query_string = 'SELECT * FROM "'+transaction_table+'" LEFT JOIN "'+identity_table+'" ON
 "'+transaction_table+'".transactionid = "'+identity_table+'".transactionid'

# run Athena query. The output is loaded to a Pandas dataframe.
dataset = pd.DataFrame()
identity_query.run(query_string=query_string,
 output_location='s3://'+default_s3_bucket_name+'/query_results/')
identity_query.wait()
dataset = identity_query.as_dataframe()
```

Dari sini Anda dapat melatih model menggunakan kumpulan data ini dan kemudian melakukan inferensi.

Menghapus grup fitur

Anda dapat menghapus grup fitur dengan `delete` fungsi tersebut.

```
feature_group.delete()
```

Contoh kode berikut adalah dari contoh deteksi penipuan.

```
identity_feature_group.delete()
transaction_feature_group.delete()
```

Untuk informasi selengkapnya, lihat API Hapus grup fitur.

Menggunakan Amazon SageMaker Feature Store di konsol

Anda dapat menggunakan Amazon SageMaker Feature Store di konsol untuk membuat, melihat, memperbarui, dan memantau grup fitur Anda. Pemantauan dalam panduan ini mencakup melihat

eksekusi pipeline dan garis keturunan grup fitur Anda. Panduan ini memberikan instruksi tentang cara mencapai tugas-tugas ini dari konsol.

Untuk contoh dan sumber daya Toko Fitur yang menggunakan Amazon SageMaker API danAWS SDK for Python (Boto3), lihatSumber daya Toko SageMaker Fitur Amazon.

Topik

- Buat grup fitur dari konsol
- Lihat detail grup fitur dari konsol
- Memperbarui grup fitur dari konsol
- Lihat eksekusi pipeline dari konsol
- Lihat silsilah dari konsol

Buat grup fitur dari konsol

Proses membuat grup fitur memiliki empat langkah:

1. Masukkan informasi grup fitur.
2. Masukkan definisi fitur.
3. Masukkan fitur yang diperlukan.
4. Masukkan tag grup fitur.

Pertimbangkan opsi mana yang sesuai dengan kasus penggunaan Anda:

- Buat toko online, toko offline, atau keduanya. Untuk informasi selengkapnya tentang perbedaan antara toko online dan offline, lihatKonsep Feature Store.
- Gunakan kunci default atau AWS Key Management Service kunci KMS Anda sendiri. Kunci default adalah AWS KMSkunci (SSE-KMS). Anda dapat mengurangi biaya AWS KMS permintaan dengan mengonfigurasi penggunaan Amazon S3 Bucket Keys di bucket Amazon S3 toko offline. Kunci Bucket Amazon S3 harus diaktifkan sebelum menggunakan bucket untuk grup fitur Anda. Untuk informasi selengkapnya tentang mengurangi biaya dengan menggunakan Kunci Bucket Amazon S3, lihat Mengurangi biaya SSE-KMS dengan Kunci Bucket Amazon S3.

  Anda dapat menggunakan kunci yang sama untuk toko online dan offline, atau memiliki kunci unik untuk masing-masing toko. Untuk informasi selengkapnya tentang AWS KMS, lihat AWS Key Management Service.

- Jika Anda membuat toko offline:
  - Putuskan apakah Anda ingin membuat bucket Amazon S3 atau menggunakan yang sudah ada. Saat menggunakan yang sudah ada, Anda harus mengetahui URL bucket Amazon S3 atau nama bucket Amazon S3 dan nama direktori kumpulan data, jika ada.
  - Pilih Amazon Resource Name (ARN) yang akan digunakan untuk menentukan IAM role. Untuk informasi selengkapnya tentang cara menemukan peran dan kebijakan terlampir, lihatMenambahkan kebijakan ke peran IAM Anda.
  - Putuskan apakah akan menggunakan format AWS Glue (default) atau Apache Iceberg tabel. Dalam kebanyakan kasus penggunaan, Anda menggunakan format Apache Iceberg tabel. Untuk informasi selengkapnya tentang format tabel, lihatGunakan Feature Store dengan SDK for Python (Boto3).

Anda dapat menggunakan konsol untuk melihat garis keturunan grup fitur. Petunjuk untuk menggunakan Feature Store di konsol bervariasi tergantung pada apakah Anda mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Buat grup fitur jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data dari panel navigasi kiri untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. Pilih Buat grup fitur.
5. Di bawah Detail grup fitur, masukkan nama grup fitur.
6. (Opsional) Masukkan deskripsi grup fitur.
7. Di bawah Konfigurasi penyimpanan grup fitur, pilih konfigurasi penyimpanan dari daftar tarik-turun. Untuk informasi tentang konfigurasi penyimpanan, lihatKonfigurasi penyimpanan Feature Store.
8. Jika Anda telah memilih untuk mengaktifkan penyimpanan online:
   a. Jika Anda hanya mengaktifkan penyimpanan online, Anda dapat memilih jenis Penyimpanan dari daftar dropdown. Untuk informasi tentang jenis penyimpanan toko online, lihatFungsi SQL baru.
   b. (Opsional) Terapkan Waktu ke Langsung (TTL) dengan mengaktifkan sakelar ke Aktif dan menentukan nilai dan unit Durasi Waktu ke Langsung. Ini akan memperbarui durasi TTL

default untuk semua catatan yang ditambahkan ke grup fitur setelah grup fitur dibuat. Untuk informasi selengkapnya tentang TTL, lihatDurasi waktu untuk tayang (TTL) untuk rekaman.

9. Jika Anda telah memilih untuk mengaktifkan penyimpanan offline:

   a. Di bawah nama bucket Amazon S3, masukkan nama bucket baru, atau masukkan URL bucket yang ada, secara manual.

   b. Dari daftar dropdown Format Tabel, pilih format tabel. Dalam kebanyakan kasus penggunaan, Anda harus menggunakan format Apache Iceberg tabel. Untuk informasi selengkapnya tentang format tabel, lihatGunakan Feature Store dengan SDK for Python (Boto3).

   c. Di bawah peran IAM ARN, pilih ARN peran IAM yang ingin Anda lampirkan ke grup fitur ini. Untuk informasi selengkapnya tentang cara menemukan peran dan kebijakan terlampir, lihatMenambahkan kebijakan ke peran IAM Anda.

   d. Jika Anda memilih untuk mengaktifkan format Tabel penyimpanan offline dan format Tabel AWS Glue (default), di bawah Katalog data, Anda dapat memilih salah satu dari dua opsi berikut:

      - Gunakan nilai default untuk Anda AWS Glue Data Catalog.
      - Berikan nama Katalog Data yang ada, nama tabel, dan nama database untuk memperluas yang adaAWS Glue Data Catalog.

10. Di bawah kunci enkripsi toko online atau daftar dropdown kunci enkripsi toko offline, pilih salah satu opsi berikut:

    - Gunakan AWS terkelola AWS KMS key (default)
    - Masukkan AWS KMS key ARN dan masukkan ARN kunci Anda di bawah AWS KMS kunci enkripsi toko offline ARN. Untuk informasi selengkapnyaAWS KMS, lihat Layanan Manajemen AWS Utama.

11. Jika berlaku, Anda akan memiliki opsi untuk memilih mode throughput Anda, yang memengaruhi cara Anda dikenai biaya. Di bawah mode Throughput, pilih mode dari daftar dropdown dan masukkan kapasitas baca dan tulis bila tersedia. Untuk informasi tentang mode throughput, seperti ketika mode dapat diterapkan dan unit kapasitas, lihatPengenal pasti.

12. Setelah Anda menentukan semua informasi yang diperlukan, tombol Lanjutkan muncul tersedia. Pilih Lanjutkan.

13. Di bawah Tentukan definisi fitur, Anda memiliki dua opsi untuk menyediakan skema untuk fitur Anda: editor JSON, atau editor tabel.

- Editor JSON: Di tab JSON, masukkan atau salin dan tempel definisi fitur Anda dalam format JSON.
- Editor tabel: Di tab Tabel, masukkan nama fitur fitur dan pilih tipe data yang sesuai untuk setiap fitur di grup fitur Anda. Pilih + Tambahkan definisi fitur untuk menyertakan lebih banyak fitur. Ketahuilah bahwa Anda tidak dapat menghapus definisi fitur dari grup fitur Anda. Namun, Anda dapat menambahkan dan memperbarui definisi fitur setelah grup fitur dibuat.

Setidaknya harus ada dua fitur dalam grup fitur yang mewakili pengenal catatan dan waktu peristiwa:

- Jenis fitur rekaman dapat berupa string, fraksional, atau integral.
- Waktu acara Jenis fitur harus berupa string atau pecahan. Namun, jika Anda memilih format Iceberg tabel, waktu acara harus berupa string.

14. Setelah semua fitur disertakan, pilih Lanjutkan.
15. Di bawah Pilih fitur yang diperlukan, Anda harus menentukan pengenal catatan dan fitur waktu acara. Lakukan ini dengan memilih nama fitur di bawah Record identifier feature name dan Event time feature name dropdown list, masing-masing.
16. Setelah Anda memilih fitur pengenal rekaman dan waktu acara, pilih Lanjutkan.
17. (Opsional) Untuk menambahkan tanda pada grup fitur, pilih Tambahkan tanda baru. Kemudian masukkan kunci tag dan nilai yang sesuai di bawah Kunci dan Nilai, masing-masing.
18. Pilih Lanjutkan.
19. Di bawah Tinjau grup fitur, tinjau informasi grup fitur. Untuk mengedit langkah apa pun, pilih tombol Edit yang sesuai dengan langkah itu. Ini membawa Anda ke langkah yang sesuai untuk mengedit. Untuk kembali ke langkah 5, pilih Lanjutkan sampai Anda kembali ke langkah 5.
20. Setelah Anda menyelesaikan pengaturan untuk grup fitur Anda, pilih Buat grup fitur.

    Jika masalah terjadi selama penyiapan, pesan peringatan pop-up muncul di bagian bawah halaman dengan tips untuk menyelesaikan masalah. Anda dapat kembali ke langkah sebelumnya untuk memperbaiki masalah dengan memilih Edit untuk langkah dengan konflik.

    Setelah grup fitur berhasil dibuat, pesan pop-up hijau muncul di bagian bawah halaman. Grup fitur baru juga muncul di katalog grup fitur Anda.

Lihat detail grup fitur dari konsol

Anda dapat melihat detail grup fitur Anda setelah grup fitur berhasil dibuat di Toko Fitur.

Anda dapat menggunakan konsol atau Amazon SageMaker Feature Store API untuk melihat detail grup fitur Anda. Petunjuk untuk menggunakan Feature Store melalui konsol tergantung pada apakah Anda telah mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Lihat detail grup fitur jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data di panel navigasi kiri, untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. (Opsional) Untuk melihat grup fitur Anda, pilih Akun saya. Untuk melihat grup fitur bersama, pilih Lintas akun.
5. Di bawah tab Katalog grup fitur, pilih nama grup fitur Anda dari daftar. Ini membuka halaman grup fitur.
6. Pada tab Fitur, Anda dapat menemukan daftar semua fitur. Gunakan filter untuk menyempurnakan daftar Anda. Pilih fitur untuk melihat detailnya.
7. Di bawah tab Detail dan subtab Informasi, Anda dapat meninjau informasi grup fitur Anda. Ini termasuk eksekusi terbaru, pengaturan penyimpanan offline, pengaturan penyimpanan online, dan banyak lagi.
8. Di bawah tab Detail dan subtab Tag, Anda dapat meninjau tag grup fitur Anda. Pilih Tambahkan tag baru untuk menambahkan tag baru atau Hapus untuk menghapus tag.
9. Di bawah tab Eksekusi Pipeline, Anda dapat melihat pipeline terkait atau eksekusi pipeline untuk grup fitur Anda.
10. Di bawah tab Lineage, Anda dapat melihat garis keturunan grup fitur Anda.

Memperbarui grup fitur dari konsol

Anda dapat memperbarui grup fitur setelah grup fitur berhasil dibuat di Toko Fitur.

Anda dapat menggunakan konsol atau Amazon SageMaker Feature Store API untuk memperbarui grup fitur. Petunjuk untuk menggunakan Feature Store melalui konsol tergantung pada apakah Anda telah mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Perbarui grup fitur jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data di panel navigasi kiri, untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. (Opsional) Untuk melihat grup fitur Anda, pilih Akun saya. Untuk melihat grup fitur bersama, pilih Lintas akun.
5. Di bawah tab Katalog grup fitur, cari dan pilih nama grup fitur Anda dari daftar. Ini membuka halaman grup fitur.
6. Pilih Perbarui grup fitur.
7. (Opsional) Jika berlaku, Anda dapat mengubah mode throughput Anda, yang memengaruhi cara Anda dikenai biaya. Di bawah mode Throughput, pilih mode dari daftar dropdown dan masukkan kapasitas baca dan tulis bila tersedia. Untuk informasi tentang mode throughput, seperti ketika mode dapat diterapkan dan unit kapasitas, lihatPengenal pasti.
8. (Opsional) Jika grup fitur Anda menggunakan toko online, Anda dapat memperbarui Time to Live (TTL) default. Jika TTL belum diaktifkan untuk grup fitur, alihkan tombol sakelar di bawah Time to Live (TTL) ke On. Anda dapat menentukan nilai TTL dan unit di bawah Durasi Time to Live. Ini akan memperbarui durasi TTL default untuk semua catatan yang ditambahkan ke grup fitur setelah grup fitur diperbarui.
9. (Opsional) Anda dapat menambahkan definisi fitur ke grup fitur Anda tetapi perlu diketahui bahwa Anda tidak dapat menghapus definisi fitur dari grup fitur Anda. Untuk menambahkan definisi fitur, pilih + Tambahkan definisi fitur dan kemudian tentukan nama definisi fitur baru di bawah kolom Nama dan pilih jenis fitur di bawah Tipe fitur kolom.
10. Pilih Simpan perubahan.
11. Untuk mengonfirmasi perubahan, pilih Konfirmasi.

Lihat eksekusi pipeline dari konsol

Anda dapat melihat informasi eksekusi pipeline terbaru untuk fitur atau grup fitur di bawah eksekusi Pipeline. Anda juga bisa mendapatkan tautan ke pipeline, eksekusi, kode, dan informasi eksekusi berguna lainnya.

Anda dapat menggunakan konsol untuk melihat eksekusi pipeline Anda. Petunjuk untuk menggunakan Feature Store melalui konsol tergantung pada apakah Anda telah mengaktifkan

SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Lihat eksekusi pipeline jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data di panel navigasi kiri, untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. (Opsional) Untuk melihat grup fitur Anda, pilih Akun saya. Untuk melihat grup fitur bersama, pilih Lintas akun.
5. Pilih grup fitur atau fitur untuk melihat eksekusi pipeline mereka.
6. Pilih tab Eksekusi Pipeline.
7. Cari pipeline dari daftar dropdown Select a pipeline.
8. Anda dapat melihat tautan untuk rincian pipeline, eksekusi, dan kode. Anda juga dapat melihat pemilik eksekusi, status, tanggal, dan durasi.

Lihat silsilah dari konsol

Anda dapat melihat garis keturunan grup fitur. Silsilah mencakup informasi tentang kode eksekusi alur kerja pemrosesan fitur Anda, sumber data apa yang digunakan, dan bagaimana mereka diserap ke grup fitur atau fitur.

Anda dapat menggunakan konsol untuk melihat garis keturunan grup fitur. Petunjuk tentang penggunaan Feature Store melalui konsol tergantung pada apakah Anda telah mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Lihat silsilah jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data dari panel navigasi kiri untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. (Opsional) Untuk melihat grup fitur Anda, pilih Akun saya. Untuk melihat grup fitur bersama, pilih Lintas akun.
5. Pilih grup fitur atau fitur untuk melihat detail garis keturunannya.

6. Pilih tab Lineage.
7. Pilih grup fitur atau node pipeline untuk memperluas node. Ini berisi informasi selengkapnya tentang grup fitur atau pipeline.
8. Anda dapat memperbesar, memperkecil, atau memasukkan kembali grafik garis keturunan dengan menggunakan tombol di kiri bawah layar.
9. Anda dapat bergerak melalui peta garis keturunan saat Anda memilih dan menyeret layar. Untuk memindahkan peta garis keturunan Anda menggunakan node sebagai titik fokus, Anda dapat menekan Tab atau Shift+Tab untuk beralih antar node.
10. Jika berlaku, Anda dapat menavigasi garis keturunan hulu (kiri, sebelumnya) atau hilir (kanan, terbaru). Lakukan ini dengan memilih node dan kemudian memilih Query upstream lineage atau Query hilir silsilah.

Menghapus grup fitur

Anda dapat menggunakan konsol atau Amazon SageMaker Feature Store API untuk menghapus grup fitur Anda. Petunjuk tentang penggunaan Feature Store melalui konsol bergantung pada apakah Anda telah mengaktifkan Studio atau Studio Classic sebagai pengalaman default Anda. Untuk informasi selengkapnya tentang perbedaan antara keduanya, atau cara mengubah default Anda, lihat SageMaker Studio Amazon.

Bagian berikut memberikan ikhtisar tentang cara menghapus grup fitur.

Topik

- Hapus grup fitur menggunakan konsol
- Hapus grup fitur contoh kode Python

Hapus grup fitur menggunakan konsol

Bagian ini menunjukkan dua cara untuk menghapus grup fitur di konsol, tergantung pada pengalaman default Anda: Studio atau Studio Classic.

Hapus grup fitur jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio Classic.
2. Pilih Data di panel navigasi kiri untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.

4. (Opsional) Untuk melihat grup fitur Anda, pilih Akun saya. Untuk melihat grup fitur bersama, pilih Lintas akun.
5. Di tab Katalog Grup Fitur, pilih grup fitur yang akan dihapus di bawah Nama grup fitur.
6. Pilih Hapus grup fitur.
7. Di jendela pop-up, konfirmasikan penghapusan dengan memasukkan **`delete`** di bidang, lalu pilih Hapus.

Hapus grup fitur contoh kode Python

Kode berikut menggunakan operasi `DeleteFeatureGroup`API untuk menghapus grup fitur Anda menggunakanAWS SDK for Python (Boto3). Ini mengasumsikan bahwa Anda telah menyiapkan Feature Store dan membuat grup fitur. Untuk informasi selengkapnya tentang memulai, lihatPengantar buku catatan contoh Toko Fitur.

```
import sagemaker
from sagemaker.feature_store.feature_group import FeatureGroup

sagemaker_session = sagemaker.Session()
fg_name = 'your-feature-group-name'

my_fg = FeatureGroup(name=fg_name, sagemaker_session=sagemaker_session)
my_fg.delete()
```

Sumber data dan konsumsi

Rekaman ditambahkan ke grup fitur Anda melalui konsumsi. Bergantung pada kasus penggunaan yang Anda inginkan, catatan yang tertelan dapat disimpan dalam grup fitur atau tidak. Ini tergantung pada konfigurasi penyimpanan, jika grup fitur Anda menggunakan toko offline atau online. Toko offline digunakan sebagai database historis, yang biasanya digunakan untuk eksplorasi data, pelatihan model pembelajaran mesin (ML), dan inferensi batch. Toko online digunakan sebagai pencarian catatan waktu nyata, yang biasanya digunakan untuk penyajian model ML. Untuk informasi lebih lanjut tentang konsep dan konsumsi Toko Fitur, lihat. Konsep Feature Store

Ada beberapa cara untuk membawa data Anda ke Amazon SageMaker Feature Store. Feature Store menawarkan panggilan API tunggal untuk konsumsi data `PutRecord` yang disebut yang memungkinkan Anda untuk menelan data dalam batch atau dari sumber streaming. Anda dapat menggunakan Amazon SageMaker Data Wrangler untuk merekayasa fitur dan kemudian

memasukkan fitur Anda ke dalam Toko Fitur Anda. Anda juga dapat menggunakan Amazon EMR untuk konsumsi data batch melalui konektor Spark.

Dalam topik berikut kita akan membahas perbedaan antara

Topik

Penyerapan data

Anda dapat menggunakan sumber streaming seperti Kafka atau Kinesis sebagai sumber data, tempat catatan diekstraksi, dan langsung memasukkan catatan ke toko online untuk pelatihan, inferensi, atau pembuatan fitur. Rekaman dapat dimasukkan ke dalam grup fitur Anda dengan menggunakan panggilan `PutRecord` API sinkron. Karena ini adalah panggilan API sinkron, ini memungkinkan sejumlah kecil pembaruan didorong dalam satu panggilan API. Ini memungkinkan Anda untuk mempertahankan kesegaran nilai fitur yang tinggi dan mempublikasikan nilai segera setelah pembaruan terdeteksi. Ini juga disebut fitur streaming.

Data Wrangler dengan Toko Fitur

Data Wrangler adalah fitur Studio Classic yang menyediakan end-to-end solusi untuk mengimpor, menyiapkan, mengubah, membuat fitur, dan menganalisis data. Data Wrangler memungkinkan Anda untuk merekayasa fitur Anda dan mencernanya ke dalam grup fitur toko online atau offline Anda.

Petunjuk berikut mengekspor buku catatan Jupyter yang berisi semua kode sumber yang diperlukan untuk membuat grup fitur Toko Fitur yang menambahkan fitur Anda dari Data Wrangler ke toko online atau offline.

Petunjuk tentang mengekspor aliran data Wrangler Data Anda ke Feature Store di konsol bervariasi tergantung pada apakah Anda mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Ekspor aliran data Data Wrangler Anda ke Feature Store jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data dari panel kiri, untuk memperluas daftar dropdown.

3. Dari daftar dropdown, pilih Data Wrangler.
4. Jika Anda memiliki instance Amazon SageMaker Canvas yang sudah berjalan, pilih Open Canvas.

   Jika Anda tidak memiliki instance SageMaker Canvas yang berjalan, pilih Jalankan di Canvas.
5. Pada konsol SageMaker Canvas, pilih Data Wrangler di panel navigasi sebelah kiri.
6. Pilih Alur data untuk melihat aliran data Anda.
7. Pilih + untuk memperluas daftar dropdown.
8. Pilih Ekspor aliran data untuk memperluas daftar dropdown.
9. Pilih Simpan ke Toko SageMaker Fitur (melalui JupyterLab Notebook).
10. Di bawah Mengekspor alur data sebagai buku catatan, pilih salah satu opsi berikut:

    - Unduh salinan lokal untuk mengunduh aliran data ke mesin lokal Anda.
    - Ekspor ke lokasi S3 untuk mengunduh aliran data ke lokasi Layanan Penyimpanan Sederhana Amazon dan masukkan lokasi Amazon S3 atau pilih Jelajahi untuk menemukan lokasi Amazon S3 Anda.
11. Pilih Ekspor.

Setelah grup fitur dibuat, Anda juga dapat memilih dan menggabungkan data di beberapa grup fitur untuk membuat fitur rekayasa baru di Data Wrangler dan kemudian mengekspor kumpulan data Anda ke bucket Amazon S3.

Untuk informasi selengkapnya tentang cara mengekspor ke Toko Fitur, lihat Mengekspor ke Toko SageMaker Fitur.

Konsumsi batch dengan Amazon SageMaker Feature Store Spark

Amazon SageMaker Feature Store Spark adalah konektor Spark yang menghubungkan perpustakaan Spark ke Feature Store. Feature Store Spark menyederhanakan konsumsi data dari Spark `DataFrame` ke grup fitur. Feature Store mendukung penyerapan data batch dengan Spark, menggunakan pipeline ETL yang ada, di Amazon EMR, GIS, pekerjaan, pekerjaan Amazon SageMaker ProcessingAWS Glue, atau notebook. SageMaker

Metode untuk menginstal dan mengimplementasikan konsumsi data batch disediakan untuk pengembang Python dan Scala. Pengembang Python dapat menggunakan pustaka `sagemaker-feature-store-pyspark` Python open-source untuk pengembangan lokal, instalasi di Amazon EMR, dan untuk Notebook Jupyter dengan mengikuti petunjuk di repositori Amazon Feature Store

Spark. SageMaker GitHub Pengembang scala dapat menggunakan konektor Feature Store Spark yang tersedia di repositori pusat Amazon SageMaker Feature Store Spark SDK Maven.

Anda dapat menggunakan konektor Spark untuk menyerap data dengan cara berikut, tergantung pada apakah toko online, toko offline, atau keduanya diaktifkan.

1. Ingest secara default - Jika toko online diaktifkan, konektor Spark pertama-tama menyerap kerangka data Anda ke toko online menggunakan API. PutRecord Hanya catatan dengan waktu acara terbesar yang tersisa di toko online. Jika toko offline diaktifkan, dalam waktu 15 menit Toko Fitur menyerap kerangka data Anda ke toko offline. Untuk informasi lebih lanjut tentang cara kerja toko online dan offline, lihatKonsep Feature Store.

   Anda dapat mencapai ini dengan tidak menentukan `target_stores` dalam metode. `.ingest_data(...)`
2. Konsumsi langsung toko offline - Jika toko offline diaktifkan, batch konektor Spark menyerap kerangka data Anda langsung ke toko offline. Menelan kerangka data langsung ke toko offline tidak memperbarui toko online.

   Anda dapat mencapai ini dengan mengatur `target_stores=["OfflineStore"]` `.ingest_data(...)` metode.
3. Hanya toko online - Jika toko online diaktifkan, konektor Spark menyerap kerangka data Anda ke toko online menggunakan API. PutRecord Menelan kerangka data langsung ke toko online tidak memperbarui toko offline.

   Anda dapat mencapai ini dengan mengatur `target_stores=["OnlineStore"]` `.ingest_data(...)` metode.

Untuk informasi tentang menggunakan metode konsumsi yang berbeda, lihat. Contoh implementasi

Topik

- Instalasi Spark Store
- Mengambil JAR untuk Feature Store Spark
- Contoh implementasi

Instalasi Spark Store

Pengguna scala

Feature Store Spark SDK tersedia di repositori pusat Amazon SageMaker Feature Store Spark SDK Maven untuk pengguna Scala.

Persyaratan

- Spark >= 3.0.0 dan <= 3.3.0
- `iceberg-spark-runtime`>= 0.14.0
- Skala >= 2.12.x
- Amazon EMR >= 6.1.0 (hanya jika Anda menggunakan Amazon EMR)

Deklarasikan ketergantungan di POM.xml

Konektor Feature Store Spark memiliki ketergantungan pada perpustakaan. `iceberg-spark-runtime` Oleh karena itu, Anda harus menambahkan versi `iceberg-spark-runtime` pustaka yang sesuai ke dependensi jika Anda memasukkan data ke dalam grup fitur yang telah Anda buat secara otomatis dengan format tabel Iceberg. Misalnya, jika Anda menggunakan Spark 3.1, Anda harus mendeklarasikan hal berikut di proyek Anda: `POM.xml`

```
 <dependency>
 <groupId>software.amazon.sagemaker.featurestore</groupId>
 <artifactId>sagemaker-feature-store-spark-sdk_2.12</artifactId>
 <version>1.0.0</version>
 </dependency>

 <dependency>
   <groupId>org.apache.iceberg</groupId>
   <artifactId>iceberg-spark-runtime-3.1_2.12</artifactId>
   <version>0.14.0</version>
</dependency>
```

Pengguna Python

Feature Store Spark SDK tersedia di repositori SageMaker Amazon Feature Store GitHub Spark sumber terbuka.

Persyaratan

- Spark >= 3.0.0 dan <= 3.3.0

- Amazon EMR >= 6.1.0 (hanya jika Anda menggunakan Amazon EMR)
- Kernel = `conda_python3`

Kami merekomendasikan pengaturan `$SPARK_HOME` ke direktori tempat Anda menginstal Spark. Selama instalasi, Feature Store mengunggah JAR yang diperlukan ke`SPARK_HOME`, sehingga dependensi dimuat secara otomatis. Spark memulai JVM diperlukan untuk membuat perpustakaan ini PySpark berfungsi.

Instalasi lokal

Untuk menemukan info lebih lanjut tentang instalasi, aktifkan mode verbose dengan menambahkan `--verbose` ke perintah instalasi berikut.

```
pip3 install sagemaker-feature-store-pyspark-3.1 --no-binary :all:
```

Instalasi di Amazon EMR

Buat klaster Amazon EMR dengan rilis versi 6.1.0 atau kemudian. Aktifkan SSH untuk membantu Anda memecahkan masalah apa pun.

Anda dapat melakukan salah satu hal berikut untuk menginstal perpustakaan:

- Buat langkah khusus dalam Amazon EMR.
- Connect ke klaster Anda menggunakan SSH dan instal perpustakaan dari sana.

> **Note**
>
> Informasi berikut menggunakan Spark versi 3.1, tetapi Anda dapat menentukan versi apa pun yang memenuhi persyaratan.

```
export SPARK_HOME=/usr/lib/spark
sudo -E pip3 install sagemaker-feature-store-pyspark-3.1 --no-binary :all: --verbose
```

> **Note**
>
> Jika Anda ingin menginstal JAR dependen secara otomatis ke SPARK_HOME, jangan gunakan langkah bootstrap.

Instalasi pada instance SageMaker notebook

Instal versi PySpark yang kompatibel dengan konektor Spark menggunakan perintah berikut:

```
!pip3 install pyspark==3.1.1
!pip3 install sagemaker-feature-store-pyspark-3.1 --no-binary :all:
```

Jika Anda melakukan batch ingestion ke toko offline, dependensi tidak berada dalam lingkungan instance notebook.

```
from pyspark.sql import SparkSession
import feature_store_pyspark

extra_jars = ",".join(feature_store_pyspark.classpath_jars())

spark = SparkSession.builder \
    .config("spark.jars", extra_jars) \
    .config("spark.jars.packages", "org.apache.hadoop:hadoop-
aws:3.2.1,org.apache.hadoop:hadoop-common:3.2.1") \
    .getOrCreate()
```

Instalasi pada notebook dengan GIS

> **Important**
>
> Pengguna harus menggunakan AWS Glue versi 2.0 atau lebih baru.

Gunakan informasi berikut ini untuk membantu Anda menginstal PySpark konektor dalam Sesi AWS Glue Interaktif (GIS).

Amazon SageMaker Feature Store Spark memerlukan JAR konektor Spark tertentu selama inisialisasi sesi untuk diunggah ke bucket Amazon S3 Anda. Untuk informasi lebih lanjut tentang

mengunggah JAR yang diperlukan ke bucket S3 Anda, lihat. Mengambil JAR untuk Feature Store Spark

Setelah Anda mengunggah JAR, Anda harus menyediakan sesi GIS dengan JAR menggunakan perintah berikut.

```
%extra_jars s3://<YOUR_BUCKET>/spark-connector-jars/sagemaker-feature-store-spark-
sdk.jar
```

Untuk menginstal Feature Store Spark di AWS Glue runtime, gunakan perintah `%additional_python_modules` ajaib di dalam notebook GIS. AWS Glueberjalan `pip` ke modul yang telah Anda tentukan di bawah`%additional_python_modules`.

```
%additional_python_modules sagemaker-feature-store-pyspark-3.1
```

Sebelum memulai AWS Glue sesi, Anda harus menggunakan kedua perintah sihir sebelumnya.

Instalasi pada suatu AWS Glue pekerjaan

> **Important**
>
> Pengguna harus menggunakan AWS Glue versi 2.0 atau lebih baru.

Untuk menginstal konektor Spark pada AWS Glue pekerjaan, gunakan `--extra-jars` argumen untuk menyediakan JAR yang diperlukan dan `--additional-python-modules` untuk menginstal Spark Connector sebagai parameter pekerjaan ketika Anda membuat AWS Glue pekerjaan seperti yang ditunjukkan pada contoh berikut. Untuk informasi lebih lanjut tentang mengunggah JAR yang diperlukan ke bucket S3 Anda, lihat. Mengambil JAR untuk Feature Store Spark

```
glue_client = boto3.client('glue', region_name=region)
response = glue_client.create_job(
    Name=pipeline_id,
    Description='Feature Store Compute Job',
    Role=glue_role_arn,
    ExecutionProperty={'MaxConcurrentRuns': max_concurrent_run},
    Command={
        'Name': 'glueetl',
        'ScriptLocation': script_location_uri,
        'PythonVersion': '3'
    },
```

```
    DefaultArguments={
        '--TempDir': temp_dir_location_uri,
        '--additional-python-modules': 'sagemaker-feature-store-pyspark-3.1',
        '--extra-jars': "s3:/<YOUR_BUCKET>/spark-connector-jars/sagemaker-feature-
store-spark-sdk.jar",
        ...
    },
    MaxRetries=3,
    NumberOfWorkers=149,
    Timeout=2880,
    GlueVersion='3.0',
    WorkerType='G.2X'
)
```

Instalasi pada pekerjaan SageMaker Pemrosesan Amazon

Untuk menggunakan pekerjaan Feature Store Spark dengan Amazon SageMaker Processing, bawalah gambar Anda sendiri. Untuk informasi selengkapnya tentang membawa gambar Anda, lihatBawa SageMaker gambar Anda sendiri. Tambahkan langkah instalasi ke Dockerfile. Setelah mendorong image Docker ke repositori Amazon ECR, Anda dapat menggunakannya PySparkProcessor untuk membuat pekerjaan pemrosesan. Untuk informasi selengkapnya tentang membuat pekerjaan pemrosesan dengan PySpark prosesor, lihatPemrosesan Data dengan Apache Spark.

Berikut ini adalah contoh menambahkan langkah instalasi ke Dockerfile.

```
FROM <ACCOUNT_ID>.dkr.ecr.<AWS_REGION>.amazonaws.com/sagemaker-spark-processing:3.1-
cpu-py38-v1.0

RUN /usr/bin/python3 -m pip install sagemaker-feature-store-pyspark-3.1 --no-
binary :all: --verbose
```

Mengambil JAR untuk Feature Store Spark

Untuk mengambil JAR ketergantungan Feature Store Spark, Anda harus menginstal konektor Spark dari repositori Python Package Index (PyPI) menggunakan di lingkungan Python apa pun dengan akses jaringan. `pip` Notebook SageMaker Jupyter adalah contoh lingkungan Python dengan akses jaringan.

Perintah berikut menginstal konektor Spark.

```
!pip install sagemaker-feature-store-pyspark-3.1
```

Setelah menginstal Feature Store Spark, Anda dapat mengambil lokasi JAR dan mengunggah JAR ke Amazon S3.

`feature-store-pyspark-dependency-jars`Perintah menyediakan lokasi JAR ketergantungan yang diperlukan yang ditambahkan oleh Feature Store Spark. Anda dapat menggunakan perintah untuk mengambil JAR dan unggah ke Amazon S3.

```
jar_location = !feature-store-pyspark-dependency-jars
jar_location = jar_location[0]

s3_client = boto3.client("s3")
s3_client.upload_file(jar_location, "<YOUR_BUCKET>","spark-connector-jars/sagemaker-
feature-store-spark-sdk.jar")
```

Contoh implementasi

Example Python script

FeatureStoreBatchIngestion.py

```
from pyspark.sql import SparkSession
from feature_store_pyspark.FeatureStoreManager import FeatureStoreManager
import feature_store_pyspark

spark = SparkSession.builder \
                    .getOrCreate()

# Construct test DataFrame
columns = ["RecordIdentifier", "EventTime"]
data = [("1","2021-03-02T12:20:12Z"), ("2", "2021-03-02T12:20:13Z"), ("3",
 "2021-03-02T12:20:14Z")]

df = spark.createDataFrame(data).toDF(*columns)
```

```
# Initialize FeatureStoreManager with a role arn if your feature group is created by
 another account
feature_store_manager= FeatureStoreManager("arn:aws:iam::111122223333:role/role-
arn")

# Load the feature definitions from input schema. The feature definitions can be
 used to create a feature group
feature_definitions = feature_store_manager.load_feature_definitions_from_schema(df)

feature_group_arn = "arn:aws:sagemaker:<AWS_REGION>:<ACCOUNT_ID>:feature-
group/<YOUR_FEATURE_GROUP_NAME>"

# Ingest by default. The connector will leverage PutRecord API to ingest your data
 in stream
# https://docs.aws.amazon.com/sagemaker/latest/APIReference/
API_feature_store_PutRecord.html
feature_store_manager.ingest_data(input_data_frame=df,
 feature_group_arn=feature_group_arn)

# To select the target stores for ingestion, you can specify the target store as the
 paramter
# If OnlineStore is selected, the connector will leverage PutRecord API to ingest
 your data in stream
feature_store_manager.ingest_data(input_data_frame=df,
 feature_group_arn=feature_group_arn, target_stores=["OfflineStore", "OnlineStore"])

# If only OfflineStore is selected, the connector will batch write the data to
 offline store directly
feature_store_manager.ingest_data(input_data_frame=df,
 feature_group_arn=feature_group_arn, target_stores=["OfflineStore"])

# To retrieve the records failed to be ingested by spark connector
failed_records_df = feature_store_manager.get_failed_stream_ingestion_data_frame()
```

Kirim pekerjaan Spark dengan contoh skrip Python

PySpark Versi ini membutuhkan JAR ekstra dependen untuk diimpor, jadi langkah tambahan diperlukan untuk menjalankan aplikasi Spark.

Jika Anda tidak menentukan `SPARK_HOME` selama instalasi, maka Anda harus memuat JAR yang diperlukan di JVM saat berjalan. `spark-submit feature-store-pyspark-dependency-`

`jars`adalah skrip Python yang diinstal oleh perpustakaan Spark untuk secara otomatis mengambil jalur ke semua JAR untuk Anda.

```
spark-submit --jars `feature-store-pyspark-dependency-
jars` FeatureStoreBatchIngestion.py
```

Jika Anda menjalankan aplikasi ini di Amazon EMR, kami sarankan Anda menjalankan aplikasi dalam mode klien, sehingga Anda tidak perlu mendistribusikan JAR dependen ke node tugas lain. Tambahkan satu langkah lagi di cluster EMR Amazon dengan argumen Spark yang mirip dengan berikut ini:

```
spark-submit --deploy-mode client --master yarn s3:/<PATH_TO_SCRIPT>/
FeatureStoreBatchIngestion.py
```

Example Scala script

FeatureStoreBatchIngestion.scala

```
import software.amazon.sagemaker.featurestore.sparksdk.FeatureStoreManager
import org.apache.spark.sql.types.{StringType, StructField, StructType}
import org.apache.spark.sql.{Row, SparkSession}

object TestSparkApp {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder().getOrCreate()

    // Construct test DataFrame
    val data = List(
      Row("1", "2021-07-01T12:20:12Z"),
      Row("2", "2021-07-02T12:20:13Z"),
      Row("3", "2021-07-03T12:20:14Z")
    )

    val schema = StructType(
      List(StructField("RecordIdentifier", StringType), StructField("EventTime",
 StringType))
```

```
    )

    val df = spark.createDataFrame(spark.sparkContext.parallelize(data), schema)

    // Initialize FeatureStoreManager with a role arn if your feature group is
 created by another account
    val featureStoreManager = new
 FeatureStoreManager("arn:aws:iam::111122223333:role/role-arn")

    // Load the feature definitions from input schema. The feature definitions can
 be used to create a feature group
    val featureDefinitions =
 featureStoreManager.loadFeatureDefinitionsFromSchema(df)

    val featureGroupArn = "arn:aws:sagemaker:<AWS_REGION>:<ACCOUNT_ID>:feature-
group/<YOUR_FEATURE_GROUP_NAME>"

    // Ingest by default. The connector will leverage PutRecord API to ingest your
 data in stream
    // https://docs.aws.amazon.com/sagemaker/latest/APIReference/
API_feature_store_PutRecord.html
    featureStoreManager.ingestData(df, featureGroupArn)

    // To select the target stores for ingestion, you can specify the target store
 as the paramter
    // If OnlineStore is selected, the connector will leverage PutRecord API to
 ingest your data in stream
    featureStoreManager.ingestData(df, featureGroupArn, List("OfflineStore",
 "OnlineStore"))

    // If only OfflineStore is selected, the connector will batch write the data to
 offline store directly
    featureStoreManager.ingestData(df, featureGroupArn, ["OfflineStore"])

    // To retrieve the records failed to be ingested by spark connector
    val failedRecordsDf = featureStoreManager.getFailedStreamIngestionDataFrame()
  }
}
```

Kirim pekerjaan Spark

Scala

Anda harus dapat menggunakan Feature Store Spark sebagai dependensi normal. Tidak diperlukan instruksi tambahan untuk menjalankan aplikasi di semua platform.

Fungsi JSON

Amazon SageMaker Feature Store Feature Processing adalah kemampuan yang dapat digunakan untuk mengubah data mentah menjadi fitur machine learning (ML). Ini memberi Anda SDK Prosesor Fitur yang dengannya Anda dapat mengubah dan menyerap data dari sumber data batch ke dalam grup fitur Anda. Dengan kemampuan ini, Feature Store menangani infrastruktur yang mendasarinya termasuk menyediakan lingkungan komputasi dan membuat serta memelihara SageMaker Pipelines untuk memuat dan menyerap data. Dengan cara ini Anda dapat fokus pada definisi prosesor fitur Anda yang mencakup fungsi transformasi (misalnya, jumlah tampilan produk, rata-rata nilai transaksi), sumber (tempat menerapkan transformasi ini), dan sink (tempat menulis nilai fitur yang dihitung ke).

Fitur Pipa prosesor adalah SageMaker pipa Pipelines. Sebagai SageMaker Pipelines, Anda juga dapat melacak pipeline Prosesor Fitur terjadwal dengan SageMaker garis keturunan di konsol. Untuk informasi selengkapnya tentang SageMaker Lineage, lihat Pelacakan SageMaker Silsilah Amazon Ini termasuk melacak eksekusi terjadwal, memvisualisasikan garis keturunan untuk melacak fitur kembali ke sumber datanya, dan melihat pemroses fitur bersama dalam satu lingkungan. Untuk informasi tentang penggunaan Feature Store dengan konsol, lihatLihat eksekusi pipeline dari konsol.

Topik

- SDK Prosesor Fitur Toko Fitur
- Menjalankan Prosesor Fitur Toko Fitur dari jarak jauh
- Membuat dan menjalankan saluran pipa Prosesor Fitur Toko Fitur
- Eksekusi terjadwal dan berbasis acara untuk pipeline Prosesor Fitur
- Pantau SageMaker Pipeline Prosesor Fitur Fitur Amazon Feature Store
- Izin IAM dan peran eksekusi
- Fitur Pembatasan, batas, dan kuota prosesor
- Sumber data
- Contoh kode Pemrosesan Fitur untuk kasus penggunaan umum

SDK Prosesor Fitur Toko Fitur

Deklarasikan definisi Prosesor Fitur Toko Fitur dengan mendekorasi fungsi transformasi Anda dengan dekorator. `@feature_processor` SageMaker SDK for Python (Boto3) SDK for Python (Boto3) secara otomatis memuat data dari sumber data input yang dikonfigurasi, menerapkan fungsi transformasi yang didekorasi, dan kemudian menyerap data yang diubah ke grup fitur target. Fungsi transformasi yang didekorasi harus sesuai dengan tanda tangan yang diharapkan dari `@feature_processor` dekorator. Untuk informasi selengkapnya tentang `@feature_processor` dekorator, lihat @feature_processor Decorator di Amazon SageMaker Feature Store Read the Docs.

Dengan `@feature_processor` dekorator, fungsi transformasi Anda berjalan di lingkungan runtime Spark di mana argumen input yang diberikan ke fungsi Anda dan nilai pengembaliannya adalah Spark. DataFrames Jumlah parameter input dalam fungsi transformasi Anda harus sesuai dengan jumlah input yang dikonfigurasi di `@feature_processor` dekorator.

Untuk informasi selengkapnya tentang `@feature_processor` dekorator, lihat Feature Processor Feature Store SDK for Python (Boto3).

Kode berikut adalah contoh dasar tentang cara menggunakan `@feature_processor` dekorator. Untuk contoh kasus penggunaan yang lebih spesifik, lihatContoh kode Pemrosesan Fitur untuk kasus penggunaan umum.

Feature Processor SDK dapat diinstal dari SageMaker Python SDK dan tambahannya menggunakan perintah berikut.

```
pip install sagemaker[feature-processor]
```

Dalam contoh berikut, *`us-east-1`* adalah wilayah sumber daya, *`111122223333`* adalah ID akun pemilik sumber daya, dan *`your-feature-group-name`* merupakan nama grup fitur.

Berikut ini adalah definisi prosesor fitur dasar, di mana `@feature_processor` dekorator mengonfigurasi input CSV dari Amazon S3 untuk dimuat dan disediakan ke fungsi transformasi Anda (misalnya,`transform`), dan menyiapkannya untuk dikonsumsi ke grup fitur. Baris terakhir menjalankannya.

```
from sagemaker.feature_store.feature_processor import CSVDataSource, feature_processor

CSV_DATA_SOURCE = CSVDataSource('s3://your-bucket/prefix-to-csv/')
OUTPUT_FG = 'arn:aws:sagemaker:us-east-1:111122223333:feature-group/your-feature-group-
name'
```

```
@feature_processor(inputs=[CSV_DATA_SOURCE], output=OUTPUT_FG)
def transform(csv_input_df):
   return csv_input_df

transform()
```

`@feature_processor`Parameternya meliputi:

- `inputs`(List [str]): Daftar sumber data yang digunakan dalam Prosesor Fitur Toko Fitur Anda. Jika sumber data Anda adalah grup fitur atau disimpan di Amazon S3, Anda mungkin dapat menggunakan definisi sumber data yang disediakan Toko Fitur untuk pemroses fitur. Untuk daftar lengkap definisi sumber data yang disediakan Toko Fitur, lihat Sumber Data Prosesor Fitur di Toko SageMaker Fitur Amazon Membaca Dokumen.
- `output`(str): ARN dari grup fitur untuk menelan output dari fungsi yang didekorasi.
- `target_stores`(Opsional [List [str]]): Daftar toko (misalnya, `OnlineStore` atau`OfflineStore`) untuk dicerna ke output. Jika tidak ditentukan, data akan dicerna ke semua penyimpanan yang diaktifkan grup fitur keluaran.
- `parameters`(Dict [str, Any]): Kamus yang akan disediakan untuk fungsi transformasi Anda.
- `enable_ingestion`(bool): Bendera untuk menunjukkan apakah output fungsi transformasi dicerna ke grup fitur keluaran. Bendera ini berguna selama fase pengembangan. Jika tidak ditentukan, konsumsi diaktifkan.

Parameter fungsi dibungkus opsional (disediakan sebagai argumen jika disediakan dalam tanda tangan fungsi) meliputi:

- `params`(Dict [str, Any]): Kamus didefinisikan dalam parameter. `@feature_processor` Ini juga berisi parameter yang dikonfigurasi sistem yang dapat direferensikan dengan kuncisystem, seperti `scheduled_time` parameter.
- `spark`(SparkSession): Referensi ke SparkSession instance yang diinisialisasi untuk Aplikasi Spark.

Kode berikut adalah contoh penggunaan `spark` parameter `params` dan.

```
from sagemaker.feature_store.feature_processor import CSVDataSource, feature_processor

CSV_DATA_SOURCE = CSVDataSource('s3://your-bucket/prefix-to-csv/')
```

```
OUTPUT_FG = 'arn:aws:sagemaker:us-east-1:111122223333:feature-group/your-feature-group-name'

@feature_processor(inputs=[CSV_DATA_SOURCE], output=OUTPUT_FG)
def transform(csv_input_df, params, spark):

   scheduled_time = params['system']['scheduled_time']
   csv_input_df.createOrReplaceTempView('csv_input_df')
   return spark.sql(f'''
        SELECT *
        FROM csv_input_df
        WHERE date_add(event_time, 1) >= {scheduled_time}
   ''')

transform()
```

Parameter `scheduled_time` sistem (disediakan dalam `params` argumen untuk fungsi Anda) adalah nilai penting untuk mendukung percobaan ulang setiap eksekusi. Nilai dapat membantu mengidentifikasi eksekusi Prosesor Fitur secara unik dan dapat digunakan sebagai titik referensi untuk input berbasis daterange (misalnya, hanya memuat data 24 jam terakhir) untuk menjamin rentang input yang independen dari waktu eksekusi aktual kode. Jika Prosesor Fitur berjalan sesuai jadwal (lihatEksekusi terjadwal dan berbasis acara untuk pipeline Prosesor Fitur) maka nilainya ditetapkan ke waktu yang dijadwalkan untuk dijalankan. Argumen dapat diganti selama eksekusi sinkron menggunakan API eksekusi SDK untuk mendukung kasus penggunaan seperti pengisian ulang data atau menjalankan kembali eksekusi masa lalu yang tidak terjawab. Nilainya adalah waktu saat ini jika Prosesor Fitur berjalan dengan cara lain.

Untuk informasi tentang penulisan kode Spark, lihat Panduan Pemrograman SQL Spark.

Untuk contoh kode lainnya untuk kasus penggunaan umum, lihat. Contoh kode Pemrosesan Fitur untuk kasus penggunaan umum

Perhatikan bahwa fungsi transformasi yang didekorasi dengan `@feature_processor` tidak mengembalikan nilai. Untuk menguji fungsi Anda secara terprogram, Anda dapat menghapus atau menambal `@feature_processor` dekorator sedemikian rupa sehingga berfungsi sebagai pass-through ke fungsi yang dibungkus. Untuk detail selengkapnya tentang `@feature_processor` dekorator, lihat Amazon SageMaker Feature Store Python SDK.

Menjalankan Prosesor Fitur Toko Fitur dari jarak jauh

Untuk menjalankan Prosesor Fitur Anda pada kumpulan data besar yang membutuhkan perangkat keras yang lebih kuat daripada yang tersedia secara lokal, Anda dapat menghias kode Anda dengan `@remote` dekorator untuk menjalankan kode Python lokal Anda sebagai pekerjaan pelatihan terdistribusi tunggal atau multi-node. SageMaker Untuk informasi selengkapnya tentang menjalankan kode Anda sebagai pekerjaan SageMaker pelatihan, lihatJalankan kode lokal Anda sebagai pekerjaan SageMaker pelatihan.

Berikut ini adalah contoh penggunaan `@remote` dekorator bersama dengan `@feature_processor` dekorator.

```
from sagemaker.remote_function.spark_config import SparkConfig
from sagemaker.remote_function import remote
from sagemaker.feature_store.feature_processor import CSVDataSource, feature_processor

CSV_DATA_SOURCE = CSVDataSource('s3://bucket/prefix-to-csv/')
OUTPUT_FG = 'arn:aws:sagemaker:us-east-1:123456789012:feature-group/feature-group'

@remote(
    spark_config=SparkConfig(),
    instance_type="ml.m5.2xlarge",
    dependencies="/local/requirements.txt"
)
@feature_processor(
    inputs=[CSV_DATA_SOURCE],
    output=OUTPUT_FG,
)
def transform(csv_input_df):
   return csv_input_df

transform()
```

`spark_config`Parameter menunjukkan bahwa pekerjaan jarak jauh berjalan sebagai aplikasi Spark. `SparkConfig`Instance ini dapat digunakan untuk mengkonfigurasi Konfigurasi Spark dan memberikan dependensi tambahan ke aplikasi Spark seperti file Python, JAR, dan file.

Untuk iterasi yang lebih cepat saat mengembangkan kode pemrosesan fitur, Anda dapat menentukan `keep_alive_period_in_seconds` argumen di `@remote` dekorator untuk mempertahankan sumber daya yang dikonfigurasi di kolam hangat untuk pekerjaan pelatihan berikutnya. Untuk

informasi selengkapnya tentang kolam hangat, lihat `KeepAlivePeriodInSeconds` di panduan Referensi API.

Kode berikut adalah contoh lokal `requirements.txt`:

```
sagemaker>=2.167.0
```

Ini akan menginstal versi SageMaker SDK yang sesuai dalam pekerjaan jarak jauh yang diperlukan untuk mengeksekusi metode yang dijelaskan oleh. `@feature-processor`

Membuat dan menjalankan saluran pipa Prosesor Fitur Toko Fitur

Feature Processor SDK menyediakan API untuk mempromosikan Definisi Prosesor Fitur Anda ke dalam SageMaker Pipeline yang dikelola sepenuhnya. Untuk informasi lebih lanjut tentang SageMaker Pipelines, lihatSageMaker Ikhtisar Pipelines. Untuk mengonversi Definisi Prosesor Fitur menjadi SageMaker Pipeline, gunakan `to_pipeline` API dengan definisi Prosesor Fitur Anda. Anda dapat menjadwalkan eksekusi Definisi Prosesor Fitur dapat dijadwalkan, memantau secara operasional dengan CloudWatch metrik, dan mengintegrasikannya EventBridge untuk bertindak sebagai sumber acara atau pelanggan. Untuk informasi selengkapnya tentang pemantauan jaringan pipa yang dibuat dengan SageMaker Pipelines, lihat. Pantau SageMaker Pipeline Prosesor Fitur Fitur Amazon Feature Store

Untuk melihat pipeline Prosesor Fitur, lihatLihat eksekusi pipeline dari konsol.

Jika fungsi Anda juga dihiasi dengan `@remote` dekorator, maka konfigurasinya dibawa ke pipa Prosesor Fitur. Anda dapat menentukan konfigurasi lanjutan seperti jenis dan hitungan instans komputasi, dependensi runtime, konfigurasi jaringan dan keamanan menggunakan dekorator. `@remote`

Contoh berikut menggunakan `execute` API `to_pipeline` dan.

```
from sagemaker.feature_store.feature_processor import (
    execute, to_pipeline, describe, TransformationCode
)

pipeline_name="feature-processor-pipeline"
pipeline_arn = to_pipeline(
    pipeline_name=pipeline_name,
    step=transform,
    transformation_code=TransformationCode(s3_uri="s3://bucket/prefix"),
```

```
)

pipeline_execution_arn = execute(
    pipeline_name=pipeline_name
)
```

`to_pipeline`API secara semantik merupakan operasi upsert. Ini memperbarui pipa jika sudah ada; jika tidak, itu membuat pipa.

`to_pipeline`API secara opsional menerima URI Amazon S3 yang mereferensikan file yang berisi definisi Prosesor Fitur untuk mengaitkannya dengan pipeline Prosesor Fitur untuk melacak fungsi transformasi dan versinya dalam garis keturunan pembelajaran SageMaker mesinnya.

Untuk mengambil daftar setiap pipeline Prosesor Fitur di akun Anda, Anda dapat menggunakan `list_pipelines` API. Permintaan berikutnya ke `describe` API menampilkan detail yang terkait dengan pipeline Prosesor Fitur termasuk, namun tidak terbatas pada, SageMaker Pipelines dan detail jadwal.

Contoh berikut menggunakan `describe` API `list_pipelines` dan.

```
from sagemaker.feature_store.feature_processor import list_pipelines, describe

feature_processor_pipelines = list_pipelines()

pipeline_description = describe(
    pipeline_name = feature_processor_pipelines[0]
)
```

Eksekusi terjadwal dan berbasis acara untuk pipeline Prosesor Fitur

Eksekusi pipeline Pemrosesan SageMaker Fitur Amazon Feature Store dapat dikonfigurasi untuk memulai secara otomatis dan asinkron berdasarkan jadwal yang telah dikonfigurasi sebelumnya atau sebagai hasil dari peristiwa layanan lain. AWS Misalnya, Anda dapat menjadwalkan pipeline Pemrosesan Fitur untuk dieksekusi pada bulan pertama setiap bulan atau menghubungkan dua saluran pipa bersama-sama sehingga pipeline target dijalankan secara otomatis setelah eksekusi pipa sumber selesai.

Topik

- Eksekusi berdasarkan jadwal
- Eksekusi berbasis acara

Eksekusi berdasarkan jadwal

Feature Processor SDK menyediakan `schedule`API untuk menjalankan pipeline Prosesor Fitur secara berulang dengan integrasi Amazon EventBridge Scheduler. Jadwal dapat ditentukan dengan`at`,`rate`, atau `cron` ekspresi menggunakan `ScheduleExpression`parameter dengan ekspresi yang sama didukung oleh Amazon EventBridge. API jadwal secara semantik merupakan operasi upsert karena memperbarui jadwal jika sudah ada; jika tidak, itu membuatnya. Untuk informasi selengkapnya tentang EventBridge ekspresi dan contoh, lihat Jenis jadwal pada EventBridge Penjadwal di Panduan Pengguna EventBridge Penjadwal.

Contoh berikut menggunakan `schedule`API Processor Fitur, menggunakan`at`,`rate`, dan `cron` ekspresi.

```
from sagemaker.feature_store.feature_processor import schedule
pipeline_name='feature-processor-pipeline'

event_bridge_schedule_arn = schedule(
    pipeline_name=pipeline_name,
    schedule_expression="at(2020-11-30T00:00:00)"
)

event_bridge_schedule_arn = schedule(
    pipeline_name=pipeline_name,
    schedule_expression="rate(24 hours)"
)

event_bridge_schedule_arn = schedule(
    pipeline_name=pipeline_name,
    schedule_expression="cron(0 0-23/1 ? * * 2023-2024)"
)
```

Zona waktu default untuk input tanggal dan waktu di `schedule` API ada di UTC. Untuk informasi selengkapnya tentang ekspresi jadwal EventBridge Scheduler, lihat `ScheduleExpression`di dokumentasi Referensi API EventBridge Scheduler.

Eksekusi pipeline Prosesor Fitur Terjadwal menyediakan fungsi transformasi Anda dengan waktu eksekusi terjadwal, untuk digunakan sebagai token idempotensi atau titik referensi tetap untuk input berbasis rentang tanggal. Untuk menonaktifkan (yaitu, menjeda) atau mengaktifkan kembali jadwal, gunakan `state` parameter `schedule`API dengan `'DISABLED'` atau`'ENABLED'`, masing-masing.

Untuk informasi tentang Prosesor Fitur, lihatFitur Sumber data SDK Prosesor.

Eksekusi berbasis acara

Pipeline Pemrosesan Fitur dapat dikonfigurasi untuk mengeksekusi secara otomatis ketika suatu AWS peristiwa terjadi. Feature Processing SDK menyediakan `put_trigger`fungsi yang menerima daftar peristiwa sumber dan pipeline target. Peristiwa sumber harus berupa contoh `FeatureProcessorPipelineEvent`, yang menentukan peristiwa status pipeline dan eksekusi.

`put_trigger`Fungsi ini mengonfigurasi EventBridge aturan Amazon dan menargetkan untuk merutekan peristiwa dan memungkinkan Anda menentukan pola EventBridge peristiwa untuk merespons AWS peristiwa apa pun. Untuk informasi tentang konsep ini, lihat EventBridge aturan Amazon, target, dan pola peristiwa.

Pemicu dapat diaktifkan atau dinonaktifkan. EventBridge akan memulai eksekusi pipeline target menggunakan peran yang disediakan dalam `role_arn` parameter `put_trigger` API. Peran eksekusi digunakan secara default jika SDK digunakan di lingkungan Amazon SageMaker Studio Classic atau Notebook. Untuk informasi tentang cara mendapatkan peran eksekusi Anda, lihatDapatkan peran eksekusi.

Contoh berikut mengatur:

- SageMaker Pipeline menggunakan `to_pipeline` API, yang menggunakan nama pipeline target (`target-pipeline`) dan fungsi transformasi Anda (`transform`). Untuk informasi tentang Prosesor Fitur dan fungsi transformasi, lihatFitur Sumber data SDK Prosesor.
- Pemicu menggunakan `put_trigger` API, yang digunakan `FeatureProcessorPipelineEvent` untuk acara dan nama pipeline target Anda (`target-pipeline`).

  `FeatureProcessorPipelineEvent`Mendefinisikan pemicu kapan status source pipeline (`source-pipeline`) Anda menjadi`Succeeded`. Untuk informasi tentang fungsi acara Pipeline Prosesor Fitur, lihat `FeatureProcessorPipelineEvent`di Toko Fitur Baca Dokumen.

```
from sagemaker.feature_store.feature_processor import put_trigger, to_pipeline,
 FeatureProcessorPipelineEvent

to_pipeline(pipeline_name="target-pipeline", step=transform)

put_trigger(
    source_pipeline_events=[
        FeatureProcessorPipelineEvent(
            pipeline_name="source-pipeline",
```

```
            status=["Succeeded"]
        )
    ],
    target_pipeline="target-pipeline"
)
```

Untuk contoh penggunaan pemicu berbasis peristiwa untuk membuat eksekusi berkelanjutan dan percobaan ulang otomatis untuk pipeline Prosesor Fitur Anda, lihat. Eksekusi berkelanjutan dan percobaan ulang otomatis menggunakan pemicu berbasis peristiwa

Untuk contoh menggunakan pemicu berbasis peristiwa untuk membuat streaming berkelanjutan dan percobaan ulang otomatis menggunakan pemicu berbasis peristiwa, lihat. Streaming contoh sumber data kustom

Pantau SageMaker Pipeline Prosesor Fitur Fitur Amazon Feature Store

AWSmenyediakan alat pemantauan untuk mengawasi SageMaker sumber daya Amazon dan aplikasi Anda secara real time, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan. Feature Store Feature Processor SageMaker pipelines adalah Pipelines, sehingga tersedia mekanisme dan integrasi pemantauan standar. Metrik operasional seperti kegagalan eksekusi dapat dipantau melalui CloudWatch metrik Amazon dan peristiwa Amazon. EventBridge

Untuk informasi selengkapnya tentang cara memantau dan mengoperasionalkan Prosesor Fitur Toko Fitur, lihat sumber daya berikut:

- Memantau AWS sumber daya yang disediakan saat menggunakan Amazon SageMaker- Panduan umum tentang kegiatan pemantauan dan audit untuk SageMaker sumber daya.
- SageMaker Metrik Pipelines- CloudWatch Metrik yang dipancarkan oleh Pipelines. SageMaker
- Perubahan status eksekusi pipa- EventBridge peristiwa yang dipancarkan untuk SageMaker Pipelines dan eksekusi.
- Memecahkan Masalah Pipa Pembuatan SageMaker Model Amazon- Kiat debugging dan pemecahan masalah umum untuk Pipelines. SageMaker

Fitur Toko Fitur Log eksekusi prosesor dapat ditemukan di Amazon CloudWatch Logs di bawah grup `/aws/sagemaker/TrainingJobs` log, di mana Anda dapat menemukan aliran log eksekusi menggunakan konvensi pencarian. Untuk eksekusi yang dibuat dengan langsung menjalankan fungsi yang `@feature_processor` didekorasi, Anda dapat menemukan log di konsol lingkungan eksekusi

lokal Anda. Untuk eksekusi yang `@remote` didekorasi, nama aliran CloudWatch Log berisi nama fungsi dan stempel waktu eksekusi. Untuk eksekusi pipeline Prosesor Fitur, aliran CloudWatch Log untuk langkah tersebut berisi `feature-processor` string dan ID eksekusi pipeline.

Saluran pipa Prosesor Fitur Toko Fitur dan status eksekusi terbaru dapat ditemukan di Amazon SageMaker Studio Classic untuk grup fitur tertentu di UI Toko Fitur. Grup fitur yang terkait dengan pipeline Prosesor Fitur sebagai input atau output ditampilkan di UI. Selain itu, tampilan garis keturunan dapat memberikan konteks ke dalam eksekusi hulu, seperti jaringan pipa Prosesor Fitur yang memproduksi data dan sumber data, untuk debugging lebih lanjut. Untuk informasi selengkapnya tentang penggunaan tampilan garis keturunan menggunakan Studio Classic, lihat. Lihat silsilah dari konsol

Izin IAM dan peran eksekusi

Untuk menggunakan Amazon SageMaker Python SDK memerlukan izin untuk berinteraksi. Layanan AWS Kebijakan berikut diperlukan untuk fungsionalitas Prosesor Fitur lengkap. Anda dapat melampirkan AmazonSageMakerFullAccessdan Kebijakan AmazonEventBridgeSchedulerFullAccessAWSTerkelola yang dilampirkan ke peran IAM Anda. Untuk informasi tentang melampirkan kebijakan ke peran IAM Anda, lihat. Menambahkan kebijakan ke peran IAM Anda Lihat contoh-contoh berikut untuk detailnya.

Kebijakan kepercayaan dari peran yang diterapkan kebijakan ini harus memungkinkan prinsip “scheduler.amazonaws.com”, “sagemaker.amazonaws.com”, dan “glue.amazonaws.com”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "scheduler.amazonaws.com",
                    "sagemaker.amazonaws.com",
                    "glue.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
```

```
}
```

Fitur Pembatasan, batas, dan kuota prosesor

Amazon SageMaker Feature Store Feature Processing mengandalkan pelacakan garis keturunan SageMaker machine learning (ML). Prosesor Fitur Toko Fitur menggunakan konteks garis keturunan untuk mewakili dan melacak Pipa Pemrosesan Fitur dan versi Pipa. Setiap Prosesor Fitur Toko Fitur mengkonsumsi setidaknya dua konteks garis keturunan (satu untuk Pipeline Pemrosesan Fitur dan satu lagi untuk versi). Jika sumber data input atau output dari Feature Processing Pipeline berubah, konteks garis keturunan tambahan akan dibuat. Anda dapat memperbarui batas garis keturunan SageMaker ML dengan menjangkau AWS dukungan untuk peningkatan batas. Batas default untuk sumber daya yang digunakan oleh Prosesor Fitur Toko Fitur adalah sebagai berikut. Untuk informasi tentang pelacakan garis keturunan SageMaker ML, lihat. Pelacakan SageMaker Silsilah Amazon

Untuk informasi selengkapnya tentang SageMaker kuota, lihat SageMaker titik akhir dan kuota Amazon.

Batas garis keturunan per Wilayah

- Konteks - 500 (batas lunak)
- Artefak - 6.000 (batas lunak)
- Asosiasi - 6.000 (batas lunak)

Batas Pelatihan per Wilayah

- Waktu aktif terpanjang untuk tugas pelatihan — 432.000 detik
- Jumlah instans maksimal per tugas pelatihan — 20
- Jumlah maksimum `CreateTrainingJob` permintaan yang dapat Anda buat, per detik, di akun ini di Wilayah saat ini — 1 TPS
- Pertahankan periode hidup untuk penggunaan kembali cluster — 3.600 detik

Jumlah maksimum Pipa dan eksekusi pipa bersamaan per Wilayah

- Jumlah maksimum saluran pipa yang diizinkan per akun - 500
- Jumlah maksimum eksekusi pipa bersamaan yang diizinkan per akun — 20
- Waktu di mana waktu eksekusi pipa habis - 672 jam

Sumber data

Amazon SageMaker Feature Store Feature Processing mendukung beberapa sumber data. Feature Processor SDK for Python (Boto3) menyediakan konstruksi untuk memuat data dari grup fitur atau objek yang disimpan di Amazon S3. Selain itu, Anda dapat membuat sumber data khusus untuk memuat data dari sumber data lain. Untuk informasi tentang sumber data yang disediakan Toko Fitur, lihat Sumber data Prosesor Fitur Feature Store Python SDK.

Topik

- Fitur Sumber data SDK Prosesor
- Sumber data kustom
- Contoh sumber data kustom

Fitur Sumber data SDK Prosesor

Amazon SageMaker Feature Store Feature Processor SDK for Python (Boto3) menyediakan konstruksi untuk memuat data dari grup fitur atau objek yang disimpan di Amazon S3. Untuk daftar lengkap definisi sumber data yang disediakan Toko Fitur, lihat Sumber data Prosesor Fitur Feature Store Python SDK.

Untuk contoh tentang cara menggunakan definisi sumber data SDK Python Toko Fitur, lihat. Contoh kode Pemrosesan Fitur untuk kasus penggunaan umum

FeatureGroupDataSource

`FeatureGroupDataSource`Ini digunakan untuk menentukan grup fitur sebagai sumber data input untuk Prosesor Fitur. Data dapat dimuat dari grup fitur toko offline. Mencoba memuat data Anda dari grup fitur toko online akan menghasilkan kesalahan validasi. Anda dapat menentukan offset awal dan akhir untuk membatasi data yang dimuat ke rentang waktu tertentu. Misalnya, Anda dapat menentukan offset awal '14 hari' untuk memuat hanya dua minggu terakhir data, dan Anda juga dapat menentukan offset akhir '7 hari' untuk membatasi input ke data minggu sebelumnya.

Fitur Store menyediakan definisi sumber data

Feature Store Python SDK berisi definisi sumber data yang dapat digunakan untuk menentukan berbagai sumber data input untuk Prosesor Fitur. Ini termasuk sumber tabel CSV, Parket, dan Gunung Es. Untuk daftar lengkap definisi sumber data yang disediakan Toko Fitur, lihat Sumber data Prosesor Fitur Feature Store Python SDK.

Sumber data kustom

Pada halaman ini kita akan menjelaskan cara membuat kelas sumber data kustom dan menunjukkan beberapa contoh penggunaan. Dengan sumber data khusus, Anda dapat menggunakan API yang disediakan SageMaker SDK for Python (Boto3) dengan cara yang sama seperti jika Anda menggunakan sumber data yang disediakan Amazon Feature Store. SageMaker

Untuk menggunakan sumber data khusus untuk mengubah dan menyerap data ke dalam grup fitur menggunakan Pemrosesan Fitur, Anda perlu memperluas `PySparkDataSource` kelas dengan anggota dan fungsi kelas berikut.

- `data_source_name`(str): nama arbitrer untuk sumber data. Misalnya, Amazon Redshift, Snowflake, atau Glue Catalog ARN.
- `data_source_unique_id`(str): pengenal unik yang mengacu pada sumber daya tertentu yang diakses. Misalnya, nama tabel, DDB Tabel ARN, awalan Amazon S3. Semua penggunaan yang sama `data_source_unique_id` dalam sumber data kustom akan dikaitkan dengan sumber data yang sama dalam tampilan garis keturunan. Lineage mencakup informasi tentang kode eksekusi alur kerja pemrosesan fitur, sumber data apa yang digunakan, dan bagaimana mereka dimasukkan ke dalam grup fitur atau fitur. Untuk informasi tentang melihat silsilah grup fitur di Studio, lihat. Lihat silsilah dari konsol
- `read_data`(func): metode yang digunakan untuk terhubung dengan prosesor fitur. Mengembalikan frame data Spark. Sebagai contoh, lihat Contoh sumber data kustom.

Keduanya `data_source_name` dan `data_source_unique_id` digunakan untuk mengidentifikasi entitas garis keturunan Anda secara unik. Berikut ini adalah contoh untuk kelas sumber data kustom bernama`CustomDataSource`.

```
from sagemaker.feature_store.feature_processor import PySparkDataSource
from pyspark.sql import DataFrame

class CustomDataSource(PySparkDataSource):

    data_source_name = "custom-data-source-name"
    data_source_unique_id = "custom-data-source-id"

    def read_data(self, parameter, spark) -> DataFrame:
        your own code here to read data into a Spark dataframe
        return dataframe
```

Contoh sumber data kustom

Bagian ini memberikan contoh implementasi sumber data khusus untuk Prosesor Fitur. Untuk informasi selengkapnya tentang sumber data khusus, lihatSumber data kustom.

Keamanan adalah tanggung jawab bersama antara AWS dan pelanggan kami. AWSbertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan diAWS Cloud. Pelanggan bertanggung jawab atas semua konfigurasi keamanan dan tugas manajemen yang diperlukan. Misalnya, rahasia seperti kredensyal akses ke penyimpanan data tidak boleh dikodekan keras dalam sumber data kustom Anda. Anda dapat menggunakan AWS Secrets Manager untuk mengelola kredensyal ini. Untuk informasi tentang Secrets Manager, lihat Apa ituAWS Secrets Manager? dalam panduan AWS Secrets Manager pengguna. Contoh berikut akan menggunakan Secrets Manager untuk kredensialmu.

Topik

- Contoh sumber data kustom Amazon Redshift Clusters (JDBC)
- Contoh sumber data kustom Snowflake
- Contoh sumber data kustom Databricks (JDBC)
- Streaming contoh sumber data kustom

Contoh sumber data kustom Amazon Redshift Clusters (JDBC)

Amazon Redshift menawarkan driver JDBC yang dapat digunakan untuk membaca data dengan Spark. Untuk informasi tentang cara mengunduh driver Amazon Redshift JDBC, lihat Mengunduh driver Amazon Redshift JDBC, versi 2.1.

Untuk membuat kelas sumber data Amazon Redshift kustom, Anda harus menimpa `read_data` metode dari. Sumber data kustom

Untuk terhubung dengan cluster Amazon Redshift, Anda memerlukan:

- URL JDBC Amazon Redshift () *`jdbc-url`*

  Untuk informasi tentang mendapatkan URL Amazon Redshift JDBC Anda, lihat Mendapatkan URL JDBC di Panduan Pengembang Basis Data Amazon Redshift.
- Nama pengguna Amazon Redshift (*`redshift-user`*) dan kata sandi () *`redshift-password`*

  Untuk informasi tentang cara membuat dan mengelola pengguna database menggunakan perintah Amazon Redshift SQL, lihat Pengguna di Panduan Pengembang Database Amazon Redshift.

- Nama tabel Amazon Redshift () *redshift-table-name*

  Untuk informasi tentang cara membuat tabel dengan beberapa contoh, lihat MEMBUAT TABEL di Panduan Developer Basis Data Amazon Redshift.

- (Opsional) Jika menggunakan Secrets Manager, Anda memerlukan nama rahasia (*secret-redshift-account-info*) tempat menyimpan nama pengguna dan kata sandi akses Amazon Redshift di Secrets Manager.

  Untuk informasi tentang Secrets Manager, lihat Menemukan rahasia AWS Secrets Manager di Panduan AWS Secrets Manager Pengguna.

- Wilayah AWS (*your-region*)

  Untuk informasi tentang mendapatkan nama wilayah sesi Anda saat ini menggunakan SDK for Python (Boto3), lihat region_name dalam dokumentasi Boto3.

Contoh berikut menunjukkan cara mengambil URL JDBC dan token akses pribadi dari Secrets Manager dan mengganti `read_data` untuk kelas sumber data kustom Anda,. `DatabricksDataSource`

```
from sagemaker.feature_store.feature_processor import PySparkDataSource
import json
import boto3


class RedshiftDataSource(PySparkDataSource):

    data_source_name = "Redshift"
    data_source_unique_id = "redshift-resource-arn"

    def read_data(self, spark, params):
        url = "jdbc-url?user=redshift-user&password=redshift-password"
        aws_iam_role_arn = "redshift-command-access-role"
        secret_name = "secret-redshift-account-info"
        region_name = "your-region"

        session = boto3.session.Session()
        sm_client = session.client(
            service_name='secretsmanager',
            region_name=region_name,
        )
```

```
        secrets = json.loads(sm_client.get_secret_value(SecretId=secret_name)
["SecretString"])
        jdbc_url = url.replace("jdbc-url", secrets["jdbcurl"]).replace("redshift-user",
 secrets['username']).replace("redshift-password", secrets['password'])

        return spark.read \
             .format("jdbc") \
             .option("url", url) \
             .option("driver", "com.amazon.redshift.Driver") \
             .option("dbtable", "redshift-table-name") \
             .option("tempdir", "s3a://your-bucket-name/your-bucket-prefix") \
             .option("aws_iam_role", aws_iam_role_arn) \
             .load()
```

Contoh berikut menunjukkan cara menghubungkan `RedshiftDataSource` ke `feature_processor` dekorator Anda.

```
from sagemaker.feature_store.feature_processor import feature_processor

@feature_processor(
    inputs=[RedshiftDataSource()],
    output="feature-group-arn",
    target_stores=["OfflineStore"],
    spark_config={"spark.jars.packages": "com.amazon.redshift:redshift-
jdbc42:2.1.0.16"}
)
def transform(input_df):
    return input_df
```

Untuk menjalankan pekerjaan prosesor fitur dari jarak jauh, Anda perlu menyediakan driver jdbc dengan mendefinisikan `SparkConfig` dan meneruskannya ke dekorator. `@remote`

```
from sagemaker.remote_function import remote
from sagemaker.remote_function.spark_config import SparkConfig

config = {
    "Classification": "spark-defaults",
    "Properties": {
      "spark.jars.packages": "com.amazon.redshift:redshift-jdbc42:2.1.0.16"
    }
}
```

```
@remote(
    spark_config=SparkConfig(configuration=config),
    instance_type="ml.m5.2xlarge",
)
@feature_processor(
    inputs=[RedshiftDataSource()],
    output="feature-group-arn",
    target_stores=["OfflineStore"],
)
def transform(input_df):
    return input_df
```

Contoh sumber data kustom Snowflake

Snowflake menyediakan konektor Spark yang dapat digunakan untuk dekorator Anda. `feature_processor` Untuk informasi tentang konektor Snowflake untuk Spark, lihat Snowflake Connector for Spark di dokumentasi Snowflake.

Untuk membuat kelas sumber data Snowflake kustom, Anda harus mengganti `read_data` metode dari Sumber data kustom dan menambahkan paket konektor Spark ke classpath Spark.

Untuk terhubung dengan sumber data Snowflake yang Anda butuhkan:

- URL kepingan salju () *sf-url*

  Untuk informasi tentang URL untuk mengakses antarmuka web Snowflake, lihat Pengenal Akun dalam dokumentasi Snowflake.
- Database kepingan salju () *sf-database*

  Untuk informasi tentang mendapatkan nama database Anda menggunakan Snowflake, lihat CURRENT_DATABASE dalam dokumentasi Snowflake.
- Skema basis data kepingan salju () *sf-schema*

  Untuk informasi tentang mendapatkan nama skema Anda menggunakan Snowflake, lihat CURRENT_SCHEMA di dokumentasi Snowflake.
- Gudang kepingan salju () *sf-warehouse*

  Untuk informasi tentang mendapatkan nama gudang Anda menggunakan Snowflake, lihat CURRENT_WAREHOUSE di dokumentasi Snowflake.

- Nama tabel kepingan salju () *sf-table-name*
- (Opsional) Jika menggunakan Secrets Manager, Anda akan memerlukan nama rahasia (*secret-snowflake-account-info*) tempat Anda menyimpan nama pengguna dan kata sandi akses Snowflake di Secrets Manager.

  Untuk informasi tentang Secrets Manager, lihat Menemukan rahasia AWS Secrets Manager di Panduan AWS Secrets Manager Pengguna.
- Wilayah AWS (*your-region*)

  Untuk informasi tentang mendapatkan nama wilayah sesi Anda saat ini menggunakan SDK for Python (Boto3), lihat region_name dalam dokumentasi Boto3.

Contoh berikut menunjukkan cara mengambil nama pengguna dan kata sandi Snowflake dari Secrets Manager dan mengganti `read_data` fungsi untuk kelas sumber data kustom Anda. `SnowflakeDataSource`

```
from sagemaker.feature_store.feature_processor import PySparkDataSource
from sagemaker.feature_store.feature_processor import feature_processor
import json
import boto3


class SnowflakeDataSource(PySparkDataSource):

    sf_options = {
        "sfUrl" : "sf-url",
        "sfDatabase" : "sf-database",
        "sfSchema" : "sf-schema",
        "sfWarehouse" : "sf-warehouse",
    }

    data_source_name = "Snowflake"
    data_source_unique_id = "sf-url"

    def read_data(self, spark, params):
        secret_name = "secret-snowflake-account-info"
        region_name = "your-region"

        session = boto3.session.Session()
        sm_client = session.client(
            service_name='secretsmanager',
```

```
            region_name=region_name,
        )

        secrets = json.loads(sm_client.get_secret_value(SecretId=secret_name)
["SecretString"])
        self.sf_options["sfUser"] = secrets.get("username")
        self.sf_options["sfPassword"] = secrets.get("password")

        return spark.read.format("net.snowflake.spark.snowflake") \
                        .options(**self.sf_options) \
                        .option("dbtable", "sf-table-name") \
                        .load()
```

Contoh berikut menunjukkan cara menghubungkan `SnowflakeDataSource` ke `feature_processor` dekorator Anda.

```
from sagemaker.feature_store.feature_processor import feature_processor

@feature_processor(
    inputs=[SnowflakeDataSource()],
    output=feature-group-arn,
    target_stores=["OfflineStore"],
    spark_config={"spark.jars.packages": "net.snowflake:spark-snowflake_2.12:2.12.0-
spark_3.3"}
)
def transform(input_df):
    return input_df
```

Untuk menjalankan pekerjaan prosesor fitur dari jarak jauh, Anda perlu menyediakan paket melalui mendefinisikan `SparkConfig` dan meneruskannya ke `@remote` dekorator. Paket Spark dalam contoh berikut sedemikian rupa sehingga `spark-snowflake_2.12` merupakan versi Feature Processor Scala, `2.12.0` adalah versi Snowflake yang ingin Anda gunakan, dan `spark_3.3` merupakan versi Feature Processor Spark.

```
from sagemaker.remote_function import remote
from sagemaker.remote_function.spark_config import SparkConfig

config = {
    "Classification": "spark-defaults",
    "Properties": {
      "spark.jars.packages": "net.snowflake:spark-snowflake_2.12:2.12.0-spark_3.3"
    }
```

```
}

@remote(
    spark_config=SparkConfig(configuration=config),
    instance_type="ml.m5.2xlarge",
)
@feature_processor(
    inputs=[SnowflakeDataSource()],
    output="feature-group-arn>",
    target_stores=["OfflineStore"],
)
def transform(input_df):
    return input_df
```

Contoh sumber data kustom Databricks (JDBC)

Spark dapat membaca data dari Databricks dengan menggunakan driver Databricks JDBC. Untuk informasi tentang driver JDBC Databricks, lihat Mengkonfigurasi driver Databricks ODBC dan JDBC dalam dokumentasi Databricks.

> **Note**
>
> Anda dapat membaca data dari database lain dengan memasukkan driver JDBC yang sesuai di Spark classpath. Untuk informasi selengkapnya, lihat JDBC To Other Databases di Spark SQL Guide.

Untuk membuat kelas sumber data Databricks kustom, Anda harus mengganti `read_data` metode dari Sumber data kustom dan menambahkan jar JDBC ke classpath Spark.

Untuk terhubung dengan sumber data Databricks yang Anda butuhkan:

- URL Databricks () *`databricks-url`*

  Untuk informasi tentang URL Databricks Anda, lihat Membangun URL koneksi untuk driver Databricks dalam dokumentasi Databricks.
- Databricks token akses pribadi () *`personal-access-token`*

  Untuk informasi tentang token akses Databricks Anda, lihat otentikasi token akses pribadi Databricks dalam dokumentasi Databricks.

- Nama katalog data (*db-catalog*)

  Untuk informasi tentang nama katalog Databricks Anda, lihat Nama katalog dalam dokumentasi Databricks.

- Nama skema () *db-schema*

  Untuk informasi tentang nama skema Databricks Anda, lihat Nama skema dalam dokumentasi Databricks.

- Nama tabel (*db-table-name*)

  Untuk informasi tentang nama tabel Databricks Anda, lihat Nama tabel dalam dokumentasi Databricks.

- (Opsional) Jika menggunakan Secrets Manager, Anda memerlukan nama rahasia (*secret-databricks-account-info*) tempat menyimpan nama pengguna dan kata sandi akses Databricks di Secrets Manager.

  Untuk informasi tentang Secrets Manager, lihat Menemukan rahasia AWS Secrets Manager di Panduan AWS Secrets Manager Pengguna.

- Wilayah AWS (*your-region*)

  Untuk informasi tentang mendapatkan nama wilayah sesi Anda saat ini menggunakan SDK for Python (Boto3), lihat region_name dalam dokumentasi Boto3.

Contoh berikut menunjukkan cara mengambil URL JDBC dan token akses pribadi dari Secrets Manager dan menimpa `read_data` untuk kelas sumber data kustom Anda,. `DatabricksDataSource`

```
from sagemaker.feature_store.feature_processor import PySparkDataSource
import json
import boto3


class DatabricksDataSource(PySparkDataSource):

    data_source_name = "Databricks"
    data_source_unique_id = "databricks-url"

    def read_data(self, spark, params):
        secret_name = "secret-databricks-account-info"
```

```
        region_name = "your-region"

        session = boto3.session.Session()
        sm_client = session.client(
            service_name='secretsmanager',
            region_name=region_name,
        )

        secrets = json.loads(sm_client.get_secret_value(SecretId=secret_name)
["SecretString"])
        jdbc_url = secrets["jdbcurl"].replace("personal-access-token", secrets['pwd'])

        return spark.read.format("jdbc") \
                        .option("url", jdbc_url) \
                        .option("dbtable","`db-catalog`.`db-schema`.`db-table-name`") \
                        .option("driver", "com.simba.spark.jdbc.Driver") \
                        .load()
```

Contoh berikut menunjukkan cara mengunggah jar driver JDBC,*jdbc-jar-file-name*.jar, ke Amazon S3 untuk menambahkannya ke classpath Spark. Untuk informasi tentang mengunduh driver Spark JDBC (*jdbc-jar-file-name*.jar) dari Databricks, lihat Mengunduh Driver JDBC di situs web Databricks.

```
from sagemaker.feature_store.feature_processor import feature_processor

@feature_processor(
    inputs=[DatabricksDataSource()],
    output=feature-group-arn,
    target_stores=["OfflineStore"],
    spark_config={"spark.jars": "s3://your-bucket-name/your-bucket-prefix/jdbc-jar-
file-name.jar"}
)
def transform(input_df):
    return input_df
```

Untuk menjalankan pekerjaan prosesor fitur dari jarak jauh, Anda perlu menyediakan stoples dengan mendefinisikan `SparkConfig` dan meneruskannya ke dekorator. `@remote`

```
from sagemaker.remote_function import remote
from sagemaker.remote_function.spark_config import SparkConfig

config = {
```

```
    "Classification": "spark-defaults",
    "Properties": {
      "spark.jars": "s3://your-bucket-name/your-bucket-prefix/jdbc-jar-file-name.jar"
    }
}

@remote(
    spark_config=SparkConfig(configuration=config),
    instance_type="ml.m5.2xlarge",
)
@feature_processor(
    inputs=[DatabricksDataSource()],
    output="feature-group-arn",
    target_stores=["OfflineStore"],
)
def transform(input_df):
    return input_df
```

Streaming contoh sumber data kustom

Anda dapat terhubung ke sumber data streaming seperti Amazon Kinesis, dan penulis mengubah dengan Spark Structured Streaming untuk membaca dari sumber data streaming. Untuk informasi tentang konektor Kinesis, lihat Konektor Kinesis untuk Streaming Terstruktur Spark di. GitHub Untuk informasi tentang Amazon Kinesis, lihat Apa itu Amazon Kinesis Data Streams? di Panduan Pengembang Amazon Kinesis.

Untuk membuat kelas sumber data Amazon Kinesis kustom, Anda perlu memperluas `BaseDataSource` kelas dan mengganti metode dari`read_data`. Sumber data kustom

Untuk terhubung ke Amazon Kinesis data stream, Anda membutuhkan:

- Kinesis ARN () *`kinesis-resource-arn`*

  Untuk informasi tentang ARN aliran data Kinesis, lihat Amazon Resource Name (ARN) untuk Kinesis Data Streams di Panduan Pengembang Amazon Kinesis Kinesis.
- Nama aliran data Kinesis () *`kinesis-stream-name`*
- Wilayah AWS (*`your-region`*)

  Untuk informasi tentang mendapatkan nama wilayah sesi Anda saat ini menggunakan SDK for Python (Boto3), lihat region_name dalam dokumentasi Boto3.

```
from sagemaker.feature_store.feature_processor import BaseDataSource
from sagemaker.feature_store.feature_processor import feature_processor

class KinesisDataSource(BaseDataSource):

    data_source_name = "Kinesis"
    data_source_unique_id = "kinesis-resource-arn"

    def read_data(self, spark, params):
        return spark.readStream.format("kinesis") \
            .option("streamName", "kinesis-stream-name") \
            .option("awsUseInstanceProfile", "false") \
            .option("endpointUrl", "https://kinesis.your-region.amazonaws.com")
            .load()
```

Contoh berikut menunjukkan cara menghubungkan `KinesisDataSource` ke `feature_processor` dekorator Anda.

```
from sagemaker.remote_function import remote
from sagemaker.remote_function.spark_config import SparkConfig
import feature_store_pyspark.FeatureStoreManager as fsm

def ingest_micro_batch_into_fg(input_df, epoch_id):
    feature_group_arn = "feature-group-arn"
    fsm.FeatureStoreManager().ingest_data(
        input_data_frame = input_df,
        feature_group_arn = feature_group_arn
    )

@remote(
    spark_config=SparkConfig(
        configuration={
            "Classification": "spark-defaults",
            "Properties":{
                "spark.sql.streaming.schemaInference": "true",
                "spark.jars.packages": "com.roncemer.spark/spark-sql-
kinesis_2.13/1.2.2_spark-3.2"
            }
        }
    ),
    instance_type="ml.m5.2xlarge",
    max_runtime_in_seconds=2419200 # 28 days
)
```

```
@feature_processor(
    inputs=[KinesisDataSource()],
    output="feature-group-arn"
)
def transform(input_df):
    output_stream = (
        input_df.selectExpr("CAST(rand() AS STRING) as partitionKey", "CAST(data AS
 STRING)")
        .writeStream.foreachBatch(ingest_micro_batch_into_fg)
        .trigger(processingTime="1 minute")
        .option("checkpointLocation", "s3a://checkpoint-path")
        .start()
    )
    output_stream.awaitTermination()
```

Dalam contoh kode di atas, kami menggunakan beberapa opsi Streaming Terstruktur Spark saat mengalirkan batch mikro ke grup fitur Anda. Untuk daftar lengkap opsi, lihat Panduan Pemrograman Streaming Terstruktur di dokumentasi Apache Spark.

- Mode `foreachBatch` wastafel adalah fitur yang memungkinkan Anda menerapkan operasi dan menulis logika pada data keluaran setiap batch mikro dari kueri streaming.

  Untuk informasi tentang`foreachBatch`, lihat Menggunakan Foreach dan ForeachBatch di Panduan Pemrograman Streaming Terstruktur Apache Spark.

- `checkpointLocation`Opsi ini secara berkala menyimpan keadaan aplikasi streaming. Log streaming disimpan di lokasi *`s3a://checkpoint-path`* pos pemeriksaan.

  Untuk informasi tentang `checkpointLocation` opsi, lihat Memulihkan dari Kegagalan dengan Checkpointing di Panduan Pemrograman Streaming Terstruktur Apache Spark.

- `trigger`Pengaturan menentukan seberapa sering pemrosesan batch mikro dipicu dalam aplikasi streaming. Dalam contoh, jenis pemicu waktu pemrosesan digunakan dengan interval batch mikro satu menit, yang ditentukan oleh. `trigger(processingTime="1 minute")` Untuk mengisi ulang dari sumber aliran, Anda dapat menggunakan tipe pemicu yang tersedia sekarang, yang ditentukan oleh. `trigger(availableNow=True)`

  Untuk daftar lengkap `trigger` jenis, lihat Pemicu dalam Panduan Pemrograman Streaming Terstruktur Apache Spark.

Streaming berkelanjutan dan percobaan ulang otomatis menggunakan pemicu berbasis peristiwa

Prosesor Fitur menggunakan SageMaker Pelatihan sebagai infrastruktur komputasi dan memiliki batas waktu proses maksimum 28 hari. Anda dapat menggunakan pemicu berbasis peristiwa untuk memperpanjang streaming berkelanjutan Anda untuk jangka waktu yang lebih lama dan pulih dari kegagalan sementara. Untuk informasi selengkapnya tentang eksekusi berdasarkan jadwal dan acara, lihatEksekusi terjadwal dan berbasis acara untuk pipeline Prosesor Fitur.

Berikut ini adalah contoh pengaturan pemicu berbasis peristiwa untuk menjaga saluran Prosesor Fitur streaming tetap berjalan terus menerus. Ini menggunakan fungsi transformasi streaming yang didefinisikan dalam contoh sebelumnya. Pipeline target dapat dikonfigurasi untuk dipicu ketika `FAILED` peristiwa `STOPPED` atau terjadi untuk eksekusi pipeline sumber. Perhatikan bahwa pipeline yang sama digunakan sebagai sumber dan target sehingga berjalan terus menerus.

```
import sagemaker.feature_store.feature_processor as fp
from sagemaker.feature_store.feature_processor import FeatureProcessorPipelineEvent
from sagemaker.feature_store.feature_processor import
 FeatureProcessorPipelineExecutionStatus

streaming_pipeline_name = "streaming-pipeline"
streaming_pipeline_arn = fp.to_pipeline(
    pipeline_name = streaming_pipeline_name,
    step = transform # defined in previous section
)

fp.put_trigger(
    source_pipeline_events=FeatureProcessorPipelineEvents(
        pipeline_name=source_pipeline_name,
        pipeline_execution_status=[
            FeatureProcessorPipelineExecutionStatus.STOPPED,
            FeatureProcessorPipelineExecutionStatus.FAILED]
    ),
    target_pipeline=target_pipeline_name
)
```

Contoh kode Pemrosesan Fitur untuk kasus penggunaan umum

Contoh-contoh berikut memberikan contoh kode Pemrosesan Fitur untuk kasus penggunaan umum. Untuk contoh notebook yang lebih detail yang menampilkan kasus penggunaan tertentu, lihat Notebook Pemrosesan SageMaker Fitur Amazon Feature Store.

Dalam contoh berikut, *us-east-1* adalah wilayah sumber daya, *111122223333* adalah ID akun pemilik sumber daya, dan *your-feature-group-name* merupakan nama grup fitur.

Kumpulan `transactions` data yang digunakan dalam contoh berikut memiliki skema berikut:

```
'FeatureDefinitions': [
  {'FeatureName': 'txn_id', 'FeatureType': 'String'},
  {'FeatureName': 'txn_time', 'FeatureType': 'String'},
  {'FeatureName': 'credit_card_num', 'FeatureType': 'String'},
  {'FeatureName': 'txn_amount', 'FeatureType': 'Fractional'}
]
```

Topik

- Menggabungkan data dari berbagai sumber data
- Agregat jendela geser
- Agregat jendela tumbling
- Promosi dari toko offline ke toko online
- Transformasi dengan perpustakaan Pandas
- Eksekusi berkelanjutan dan percobaan ulang otomatis menggunakan pemicu berbasis peristiwa

Menggabungkan data dari berbagai sumber data

```
@feature_processor(
    inputs=[
        CSVDataSource('s3://bucket/customer'),
        FeatureGroupDataSource('transactions')
    ],
    output='arn:aws:sagemaker:us-east-1:111122223333:feature-group/your-feature-group-name'
)
def join(transactions_df, customer_df):
  '''Combine two data sources with an inner join on a common column'''

  return transactions_df.join(
    customer_df, transactions_df.customer_id == customer_df.customer_id, "inner"
  )
```

Agregat jendela geser

```
@feature_processor(
    inputs=[FeatureGroupDataSource('transactions')],
```

```
    output='arn:aws:sagemaker:us-east-1:111122223333:feature-group/your-feature-group-
name'
)
def sliding_window_aggregates(transactions_df):
    '''Aggregates over 1-week windows, across 1-day sliding windows.'''
    from pyspark.sql.functions import window, avg, count

    return (
        transactions_df
            .groupBy("credit_card_num", window("txn_time", "1 week", "1 day"))
            .agg(avg("txn_amount").alias("avg_week"), count("*").alias("count_week"))
            .orderBy("window.start")
            .select("credit_card_num", "window.start", "avg_week", "count_week")
    )
```

Agregat jendela tumbling

```
@feature_processor(
    inputs=[FeatureGroupDataSource('transactions')],
    output='arn:aws:sagemaker:us-east-1:111122223333:feature-group/your-feature-group-
name'
)
def tumbling_window_aggregates(transactions_df, spark):
    '''Aggregates over 1-week windows, across 1-day tumbling windows, as a SQL
 query.'''

    transactions_df.createOrReplaceTempView('transactions')
    return spark.sql(f'''
        SELECT credit_card_num, window.start, AVG(amount) AS avg, COUNT(*) AS count
        FROM transactions
        GROUP BY credit_card_num, window(txn_time, "1 week")
        ORDER BY window.start
    ''')
```

Promosi dari toko offline ke toko online

```
@feature_processor(
    inputs=[FeatureGroupDataSource('transactions')],
    target_stores=['OnlineStore'],
    output='arn:aws:sagemaker:us-east-1:111122223333:feature-group/transactions'
)
def offline_to_online():
```

```
    '''Move data from the offline store to the online store of the same feature
 group.'''

    transactions_df.createOrReplaceTempView('transactions')
    return spark.sql(f'''
        SELECT txn_id, txn_time, credit_card_num, amount
        FROM
            (SELECT *,
            row_number()
            OVER
                (PARTITION BY txn_id
                ORDER BY "txn_time" DESC, Api_Invocation_Time DESC, write_time DESC)
            AS row_number
            FROM transactions)
        WHERE row_number = 1
    ''')
```

Transformasi dengan perpustakaan Pandas

Transformasi dengan perpustakaan Pandas

```
@feature_processor(
    inputs=[FeatureGroupDataSource('transactions')],
    target_stores=['OnlineStore'],
    output='arn:aws:sagemaker:us-east-1:111122223333:feature-group/transactions'
)
def pandas(transactions_df):
    '''Author transformations using the Pandas interface.

    Requires PyArrow to be installed via pip.
    For more details: https://spark.apache.org/docs/latest/api/python/user_guide/
pandas_on_spark
    '''
    import pyspark.pandas as ps

    # PySpark DF to Pandas-On-Spark DF (Distributed DF with Pandas interface).
    pandas_on_spark_df = transactions_df.pandas_api()
    # Pandas-On-Spark DF to Pandas DF (Single Machine Only).
    pandas_df = pandas_on_spark_df.to_pandas()

    # Reverse: Pandas DF to Pandas-On-Spark DF
    pandas_on_spark_df = ps.from_pandas(pandas_df)
    # Reverse: Pandas-On-Spark DF to PySpark DF
```

```
    spark_df = pandas_on_spark_df.to_spark()

    return spark_df
```

Eksekusi berkelanjutan dan percobaan ulang otomatis menggunakan pemicu berbasis peristiwa

```
from sagemaker.feature_store.feature_processor import put_trigger, to_pipeline,
 FeatureProcessorPipelineEvent
from sagemaker.feature_store.feature_processor import
 FeatureProcessorPipelineExecutionStatus

streaming_pipeline_name = "target-pipeline"

to_pipeline(
    pipeline_name=streaming_pipeline_name,
    step=transform
)

put_trigger(
    source_pipeline_events=[
        FeatureProcessorPipelineEvent(
            pipeline_name=streaming_pipeline_name,
            pipeline_execution_status=[
            FeatureProcessorPipelineExecutionStatus.STOPPED,
            FeatureProcessorPipelineExecutionStatus.FAILED]
        )
    ],
    target_pipeline=streaming_pipeline_name
)
```

Durasi waktu untuk tayang (TTL) untuk rekaman

Amazon SageMaker Feature Store menyediakan opsi agar catatan dihapus dengan keras dari toko online setelah durasi waktu tercapai, dengan durasi time to live (TTL) (`TtlDuration`). Catatan akan kedaluwarsa setelah catatan `EventTime` ditambah `TtlDuration` tercapai, atau `ExpiresAt` = `EventTime` +`TtlDuration`. `TtlDuration`Dapat diterapkan pada tingkat grup fitur, di mana semua catatan dalam grup fitur akan memiliki secara `TtlDuration` default, atau pada tingkat catatan individu. Jika tidak `TtlDuration` ditentukan, nilai defaultnya adalah `null` dan catatan akan tetap berada di toko online sampai ditimpa.

Catatan yang dihapus menggunakan `TtlDuration` sulit dihapus, atau sepenuhnya dihapus dari toko online, dan catatan yang dihapus ditambahkan ke toko offline. Untuk informasi selengkapnya tentang mode penghapusan dan penghapusan keras, lihat `DeleteRecord`di panduan Referensi Amazon SageMaker API. Ketika catatan dihapus dengan keras, itu segera menjadi tidak dapat diakses menggunakan API Toko Fitur.

> **Important**
>
> TTL biasanya menghapus item kedaluwarsa dalam waktu beberapa hari Bergantung pada ukuran dan tingkat aktivitas tabel, operasi penghapusan aktual dari item yang kedaluwarsa dapat bervariasi. Karena TTL dimaksudkan sebagai proses latar belakang, sifat kapasitas yang digunakan untuk kedaluwarsa dan menghapus item melalui TTL adalah variabel (tetapi gratis). Untuk informasi selengkapnya tentang cara item dihapus dari tabel DynamoDB, lihat Cara kerjanya: DynamoDB Time to Live (TTL).

`TtlDuration`harus berupa kamus yang berisi a `Unit` dan a`Value`, di mana `Unit` harus berupa string dengan nilai “Detik”, “Menit”, “Jam”, “Hari”, atau “Minggu” dan `Value` harus bilangan bulat lebih besar dari atau sama dengan 1. `TtlDuration`dapat diterapkan saat menggunakan`CreateFeatureGroup`,`UpdateFeatureGroup`, dan `PutRecord` API. Lihat sintaks permintaan dan respons dalam dokumentasi SDK for Python (Boto3) untuk,, dan API. `CreateFeatureGroup``UpdateFeatureGroup``PutRecord`

- Ketika `TtlDuration` diterapkan pada tingkat grup fitur (menggunakan `CreateFeatureGroup` atau `UpdateFeatureGroup` API), yang diterapkan `TtlDuration` menjadi default `TtlDuration` untuk semua catatan yang ditambahkan ke grup fitur dari titik waktu API dipanggil. Saat menerapkan `TtlDuration` dengan `UpdateFeatureGroup` API, ini tidak akan menjadi default `TtlDuration` untuk catatan yang dibuat sebelum API dipanggil.
- Ketika `TtlDuration` diterapkan pada tingkat rekaman (misalnya, menggunakan `PutRecord` API), `TtlDuration` durasi berlaku untuk rekaman itu dan digunakan sebagai pengganti default tingkat grup fitur`TtlDuration`.
- Ketika `TtlDuration` diterapkan pada tingkat grup fitur, mungkin perlu beberapa menit `TtlDuration` untuk mulai berlaku.
- Jika `TtlDuration` digunakan ketika tidak ada toko online, Anda akan menerima `Validation Exception (400)` kesalahan.

Contoh kode berikut menunjukkan cara menerapkan `TtlDuration` saat memperbarui grup fitur, sehingga catatan yang ditambahkan ke grup fitur setelah menjalankan API secara default akan kedaluwarsa empat minggu setelah waktu acara mereka.

```
import boto3

sagemaker_client = boto3.client("sagemaker")
feature_group_name = '<YOUR_FEATURE_GROUP_NAME>'

sagemaker_client.update_feature_group(
    FeatureGroupName=feature_group_name,
    OnlineStoreConfig={
        TtlDuration:{
            Unit: "Weeks",
            Value: 4
        }
    }
)
```

Anda dapat menggunakan `DescribeFeatureGroup` API untuk melihat default`TtlDuration`.

Untuk melihat waktu kedaluwarsa, `ExpiresAt` (dalam format ISO-8601 waktu UTC), saat menggunakan API atau yang harus Anda atur. `GetRecord` `BatchGetRecord` `ExpirationTimeResponse` `ENABLED` Lihat sintaks permintaan dan respons dalam dokumentasi SDK for Python (Boto3) untuk,, dan API. `DescribeFeatureGroup` `GetRecord` `BatchGetRecord`

Kemampuan dan akses grup fitur lintas akun

Ilmuwan data dan insinyur data dapat memperoleh manfaat dari mengeksplorasi dan mengakses fitur yang menjangkau banyak akun, untuk mempromosikan konsistensi data, merampingkan kolaborasi, dan mengurangi duplikasi upaya.

Dengan Amazon SageMaker Feature Store, Anda dapat berbagi sumber daya grup fitur di seluruh akun. Sumber daya yang dapat dibagikan di Toko Fitur adalah entitas grup fitur atau katalog grup fitur, tempat katalog grup fitur berisi semua entitas grup fitur di akun Anda. Akun pemilik sumber daya berbagi sumber daya dengan akun konsumen sumber daya. Ada dua kategori izin yang berbeda yang terkait dengan berbagi sumber daya:

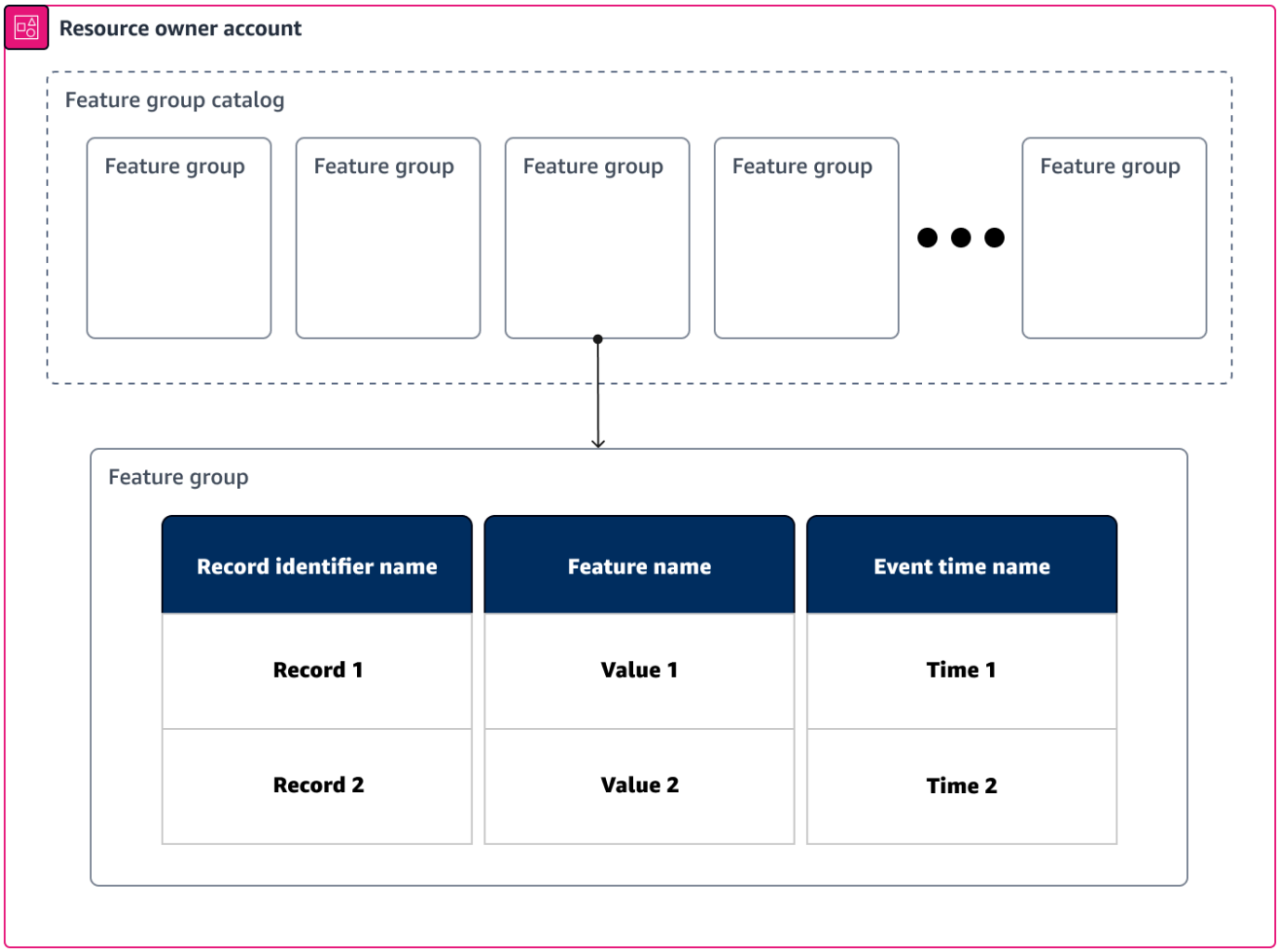
- Izin Discoverability: Discoverability berarti dapat melihat nama grup fitur dan metadata. Saat Anda membagikan katalog grup fitur dan memberikan izin untuk dapat ditemukan, semua entitas grup

fitur di akun yang Anda bagikan (akun pemilik sumber daya) dapat ditemukan oleh akun yang Anda bagikan (akun konsumen sumber daya). Misalnya, jika Anda membuat katalog grup fitur di akun pemilik sumber daya dapat ditemukan ke akun konsumen sumber daya, maka prinsipal akun konsumen sumber daya dapat melihat semua grup fitur yang terdapat dalam akun pemilik sumber daya. Ini berarti kemampuan untuk ditemukan adalah “semua atau tidak sama sekali” di tingkat akun (regionalisasi). Izin ini diberikan ke akun konsumen sumber daya dengan menggunakan tipe sumber daya katalog grup fitur.

- Izin akses: Saat Anda memberikan izin akses, Anda melakukannya di tingkat sumber daya grup fitur (bukan di tingkat akun). Ini memberi Anda kontrol yang lebih terperinci atas pemberian akses ke data. Jenis izin akses yang dapat diberikan adalah: read-only, read-write, dan admin. Misalnya, Anda hanya dapat memilih grup fitur tertentu dari akun pemilik sumber daya agar dapat diakses oleh kepala sekolah akun konsumen sumber daya, tergantung pada kebutuhan bisnis Anda. Izin ini diberikan ke akun konsumen sumber daya dengan menggunakan tipe sumber daya grup fitur dan menentukan entitas grup fitur.

Perbedaan antara kemampuan untuk ditemukan dan akses penting untuk diingat ketika Anda mengatur berbagi lintas akun. Selain itu, metode berbagi sumber daya berbeda tergantung pada apakah Anda berbagi grup fitur online atau offline. Untuk informasi tentang grup fitur online dan offline, lihatKonsep Feature Store. Dalam topik berikut, Anda dapat mempelajari cara menerapkan kemampuan untuk dapat ditemukan dan izin akses ke sumber daya bersama Anda.

Diagram contoh berikut memvisualisasikan sumber daya katalog grup fitur versus entitas sumber daya grup fitur. Katalog grup fitur berisi semua entitas grup fitur Anda dan dapat dibagikan menggunakan izin untuk dapat ditemukan. Ketika diberikan izin untuk dapat ditemukan, akun konsumen sumber daya dapat mencari dan menemukan semua entitas grup fitur dalam akun pemilik sumber daya. Entitas grup fitur berisi data pembelajaran mesin Anda dan dapat dibagikan menggunakan izin akses. Ketika diberikan izin akses, akun konsumen sumber daya dapat mengakses data grup fitur, dengan akses ditentukan oleh izin akses yang relevan.



Topik

- Mengaktifkan kemampuan penemuan lintas akun
- Mengaktifkan akses lintas akun

Mengaktifkan kemampuan penemuan lintas akun

Dengan AWS Resource Access Manager (AWS RAM) Anda dapat berbagi katalog grup fitur dengan aman, yang berisi semua grup fitur dan sumber daya fitur, dengan lainnyaAkun AWS. Ini memungkinkan anggota tim Anda mencari dan menemukan grup fitur dan fitur yang menjangkau beberapa akun, mempromosikan konsistensi data, merampingkan kolaborasi, dan mengurangi duplikasi upaya.

Akun pemilik sumber daya dapat berbagi sumber daya dengan individu lain Akun AWS dengan memberikan izin menggunakan. AWS RAM Akun konsumen sumber daya adalah Akun AWS dengan siapa sumber daya dibagikan, dibatasi oleh izin yang diberikan dari akun pemilik sumber daya. Jika Anda adalah organisasi, Anda mungkin ingin memanfaatkannyaAWS Organizations, yang dengannya Anda dapat berbagi sumber daya dengan individuAkun AWS, dengan semua akun di organisasi Anda, atau di Unit Organisasi (OU), tanpa harus menerapkan izin ke setiap akun. Untuk video instruksional dan informasi lebih lanjut tentang AWS RAM konsep dan manfaat, lihat Apa itu? AWS Resource Access Manager dalam AWS RAM User Guide.

Bagian ini mencakup bagaimana akun pemilik sumber daya dapat memilih katalog grup fitur dan memberikan hak istimewa untuk dapat ditemukan ke akun konsumen sumber daya, dan kemudian bagaimana akun konsumen sumber daya dengan hak istimewa dapat ditemukan dapat menggunakan pencarian dan menemukan grup fitur dalam akun pemilik sumber daya. Izin discoverability tidak memberikan izin akses (read-only, read-write, atau admin). Izin akses diberikan pada tingkat sumber daya dan bukan di tingkat akun. Untuk informasi tentang pemberian izin akses, lihat. Mengaktifkan akses lintas akun

Topik berikut membahas cara membagikan katalog grup fitur dan cara mencari sumber daya bersama dengan izin dapat ditemukan yang diterapkan.

Topik

- Bagikan katalog grup fitur Anda
- Cari sumber daya yang dapat ditemukan

Bagikan katalog grup fitur Anda

Katalog grup fitur,`DefaultFeatureGroupCatalog`, berisi semua entitas grup fitur yang dimiliki oleh akun pemilik sumber daya. Katalog dapat dibagikan oleh akun pemilik sumber daya untuk memberikan kemampuan ditemukan ke satu atau beberapa akun konsumen sumber daya, dengan membuat pembagian sumber daya di AWS Resource Access Manager ()AWS RAM. Grup fitur adalah sumber daya utama di Amazon SageMaker Feature Store dan terdiri dari definisi fitur dan catatan yang dikelola oleh Feature Store. Untuk informasi selengkapnya tentang grup fitur, lihatKonsep Feature Store.

Discoverability berarti bahwa akun konsumen sumber daya dapat mencari sumber daya yang dapat ditemukan dan melihatnya seolah-olah mereka berada di akun mereka sendiri (tidak termasuk tag). Saat mengizinkan katalog grup fitur dapat ditemukan, akun konsumen sumber daya secara

default tidak diberikan izin akses (hanya-baca, baca tulis, atau admin). Izin akses diberikan pada tingkat sumber daya dan bukan di tingkat akun. Untuk informasi tentang pemberian izin akses, lihat. Mengaktifkan akses lintas akun

Untuk mengaktifkan kemampuan penemuan lintas akun, Anda harus menentukan Katalog SageMaker Sumber Daya dan katalog grup fitur saat menggunakan instruksi AWS RAMBuat berbagi sumber daya di panduan AWS RAM pengembang. Berikut ini kami memberikan spesifikasi untuk menggunakan instruksi AWS RAM konsol.

1. Tentukan detail berbagi sumber daya:

   - Jenis sumber daya: Pilih Katalog SageMaker Sumber Daya.
   - ARN: Pilih katalog grup fitur ARN dengan format: `arn:aws:sagemaker:`*`us-east-1`*`:`*`111122223333`*`:sagemaker-catalog/DefaultFeatureGroupCatalog`

     *`us-east-1`*adalah wilayah sumber daya dan *`111122223333`*merupakan ID akun pemilik sumber daya.
   - ID Sumber Daya: Pilih`DefaultFeatureGroupCatalog`.
2. Izin terkelola asosiasi:

   - Izin terkelola: Pilih`AWSRAMPermissionSageMakerCatalogResourceSearch`.
3. Berikan akses ke kepala sekolah:

   - Pilih tipe utama (Akun AWS, Organisasi, atau Unit Organisasi) dan masukkan ID yang sesuai.

     Jika Anda adalah organisasi, Anda mungkin ingin memanfaatkannyaAWS Organizations, yang dengannya Anda dapat berbagi sumber daya dengan individuAkun AWS, dengan semua akun di organisasi Anda, atau dengan Unit Organisasi (OU), tanpa harus menerapkan izin ke setiap akun. Untuk informasi selengkapnya tentang berbagi sumber daya dan memberikan izin di dalamnyaAWS, lihat Mengaktifkan berbagi sumber daya AWS Organizations di dalam Panduan AWS Resource Access Manager Pengembang.
4. Tinjau dan buat:

   - Tinjau lalu pilih Buat berbagi sumber daya.

Mungkin perlu waktu beberapa menit agar asosiasi berbagi sumber daya dan prinsipal, atau akun konsumen sumber daya, selesai diselesaikan. Setelah pembagian sumber daya dan asosiasi utama ditetapkan, akun konsumen sumber daya yang ditentukan menerima undangan untuk bergabung

dengan pembagian sumber daya. Akun konsumen sumber daya dapat melihat dan menerima undangan dengan membuka halaman Shared with me: Resource shares di AWS RAM konsol. Untuk informasi selengkapnya tentang menerima dan melihat sumber dayaAWS RAM, lihat Mengakses AWS sumber daya yang dibagikan dengan Anda. Undangan tidak dikirim dalam kasus ini:

- Jika Anda adalah bagian dari sebuah organisasi di AWS Organizations dan berbagi di organisasi Anda diaktifkan, maka kepala sekolah di organisasi secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.
- Jika Anda berbagi dengan Akun AWS yang memiliki sumber daya, maka prinsipal di akun itu secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.

Untuk informasi selengkapnya tentang menerima dan menggunakan pembagian sumber daya, lihatCari sumber daya yang dapat ditemukan.

Bagikan katalog grup fitur menggunakan AWS SDK for Python (Boto3)

Anda dapat menggunakan AWS RAM API AWS SDK for Python (Boto3) for untuk membuat pembagian sumber daya. Kode berikut adalah contoh ID akun pemilik sumber daya di *111122223333*wilayah *us-east-1* yang membuat pembagian sumber daya bernama *test-cross-account-catalog*, berbagi katalog grup fitur dengan ID akun konsumen sumber daya. *444455556666* Untuk menggunakan Python SDK untuk AWS RAM API, lampirkan `AWSRAMPermissionSageMakerCatalogResourceSearch` kebijakan dengan peran eksekusi. Lihat AWS RAMAPI untuk detail selengkapnya.

```
#Call list resource catalogs as a prerequisite for RAM share
sagemaker_client.list_resource_catalogs()

# Share DefaultFeatureGroupCatalog with other account
ram_client = boto3.client("ram")
response = ram_client.create_resource_share(
    name='test-cross-account-catalog', # Change to your custom resource share name
    resourceArns=[
        'arn:aws:sagemaker:us-east-1:111122223333:sagemaker-catalog/' +
 'DefaultFeatureGroupCatalog', # Change 111122223333 to the resource owner account ID
    ],
    principals=[
        '444455556666', # Change 444455556666 to the resource consumer account ID
    ],
    permissionArns = ["arn:aws:ram::aws:permission/
AWSRAMPermissionSageMakerCatalogResourceSearch"] #
```

```
 AWSRAMPermissionSageMakerCatalogResourceSearch is the only policy allowed for
 SageMaker Catalog
)
```

Kepala sekolah adalah aktor dalam sistem keamanan. Dalam kebijakan berbasis sumber daya, prinsipal yang diizinkan adalah pengguna IAM, peran IAM, akun root, atau layanan lain. AWS

Cari sumber daya yang dapat ditemukan

Akun pemilik sumber daya harus memberikan izin ke akun konsumen sumber daya agar dapat ditemukan atau akses (hanya-baca, tulis baca, atau admin) hak istimewa dengan sumber daya bersama. Di bagian berikut, kami memberikan petunjuk tentang cara menerima undangan ke sumber daya bersama dan contoh yang menunjukkan cara mencari grup fitur yang dapat ditemukan.

Terima undangan ke sumber daya bersama

Sebagai akun konsumen sumber daya, Anda menerima undangan untuk bergabung dengan pembagian sumber daya setelah akun pemilik sumber daya memberikan izin. Untuk menerima undangan ke sumber daya bersama, buka halaman Berbagi dengan saya: Berbagi sumber daya di AWS RAM konsol untuk melihat dan menanggapi undangan. Undangan tidak dikirim dalam kasus ini:

- Jika Anda adalah bagian dari sebuah organisasi di AWS Organizations dan berbagi di organisasi Anda diaktifkan, maka kepala sekolah di organisasi secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.
- Jika Anda berbagi dengan Akun AWS yang memiliki sumber daya, maka prinsipal di akun itu secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.

Untuk informasi selengkapnya tentang menerima dan menggunakan pembagian sumber dayaAWS RAM, lihat Menanggapi undangan berbagi sumber daya.

Cari contoh grup fitur yang dapat ditemukan

Setelah sumber daya dibagikan dengan akun konsumen sumber daya dengan izin dapat ditemukan diterapkan, akun konsumen sumber daya dapat mencari dan menemukan sumber daya bersama di Amazon SageMaker Feature Store menggunakan UI konsol dan Feature Store SDK. Perhatikan bahwa Anda tidak dapat mencari di tag untuk sumber daya lintas akun. Jumlah maksimum katalog grup fitur yang dapat dilihat adalah 1000. Untuk informasi selengkapnya tentang pemberian izin untuk ditemukan, lihat. Mengaktifkan kemampuan penemuan lintas akun

Untuk detail tentang melihat grup fitur bersama di konsol, lihatTemukan grup fitur di Toko Fitur.

Dalam contoh berikut, akun konsumen sumber daya menggunakan SageMaker pencarian untuk mencari sumber daya yang dapat ditemukan ketika `CrossAccountFilterOption` disetel ke: `"CrossAccount"`

```
from sagemaker.session import Session

sagemaker_session = Session(boto_session=boto_session)

sagemaker_session.search(
    resource="FeatureGroup",
    search_expression={
        "Filters": [
            {
                "Name": "FeatureGroupName",
                "Value": "MyFeatureGroup",
                "Operator": "Contains",
            }
        ],
        "Operator": "And",
    },
    sort_by="Name",
    sort_order="Ascending",
    next_token="token",
    max_results=50,
    CrossAccountFilterOption="CrossAccount"
)
```

Untuk informasi selengkapnya tentang SageMaker penelusuran dan parameter permintaan, lihat Pencarian di Referensi Amazon SageMaker API.

Mengaktifkan akses lintas akun

Izin akses adalah izin read-only, read-write, dan admin. Nama izin, deskripsi, dan daftar API tertentu yang tersedia untuk setiap izin tercantum di bawah ini:

- Izin hanya-baca (`AWSRAMPermissionFeatureGroupReadOnly`): Hak istimewa baca memungkinkan akun konsumen sumber daya membaca catatan dalam grup fitur bersama serta melihat detail dan metadata.
  - `DescribeFeatureGroup`: Mengambil detail tentang grup fitur dan konfigurasinya
  - `DescribeFeatureMetadata`: Menampilkan metadata untuk fitur dalam grup fitur
  - `BatchGetRecord`: Mengambil sekumpulan catatan dari grup fitur

  - `GetRecord`: Mengambil catatan dari grup fitur
- Izin baca-tulis (`AWSRAMPermissionSagemakerFeatureGroupReadWrite`): Hak istimewa baca-tulis memungkinkan akun konsumen sumber daya untuk menulis catatan, dan menghapus catatan dari, grup fitur bersama, selain izin baca.
  - `PutRecord`: Menulis catatan ke grup fitur
  - `DeleteRecord`: Menghapus catatan dari grup fitur
  - API terdaftar di `AWSRAMPermissionFeatureGroupReadOnly`
- Izin admin (`AWSRAMPermissionSagemakerFeatureGroupAdmin`): Hak istimewa admin memungkinkan akun konsumen sumber daya untuk memperbarui deskripsi dan parameter fitur dalam grup fitur bersama, memperbarui konfigurasi grup fitur bersama, selain izin baca-tulis.
  - `DescribeFeatureMetadata`: Menampilkan metadata untuk fitur dalam grup fitur
  - `UpdateFeatureGroup`: Memperbarui konfigurasi grup fitur
  - `UpdateFeatureMetadata`: Memperbarui deskripsi dan parameter fitur dalam grup fitur
  - API terdaftar di `AWSRAMPermissionSagemakerFeatureGroupReadWrite`

Dalam topik berikut, Anda dapat mempelajari cara berbagi toko online dan grup fitur offline — ada perbedaan antara keduanya dalam hal berbagi.

Topik

- Bagikan grup fitur online dengan AWS Resource Access Manager
- Akses toko offline lintas akun

Bagikan grup fitur online dengan AWS Resource Access Manager

Dengan AWS Resource Access Manager (AWS RAM) Anda dapat berbagi grup SageMaker fitur online Amazon Feature Store dengan aman dengan lainnyaAkun AWS. Anggota tim Anda dapat menjelajahi dan mengakses grup fitur yang menjangkau beberapa akun, mempromosikan konsistensi data, merampingkan kolaborasi, dan mengurangi duplikasi upaya.

Akun pemilik sumber daya dapat berbagi sumber daya dengan individu lain Akun AWS dengan memberikan izin menggunakan. AWS RAM Akun konsumen sumber daya adalah Akun AWS dengan siapa sumber daya dibagikan, dibatasi oleh izin yang diberikan dari akun pemilik sumber daya. Jika Anda adalah organisasi, Anda mungkin ingin memanfaatkannyaAWS Organizations, yang dengannya Anda dapat berbagi sumber daya dengan individuAkun AWS, dengan semua akun di

organisasi Anda, atau di Unit Organisasi (OU), tanpa harus menerapkan izin ke setiap akun. Untuk video instruksional dan informasi lebih lanjut tentang AWS RAM konsep dan manfaat, lihat Apa itu? AWS Resource Access Manager dalam AWS RAM User Guide.

Perhatikan bahwa ada batas maksimum lunak untuk transaksi per detik (TPS) per API perAkun AWS. Batas TPS maksimum berlaku untuk semua transaksi pada sumber daya dalam akun pemilik sumber daya, sehingga transaksi dari akun konsumen sumber daya juga diperhitungkan dalam batas maksimum ini. Untuk informasi tentang layanan dan cara meminta kenaikan kuota, lihat AWSService quotas.

Bagian ini mencakup cara akun pemilik sumber daya dapat memilih grup fitur dan memberikan hak akses (hanya-baca, baca tulis, dan admin) ke akun konsumen sumber daya, lalu bagaimana akun konsumen sumber daya dengan hak akses dapat menggunakan grup fitur tersebut. Izin akses tidak memungkinkan akun konsumen sumber daya untuk mencari dan menemukan grup fitur. Agar akun konsumen sumber daya dapat mencari dan menemukan grup fitur dari akun pemilik sumber daya, akun pemilik sumber daya harus memberikan izin untuk dapat ditemukan ke akun konsumen sumber daya, di mana semua grup fitur dalam akun pemilik sumber daya dapat ditemukan oleh akun konsumen sumber daya. Untuk informasi selengkapnya tentang pemberian izin untuk ditemukan, lihat. Mengaktifkan kemampuan penemuan lintas akun

Topik berikut menunjukkan cara membagikan sumber daya toko online Feature Store menggunakan AWS RAM konsol. Untuk informasi tentang berbagi sumber daya dan memberikan izin dalam AWS menggunakan AWS RAM konsol atau AWS Command Line Interface (AWS CLI), lihat Berbagi sumber daya Anda AWS.

Topik

- Bagikan entitas grup fitur Anda
- Gunakan sumber daya bersama toko online dengan izin akses

Bagikan entitas grup fitur Anda

Sebagai akun pemilik sumber daya, Anda dapat menggunakan jenis sumber daya grup fitur untuk Amazon SageMaker Feature Store untuk berbagi entitas grup fitur, dengan membuat pembagian sumber daya di AWS Resource Access Manager (AWS RAM).

Gunakan petunjuk berikut bersama dengan petunjuk Berbagi AWS sumber daya Anda di Panduan AWS RAM Pengguna.

Saat membagikan jenis sumber daya grup fitur menggunakan AWS RAM konsol, Anda perlu membuat pilihan berikut.

1. Tentukan detail berbagi sumber daya:

   - Jenis sumber daya: Pilih Grup SageMaker Fitur.
   - ARN: Pilih grup fitur ARN Anda dengan format:. `arn:aws:sagemaker:us-east-1:111122223333:feature-group/`*`your-feature-group-name`*

     `us-east-1`adalah wilayah sumber daya, `111122223333` adalah ID akun pemilik sumber daya, dan *`your-feature-group-name`* merupakan grup fitur yang Anda bagikan.
   - ID Sumber Daya: Pilih grup fitur*`your-feature-group-name`*, yang ingin Anda berikan izin akses.
2. Izin terkelola asosiasi:

   - Izin terkelola: Pilih izin akses. Untuk informasi selengkapnya tentang izin akses, lihat Mengaktifkan akses lintas akun.
3. Berikan akses ke kepala sekolah:

   - Pilih tipe utama (Akun AWS, Organisasi, Unit Organisasi, peran IAM, atau pengguna IAM) dan masukkan ID atau ARN yang sesuai.
4. Tinjau dan buat:

   - Tinjau lalu pilih Buat berbagi sumber daya.

Pemberian izin akses apa pun tidak memberikan izin untuk dapat ditemukan kepada akun konsumen sumber daya, sehingga akun konsumen sumber daya dengan izin akses tidak dapat mencari dan menemukan grup fitur tersebut. Agar akun konsumen sumber daya dapat mencari dan menemukan grup fitur dari akun pemilik sumber daya, akun pemilik sumber daya harus memberikan izin untuk dapat ditemukan ke akun konsumen sumber daya, di mana semua grup fitur dalam akun pemilik sumber daya dapat ditemukan oleh akun konsumen sumber daya. Untuk informasi selengkapnya tentang pemberian izin untuk ditemukan, lihat. Mengaktifkan kemampuan penemuan lintas akun

Jika akun konsumen sumber daya hanya diberikan izin akses, entitas grup fitur masih dapat dilihat. AWS RAM Untuk melihat sumber dayaAWS RAM, lihat Mengakses AWS sumber daya yang dibagikan dengan Anda di Panduan AWS RAM Pengguna.

Mungkin perlu waktu beberapa menit agar asosiasi berbagi sumber daya dan prinsipal, atau akun konsumen sumber daya, selesai diselesaikan. Setelah pembagian sumber daya dan asosiasi utama ditetapkan, akun konsumen sumber daya yang ditentukan menerima undangan untuk bergabung dengan pembagian sumber daya. Akun konsumen sumber daya dapat melihat dan menerima undangan dengan membuka halaman Shared with me: Resource shares di AWS RAM konsol. Undangan tidak dikirim dalam kasus ini:

- Jika Anda adalah bagian dari sebuah organisasi di AWS Organizations dan berbagi di organisasi Anda diaktifkan, maka kepala sekolah di organisasi secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.
- Jika Anda berbagi dengan Akun AWS yang memiliki sumber daya, maka prinsipal di akun itu secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.

Untuk informasi selengkapnya tentang menerima dan menggunakan pembagian sumber dayaAWS RAM, lihat Menggunakan AWS sumber daya bersama di Panduan AWS RAM Pengguna.

Bagikan grup fitur toko online menggunakan AWS SDK for Python (Boto3)

Anda dapat menggunakan AWS RAM API AWS SDK for Python (Boto3) for untuk membuat pembagian sumber daya. Kode berikut adalah contoh ID akun pemilik sumber daya yang `111122223333` membuat pembagian sumber daya bernama`'test-cross-account-fg'`, berbagi grup fitur bernama `'my-feature-group'` dengan ID akun konsumen sumber daya `444455556666` sambil memberikan `AWSRAMPermissionSageMakerFeatureGroupReadOnly` izin. Untuk informasi selengkapnya tentang izin akses, lihat Mengaktifkan akses lintas akun. Untuk menggunakan Python SDK untuk AWS RAM API, Anda harus melampirkan kebijakan terkelola akses AWS RAM penuh dengan peran eksekusi. Lihat create_resource_share API untuk detail AWS RAM selengkapnya.

```
import boto3

# Choose feature group name
feature_group_name = 'my-feature-group' # Change to your feature group name

# Share 'my-feature-group' with other account
ram_client = boto3.client("ram")
response = ram_client.create_resource_share(
    name='test-cross-account-fg', # Change to your custom resource share name
    resourceArns=[
```

```
        'arn:aws:sagemaker:us-east-1:111122223333:feature-group/' + feature_group_name,
 # Change 111122223333 to the resource owner account ID
    ],
    principals=[
        '444455556666', # Change 444455556666 to the resource consumer account ID
    ],
    permissionArns = ["arn:aws:ram::aws:permission/
AWSRAMPermissionSageMakerFeatureGroupReadOnly"]
)
```

Kepala sekolah adalah aktor dalam sistem keamanan. Dalam kebijakan berbasis sumber daya, prinsip yang diizinkan adalah pengguna IAM, peran IAM, akun root, atau lainnya. Layanan AWS

Gunakan sumber daya bersama toko online dengan izin akses

Akun pemilik sumber daya harus memberikan izin ke akun konsumen sumber daya untuk memungkinkan hak istimewa yang dapat ditemukan, hanya-baca, tulis, atau admin dengan sumber daya bersama. Di bagian berikut, kami memberikan petunjuk tentang cara menerima undangan untuk mengakses sumber daya bersama dan memberikan contoh yang menunjukkan cara melihat dan berinteraksi dengan grup fitur bersama.

Menerima undangan untuk mengakses sumber daya bersama menggunakan AWS RAM

Sebagai akun konsumen sumber daya, Anda akan menerima undangan untuk bergabung dengan pembagian sumber daya setelah akun pemilik sumber daya memberikan izin. Untuk menerima undangan ke sumber daya bersama, buka halaman Berbagi dengan saya: Berbagi sumber daya di AWS RAM konsol untuk melihat dan menanggapi undangan. Undangan tidak dikirim dalam kasus ini:

- Jika Anda adalah bagian dari sebuah organisasi di AWS Organizations dan berbagi di organisasi Anda diaktifkan, maka kepala sekolah di organisasi secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.
- Jika Anda berbagi dengan Akun AWS yang memiliki sumber daya, maka prinsipal di akun itu secara otomatis mendapatkan akses ke sumber daya bersama tanpa undangan.

Untuk informasi selengkapnya tentang menerima dan menggunakan pembagian sumber dayaAWS RAM, lihat Menggunakan AWS sumber daya bersama di Panduan AWS RAM Pengguna.

Melihat sumber daya bersama di AWS RAM konsol

Pemberian izin akses apa pun tidak memberikan izin untuk dapat ditemukan kepada akun konsumen sumber daya, sehingga akun konsumen sumber daya dengan izin akses tidak dapat mencari dan

menemukan grup fitur tersebut. Agar akun konsumen sumber daya dapat mencari dan menemukan grup fitur dari akun pemilik sumber daya, akun pemilik sumber daya harus memberikan izin untuk dapat ditemukan ke akun konsumen sumber daya, di mana semua grup fitur dalam akun pemilik sumber daya dapat ditemukan oleh akun konsumen sumber daya. Untuk informasi selengkapnya tentang pemberian izin untuk ditemukan, lihat. Mengaktifkan kemampuan penemuan lintas akun

Untuk melihat sumber daya bersama di AWS RAM konsol, buka halaman Shared with me: Resource shares di AWS RAM konsol.

Membaca dan menulis tindakan dengan contoh grup fitur bersama

Setelah akun konsumen sumber daya Anda diberikan izin yang sesuai oleh akun pemilik sumber daya, Anda dapat melakukan tindakan pada sumber daya bersama menggunakan Feature Store SDK. Anda dapat melakukan ini dengan menyediakan sumber daya ARN sebagai. `FeatureGroupName` Untuk mendapatkan ARN Grup Fitur, Anda dapat menggunakan AWS SDK for Python (Boto3) `DescribeFeatureGroup`fungsi atau menggunakan UI konsol. Untuk informasi tentang penggunaan UI konsol tersebut untuk melihat detail grup fitur, lihatLihat detail grup fitur dari konsol.

Contoh berikut menggunakan `PutRecord` dan `GetRecord` dengan entitas grup fitur bersama. Lihat sintaks permintaan dan respons dalam AWS SDK for Python (Boto3) dokumentasi untuk `PutRecord`dan `GetRecordAPIs`.

```
import boto3

sagemaker_featurestore_runtime = boto3.client('sagemaker-featurestore-runtime')

# Put record into feature group named 'test-fg' within the resource owner account ID
 111122223333
featurestore_runtime.put_record(
    FeatureGroupName="arn:aws:sagemaker:us-east-1:111122223333:feature-group/test-fg",
    Record=[value.to_dict() for value in record] # You will need to define record prior
 to calling PutRecord
)
```

```
import boto3

sagemaker_featurestore_runtime = boto3.client('sagemaker-featurestore-runtime')

# Choose record identifier
record_identifier_value = str(2990130)
```

```
# Get record from feature group named 'test-fg' within the resource owner account ID
 111122223333
featurestore_runtime.get_record(
    FeatureGroupName="arn:aws:sagemaker:us-east-1:111122223333:feature-group/test-fg",
    RecordIdentifierValueAsString=record_identifier_value
)
```

Untuk informasi selengkapnya tentang pemberian izin ke entitas grup fitur, lihat. Bagikan entitas grup fitur Anda

Akses toko offline lintas akun

Amazon SageMaker Feature Store memungkinkan pengguna untuk membuat grup fitur dalam satu akun (Akun A) dan mengonfigurasinya dengan toko offline menggunakan bucket Amazon S3 di akun lain (Akun B). Anda dapat mengatur ini menggunakan langkah-langkah di bagian berikut.

Topik

- Langkah 1: Mengatur peran akses toko offline di Akun A
- Langkah 2: Siapkan bucket Amazon S3 toko offline di Akun B
- Langkah 3: Mengatur kunci AWS KMS enkripsi toko offline di Akun A
- Langkah 4: Membuat grup fitur di Akun A

Langkah 1: Mengatur peran akses toko offline di Akun A

Pertama, siapkan peran Amazon SageMaker Feature Store untuk menulis data ke toko offline. Cara termudah untuk mencapainya adalah dengan membuat peran baru menggunakan `AmazonSageMakerFeatureStoreAccess` kebijakan atau menggunakan peran yang ada yang sudah memiliki `AmazonSageMakerFeatureStoreAccess` kebijakan terlampir. Dokumen ini mengacu pada kebijakan ini sebagai`Account-A-Offline-Feature-Store-Role-ARN`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        }
    ]
}
```

Cuplikan kode sebelumnya menunjukkan kebijakan. `AmazonSageMakerFeatureStoreAccess` `Resource`Bagian kebijakan dicakup secara default ke bucket S3 dengan nama yang berisi`SageMaker`,, atau. `Sagemaker` `sagemaker` Ini berarti bucket Amazon S3 toko offline yang digunakan harus mengikuti konvensi penamaan ini. Jika ini bukan kasus Anda, atau jika Anda ingin memperluas cakupan sumber daya, Anda dapat menyalin dan menempelkan kebijakan tersebut ke kebijakan bucket Amazon S3 di konsol, menyesuaikan `Resource` bagiannya`arn:aws:s3:::your-offline-store-bucket-name`, lalu melampirkan peran tersebut.

Selain itu, peran ini harus memiliki AWS KMS izin yang dilampirkan. Minimal, diperlukan `kms:GenerateDataKey` izin untuk dapat menulis ke toko offline menggunakan kunci yang dikelola pelanggan Anda. Lihat Langkah 3 untuk mempelajari mengapa kunci yang dikelola pelanggan diperlukan untuk skenario lintas akun dan cara mengaturnya. Berikut adalah contoh kebijakan inline:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:*:Account-A-Account-Id:key/*"
        }
    ]
}
```

`Resource`Bagian dari kebijakan ini dicakup ke kunci apa pun di Akun A. Untuk lebih lanjut menjelaskan hal ini, setelah menyiapkan kunci KMS toko offline di Langkah 3, kembali ke kebijakan ini dan ganti dengan kunci ARN.

Langkah 2: Siapkan bucket Amazon S3 toko offline di Akun B

Buat bucket Amazon S3 di Akun B. Jika Anda menggunakan `AmazonSageMakerFeatureStoreAccess` kebijakan default, nama bucket harus menyertakan`SageMaker`,`Sagemaker`, atau. `sagemaker` Edit kebijakan bucket seperti yang ditunjukkan pada contoh berikut untuk memungkinkan Akun A membaca dan menulis objek.

Dokumen ini mengacu pada contoh kebijakan bucket berikut sebagai`Account-B-Offline-Feature-Store-Bucket`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3CrossAccountBucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetBucketAcl"
            ],
            "Principal": {
                "AWS": [
                    "*Account-A-Offline-Feature-Store-Role-ARN*"
                ],
            },
            "Resource": [
                "arn:aws:s3:::offline-store-bucket-name/*",
                "arn:aws:s3:::offline-store-bucket-name"
            ]
        }
    ]
}
```

Dalam kebijakan sebelumnya, prinsipnya adalah, yang merupakan `Account-A-Offline-Feature-Store-Role-ARN` peran yang dibuat di Akun A pada Langkah 1 dan diberikan ke Amazon SageMaker Feature Store untuk menulis ke toko offline. Anda dapat memberikan beberapa peran ARN di bawah. `Principal`

Langkah 3: Mengatur kunci AWS KMS enkripsi toko offline di Akun A

Amazon SageMaker Feature Store memastikan bahwa enkripsi sisi server selalu diaktifkan untuk objek Amazon S3 di toko offline. Untuk kasus penggunaan lintas akun, Anda harus memberikan kunci yang dikelola pelanggan sehingga Anda mengendalikan siapa yang dapat menulis ke toko offline (dalam hal ini, `Account-A-Offline-Feature-Store-Role-ARN` dari Akun A) dan siapa yang dapat membaca dari toko offline (dalam hal ini, identitas dari Akun B).

Dokumen ini mengacu pada contoh kebijakan kunci berikut sebagai`Account-A-Offline-Feature-Store-KMS-Key-ARN`.

```
{
    "Version": "2012-10-17",
    "Id": "key-consolepolicy-3",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Account-A-Account-Id:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::Account-A-Account-Id:role/Administrator",
                ]
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
```

```
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow Feature Store to get information about the customer managed
key",
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker.amazonaws.com"
            },
            "Action": [
                "kms:Describe*",
                "kms:Get*",
                "kms:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "*Account-A-Offline-Feature-Store-Role-ARN*",
                    "*arn:aws:iam::Account-B-Account-Id:root*"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:RetireGrant",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:ListAliases",
                "kms:ListGrants"
            ],
            "Resource": "*",
        }
```

```
    ]
}
```

Langkah 4: Membuat grup fitur di Akun A

Selanjutnya, buat grup fitur di Akun A, dengan bucket Amazon S3 toko offline di Akun B. Untuk melakukan ini, berikan parameter berikut untuk`RoleArn`,, dan `OfflineStoreConfig.S3StorageConfig.KmsKeyIdOfflineStoreConfig.S3StorageConfig.S3` masing-masing:

- Berikan `Account-A-Offline-Feature-Store-Role-ARN` sebagai`RoleArn`.
- Menyediakan `Account-A-Offline-Feature-Store-KMS-Key-ARN` untuk`OfflineStoreConfig.S3StorageConfig.KmsKeyId`.
- Menyediakan `Account-B-Offline-Feature-Store-Bucket` untuk`OfflineStoreConfig.S3StorageConfig.S3Uri`.

Konfigurasi penyimpanan Feature Store

Amazon SageMaker Feature Store terdiri dari toko online dan toko offline. Toko online memungkinkan pencarian fitur real-time untuk inferensi, sedangkan toko offline berisi data historis untuk pelatihan model dan inferensi batch. Saat membuat grup fitur, Anda memiliki opsi untuk mengaktifkan toko online, toko offline, atau keduanya. Saat Anda mengaktifkan keduanya, keduanya disinkronkan untuk menghindari perbedaan antara data pelatihan dan penyajian. Untuk informasi selengkapnya tentang toko online dan offline serta konsep Toko Fitur lainnya, lihatKonsep Feature Store.

Topik berikut membahas jenis penyimpanan toko online dan format tabel toko offline.

Topik

- Fungsi SQL baru
- Fungsi SQL baru
- Pengenal pasti

Fungsi SQL baru

Toko online adalah toko data latensi rendah dan ketersediaan tinggi yang menyediakan pencarian fitur secara real-time. Hal ini biasanya digunakan untuk penyajian model machine learning (ML).

Anda dapat memilih antara toko online standar (`Standard`) atau toko online tingkat dalam memori (`InMemory`), pada saat Anda membuat grup fitur. Dengan cara ini, Anda dapat memilih jenis penyimpanan yang paling cocok dengan pola baca dan tulis untuk aplikasi tertentu, sambil mempertimbangkan kinerja dan biaya. Untuk detail lebih lanjut tentang harga, lihat SageMaker Harga Amazon.

Toko online berisi `StorageType` opsi berikut. Untuk informasi lebih lanjut tentang konten toko online, lihat `OnlineStoreConfig`.

Jenis penyimpanan tingkat standar

`Standard`Tingkat adalah penyimpanan data latensi rendah terkelola untuk grup fitur toko online. Ini menyediakan pengambilan data cepat untuk layanan model ML untuk aplikasi Anda. `Standard`adalah jenis penyimpanan default.

Jenis penyimpanan tingkat dalam memori

`InMemory`Tingkat adalah penyimpanan data terkelola untuk grup fitur toko online yang mendukung pengambilan latensi sangat rendah. Ini menyediakan pengambilan data real-time skala besar untuk penyajian model ML yang digunakan untuk aplikasi throughput tinggi. `InMemory`Tingkat ini didukung oleh Amazon ElastiCache untuk Redis. Untuk informasi selengkapnya, lihat Apa itu Amazon ElastiCache untuk Redis? .

`InMemory`Tingkat toko online mendukung jenis koleksi, yaitu daftar, set, dan vektor. Untuk informasi selengkapnya tentang jenis `InMemory` koleksi, lihatJenis koleksi.

Feature Store menyediakan latensi rendah membaca dan menulis ke toko online. Latensi aplikasi terutama terdiri dari dua komponen utama: infrastruktur atau latensi jaringan dan latensi API Feature Store. Pengurangan latensi jaringan membantu mendapatkan pembacaan dan penulisan latensi terendah ke Feature Store. Anda dapat mengurangi latensi jaringan ke Feature Store dengan menerapkan AWS PrivateLink ke titik akhir Feature Store Runtime. DenganAWS PrivateLink, Anda dapat mengakses secara pribadi semua operasi API Runtime Store Fitur dari Amazon Virtual Private Cloud (VPC) Anda dengan cara yang dapat diskalakan dengan menggunakan titik akhir VPC antarmuka. AWS PrivateLinkPenerapan dengan `privateDNSEnabled` opsi ditetapkan sebagai true:

- Itu membuat semua lalu lintas baca/tulis Toko Fitur dalam VPC Anda.
- Itu membuat lalu lintas di AZ yang sama dengan klien yang berasal saat menggunakan Feature Store. Ini menghindari “lompatan” antara AZ yang mengurangi latensi jaringan.

Ikuti langkah-langkah di Akses AWS layanan menggunakan titik akhir VPC antarmuka untuk penyiapan AWS PrivateLink ke Feature Store. Nama layanan untuk Feature Store Runtime in AWS PrivateLink adalah`com.amazonaws.region.sagemaker.featurestore-runtime`.

Skala toko online `InMemory` tingkat secara otomatis berdasarkan penggunaan dan permintaan penyimpanan. Penskalaan otomatis dapat memakan waktu beberapa menit untuk beradaptasi dengan pola penggunaan baru jika berubah dengan cepat. Selama penskalaan otomatis:

- Operasi tulis ke grup fitur mungkin menerima kesalahan pelambatan. Anda harus mencoba kembali permintaan Anda beberapa menit kemudian.
- Operasi baca ke grup fitur mungkin menerima kesalahan pelambatan. Strategi coba lagi standar cocok dalam kasus ini.
- Operasi baca mungkin melihat peningkatan latensi.

Ukuran maksimum grup fitur `InMemory` tingkat default adalah 50 GiB.

Perhatikan bahwa `InMemory` tingkat saat ini hanya mendukung grup fitur online, bukan grup fitur online+offline, jadi tidak ada replikasi antara toko online dan offline untuk tingkat tersebut. `InMemory` Selain itu, `InMemory` tingkat saat ini tidak mendukung kunci KMS yang dikelola pelanggan.

Fungsi SQL baru

Toko offline digunakan untuk data historis ketika pengambilan sub-detik tidak diperlukan. Ini biasanya digunakan untuk eksplorasi data, pelatihan model, dan inferensi batch.

Saat Anda mengaktifkan toko online dan offline untuk grup fitur Anda, keduanya menyimpan sinkronisasi untuk menghindari perbedaan antara data pelatihan dan penyajian. Harap dicatat bahwa grup fitur toko online dengan jenis `InMemory` penyimpanan diaktifkan saat ini tidak mendukung grup fitur yang sesuai di toko offline (tidak ada replikasi online ke offline). Untuk informasi selengkapnya tentang penyajian model ML di Amazon SageMaker Feature Store, lihatFungsi SQL baru.

Toko offline berisi `TableFormat` opsi berikut. Untuk informasi tentang konten toko offline, lihat `OfflineStoreConfig`di Referensi Amazon SageMaker API.

Format tabel Glue

`Glue`Format (default) adalah format tabel tipe Hive standar untukAWS Glue. DenganAWS Glue, Anda dapat menemukan, menyiapkan, memindahkan, dan mengintegrasikan data dari berbagai

sumber. Ini juga mencakup produktivitas tambahan dan perkakas operasi data untuk menulis, menjalankan pekerjaan, dan mengimplementasikan alur kerja bisnis. Untuk informasi lebih lanjut tentangAWS Glue, lihat Apa ituAWS Glue? .

Format tabel gunung es

`Iceberg`Format (disarankan) adalah format tabel terbuka untuk tabel analitik yang sangat besar. Dengan`Iceberg`, Anda dapat memadatkan file data kecil menjadi lebih sedikit file besar di partisi, menghasilkan kueri yang jauh lebih cepat. Operasi pemadatan ini bersamaan dan tidak memengaruhi operasi baca dan tulis yang sedang berlangsung pada grup fitur. Untuk informasi selengkapnya tentang pemadatan, lihat Mengoptimalkan tabel Gunung Es di Panduan Pengguna Amazon Athena.

`Iceberg`mengelola koleksi besar file sebagai tabel dan mendukung operasi danau data analitik modern. Jika Anda memilih `Iceberg` opsi saat membuat grup fitur baru, Amazon SageMaker Feature Store membuat `Iceberg` tabel menggunakan format file Parket, dan mendaftarkan tabel dengan file. AWS Glue Data Catalog Untuk informasi selengkapnya tentang format `Iceberg` tabel, lihat Menggunakan tabel Apache Iceberg.

> **Important**
>
> Perhatikan bahwa untuk grup fitur dalam format `Iceberg` tabel, Anda harus menentukan `String` sebagai jenis fitur untuk waktu acara. Jika Anda menentukan jenis lainnya, Anda tidak dapat membuat grup fitur dengan sukses.

Pengenal pasti

Amazon SageMaker Feature Store menyediakan dua model harga untuk dipilih: mode throughput on-demand (`On-demand`) dan provisioned (`Provisioned`). `On-demand`bekerja paling baik untuk lalu lintas yang kurang dapat diprediksi, sementara `Provisioned` bekerja paling baik untuk lalu lintas yang konsisten dan dapat diprediksi.

Anda memiliki opsi untuk beralih antara `On-demand` dan mode `Provisioned` throughput untuk grup fitur tertentu, untuk mengakomodasi periode di mana pola lalu lintas aplikasi berubah atau kurang dapat diprediksi. Anda hanya dapat memperbarui mode throughput grup fitur Anda menjadi `On-demand` sekali dalam periode 24 jam. Mode throughput dapat diperbarui secara terprogram menggunakan UpdateFeatureGroupAPI atau melalui UI konsol. Untuk informasi selengkapnya tentang penggunaan konsol, lihatMenggunakan Amazon SageMaker Feature Store di konsol.

Anda dapat menggunakan mode `Provisioned` throughput dengan grup fitur khusus offline atau grup fitur dengan jenis penyimpanan. `Standard` Untuk konfigurasi penyimpanan lainnya, mode `On-demand` throughput digunakan. Untuk informasi tentang konfigurasi penyimpanan online dan offline, lihat Fungsi SQL baru danFungsi SQL baru, masing-masing.

Untuk detail selengkapnya tentang harga, lihat SageMaker Harga Amazon.

Topik

- Mode throughput sesuai permintaan
- Mode throughput yang disediakan
- Metrik mode Throughput
- Batas mode throughput

Mode throughput sesuai permintaan

Mode throughput `On-demand` (default) berfungsi paling baik saat Anda menggunakan grup fitur dengan beban kerja yang tidak diketahui, lalu lintas aplikasi yang tidak dapat diprediksi, dan Anda tidak dapat memperkirakan persyaratan kapasitas.

`On-demand`Mode ini menagih biaya untuk membaca dan tulis yang dilakukan aplikasi Anda pada grup fitur Anda. Anda tidak perlu menentukan berapa banyak throughput baca dan tulis yang Anda harapkan untuk dilakukan aplikasi Anda karena Feature Store langsung mengakomodasi beban kerja Anda saat naik atau turun. Anda hanya membayar untuk apa yang Anda gunakan, yang diukur dalam `ReadRequestsUnits` dan`WriteRequestsUnits`.

Anda dapat mengaktifkan mode `On-demand` throughput menggunakan UpdateFeatureGroupAPI CreateFeatureGroupatau melalui UI konsol. Untuk informasi selengkapnya tentang penggunaan UI konsol, lihatMenggunakan Amazon SageMaker Feature Store di konsol.

> **Important**
>
> Anda hanya dapat memperbarui mode throughput grup fitur Anda menjadi `On-demand` sekali dalam periode 24 jam.

Mode throughput yang disediakan

Mode `Provisioned` throughput bekerja paling baik saat Anda menggunakan grup fitur dengan beban kerja yang dapat diprediksi dan Anda dapat memperkirakan persyaratan kapasitas untuk mengontrol biaya. Ini dapat membuatnya lebih hemat biaya untuk beban kerja tertentu di mana Anda dapat mengantisipasi persyaratan throughput sebelumnya.

Saat Anda mengatur grup fitur ke `Provisioned` mode, tentukan unit kapasitas yang merupakan jumlah maksimum kapasitas yang dapat dikonsumsi aplikasi dari grup fitur. Jika aplikasi Anda melebihi kapasitas `Provisioned` throughput ini, aplikasi tunduk pada permintaan throttling.

Berikut ini mencakup informasi tentang unit kapasitas baca dan tulis.

- Mengambil satu record hingga 4 KB menggunakan `GetRecord` API akan mengkonsumsi setidaknya 1 RCU (unit kapasitas baca). Mengambil muatan yang lebih besar mungkin membutuhkan lebih banyak. Jumlah total unit kapasitas baca yang diperlukan tergantung pada ukuran item, termasuk metadata per catatan kecil yang ditambahkan oleh layanan Feature Store.
- Permintaan tulis tunggal dengan muatan 1 KB menggunakan `PutRecord` API akan menggunakan setidaknya 1 WCU (unit kapasitas tulis), dengan muatan pecahan dibulatkan ke KB terdekat. Ini mungkin mengkonsumsi lebih banyak tergantung pada waktu acara, status penghapusan catatan, dan status time to live (TTL). Untuk informasi selengkapnya tentang TTL, lihat[Durasi waktu untuk tayang (TTL) untuk rekaman](#).

> **Important**
>
> Saat mengatur unit kapasitas Anda, pertimbangkan hal berikut:
>
> - Anda akan dikenakan biaya untuk kapasitas baca dan tulis yang Anda berikan untuk grup fitur Anda, bahkan jika Anda tidak sepenuhnya memanfaatkan kapasitas tersebut`Provisioned`.
> - Jika Anda menyetel kapasitas baca atau tulis terlalu rendah, permintaan Anda mungkin mengalami pembatasan.
> - Dalam beberapa kasus, catatan dapat menggunakan unit kapasitas tambahan karena metadata tingkat rekaman yang ditambahkan oleh layanan Feature Store untuk mengaktifkan berbagai fitur.
> - Mengambil hanya sebagian fitur yang menggunakan `GetRecord` atau `BatchGetRecord` API akan tetap menggunakan RCU yang sesuai dengan seluruh catatan.

- Untuk kapasitas tulis, Anda harus menyediakan 2x kapasitas puncak baru-baru ini untuk menghindari pelambatan saat melakukan pengisian ulang atau konsumsi massal yang dapat menghasilkan sejumlah besar penulisan catatan sejarah. Ini karena menulis catatan sejarah menghabiskan kapasitas menulis tambahan.
- Toko Fitur saat ini tidak mendukung penskalaan otomatis untuk `Provisioned` mode.

Anda dapat mengaktifkan mode `On-demand` throughput menggunakan UpdateFeatureGroupAPI CreateFeatureGroupatau melalui UI konsol. Untuk informasi selengkapnya tentang penggunaan UI konsol, lihatMenggunakan Amazon SageMaker Feature Store di konsol.

Berikut ini menjelaskan bagaimana Anda dapat menambah atau mengurangi throughput RCU dan WCU untuk grup fitur Anda saat `Provisioned` mode diaktifkan.

Meningkatkan throughput yang disediakan

Anda dapat meningkatkan RCU atau WCU sesering yang diperlukan menggunakan UpdateFeatureGroupAPI atau UI konsol.

Mengurangi throughput yang disediakan

Anda dapat mengurangi RCU dan WCU (atau keduanya) untuk grup fitur yang menggunakan UpdateFeatureGroupAPI atau UI konsol.

Ada kuota default pada jumlah penurunan `Provisioned` kapasitas yang dapat Anda lakukan pada grup fitur Anda per hari. Satu hari ditentukan berdasarkan Waktu Universal Terkoordinasi (UTC). Pada hari tertentu, Anda dapat memulai dengan melakukan hingga empat penurunan dalam satu jam selama Anda belum melakukan penurunan lainnya pada hari tersebut. Selanjutnya, Anda dapat melakukan satu penurunan tambahan per jam selama tidak ada penurunan pada jam sebelumnya. Hal ini secara efektif menjadikan jumlah maksimum penurunan dalam sehari menjadi 27 kali (4 penurunan dalam satu jam pertama, dan 1 penurunan untuk masing-masing jendela 1 jam berikutnya dalam sehari).

Metrik mode Throughput

Grup fitur dalam `On-demand` mode akan memancarkan `ConsumedReadRequestsUnits` dan `ConsumedWriteRequestsUnits` metrik. Grup fitur dalam `Provisioned` mode akan memancarkan `ConsumedReadCapacityUnits` dan `ConsumedWriteCapacityUnits` metrik. Untuk informasi selengkapnya tentang metrik Toko Fitur, lihatMetrik Toko SageMaker Fitur Amazon.

Batas mode throughput

Masing-masing Akun AWS memiliki kuota atau batasan layanan default yang diterapkan untuk membantu memastikan ketersediaan dan mengelola risiko penagihan. Untuk informasi tentang kuota dan batas default, lihatKuota, aturan penamaan, dan tipe data.

Dalam beberapa kasus, batasan ini mungkin lebih rendah dari yang dinyatakan dalam dokumentasi. Jika Anda membutuhkan batas yang lebih tinggi, Anda dapat mengajukan permintaan kenaikan. Sebaiknya lakukan sebelum mencapai batas saat ini untuk menghindari gangguan pada pekerjaan Anda. Untuk informasi tentang layanan dan cara meminta kenaikan kuota, lihat AWSService quotas.

Jenis koleksi

Jenis koleksi menyediakan cara untuk mengatur dan menyusun data untuk pengambilan dan analisis yang efisien. Mereka digunakan dalam database ML untuk menentukan skema dataset dan elemen-elemennya. Di Amazon SageMaker Feature Store, jenis koleksi yang didukung mencakup daftar, set, dan vektor.

Koleksi adalah pengelompokan elemen di mana setiap elemen dalam koleksi harus memiliki jenis fitur yang sama (`String`,`Integral`, atau`Fractional`). Misalnya, koleksi dapat berisi elemen dengan semua jenis fitur elemen sebagai`Fractional`, tetapi koleksi tidak dapat berisi elemen dengan beberapa jenis fitur sebagai `Fractional` dan beberapa jenis fitur sebagai`String`.

Hanya grup fitur toko `InMemory` online yang saat ini mendukung jenis koleksi. Daftar berikut menjelaskan opsi tipe koleksi.

Daftar: Kumpulan elemen yang diurutkan.

- Panjang daftar ditentukan oleh berapa banyak elemen dalam koleksi.
- Contoh: Anda dapat memiliki daftar seperti ['a', 'b', 'a'], karena daftar mempertahankan urutan dan dapat memiliki elemen pengulangan.

Set: Koleksi elemen unik yang tidak berurutan.

- Panjang set ditentukan oleh berapa banyak elemen unik dalam koleksi.
- Contoh: Anda tidak dapat memiliki himpunan seperti ['a', 'b', 'a'], karena mengandung elemen pengulangan. Himpunan akan memiliki elemen ['a', 'b'], karena himpunan hanya berisi elemen unik.

Vektor: Daftar khusus yang mewakili array elemen ukuran tetap. Urutan elemen memiliki signifikansi, sehingga posisi elemen mewakili properti tertentu dari data.

- Elemen-elemen dalam jenis koleksi vektor harus memiliki tipe `Fractional` fitur.
- Anda mungkin hanya memiliki satu jenis koleksi vektor per grup fitur `InMemory` tingkat toko online.
- Dimensi (jumlah elemen dalam vektor) vektor ditentukan sebelumnya oleh Anda dan ditentukan menggunakan`VectorDimension`. Batas dimensi maksimal adalah 8192.
- Contoh: Anda dapat memiliki vektor seperti [4.2, -6.3, 4.2], di mana elemen pertama, kedua, dan ketiga dapat mewakili posisi x, y, dan z dalam ruang fisik.

Tidak ada batasan panjang koleksi, asalkan tidak melebihi ukuran maksimum rekaman. Untuk ukuran maksimum rekaman, lihatKuota, aturan penamaan, dan tipe data.

Menambahkan fitur dan catatan ke grup fitur

Anda dapat menggunakan Amazon SageMaker Feature Store API atau konsol untuk memperbarui dan mendeskripsikan grup fitur serta menambahkan fitur dan catatan ke grup fitur Anda. Grup fitur adalah objek yang berisi data Anda dan fitur menjelaskan kolom dalam tabel. Saat Anda menambahkan fitur ke grup fitur, Anda secara efektif menambahkan kolom ke tabel. Saat Anda menambahkan catatan baru ke grup fitur, Anda mengisi nilai untuk fitur yang terkait dengan pengenal rekaman tertentu. Untuk informasi selengkapnya tentang konsep Toko Fitur, lihatKonsep Feature Store.

Setelah berhasil menambahkan fitur ke grup fitur, Anda tidak dapat menghapus fitur tersebut. Fitur yang telah Anda tambahkan tidak menambahkan data apa pun ke catatan Anda. Anda dapat menambahkan catatan baru ke grup fitur atau menimpa mereka menggunakan PutRecordAPI. Untuk contoh tentang memperbarui, mendeskripsikan, dan menempatkan catatan ke dalam grup fitur, lihatContoh kode.

Anda dapat menggunakan konsol untuk menambahkan fitur ke grup fitur. Untuk informasi selengkapnya tentang cara memperbarui grup fitur menggunakan konsol, lihatMemperbarui grup fitur dari konsol.

Bagian berikut memberikan ikhtisar penggunaan Feature Store API untuk menambahkan fitur ke grup fitur diikuti dengan contoh. Dengan API, Anda juga dapat menambahkan atau menimpa catatan setelah memperbarui grup fitur.

Topik

API

Gunakan `UpdateFeatureGroup`operasi untuk menambahkan fitur ke grup fitur.

Anda dapat menggunakan `DescribeFeatureGroup`operasi untuk melihat apakah Anda telah berhasil menambahkan fitur.

Untuk menambah atau menimpa catatan, gunakan `PutRecord`operasi.

Untuk melihat pembaruan yang telah Anda buat pada rekaman, gunakan `GetRecord`operasi. Untuk melihat pembaruan yang telah Anda buat pada beberapa catatan, gunakan `BatchGetRecord`operasi. Ini dapat memakan waktu hingga lima menit hingga pembaruan yang Anda buat muncul.

Anda dapat menggunakan kode contoh di bagian berikut untuk berjalan melalui penambahan fitur dan catatan menggunakanAWS SDK for Python (Boto3).

Contoh kode

Kode contoh memandu Anda melalui proses berikut:

1. Menambahkan fitur ke grup fitur
2. Memverifikasi bahwa Anda telah berhasil menambahkannya
3. Menambahkan catatan ke grup fitur
4. Memverifikasi bahwa Anda telah menambahkannya dengan sukses

Langkah 1: Menambahkan fitur ke grup fitur

Kode berikut menggunakan `UpdateFeatureGroup`operasi untuk menambahkan fitur baru ke grup fitur. Ini mengasumsikan bahwa Anda telah menyiapkan Feature Store dan membuat grup fitur. Untuk informasi selengkapnya tentang memulai, lihatPengantar buku catatan contoh Toko Fitur.

```
import boto3

sagemaker_client = boto3.client("sagemaker")
```

```
sagemaker_client.update_feature_group(
    FeatureGroupName=feature_group_name,
    FeatureAdditions=[
        {"FeatureName": "new-feature-1", "FeatureType": "Integral"},
        {"FeatureName": "new-feature-2", "FeatureType": "Fractional"},
        {"FeatureName": "new-feature-3", "FeatureType": "String"}
    ]
)
```

Kode berikut menggunakan DescribeFeatureGroupoperasi untuk memeriksa status pembaruan. Jika LastUpdateStatusbidangnya`Successful`, Anda telah berhasil menambahkan fitur.

```
sagemaker_client.describe_feature_group(
    FeatureGroupName=feature_group_name
)
```

Langkah 2: Menambahkan catatan baru ke grup fitur

Kode berikut menggunakan PutRecordoperasi untuk menambahkan catatan ke grup fitur yang telah Anda buat.

```
record_identifier_value = 'new_record'

sagemaker_featurestore_runtime_client = boto3.client("sagemaker-featurestore-runtime")

sagemaker_runtime_client.put_record(
    FeatureGroupName=feature_group_name,
    Record=[
        {
            'FeatureName': "record-identifier-feature-name",
            'ValueAsString': record_identifier_value
        },
        {
            'FeatureName': "event-time-feature",
            'ValueAsString': "timestamp-that-feature-store-returns"
        },
        {
```

```
            'FeatureName': "new-feature-1",
            'ValueAsString': "value-as-string"
        },
        {
            'FeatureName': "new-feature-2",
            'ValueAsString': "value-as-string"
        },
        {
            'FeatureName': "new-feature-3",
            'ValueAsString': "value-as-string"
        },
    ]
)
```

Gunakan GetRecordoperasi untuk melihat rekaman mana di grup fitur Anda yang tidak memiliki data untuk fitur yang telah Anda tambahkan. Anda dapat menggunakan PutRecordoperasi untuk menimpa catatan yang tidak memiliki data untuk fitur yang telah Anda tambahkan.

Temukan fitur di grup fitur Anda

Dengan Amazon SageMaker Feature Store, Anda dapat mencari fitur yang Anda buat di grup fitur Anda. Anda dapat mencari melalui semua fitur Anda tanpa perlu memilih grup fitur terlebih dahulu. Fungsionalitas pencarian membantu menemukan fitur yang relevan dengan kasus penggunaan Anda.

> **Note**
>
> Grup fitur tempat Anda mencari fitur harus ada di dalam Wilayah AWS danAkun AWS. Untuk grup fitur bersama, grup fitur harus dibuat dapat ditemukan oleh Anda. Akun AWS Untuk petunjuk selengkapnya tentang cara membagikan katalog grup fitur dan memberikan kemampuan untuk ditemukan, lihat. Bagikan katalog grup fitur Anda

Jika Anda berada di tim, dan rekan tim mencari fitur untuk digunakan dalam model mereka, mereka dapat mencari melalui fitur di semua grup fitur.

Anda dapat menambahkan parameter dan deskripsi yang dapat dicari untuk membuat fitur Anda lebih mudah ditemukan. Untuk informasi selengkapnya, lihat Menambahkan metadata yang dapat dicari ke fitur Anda.

Anda dapat mencari fitur menggunakan konsol atau dengan menggunakan operasi `Search`API di SageMaker. Tabel berikut mencantumkan semua metadata yang dapat dicari dan apakah Anda dapat mencarinya di konsol atau dengan API.

Pengenal metadata	Fungsi JSON	Dapat dicari di konsol?
Fungsi SQL baru	AllParameters	Ya
Waktu pembuatan	CreationTime	Ya
Deskripsi	Deskripsi	Ya
Nama grup	FeatureGroupName	Tidak
Nama	FeatureName	Ya
Fitur	FeatureType	Tidak
Waktu terakhir dimodifikasi	LastModifiedTime	Tidak
Parameter	Parameter. *kunci*	Ya

Cara mencari fitur Anda

Petunjuk untuk menggunakan Feature Store melalui konsol tergantung pada apakah Anda telah mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda. Pilih salah satu petunjuk berikut berdasarkan kasus penggunaan Anda.

Cari fitur jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data di panel navigasi kiri untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. (Opsional) Untuk melihat fitur Anda, pilih Akun saya. Untuk melihat fitur bersama, pilih Cross account.
5. Di bawah tab Katalog Fitur, pilih Akun saya untuk melihat grup fitur Anda.

6. Di bawah tab Katalog Fitur, pilih Lintas akun untuk melihat grup fitur yang dibuat orang lain yang dapat ditemukan oleh Anda. Di bawah Dibuat oleh, Anda dapat melihat ID akun pemilik sumber daya.
7. Anda dapat mencari fitur Anda di daftar dropdown Penelusuran:

   - (Opsional) Untuk memfilter pencarian Anda, pilih ikon filter di sebelah daftar dropdown Pencarian. Anda dapat menggunakan filter untuk menentukan parameter atau rentang tanggal dalam hasil pencarian Anda. Jika Anda mencari parameter, tentukan kunci dan nilainya. Untuk menemukan fitur Anda, tentukan rentang waktu, atau hapus (batal pilihan) kolom yang tidak ingin Anda kueri.
   - Untuk sumber daya bersama, Anda hanya dapat mengedit metadata grup fitur atau definisi fitur jika Anda memiliki izin akses yang tepat yang diberikan dari akun pemilik sumber daya. Izin dapat ditemukan saja tidak akan memungkinkan Anda untuk mengedit metadata atau definisi fitur. Untuk informasi lebih lanjut tentang izin akses, lihat. Mengaktifkan akses lintas akun

Cari fitur Anda menggunakan SDK for Python (Boto3)

Kode di bagian ini menggunakan `Search`operasi di AWS SDK for Python (Boto3) untuk menjalankan kueri penelusuran untuk menemukan fitur di grup fitur Anda. Untuk informasi tentang bahasa lain untuk mengirimkan kueri, lihat Lihat Juga di Referensi Amazon SageMaker API.

Untuk contoh dan sumber daya Toko Fitur lainnya, lihatSumber daya Toko SageMaker Fitur Amazon.

Kode berikut menunjukkan contoh kueri penelusuran yang berbeda menggunakan API:

```
# Return all features in your feature groups
sagemaker_client.search(
    Resource="FeatureMetadata",
)

# Search for all features that belong to a feature group that contain the "ver"
 substring
sagemaker_client.search(
    Resource="FeatureMetadata",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
```

```
                'Operator': 'Contains',
                'Value': 'ver'
            },
        ]
    }
)

# Search for all features that belong to a feature group that have the EXACT name
 "airport"
sagemaker_client.search(
    Resource="FeatureMetadata",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Equals',
                'Value': 'airport'
            },
        ]
    }
)

# Search for all features that belong to a feature group that contains the name "ver"
AND have a name that contains "wha"
AND have a parameter (key or value) that contains "hea"

sagemaker_client.search(
    Resource="FeatureMetadata",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Contains',
                'Value': 'ver'
            },
            {
                'Name': 'FeatureName',
                'Operator': 'Contains',
                'Value': 'wha'
            },
            {
                'Name': 'AllParameters',
                'Operator': 'Contains',
                'Value': 'hea'
```

```
            },
        ]
    }
)

# Search for all features that belong to a feature group with substring "ver" in its
 name
OR features that have a name that contain "wha"
OR features that have a parameter (key or value) that contains "hea"

sagemaker_client.search(
    Resource="FeatureMetadata",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Contains',
                'Value': 'ver'
            },
            {
                'Name': 'FeatureName',
                'Operator': 'Contains',
                'Value': 'wha'
            },
            {
                'Name': 'AllParameters',
                'Operator': 'Contains',
                'Value': 'hea'
            },
        ],
        'Operator': 'Or' # note that this is explicitly set to "Or"- the default is
 "And"
    }
)


# Search for all features that belong to a feature group with substring "ver" in its
 name
OR features that have a name that contain "wha"
OR parameters with the value 'Sage' for the 'org' key

sagemaker_client.search(
    Resource="FeatureMetadata",
    SearchExpression={
```

```
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Contains',
                'Value': 'ver'
            },
            {
                'Name': 'FeatureName',
                'Operator': 'Contains',
                'Value': 'wha'
            },
            {
                'Name': 'Parameters.org',
                'Operator': 'Contains',
                'Value': 'Sage'
            },
        ],
        'Operator': 'Or' # note that this is explicitly set to "Or"- the default is
 "And"
    }
)
```

Temukan grup fitur di Toko Fitur

Dengan Amazon SageMaker Feature Store, Anda dapat mencari grup fitur menggunakan konsol atau operasi Pencarian. Anda dapat menggunakan fungsi pencarian untuk menemukan fitur dan grup fitur yang relevan dengan model yang Anda buat. Anda dapat menggunakan fungsi pencarian untuk dengan cepat menemukan grup fitur yang relevan dengan kasus penggunaan Anda.

> **Note**
>
> Grup fitur yang Anda cari harus berada di dalam Wilayah AWS dan AWS akun Anda, atau dibagikan dan dibuat dapat ditemukan oleh Anda. Akun AWS Untuk informasi selengkapnya tentang cara membagikan katalog grup fitur dan memberikan kemampuan untuk menemukan, lihat. Bagikan katalog grup fitur Anda

Tabel berikut menunjukkan bidang yang dapat dicari dan apakah Anda dapat menggunakan konsol untuk mencari bidang tertentu.

Anda dapat mencari fitur menggunakan Amazon SageMaker Studio Classic atau `Search`operasi di SageMaker API. Tabel berikut mencantumkan semua metadata yang dapat dicari dan apakah Anda dapat mencarinya di konsol. Tag dapat dicari untuk grup fitur Anda sendiri tetapi tidak dapat dicari untuk grup fitur yang dapat ditemukan oleh Anda.

Metadata yang dapat dicari	Nama bidang	Dapat dicari di konsol?	Dapat dicari dengan akun silang?
Semua Tag	AllTags	Ya	Tidak
Alasan Kegagalan Penciptaan	FailureReason	Tidak	Tidak
Membuat Alur	FeatureGroupStatus	Ya	Ya
Waktu pembuatan	CreationTime	Ya	Ya
Deskripsi	Deskripsi	Ya	Ya
Nama Fitur Waktu Acara	EventTimeFeatureName	Tidak	Tidak
Definisi Fitur	FeatureDefinitions	Tidak	Tidak
Nama grup	FeatureGroupARN	Tidak	Tidak
Nama Grup Fitur	FeatureGroupName	Ya	Ya
Konfigurasi Toko Offline	OfflineStoreConfig	Tidak	Tidak
Status Toko Offline	OfflineStoreStatus	Ya	Ya
Status Pembaruan Terakhir	LastUpdateStatus	Tidak	Tidak
Nama Fitur Record Identfier	RecordIdentifierFeatureName	Ya	Ya
Tag	Tag. `key`	Ya	Tidak

Cara menemukan grup fitur

Anda dapat menggunakan konsol atau Amazon SageMaker Feature Store API untuk menemukan grup fitur Anda. Petunjuk untuk menggunakan Feature Store melalui konsol tergantung pada apakah Anda telah mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Temukan grup fitur jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data di panel navigasi kiri untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. (Opsional) Untuk melihat grup fitur Anda, pilih Akun saya. Untuk melihat grup fitur bersama, pilih Lintas akun.
5. Di bawah tab Katalog Grup Fitur, pilih Akun saya untuk melihat grup fitur Anda.
6. Di bawah tab Katalog Grup Fitur, pilih Lintas akun untuk melihat grup fitur yang dibuat orang lain yang dapat ditemukan oleh Anda. Di bawah Dibuat oleh, Anda dapat melihat ID akun pemilik sumber daya.
7. Anda dapat mencari grup fitur Anda di daftar dropdown Penelusuran:

   - (Opsional) Untuk memfilter pencarian Anda, pilih ikon filter di sebelah daftar dropdown Pencarian. Anda dapat menggunakan filter untuk menentukan parameter atau rentang tanggal dalam hasil pencarian Anda. Jika Anda mencari parameter, tentukan kunci dan nilainya. Untuk menemukan grup fitur, Anda dapat menentukan rentang waktu, menghapus (membatalkan pilihan) kolom yang tidak ingin Anda kueri, memilih toko untuk dicari, atau mencari berdasarkan status.
   - Untuk sumber daya bersama, Anda hanya dapat mengedit metadata grup fitur atau definisi fitur jika Anda memiliki izin akses yang tepat yang diberikan dari akun pemilik sumber daya. Izin dapat ditemukan saja tidak akan memungkinkan Anda untuk mengedit metadata atau definisi fitur. Untuk informasi lebih lanjut tentang izin akses, lihat. Mengaktifkan akses lintas akun

Temukan grup fitur menggunakan SDK for Python (Boto3)

Kode di bagian ini menggunakan `Search`operasi di AWS SDK for Python (Boto3) untuk menjalankan permintaan pencarian untuk menemukan grup fitur. Untuk informasi tentang bahasa lain untuk mengirimkan kueri, lihat Lihat Juga di Referensi Amazon SageMaker API.

Untuk contoh dan sumber daya Toko Fitur lainnya, lihatSumber daya Toko SageMaker Fitur Amazon.

Kode berikut menunjukkan contoh kueri penelusuran yang berbeda menggunakan API:

```
# Return all feature groups
sagemaker_client.search(
    Resource="FeatureGroups",
)

# Search for feature groups that are shared with your account
sagemaker_session.search(
    resource="FeatureGroup",
    search_expression={
        "Filters": [
            {
                "Name": "FeatureGroupName",
                "Value": "MyFeatureGroup",
                "Operator": "Contains",
            }
        ],
        "Operator": "And",
    },
    sort_by="Name",
    sort_order="Ascending",
    next_token="token",
    max_results=50,
    CrossAccountFilterOption="SameAccount"
)

# Search for all feature groups with a name that contains the "ver" substring
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Contains',
                'Value': 'ver'
```

```
            },
        ]
    }
)

# Search for all feature groups that have the EXACT name "airport"
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Equals',
                'Value': 'airport'
            },
        ]
    }
)

# Search for all feature groups that contains the name "ver"
# AND have a record identifier feature name that contains "wha"
# AND have a tag (key or value) that contains "hea"
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Contains',
                'Value': 'ver'
            },
            {
                'Name': 'RecordIdentifierFeatureName',
                'Operator': 'Contains',
                'Value': 'wha'
            },
            {
                'Name': 'AllTags',
                'Operator': 'Contains',
                'Value': 'hea'
            },
        ]
    }
)
```

```
# Search for all feature groups with substring "ver" in its name
# OR feature groups that have a record identifier feature name that contains "wha"
# OR feature groups that have a tag (key or value) that contains "hea"
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Contains',
                'Value': 'ver'
            },
            {
                'Name': 'RecordIdentifierFeatureName',
                'Operator': 'Contains',
                'Value': 'wha'
            },
            {
                'Name': 'AllTags',
                'Operator': 'Contains',
                'Value': 'hea'
            },
        ],
        'Operator': 'Or' # note that this is explicitly set to "Or"- the default is
 "And"
    }
)


# Search for all feature groups with substring "ver" in its name
# OR feature groups that have a record identifier feature name that contains "wha"
# OR tags with the value 'Sage' for the 'org' key
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'FeatureGroupName',
                'Operator': 'Contains',
                'Value': 'ver'
            },
            {
                'Name': 'RecordIdentifierFeatureName',
```

```
                'Operator': 'Contains',
                'Value': 'wha'
            },
            {
                'Name': 'Tags.org',
                'Operator': 'Contains',
                'Value': 'Sage'
            },
        ],
        'Operator': 'Or' # note that this is explicitly set to "Or"- the default is
 "And"
    }
)

# Search for all offline only feature groups
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'OnlineStoreConfig.EnableOnlineStore',
                'Operator': 'NotEquals',
                'Value': 'true'
            },
            {
                'Name': 'OfflineStoreConfig.S3StorageConfig.S3Uri',
                'Operator': 'Exists'
            }
        ]
    }
)

# Search for all online only feature groups
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'OnlineStoreConfig.EnableOnlineStore',
                'Operator': 'Equals',
                'Value': 'true'
            },
            {
                'Name': 'OfflineStoreConfig.S3StorageConfig.S3Uri',
```

```
                'Operator': 'NotExists'
            }
        ]
    }
)

# Search for all feature groups that are BOTH online and offline
sagemaker_client.search(
    Resource="FeatureGroups",
    SearchExpression={
        'Filters': [
            {
                'Name': 'OnlineStoreConfig.EnableOnlineStore',
                'Operator': 'Equals',
                'Value': 'true'
            },
            {
                'Name': 'OfflineStoreConfig.S3StorageConfig.S3Uri',
                'Operator': 'Exists'
            }
        ]
    }
)
```

Anda juga dapat menggunakan python SDK dari AWS RAM API untuk membuat pembagian sumber daya. Tanda tangan API diberikan di bawah ini. Untuk menggunakan python SDK AWS RAM API, Anda perlu melampirkan kebijakan terkelola akses AWS RAM penuh dengan Peran eksekusi.

```
response = client.create_resource_share(
    name='string',
    resourceArns=[
        'string',
    ],
    principals=[
        'string',
    ],
    tags=[
        {
            'key': 'string',
            'value': 'string'
        },
    ],
```

```
    allowExternalPrincipals=True|False,
    clientToken='string',
    permissionArns=[
        'string',
    ]
)
```

Menambahkan metadata yang dapat dicari ke fitur Anda

Di Amazon SageMaker Feature Store, Anda dapat mencari melalui semua fitur Anda. Untuk membuat fitur Anda lebih mudah ditemukan, Anda dapat menambahkan metadata ke dalamnya. Anda dapat menambahkan jenis metadata berikut:

- Deskripsi — Deskripsi fitur yang dapat dicari.
- Parameter — Pasangan nilai kunci yang dapat dicari.

Deskripsi dapat memiliki hingga 255 karakter. Untuk parameter, Anda harus menentukan pasangan kunci-nilai dalam pencarian Anda. Anda dapat menambahkan hingga 25 parameter.

Untuk memperbarui metadata fitur, Anda dapat menggunakan konsol atau operasi. `UpdateFeatureMetadata`

Cara menambahkan metadata yang dapat dicari ke fitur Anda

Anda dapat menggunakan konsol atau Amazon SageMaker Feature Store API untuk menambahkan metadata yang dapat dicari ke fitur Anda. Petunjuk untuk menggunakan Feature Store melalui konsol bergantung pada apakah Anda telah mengaktifkan SageMaker Studio Amazon atau Amazon SageMaker Studio Klasik sebagai pengalaman default Anda.

Tambahkan metadata yang dapat dicari ke fitur jika Studio adalah pengalaman default Anda (konsol)

1. Buka konsol Studio dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio.
2. Pilih Data di panel navigasi kiri, untuk memperluas daftar dropdown.
3. Dari daftar dropdown, pilih Toko Fitur.
4. (Opsional) Untuk melihat fitur Anda, pilih Akun saya. Untuk melihat fitur bersama, pilih Cross account.
5. Untuk melihat grup fitur Anda, di bawah tab Katalog Fitur, pilih Akun saya.

6. Di bawah tab Katalog Fitur, pilih Lintas akun untuk melihat grup fitur yang orang lain buat dapat ditemukan oleh Anda. Di bawah Dibuat oleh, Anda dapat melihat ID akun pemilik sumber daya dari grup fitur.
7. Anda dapat mencari fitur Anda di daftar dropdown Penelusuran.

   - (Opsional) Untuk memfilter pencarian Anda, pilih ikon filter di sebelah daftar dropdown Pencarian. Anda dapat menggunakan filter untuk menentukan parameter atau rentang tanggal dalam hasil pencarian Anda. Jika Anda mencari parameter, tentukan kunci dan nilainya. Untuk menemukan fitur Anda dengan lebih mudah, Anda dapat menentukan rentang waktu atau membatalkan pilihan kolom yang tidak ingin Anda kueri.
   - Untuk sumber daya bersama, Anda hanya dapat mengedit metadata grup fitur atau definisi fitur jika Anda memiliki izin akses yang tepat yang diberikan dari akun pemilik sumber daya. Memiliki izin dapat ditemukan saja tidak memungkinkan Anda untuk mengedit metadata atau definisi fitur. Untuk informasi lebih lanjut tentang izin akses, lihat. Mengaktifkan akses lintas akun
8. Pilih fitur Anda.
9. Pilih Edit metadata.
10. Di bidang Deskripsi, tambahkan atau perbarui deskripsi.
11. Di bidang Parameter di bawah Parameter, tentukan pasangan kunci-nilai untuk parameter.
12. (Opsional) Pilih Tambahkan parameter baru untuk menambahkan parameter lain.
13. Pilih Simpan perubahan.
14. Pilih Konfirmasi.

Tambahkan metadata yang dapat dicari ke fitur Anda menggunakan SDK for Python (Boto3)

Kode di bagian ini menggunakan `UpdateFeatureMetadata`operasi di AWS SDK for Python (Boto3) untuk menambahkan metadata yang dapat dicari ke fitur Anda untuk skenario yang berbeda. Untuk informasi tentang bahasa lain untuk mengirimkan kueri, lihat Lihat Juga di Referensi Amazon SageMaker API.

Untuk contoh dan sumber daya Toko Fitur lainnya, lihatSumber daya Toko SageMaker Fitur Amazon.

Add a list of parameters to a feature

Untuk menambahkan daftar parameter ke fitur, tentukan nilai untuk bidang berikut:

- `FeatureGroupName`

- `Feature`
- `Parameters`

Kode contoh berikut menggunakan AWS SDK for Python (Boto3) untuk menambahkan dua parameter.

```
sagemaker_client.update_feature_metadata(
    FeatureGroupName="feature_group_name",
    FeatureName="feature-name",
    ParameterAdditions=[
        {"Key": "example-key-0", "Value": "example-value-0"},
        {"Key": "example-key-1", "Value": "example-value-1"},
    ]
)
```

Add a description to a feature

Untuk menambahkan deskripsi ke fitur, tentukan nilai untuk kolom berikut:

- `FeatureGroupName`
- `Feature`
- `Description`

```
sagemaker_client.update_feature_metadata(
    FeatureGroupName="feature-group-name",
    FeatureName="feature-name",
    Description="description"
)
```

Remove parameters for a feature

Untuk menghapus semua parameter untuk fitur, lakukan hal berikut ini.

Tentukan nilai dari bidang berikut:

- `FeatureGroupName`

- `Feature`

Tentukan kunci untuk parameter yang Anda hapus di bawah`ParameterRemovals`.

```
sagemaker_client.update_feature_metadata(
    FeatureGroupName="feature_group_name",
    FeatureName="feature-name",
        ParameterRemovals=[
        {"Key": "example-key-0"},
        {"Key": "example-key-1"},
    ]
)
```

Remove the description for a feature

Untuk menghapus deskripsi fitur, lakukan hal berikut ini.

Tentukan nilai dari bidang berikut:

- `FeatureGroupName`
- `Feature`

Tentukan string kosong untuk`Description`.

```
sagemaker_client.update_feature_metadata(
    FeatureGroupName="feature-group-name",
    FeatureName="feature-name",
    Description=""
)
```

Contoh kode

Setelah memperbarui metadata untuk suatu fitur, Anda dapat menggunakan [`DescribeFeatureMetadata`]operasi untuk melihat pembaruan yang telah Anda buat.

Kode berikut melewati alur kerja contoh menggunakan file. AWS SDK for Python (Boto3) Kode contoh tersebut melakukan hal berikut:

1. Mengatur SageMaker lingkungan Anda.
2. Membuat grup fitur.
3. Menambahkan fitur ke grup.
4. Menambahkan metadata ke fitur.

Untuk contoh dan sumber daya Toko Fitur lainnya, lihatSumber daya Toko SageMaker Fitur Amazon.

Langkah 1: Siapkan

Untuk mulai menggunakan Feature Store, buat SageMaker, boto3 dan sesi Feature Store. Kemudian atur bucket S3 yang ingin Anda gunakan untuk fitur Anda. Ini adalah toko offline Anda. Kode berikut menggunakan bucket SageMaker default dan menambahkan awalan kustom ke dalamnya.

> **Note**
>
> Peran yang Anda gunakan harus memiliki kebijakan terkelola berikut yang melekat padanya: `AmazonS3FullAccess` dan`AmazonSageMakerFeatureStoreAccess`.

```
# SageMaker Python SDK version 2.x is required
%pip install 'sagemaker>=2.0.0'
import sagemaker
import sys
```

```


import boto3
import pandas as pd
import numpy as np
import io
from sagemaker.session import Session
from sagemaker import get_execution_role
from botocore.exceptions import ClientError


prefix = 'sagemaker-featurestore-introduction'
role = get_execution_role()

sagemaker_session = sagemaker.Session()
region = sagemaker_session.boto_region_name
```

```
s3_bucket_name = sagemaker_session.default_bucket()
sagemaker_client = boto_session.client(service_name='sagemaker', region_name=region)
```

Langkah 2: Buat grup fitur dan tambahkan fitur

Kode berikut adalah contoh membuat grup fitur dengan definisi fitur.

```
feature_group_name = "test-for-feature-metadata"
feature_definitions = [
    {"FeatureName": "feature-1", "FeatureType": "String"},
    {"FeatureName": "feature-2", "FeatureType": "String"},
    {"FeatureName": "feature-3", "FeatureType": "String"},
    {"FeatureName": "feature-4", "FeatureType": "String"},
    {"FeatureName": "feature-5", "FeatureType": "String"}
]
try:
    sagemaker_client.create_feature_group(
        FeatureGroupName=feature_group_name,
        RecordIdentifierFeatureName="feature-1",
        EventTimeFeatureName="feature-2",
        FeatureDefinitions=feature_definitions,
        OnlineStoreConfig={"EnableOnlineStore": True}
    )
except ClientError as e:
    if e.response["Error"]["Code"] == "ResourceInUse":
        pass
    else:
        raise e
```

Langkah 3: Tambahkan metadata

Sebelum Anda menambahkan metadata, gunakan `DescribeFeatureGroup` operasi untuk memastikan bahwa status grup fitur adalah. `Created`

```
sagemaker_client.describe_feature_group(
        FeatureGroupName=feature_group_name
    )
```

Tambahkan deskripsi ke fitur.

```
sagemaker_client.update_feature_metadata(
    FeatureGroupName=feature_group_name,
    FeatureName="feature-1",
    Description="new description"
)
```

Anda dapat menggunakan DescribeFeatureMetadataoperasi untuk melihat apakah Anda berhasil memperbarui deskripsi untuk grup fitur.

```
    sagemaker_client.describe_feature_metadata(
    FeatureGroupName=feature_group_name,
    FeatureName="feature-1"
)
```

Anda juga dapat menggunakannya untuk menambahkan parameter ke grup fitur.

```
sagemaker_client.update_feature_metadata(
    FeatureGroupName=feature_group_name,
    FeatureName="feature-1",
    ParameterAdditions=[
        {"Key": "team", "Value": "featurestore"},
        {"Key": "org", "Value": "sagemaker"},
    ]
)
```

Anda dapat menggunakan DescribeFeatureMetadataoperasi lagi untuk melihat apakah Anda telah berhasil menambahkan parameter.

```
    sagemaker_client.describe_feature_metadata(
    FeatureGroupName=feature_group_name,
    FeatureName="feature-1"
)
```

Membuat kumpulan data dari grup fitur Anda

Setelah grup fitur Feature Store dibuat di toko offline, Anda dapat memilih untuk menggunakan metode berikut untuk mendapatkan data Anda:

- Menggunakan Amazon SageMaker Python SDK
- Menjalankan kueri SQL di Amazon Athena

> **Important**
>
> Feature Store membutuhkan data untuk didaftarkan dalam katalog AWS Glue data. Secara default, Feature Store secara otomatis membuat katalog AWS Glue data saat Anda membuat grup fitur.

Setelah membuat grup fitur untuk toko offline dan mengisinya dengan data, Anda dapat membuat kumpulan data dengan menjalankan kueri atau menggunakan SDK untuk menggabungkan data yang disimpan di toko offline dari grup fitur yang berbeda. Anda juga dapat bergabung dengan grup fitur ke kerangka data panda tunggal. Anda dapat menggunakan Amazon Athena untuk menulis dan menjalankan kueri SQL.

> **Note**
>
> Untuk memastikan bahwa data Anda mutakhir, Anda dapat mengatur AWS Glue crawler agar berjalan sesuai jadwal.
> Untuk menyiapkan AWS Glue crawler, tentukan peran IAM yang digunakan crawler untuk mengakses bucket Amazon S3 toko offline. Untuk informasi lebih lanjut, lihat Membuat peran IAM.
> Untuk informasi lebih lanjut tentang cara menggunakan AWS Glue dan Athena untuk membangun kumpulan data pelatihan untuk pelatihan model dan inferensi, lihat. Gunakan Feature Store dengan SDK for Python (Boto3)

Menggunakan Amazon SageMaker Python SDK untuk mendapatkan data dari grup fitur

Anda dapat menggunakan API Toko Fitur untuk membuat kumpulan data dari grup fitur Anda. Ilmuwan data membuat kumpulan data ML untuk pelatihan dengan mengambil data fitur ML dari satu atau beberapa grup fitur di toko offline. Gunakan `create_dataset()` fungsi untuk membuat dataset. Anda dapat menggunakan SDK untuk melakukan hal berikut:

- Buat kumpulan data dari beberapa grup fitur.
- Buat kumpulan data dari grup fitur dan bingkai data panda.

Secara default, Feature Store tidak menyertakan catatan yang telah Anda hapus dari kumpulan data. Itu juga tidak termasuk catatan duplikat. Rekaman duplikat memiliki ID rekaman dan nilai stempel waktu di kolom waktu acara.

Sebelum Anda menggunakan SDK untuk membuat kumpulan data, Anda harus memulai sesi. SageMaker Gunakan kode berikut untuk memulai sesi.

```
import boto3
from sagemaker.session import Session
from sagemaker.feature_store.feature_store import FeatureStore

region = boto3.Session().region_name
boto_session = boto3.Session(region_name=region)

sagemaker_client = boto_session.client(
    service_name="sagemaker", region_name=region
 )
featurestore_runtime = boto_session.client(
    service_name="sagemaker-featurestore-runtime",region_name=region
)

feature_store_session = Session(
    boto_session=boto_session,
    sagemaker_client=sagemaker_client,
    sagemaker_featurestore_runtime_client=featurestore_runtime,
)

feature_store = FeatureStore(feature_store_session)
```

Kode berikut menunjukkan contoh pembuatan dataset dari beberapa grup fitur. Cuplikan kode berikut menggunakan contoh grup fitur "*base_fg_name “," first_fg_name “, dan" second_fg_name “, yang mungkin tidak ada atau memiliki skema yang sama di dalam Toko* Fitur Anda. Disarankan untuk mengganti grup fitur ini dengan grup fitur yang ada di dalam Toko Fitur Anda. Untuk informasi tentang cara membuat grup fitur, lihatLangkah 3: Membuat grup fitur.

```
from sagemaker.feature_store.feature_group import FeatureGroup

s3_bucket_name = "offline-store-sdk-test"

base_fg_name = "base_fg_name"
base_fg = FeatureGroup(name=base_fg_name, sagemaker_session=feature_store_session)

first_fg_name = "first_fg_name"
first_fg = FeatureGroup(name=first_fg_name, sagemaker_session=feature_store_session)

second_fg_name = "second_fg_name"
second_fg = FeatureGroup(name=second_fg_name, sagemaker_session=feature_store_session)

feature_store = FeatureStore(feature_store_session)
builder = feature_store.create_dataset(
    base=base_fg,
    output_path=f"s3://{DOC-EXAMPLE-BUCKET1}",
).with_feature_group(first_fg
).with_feature_group(second_fg, "base_id", ["base_feature_1"])
```

Kode berikut menunjukkan contoh membuat dataset dari beberapa grup fitur dan kerangka data panda.

```
base_data = [[1, 187512346.0, 123, 128],
             [2, 187512347.0, 168, 258],
             [3, 187512348.0, 125, 184],
             [1, 187512349.0, 195, 206]]
base_data_df = pd.DataFrame(
    base_data,
    columns=["base_id", "base_time", "base_feature_1", "base_feature_2"]
)

builder = feature_store.create_dataset(
    base=base_data_df,
```

```
    event_time_identifier_feature_name='base_time',
    record_identifier_feature_name='base_id',
    output_path=f"s3://{s3_bucket_name}"
).with_feature_group(first_fg
).with_feature_group(second_fg, "base_id", ["base_feature_1"])
```

Feature Store API memberi Anda metode pembantu untuk `create_dataset` fungsi tersebut. Anda dapat menggunakannya untuk melakukan hal berikut:

- Buat kumpulan data dari beberapa grup fitur.
- Buat kumpulan data dari beberapa grup fitur dan kerangka data panda.
- Buat kumpulan data dari satu grup fitur dan kerangka data panda.
- Buat kumpulan data menggunakan gabungan akurat titik waktu di mana catatan dalam grup fitur yang digabungkan mengikuti secara berurutan.
- Buat kumpulan data dengan catatan duplikat, alih-alih mengikuti perilaku default fungsi.
- Buat kumpulan data dengan catatan yang dihapus, alih-alih mengikuti perilaku default fungsi.
- Buat kumpulan data untuk periode waktu yang Anda tentukan.
- Simpan dataset sebagai file CSV.
- Simpan dataset sebagai kerangka data panda.

Grup fitur dasar adalah konsep penting untuk bergabung. Grup fitur dasar adalah grup fitur yang memiliki grup fitur lain atau kerangka data panda yang bergabung dengannya. Data per set

Anda dapat menambahkan metode opsional berikut ke `create_dataset` fungsi untuk mengonfigurasi cara Anda membuat kumpulan data:

- `with_feature_group`— Melakukan penggabungan batin antara grup fitur dasar dan grup fitur lain menggunakan pengenal catatan dan nama fitur target di grup fitur dasar. Berikut ini memberikan informasi tentang parameter yang Anda tentukan:
  - `feature_group`— Grup fitur yang Anda ikuti.
  - `target_feature_name_in_base`— Nama fitur di grup fitur dasar yang Anda gunakan sebagai kunci dalam bergabung. Pengenal rekaman di grup fitur lain adalah kunci lain yang digunakan Feature Store dalam bergabung.

  - `included_feature_names`— Daftar string yang mewakili nama fitur dari grup fitur dasar. Anda dapat menggunakan bidang untuk menentukan fitur yang ingin Anda sertakan dalam kumpulan data.
  - `feature_name_in_target`— String opsional yang mewakili fitur dalam grup fitur target yang akan dibandingkan dengan fitur target di grup fitur dasar.
  - `join_comparator`— Opsional `JoinComparatorEnum` mewakili komparator yang digunakan saat bergabung dengan fitur target di grup fitur dasar dan fitur dalam grup fitur target. `JoinComparatorEnum`Nilai-nilai ini dapat `GREATER_THANGREATER_THAN_OR_EQUAL_TO`,,`LESS_THAN`,`LESS_THAN_OR_EQUAL_TO`, `NOT_EQUAL_TO` atau secara `EQUALS` default.
  - `join_type`— Opsional `JoinTypeEnum` mewakili jenis gabungan antara kelompok fitur dasar dan target. `JoinTypeEnum`Nilai-nilai ini dapat`LEFT_JOIN`,`RIGHT_JOIN`,`FULL_JOIN`, `CROSS_JOIN` atau secara `INNER_JOIN` default.
- `with_event_time_range`— Membuat kumpulan data menggunakan rentang waktu acara yang Anda tentukan.
- `as_of`— Membuat kumpulan data hingga stempel waktu yang Anda tentukan. Misalnya, jika Anda menentukan `datetime(2021, 11, 28, 23, 55, 59, 342380)` sebagai nilainya, buat kumpulan data hingga 28 November 2021.
- `point_time_accurate_join`— Membuat kumpulan data di mana semua nilai waktu acara dari grup fitur dasar kurang dari semua nilai waktu acara grup fitur atau kerangka data panda yang Anda ikuti.
- `include_duplicated_records`— Menyimpan nilai duplikat dalam grup fitur.
- `include_deleted_records`— Menyimpan nilai yang dihapus dalam grup fitur.
- `with_number_of_recent_records_by_record_identifier`— Bilangan bulat yang Anda tentukan untuk menentukan berapa banyak catatan terbaru yang muncul di kumpulan data.
- `with_number_of_records_by_record_identifier`— Bilangan bulat yang mewakili berapa banyak catatan yang muncul dalam dataset.

Setelah mengonfigurasi kumpulan data, Anda dapat menentukan output menggunakan salah satu metode berikut:

- `to_csv_file`— Menyimpan dataset sebagai file CSV.
- `to_dataframe`— Menyimpan dataset sebagai kerangka data panda.

Anda dapat mengambil data yang datang setelah periode waktu tertentu. Kode berikut mengambil data setelah stempel waktu.

```
fg1 = FeatureGroup("example-feature-group-1")
feature_store.create_dataset(
    base=fg1,
    output_path="s3://example-S3-path"
).with_number_of_records_from_query_results(5).to_csv_file()
```

Anda juga dapat mengambil data dari periode waktu tertentu. Anda dapat menggunakan kode berikut untuk mendapatkan data untuk rentang waktu tertentu:

```
fg1 = FeatureGroup("fg1")
feature_store.create_dataset(
    base=fg1,
    output_path="example-S3-path"
).with_event_time_range(
    datetime(2021, 11, 28, 23, 55, 59, 342380),
    datetime(2020, 11, 28, 23, 55, 59, 342380)
).to_csv_file() #example time range specified in datetime functions
```

Anda mungkin ingin menggabungkan beberapa grup fitur ke kerangka data panda di mana nilai waktu acara grup fitur terjadi selambat-lambatnya waktu peristiwa bingkai data. Gunakan kode berikut sebagai templat untuk membantu Anda melakukan gabungan.

```
fg1 = FeatureGroup("fg1")
fg2 = FeatureGroup("fg2")
events = [['2020-02-01T08:30:00Z', 6, 1],
          ['2020-02-02T10:15:30Z', 5, 2],
          ['2020-02-03T13:20:59Z', 1, 3],
          ['2021-01-01T00:00:00Z', 1, 4]]
df = pd.DataFrame(events, columns=['event_time', 'customer-id', 'title-id'])
feature_store.create_dataset(
    base=df,
    event_time_identifier_feature_name='event_time',
    record_identifier_feature_name='customer_id',
    output_path="s3://example-S3-path"
).with_feature_group(fg1, "customer-id"
).with_feature_group(fg2, "title-id"
).point_in_time_accurate_join(
).to_csv_file()
```

Anda juga dapat mengambil data yang datang setelah periode waktu tertentu. Kode berikut mengambil data setelah waktu yang ditentukan oleh stempel waktu dalam metode. `as_of`

```
fg1 = FeatureGroup("fg1")
feature_store.create_dataset(
    base=fg1,
    output_path="s3://example-s3-file-path"
).as_of(datetime(2021, 11, 28, 23, 55, 59, 342380)
).to_csv_file() # example datetime values
```

Contoh kueri Amazon Athena

Anda dapat menulis kueri di Amazon Athena untuk membuat kumpulan data dari grup fitur Anda. Anda juga dapat menulis kueri yang membuat kumpulan data dari grup fitur dan kerangka data panda tunggal.

Eksplorasi Interaktif

Query ini memilih 1000 record pertama.

```
SELECT *
FROM <FeatureGroup.DataCatalogConfig.DatabaseName>.<FeatureGroup.DataCatalogConfig.TableName>
LIMIT 1000
```

Snapshot terbaru tanpa duplikat

Kueri ini memilih catatan non-duplikat terbaru.

```
SELECT *
FROM
    (SELECT *,
         row_number()
        OVER (PARTITION BY <RecordIdentiferFeatureName>
    ORDER BY  <EventTimeFeatureName> desc, Api_Invocation_Time DESC, write_time DESC)
 AS row_num
    FROM
 <FeatureGroup.DataCatalogConfig.DatabaseName>.<FeatureGroup.DataCatalogConfig.TableName>)
WHERE row_num = 1;
```

Snapshot terbaru tanpa duplikat dan catatan yang dihapus di toko offline

Kueri ini menyaring semua catatan yang dihapus dan memilih catatan non-duplikat dari toko offline.

```
SELECT *
FROM
    (SELECT *,
         row_number()
        OVER (PARTITION BY <RecordIdentiferFeatureName>
    ORDER BY  <EventTimeFeatureName> desc, Api_Invocation_Time DESC, write_time DESC)
 AS row_num
    FROM
 <FeatureGroup.DataCatalogConfig.DatabaseName>.<FeatureGroup.DataCatalogConfig.TableName>)
WHERE row_num = 1 and
NOT is_deleted;
```

Perjalanan Waktu tanpa duplikat dan catatan yang dihapus di toko offline

Kueri ini menyaring catatan yang dihapus dan memilih catatan non-duplikat dari titik waktu tertentu.

```
SELECT *
FROM
    (SELECT *,
         row_number()
        OVER (PARTITION BY <RecordIdentiferFeatureName>
    ORDER BY  <EventTimeFeatureName> desc, Api_Invocation_Time DESC, write_time DESC)
 AS row_num
    FROM
 <FeatureGroup.DataCatalogConfig.DatabaseName>.<FeatureGroup.DataCatalogConfig.TableName>
    where <EventTimeFeatureName> <= timestamp '<timestamp>')
    -- replace timestamp '<timestamp>' with just <timestamp>  if EventTimeFeature is of
 type fractional
WHERE row_num = 1 and
NOT is_deleted
```

Hapus catatan dari grup fitur Anda

Anda dapat menggunakan Amazon SageMaker Feature Store API untuk menghapus catatan dari grup fitur Anda. Grup fitur adalah objek yang berisi data pembelajaran mesin (ML) Anda, di mana kolom data Anda dijelaskan oleh fitur dan data Anda terkandung dalam catatan. Catatan berisi nilai untuk fitur yang terkait dengan pengenal catatan tertentu.

Ada dua konfigurasi penyimpanan untuk grup fitur Anda: toko online dan toko offline. Toko online hanya menyimpan catatan dengan waktu acara terbaru dan biasanya digunakan untuk pencarian

real-time untuk inferensi ML. Toko offline menyimpan semua catatan dan bertindak sebagai database historis dan biasanya digunakan untuk eksplorasi fitur, pelatihan ML, dan inferensi batch.

Untuk informasi selengkapnya tentang konsep Toko Fitur, lihatPenyempurnaan satu spasi.

Ada dua cara untuk menghapus catatan dari grup fitur Anda, dan perilakunya berbeda tergantung pada konfigurasi penyimpanan. Dalam topik berikut kami akan menjelaskan cara menghapus catatan lunak dan keras dari toko online dan offline dan memberikan contoh.

Topik

Hapus catatan dari toko online

Anda dapat menghapus catatan dengan lembut atau keras dari toko online menggunakan `DeleteRecord` API dengan menggunakan parameter `DeletionMode` permintaan untuk menentukan `SoftDelete` (default) atau`HardDelete`. Untuk informasi selengkapnya tentang `DeleteRecord` API, lihat `DeleteRecord`di Referensi Amazon SageMaker API.

Dengan toko online:

- Saat Anda menghapus lunak (default), catatan tidak lagi dapat diambil oleh GetRecord atau BatchGetRecord dan nilai kolom fitur disetel ke`null`, kecuali untuk nilai `RecordIdentifer` dan `EventTime` fitur.
- Ketika Anda sulit menghapus, catatan sepenuhnya dihapus dari toko online.

Dalam kedua kasus, Feature Store menambahkan penanda rekaman yang dihapus ke file. `OfflineStore` Penanda rekaman yang dihapus adalah catatan yang `RecordIdentifer` sama dengan aslinya, tetapi dengan `is_deleted` nilai yang disetel ke`True`, `EventTime` disetel ke input hapus`EventTime`, dan nilai fitur lainnya yang disetel ke`null`.

Perhatikan bahwa yang `EventTime` ditentukan `DeleteRecord` harus ditetapkan lebih lambat `EventTime` dari catatan yang ada di `OnlineStore` untuk yang sama`RecordIdentifer`. Jika tidak, penghapusan tidak terjadi:

- Untuk`SoftDelete`, catatan yang ada (tidak dihapus) tetap ada di`OnlineStore`, meskipun penanda catatan hapus masih ditulis ke file`OfflineStore`.

- `HardDeletereturnEventTime`: `400 ValidationException` untuk menunjukkan bahwa operasi penghapusan gagal. Tidak ada penanda catatan hapus yang ditulis ke file`OfflineStore`.

Contoh berikut menggunakan operasi SDK for Python (Boto3) untuk `delete_record`menghapus rekaman dari grup fitur. Untuk menghapus catatan dari grup fitur, Anda perlu:

- Nama grup fitur (*`feature-group-name`*)
- Catat nilai pengenal sebagai string () *`record-identifier-value`*
- Waktu acara penghapusan () *`deletion-event-time`*

  Waktu acara penghapusan harus lebih lambat dari waktu acara rekaman yang ingin Anda hapus.

Contoh penghapusan lunak toko online

Untuk penghapusan lunak, Anda perlu menggunakan `DeleteRecord` API dan dapat menggunakan default `DeletionMode` atau mengatur `DeletionMode` ke`SoftDelete`.

```
import boto3
client = boto3.client('sagemaker-featurestore-runtime')

client.delete_record(
    FeatureGroupName='feature-group-name',
    RecordIdentifierValueAsString='record-identifier-value',
    EventTime='deletion-event-time',
    TargetStores=[
        'OnlineStore',
    ],
    DeletionMode='SoftDelete'
)
```

Contoh hard delete toko online

Untuk hard delete, Anda perlu menggunakan `DeleteRecord` API dan atur `DeletionMode` ke`HardDelete`.

```
import boto3
client = boto3.client('sagemaker-featurestore-runtime')

client.delete_record(
```

```
    FeatureGroupName='feature-group-name',
    RecordIdentifierValueAsString='record-identifier-value',
    EventTime='deletion-event-timestamp',
    TargetStores=[
        'OnlineStore',
    ],
    DeletionMode='HardDelete'
)
```

Hapus catatan dari toko offline

Dengan Amazon SageMaker Feature Store Anda dapat menghapus catatan dengan lembut dan keras dari format tabel `OfflineStore` Iceberg. Dengan format tabel `OfflineStore` Iceberg:

- Ketika Anda menghapus catatan versi terbaru dari file tabel Iceberg tidak akan berisi catatan, tetapi versi sebelumnya masih akan berisi catatan dan dapat diakses menggunakan perjalanan waktu. Untuk informasi tentang perjalanan waktu, lihat Menanyakan data tabel Gunung Es dan melakukan perjalanan waktu di panduan pengguna Athena.
- Ketika Anda sulit menghapus catatan Anda menghapus versi sebelumnya dari tabel Iceberg yang berisi catatan. Dalam hal ini Anda harus menentukan versi tabel Iceberg yang ingin Anda hapus.

Dapatkan nama tabel Iceberg Anda

Untuk menghapus lunak dan keras dari tabel `OfflineStore` Iceberg Anda, Anda harus mendapatkan nama tabel Iceberg Anda,. *`iceberg-table-name`* Instruksi berikut mengasumsikan Anda telah menggunakan Feature Store untuk membuat grup fitur menggunakan konfigurasi penyimpanan toko offline menggunakan format tabel Iceberg, dengan `DisableGlueTableCreation = False` (default). Untuk informasi lebih lanjut tentang pembuatan grup fitur, lihatMemulai dengan Amazon SageMaker Feature Store.

Untuk mendapatkan Anda*`iceberg-table-name`*, gunakan `DescribeFeatureGroup`API untuk mendapatkan `DataCatalogConfig`. Ini berisi metadata tabel Glue yang berfungsi sebagai katalog data untuk. `OfflineStore` Yang di `TableName` dalamnya `DataCatalogConfig` adalah milikmu*`iceberg-table-name`*.

Amazon Athena toko offline contoh penghapusan lunak dan keras

Instruksi berikut menggunakan Amazon Athena untuk menghapus lunak kemudian menghapus catatan dari tabel `OfflineStore` Iceberg. Ini mengasumsikan bahwa catatan yang ingin Anda

hapus `OfflineStore` adalah penanda catatan yang dihapus. Untuk informasi tentang penanda rekaman yang dihapus di Anda`OfflineStore`, lihatHapus catatan dari toko online.

1. Dapatkan nama tabel Iceberg Anda,. *`iceberg-table-name`* Untuk informasi tentang cara mendapatkan nama tabel Gunung Es Anda, lihat. Dapatkan nama tabel Iceberg Anda
2. Jalankan `DELETE` perintah untuk menghapus catatan lunak pada`OfflineStore`, sehingga versi terbaru (atau snapshot) dari tabel Iceberg tidak akan berisi catatan. Contoh berikut menghapus catatan di mana `is_deleted 'True'` dan versi waktu peristiwa sebelumnya dari catatan tersebut. Anda dapat menambahkan kondisi tambahan berdasarkan fitur lain untuk membatasi penghapusan. Untuk informasi lebih lanjut tentang penggunaan `DELETE` dengan Athena, lihat `DELETE` di panduan pengguna Athena.

   ```
   DELETE FROM iceberg-table-name WHERE record-id-feature-name IS IN ( SELECT record-id-feature-name FROM iceberg-table-name WHERE is_deleted = 'True')
   ```

   Catatan yang dihapus lunak masih dapat dilihat pada versi file sebelumnya dengan melakukan perjalanan waktu. Untuk informasi tentang melakukan perjalanan waktu, lihat Menanyakan data tabel Gunung Es dan melakukan perjalanan waktu di panduan pengguna Athena.
3. Hapus catatan dari versi sebelumnya dari tabel Iceberg Anda untuk menghapus catatan dari: `OfflineStore`

   a. Jalankan `OPTIMIZE` perintah untuk menulis ulang file data ke dalam tata letak yang lebih dioptimalkan, berdasarkan ukuran dan jumlah file hapus terkait. Untuk informasi selengkapnya tentang mengoptimalkan tabel Iceberg dan sintaksnya, lihat Mengoptimalkan tabel Gunung Es di panduan pengguna Athena.

      ```
      OPTIMIZE iceberg-table-name REWRITE DATA USING BIN_PACK
      ```

   b. (Opsional, hanya perlu dijalankan sekali) Jalankan `ALTER TABLE` perintah untuk mengubah nilai set tabel Iceberg, dan atur kapan versi file sebelumnya harus dihapus dengan keras sesuai dengan spesifikasi Anda. Ini dapat dilakukan dengan menetapkan nilai `vacuum_min_snapshots_to_keep` dan `vacuum_max_snapshot_age_seconds` properti. Untuk informasi selengkapnya tentang mengubah properti kumpulan tabel Iceberg, lihat MENGUBAH PROPERTI SET TABEL di panduan pengguna Athena. Untuk informasi selengkapnya tentang pasangan nilai kunci properti tabel Iceberg, lihat Properti tabel di panduan pengguna Athena.

```
ALTER TABLE iceberg-table-name SET TBLPROPERTIES (
  'vacuum_min_snapshots_to_keep'='your-specified-value',
  'vacuum_max_snapshot_age_seconds'='your-specified-value'
)
```

c. Jalankan `VACUUM` perintah untuk menghapus file data yang tidak lagi diperlukan untuk tabel Iceberg Anda, tidak direferensikan oleh versi saat ini. `VACUUM`Perintah harus dijalankan setelah catatan yang dihapus tidak lagi direferensikan dalam snapshot saat ini. Misalnya, `vacuum_max_snapshot_age_seconds` setelah penghapusan. Untuk informasi lebih lanjut tentang `VACUUM` Athena dan sintaksnya, lihat. `VACUUM`

```
VACUUM iceberg-table-name
```

Contoh penghapusan lunak dan keras toko offline Apache Spark

Untuk menghapus catatan lunak dan kemudian keras dari tabel `OfflineStore` Iceberg menggunakan Apache Spark, Anda dapat mengikuti instruksi yang sama seperti di Amazon Athena toko offline contoh penghapusan lunak dan keras atas, tetapi menggunakan prosedur Spark. Untuk daftar lengkap prosedur, lihat Prosedur Spark dalam dokumentasi Apache Iceberg.

- Saat menghapus lunak dari`OfflineStore`: alih-alih menggunakan perintah di Athena, gunakan `DELETE` perintah di Apache `DELETE FROM`Spark.
- Untuk menghapus catatan dari versi sebelumnya dari tabel Iceberg Anda untuk menghapus catatan dari: `OfflineStore`
  - Saat mengubah konfigurasi tabel Iceberg Anda: alih-alih menggunakan `ALTER TABLE` perintah dari Athena, gunakan prosedur. `expire_snapshots`
  - Untuk menghapus file data yang tidak lagi diperlukan dari tabel Iceberg Anda: alih-alih menggunakan `VACUUM` perintah di Athena, gunakan prosedur. `remove_orphan_files`

Logging operasi Feature Store dengan menggunakan AWS CloudTrail

Amazon SageMaker Feature Store terintegrasi denganAWS CloudTrail, sebuah layanan yang menyediakan catatan tindakan yang dilakukan oleh pengguna, peran, atau AWS layanan di Feature Store. CloudTrail menangkap semua panggilan API untuk Feature Store yang tercantum di halaman

ini. Peristiwa yang dicatat mencakup panggilan API dari manajemen sumber daya Feature Store dan operasi data. Ketika membuat jejak, Anda mengaktifkan pengiriman CloudTrail peristiwa berkelanjutan dari Feature Store ke bucket Amazon S3. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Feature Store, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail lainnya.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Acara manajemen

Operasi penangkapan peristiwa manajemen yang dilakukan pada sumber daya Toko Fitur di AWS akun Anda. Misalnya, log yang dihasilkan dari peristiwa manajemen memberikan visibilitas jika pengguna membuat atau menghapus Toko Fitur. Peristiwa pengelolaan log API berikut dengan Amazon SageMaker Feature Store.

- `CreateFeatureGroup`
- `DeleteFeatureGroup`
- `DescribeFeatureGroup`
- `UpdateFeatureGroup`

Panggilan SageMaker API Amazon dan peristiwa manajemen dicatat secara default saat Anda membuat akun, seperti yang dijelaskan dalamLog Panggilan SageMaker API Amazon dengan AWS CloudTrail. Untuk informasi selengkapnya, lihat Mencatat kejadian manajemen untuk jejak.

Data Peristiwa

Peristiwa data menangkap operasi bidang data yang dilakukan menggunakan sumber daya Penyimpanan Fitur di AWS akun Anda. Misalnya, log yang dihasilkan dari peristiwa data memberikan visibilitas jika pengguna menambahkan atau menghapus catatan dalam grup fitur. API berikut mencatat peristiwa data dengan Amazon SageMaker Feature Store.

- `BatchGetRecord`
- `DeleteRecord`
- `GetRecord`
- `PutRecord`

Peristiwa data tidak dicatat oleh CloudTrail jejak secara default. Untuk mengaktifkan pencatatan peristiwa data, aktifkan pencatatan aktivitas API bidang data di CloudTrail. Untuk informasi selengkapnya, CloudTrail lihat Mencatat kejadian data untuk jejak.

Berikut ini adalah contoh CloudTrail peristiwa untuk panggilan `PutRecord` API:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "USERPRINCIPALID",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "USERACCESSKEYID",
        "userName": "your-user-name"
    },
    "eventTime": "2023-01-01T01:00:00Z",
    "eventSource": "sagemaker.amazonaws.com",
    "eventName": "PutRecord",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "your-user-agent",
    "requestParameters": {
        "featureGroupName": "your-feature-group-name"
    },
    "responseElements": null,
    "requestID": "request-id",
    "eventID": "event-id",
    "readOnly": false,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::SageMaker::FeatureGroup",
            "ARN": "arn:aws:sagemaker:us-east-1:123456789012:feature-group/your-
feature-group-name"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        ...
```

```
    }
}
```

Keamanan dan kontrol akses

Amazon SageMaker Feature Store memungkinkan Anda membuat dua jenis toko: toko online atau toko offline. Toko online digunakan untuk kasus penggunaan inferensi real-time latensi rendah sedangkan toko offline digunakan untuk pelatihan dan kasus penggunaan inferensi batch. Saat Anda membuat grup fitur untuk penggunaan online atau offline, Anda dapat memberikan kunci yang dikelola AWS Key Management Service pelanggan untuk mengenkripsi semua data Anda saat istirahat. Jika Anda tidak memberikan AWS KMS kunci maka kami memastikan bahwa data Anda dienkripsi di sisi server menggunakan kunci yang AWS dimiliki atau AWS KMS kunci AWS terkelolaAWS KMS. Saat membuat grup fitur, Anda dapat memilih jenis penyimpanan dan secara opsional memberikan AWS KMS kunci untuk mengenkripsi data, lalu Anda dapat memanggil berbagai API untuk manajemen data sepertiPutRecord,,. GetRecord DeleteRecord

Fitur Store memungkinkan Anda untuk memberikan atau menolak akses ke individu di tingkat grup fitur dan memungkinkan akses lintas akun ke Toko Fitur. Misalnya, Anda dapat mengatur akun pengembang untuk mengakses toko offline untuk pelatihan model dan eksplorasi yang tidak memiliki akses tulis ke akun produksi. Anda dapat mengatur akun produksi untuk mengakses toko online dan offline. Toko Fitur menggunakan AWS KMS kunci pelanggan unik untuk enkripsi data offline dan toko online saat istirahat. Kontrol akses diaktifkan melalui API dan akses AWS KMS kunci. Anda juga dapat membuat kontrol akses tingkat grup fitur.

Untuk informasi selengkapnya tentang kunci terkelola pelanggan, lihat kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang AWS KMS, lihat AWS KMS.

Menggunakan AWS KMS izin untuk Amazon SageMaker Feature Store

Enkripsi saat istirahat melindungi Toko Fitur di bawah kunci yang dikelola AWS KMS pelanggan. Secara default, ia menggunakan kunci yang dikelola pelanggan yang AWS dimiliki untuk OnlineStore dan kunci yang AWS dikelola pelanggan yang dikelola untuk OfflineStore. Feature Store mendukung opsi untuk mengenkripsi toko online atau offline Anda di bawah kunci yang dikelola pelanggan. Anda dapat memilih kunci yang dikelola pelanggan untuk Toko Fitur saat Anda membuat toko online atau offline, dan mereka dapat berbeda untuk setiap toko.

Feature Store hanya mendukung kunci yang dikelola pelanggan simetris. Anda tidak dapat menggunakan kunci yang dikelola pelanggan asimetris untuk mengenkripsi data Anda di toko online

atau offline Anda. Untuk bantuan menentukan apakah kunci yang dikelola pelanggan simetris atau asimetris, lihat Mengidentifikasi kunci yang dikelola pelanggan simetris dan asimetris.

Saat Anda menggunakan kunci yang dikelola pelanggan, Anda dapat memanfaatkan fitur-fitur berikut:

- Anda membuat dan mengelola kunci yang dikelola pelanggan, termasuk pengaturan kebijakan utama, kebijakan IAM, dan hibah untuk mengontrol akses ke kunci yang dikelola pelanggan. Anda dapat mengaktifkan dan menonaktifkan kunci yang dikelola pelanggan, mengaktifkan dan menonaktifkan rotasi kunci otomatis, dan menghapus kunci yang dikelola pelanggan bila tidak lagi digunakan.
- Anda dapat menggunakan kunci yang dikelola pelanggan dengan material kunci yang diimpor atau kunci yang dikelola pelanggan di penyimpanan kunci kustom yang Anda miliki dan kelola.
- Anda dapat mengaudit enkripsi dan dekripsi toko online atau offline Anda dengan memeriksa panggilan API ke dalam log. AWS KMSAWS CloudTrail

Anda tidak membayar biaya bulanan untuk kunci yang dikelola pelanggan yang AWS dimiliki. Kunci yang dikelola pelanggan akan dikenakan biaya untuk setiap panggilan API dan AWS Key Management Service kuota berlaku untuk setiap kunci yang dikelola pelanggan.

Mengotorisasi penggunaan Kunci yang dikelola pelanggan untuk toko online Anda

Jika Anda menggunakan kunci yang dikelola pelanggan untuk melindungi toko online Anda, kebijakan pada kunci yang dikelola pelanggan tersebut harus memberikan izin Toko Fitur untuk menggunakannya atas nama Anda. Anda memiliki kontrol penuh atas kebijakan dan hibah pada kunci yang dikelola pelanggan.

Toko Fitur tidak memerlukan otorisasi tambahan untuk menggunakan kunci KMS yang AWS dimiliki default untuk melindungi toko online atau offline Anda di akun AndaAWS.

Kebijakan kunci yang dikelola pelanggan

Bila Anda memilih kunci yang dikelola pelanggan untuk melindungi Toko Online Anda, Toko Fitur harus memiliki izin untuk menggunakan kunci yang dikelola pelanggan atas nama prinsipal yang membuat pilihan. Prinsipal tersebut, pengguna atau peran, harus memiliki izin pada kunci yang dikelola pelanggan yang diperlukan Feature Store. Anda dapat memberikan izin ini dalam kebijakan kunci, kebijakan IAM, atau pemberian izin. Minimal, Toko Fitur memerlukan izin berikut pada kunci yang dikelola pelanggan:

- “kms:Encrypt”, “kms:Decrypt”, “kms: “, “kms: DescribeKey “, “kms: CreateGrant “, “kms: RetireGrant “, “kms: “, ReEncryptFrom “kms: “, “kms: ReEncryptTo “, “kms:GenerateDataKey” ListAliases ListGrants RevokeGrant

Sebagai contoh, kebijakan kunci berikut hanya menyediakan izin yang diperlukan. Kebijakan ini memiliki efek sebagai berikut:

- Memungkinkan Toko Fitur untuk menggunakan kunci yang dikelola pelanggan dalam operasi kriptografi dan membuat hibah, tetapi hanya ketika bertindak atas nama prinsipal di akun yang memiliki izin untuk menggunakan Toko Fitur Anda. Jika prinsipal yang ditentukan dalam pernyataan kebijakan tidak memiliki izin untuk menggunakan Toko Fitur Anda, bahkan ketika itu berasal dari layanan Feature Store.
- Kunci ViaService kondisi kms memungkinkan izin hanya ketika permintaan berasal dari FeatureStore prinsipal yang tercantum dalam pernyataan kebijakan. Pengguna utama ini tidak dapat memanggil operasi ini secara langsung. Nilai untuk `kms:ViaService` seharusnya`sagemaker.*.amazonaws.com`.

  > **Note**
  >
  > Kunci `kms:ViaService` kondisi hanya dapat digunakan untuk AWS KMS kunci yang dikelola pelanggan toko online, dan tidak dapat digunakan untuk toko offline. Jika Anda menambahkan kondisi khusus ini ke kunci yang dikelola pelanggan Anda, dan menggunakan AWS KMS kunci yang sama untuk toko online dan offline, maka itu akan gagal operasi `CreateFeatureGroup` API.

- Memberikan administrator kunci yang dikelola pelanggan akses hanya-baca ke kunci yang dikelola pelanggan dan izin untuk mencabut hibah, termasuk hibah yang digunakan Toko Fitur untuk melindungi data Anda.

Sebelum menggunakan kebijakan kunci contoh, ganti contoh prinsipal dengan prinsipal aktual dari akun Anda. AWS

```
{"Id": "key-policy-feature-store",
   "Version":"2012-10-17",
   "Statement": [
     {"Sid" : "Allow access through Amazon SageMaker Feature Store for all principals
 in the account that are authorized to use  Amazon SageMaker Feature Store",
```

```
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:user/featurestore-user"},
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:RetireGrant",
        "kms:ReEncryptFrom",
        "kms:ReEncryptTo",
        "kms:GenerateDataKey",
        "kms:ListAliases",
        "kms:ListGrants"
      ],
      "Resource": "*",
      "Condition": {"StringLike": {"kms:ViaService" : "sagemaker.*.amazonaws.com"
         }
      }
    },
    {"Sid":  "Allow administrators to view the customer managed key and revoke
grants",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:role/featurestore-admin"
       },
      "Action": [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "kms:RevokeGrant"
      ],
      "Resource": "*"
    },
    {"Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
       "Principal": {"AWS": "arn:aws:iam::123456789:root"
       },
       "Action": "kms:*",
       "Resource": "*"
    }
  ]
}
```

Menggunakan hibah untuk mengotorisasi Toko Fitur

Selain kebijakan kunci, Feature Store menggunakan hibah untuk mengatur izin pada kunci yang dikelola pelanggan. Untuk melihat hibah pada kunci yang dikelola pelanggan di akun Anda, gunakan `ListGrants` operasi. Feature Store tidak memerlukan hibah, atau izin tambahan, untuk menggunakan kunci yang AWS dikelola pelanggan untuk melindungi toko online Anda.

Feature Store menggunakan izin hibah saat melakukan pemeliharaan sistem latar belakang dan tugas perlindungan data berkelanjutan.

Setiap hibah spesifik untuk toko online. Jika akun menyertakan beberapa toko yang dienkripsi di bawah kunci yang dikelola pelanggan yang sama, akan ada hibah unik per `FeatureGroup` menggunakan kunci yang dikelola pelanggan yang sama.

Kebijakan kunci juga dapat memungkinkan akun untuk mencabut hibah pada kunci yang dikelola pelanggan. Namun, jika Anda mencabut hibah pada toko online terenkripsi yang aktif, Feature Store tidak akan dapat melindungi dan memelihara toko.

Memantau interaksi Feature Store dengan AWS KMS

Jika Anda menggunakan kunci yang dikelola pelanggan untuk melindungi toko online atau offline Anda, Anda dapat menggunakan AWS CloudTrail log untuk melacak permintaan yang dikirimkan Feature Store atas nama Anda. AWS KMS

Mengakses data di toko online Anda

Penelepon (baik pengguna atau peran) ke SEMUA DataPlane operasi (Put, Get, DeleteRecord) harus memiliki izin di bawah ini pada kunci yang dikelola pelanggan:

```
"kms:Decrypt"
```

Mengotorisasi penggunaan kunci yang dikelola pelanggan untuk toko offline Anda

RoLearn yang diteruskan sebagai parameter `createFeatureGroup` harus memiliki izin di bawah ini untuk: OfflineStore KmsKeyId

```
"kms:GenerateDataKey"
```

> **Note**
>
> Kebijakan utama untuk toko online juga berfungsi untuk toko offline, hanya ketika `kms:ViaService` kondisinya tidak ditentukan.

> **Important**
>
> Anda dapat menentukan kunci AWS KMS enkripsi untuk mengenkripsi lokasi Amazon S3 yang digunakan untuk feature store offline saat membuat grup fitur. Jika kunci AWS KMS enkripsi tidak ditentukan, secara default kami mengenkripsi semua data saat istirahat menggunakan AWS KMS kunci. Dengan mendefinisikan kunci tingkat ember Anda untuk SSE, Anda dapat mengurangi biaya AWS KMS permintaan hingga 99 persen.

Kuota, aturan penamaan, dan tipe data

Terminologi kuota

- Read Request Unit (RRU): Ukuran throughput baca, di mana jumlah RRU per permintaan baca sama dengan plafon ukuran catatan baca yang dibagi menjadi potongan 4KB. RRU minimum per permintaan adalah 0.
- Write Request Unit (WRU): Ukuran throughput tulis, di mana jumlah WRU per permintaan tulis sama dengan plafon ukuran catatan tertulis yang dibagi menjadi potongan 1KB. WRU minimum per permintaan adalah 1 (termasuk operasi hapus).

Kuota dan Batas

> **Note**
>
> Batas lunak dapat ditingkatkan berdasarkan kebutuhan Anda.

- Jumlah maksimum grup fitur per AWS akun: Batas lunak 100.
- Jumlah maksimum definisi fitur per grup fitur: 2500.
- Jumlah maksimum RRU per pengidentifikasi catatan: 2400 RRU per detik.

- Jumlah maksimum WRU per pengenal catatan: 500 WRU per detik.
- Max Read Capacity Units (RCU) yang dapat disediakan pada satu grup fitur: 40000 RCU.
- Max Write Capacity Units (WCU) yang dapat disediakan pada satu grup fitur: 40000 WCU.
- Unit Kapasitas Baca Maks yang dapat disediakan di semua grup fitur di suatu wilayah: 80000 RCU.
- Unit Kapasitas Tulis Maks yang dapat disediakan di semua grup fitur di suatu wilayah: 80000 WCU.
- Transaksi Maksimum per detik (TPS) per API perAkun AWS: Batas lunak 10000 TPS per API tidak termasuk panggilan `BatchGetRecord` API, yang memiliki batas lunak 500 TPS.
- Ukuran maksimum catatan: 350KB.
- Ukuran maksimum pengenal catatan: 2KB.
- Ukuran maksimum nilai fitur: 350KB.
- Jumlah maksimum alur kerja pembuatan grup fitur bersamaan: 4.
- BatchGetRecord API: Dapat berisi sebanyak 100 catatan dan dapat menanyakan hingga 100 grup fitur.

Untuk informasi tentang layanan kuota dan cara meminta kenaikan kuota, lihat AWSlayanan.

Peraturan penamaan

- Kata Cadangan: Berikut ini adalah kata-kata yang dicadangkan dan tidak dapat digunakan sebagai nama fitur dalam definisi fitur:`is_deleted`,`write_time`, dan`api_invocation_time`.

Tipe Data

- String Tipe Fitur: String adalah Unicode dengan pengkodean biner UTF-8. Panjang minimum string dapat nol, panjang maksimum dibatasi oleh ukuran maksimum catatan.
- Jenis Fitur Pecahan: Nilai fitur pecahan harus sesuai dengan nomor floating point presisi ganda seperti yang didefinisikan oleh standar IEEE 754.
- Tipe Fitur Integral: Toko Fitur mendukung nilai integral dalam kisaran bilangan bulat bertanda 64-bit. Nilai minimum -2^{63} dan nilai maksimum: 2^{63} - 1.
- Fitur Waktu Acara: Semua grup fitur memiliki fitur waktu acara dengan presisi nanodetik. Setiap waktu peristiwa dengan presisi lebih rendah dari nanodetik akan menyebabkan ketidakcocokan ke belakang. Fitur ini dapat memiliki jenis fitur baik String atau Fractional.

- Waktu acara string diterima dalam format ISO-8601, dalam waktu UTC, sesuai dengan pola: [yyyy-mm-dd't'hh:mm:ssz, yyyy-mm-dd't'hh:mm:sssssssssz].
- Nilai waktu peristiwa pecahan diterima sebagai detik dari zaman unix. Waktu acara harus dalam kisaran [0000-01-01T 00:00:00.000 000000Z, 9999-12-31T 23:59:59.999 999999Z]. Untuk grup fitur dalam format `Iceberg` tabel, Anda hanya dapat menggunakan tipe String untuk waktu acara.

Amazon SageMaker Feature Store format data toko offline

Amazon SageMaker Feature Store mendukung format tabel Apache Iceberg AWS Glue dan Apache untuk toko offline. Anda dapat memilih format tabel saat membuat grup fitur baru. AWS Glueadalah format default.

Amazon SageMaker Feature Store data toko offline disimpan di bucket Amazon S3 di akun Anda. Saat Anda meneleponPutRecord, data Anda di-buffer, di-batch, dan ditulis ke Amazon S3 dalam waktu 15 menit. Feature Store hanya mendukung format file Parket saat menulis data Anda ke toko offline Anda. Khususnya, ketika data Anda ditulis ke toko offline Anda, data dapat diambil dari bucket Amazon S3 Anda dalam format Parket. Setiap file dapat berisi beberapa `Record` s.

Untuk format Iceberg, Feature Store menyimpan metadata tabel dalam bucket Amazon S3 yang sama yang Anda gunakan untuk menyimpan data toko offline. Anda dapat menemukannya di bawah `metadata` awalan.

Feature Store juga mengekspos OfflineStoreConfig.S3. StorageConfig ResolvedOutputBidang S3Uri, yang dapat ditemukan dari dalam panggilan DescribeFeatureGroupAPI. Ini adalah jalur S3 di mana file untuk grup fitur tertentu ditulis.

Bidang tambahan berikut ditambahkan oleh Feature Store ke setiap rekaman saat disimpan di toko offline:

- api_invocation_time — Stempel waktu saat layanan menerima panggilan atau. `PutRecord` `DeleteRecord` Jika menggunakan konsumsi terkelola (misalnya Data Wrangler), ini adalah stempel waktu saat data ditulis ke toko offline.
- write_time — Stempel waktu ketika data ditulis ke toko offline. Dapat digunakan untuk membangun kueri terkait perjalanan waktu.
- is_deleted - secara `False` default. Jika `DeleteRecord` dipanggil, yang baru `Record` dimasukkan ke dalam `RecordIdentifierValue` dan diatur ke `True` dalam toko offline.

Amazon SageMaker Feature Store struktur URI toko offline

Dalam contoh berikut *DOC-EXAMPLE-BUCKET* adalah bucket Amazon S3 dalam akun Anda, *example-prefix* adalah awalan contoh Anda, *111122223333* adalah ID akun Anda, *Wilayah AWS* adalah wilayah Anda, *feature-group-name* adalah nama grup fitur Anda.

AWS Glueformat tabel

Catatan di toko offline yang disimpan menggunakan format AWS Glue tabel dipartisi berdasarkan waktu acara menjadi partisi per jam. Anda tidak dapat mengkonfigurasi skema partisi. Struktur URI berikut menunjukkan organisasi file Parket menggunakan AWS Glue format:

```
s3://DOC-EXAMPLE-BUCKET/example-prefix/111122223333/sagemaker/Wilayah AWS/offline-
store/feature-group-name-feature-group-creation-time/data/year=year/month=month/
day=day/hour=hour/timestamp_of_latest_event_time_in_file_16-random-alphanumeric-
digits.parquet
```

Contoh berikut adalah lokasi output dari file Parket untuk file dengan *feature-group-name* ascustomer-purchase-history-patterns:

```
s3://DOC-EXAMPLE-BUCKET/example-prefix/111122223333/sagemaker/Wilayah AWS/offline-
store/customer-purchase-history-patterns-1593511200/data/year=2020/month=06/day=31/
hour=00/20200631T064401Z_108934320012Az11.parquet
```

Format tabel gunung es

Catatan di toko offline yang disimpan dalam format tabel Iceberg dipartisi berdasarkan waktu acara menjadi partisi harian. Anda tidak dapat mengkonfigurasi skema partisi. Struktur URI berikut menunjukkan organisasi file data yang disimpan dalam format tabel Iceberg:

```
s3://DOC-EXAMPLE-BUCKET/example-prefix/111122223333/sagemaker/Wilayah AWS/offline-
store/feature-group-name-feature-group-creation-time/data/8-random-alphanumeric-
digits/event-time-feature-name_trunc=event-time-year-event-time-month-event-time-day/
timestamp-of-latest-event-time-in-file_16-random-alphanumeric-digits.parquet
```

Contoh berikut adalah lokasi output dari file Parket untuk file dengan *feature-group-name* ascustomer-purchase-history-patterns, dan *event-time-feature-name* adalahEventTime:

```
s3://DOC-EXAMPLE-BUCKET/example-prefix/111122223333/sagemaker/Wilayah AWS/
offline-store/customer-purchase-history-patterns-1593511200/data/0aec19ca/
EventTime_trunc=2022-11-09/20221109T215231Z_yolTtpyuWbkaeGIl.parquet
```

Contoh berikut adalah lokasi file metadata untuk file data yang disimpan dalam format tabel Iceberg.

```
s3://DOC-EXAMPLE-BUCKET/example-prefix/111122223333/sagemaker/Wilayah AWS/offline-
store/feature-group-name-feature-group-creation-time/metadata/
```

Sumber daya Toko SageMaker Fitur Amazon

Berikut ini mencantumkan sumber daya yang tersedia untuk pengguna Amazon SageMaker Feature Store. Untuk halaman utama Toko Fitur, lihat Amazon SageMaker Feature Store.

Feature Store contoh notebook dan lokakarya

Untuk mulai menggunakan Amazon SageMaker Feature Store, Anda dapat memilih dari berbagai contoh notebook Jupyter dari tabel berikut. Jika ini adalah pertama kalinya Anda menggunakan Feature Store, cobalah buku catatan Pengantar ke Feature Store. Untuk menjalankan buku catatan ini, Anda harus melampirkan kebijakan ini ke peran eksekusi IAM Anda:. `AmazonSageMakerFeatureStoreAccess`

Lihat Peran IAM untuk mengakses peran Anda dan lampirkan kebijakan ini. Untuk panduan tentang cara melihat kebijakan yang dilampirkan pada peran dan cara menambahkan kebijakan ke peran Anda, lihat Menambahkan kebijakan ke peran IAM Anda.

Tabel berikut mencantumkan berbagai sumber daya untuk membantu Anda memulai dengan Toko Fitur. Tabel ini berisi contoh, instruksi, dan contoh buku catatan untuk memandu Anda dalam cara menggunakan Feature Store untuk pertama kalinya untuk kasus penggunaan tertentu. Kode dalam sumber daya ini menggunakan SageMaker SDK for Python (Boto3).

Halaman	Deskripsi
Mulai dengan Amazon SageMaker Feature Store di Baca Dokumen.	Daftar contoh notebook untuk memperkenalkan Anda ke Feature Store dan fitur-fiturnya untuk membantu Anda memulai.

Halaman	Deskripsi
Panduan Toko SageMaker Fitur Amazon di Baca Dokumen.	Panduan Toko Fitur tentang cara mengatur, membuat grup fitur, memuat data ke grup fitur, dan cara menggunakan Feature Store secara umum.
end-to-end Lokakarya Amazon SageMaker Feature Store di repositori `aws-samples` Github	Lokakarya Toko end-to-end Fitur.
Feature Store contoh notebook dalam SageMaker contoh repositori notebook.	Notebook contoh kasus penggunaan khusus untuk Feature Store.

Fitur Store Python SDK dan API

Python Software Development Kit (SDK) dan Application Programming Interface (API) adalah alat yang digunakan untuk membuat aplikasi perangkat lunak. Feature Store SDK for Python (Boto3) dan API tercantum dalam tabel berikut.

Halaman	Deskripsi
Fitur Store API di Amazon SageMaker Python SDK Baca Dokumen	API Toko Fitur di Baca Dokumen.
Fitur Store Python SDK di repositori Amazon Python SageMaker SDK Github	Penyimpanan Fitur Python SDK Github repositori.
Operasi Feature Store Runtime dan tipe data dalam dokumentasi SDK for Python (Boto3)	Klien Runtime Store Fitur yang berisi semua operasi API bidang data dan tipe data untuk Feature Store.
Runtime Toko SageMaker Fitur Amazon di Referensi Amazon SageMaker API	Beberapa tindakan tingkat grup fitur yang didukung oleh Feature Store. Jika operasi API atau tipe data yang Anda cari tidak tercantum di sini, silakan gunakan pencarian di panduan ini.

Halaman	Deskripsi
Runtime Toko SageMaker Fitur Amazon di Referensi Amazon SageMaker API	Rekam tindakan tingkat yang didukung oleh Feature Store. Jika operasi API atau tipe data yang Anda cari tidak tercantum di sini, silakan gunakan pencarian di panduan ini.

Melatih model pembelajaran mesin

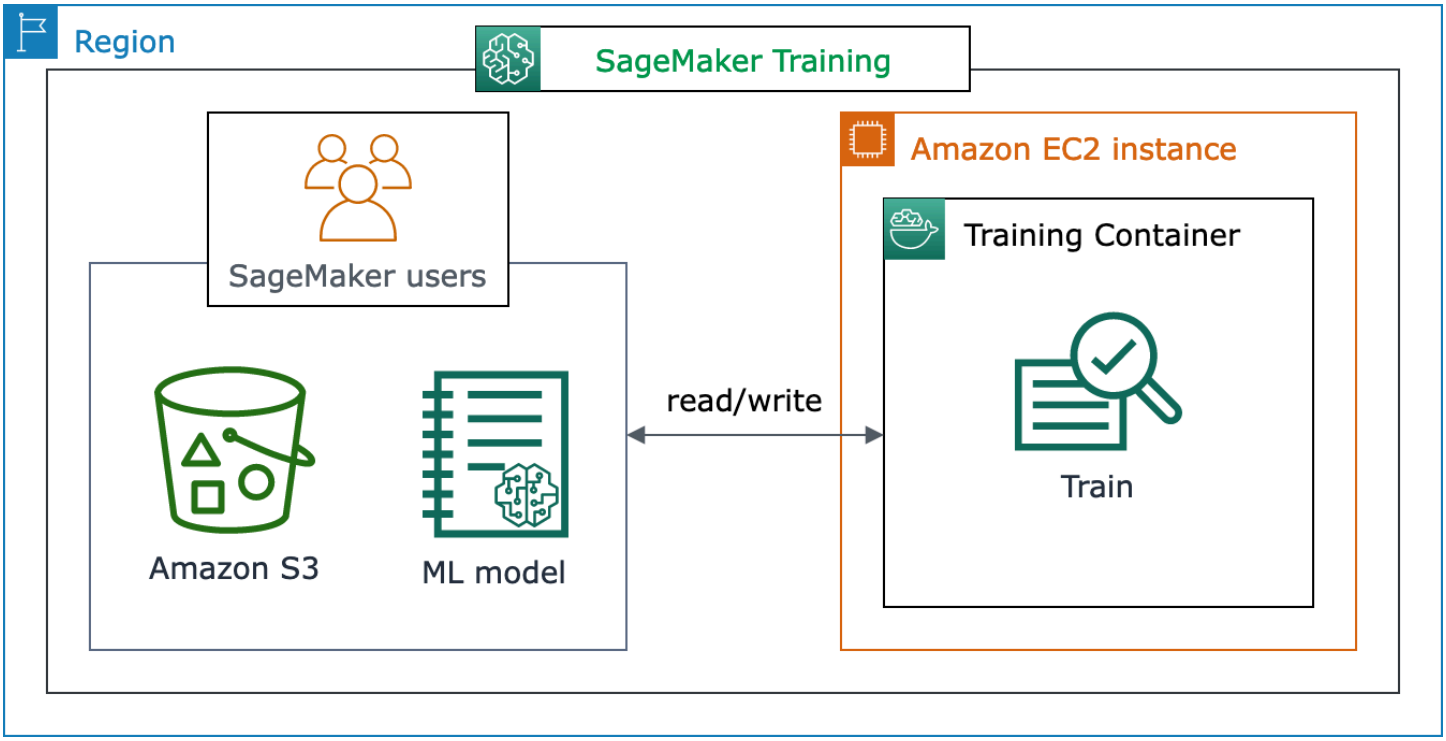
Tahap pelatihan siklus hidup pembelajaran mesin (ML) lengkap mencakup dari mengakses kumpulan data pelatihan Anda hingga menghasilkan model akhir dan memilih model berkinerja terbaik untuk penerapan. Bagian berikut memberikan gambaran umum tentang fitur SageMaker pelatihan yang tersedia dan sumber daya dengan informasi teknis yang mendalam untuk masing-masing.

Alur kerja pelatihan paling sederhana di SageMaker

Jika Anda menggunakan SageMaker untuk pertama kalinya dan ingin menemukan solusi ML cepat untuk melatih model pada kumpulan data Anda, pertimbangkan untuk menggunakan solusi tanpa kode atau kode rendah seperti SageMaker Canvas, SageMaker JumpStartdalam SageMaker Studio Classic, atau Autopilot. SageMaker

Untuk pengalaman pengkodean menengah, pertimbangkan untuk menggunakan notebook SageMaker Studio Classic atau Instans SageMaker Notebook. Untuk memulai, ikuti petunjuk di the section called “Langkah 4: Latih Model” panduan SageMaker Memulai. Kami merekomendasikan ini untuk kasus penggunaan di mana Anda membuat model dan skrip pelatihan Anda sendiri menggunakan kerangka kerja HTML.

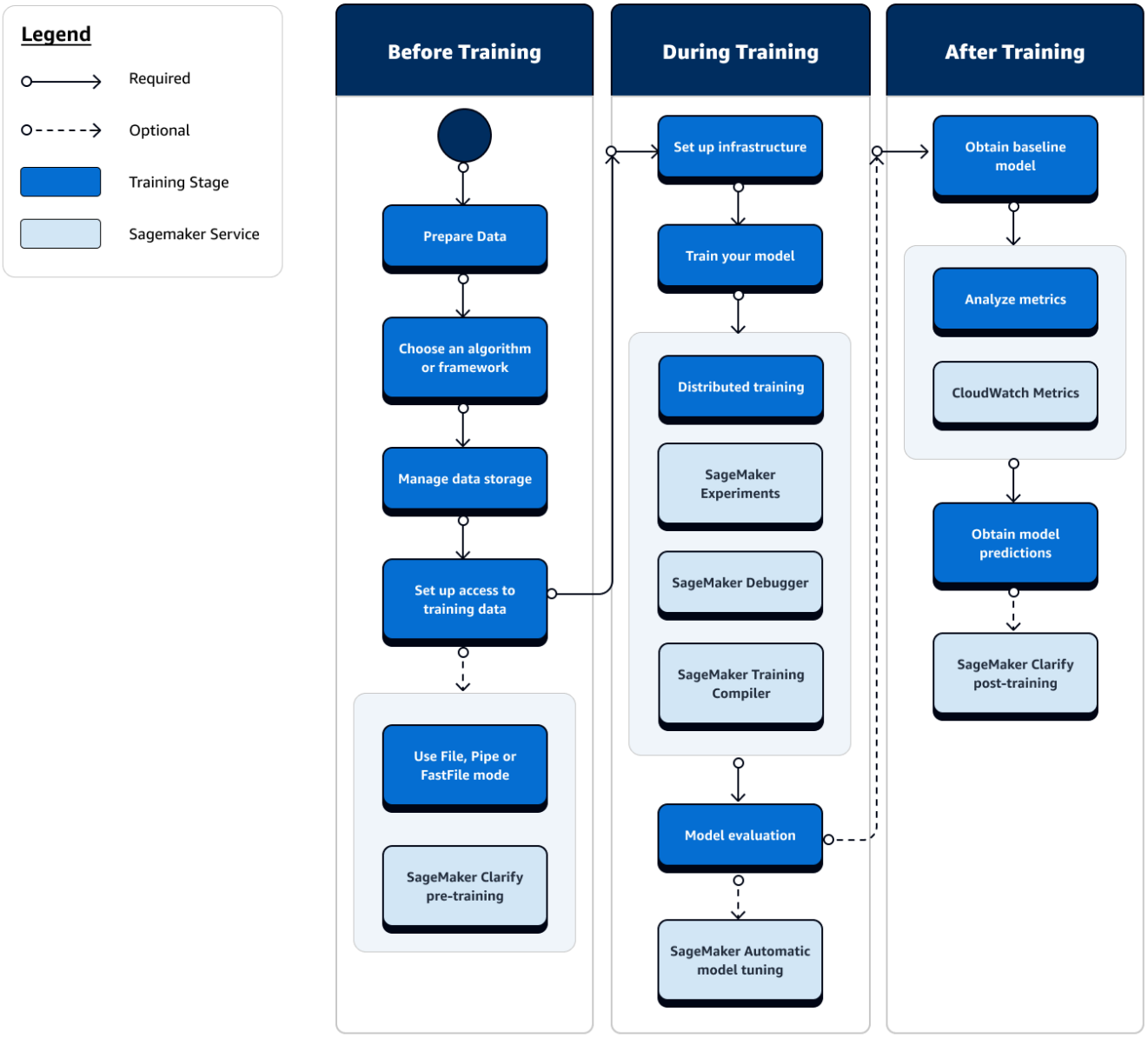
Diagram arsitektur berikut menunjukkan cara SageMaker mengelola pekerjaan pelatihan ML dan menyediakan instans Amazon EC2 atas nama pengguna. SageMaker Anda sebagai SageMaker pengguna dapat membawa kumpulan data pelatihan Anda sendiri, menyimpannya ke Amazon S3. Anda dapat memilih pelatihan model ML dari algoritme SageMaker bawaan yang tersedia, atau membawa skrip pelatihan Anda sendiri dengan model yang dibuat dengan kerangka kerja pembelajaran mesin yang populer.



Tampilan penuh alur kerja dan fitur SageMaker Pelatihan

Perjalanan penuh pelatihan ML melibatkan tugas-tugas di luar konsumsi data ke model ML, model pelatihan tentang instance komputasi, dan memperoleh artefak dan output model. Anda perlu mengevaluasi setiap fase sebelum, selama, dan setelah pelatihan untuk memastikan model Anda dilatih dengan baik untuk memenuhi akurasi target untuk tujuan Anda.

Diagram alir berikut menunjukkan ikhtisar tingkat tinggi dari tindakan Anda (dalam kotak biru) dan fitur SageMaker Pelatihan yang tersedia (dalam kotak biru muda) selama fase pelatihan siklus hidup ML.



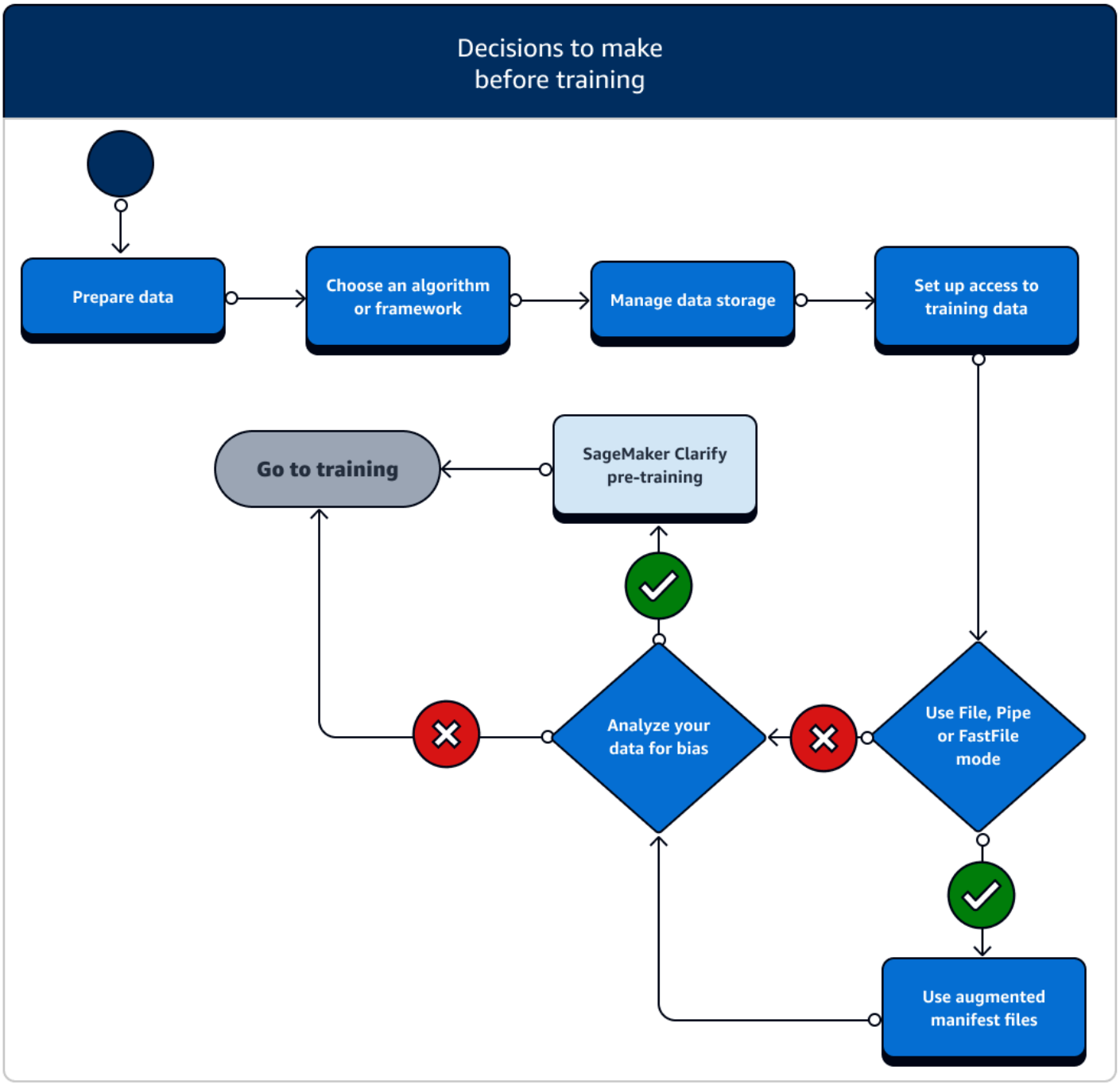
Bagian-bagian berikut memandu Anda melalui setiap fase pelatihan yang digambarkan dalam diagram alir sebelumnya dan fitur berguna yang ditawarkan oleh SageMaker seluruh tiga sub-tahap pelatihan ML.

Topik

- Sebelum pelatihan
- Selama pelatihan
- Setelah pelatihan

Sebelum pelatihan

Ada sejumlah skenario pengaturan sumber daya data dan akses yang perlu Anda pertimbangkan sebelum pelatihan. Lihat diagram berikut dan detail dari setiap tahap sebelum pelatihan untuk memahami keputusan apa yang perlu Anda buat.



- Siapkan data: Sebelum pelatihan, Anda harus menyelesaikan pembersihan data dan rekayasa fitur selama tahap persiapan data. SageMaker memiliki beberapa pelabelan dan alat rekayasa

fitur untuk membantu Anda. Lihat Label Data, Mempersiapkan dan Menganalisis Kumpulan Data, Memproses Data, dan Membuat, Menyimpan, dan Berbagi Fitur untuk informasi selengkapnya.

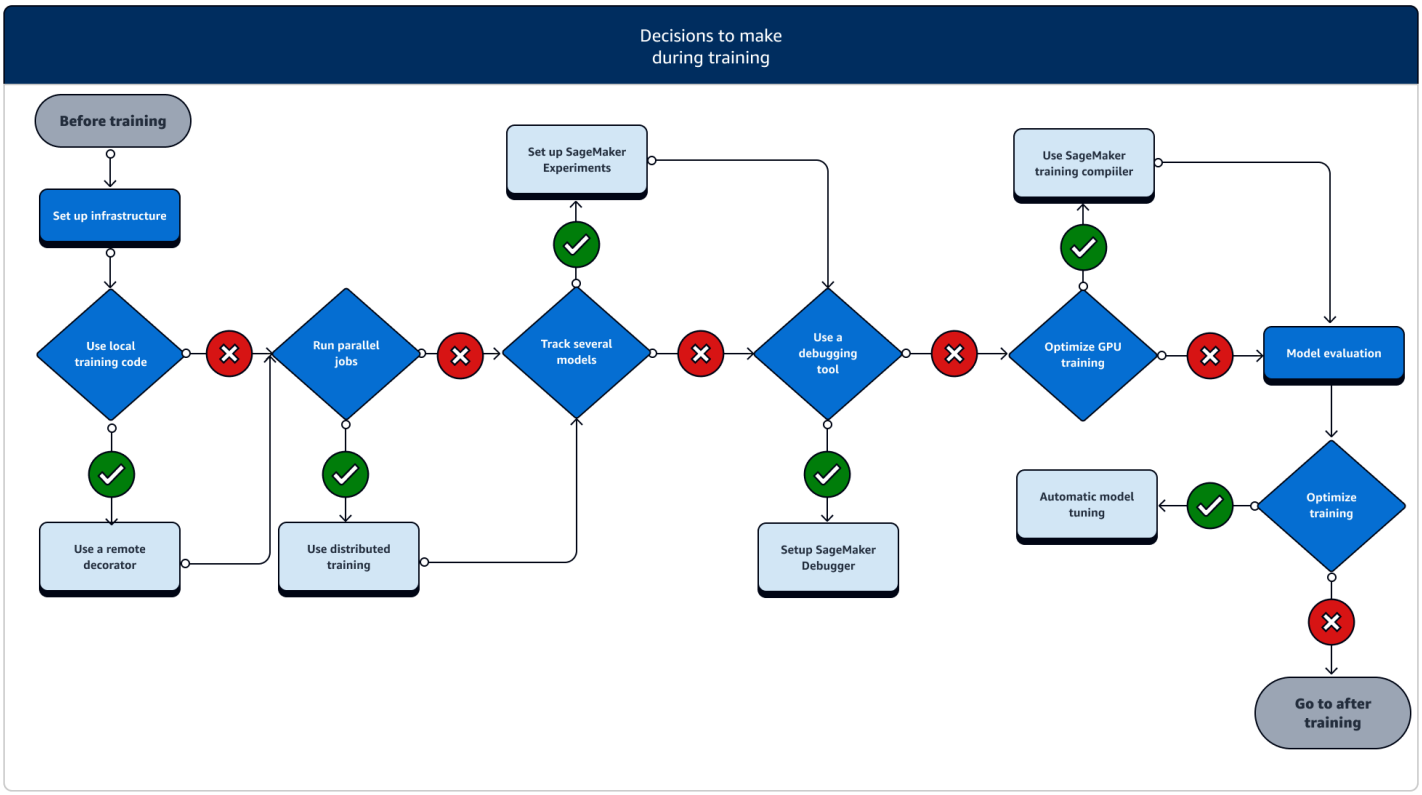
- Pilih algoritma atau kerangka kerja: Bergantung pada seberapa banyak penyesuaian yang Anda butuhkan, ada opsi berbeda untuk algoritme dan kerangka kerja.
  - Jika Anda lebih suka implementasi kode rendah dari algoritma pra-bangun, gunakan salah satu algoritma bawaan yang ditawarkan oleh. SageMaker Untuk informasi selengkapnya, lihat Memilih Algoritma.
  - Jika Anda membutuhkan lebih banyak fleksibilitas untuk menyesuaikan model Anda, jalankan skrip pelatihan Anda menggunakan kerangka kerja dan toolkit pilihan Anda di dalamnya. SageMaker Untuk informasi selengkapnya, lihat Kerangka Kerja dan Toolkit ML.
  - Untuk memperluas image SageMaker Docker yang sudah dibuat sebelumnya sebagai gambar dasar wadah Anda sendiri, lihat Menggunakan gambar Docker yang sudah dibuat sebelumnya SageMaker .
  - Untuk membawa kontainer Docker kustom Anda SageMaker, lihat Mengadaptasi wadah Docker Anda sendiri untuk bekerja dengannya. SageMaker Anda perlu menginstal sagemaker-training-toolkitke wadah Anda.
- Mengelola penyimpanan data: Memahami pemetaan antara penyimpanan data (seperti Amazon S3, Amazon EFS, atau Amazon FSx) dan wadah pelatihan yang berjalan di instans komputasi Amazon EC2. SageMaker membantu memetakan jalur penyimpanan dan jalur lokal dalam wadah pelatihan. Anda juga dapat menentukannya secara manual. Setelah pemetaan selesai, pertimbangkan untuk menggunakan salah satu mode transmisi data: File, Pipa, dan FastFile mode. Untuk mempelajari cara SageMaker memetakan jalur penyimpanan, lihat Melatih Folder Penyimpanan.
- Siapkan akses ke data pelatihan: Gunakan SageMaker Domain Amazon, profil pengguna Domain, IAM, Amazon VPC, AWS KMS dan untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.
  - Untuk administrasi akun, lihat SageMaker Domain Amazon.
  - Untuk referensi lengkap tentang kebijakan dan keamanan IAM, lihat Keamanan di Amazon SageMaker.
- Streaming data input Anda: SageMaker menyediakan tiga mode input data, File, Pipe, dan FastFile. Mode input default adalah mode File, yang memuat seluruh kumpulan data selama menginisialisasi pekerjaan pelatihan. Untuk mempelajari praktik terbaik umum untuk streaming data dari penyimpanan data Anda ke wadah pelatihan, lihat Mengakses Data Pelatihan.

Dalam kasus mode Pipe, Anda juga dapat mempertimbangkan untuk menggunakan file manifes tambahan untuk mengalirkan data Anda langsung dari Amazon Simple Storage Service (Amazon S3) dan melatih model Anda. Menggunakan mode pipa mengurangi ruang disk karena Amazon Elastic Block Store hanya perlu menyimpan artefak model akhir Anda, daripada menyimpan kumpulan data pelatihan lengkap Anda. Untuk informasi selengkapnya, lihat Menyediakan Metadata Set Data ke Pekerjaan Pelatihan dengan File Manifes Tertambah.

- Analisis data Anda untuk bias: Sebelum pelatihan, Anda dapat menganalisis kumpulan data dan memodelkan bias terhadap grup yang tidak disukai sehingga Anda dapat memeriksa apakah model Anda mempelajari kumpulan data yang tidak bias menggunakan Clarify. SageMaker
- Pilih SageMaker SDK mana yang akan digunakan: Ada dua cara untuk meluncurkan pekerjaan pelatihan SageMaker: menggunakan SDK SageMaker Python tingkat tinggi, atau menggunakan SageMaker API tingkat rendah untuk SDK for Python (Boto3) atau. AWS CLI SageMaker Python SDK mengabstraksi SageMaker API tingkat rendah untuk menyediakan alat yang nyaman. Seperti disebutkan di atasthe section called “Alur kerja pelatihan paling sederhana di SageMaker”, Anda juga dapat mengejar opsi tanpa kode atau kode minimal menggunakan SageMaker Canvas, SageMaker JumpStart dalam SageMaker Studio Classic, atau Autopilot. SageMaker

Selama pelatihan

Selama pelatihan, Anda perlu terus meningkatkan stabilitas pelatihan, kecepatan pelatihan, efisiensi pelatihan sambil menskalakan sumber daya komputasi, pengoptimalan biaya, dan, yang paling penting, kinerja model. Baca terus untuk informasi lebih lanjut tentang tahapan pelatihan selama dan fitur SageMaker Pelatihan yang relevan.



- Siapkan infrastruktur: Pilih jenis instans dan alat manajemen infrastruktur yang tepat untuk kasus penggunaan Anda. Anda dapat memulai dari contoh kecil dan meningkatkan tergantung pada beban kerja Anda. Untuk melatih model pada kumpulan data tabular, mulailah dengan instance CPU terkecil dari keluarga instance C4 atau C5. Untuk melatih model besar untuk visi komputer atau pemrosesan bahasa alami, mulailah dengan instance GPU terkecil dari keluarga instance P2, P3, G4dn atau G5. Anda juga dapat mencampur berbagai jenis instance dalam klaster, atau menyimpan instance di kolam hangat menggunakan alat manajemen instance berikut yang ditawarkan oleh SageMaker. Anda juga dapat menggunakan cache persisten untuk mengurangi latensi dan waktu yang dapat ditagih pada pekerjaan pelatihan berulang selama pengurangan latensi dari kolam hangat saja. Untuk mempelajari lebih lanjut, lihat topik berikut.
  - Melatih Menggunakan Cluster Heterogen
  - Melatih Menggunakan Kolam Hangat yang SageMaker Dikelola
  - Menggunakan cache persisten

  Anda harus memiliki kuota yang cukup untuk menjalankan pekerjaan pelatihan. Jika Anda menjalankan pekerjaan pelatihan Anda pada contoh di mana Anda memiliki kuota yang tidak mencukupi, Anda akan menerima `ResourceLimitExceeded` kesalahan. Untuk memeriksa kuota yang tersedia saat ini di akun Anda, gunakan konsol Service Quotas Anda. Untuk mempelajari cara

meminta peningkatan kuota, lihat Wilayah dan Kuota yang Didukung. Juga, untuk menemukan informasi harga dan jenis instans yang tersedia tergantung padaWilayah AWS, cari tabel di halaman SageMaker Harga Amazon.

- Jalankan pekerjaan pelatihan dari kode lokal: Anda dapat membuat anotasi kode lokal Anda dengan dekorator jarak jauh untuk menjalankan kode Anda sebagai pekerjaan SageMaker pelatihan dari dalam Amazon SageMaker Studio Classic, SageMaker notebook Amazon, atau dari lingkungan pengembangan terintegrasi lokal Anda. Untuk informasi selengkapnya, lihat Jalankan kode lokal Anda sebagai pekerjaan SageMaker pelatihan.
- Lacak pekerjaan pelatihan: Pantau dan lacak pekerjaan pelatihan Anda menggunakan SageMaker Eksperimen, SageMaker Debugger, atau Amazon. CloudWatch Anda dapat menonton kinerja model dalam hal akurasi dan konvergensi, dan menjalankan analisis komparatif metrik antara beberapa pekerjaan pelatihan dengan menggunakan Eksperimen. SageMaker Anda dapat menonton tingkat pemanfaatan sumber daya komputasi dengan menggunakan alat profil SageMaker Debugger atau Amazon. CloudWatch Untuk mempelajari lebih lanjut, lihat topik berikut.
  - Kelola Machine Learning dengan Amazon SageMaker Experiments
  - Pekerjaan Pelatihan Profil Menggunakan Amazon SageMaker Debugger
  - Memantau dan Menganalisis Menggunakan CloudWatch Metrik

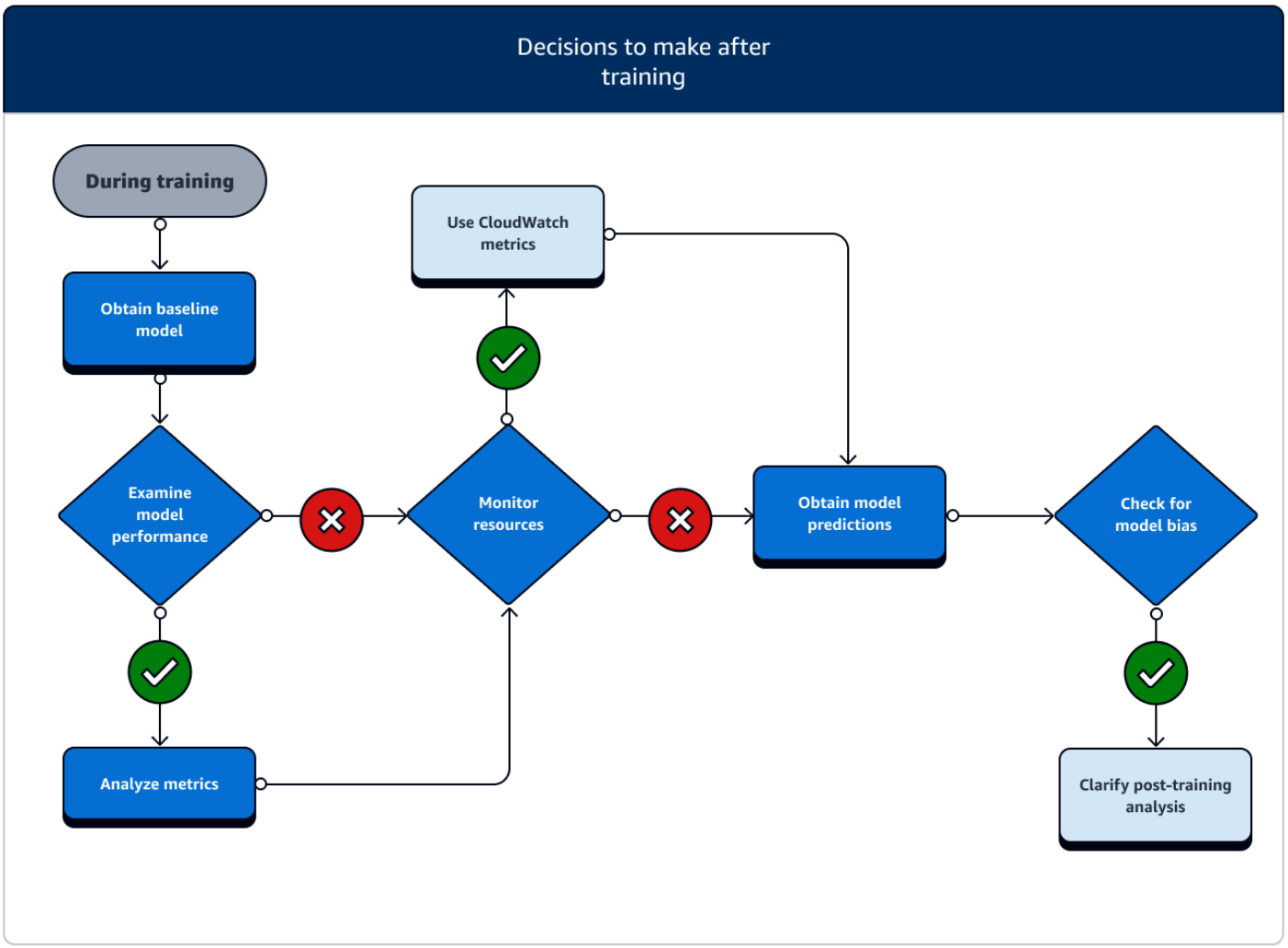
  Selain itu, untuk tugas pembelajaran mendalam, gunakan alat SageMaker debugging model Amazon Debugger dan aturan bawaan untuk mengidentifikasi masalah yang lebih kompleks dalam proses konvergensi model dan pembaruan bobot.
- Pelatihan terdistribusi: Jika pekerjaan pelatihan Anda memasuki tahap yang stabil tanpa putus karena kesalahan konfigurasi infrastruktur atau out-of-memory masalah pelatihan, Anda mungkin ingin menemukan lebih banyak opsi untuk meningkatkan skala pekerjaan Anda dan menjalankan dalam jangka waktu yang lama selama berhari-hari dan bahkan berbulan-bulan. Saat Anda siap untuk meningkatkan, pertimbangkan pelatihan terdistribusi. SageMaker menyediakan berbagai opsi untuk komputasi terdistribusi dari beban kerja ML ringan hingga beban kerja pembelajaran mendalam yang berat.

  Untuk tugas pembelajaran mendalam yang melibatkan pelatihan model yang sangat besar pada kumpulan data yang sangat besar, pertimbangkan untuk menggunakan salah satu strategi pelatihan SageMaker terdistribusi untuk meningkatkan dan mencapai paralelisme data, paralelisme model, atau kombinasi keduanya. Anda juga dapat menggunakan SageMaker Training Compiler untuk mengkompilasi dan mengoptimalkan grafik model pada instance GPU. SageMaker Fitur-fitur ini mendukung kerangka pembelajaran mendalam seperti PyTorch, TensorFlow, dan Hugging Face Transformers.

- Penyetelan hiperparameter model: Setel hiperparameter model Anda menggunakan Penyetelan Model Otomatis dengan. SageMaker SageMaker menyediakan metode penyetelan hiperparameter seperti pencarian kisi dan pencarian Bayesian, meluncurkan pekerjaan penyetelan hiperparameter paralel dengan fungsionalitas penghentian awal untuk pekerjaan penyetelan hiperparameter yang tidak meningkatkan.
- Checkpointing dan penghematan biaya dengan instans Spot: Jika waktu pelatihan bukan masalah besar, Anda dapat mempertimbangkan untuk mengoptimalkan biaya pelatihan model dengan instans Spot terkelola. Perhatikan bahwa Anda harus mengaktifkan checkpointing untuk pelatihan Spot agar tetap memulihkan dari jeda pekerjaan intermiten karena penggantian instans Spot. Anda juga dapat menggunakan fungsionalitas pos pemeriksaan untuk mencadangkan model Anda jika terjadi pemutusan hubungan kerja pelatihan yang tidak terduga. Untuk mempelajari lebih lanjut, lihat topik berikut.
  - Pelatihan Spot Terkelola
  - Gunakan Checkpoints

Setelah pelatihan

Setelah pelatihan, Anda mendapatkan artefak model akhir untuk digunakan untuk penerapan dan inferensi model. Ada tindakan tambahan yang terlibat dalam fase setelah pelatihan seperti yang ditunjukkan pada diagram berikut.

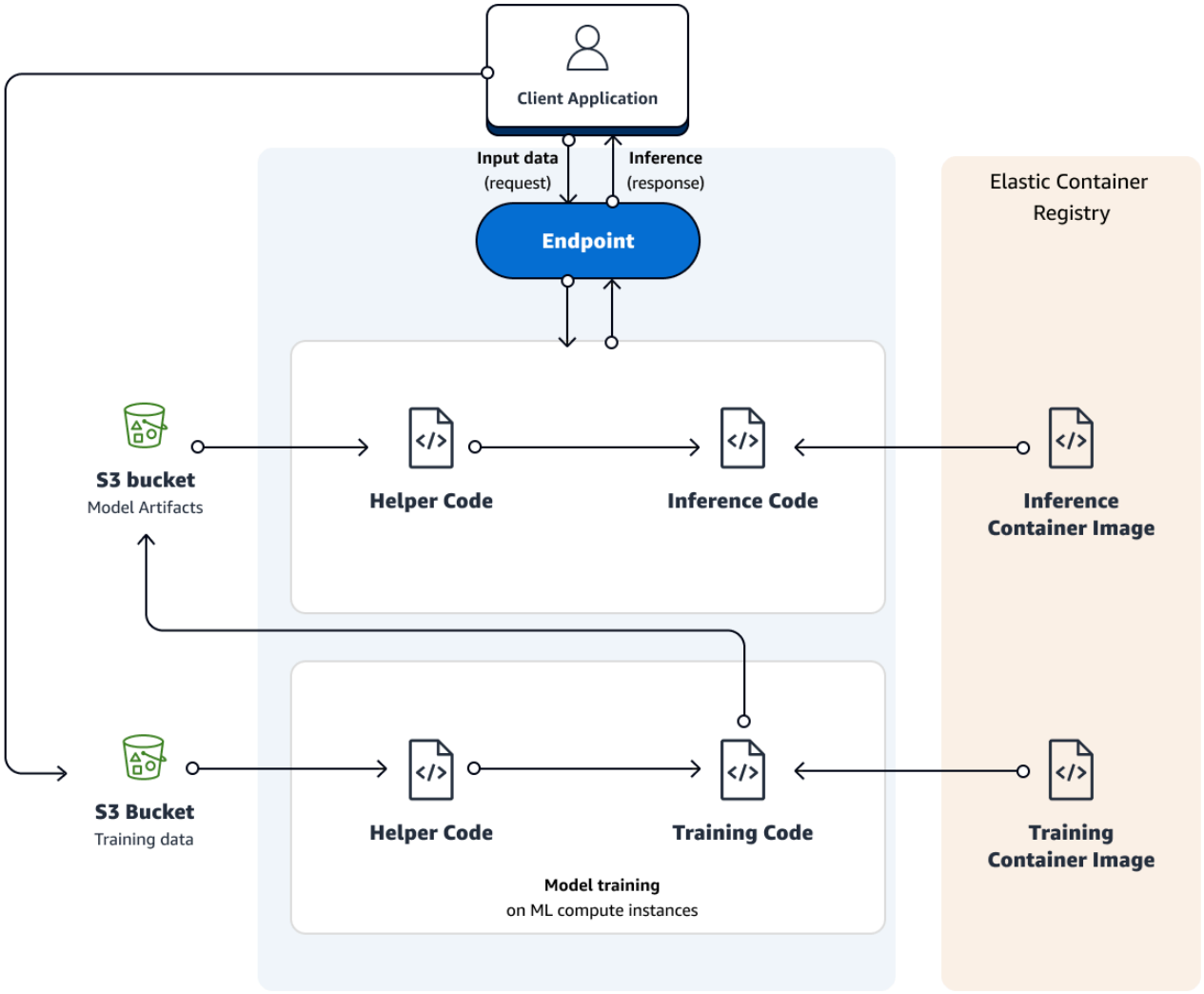


- Dapatkan model dasar: Setelah Anda memiliki artefak model, Anda dapat mengaturnya sebagai model dasar. Pertimbangkan tindakan pasca-pelatihan berikut dan gunakan SageMaker fitur sebelum beralih ke penerapan model ke produksi.
- Periksa kinerja model dan periksa bias: Gunakan CloudWatch Metrik Amazon dan SageMaker Klarifikasi untuk bias pasca-pelatihan untuk mendeteksi bias apa pun dalam data dan model yang masuk dari waktu ke waktu terhadap baseline. Anda perlu mengevaluasi data baru dan prediksi model Anda terhadap data baru secara teratur atau secara real time. Dengan menggunakan fitur-fitur ini, Anda dapat menerima peringatan tentang perubahan atau anomali akut, serta perubahan bertahap atau penyimpangan dalam data dan model.
- Anda juga dapat menggunakan fungsionalitas Pelatihan Inkremental SageMaker untuk memuat dan memperbarui model Anda (atau menyempurnakan) dengan kumpulan data yang diperluas.

- Anda dapat mendaftarkan pelatihan model sebagai langkah dalam SageMakerPipeline Anda atau sebagai bagian dari fitur Alur Kerja lain yang ditawarkan oleh SageMaker untuk mengatur siklus hidup ML penuh.

Latih Model dengan Amazon SageMaker

Diagram berikut menunjukkan bagaimana Anda melatih dan menerapkan model dengan Amazon SageMaker. Kode pelatihan Anda mengakses data pelatihan Anda dan mengeluarkan artefak model dari bucket S3. Kemudian Anda dapat membuat permintaan ke titik akhir model untuk menjalankan inferensi. Anda dapat menyimpan gambar wadah pelatihan dan inferensi di Amazon Elastic Container Registry (ECR).



Panduan berikut menyoroti dua komponen SageMaker: pelatihan model dan penerapan model.

Untuk melatih model SageMaker, Anda membuat pekerjaan pelatihan. Pekerjaan pelatihan mencakup informasi berikut:

- URL bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) tempat Anda menyimpan data pelatihan.
- Sumber daya komputasi yang ingin Anda gunakan SageMaker untuk pelatihan model. Sumber daya komputasi adalah instance komputasi pembelajaran mesin (ML) yang dikelola oleh. SageMaker
- URL bucket S3 tempat Anda ingin menyimpan output pekerjaan.
- Jalur Amazon Elastic Container Registry tempat kode pelatihan disimpan. Untuk informasi selengkapnya, lihat Jalur Registri Docker dan Kode Contoh.

> **Note**
>
> Dataset input Anda harus Wilayah AWS sama dengan pekerjaan pelatihan Anda.

Anda memiliki opsi berikut untuk algoritma pelatihan:

- Gunakan algoritme yang disediakan oleh SageMaker — SageMaker menyediakan lusinan algoritma pelatihan bawaan dan ratusan model yang telah dilatih sebelumnya. Jika salah satu dari ini memenuhi kebutuhan Anda, ini adalah out-of-the-box solusi yang bagus untuk pelatihan model cepat. Untuk daftar algoritma yang disediakan oleh SageMaker, lihatGunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih. Untuk mencoba latihan yang menggunakan algoritma yang disediakan oleh SageMaker, lihatMemulai. Anda juga dapat menggunakan SageMaker JumpStart algoritme dan model melalui UI Studio Classic.
- Gunakan SageMaker Debugger —untuk memeriksa parameter dan data pelatihan selama proses pelatihan saat bekerja dengan kerangka kerja pembelajaran TensorFlow,, PyTorch dan Apache MXNet atau algoritma XGBoost. Debugger secara otomatis mendeteksi dan memperingatkan pengguna untuk kesalahan yang umum terjadi seperti nilai parameter menjadi terlalu besar atau kecil. Untuk informasi selengkapnya tentang menggunakan Debugger, lihat. Gunakan Amazon SageMaker Debugger untuk men-debug dan meningkatkan kinerja model Notebook sampel debugger tersedia di Amazon SageMaker Debugger Sampel.
- Gunakan Apache Spark dengan SageMaker — SageMaker menyediakan perpustakaan yang dapat Anda gunakan di Apache Spark untuk melatih model. SageMaker Menggunakan

perpustakaan yang disediakan oleh SageMaker mirip dengan menggunakan Apache Spark MLLib. Untuk informasi selengkapnya, lihat Gunakan Apache Spark dengan Amazon SageMaker.

- Kirim kode kustom untuk dilatih dengan kerangka pembelajaran mendalam —Anda dapat mengirimkan kode Python kustom yang menggunakan TensorFlow,, PyTorch atau Apache MXNet untuk pelatihan model. Lihat informasi selengkapnya di Gunakan TensorFlow dengan Amazon SageMaker, Gunakan PyTorch dengan Amazon SageMaker, dan Gunakan Apache MxNet dengan Amazon SageMaker.
- Gunakan algoritme kustom Anda sendiri —Letakkan kode Anda bersama-sama sebagai gambar Docker dan tentukan jalur registri gambar dalam panggilan API. SageMaker `CreateTrainingJob` Untuk informasi selengkapnya, lihat Gunakan kontainer Docker untuk membuat model.
- Gunakan algoritma yang Anda berlangganan dari AWS Marketplace —Untuk informasi, lihatTemukan dan Berlangganan Algoritma dan Paket Model di AWS Marketplace.

Setelah Anda membuat pekerjaan pelatihan, SageMaker luncurkan instance komputasi ML dan gunakan kode pelatihan dan kumpulan data pelatihan untuk melatih model. Ini menyimpan artefak model yang dihasilkan dan output lainnya dalam ember S3 yang Anda tentukan untuk tujuan itu.

Anda dapat membuat pekerjaan pelatihan dengan SageMaker konsol atau API. Untuk informasi tentang membuat pekerjaan pelatihan dengan API, lihat `CreateTrainingJob`API.

Saat Anda membuat tugas pelatihan dengan API, SageMaker mereplikasi seluruh kumpulan data pada instance komputasi HTML secara default. Untuk membuat SageMaker replikasi subset data pada setiap instance komputasi ML. Anda harus menyetel field ke`S3DataDistributionType`. `ShardedByS3Key` Anda dapat mengatur bidang ini menggunakan SDK tingkat rendah. Untuk informasi selengkapnya, lihat `S3DataDistributionType` di `S3DataSource`.

> **Important**
>
> Untuk mencegah container algoritme Anda bersaing untuk memori, kami menyimpan memori untuk proses sistem SageMaker penting kami pada instance komputasi ML Anda dan oleh karena itu Anda tidak dapat mengharapkan untuk melihat semua memori untuk jenis instans Anda.

Pilih algoritme

Pembelajaran mesin dapat membantu Anda menyelesaikan tugas empiris yang memerlukan semacam inferensi induktif. Tugas ini melibatkan induksi karena menggunakan data untuk melatih algoritme untuk membuat kesimpulan yang dapat digeneralisasi. Ini berarti bahwa algoritme dapat membuat prediksi atau keputusan yang dapat diandalkan secara statistik, atau menyelesaikan tugas lain ketika diterapkan pada data baru yang tidak digunakan untuk melatihnya.

Untuk membantu Anda memilih algoritma terbaik untuk tugas Anda, kami mengklasifikasikan tugas-tugas ini pada berbagai tingkat abstraksi. Pada tingkat abstraksi tertinggi, pembelajaran mesin mencoba menemukan pola atau hubungan antara fitur atau item yang kurang terstruktur, seperti teks dalam kumpulan data. Teknik pengenalan pola dapat diklasifikasikan ke dalam paradigma pembelajaran mesin yang berbeda, yang masing-masing membahas jenis masalah tertentu. Saat ini ada tiga paradigma dasar untuk pembelajaran mesin yang digunakan untuk mengatasi berbagai jenis masalah:

- Pembelajaran yang diawasi
- Pembelajaran tanpa pengawasan
- Pembelajaran penguatan

Jenis masalah yang dapat diatasi oleh setiap paradigma pembelajaran diidentifikasi dengan mempertimbangkan kesimpulan (atau prediksi, keputusan, atau tugas lain) yang ingin Anda buat dari jenis data yang Anda miliki atau dapat kumpulkan. Paradigma pembelajaran mesin menggunakan metode algoritmik untuk mengatasi berbagai jenis masalah mereka. Algoritma menyediakan resep untuk memecahkan masalah ini.

Namun, banyak algoritma, seperti jaringan saraf, dapat digunakan dengan paradigma pembelajaran yang berbeda dan pada berbagai jenis masalah. Beberapa algoritma juga dapat mengatasi jenis masalah tertentu. Beberapa algoritma lebih umum berlaku dan yang lain cukup spesifik untuk jenis tertentu dari tujuan dan data. Jadi pemetaan antara algoritma pembelajaran mesin dan jenis masalah adalah many-to-many. Juga, ada berbagai opsi implementasi yang tersedia untuk algoritma.

Bagian berikut memberikan panduan mengenai opsi implementasi, paradigma pembelajaran mesin, dan algoritma yang sesuai untuk berbagai jenis masalah.

Topik

- Pilih implementasi algoritma

- Jenis masalah untuk paradigma pembelajaran mesin dasar
- Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih
- Gunakan Pembelajaran Penguatan dengan Amazon SageMaker

Pilih implementasi algoritma

Setelah memilih algoritma, Anda harus memutuskan implementasi mana yang ingin Anda gunakan. Amazon SageMaker mendukung tiga opsi implementasi yang membutuhkan peningkatan tingkat upaya.

- Model pra-terlatihmembutuhkan sedikit usaha dan model siap untuk menyebarkan atau untuk menyempurnakan dan menyebarkan menggunakan SageMaker JumpStart.
- algoritme bawaanmembutuhkan lebih banyak usaha dan skala jika kumpulan data besar dan sumber daya yang signifikan diperlukan untuk melatih dan menyebarkan model.
- Jika tidak ada solusi built-in yang berfungsi, cobalah untuk mengembangkan solusi yang menggunakangambar pra-dibuat untuk mesin dan kerangka pembelajaran mendalamuntuk kerangka kerja yang didukung seperti Scikit-Learn, TensorFlow, PyTorch, MXNet, atau Chainer.
- Jika Anda perlu menjalankan paket kustom atau menggunakan kode apa pun yang bukan bagian dari kerangka kerja yang didukung atau tersedia melalui PyPi, maka Anda perlu membangungambar Docker kustom Anda sendiriyang dikonfigurasi untuk menginstal paket atau perangkat lunak yang diperlukan. Gambar kustom juga harus didorong ke repositori online seperti Amazon Elastic Container Registry.

Topik

- Gunakan algoritme bawaan
- Gunakan mode skrip dalam kerangka kerja yang didukung
- Menggunakan image Docker kustom

Panduan implementasi algoritme

Implementasi	Membutuhkan kode	Pra-kode algoritme	Support untuk paket pihak ketiga	Support untuk kode kustom	Tingkat upaya
bawaan	Tidak	Ya	Tidak	Tidak	Rendah

Implementasi	Membutuhkan kode	Pra-kode algoritme	Support untuk paket pihak ketiga	Support untuk kode kustom	Tingkat upaya
Scikit-belajar	Ya	Ya	PyPi cuma	Ya	Sedang
Percikan	Ya	Ya	PyPi cuma	Ya	Sedang
XGBoost (open source)	Ya	Ya	PyPi cuma	Ya	Sedang
TensorFlow	Ya	Tidak	PyPi cuma	Ya	Tinggi medium
PyTorch	Ya	Tidak	PyPi cuma	Ya	Tinggi medium
MXNet	Ya	Tidak	PyPi cuma	Ya	Tinggi medium
Chainer	Ya	Tidak	PyPi cuma	Ya	Tinggi medium
Citra kustom	Ya	Tidak	Ya, dari sumber mana pun	Ya	Tinggi

Gunakan algoritme bawaan

Saat memilih algoritme untuk jenis masalah dan data Anda, opsi termudah adalah menggunakan salah satu Amazon SageMakerbuilt-in algoritma. Algoritme bawaan ini hadir dengan dua manfaat utama.

- Algoritma bawaan tidak memerlukan pengkodean untuk mulai menjalankan eksperimen. Satu-satunya input yang perlu Anda berikan adalah data, hyperparameter, dan sumber daya komputasi. Hal ini memungkinkan Anda untuk menjalankan eksperimen lebih cepat, dengan lebih sedikit overhead untuk melacak hasil dan perubahan kode.

- Algoritma bawaan dilengkapi dengan paralelisasi di beberapa instance komputasi dan dukungan GPU langsung dari kotak untuk semua algoritma yang berlaku (beberapa algoritma mungkin tidak disertakan karena keterbatasan yang melekat). Jika Anda memiliki banyak data untuk melatih model Anda, sebagian besar algoritme bawaan dapat dengan mudah disesuaikan untuk memenuhi permintaan. Bahkan jika Anda sudah memiliki model pra-terlatih, mungkin masih lebih mudah untuk menggunakan wajar di SageMaker dan masukan hyper-parameter yang sudah Anda ketahui daripada port itu, menggunakan mode script pada kerangka kerja yang didukung.

Untuk informasi lebih lanjut tentang algoritme bawaan yang disediakan oleh SageMaker, lihatGunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih.

Untuk informasi penting tentang jalur registri docker, tipe instans EC2 yang direkomendasikan, dan CloudWatch log umum untuk semua algoritma built-in yang disediakan oleh SageMaker, lihatInformasi Umum Tentang Algoritma Bawaan.

Gunakan mode skrip dalam kerangka kerja yang didukung

Jika algoritme yang ingin Anda gunakan untuk model Anda tidak didukung oleh pilihan bawaan dan Anda merasa nyaman mengkodekan solusi Anda sendiri, maka Anda harus mempertimbangkan untuk menggunakan Amazon SageMakerkerangka kerja yang didukung. Ini disebut sebagai “mode skrip” karena Anda menulis kode kustom Anda (skrip) dalam file teks dengan `.py`ekstensi. Seperti tabel di atas menunjukkan, SageMaker mendukung sebagian besar kerangka pembelajaran mesin populer. Framework ini dimuat sebelumnya dengan framework yang sesuai dan beberapa paket Python tambahan, seperti Pandas dan NumPy, sehingga Anda bisa tulis kode Anda sendiri untuk melatih algoritme. Kerangka kerja ini juga memungkinkan Anda untuk menginstal paket Python yang di-host di PyPi dengan menyertakan file requirements.txt dengan kode pelatihan Anda atau untuk menyertakan direktori kode Anda sendiri. R juga didukung secara native di SageMaker kernel notebook. Beberapa kerangka kerja, seperti scikit-learn dan Spark MM, memiliki algoritma pra-kode yang dapat Anda gunakan dengan mudah, sementara kerangka kerja lainnya seperti TensorFlow dan PyTorchmungkin mengharuskan Anda untuk menerapkan algoritma sendiri. Satu-satunya batasan saat menggunakan gambar kerangka kerja yang didukung adalah Anda tidak dapat mengimpor paket perangkat lunak apa pun yang tidak di-host PyPi atau yang belum disertakan dengan gambar kerangka kerja.

Untuk informasi lebih lanjut tentang kerangka kerja yang didukung oleh SageMaker, lihatKerangka Kerja dan Bahasa Machine Learning.

Menggunakan image Docker kustom

Amazon SageMakerAlgoritma bawaan dan kerangka kerja yang didukung harus mencakup sebagian besar kasus penggunaan, tetapi ada kalanya Anda mungkin perlu menggunakan algoritme dari paket yang tidak termasuk dalam kerangka kerja yang didukung. Anda mungkin juga memiliki model pra-terlatih memilih atau bertahan di suatu tempat yang perlu Anda gunakan. SageMaker menggunakan gambar Docker untuk menyelenggarakan pelatihan dan penyajian semua model, sehingga Anda dapat menyediakan gambar Docker kustom Anda sendiri jika paket atau perangkat lunak yang Anda butuhkan tidak termasuk dalam kerangka kerja yang didukung. Ini mungkin paket Python Anda sendiri atau algoritma dikodekan dalam bahasa seperti Stan atau Julia. Untuk gambar-gambar ini Anda juga harus mengkonfigurasi pelatihan algoritma dan melayani model dengan benar di Dockerfile Anda. Ini membutuhkan pengetahuan menengah tentang Docker dan tidak disarankan kecuali Anda merasa nyaman menulis algoritma pembelajaran mesin Anda sendiri. Gambar Docker Anda harus diunggah ke repositori online, seperti Amazon Elastic Container Registry (ECR) sebelum Anda dapat melatih dan melayani model Anda dengan benar.

Untuk informasi lebih lanjut tentang image Docker kustom di SageMaker, lihatGunakan kontainer Docker untuk membuat model.

Jenis masalah untuk paradigma pembelajaran mesin dasar

Tiga bagian berikut menjelaskan jenis masalah utama yang ditangani oleh tiga paradigma dasar untuk pembelajaran mesin. Untuk daftar algoritma built-in yang SageMaker menyediakan untuk mengatasi jenis masalah ini, lihatGunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih.

Topik

- Pembelajaran yang diawasi
- Pembelajaran tanpa pengawasan
- Pembelajaran penguatan

Pembelajaran yang diawasi

Jika kumpulan data Anda terdiri dari fitur atau atribut (input) yang berisi nilai target (output), maka Anda memiliki masalah pembelajaran yang diawasi. Jika nilai target Anda kategoris (matematis diskrit), maka Anda memilikiKlasifikasi masalah. Ini adalah praktik standar untuk membedakan biner dari klasifikasi multiclass.

- Klasifikasi bineradalah jenis pembelajaran yang diawasi yang menugaskan individu ke salah satu dari dua kelas yang telah ditentukan dan saling eksklusif berdasarkan atribut individu. Hal ini diawasi karena model dilatih menggunakan contoh di mana atribut disediakan dengan objek berlabel benar. Diagnosis medis untuk apakah seseorang memiliki penyakit atau tidak berdasarkan hasil tes diagnostik adalah contoh klasifikasi biner.
- Klasifikasi multiclassadalah jenis pembelajaran yang diawasi yang menugaskan individu ke salah satu dari beberapa kelas berdasarkan atribut individu. Hal ini diawasi karena model dilatih menggunakan contoh di mana atribut disediakan dengan objek berlabel benar. Contohnya adalah prediksi topik yang paling relevan dengan dokumen teks. Sebuah dokumen dapat diklasifikasikan sebagai tentang agama, politik, atau keuangan, atau sebagai tentang salah satu dari beberapa kelas topik yang telah ditetapkan lainnya.

Jika nilai target yang Anda coba prediksi secara matematis terus menerus, maka Anda memilikiregresimasalah. Regresi memperkirakan nilai variabel target dependen berdasarkan satu atau lebih variabel atau atribut lain yang berkorelasi dengannya. Contohnya adalah prediksi harga rumah menggunakan fitur seperti jumlah kamar mandi dan kamar tidur serta cuplikan persegi rumah dan taman. Analisis regresi dapat membuat model yang mengambil satu atau lebih fitur ini sebagai masukan dan memprediksi harga rumah.

Untuk informasi lebih lanjut tentang algoritma pembelajaran terawasi bawaan yang disediakan oleh SageMaker, lihatPembelajaran yang Diawasi.

Pembelajaran tanpa pengawasan

Jika kumpulan data Anda terdiri dari fitur atau atribut (input) yang tidak mengandung label atau nilai target (output), maka Anda memiliki masalah pembelajaran tanpa pengawasan. Dalam jenis masalah ini, output harus diprediksi berdasarkan pola yang ditemukan dalam data input. Tujuan dalam masalah pembelajaran tanpa pengawasan adalah untuk menemukan pola seperti pengelompokan dalam data. Ada berbagai macam tugas atau jenis masalah yang pembelajaran tanpa pengawasan dapat diterapkan. Komponen utama dan analisis klaster adalah dua metode utama yang biasa digunakan untuk data preprocessing. Berikut adalah daftar singkat jenis masalah yang dapat diatasi dengan pembelajaran tanpa pengawasan:

- Dimensi reduksibiasanya merupakan bagian dari langkah eksplorasi data yang digunakan untuk menentukan fitur yang paling relevan untuk digunakan untuk konstruksi model. Idenya adalah untuk mengubah data dari ruang dimensi tinggi dan jarang penduduknya menjadi ruang dimensi rendah yang mempertahankan sifat paling signifikan dari data asli. Ini memberikan kelegaan bagi kutukan dimensi yang dapat timbul dengan data berdimensi tinggi yang jarang

penduduknya dimana analisis statistik menjadi bermasalah. Ini juga dapat digunakan untuk membantu memahami data, mengurangi data dimensi tinggi ke dimensi yang lebih rendah yang dapat divisualisasikan.

- Analisis klasteradalah kelas teknik yang digunakan untuk mengklasifikasikan objek atau kasus ke dalam kelompok yang disebut cluster. Ini mencoba untuk menemukan pengelompokan diskrit dalam data, di mana anggota kelompok yang sama mungkin satu sama lain dan berbeda mungkin dari anggota kelompok lain. Anda menentukan fitur atau atribut yang Anda inginkan algoritma untuk digunakan untuk menentukan kesamaan, memilih fungsi jarak untuk mengukur kesamaan, dan menentukan jumlah cluster yang akan digunakan dalam analisis.
- Deteksi anomaliadalah identifikasi item langka, peristiwa, atau pengamatan dalam kumpulan data yang menimbulkan kecurigaan karena mereka berbeda secara signifikan dari sisa data. Identifikasi item anomali dapat digunakan, misalnya, untuk mendeteksi penipuan bank atau kesalahan medis. Anomali juga disebut sebagai outliers, hal baru, kebisingan, penyimpangan, dan pengecualian.
- Estimasi kepadatanadalah pembangunan perkiraan fungsi kerapatan probabilitas dasar yang tidak dapat diamati berdasarkan data yang diamati. Penggunaan alami estimasi kepadatan adalah untuk eksplorasi data. Perkiraan kepadatan dapat menemukan fitur seperti kemiringan dan multimodalitas dalam data. Bentuk estimasi kepadatan yang paling dasar adalah histogram rescaled.

SageMaker menyediakan beberapa algoritma pembelajaran mesin bawaan yang dapat Anda gunakan untuk tugas pembelajaran tanpa pengawasan ini. Untuk informasi lebih lanjut tentang algoritma tanpa pengawasan bawaan yang disediakan oleh SageMaker, lihatPembelajaran Tanpa Pengawasan.

Pembelajaran penguatan

Pembelajaran penguatan adalah jenis pembelajaran yang didasarkan pada interaksi dengan lingkungan. Jenis pembelajaran ini digunakan oleh agen yang harus mempelajari perilaku melalui trial-and-error interaksi dengan lingkungan yang dinamis di mana tujuannya adalah untuk memaksimalkan imbalan jangka panjang yang diterima agen sebagai hasil dari tindakannya. Hadiah dimaksimalkan dengan trading off mengeksplorasi tindakan yang memiliki imbalan yang tidak pasti dengan mengeksploitasi tindakan yang telah diketahui imbalan.

Untuk informasi lebih lanjut tentang SageMakerkerangka kerja, toolkit, dan lingkungan untuk pembelajaran penguatan, lihatGunakan Pembelajaran Penguatan dengan Amazon SageMaker.

Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih

Amazon SageMaker menyediakan rangkaian algoritme bawaan, model pra-terlatih, dan templat solusi pra-bangun untuk membantu ilmuwan data dan praktisi pembelajaran mesin memulai pelatihan dan penerapan model pembelajaran mesin dengan cepat. Bagi seseorang yang baru mengenal SageMaker, memilih algoritma yang tepat untuk kasus penggunaan khusus Anda bisa menjadi tugas yang menantang. Tabel berikut menyediakan lembar contekan cepat yang menunjukkan bagaimana Anda dapat memulai dengan contoh masalah atau kasus penggunaan dan menemukan algoritma bawaan yang sesuai yang ditawarkan oleh SageMaker yang valid untuk jenis masalah tersebut. Panduan tambahan yang diselenggarakan oleh paradigma pembelajaran (diawasi dan tidak diawasi) dan domain data penting (teks dan gambar) disediakan di bagian berikut tabel.

Tabel: Memetakan kasus penggunaan ke algoritme bawaan

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Berikut beberapa contoh dari 15 jenis masalah yang dapat diatasi oleh model pra-terlatih dan templat solusi pra-bangun yang disediakan oleh: SageMaker JumpStart Penjawab pertanyaan: chatbot yang menghasilkan jawaban untuk pertanyaan yang diberikan.	Model pra-terlatih dan templat solusi pra-bangun	Klasifikasi Gambar Klasifikasi Tabular Regresi Tabular Klasifikasi Teks Deteksi Objek Penyematan Teks Menjawab Pertanyaan Klasifikasi Pasangan Kalimat	Gambar, Teks, Tabular	Model populer, termasuk Mobilenet, YOLO, Faster R-CNN, BERT, LightGBM, dan CatBoost Untuk daftar model pra-terlatih yang tersedia, lihat JumpStart Model. Untuk daftar templat solusi bawaan yang tersedia, lihat JumpStart Solusi.

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Analisis teks: menganalisis teks dari model khusus untuk domain industri seperti keuangan.		Penyematan Gambar Pengakuan Entitas Bernama Segmentasi Instance Generasi Teks Ringkasan Teks Segmentasi Semantik Terjemahan Mesin		
Memprediksi jika item termasuk dalam kategori: filter spam email	Pembelajaran yang Diawasi	Klasifikasi biner/ multi-kelas	Tabular	AutoGluon-Tabular, CatBoost, Algoritme faktorisasi, Algoritme k-Nearest Neighbor (k-NN), LightGBM, Algoritma Linear, TabTransformer, Algoritma XGBoost

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Memprediksi nilai numerik/kontinu: perkiraan nilai rumah		Regresi	Tabular	AutoGluon-Tabular, CatBoost, Algoritme faktorisasi, Algoritme k-Nearest Neighbor (k-NN), LightGBM, Algoritma Linear, TabTransformer, Algoritma XGBoost
Berdasarkan data historis untuk suatu perilaku, prediksi perilaku masa depan: memprediksi penjualan pada produk baru berdasarkan data penjualan sebelumnya.		Peramalan deret waktu	Tabular	Algoritma Peramalan Deepar

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Tingkatkan penyematan data objek dimensi tinggi: identifikasi tiket dukungan duplikat atau temukan perutean yang benar berdasarkan kesamaan teks dalam tiket		Embeddings: mengubah objek berdimensi tinggi menjadi ruang dimensi rendah.	Tabular	Algoritma Object2Vec
Jatuhkan kolom tersebut dari kumpulan data yang memiliki hubungan lemah dengan variabel label/target: warna mobil saat memprediksi jarak tempuhnya.	Pembelajaran Tanpa Pengawasan	Rekayasa fitur: pengurangan dimensi	Tabular	Algoritma Analisis Komponen Utama (PCA)
Mendeteksi perilaku abnormal dalam aplikasi: temukan saat sensor IoT mengirimkan pembacaan abnormal		Deteksi anomali	Tabular	Algoritma Acak Cut Forest (RCF)

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Lindungi aplikasi Anda dari pengguna yang mencurigakan: deteksi jika alamat IP yang mengakses layanan mungkin berasal dari aktor yang buruk		Deteksi anomali IP	Tabular	Wawasan IP
Kelompokkan objek/data serupa bersama-sama: temukan pelanggan dengan pengeluaran tinggi, menengah, dan rendah dari riwayat transaksi mereka		Pengelompokan atau pengelompokan	Tabular	Algoritma K-Means

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Atur satu set dokumen ke dalam topik (tidak diketahui sebelumnya): tandai dokumen sebagai milik kategori medis berdasarkan istilah yang digunakan dalam dokumen.		Pemodelan topik	Teks	Laten Dirichlet Alokasi (LDA) Algoritma, Algoritma Model Topik Saraf (NTM)
Tetapkan kategori yang telah ditentuka n sebelumny a ke dokumen dalam korpus: kategorikan buku di perpustakaan ke dalam disiplin akademis	Analisis Tekstual	Klasifikasi teks	Teks	BlazingTe xtalgoritma, Klasifikasi Teks - TensorFlow
Konversi teks dari satu bahasa ke bahasa lain: Spanyol ke Inggris		Terjemahan mesin algoritma	Teks	Algoritma Urutan ke Urutan

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Meringkas korpus teks panjang: abstrak untuk paper penelitian		Ringkasan teks	Teks	Algoritma Urutan ke Urutan
Konversi file audio menjadi teks: transkripsikan percakapan pusat panggilan untuk analisis lebih lanjut		S peech-to-text	Teks	Algoritma Urutan ke Urutan
Label/tag gambar berdasarkan konten gambar: peringatan tentang konten dewasa dalam gambar	Pemrosesan Gambar	Klasifikasi gambar dan multi-label	Citra	Klasifikasi Gambar - MXNet
Klasifikasi sesuatu dalam gambar menggunakan pembelajaran transfer.		Klasifikasi gambar	Citra	Klasifikasi Gambar - TensorFlow

Contoh masalah dan kasus penggunaan	Paradigma atau domain pembelajaran	Jenis masalah	Format masukan data	Algoritma bawaan
Mendeteksi orang dan objek dalam gambar: polisi meninjau galeri foto besar untuk orang hilang		Deteksi dan klasifikasi objek	Citra	Deteksi, Deteksi Objek - TensorFlow
Tandai setiap piksel gambar satu per satu dengan kategori: mobil self-driv ing bersiap untuk mengidentifikasi objek dengan cara mereka		Visi komputer	Citra	Algoritme segmentasi semantik

Untuk informasi penting tentang jalur registri Docker, format data, jenis instans Amazon EC2 yang dimulai ulang, dan CloudWatch log yang umum untuk semua algoritme bawaan yang disediakan oleh, lihat. SageMaker Informasi Umum Tentang Algoritma Bawaan

Bagian berikut memberikan panduan tambahan untuk algoritme SageMaker bawaan Amazon yang dikelompokkan berdasarkan paradigma pembelajaran yang diawasi dan tidak diawasi. Untuk deskripsi paradigma pembelajaran ini dan jenis masalah terkaitnya, lihat. Pilih algoritme Bagian juga disediakan untuk algoritme SageMaker bawaan yang tersedia untuk menangani dua domain pembelajaran mesin penting: analisis tekstual dan pemrosesan gambar.

- Model dan Templat Solusi Pra-terlatih
- Pembelajaran yang Diawasi
- Pembelajaran Tanpa Pengawasan
- Analisis Tekstual

- Pemrosesan Gambar

Model dan Templat Solusi Pra-terlatih

SageMaker JumpStart menyediakan berbagai model pra-terlatih, templat solusi pra-bangun, dan contoh untuk jenis masalah populer yang menggunakan SageMaker SDK serta Studio Classic. Untuk informasi lebih lanjut tentang model ini, solusi, dan contoh notebook yang disediakan oleh SageMaker JumpStart, lihatSageMaker JumpStart.

Pembelajaran yang Diawasi

Amazon SageMaker menyediakan beberapa algoritma tujuan umum bawaan yang dapat digunakan untuk masalah klasifikasi atau regresi.

- AutoGluon-Tabular—kerangka AutoML open-source yang berhasil dengan menyamai model dan menumpuknya dalam beberapa lapisan.
- CatBoost—implementasi algoritme pohon yang ditingkatkan gradien yang memperkenalkan peningkatan berurutan dan algoritme inovatif untuk memproses fitur kategoris.
- Algoritme faktorisasi—perpanjangan dari model linier yang dirancang untuk menangkap interaksi secara ekonomis antara fitur dalam kumpulan data jarang berdimensi tinggi.
- Algoritme k-Nearest Neighbor (k-NN)—metode non-parametrik yang menggunakan k titik berlabel terdekat untuk menetapkan label ke titik data baru untuk klasifikasi atau nilai target yang diprediksi dari rata-rata k titik terdekat untuk regresi.
- LightGBM—implementasi algoritma pohon yang ditingkatkan gradien yang menambahkan dua teknik baru untuk meningkatkan efisiensi dan skalabilitas: Pengambilan Sampel Satu Sisi Berbasis Gradien (GOSS) dan Bundling Fitur Eksklusif (EFB).
- Algoritma Linear—mempelajari fungsi linier untuk regresi atau fungsi ambang linier untuk klasifikasi.
- TabTransformer—arsitektur pemodelan data tabular mendalam baru yang dibangun di atas self-attention-based Transformers.
- Algoritma XGBoost—implementasi algoritma pohon yang ditingkatkan gradien yang menggabungkan ansambel perkiraan dari serangkaian model yang lebih sederhana dan lebih lemah.

Amazon SageMaker juga menyediakan beberapa algoritma pembelajaran terawasi bawaan yang digunakan untuk tugas yang lebih khusus selama rekayasa fitur dan peramalan dari data deret waktu.

- Algoritma Object2Vec—algoritma multi-tujuan baru yang sangat dapat disesuaikan yang digunakan untuk rekayasa fitur. Ini dapat mempelajari penyematan padat dimensi rendah dari objek dimensi tinggi untuk menghasilkan fitur yang meningkatkan efisiensi pelatihan untuk model hilir. Meskipun ini adalah algoritma yang diawasi, karena memerlukan data berlabel untuk pelatihan, ada banyak skenario di mana label hubungan dapat diperoleh murni dari pengelompokan alami dalam data, tanpa anotasi manusia yang eksplisit.
- Algoritma Peramalan Deepar—algoritma pembelajaran yang diawasi untuk meramalkan deret waktu skalar (satu dimensi) menggunakan jaringan saraf berulang (RNN).

Pembelajaran Tanpa Pengawasan

Amazon SageMaker menyediakan beberapa algoritma bawaan yang dapat digunakan untuk berbagai tugas pembelajaran tanpa pengawasan seperti pengelompokan, pengurangan dimensi, pengenalan pola, dan deteksi anomali.

- Algoritma Analisis Komponen Utama (PCA)—mengurangi dimensi (jumlah fitur) dalam kumpulan data dengan memproyeksikan titik data ke beberapa komponen utama pertama. Tujuannya adalah untuk menyimpan informasi atau variasi sebanyak mungkin. Untuk matematikawan, komponen utama adalah vektor eigen dari matriks kovarians data.
- Algoritma K-Means—menemukan pengelompokan diskrit dalam data, di mana anggota kelompok semirip mungkin satu sama lain dan berbeda mungkin dari anggota kelompok lain.
- Wawasan IP—mempelajari pola penggunaan untuk alamat IPv4. Ini dirancang untuk menangkap asosiasi antara alamat IPv4 dan berbagai entitas, seperti ID pengguna atau nomor akun.
- Algoritma Acak Cut Forest (RCF)—mendeteksi titik data anomali dalam kumpulan data yang menyimpang dari data yang terstruktur atau berpola dengan baik.

Analisis Tekstual

SageMaker menyediakan algoritma yang disesuaikan dengan analisis dokumen tekstual yang digunakan dalam pemrosesan bahasa alami, klasifikasi atau ringkasan dokumen, pemodelan atau klasifikasi topik, dan transkripsi atau terjemahan bahasa.

- BlazingTextalgoritma—implementasi yang sangat dioptimalkan dari Word2vec dan algoritma klasifikasi teks yang menskalakan ke kumpulan data besar dengan mudah. Ini berguna untuk banyak tugas pemrosesan bahasa alami hilir (NLP).

- Algoritma Urutan ke Urutan—algoritma yang diawasi yang biasa digunakan untuk terjemahan mesin saraf.
- Laten Dirichlet Alokasi (LDA) AlgoritmaSebuah algoritma yang cocok untuk menentukan topik dalam satu set dokumen. Ini adalah algoritma tanpa pengawasan, yang berarti tidak menggunakan data contoh dengan jawaban selama pelatihan.
- Algoritma Model Topik Saraf (NTM)—teknik lain yang tidak diawasi untuk menentukan topik dalam satu set dokumen, menggunakan pendekatan jaringan saraf.
- Klasifikasi Teks - TensorFlow—algoritma yang diawasi yang mendukung pembelajaran transfer dengan model terlatih yang tersedia untuk klasifikasi teks.

Pemrosesan Gambar

SageMaker juga menyediakan algoritma pemrosesan gambar yang digunakan untuk klasifikasi gambar, deteksi objek, dan visi komputer.

- Klasifikasi Gambar - MXNet—menggunakan contoh data dengan jawaban (disebut sebagai algoritma yang diawasi). Gunakan algoritma ini untuk mengklasifikasikan gambar.
- Klasifikasi Gambar - TensorFlow—menggunakan model TensorFlow Hub terlatih untuk menyempurnakan tugas-tugas tertentu (disebut sebagai algoritma yang diawasi). Gunakan algoritma ini untuk mengklasifikasikan gambar.
- Algoritme segmentasi semantik—menyediakan pendekatan tingkat piksel berbutir halus untuk mengembangkan aplikasi visi komputer.
- Deteksi—mendeteksi dan mengklasifikasikan objek dalam gambar menggunakan satu jaringan saraf dalam. Ini adalah algoritma pembelajaran yang diawasi yang mengambil gambar sebagai input dan mengidentifikasi semua contoh objek dalam adegan gambar.
- Deteksi Objek - TensorFlow—mendeteksi kotak pembatas dan label objek dalam gambar. Ini adalah algoritma pembelajaran yang diawasi yang mendukung pembelajaran transfer dengan model terlatih TensorFlow yang tersedia.

Topik

- Informasi Umum Tentang Algoritma Bawaan
- Built-inSageMakerAlgoritma untuk Data Tabular
- SageMaker Algoritma Bawaan untuk Data Teks
- Built-inSageMakerAlgoritma untuk Data Time-Series

- Built-in tanpa pengawasanSageMakerAlgoritma
- SageMaker Algoritma Built-in untuk Visi Komputer

Informasi Umum Tentang Algoritma Bawaan

Tabel berikut mencantumkan parameter untuk masing-masing algoritma yang disediakan oleh Amazon SageMaker.

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas instans	Dapat diparalelkan	
AutoGluon-Tabular	pelatihan dan validasi (opsional)	File	CSV	CPU atau GPU (hanya satu contoh)	Tidak	
BlazingText	melatih	File atau Pipa	File teks (satu kalimat per baris dengan token yang dipisahkan spasi)	CPU atau GPU (hanya satu contoh)	Tidak	
CatBoost	pelatihan dan validasi (opsional)	File	CSV	CPU (hanya satu contoh)	Tidak	
Peramalan DeepAR	melatih dan (opsional) tes	File	Garis JSON atau Parket	CPU atau GPU	Ya	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas instans	Dapat diparalelkan	
Mesin Faktorisasi	melatih dan (opsional) tes	File atau Pipa	Protobuf Recordio	CPU (GPU untuk data padat)	Ya	
Klasifikasi Gambar - MXNet	melatih dan validasi, (opsional) train_lst, validation_lst, dan model	File atau Pipa	RecorDo atau file gambar (.jpg atau.png)	GPU	Ya	
Klasifikasi Gambar - TensorFlow	pelatihan dan validasi	File	file gambar (.jpg, .jpeg, atau .png)	CPU atau GPU	Ya (hanya di beberapa GPU dalam satu instance)	
Wawasan IP	melatih dan (opsional) validasi	File	CSV	CPU atau GPU	Ya	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas instans	Dapat diparalelkan	
K-Berarti	melatih dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPUCommon (perangkat GPU tunggal pada satu atau beberapa instance)	Tidak	
K-Tetangga Terdekat (K-nn)	melatih dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPU (perangkat GPU tunggal pada satu atau beberapa instance)	Ya	
LDA	melatih dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU (hanya satu contoh)	Tidak	
LightGBM	pelatihan/pelatihan dan validasi (opsional)	File	CSV	CPU	Ya	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas instans	Dapat diparalelkan	
Linear Learner	melatih dan (opsional) validasi, tes, atau keduanya	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPU	Ya	
Model Topik Saraf	melatih dan (opsional) validasi, tes, atau keduanya	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPU	Ya	
Object2Vec	melatih dan (opsional) validasi, tes, atau keduanya	File	Garis JSON	CPU atau GPU (hanya satu contoh)	Tidak	
Deteksi Objek - MxNet	melatih dan validasi, (opsional) train_annotation, validation_annotation, dan model	File atau Pipa	RecorDo atau file gambar (.jpg atau.png)	GPU	Ya	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas instans	Dapat diparalelkan	
Deteksi Objek - TensorFlow	pelatihan dan validasi	File	file gambar (.jpg, .jpeg, atau .png)	GPU	Ya (hanya di beberapa GPU dalam satu instance)	
PCA	melatih dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPU	Ya	
Random Cut Forest	melatih dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU	Ya	
Segmentasi Semantik	melatih dan validasi, train_annotation, validation_annotation, dan (opsional) label_map dan model	File atau Pipa	File gambar	GPU (hanya satu contoh)	Tidak	
Pemodelan Seq2Seq	melatih, validasi, dan vocab	File	Protobuf Recordio	GPU (hanya satu contoh)	Tidak	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas instans	Dapat diparalelkan	
TabTransformer	pelatihan dan validasi (opsional)	File	CSV	CPU atau GPU (hanya satu contoh)	Tidak	
Klasifikasi Teks - TensorFlow	pelatihan dan validasi	File	CSV	CPU atau GPU	Ya (hanya di beberapa GPU dalam satu instance)	
XGBoost (0,90-1, 0,90-2, 1,0-1, 1,2-1, 1,2-21)	melatih dan (opsional) validasi	File atau Pipa	CSV, LibSVM, atau Parket	CPU (atau GPU untuk 1.2-1)	Ya	

Algoritma yang dapat diparalelkan dapat digunakan pada beberapa instance komputasi untuk pelatihan terdistribusi.

Topik berikut memberikan informasi tentang format data, jenis instans Amazon EC2 yang direkomendasikan, dan CloudWatch log yang umum untuk semua algoritme bawaan yang disediakan oleh Amazon. SageMaker

> **Note**
>
> Untuk mencari URI image Docker dari algoritma bawaan yang dikelola oleh SageMaker, lihat Docker Registry Paths and Example Code.

Topik

- Format Data Umum untuk Algoritma Bawaan
- Jenis Instance untuk Algoritma Bawaan
- Log untuk Algoritma Bawaan

Format Data Umum untuk Algoritma Bawaan

Topik berikut menjelaskan format data untuk algoritme yang disediakan oleh Amazon SageMaker.

Topik

- Format Data Umum untuk Pelatihan
- Format Data Umum untuk Inferensi

Format Data Umum untuk Pelatihan

Untuk mempersiapkan pelatihan, Anda dapat memproses data Anda menggunakan berbagai AWS layanan, termasuk Amazon EMR, Amazon RedshiftAWS Glue, Amazon Relational Database Service, dan Amazon Athena. Setelah preprocessing, publikasikan data ke bucket Amazon S3. Untuk pelatihan, data harus melalui serangkaian konversi dan transformasi, termasuk:

- Serialisasi data pelatihan (ditangani oleh Anda)
- Deserialisasi data pelatihan (ditangani oleh algoritma)
- Serialisasi model pelatihan (ditangani oleh algoritma)
- Deserialisasi model terlatih (opsional, ditangani oleh Anda)

Saat menggunakan Amazon SageMaker di bagian pelatihan algoritme, pastikan untuk mengunggah semua data sekaligus. Jika lebih banyak data ditambahkan ke lokasi itu, panggilan pelatihan baru perlu dibuat untuk membangun model baru.

Topik

- Jenis Konten yang Didukung oleh Algoritma Bawaan
- Menggunakan Mode Pipa
- Menggunakan Format CSV
- Menggunakan Format RecorDio

- Deserialisasi Model Terlatih

Jenis Konten yang Didukung oleh Algoritma Bawaan

Tabel berikut mencantumkan beberapa `ContentType`nilai yang didukung umum dan algoritma yang menggunakannya:

ContentTypes untuk Algoritma Bawaan

ContentType	Algoritme
aplikasi/x-image	Algoritma Deteksi Objek, Segmentasi Semantik
aplikasi/x-recordio	Algoritma Deteksi Objek
aplikasi/ x-recordio-protobuf	Mesin Faktorisasi, K-Means, K-NN, Alokasi Dirichlet Laten, Linear Learner, NTM, PCA, RCF, Urutan-ke-Urutan
aplikasi/jsonlines	BlazingText, DeepAR
gambar/jpeg	Algoritma Deteksi Objek, Segmentasi Semantik
gambar/png	Algoritma Deteksi Objek, Segmentasi Semantik
teks/csv	Wawasan IP, K-Means, K-NN, Alokasi Dirichlet Laten, Pelajar Linear, NTM, PCA, RCF, XGBoost
teks/libsvm	XGBoost

Untuk ringkasan parameter yang digunakan oleh setiap algoritma, lihat dokumentasi untuk algoritma individu atau tabel ini.

Menggunakan Mode Pipa

Dalam mode Pipe, pekerjaan latihan Anda mengalirkan data langsung dari Amazon Simple Storage Service (Amazon S3). Streaming dapat memberikan waktu mulai yang lebih cepat untuk pekerjaan pelatihan dan throughput yang lebih baik. Ini berbeda dengan mode File, di mana data Anda dari Amazon S3 disimpan pada volume instans pelatihan. Mode file menggunakan ruang disk untuk menyimpan artefak model akhir dan kumpulan data pelatihan lengkap Anda. Dengan streaming data Anda langsung dari Amazon S3 dalam mode Pipa, Anda mengurangi ukuran volume Amazon

Elastic Block Store dari instans pelatihan Anda. Mode pipa hanya membutuhkan ruang disk yang cukup untuk menyimpan artefak model akhir Anda. Lihat `AlgorithmSpecification`untuk detail tambahan tentang mode input pelatihan.

Menggunakan Format CSV

Banyak SageMaker algoritma Amazon mendukung pelatihan dengan data dalam format CSV. Untuk menggunakan data dalam format CSV untuk pelatihan, dalam spesifikasi saluran data input, tentukan **`text/csv`** sebagai. `ContentType` Amazon SageMaker mengharuskan file CSV tidak memiliki catatan header dan variabel target ada di kolom pertama. Untuk menjalankan algoritme pembelajaran tanpa pengawasan yang tidak memiliki target, tentukan jumlah kolom label dalam jenis konten. Misalnya, dalam hal ini**`'content_type=text/csv;label_size=0'`**. Untuk contoh buku catatan yang menggunakan format CSV, lihat Prediksi Kanker Payudara. Untuk informasi selengkapnya, lihat Sekarang menggunakan mode Pipa dengan kumpulan data CSV untuk pelatihan lebih cepat tentang algoritme bawaan Amazon SageMaker.

Menggunakan Format RecorDio

Dalam format protobuf Recordio, SageMaker mengubah setiap pengamatan dalam dataset menjadi representasi biner sebagai satu set float 4-byte, lalu memuatnya di bidang nilai protobuf. Jika Anda menggunakan Python untuk persiapan data Anda, kami sangat menyarankan Anda menggunakan transformasi yang ada ini. Namun, jika Anda menggunakan bahasa lain, file definisi protobuf di bawah ini menyediakan skema yang Anda gunakan untuk mengonversi data Anda ke format protobuf. SageMaker

> **Note**
>
> Untuk contoh yang menunjukkan cara mengonversi array NumPy yang umum digunakan ke dalam format protobuf RecorDio, lihat Pengantar Mesin Faktorisasi dengan MNIST.

```
syntax = "proto2";

 package aialgs.data;

 option java_package = "com.amazonaws.aialgorithms.proto";
 option java_outer_classname = "RecordProtos";

 // A sparse or dense rank-R tensor that stores data as doubles (float64).
 message Float32Tensor   {
```

```
    // Each value in the vector. If keys is empty, this is treated as a
    // dense vector.
    repeated float values = 1 [packed = true];

    // If key is not empty, the vector is treated as sparse, with
    // each key specifying the location of the value in the sparse vector.
    repeated uint64 keys = 2 [packed = true];

    // An optional shape that allows the vector to represent a matrix.
    // For example, if shape = [ 10, 20 ], floor(keys[i] / 20) gives the row,
    // and keys[i] % 20 gives the column.
    // This also supports n-dimensonal tensors.
    // Note: If the tensor is sparse, you must specify this value.
    repeated uint64 shape = 3 [packed = true];
}

// A sparse or dense rank-R tensor that stores data as doubles (float64).
message Float64Tensor {
    // Each value in the vector. If keys is empty, this is treated as a
    // dense vector.
    repeated double values = 1 [packed = true];

    // If this is not empty, the vector is treated as sparse, with
    // each key specifying the location of the value in the sparse vector.
    repeated uint64 keys = 2 [packed = true];

    // An optional shape that allows the vector to represent a matrix.
    // For example, if shape = [ 10, 20 ], floor(keys[i] / 10) gives the row,
    // and keys[i] % 20 gives the column.
    // This also supports n-dimensonal tensors.
    // Note: If the tensor is sparse, you must specify this value.
    repeated uint64 shape = 3 [packed = true];
}

// A sparse or dense rank-R tensor that stores data as 32-bit ints (int32).
message Int32Tensor {
    // Each value in the vector. If keys is empty, this is treated as a
    // dense vector.
    repeated int32 values = 1 [packed = true];

    // If this is not empty, the vector is treated as sparse with
    // each key specifying the location of the value in the sparse vector.
    repeated uint64 keys = 2 [packed = true];
```

```
    // An optional shape that allows the vector to represent a matrix.
    // For Exmple, if shape = [ 10, 20 ], floor(keys[i] / 10) gives the row,
    // and keys[i] % 20 gives the column.
    // This also supports n-dimensonal tensors.
    // Note: If the tensor is sparse, you must specify this value.
    repeated uint64 shape = 3 [packed = true];
}

// Support for storing binary data for parsing in other ways (such as JPEG/etc).
// This is an example of another type of value and may not immediately be supported.
message Bytes {
    repeated bytes value = 1;

    // If the content type of the data is known, stores it.
    // This allows for the possibility of using decoders for common formats
    // in the future.
    optional string content_type = 2;
}

message Value {
    oneof value {
        // The numbering assumes the possible use of:
        // - float16, float128
        // - int8, int16, int32
        Float32Tensor float32_tensor = 2;
        Float64Tensor float64_tensor = 3;
        Int32Tensor int32_tensor = 7;
        Bytes bytes = 9;
    }
}

message Record {
    // Map from the name of the feature to the value.
    //
    // For vectors and libsvm-like datasets,
    // a single feature with the name `values`
    // should be specified.
    map<string, Value> features = 1;

    // An optional set of labels for this record.
    // Similar to the features field above, the key used for
    // generic scalar / vector labels should be 'values'.
    map<string, Value> label = 2;
```

```
    // A unique identifier for this record in the dataset.
    //
    // Whilst not necessary, this allows better
    // debugging where there are data issues.
    //
    // This is not used by the algorithm directly.
    optional string uid = 3;

    // Textual metadata describing the record.
    //
    // This may include JSON-serialized information
    // about the source of the record.
    //
    // This is not used by the algorithm directly.
    optional string metadata = 4;

    // An optional serialized JSON object that allows per-record
    // hyper-parameters/configuration/other information to be set.
    //
    // The meaning/interpretation of this field is defined by
    // the algorithm author and may not be supported.
    //
    // This is used to pass additional inference configuration
    // when batch inference is used (e.g. types of scores to return).
    optional string configuration = 5;
}
```

Setelah membuat buffer protokol, simpan di lokasi Amazon S3 yang dapat diakses SageMaker Amazon dan yang dapat diteruskan sebagai bagian dari `InputDataConfig` in. `create_training_job`

> **Note**
>
> Untuk semua SageMaker algoritma Amazon, `ChannelName` in `InputDataConfig` harus diatur `ketrain`. Beberapa algoritma juga mendukung validasi atau pengujian. `input channels` Ini biasanya digunakan untuk mengevaluasi kinerja model dengan menggunakan dataset penahanan. Kumpulan data penahanan tidak digunakan dalam pelatihan awal tetapi dapat digunakan untuk menyetel model lebih lanjut.

Deserialisasi Model Terlatih

SageMaker Model Amazon disimpan sebagai model.tar.gz di bucket S3 yang ditentukan dalam `OutputDataConfig` `S3OutputPath` parameter `create_training_job` panggilan. Bucket S3 harus berada di AWS Region yang sama dengan instance notebook. Anda dapat menentukan sebagian besar artefak model ini saat membuat model hosting. Anda juga dapat membuka dan meninjaunya di instance notebook Anda. Ketika `model.tar.gz` untarred, itu berisi`model_algo-1`, yang merupakan objek Apache MXNet serial. Misalnya, Anda menggunakan yang berikut ini untuk memuat model k-means ke dalam memori dan melihatnya:

```
import mxnet as mx
print(mx.ndarray.load('model_algo-1'))
```

Format Data Umum untuk Inferensi

SageMaker Algoritma Amazon menerima dan menghasilkan beberapa jenis MIME yang berbeda untuk muatan HTTP yang digunakan dalam mengambil prediksi online dan mini-batch. Anda dapat menggunakan berbagai AWS layanan untuk mengubah atau memproses catatan sebelum menjalankan inferensi. Minimal, Anda perlu mengonversi data sebagai berikut:

- Serialisasi permintaan inferensi (ditangani oleh Anda)
- Deserialisasi permintaan inferensi (ditangani oleh algoritma)
- Serialisasi respons inferensi (ditangani oleh algoritma)
- Deserialisasi respons inferensi (ditangani oleh Anda)

Topik

- Konversi Data untuk Serialisasi Permintaan Inferensi
- Konversi Data untuk Deserialisasi Respon Inferensi
- Format Permintaan Umum untuk Semua Algoritma
- Gunakan Transformasi Batch dengan Algoritma Bawaan

Konversi Data untuk Serialisasi Permintaan Inferensi

Opsi jenis konten untuk permintaan inferensi SageMaker algoritme Amazon meliputi:`text/csv`,`application/json`, dan`application/x-recordio-protobuf`. Algoritma yang tidak mendukung semua jenis ini dapat mendukung jenis lain. XGBoost, misalnya, hanya mendukung `text/csv` dari daftar ini, tetapi juga mendukung. `text/libsvm`

Untuk`text/csv`, nilai argumen Body `invoke_endpoint` harus berupa string dengan koma yang memisahkan nilai untuk setiap fitur. Misalnya, rekaman untuk model dengan empat fitur mungkin terlihat seperti`1.5,16.0,14,23.0`. Setiap transformasi yang dilakukan pada data pelatihan juga harus dilakukan pada data sebelum mendapatkan inferensi. Urutan fitur penting dan harus tetap tidak berubah.

`application/json`secara signifikan lebih fleksibel dan menyediakan beberapa format yang mungkin bagi pengembang untuk digunakan dalam aplikasi mereka. Pada tingkat tinggi, di JavaScript, payload mungkin terlihat seperti berikut:

```
let request = {
  // Instances might contain multiple rows that predictions are sought for.
  "instances": [
    {
      // Request and algorithm specific inference parameters.
      "configuration": {},
      // Data in the specific format required by the algorithm.
      "data": {
         "<field name>": dataElement
       }
    }
  ]
}
```

Anda memiliki opsi berikut untuk menentukan: `dataElement`

Protokol buffer setara

```
// Has the same format as the protocol buffers implementation described for training.
let dataElement = {
  "keys": [],
  "values": [],
  "shape": []
}
```

vektor numerik sederhana

```
// An array containing numeric values is treated as an instance containing a
// single dense vector.
let dataElement = [1.5, 16.0, 14.0, 23.0]
```

```
// It will be converted to the following representation by the SDK.
let converted = {
  "features": {
    "values": dataElement
  }
}
```

Untuk beberapa catatan

```
let request = {
  "instances": [
    // First instance.
    {
      "features": [ 1.5, 16.0, 14.0, 23.0 ]
    },
    // Second instance.
    {
      "features": [ -2.0, 100.2, 15.2, 9.2 ]
    }
  ]
}
```

Konversi Data untuk Deserialisasi Respon Inferensi

SageMaker Algoritma Amazon mengembalikan JSON dalam beberapa tata letak. Pada tingkat tinggi, strukturnya adalah:

```
let response = {
  "predictions": [{
    // Fields in the response object are defined on a per algorithm-basis.
  }]
}
```

Bidang yang disertakan dalam prediksi berbeda di seluruh algoritme. Berikut ini adalah contoh output untuk algoritma k-means.

Inferensi rekor tunggal

```
let response = {
  "predictions": [{
    "closest_cluster": 5,
    "distance_to_cluster": 36.5
```

```
  }]
}
```

Inferensi multi-rekam

```
let response = {
  "predictions": [
    // First instance prediction.
    {
      "closest_cluster": 5,
      "distance_to_cluster": 36.5
    },
    // Second instance prediction.
    {
      "closest_cluster": 2,
      "distance_to_cluster": 90.3
    }
  ]
}
```

Inferensi multi-rekam dengan input protobuf

```
{
  "features": [],
  "label": {
    "closest_cluster": {
      "values": [ 5.0 ] // e.g. the closest centroid/cluster was 1.0
    },
    "distance_to_cluster": {
      "values": [ 36.5 ]
    }
  },
  "uid": "abc123",
  "metadata": "{ "created_at": '2017-06-03' }"
}
```

SageMaker algoritma juga mendukung format JSONLINES, di mana konten respons per rekaman sama dengan yang ada dalam format JSON. Struktur multi-record adalah rangkaian objek respons per rekaman yang dipisahkan oleh karakter baris baru. Konten respons untuk algoritma KMeans bawaan untuk 2 titik data input adalah:

```
{"distance_to_cluster": 23.40593910217285, "closest_cluster": 0.0}
```

```
{"distance_to_cluster": 27.250282287597656, "closest_cluster": 0.0}
```

Saat menjalankan transformasi batch, kami merekomendasikan menggunakan tipe `jsonlines` respons dengan menyetel `Accept` bidang di `CreateTransformJobRequest` to`application/jsonlines`.

Format Permintaan Umum untuk Semua Algoritma

Sebagian besar algoritma menggunakan beberapa format permintaan inferensi berikut.

Format Permintaan JSON

Jenis konten: Aplikasi/JSON

Format padat

```
let request =   {
    "instances":    [
        {
            "features": [1.5, 16.0, 14.0, 23.0]
        }
    ]
}


let request =   {
    "instances":    [
        {
            "data": {
                "features": {
                    "values": [ 1.5, 16.0, 14.0, 23.0]
                }
            }
        }
    ]
}
```

Format jarang

```
{
 "instances": [
  {"data": {"features": {
     "keys": [26, 182, 232, 243, 431],
```

```
     "shape": [2000],
     "values": [1, 1, 1, 4, 1]
    }
   }
  },
  {"data": {"features": {
     "keys": [0, 182, 232, 243, 431],
     "shape": [2000],
     "values": [13, 1, 1, 4, 1]
    }
   }
  },
 ]
}
```

Format Permintaan JSONLINES

Jenis konten: Aplikasi/JSONlines

Format padat

Sebuah catatan tunggal dalam format padat dapat direpresentasikan sebagai:

```
{ "features": [1.5, 16.0, 14.0, 23.0] }
```

atau:

```
{ "data": { "features": { "values": [ 1.5, 16.0, 14.0, 23.0] } }
```

Format Jarang

Sebuah catatan tunggal dalam format jarang direpresentasikan sebagai:

```
{"data": {"features": { "keys": [26, 182, 232, 243, 431], "shape": [2000], "values":
 [1, 1, 1, 4, 1] } } }
```

Beberapa catatan direpresentasikan sebagai rangkaian representasi rekaman tunggal di atas, dipisahkan oleh karakter baris baru:

```
{"data": {"features": { "keys": [0, 1, 3], "shape": [4], "values": [1, 4, 1] } } }
{ "data": { "features": { "values": [ 1.5, 16.0, 14.0, 23.0] } }
{ "features": [1.5, 16.0, 14.0, 23.0] }
```

Format Permintaan CSV

Jenis konten: teks/CSV; label_size = 0

> **Note**
>
> Dukungan CSV tidak tersedia untuk mesin faktorisasi.

Format Permintaan RECORDIO

Jenis konten: aplikasi/ x-recordio-protobuf

Gunakan Transformasi Batch dengan Algoritma Bawaan

Saat menjalankan transformasi batch, sebaiknya gunakan tipe respons JSONLINES alih-alih JSON, jika didukung oleh algoritme. Ini dicapai dengan mengatur `Accept` bidang di `CreateTransformJobRequest` to`application/jsonlines`.

Ketika Anda membuat pekerjaan transformasi, `SplitType` harus diatur sesuai dengan `ContentType` data input. Demikian pula, tergantung pada `Accept` bidang di`CreateTransformJobRequest`, `AssembleWith` harus diatur sesuai. Silakan gunakan tabel berikut untuk membantu mengatur bidang ini dengan tepat:

ContentType	Direkomendasikan SplitType
`application/x-recordio-protobuf`	`RecordIO`
`text/csv`	`Line`
`application/jsonlines`	`Line`
`application/json`	`None`
`application/x-image`	`None`
`image/*`	`None`
Menerima	Direkomendasikan AssembleWith
`application/x-recordio-protobuf`	`None`

Menerima	Direkomendasikan AssembleWith
`application/json`	`None`
`application/jsonlines`	`Line`

Untuk informasi selengkapnya tentang format respons untuk algoritme tertentu, lihat berikut ini:

- Format Inferensi Deepar
- Faktorisasi Mesin Respon Format
- Format Data Inferensi Wawasan IP
- Format Respons K-Means
- Format Permintaan dan Respons K-nN
- Format respons pelajar linier
- Format Respons NTM
- Format Data untuk Inferensi Object2Vec
- Embeddings Encoder untuk Object2Vec
- Format Respons
- Format Respons RCF

Jenis Instance untuk Algoritma Bawaan

Untuk melatih dan menghosting SageMaker algoritme Amazon, sebaiknya gunakan jenis instans Amazon EC2 berikut:

- ml.m5.xlarge, ml.m5.4xlarge, dan ml.m5.12xlarge
- ml.c5.xlarge, ml.c5.2xlarge, dan ml.c5.8xlarge
- ml.p3.xlarge, ml.p3.8xlarge, dan ml.p3.16xlarge

Sebagian besar SageMaker algoritma Amazon telah direkayasa untuk memanfaatkan komputasi GPU untuk pelatihan. Untuk sebagian besar pelatihan algoritme, kami mendukung instans GPU P2, P3, G4dn, dan G5. Meskipun biaya per instans lebih tinggi, GPU berlatih lebih cepat, membuatnya lebih hemat biaya. Pengecualian dicatat dalam panduan ini.

Ukuran dan jenis data dapat memiliki efek besar pada konfigurasi perangkat keras mana yang paling efektif. Ketika model yang sama dilatih secara berulang, pengujian awal di seluruh spektrum jenis instance dapat menemukan konfigurasi yang lebih hemat biaya dalam jangka panjang. Selain itu, algoritme yang melatih paling efisien pada GPU mungkin tidak memerlukan GPU untuk inferensi yang efisien. Eksperimen untuk menentukan solusi efektivitas biaya yang paling banyak. Untuk mendapatkan rekomendasi instans otomatis atau melakukan uji pemuatan khusus, gunakan Amazon SageMaker Inference Recommender.

Untuk informasi selengkapnya tentang spesifikasi SageMaker perangkat keras, lihat Jenis SageMaker Instance Amazon Amazon.

Log untuk Algoritma Bawaan

SageMaker Algoritma Amazon menghasilkan CloudWatch log Amazon, yang memberikan informasi rinci tentang proses pelatihan. Untuk melihat log, di konsol AWS manajemen, pilih, pilih Log CloudWatch, lalu pilih grup log TrainingJobs /aws/sagemaker/. Setiap pekerjaan pelatihan memiliki satu aliran log per node tempat ia dilatih. Nama log stream dimulai dengan nilai yang ditentukan dalam `TrainingJobName` parameter saat pekerjaan dibuat.

> **Note**
>
> Jika pekerjaan gagal dan log tidak muncul CloudWatch, kemungkinan kesalahan terjadi sebelum dimulainya pelatihan. Alasannya termasuk menentukan gambar pelatihan yang salah atau lokasi S3.

Isi log bervariasi menurut algoritma. Namun, Anda biasanya dapat menemukan informasi berikut:

- Konfirmasi argumen yang diberikan di awal log
- Kesalahan yang terjadi selama pelatihan
- Pengukuran akurasi algoritma atau kinerja numerik
- Pengaturan waktu untuk algoritme dan tahapan utama apa pun dalam algoritme

Kesalahan Umum

Jika pekerjaan pelatihan gagal, beberapa detail tentang kegagalan diberikan oleh nilai `FailureReason` pengembalian dalam uraian pekerjaan pelatihan, sebagai berikut:

```
sage = boto3.client('sagemaker')
```

```
sage.describe_training_job(TrainingJobName=job_name)['FailureReason']
```

Lainnya dilaporkan hanya di CloudWatch log. Kesalahan umum termasuk yang berikut:

1. Salah menentukan hyperparameter atau menentukan hyperparameter yang tidak valid untuk algoritma.

   Dari CloudWatch Log

   ```
   [10/16/2017 23:45:17 ERROR 139623806805824 train.py:48]
   Additional properties are not allowed (u'mini_batch_siz' was
   unexpected)
   ```

2. Menentukan nilai yang tidak valid untuk hyperparameter.

   FailureReason

   ```
   AlgorithmError: u'abc' is not valid under any of the given
   schemas\n\nFailed validating u'oneOf' in
   schema[u'properties'][u'feature_dim']:\n    {u'oneOf':
   [{u'pattern': u'^([1-9][0-9]*)$', u'type': u'string'},\n
   {u'minimum': 1, u'type': u'integer'}]}\
   ```

   FailureReason

   ```
   [10/16/2017 23:57:17 ERROR 140373086025536 train.py:48] u'abc'
   is not valid under any of the given schemas
   ```

3. Format file protobuf yang tidak akurat.

   Dari CloudWatch log

   ```
   [10/17/2017 18:01:04 ERROR 140234860816192 train.py:48] cannot
                    copy sequence with size 785 to array axis with dimension 784
   ```

Built-inSageMakerAlgoritma untuk Data Tabular

AmazonSageMakermenyediakan algoritma built-in yang disesuaikan dengan analisis data tabular. Data tabular mengacu pada kumpulan data apa pun yang diatur dalam tabel yang terdiri dari baris

(pengamatan) dan kolom (fitur). Built-inSageMakeralgoritma untuk data tabular dapat digunakan baik untuk klasifikasi atau masalah regresi.

- AutoGluon-Tabular—kerangka AutoKL open-source yang berhasil dengan mengansambel model dan menumpuknya dalam beberapa lapisan.
- CatBoost—implementasi dari algoritme pohon yang ditingkatkan gradien yang memperkenalkan peningkatan yang teratur dan algoritma inovatif untuk memproses fitur kategoris.
- Algoritme faktorisasi—ekstensi dari model linier yang dirancang untuk secara ekonomis menangkap interaksi antara fitur dalam dataset jarang dimensi tinggi.
- Algoritme k-Nearest Neighbor (k-NN)—a metode non-parametrik yang menggunakan k titik berlabel terdekat untuk menetapkan label ke titik data baru untuk klasifikasi atau nilai target yang diprediksi dari rata-rata titik k terdekat untuk regresi.
- LightGBM—implementasi dari algoritma pohon yang ditingkatkan gradien yang menambahkan dua teknik baru untuk meningkatkan efisiensi dan skalabilitas: Gradient berbasis One-Side Sampling (GOSS) dan Exclusive Feature Bundling (EFB).
- Algoritma Linear—learns fungsi linear untuk regresi atau fungsi ambang linear untuk klasifikasi.
- TabTransformer—a novel dalam tabular data pemodelan arsitektur dibangun di atasself-attention-basedTransformer.
- Algoritma XGBoost—implementasi dari algoritme pohon yang ditingkatkan gradien yang menggabungkan ansambel perkiraan dari serangkaian model yang lebih sederhana dan lebih lemah.

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas Instans	Paralleli zable	
AutoGluon-Tabular	pelatihan dan (opsional) validasi	File	CSV	CPU atau GPU (hanya instans tunggal)	Tidak	
CatBoost	pelatihan dan	File	CSV	CPU (hanya	Tidak	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas Instans	Paralleli zable	
	(opsional) validasi			contoh tunggal)		
Mesin Faktorisasi	kereta api dan (opsional) tes	File atau Pipa	Recordio-protobuf	CPU (GPU untuk data padat)	Ya	
K-Tetangga Terdekat (K-nN)	kereta api dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPU (perangkat GPU tunggal pada satu atau beberapa instance)	Ya	
LightGBM	pelatihan dan (opsional) validasi	File	CSV	CPU (hanya contoh tunggal)	Tidak	
Peserta Linear	melatih dan (opsional) validasi, tes, atau keduanya	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPU	Ya	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas Instans	Paralleli zable	
TabTransformer	pelatihan dan (opsional) validasi	File	CSV	CPU atau GPU (hanya instans tunggal)	Tidak	
XGBoost (0,90-1, 0,90-2, 1,0-1, 1,2-1, 1,2-21)	melatih dan (opsional) validasi	File atau Pipa	CSV, LibSVM, atau Parket	CPU (atau GPU untuk 1,2-1)	Ya	

AutoGluon-Tabular

AutoGluon-Tabular adalah kerangka kerja AutoML open-source populer yang melatih model pembelajaran mesin yang sangat akurat pada kumpulan data tabular yang belum diproses. Tidak seperti kerangka kerja AutoML yang ada yang terutama berfokus pada pemilihan model dan hiperparameter, AutoGluon -Tabular berhasil dengan menyamarkan beberapa model dan menumpuknya dalam beberapa lapisan.

Cara menggunakan SageMaker AutoGluon -Tabular

Anda dapat menggunakan AutoGluon -Tabular sebagai algoritma SageMaker bawaan Amazon. Bagian berikut menjelaskan cara menggunakan AutoGluon -Tabular dengan Python SageMaker SDK. Untuk informasi tentang cara menggunakan AutoGluon -Tabular dari Amazon SageMaker Studio Classic UI, lihat. SageMaker JumpStart

- Gunakan AutoGluon -Tabular sebagai algoritma bawaan

  Gunakan algoritma bawaan AutoGluon -Tabular untuk membangun wadah pelatihan AutoGluon -Tabular seperti yang ditunjukkan pada contoh kode berikut. Anda dapat secara otomatis melihat URI image algoritma bawaan AutoGluon -Tabular menggunakan SageMaker

`image_uris.retrieve` API (atau `get_image_uri` API jika menggunakan Amazon SageMaker Python SDK versi 2).

Setelah menentukan URI image AutoGluon -Tabular, Anda dapat menggunakan container AutoGluon -Tabular untuk membuat estimator menggunakan Estimator API dan memulai pekerjaan pelatihan SageMaker . Algoritma bawaan AutoGluon -Tabular berjalan dalam mode skrip, tetapi skrip pelatihan disediakan untuk Anda dan tidak perlu menggantinya. Jika Anda memiliki pengalaman luas menggunakan mode skrip untuk membuat pekerjaan SageMaker pelatihan, maka Anda dapat memasukkan skrip pelatihan AutoGluon -Tabular Anda sendiri.

```
from sagemaker import image_uris, model_uris, script_uris

train_model_id, train_model_version, train_scope = "autogluon-classification-
ensemble", "*", "training"
training_instance_type = "ml.p3.2xlarge"

# Retrieve the docker image
train_image_uri = image_uris.retrieve(
    region=None,
    framework=None,
    model_id=train_model_id,
    model_version=train_model_version,
    image_scope=train_scope,
    instance_type=training_instance_type
)

# Retrieve the training script
train_source_uri = script_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 script_scope=train_scope
)

train_model_uri = model_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 model_scope=train_scope
)

# Sample training data is available in this bucket
training_data_bucket = f"jumpstart-cache-prod-{aws_region}"
training_data_prefix = "training-datasets/tabular_binary/"
```

```
training_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
train"
validation_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
validation"

output_bucket = sess.default_bucket()
output_prefix = "jumpstart-example-tabular-training"

s3_output_location = f"s3://{output_bucket}/{output_prefix}/output"

from sagemaker import hyperparameters

# Retrieve the default hyperparameters for training the model
hyperparameters = hyperparameters.retrieve_default(
    model_id=train_model_id, model_version=train_model_version
)

# [Optional] Override default hyperparameters with custom values
hyperparameters[
    "auto_stack"
] = "True"
print(hyperparameters)

from sagemaker.estimator import Estimator
from sagemaker.utils import name_from_base

training_job_name = name_from_base(f"built-in-algo-{train_model_id}-training")

# Create SageMaker Estimator instance
tabular_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1,
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location
)

# Launch a SageMaker Training job by passing the S3 path of the training data
tabular_estimator.fit(
```

```
    {
        "training": training_dataset_s3_path,
        "validation": validation_dataset_s3_path,
    }, logs=True, job_name=training_job_name
)
```

Untuk informasi selengkapnya tentang cara mengatur AutoGluon -Tabular sebagai algoritma bawaan, lihat contoh notebook berikut. Bucket S3 apa pun yang digunakan dalam contoh ini harus berada di AWS Region yang sama dengan instance notebook yang digunakan untuk menjalankannya.

- Klasifikasi tabel dengan algoritma Amazon SageMaker AutoGluon -Tabular
- Regresi tabular dengan algoritma Amazon -Tabular SageMaker AutoGluon

Antarmuka Input dan Output untuk algoritma AutoGluon -Tabular

Peningkatan gradien beroperasi pada data tabular, dengan baris mewakili pengamatan, satu kolom mewakili variabel target atau label, dan kolom yang tersisa mewakili fitur.

SageMaker Implementasi AutoGluon -Tabular mendukung CSV untuk pelatihan dan inferensi:

- Untuk Pelatihan ContentType, input yang valid harus teks/csv.
- Untuk Inferensi ContentType, input yang valid harus teks/csv.

> **Note**
>
> Untuk pelatihan CSV, algoritme mengasumsikan bahwa variabel target ada di kolom pertama dan CSV tidak memiliki catatan header.
> Untuk inferensi CSV, algoritme mengasumsikan bahwa input CSV tidak memiliki kolom label.

Format input untuk data pelatihan, data validasi, dan fitur kategoris

Perhatikan cara memformat data pelatihan Anda untuk masukan ke model AutoGluon -Tabular. Anda harus menyediakan jalur ke bucket Amazon S3 yang berisi data pelatihan dan validasi Anda. Anda juga dapat menyertakan daftar fitur kategoris. Gunakan saluran `training` dan `validation` saluran untuk memberikan data input Anda. Atau, Anda hanya dapat menggunakan `training` saluran.

Gunakan kedua **`validation`** saluran **`training`** dan

Anda dapat memberikan data input Anda melalui dua jalur S3, satu untuk `training` saluran dan satu untuk `validation` saluran. Setiap jalur S3 dapat berupa awalan S3 atau jalur S3 lengkap yang menunjuk ke satu file CSV tertentu. Variabel target harus berada di kolom pertama file CSV Anda. Variabel prediktor (fitur) harus berada di kolom yang tersisa. Data validasi digunakan untuk menghitung skor validasi di akhir setiap iterasi peningkatan. Penghentian awal diterapkan ketika skor validasi berhenti membaik.

Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan file data pelatihan Anda. Jika Anda menyediakan file JSON untuk fitur kategoris, `training` saluran Anda harus menunjuk ke awalan S3 dan bukan file CSV tertentu. File ini harus berisi kamus Python di mana kuncinya adalah string `"cat_index_list"` dan nilainya adalah daftar bilangan bulat unik. Setiap bilangan bulat dalam daftar nilai harus menunjukkan indeks kolom dari fitur kategoris yang sesuai dalam file CSV data pelatihan Anda. Setiap nilai harus berupa bilangan bulat positif (lebih besar dari nol karena nol mewakili nilai target), kurang dari `Int32.MaxValue` (2147483647), dan kurang dari jumlah kolom. Seharusnya hanya ada satu file JSON indeks kategoris.

Gunakan hanya **`training`** saluran:

Sebagai alternatif, Anda dapat memberikan data input Anda melalui jalur S3 tunggal untuk `training` saluran tersebut. Path S3 ini harus menunjuk ke direktori dengan subdirektori bernama `training/` yang berisi file CSV. Anda dapat secara opsional menyertakan subdirektori lain di lokasi yang sama yang disebut `validation/` yang juga memiliki file CSV. Jika data validasi tidak disediakan, maka 20% data pelatihan Anda diambil sampelnya secara acak untuk dijadikan data validasi. Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan subdirektori data Anda.

> **Note**
>
> Untuk mode input pelatihan CSV, total memori yang tersedia untuk algoritme (jumlah instance dikalikan dengan memori yang tersedia di`InstanceType`) harus dapat menampung kumpulan data pelatihan.

SageMaker AutoGluon-Tabular menggunakan `autogluon.tabular.TabularPredictor` modul untuk membuat serial atau deserialisasi model, yang dapat digunakan untuk menyimpan atau memuat model.

Untuk menggunakan model yang dilatih dengan SageMaker AutoGluon -Tabular dengan kerangka kerja AutoGluon

- Gunakan kode Python berikut:

```
import tarfile
from autogluon.tabular import TabularPredictor

t = tarfile.open('model.tar.gz', 'r:gz')
t.extractall()

model = TabularPredictor.load(model_file_path)

# prediction with test data
# dtest should be a pandas DataFrame with column names feature_0, feature_1, ...,
 feature_d
pred = model.predict(dtest)
```

Rekomendasi instans Amazon EC2 untuk algoritme AutoGluon -Tabular

SageMaker AutoGluon-Tabular mendukung pelatihan CPU satu instans dan GPU instans tunggal. Meskipun biaya per instans lebih tinggi, GPU berlatih lebih cepat, membuatnya lebih hemat biaya. Untuk memanfaatkan pelatihan GPU, tentukan jenis instans sebagai salah satu instance GPU (misalnya, P3). SageMaker AutoGluon-Tabular saat ini tidak mendukung pelatihan multi-GPU.

AutoGluon-Notebook sampel tabel

Tabel berikut menguraikan berbagai contoh notebook yang membahas berbagai kasus penggunaan algoritma Amazon SageMaker AutoGluon -Tabular.

Judul Notebook	Deskripsi
Klasifikasi tabel dengan algoritma Amazon SageMaker AutoGluon -Tabular	Notebook ini menunjukkan penggunaan algoritma Amazon SageMaker AutoGluon -Tabular untuk melatih dan menjadi tuan rumah model klasifikasi tabel.
Regresi tabular dengan algoritma Amazon -Tabular SageMaker AutoGluon	Notebook ini menunjukkan penggunaan algoritma Amazon SageMaker AutoGluon -

Judul Notebook	Deskripsi
	Tabular untuk melatih dan menjadi tuan rumah model regresi tabular.

Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Bagaimana AutoGluon -Tabular bekerja

AutoGluon-Tabular melakukan pemrosesan data tingkat lanjut, pembelajaran mendalam, dan metode ansambel model multi-layer. Secara otomatis mengenali tipe data di setiap kolom untuk preprocessing data yang kuat, termasuk penanganan khusus bidang teks.

AutoGluon cocok untuk berbagai model mulai dari pohon yang off-the-shelf ditingkatkan hingga jaringan saraf yang disesuaikan. Model-model ini diansambel dengan cara baru: model ditumpuk dalam beberapa lapisan dan dilatih secara berlapis yang menjamin data mentah dapat diterjemahkan ke dalam prediksi berkualitas tinggi dalam batasan waktu tertentu. Proses ini mengurangi overfitting dengan membagi data dengan berbagai cara dengan pelacakan contoh yang cermat. out-of-fold

Algoritma AutoGluon -Tabular berkinerja baik dalam kompetisi pembelajaran mesin karena penanganannya yang kuat dari berbagai tipe data, hubungan, dan distribusi. Anda dapat menggunakan AutoGluon -Tabular untuk regresi, klasifikasi (biner dan multiclass), dan masalah peringkat.

Lihat diagram berikut yang menggambarkan cara kerja strategi susun multi-layer.

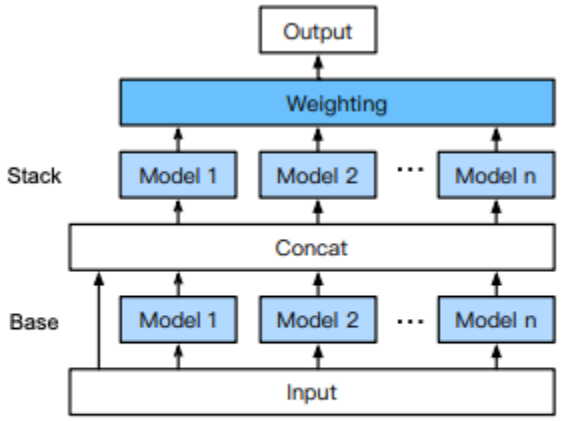


Figure 2. AutoGluon's multi-layer stacking strategy, shown here using two stacking layers and n types of base learners.

Untuk informasi selengkapnya, lihat AutoGluon-Tabular: AutoML yang Kuat dan Akurat untuk Data Terstruktur.

AutoGluon-Hiperparameter tabel

Tabel berikut berisi subset hiperparameter yang diperlukan atau paling umum digunakan untuk algoritma Amazon SageMaker AutoGluon -Tabular. Pengguna mengatur parameter ini untuk memfasilitasi estimasi parameter model dari data. Algoritma SageMaker AutoGluon -Tabular adalah implementasi dari paket -Tabular open-sourceAutoGluon.

> **Note**
>
> Hyperparameter default didasarkan pada contoh kumpulan data di file. AutoGluon-Notebook sampel tabel

Secara default, algoritma SageMaker AutoGluon -Tabular secara otomatis memilih metrik evaluasi berdasarkan jenis masalah klasifikasi. Algoritma mendeteksi jenis masalah klasifikasi berdasarkan jumlah label dalam data Anda. Untuk masalah regresi, metrik evaluasi adalah kesalahan kuadrat rata-rata akar. Untuk masalah klasifikasi biner, metrik evaluasi adalah area di bawah kurva karakteristik operasi penerima (AUC). Untuk masalah klasifikasi multikelas, metrik evaluasi adalah akurasi. Anda dapat menggunakan `eval_metric` hyperparameter untuk mengubah metrik evaluasi default. Lihat tabel berikut untuk informasi lebih lanjut tentang hiperparameter AutoGluon -Tabular, termasuk deskripsi, nilai yang valid, dan nilai default.

Nama Parameter	Deskripsi
`eval_metric`	Metrik evaluasi untuk data validasi. Jika `eval_metric` diatur ke `"auto"` nilai default, maka algoritma secara otomatis memilih metrik evaluasi berdasarkan jenis masalah klasifikasi: • `"root_mean_squared_error"` untuk regresi • `"roc_auc"` untuk klasifikasi biner • `"accuracy"` untuk klasifikasi multi-kelas Nilai yang valid: string, lihat AutoGluon dokumentasi untuk nilai yang valid. Nilai default:`"auto"`.
`presets`	Daftar konfigurasi preset untuk berbagai argumen di. `fit()` • `"best_quality"` : akurasi prediktif tinggi, waktu inferensi lebih lambat dan penggunaan disk yang lebih tinggi • `"high_quality"` : akurasi prediktif tinggi dan inferensi cepat • `"good_quality"` : akurasi prediktif yang baik dan inferensi yang sangat cepat • `"medium_quality"` : akurasi prediktif sedang, inferensi dan waktu pelatihan yang sangat cepat • `"optimize_for_deployment"` : hapus model yang tidak digunakan dan hapus artefak pelatihan • `"interpretable"` : hanya cocok dengan model berbasis aturan yang dapat ditafsirkan dari paket `imodels` Untuk lebih jelasnya, lihat AutoGluon Prediktor. Nilai yang valid: string, salah satu dari berikut ini: (`"best_quality" "high_quality" ,,good_quality" ,"medium_quality" ,"optimize_for_deployment" , or "interpretable" `).

Nama Parameter	Deskripsi
	Nilai default:"medium_quality" .
auto_stack	Apakah AutoGluon harus secara otomatis menggunakan bagging dan multi-layer stack ansambling untuk meningkat kan akurasi prediktif. Atur auto_stack ke "True" jika Anda bersedia mentolerir waktu pelatihan yang lebih lama untuk memaksimalkan akurasi prediktif. Ini secara otomatis menetapka n num_stack_levels argumen num_bag_folds dan berdasarkan properti dataset. Nilai yang valid: string, "True" atau"False". Nilai default:"False".
num_bag_folds	Jumlah lipatan yang digunakan untuk mengantongi model. num_bag_folds Kapan sama dengank, waktu pelatihan secara kasar meningkat dengan faktork. Setel num_bag_f olds ke 0 untuk menonaktifkan bagging. Ini dinonaktifkan secara default, tetapi kami sarankan menggunakan nilai antara 5 dan 10 untuk memaksimalkan kinerja prediktif. Peningkatan num_bag_folds hasil dalam model dengan bias yang lebih rendah, tetapi itu lebih rentan terhadap overfitting. Salah satunya adalah nilai yang tidak valid untuk parameter ini, dan akan menaikkan nilai. ValueError Nilai yang lebih besar dari 10 dapat menghasilkan pengembalian yang berkurang dan bahkan dapat merusak hasil keseluruhan karena overfitting. Untuk lebih meningkatkan prediksi, hindari peningkatan num_bag_folds dan sebaliknya tingkatkannum_bag_sets . Nilai yang valid: string, bilangan bulat apa pun antara (dan termasuk) "0" dan"10". Nilai default:"0".

Nama Parameter	Deskripsi
num_bag_sets	Jumlah pengulangan kfold bagging untuk dilakukan (nilai harus lebih besar dari atau sama dengan 1). Jumlah total model yang dilatih selama pengantongan sama num_bag_folds dengan*num_bag_sets . Parameter ini default ke satu jika time_limit tidak ditentukan. Parameter ini dinonaktifkan num_bag_folds jika tidak ditentukan. Nilai yang lebih besar dari satu menghasilkan kinerja prediktif yang unggul, terutama pada masalah yang lebih kecil dan dengan penumpukan diaktifkan. Nilai yang valid: integer, range: [1,20]. Nilai default:1.
num_stack_levels	Jumlah tingkat susun untuk digunakan dalam ansambel tumpukan. Secara kasar meningkatkan waktu pelatihan model dengan faktor num_stack_levels + 1. Setel parameter ini ke 0 untuk menonaktifkan ansambel tumpukan. Parameter ini dinonaktifkan secara default, tetapi kami sarankan menggunakan nilai antara 1 dan 3 untuk memaksimalkan kinerja prediktif. Untuk mencegah overfitting dan aValueError , num_bag_folds harus lebih besar dari atau sama dengan 2. Nilai yang valid: float, range: [0,3]. Nilai default:0.
refit_full	Apakah akan melatih ulang semua model pada semua data (pelatihan dan validasi) setelah prosedur pelatihan normal atau tidak. Untuk lebih jelasnya, lihat AutoGluon Prediktor. Nilai yang valid: string, "True" atau"False". Nilai default:"False".

Nama Parameter	Deskripsi
`set_best_to_refit_full`	Apakah akan mengubah model default yang digunakan prediktor untuk prediksi atau tidak. Jika `set_best_to_refit_full` disetel ke`"True"`, model default berubah ke model yang menunjukkan skor validasi tertinggi sebagai hasil dari refitting (diaktifkan oleh). `refit_full` Hanya valid jika `refit_full` disetel. Nilai yang valid: string, `"True"` atau`"False"`. Nilai default:`"False"`.
`save_space`	Apakah atau perhatikan untuk mengurangi memori dan ukuran disk prediktor dengan menghapus file model tambahan yang tidak diperlukan untuk prediksi pada data baru. Ini tidak berdampak pada akurasi inferensi. Kami merekomendasikan pengaturan `save_space` `"True"` apakah satu-satunya tujuan adalah menggunakan model terlatih untuk prediksi. Fungsionalitas lanjutan tertentu mungkin tidak lagi tersedia jika `save_space` disetel ke`"True"`. Lihat `predictor.save_space()` dokumentasi untuk lebih jelasnya. Nilai yang valid: string, `"True"` atau`"False"`. Nilai default:`"False"`.
`verbosity`	Verbositas pesan cetak. `verbosity` tingkat berkisar dari `0` ke`4`, dengan tingkat yang lebih tinggi sesuai dengan pernyataan cetak yang lebih rinci. Sebuah `verbosity` dari `0` menekan peringatan. Nilai yang valid: bilangan bulat, salah satu dari berikut ini: (`0`,`1`,`2`,`3`, atau`4`). Nilai default:2.

Menyetel model AutoGluon -Tabular

Meskipun AutoGluon -Tabular dapat digunakan dengan penyetelan model, desainnya dapat memberikan kinerja yang baik menggunakan metode susun dan ansambel, yang berarti optimasi hyperparameter tidak diperlukan. Alih-alih berfokus pada penyetelan model, AutoGluon -Tabular berhasil dengan menumpuk model dalam beberapa lapisan dan pelatihan dengan cara yang bijaksana.

Untuk informasi lebih lanjut tentang AutoGluon -Tabular hyperparameters, lihat. AutoGluon-Hiperparameter tabel

CatBoost

CatBoostadalah implementasi open-source yang populer dan berkinerja tinggi dari algoritma Gradient Boosting Decision Tree (GBDT). GBDT adalah algoritma pembelajaran yang diawasi yang mencoba memprediksi variabel target secara akurat dengan menggabungkan ansambel perkiraan dari serangkaian model yang lebih sederhana dan lebih lemah.

CatBoost memperkenalkan dua kemajuan algoritmik penting ke GBDT:

1. Implementasi peningkatan berurutan, alternatif yang digerakkan oleh permutasi untuk algoritme klasik
2. Algoritma inovatif untuk memproses fitur kategoris

Kedua teknik diciptakan untuk melawan pergeseran prediksi yang disebabkan oleh jenis kebocoran target khusus yang ada di semua implementasi algoritma peningkatan gradien yang ada saat ini.

Cara menggunakan SageMaker CatBoost

Anda dapat menggunakan CatBoost sebagai algoritma SageMaker bawaan Amazon. Bagian berikut menjelaskan cara menggunakan CatBoost dengan SageMaker Python SDK. Untuk informasi tentang cara menggunakan CatBoost dari Amazon SageMaker Studio Classic UI, lihatSageMaker JumpStart.

- Gunakan CatBoost sebagai algoritma bawaan

  Gunakan algoritma CatBoost bawaan untuk membangun wadah CatBoost pelatihan seperti yang ditunjukkan pada contoh kode berikut. Anda dapat secara otomatis melihat URI image algoritme CatBoost bawaan menggunakan SageMaker `image_uris.retrieve` API (atau `get_image_uri` API jika menggunakan Amazon SageMaker Python SDK versi 2).

Setelah menentukan URI CatBoost image, Anda dapat menggunakan CatBoost container untuk membuat estimator menggunakan SageMaker Estimator API dan memulai tugas pelatihan. Algoritma CatBoost bawaan berjalan dalam mode skrip, tetapi skrip pelatihan disediakan untuk Anda dan tidak perlu menggantinya. Jika Anda memiliki pengalaman luas menggunakan mode skrip untuk membuat pekerjaan SageMaker pelatihan, maka Anda dapat memasukkan skrip CatBoost pelatihan Anda sendiri.

```
from sagemaker import image_uris, model_uris, script_uris

train_model_id, train_model_version, train_scope = "catboost-classification-model",
 "*", "training"
training_instance_type = "ml.m5.xlarge"

# Retrieve the docker image
train_image_uri = image_uris.retrieve(
    region=None,
    framework=None,
    model_id=train_model_id,
    model_version=train_model_version,
    image_scope=train_scope,
    instance_type=training_instance_type
)

# Retrieve the training script
train_source_uri = script_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 script_scope=train_scope
)

train_model_uri = model_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 model_scope=train_scope
)

# Sample training data is available in this bucket
training_data_bucket = f"jumpstart-cache-prod-{aws_region}"
training_data_prefix = "training-datasets/tabular_multiclass/"

training_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
train"
validation_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
validation"
```

```
output_bucket = sess.default_bucket()
output_prefix = "jumpstart-example-tabular-training"

s3_output_location = f"s3://{output_bucket}/{output_prefix}/output"

from sagemaker import hyperparameters

# Retrieve the default hyperparameters for training the model
hyperparameters = hyperparameters.retrieve_default(
    model_id=train_model_id, model_version=train_model_version
)

# [Optional] Override default hyperparameters with custom values
hyperparameters[
    "iterations"
] = "500"
print(hyperparameters)

from sagemaker.estimator import Estimator
from sagemaker.utils import name_from_base

training_job_name = name_from_base(f"built-in-algo-{train_model_id}-training")

# Create SageMaker Estimator instance
tabular_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1,
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location
)

# Launch a SageMaker Training job by passing the S3 path of the training data
tabular_estimator.fit(
    {
        "training": training_dataset_s3_path,
        "validation": validation_dataset_s3_path,
    }, logs=True, job_name=training_job_name
```

```
)
```

Untuk informasi selengkapnya tentang cara mengatur CatBoost sebagai algoritma bawaan, lihat contoh buku catatan berikut.

- Klasifikasi tabel dengan Amazon SageMaker LightGBM dan algoritma CatBoost
- Regresi tabular dengan Amazon SageMaker LightGBM dan algoritma CatBoost

Antarmuka Input dan Output untuk CatBoost algoritma

Peningkatan gradien beroperasi pada data tabular, dengan baris mewakili pengamatan, satu kolom mewakili variabel target atau label, dan kolom yang tersisa mewakili fitur.

SageMaker Implementasi CatBoost dukungan CSV untuk pelatihan dan inferensi:

- Untuk Pelatihan ContentType, input yang valid harus berupa teks/csv.
- Untuk Inferensi ContentType, input yang valid harus teks/csv.

> **Note**
>
> Untuk pelatihan CSV, algoritme mengasumsikan bahwa variabel target ada di kolom pertama dan CSV tidak memiliki catatan header.
> Untuk inferensi CSV, algoritme mengasumsikan bahwa input CSV tidak memiliki kolom label.

Format input untuk data pelatihan, data validasi, dan fitur kategoris

Perhatikan cara memformat data pelatihan Anda untuk masukan ke CatBoost model. Anda harus menyediakan jalur ke bucket Amazon S3 yang berisi data pelatihan dan validasi Anda. Anda juga dapat menyertakan daftar fitur kategoris. Gunakan saluran `training` dan `validation` saluran untuk memberikan data masukan Anda. Atau, Anda hanya dapat menggunakan `training` saluran.

Gunakan kedua **`validation`** saluran **`training`** dan

Anda dapat memberikan data input Anda melalui dua jalur S3, satu untuk `training` saluran dan satu untuk `validation` saluran. Setiap jalur S3 dapat berupa awalan S3 yang menunjuk ke satu atau lebih file CSV atau jalur S3 lengkap yang menunjuk ke satu file CSV tertentu. Variabel target harus berada di kolom pertama file CSV Anda. Variabel prediktor (fitur) harus berada di kolom yang tersisa. Jika beberapa file CSV disediakan untuk `validation` saluran `training` atau, CatBoost

algoritme menggabungkan file. Data validasi digunakan untuk menghitung skor validasi di akhir setiap iterasi peningkatan. Penghentian awal diterapkan ketika skor validasi berhenti membaik.

Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan file atau file data pelatihan Anda. Jika Anda menyediakan file JSON untuk fitur kategoris, `training` saluran Anda harus menunjuk ke awalan S3 dan bukan file CSV tertentu. File ini harus berisi kamus Python di mana kuncinya adalah string `"cat_index_list"` dan nilainya adalah daftar bilangan bulat unik. Setiap bilangan bulat dalam daftar nilai harus menunjukkan indeks kolom dari fitur kategoris yang sesuai dalam file CSV data pelatihan Anda. Setiap nilai harus berupa bilangan bulat positif (lebih besar dari nol karena nol mewakili nilai target), kurang dari `Int32.MaxValue` (2147483647), dan kurang dari jumlah kolom. Seharusnya hanya ada satu file JSON indeks kategoris.

Gunakan hanya **`training`** saluran:

Sebagai alternatif, Anda dapat memberikan data input Anda melalui jalur S3 tunggal untuk `training` saluran tersebut. Jalur S3 ini harus menunjuk ke direktori dengan subdirektori bernama `training/` yang berisi satu atau lebih file CSV. Anda dapat secara opsional menyertakan subdirektori lain di lokasi yang sama yang disebut `validation/` yang juga memiliki satu atau lebih file CSV. Jika data validasi tidak disediakan, maka 20% data pelatihan Anda diambil sampelnya secara acak untuk dijadikan data validasi. Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan subdirektori data Anda.

> **Note**
>
> Untuk mode input pelatihan CSV, total memori yang tersedia untuk algoritme (jumlah instance dikalikan dengan memori yang tersedia di`InstanceType`) harus dapat menampung kumpulan data pelatihan.

SageMaker CatBoost menggunakan `catboost.CatBoostRegressor` modul `catboost.CatBoostClassifier` dan untuk membuat serial atau deserialisasi model, yang dapat digunakan untuk menyimpan atau memuat model.

Untuk menggunakan model yang dilatih SageMaker CatBoost dengan **`catboost`**

- Gunakan kode Python berikut:

```
import tarfile
```

```
from catboost import CatBoostClassifier

t = tarfile.open('model.tar.gz', 'r:gz')
t.extractall()

file_path = os.path.join(model_file_path, "model")
model = CatBoostClassifier()
model.load_model(file_path)

# prediction with test data
# dtest should be a pandas DataFrame with column names feature_0, feature_1, ...,
 feature_d
pred = model.predict(dtest)
```

Rekomendasi instans Amazon EC2 untuk algoritme CatBoost

SageMaker CatBoost saat ini hanya melatih menggunakan CPU. CatBoost adalah algoritma yang terikat memori (sebagai lawan dari compute-bound). Jadi, instance komputasi tujuan umum (misalnya, M5) adalah pilihan yang lebih baik daripada instance yang dioptimalkan komputasi (misalnya, C5). Selanjutnya, kami menyarankan Anda memiliki memori total yang cukup dalam instance yang dipilih untuk menyimpan data pelatihan.

CatBoost contoh notebook

Tabel berikut menguraikan berbagai contoh notebook yang membahas berbagai kasus penggunaan algoritma Amazon SageMaker CatBoost .

Judul Notebook	Deskripsi
Klasifikasi tabel dengan Amazon SageMaker LightGBM dan algoritma CatBoost	Notebook ini menunjukkan penggunaan SageMaker CatBoost algoritma Amazon untuk melatih dan menjadi tuan rumah model klasifikasi tabel.
Regresi tabular dengan Amazon SageMaker LightGBM dan algoritma CatBoost	Notebook ini menunjukkan penggunaan SageMaker CatBoost algoritma Amazon untuk melatih dan menjadi tuan rumah model regresi tabular.

Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Cara CatBoost Kerja

CatBoost mengimplementasikan algoritma Gradient Boosting Decision Tree (GBDT) konvensional dengan penambahan dua kemajuan algoritmik penting:

1. Implementasi peningkatan berurutan, alternatif yang digerakkan oleh permutasi untuk algoritme klasik
2. Algoritma inovatif untuk memproses fitur kategoris

Kedua teknik diciptakan untuk melawan pergeseran prediksi yang disebabkan oleh jenis kebocoran target khusus yang ada di semua implementasi algoritma peningkatan gradien yang ada saat ini.

CatBoost Algoritma ini berkinerja baik dalam kompetisi pembelajaran mesin karena penanganannya yang kuat terhadap berbagai tipe data, hubungan, distribusi, dan keragaman hiperparameter yang dapat Anda sesuaikan. Anda dapat menggunakan CatBoost untuk regresi, klasifikasi (biner dan multiclass), dan masalah peringkat.

Untuk informasi lebih lanjut tentang peningkatan gradien, lihat. Bagaimana XGBoost Bekerja Untuk detail mendalam tentang teknik GOSS dan EFB tambahan yang digunakan dalam CatBoost metode ini, lihat CatBoost: peningkatan yang tidak bias dengan fitur kategoris.

CatBoost hiperparameter

Tabel berikut berisi subset hiperparameter yang diperlukan atau paling umum digunakan untuk algoritma Amazon SageMaker CatBoost . Pengguna mengatur parameter ini untuk memfasilitasi estimasi parameter model dari data. SageMaker CatBoost Algoritma adalah implementasi dari CatBoostpaket open-source.

> **Note**
>
> Hyperparameter default didasarkan pada contoh kumpulan data di file. CatBoost contoh notebook

Secara default, SageMaker CatBoost algoritme secara otomatis memilih metrik evaluasi dan fungsi kerugian berdasarkan jenis masalah klasifikasi. CatBoost Algoritma mendeteksi jenis masalah klasifikasi berdasarkan jumlah label dalam data Anda. Untuk masalah regresi, metrik evaluasi dan fungsi kerugian keduanya merupakan kesalahan kuadrat rata-rata akar. Untuk masalah klasifikasi biner, metrik evaluasi adalah Area Under the Curve (AUC) dan fungsi kerugiannya adalah kehilangan log. Untuk masalah klasifikasi multikelas, metrik evaluasi dan fungsi kerugian adalah entropi silang multikelas. Anda dapat menggunakan `eval_metric` hyperparameter untuk mengubah metrik evaluasi default. Lihat tabel berikut untuk informasi selengkapnya tentang hyperparameters LightGBM, termasuk deskripsi, nilai valid, dan nilai default.

Nama Parameter	Deskripsi
`iterations`	Jumlah maksimum pohon yang dapat dibangun. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat positif. Nilai default:`500`.
`early_stopping_rounds`	Pelatihan akan berhenti jika satu metrik dari satu titik data validasi tidak membaik di `early_stopping_rounds` babak terakhir. Jika `early_stopping_rounds` kurang dari atau sama dengan nol, hyperparameter ini diabaikan. Nilai yang valid: bilangan bulat. Nilai default:`5`.
`eval_metric`	Metrik evaluasi untuk data validasi. Jika `eval_metric` diatur ke `"auto"` nilai default, maka algoritma secara otomatis memilih metrik evaluasi berdasarkan jenis masalah klasifikasi: • `"RMSE"`untuk regresi • `"AUC"`untuk klasifikasi biner • `"MultiClass"` untuk klasifikasi multi-kelas Nilai yang valid: string, lihat CatBoost dokumentasi untuk nilai yang valid.

Nama Parameter	Deskripsi
	Nilai default:`"auto"`.
`learning_rate`	Tingkat di mana bobot model diperbarui setelah mengerjakan setiap batch contoh pelatihan. Nilai yang valid: float, range: (`0.0`,`1.0`). Nilai default:`0.009`.
`depth`	Kedalaman pohon. Nilai yang valid: integer, range: (1,16). Nilai default:6.
`l2_leaf_reg`	Koefisien untuk jangka waktu regularisasi L2 dari fungsi biaya. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat positif. Nilai default:3.
`random_strength`	Jumlah keacakan yang digunakan untuk penilaian terbelah ketika struktur pohon dipilih. Gunakan parameter ini untuk menghindari model yang terlalu pas. Nilai yang valid: float, range: Nomor floating point positif. Nilai default:`1.0`.
`max_leaves`	Jumlah maksimum daun di pohon yang dihasilkan. Hanya dapat digunakan dengan kebijakan yang `"Lossguide"` berkemban g. Nilai yang valid: integer, range: [2,64]. Nilai default:31.

Nama Parameter	Deskripsi
`rsm`	Metode subruang acak. Persentase fitur yang akan digunakan pada setiap pemilihan split, ketika fitur dipilih lagi secara acak. Nilai yang valid: float, range: (`0.0`,`1.0`]. Nilai default:`1.0`.
`sampling_frequency`	Frekuensi untuk mengambil sampel bobot dan benda saat membangun pohon. Nilai yang valid: string, baik: (`"PerTreeL evel"` atau`"PerTree"`). Nilai default:`"PerTreeLevel"` .
`min_data_in_leaf`	Jumlah minimum sampel pelatihan dalam satu daun. CatBoost tidak mencari split baru di daun dengan jumlah sampel kurang dari nilai yang ditentukan. Hanya dapat digunakan dengan kebijakan yang `"Depthwise"` berkembang `"Lossguide"` dan berkembang. Nilai yang valid: bilangan bulat, rentang: (1atau∞). Nilai default:`1`.
`bagging_temperature`	Mendefinisikan pengaturan bootstrap Bayesian. Gunakan bootstrap Bayesian untuk menetapkan bobot acak ke objek. Jika `bagging_temperature` diatur ke`1.0`, maka bobot diambil sampelnya dari distribusi eksponensial. Jika `bagging_t emperature` diatur ke`0.0`, maka semua bobot adalah 1.0. Nilai yang valid: float, range: Float non-negatif. Nilai default:`1.0`.

Nama Parameter	Deskripsi
`boosting_type`	Skema peningkatan. “Otomatis” berarti `boosting_type` dipilih berdasarkan jenis unit pemrosesan, jumlah objek dalam kumpulan data pelatihan, dan mode pembelajaran yang dipilih. Nilai yang valid: string, salah satu dari berikut ini: (`"Auto"`,`"Ordered"` ,`"Plain"`). Nilai default:`"Auto"`.
`scale_pos_weight`	Bobot untuk kelas positif dalam klasifikasi biner. Nilai ini digunakan sebagai pengganda untuk bobot objek dari kelas positif. Nilai yang valid: float, range: Positive float. Nilai default:`1.0`.
`max_bin`	Jumlah split untuk fitur numerik. `"Auto"`berarti yang `max_bin` dipilih berdasarkan jenis unit pengolahan dan parameter lainnya. Untuk detailnya, lihat CatBoost dokumentasi. Nilai yang valid: string, baik: (`"Auto"`atau string integer dari `"1"` ke `"65535"` inklusif). Nilai default:`"Auto"`.
`grow_policy`	Kebijakan penanaman pohon. Mendefinisikan bagaimana melakukan konstruksi pohon serakah. Nilai yang valid: string, salah satu dari berikut ini: (`"SymmetricTree"` ,`"Depthwise"` , atau`"Lossguide"`). Nilai default:`"SymmetricTree"` .

Nama Parameter	Deskripsi
`random_seed`	Benih acak yang digunakan untuk pelatihan. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat non-negatif. Nilai default:`1.0`.
`thread_count`	Jumlah utas yang akan digunakan selama pelatihan. Jika `thread_count` ya`-1`, maka jumlah utas sama dengan jumlah inti prosesor. `thread_count` tidak bisa`0`. Nilai valid: integer, baik: (`-1`atau bilangan bulat positif). Nilai default:`-1`.
`verbose`	Verbositas pesan cetak, dengan tingkat yang lebih tinggi sesuai dengan pernyataan cetak yang lebih rinci. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat positif. Nilai default:`1`.

Menyetel CatBoost model

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hiperparameter pada kumpulan data pelatihan dan validasi Anda. Penyetelan model berfokus pada hiperparameter berikut:

> **Note**
>
> Fungsi kehilangan pembelajaran secara otomatis ditetapkan berdasarkan jenis tugas klasifikasi, yang ditentukan oleh jumlah bilangan bulat unik di kolom label. Untuk informasi selengkapnya, lihat CatBoost hiperparameter.

- Fungsi kehilangan belajar untuk mengoptimalkan selama pelatihan model
- Metrik evaluasi yang digunakan untuk mengevaluasi kinerja model selama validasi

- Satu set hyperparameters dan rentang nilai untuk masing-masing untuk digunakan saat menyetel model secara otomatis

Penyetelan model otomatis mencari hiperparameter pilihan Anda untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik evaluasi yang dipilih.

> **Note**
>
> Penyetelan model otomatis hanya CatBoost tersedia dari Amazon SageMaker SDK, bukan dari konsol. SageMaker

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik evaluasi dihitung oleh algoritme CatBoost

SageMaker CatBoost Algoritma menghitung metrik berikut untuk digunakan untuk validasi model. Metrik evaluasi secara otomatis ditetapkan berdasarkan jenis tugas klasifikasi, yang ditentukan oleh jumlah bilangan bulat unik di kolom label.

Nama Metrik	Deskripsi	Arah Optimasi	Pola Regex
`RMSE`	kesalahan kuadrat rata-rata akar	memperkecil	`"bestTest = ([0-9\\.]+)"`
`MAE`	berarti kesalahan absolut	memperkecil	`"bestTest = ([0-9\\.]+)"`
`MedianAbsoluteError`	kesalahan absolut median	memperkecil	`"bestTest = ([0-9\\.]+)"`
`R2`	skor r2	memaksimalkan	`"bestTest = ([0-9\\.]+)"`

Nama Metrik	Deskripsi	Arah Optimasi	Pola Regex
`Logloss`	entropi silang biner	memaksimalkan	`"bestTest = ([0-9\\.]+)"`
`Precision`	ketepatan	memaksimalkan	`"bestTest = ([0-9\\.]+)"`
`Recall`	penarikan	memaksimalkan	`"bestTest = ([0-9\\.]+)"`
`F1`	skor f1	memaksimalkan	`"bestTest = ([0-9\\.]+)"`
`AUC`	skor auc	memaksimalkan	`"bestTest = ([0-9\\.]+)"`
`MultiClass`	entropi silang multiclass	memaksimalkan	`"bestTest = ([0-9\\.]+)"`
`Accuracy`	ketepatan	memaksimalkan	`"bestTest = ([0-9\\.]+)"`
`BalancedAccuracy`	akurasi seimbang	memaksimalkan	`"bestTest = ([0-9\\.]+)"`

Hiperparameter yang dapat disetel CatBoost

Setel CatBoost model dengan hyperparameters berikut. Hiperparameter yang memiliki efek terbesar dalam mengoptimalkan metrik CatBoost evaluasi adalah:`learning_rate`,,`depth`,

`l2_leaf_reg` dan. `random_strength` Untuk daftar semua CatBoost hyperparameters, lihatCatBoost hiperparameter.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`learning_rate`	ContinuousParameterRanges	MinValue: 0,001, MaxValue: 0,01
`depth`	IntegerParameterRanges	MinValue: 4, MaxValue: 10
`l2_leaf_reg`	IntegerParameterRanges	MinValue: 2, MaxValue: 10
`random_strength`	ContinuousParameterRanges	MinValue: 0, MaxValue: 10

Algoritme faktorisasi

Algoritma Factorization Machines adalah algoritma pembelajaran yang diawasi untuk tujuan umum yang dapat Anda gunakan untuk tugas klasifikasi dan regresi. Ini adalah perpanjangan dari model linier yang dirancang untuk menangkap interaksi antara fitur dalam dataset jarang dimensi tinggi secara ekonomi. Misalnya, dalam sistem prediksi klik, model Mesin Faktorisasi dapat menangkap pola rasio klik yang diamati saat iklan dari kategori iklan tertentu ditempatkan pada halaman dari kategori halaman tertentu. Mesin faktorisasi adalah pilihan yang baik untuk tugas-tugas yang berurusan dengan dataset jarang dimensi tinggi, seperti prediksi klik dan rekomendasi item.

> **Note**
>
> Amazon SageMaker implementasi algoritma Factorization Machines hanya mempertimbangkan interaksi pasangan-bijaksana (urutan ke-2) antar fitur.

Topik

- Antarmuka Input/Output untuk Algoritma Mesin Faktorisasi
- Rekomendasi Instans EC2 untuk Algoritma Mesin Faktorisasi

- Faktorisasi Mesin Contoh Notebook
- Bagaimana Mesin Faktorisasi Bekerja
- Hyperparameter Mesin Faktorisasi
- Tune Model Mesin Faktorisasi
- Faktorisasi Mesin Respon Format

Antarmuka Input/Output untuk Algoritma Mesin Faktorisasi

Algoritma Factorization Machines dapat dijalankan baik dalam mode klasifikasi biner atau mode regresi. Dalam setiap mode, dataset dapat diberikan keujisaluran bersama dengan set data saluran kereta api. Penilaian tergantung pada mode yang digunakan. Dalam mode regresi, dataset pengujian dinilai menggunakan Root Mean Square Error (RMSE). Dalam mode klasifikasi biner, kumpulan data pengujian dinilai menggunakan Binary Cross Entropy (Log Loss), Akurasi (pada ambang = 0,5) dan Skor F1 (pada ambang = 0,5).

Untukpelatihan, algoritma Factorization Machines saat ini hanya mendukung`recordIO-protobuf`format dengan`Float32`tensor. Karena kasus penggunaannya sebagian besar pada data yang jarang,`CSV`Bukan kandidat yang baik. Pelatihan mode File dan Pipa didukung untuk protobuf yang dibungkus Recordio.

Untukinferensi, algoritma Factorization Machines mendukung`application/json`dan`x-recordio-protobuf`format.

- UntukKlasifikasi binermasalah, algoritma memprediksi skor dan label. Labelnya adalah angka dan bisa berupa`0`atau`1`. Skor adalah angka yang menunjukkan seberapa kuat algoritma percaya bahwa label seharusnya`1`. Algoritma menghitung skor pertama dan kemudian memperoleh label dari nilai skor. Jika skor lebih besar dari atau sama dengan 0,5, labelnya adalah`1`.
- Untukregresimasalah, hanya skor dikembalikan dan itu adalah nilai prediksi. Misalnya, jika Mesin Faktorisasi digunakan untuk memprediksi peringkat film, skor adalah nilai peringkat yang diprediksi.

Silakan lihatFaktorisasi Mesin Contoh Notebookuntuk detail lebih lanjut tentang format file pelatihan dan inferensi.

Rekomendasi Instans EC2 untuk Algoritma Mesin Faktorisasi

Amazon SageMaker Algoritma Factorization Machines sangat skalabel dan dapat melatih seluruh instance terdistribusi. Kami merekomendasikan pelatihan dan inferensi dengan instans CPU untuk

kumpulan data yang jarang dan padat. Dalam beberapa keadaan, pelatihan dengan satu atau lebih GPU pada data padat mungkin memberikan beberapa manfaat. Pelatihan dengan GPU hanya tersedia pada data padat. Gunakan instance CPU untuk data yang jarang. Algoritma Factorization Machines mendukung instans P2, P3, G4dn, dan G5 untuk pelatihan dan inferensi.

Faktorisasi Mesin Contoh Notebook

Untuk contoh notebook yang menggunakan SageMaker Algoritma Mesin Faktorisasi untuk menganalisis gambar digit tulisan tangan dari nol hingga sembilan dalam set data MNIST, lihatPengantar Mesin Faktorisasi dengan MNIST. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, LihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMaker Contohtab untuk melihat daftar semua SageMaker Sampel. Contoh notebook yang menggunakan algoritma Factorization Machines terletak diPengantar algoritme AmazonBagian. Untuk membuka notebook, klik pada nyaGunakantab dan pilihBuat salinan.

Bagaimana Mesin Faktorisasi Bekerja

Tugas prediksi untuk model Factorization Machines adalah untuk memperkirakan fungsi dari set fitur $x_i$ke domain target. Domain ini bernilai nyata untuk regresi dan biner untuk klasifikasi. Model Mesin Faktorisasi diawasi dan memiliki set data pelatihan (x_i, y_j) tersedia. Keuntungan yang disajikan model ini terletak pada cara menggunakan parametrization faktorisasi untuk menangkap interaksi fitur berpasangan. Hal ini dapat direpresentasikan secara matematis sebagai berikut:

$$\hat{y} = w_0 + \sum_i w_i x_i + \sum_i \sum_{j>i} < v_i, v_j > x_i x_j$$

Tiga istilah dalam persamaan ini sesuai masing-masing dengan tiga komponen model:

- W_0istilah mewakili bias global.
- $W_i$istilah linier memodelkan kekuatan i^{th}Variabel.
- <v_i, v_j> istilah faktorisasi memodelkan interaksi berpasangan antara i^{th}dan J^{th}Variabel.

Bias global dan istilah linier sama seperti pada model linier. Interaksi fitur berpasangan dimodelkan dalam istilah ketiga sebagai produk dalam dari faktor-faktor terkait yang dipelajari untuk setiap fitur. Faktor yang dipelajari juga dapat dianggap sebagai vektor embedding untuk setiap fitur. Misalnya, dalam tugas klasifikasi, jika sepasang fitur cenderung terjadi bersamaan lebih sering pada sampel berlabel positif, maka produk dalam dari faktor-faktor mereka akan menjadi besar. Dengan kata lain,

vektor embedding mereka akan dekat satu sama lain dalam kesamaan kosinus. Untuk informasi selengkapnya tentang model Mesin Faktorisasi, lihatMesin faktorisasi.

Untuk tugas regresi, model dilatih dengan meminimalkan kesalahan kuadrat antara prediksi $model_n$dan nilai target y_n. Hal ini dikenal sebagai kerugian kuadrat:

$$L = \frac{1}{N}\sum_{n}(y_n - \hat{y}_n)^2$$

Untuk tugas klasifikasi, model dilatih dengan meminimalkan kerugian entropi silang, juga dikenal sebagai kehilangan log:

$$L = \frac{1}{N}\sum_{n}[y_n \log \hat{p}_n + (1 - y_n) \log (1 - \hat{p}_n)]$$

di mana:

$$\hat{p}_n = \frac{1}{1 + e^{-\hat{y}_n}}$$

Untuk informasi selengkapnya tentang fungsi kehilangan klasifikasi, lihatFungsi kerugian untuk klasifikasi.

Hyperparameter Mesin Faktorisasi

Tabel berikut berisi hyperparameters untuk algoritma Factorization Machines. Ini adalah parameter yang ditetapkan oleh pengguna untuk memfasilitasi estimasi parameter model dari data. Hyperparameter yang diperlukan yang harus ditetapkan dicantumkan terlebih dahulu, dalam urutan abjad. Hyperparameter opsional yang dapat diatur tercantum berikutnya, juga dalam urutan abjad.

Nama Parameter	Deskripsi
`feature_dim`	Dimensi ruang fitur input. Ini bisa sangat tinggi dengan masukan yang jarang. Wajib Nilai yang valid: Integer positif. Rentang nilai yang disarankan: [10000,10000000]
`num_factors`	Dimensi faktorisasi. Wajib

Nama Parameter	Deskripsi
	Nilai yang valid: Integer positif. Rentang nilai yang disarankan: [2,1000], 64 biasanya menghasilkan hasil yang baik dan merupakan titik awal yang baik.
`predictor_type`	Jenis prediktor. • `binary_classifier` : Untuk tugas klasifikasi biner. • `regressor` : Untuk tugas regresi. Wajib Nilai yang valid: String:`binary_classifier` atau`regressor`
`bias_init_method`	Metode inisialisasi untuk istilah bias: • `normal`: Menginisialisasi bobot dengan nilai acak yang diambil sampel dari distribusi normal dengan rata-rata nol dan standar deviasi yang ditentukan oleh`bias_init_sigma` . • `uniform`: Menginisialisasi bobot dengan nilai acak yang disampel secara seragam dari rentang yang ditentukan oleh [-`bias_init_scale` , +`bias_init_scale`]. • `constant`: Menginisialisasi bobot ke nilai skalar yang ditentukan oleh`bias_init_value` . Opsional Nilai valid: `uniform`, `normal`, atau `constant` Nilai default: `normal`

Nama Parameter	Deskripsi
`bias_init_scale`	Rentang untuk inisialisasi istilah bias. Berlaku jika`bias_init_method` set ke`uniform`. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: Tidak ada
`bias_init_sigma`	Standar deviasi untuk inisialisasi istilah bias. Berlaku jika`bias_init_method` set ke`normal`. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0,01
`bias_init_value`	Nilai awal dari istilah bias. Berlaku jika`bias_init_method` set ke`constant`. Opsional Nilai yang valid: FLOAT. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: Tidak ada
`bias_lr`	Tingkat pembelajaran untuk istilah bias. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0,1

Nama Parameter	Deskripsi
`bias_wd`	Peluruhan berat untuk istilah bias. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0,01
`clip_gradient`	Parameter pengoptimal kliping gradien. Klip gradien dengan memproyeksikan ke interval [`-clip_gradient` , `+clip_gradient ]`. Opsional Nilai yang valid: Desimal Nilai default: Tidak ada
`epochs`	Jumlah zaman pelatihan untuk dijalankan. Opsional Nilai yang valid: Integer positif Nilai default: 1
`eps`	Parameter Epsilon untuk menghindari pembagian dengan 0. Opsional Nilai yang valid: FLOAT. Nilai yang disarankan: kecil. Nilai default: Tidak ada

Nama Parameter	Deskripsi
`factors_init_method`	Metode inisialisasi untuk istilah faktorisasi: • `normal`Menginisialisasi bobot dengan nilai acak yang diambil sampel dari distribusi normal dengan rata-rata nol dan deviasi standar yang ditentukan oleh`factors_init_sigma` . • `uniform`: Menginisialisasi bobot dengan nilai acak yang disampel secara seragam dari rentang yang ditentukan oleh [-`factors_init_scale` , +`factors_init_scale`]. • `constant`: Menginisialisasi bobot ke nilai skalar yang ditentukan oleh`factors_init_value` . Opsional Nilai yang benar:`uniform`,`normal`, atau`constant`. Nilai default: `normal`
`factors_init_scale`	Kisaran untuk inisialisasi istilah faktorisasi. Berlaku jika`factors_init_method` set ke`uniform`. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: Tidak ada
`factors_init_sigma`	Standar deviasi untuk inisialisasi istilah faktorisasi. Berlaku jika`factors_init_method` set ke`normal`. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0.001

Nama Parameter	Deskripsi
`factors_init_value`	Nilai awal istilah faktorisasi. Berlaku jika`factors_init_method` set ke`constant`. Opsional Nilai yang valid: FLOAT. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: Tidak ada
`factors_lr`	Tingkat pembelajaran untuk istilah faktorisasi. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0.0001
`factors_wd`	Peluruhan berat untuk istilah faktorisasi. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0.00001
`linear_lr`	Tingkat pembelajaran untuk istilah linier. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0.001

Nama Parameter	Deskripsi
`linear_init_method`	Metode inisialisasi untuk istilah linier: • `normal`Menginisialisasi bobot dengan nilai acak yang diambil sampel dari distribusi normal dengan rata-rata nol dan deviasi standar yang ditentukan oleh`linear_init_sigma` . • `uniform`Menginisialisasi bobot dengan nilai acak yang disampel secara seragam dari rentang yang ditentukan oleh [-`linear_init_scale` , +`linear_init_scale`]. • `constant`Menginisialisasi bobot ke nilai skalar yang ditentuka n oleh`linear_init_value` . Opsional Nilai yang benar:`uniform`,`normal`, atau`constant`. Nilai default: `normal`
`linear_init_scale`	Rentang untuk inisialisasi istilah linier. Berlaku jika`linear_in it_method` set ke`uniform`. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: Tidak ada
`linear_init_sigma`	Standar deviasi untuk inisialisasi istilah linear. Berlaku jika`linear_init_method` set ke`normal`. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0,01

Nama Parameter	Deskripsi
`linear_init_value`	Nilai awal dari istilah linier. Berlaku jika`linear_init_method` set ketetap. Opsional Nilai yang valid: FLOAT. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: Tidak ada
`linear_wd`	Peluruhan berat untuk istilah linier. Opsional Nilai yang valid: Mengapung non-negatif. Kisaran nilai yang disarankan: [1e-8, 512]. Nilai default: 0.001
`mini_batch_size`	Ukuran mini-batch yang digunakan untuk pelatihan. Opsional Nilai yang valid: Integer positif Nilai default: 1000
`rescale_grad`	Parameter pengoptimal penskalaan ulang gradien. Jika diatur, mengalikan gradien dengan`rescale_grad` sebelum memperbarui. Sering memilih untuk menjadi 1.0/`batch_size` . Opsional Nilai yang valid: Desimal Nilai default: Tidak ada

Tune Model Mesin Faktorisasi

Penyempurnaan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma Mesin Faktorisasi

Algoritma Mesin Faktorisasi memiliki klasifikasi biner dan jenis prediktor regresi. Jenis prediktor menentukan metrik mana yang dapat Anda gunakan untuk penyetelan model otomatis. Algoritma melaporkan`test:rmse`regressor metrik, yang dihitung selama pelatihan. Saat menyetel model untuk tugas regresi, pilih metrik ini sebagai tujuannya.

Nama Metrik	Deskripsi	Arah optimalisasi
`test:rmse`	Akar Berarti Persegi Kesalahan	Minimalkan

Algoritma Factorization Machines melaporkan tiga metrik klasifikasi biner, yang dihitung selama pelatihan. Saat menyetel model untuk tugas klasifikasi biner, pilih salah satunya sebagai tujuannya.

Nama Metrik	Deskripsi	Arah optimalisasi
`test:binary_classification_accuracy`	Akurasi	Maksimalkan
`test:binary_classification_cross_entropy`	Lintas Entropi	Minimalkan

Nama Metrik	Deskripsi	Arah optimalisasi
`test:binary_f_beta`	Beta	Maksimalkan

Hyperparameter Mesin Faktorisasi Merdu

Anda dapat menyetel hyperparameters berikut untuk algoritma Factorization Machines. Parameter inisialisasi yang berisi istilah bias, linier, dan faktorisasi bergantung pada metode inisialisasi mereka. Ada tiga metode inisialisasi:`uniform`,`normal`, dan`constant`. Metode inisialisasi ini tidak dapat dirdu sendiri. Parameter yang merdu tergantung pada pilihan metode inisialisasi ini. Misalnya, jika metode inisialisasi`uniform`, maka hanya`scale`parameter yang merdu. Secara khusus, jika`bias_init_method==uniform`, maka`bias_init_scale`,`linear_init_scale`, dan`factors_init_scale`merdu. Demikian pula, jika metode inisialisasi`normal`maka hanya`sigma`parameter yang merdu. Jika metode inisialisasi`constant`maka hanya`value`parameter yang merdu. Dependensi ini tercantum dalam tabel berikut.

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan	Dependensi
`bias_init_scale`	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_metode==seragam
`bias_init_sigma`	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_method = normal
`bias_init_value`	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_method==konstan
`bias_lr`	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	Tidak ada
`bias_wd`	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	Tidak ada

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan	Dependensi
epoch	IntegerParameterRange	MinValue: 1, MaxValue: 1000	Tidak ada
factors_init_scale	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_metode==seragam
factors_init_sigma	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_metode==normal
factors_init_value	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_method==konstan
factors_lr	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	Tidak ada
factors_wd	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512]	Tidak ada
linear_init_scale	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_metode==seragam
linear_init_sigma	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_metode==normal
linear_init_value	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	bias_init_method==konstan
linear_lr	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	Tidak ada

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan	Dependensi
`linear_wd`	ContinuousParameterRange	MinValue: 1e-8 MaxValue: 512	Tidak ada
`mini_batch_size`	IntegerParameterRange	MinValue: 100, MaxValue: 10000	Tidak ada

Faktorisasi Mesin Respon Format

Format Respons JSON

Klasifikasi biner

```
let response =   {
    "predictions":    [
        {
            "score": 0.4,
            "predicted_label": 0
        }
    ]
}
```

Regresi

```
let response =   {
    "predictions":    [
        {
            "score": 0.4
        }
    ]
}
```

Format Respons JSONLINES

Klasifikasi biner

```
{"score": 0.4, "predicted_label": 0}
```

Regresi

```
{"score": 0.4}
```

Format Respons RECORDIO

Klasifikasi biner

```
[
    Record = {
        features = {},
        label = {
            'score’: {
                keys: [],
                values: [0.4]  # float32
            },
            'predicted_label': {
                keys: [],
                values: [0.0]  # float32
            }
        }
    }
]
```

Regresi

```
[
    Record = {
        features = {},
        label = {
            'score’: {
                keys: [],
                values: [0.4]  # float32
            }
        }
    }
]
```

Algoritme k-Nearest Neighbor (k-NN)

Amazon SageMaker tetangga k-terdekat (k-nN) algoritma adalah algoritma berbasis indeks. Ini menggunakan metode non-parametrik untuk klasifikasi atau regresi. Untuk masalah klasifikasi,

algoritma querykpoin yang paling dekat dengan titik sampel dan mengembalikan label kelas yang paling sering digunakan sebagai label yang diprediksi. Untuk masalah regresi, algoritme menanyakanktitik terdekat ke titik sampel dan mengembalikan rata-rata nilai fitur mereka sebagai nilai prediksi.

Pelatihan dengan algoritma k-nN memiliki tiga langkah: pengambilan sampel, pengurangan dimensi, dan pembangunan indeks. Pengambilan sampel mengurangi ukuran dataset awal sehingga cocok dengan memori. Untuk pengurangan dimensi, algoritma mengurangi dimensi fitur data untuk mengurangi jejak model k-nN dalam memori dan latensi inferensi. Kami menyediakan dua metode metode pengurangan dimensi: proyeksi acak dan transformasi Johnson-Lindenstrauss yang cepat. Biasanya, Anda menggunakan pengurangan dimensi untuk dataset dimensi tinggi (d> 1000) untuk menghindari “kutukan dimensi” yang mengganggu analisis statistik data yang menjadi jarang saat dimensionalitas meningkat. Tujuan utama pelatihan K-nN adalah untuk membangun indeks. Indeks memungkinkan pencarian jarak yang efisien antara titik yang nilai atau label kelasnya belum ditentukan dan titik k terdekat untuk digunakan untuk inferensi.

Topik

- Antarmuka Input/Output untuk Algoritma K-nN
- Notebook k-NN Contoh
- Bagaimana Algoritma K-nN Bekerja
- Rekomendasi Instans EC2 untuk Algoritma K-nN
- hiperparameter k-NN
- Menyetel k-NN
- Format Data untuk Input Pelatihan K-nN
- Format Permintaan dan Respons K-nN

Antarmuka Input/Output untuk Algoritma K-nN

SageMaker K-nN mendukung saluran data kereta dan uji.

- MenggunakanSaluran keretauntuk data yang ingin Anda sampel dan membangun ke dalam indeks k-NN.
- MenggunakanSaluran ujiuntuk memancarkan skor dalam file log. Skor terdaftar sebagai satu baris per batch mini: akurasi untuk`classifier`, kesalahan kuadrat (mse) untuk`regressor`untuk skor.

Untuk input pelatihan, dukungan K-nN`text/csv`dan`application/x-recordio-protobuf`Format data Untuk tipe input`text/csv`, yang pertama`label_size`kolom ditafsirkan sebagai vektor label untuk baris itu. Anda dapat menggunakan mode File atau mode Pipa untuk melatih model pada data yang diformat sebagai`recordIO-wrapped-protobuf`atau sebagai`CSV`.

Untuk input inferensi, K-nN mendukung`application/json`,`application/x-recordio-protobuf`, dan`text/csv`Format data Klaster`text/csv`format menerima`label_size`dan parameter pengkodean. Ini mengasumsikan`label_size`0 dan pengkodean UTF-8.

Untuk output inferensi, K-nN mendukung`application/json`dan`application/x-recordio-protobuf`Format data Kedua format data ini juga mendukung mode keluaran verbose. Dalam mode keluaran verbose, API memberikan hasil pencarian dengan vektor jarak yang diurutkan dari terkecil ke terbesar, dan elemen yang sesuai dalam vektor label.

Untuk transformasi batch, K-nN mendukung`application/jsonlines`format data untuk input dan output. Contoh input adalah sebagai berikut:

```
content-type: application/jsonlines

{"features": [1.5, 16.0, 14.0, 23.0]}
{"data": {"features": {"values": [1.5, 16.0, 14.0, 23.0]}}
```

Sebuah output contoh adalah sebagai berikut:

```
accept: application/jsonlines

{"predicted_label": 0.0}
{"predicted_label": 2.0}
```

Untuk informasi lebih lanjut tentang format file input dan output, lihatFormat Data untuk Input Pelatihan K-nNuntuk pelatihanFormat Permintaan dan Respons K-nNuntuk kesimpulan danNotebook k-NN Contoh.

Notebook k-NN Contoh

Untuk contoh notebook yang menggunakan SageMaker algoritma tetangga k-terdekat untuk memprediksi jenis tutupan hutan belantara dari data layanan geologi dan kehutanan, lihatTipe Penutup Tetangga K-Terdekat.

Gunakan instance notebook Jupyter untuk menjalankan contoh SageMaker. Untuk mempelajari cara membuat dan membuka instance notebook Jupyter di SageMaker, lihatInstans SageMaker Notebook

Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMaker Contohtab untuk melihat daftar semua SageMaker Notebook contoh Temukan notebook K-Neighbor terdekat diPengantar algoritme AmazonBagian. Untuk membuka notebook, klik pada nyaGunakantab dan pilihBuat salinan.

Bagaimana Algoritma K-nN Bekerja

Langkah 1: Sampel

Untuk menentukan jumlah total titik data yang akan diambil sampel dari set data pelatihan, gunakan`sample_size`parameter Misalnya, jika dataset awal memiliki 1.000 titik data dan`sample_size`diatur ke 100, di mana jumlah total instance adalah 2, setiap pekerja akan mengambil sampel 50 poin. Satu set total 100 titik data akan dikumpulkan. Sampling berjalan dalam waktu linier sehubungan dengan jumlah titik data.

Langkah 2: Lakukan Pengurangan Dimensi

Implementasi algoritma k-NN saat ini memiliki dua metode pengurangan dimensi. Anda menentukan metode di`dimension_reduction_type`hiperparameter Klaster`sign`menentukan proyeksi acak, yang menggunakan proyeksi linear menggunakan matriks tanda-tanda acak, dan`fjlt`metode menentukan cepat Johnson-Lindenstrauss transformasi, metode yang didasarkan pada transformasi Fourier. Kedua metode menjaga jarak L2 dan produk dalam. Klaster`fjlt`Metode harus digunakan ketika dimensi target besar dan memiliki kinerja yang lebih baik dengan inferensi CPU. Metode berbeda dalam kompleksitas komputasi mereka. Klaster`sign`Metode membutuhkan O (ndk) waktu untuk mengurangi dimensi batch n titik dimensi d menjadi target dimensi k. Klaster`fjlt`Metode membutuhkan O (nd log (d)) waktu, tetapi konstanta yang terlibat lebih besar. Menggunakan pengurangan dimensi memperkenalkan kebisingan ke dalam data dan kebisingan ini dapat mengurangi akurasi prediksi.

Langkah 3: Bangun Indeks

Selama inferensi, algoritma query indeks untuk k-nearest-neighbors dari titik sampel. Berdasarkan referensi ke poin, algoritma membuat klasifikasi atau prediksi regresi. Itu membuat prediksi berdasarkan label kelas atau nilai yang disediakan. k-nn menyediakan tiga jenis indeks: indeks datar, indeks terbalik, dan indeks terbalik dengan kuantisasi produk. Anda menentukan jenis dengan`index_type`parameter

Serialisasi Model

Ketika algoritma k-nN selesai pelatihan, itu serializes tiga file untuk mempersiapkan inferensi.

- model_algo-1: Berisi indeks serial untuk menghitung tetangga terdekat.
- model_algo-1.labels: Berisi label serial (format biner np.float32) untuk menghitung label yang diprediksi berdasarkan hasil kueri dari indeks.
- model_algo-1.json: Berisi metadata model berformat JSON yang menyimpankdan`predictor_type`hiper-parameter dari pelatihan untuk inferensi bersama dengan negara lain yang relevan.

Dengan implementasi K-nn saat ini, Anda dapat memodifikasi file metadata untuk mengubah cara prediksi dihitung. Misalnya, Anda dapat mengubahkke 10 atau ubah`predictor_type`kepadaregresi.

```
{
  "k": 5,
  "predictor_type": "classifier",
  "dimension_reduction": {"type": "sign", "seed": 3, "target_dim": 10, "input_dim":
 20},
  "normalize": False,
  "version": "1.0"
}
```

Rekomendasi Instans EC2 untuk Algoritma K-nN

Kami merekomendasikan pelatihan pada instance CPU (seperti ml.m5.2xlarge) atau pada instans GPU. Algoritma K-nn mendukung keluarga instans GPU P2, P3, G4dn, dan G5 untuk pelatihan dan inferensi.

Permintaan inferensi dari CPU umumnya memiliki latensi rata-rata yang lebih rendah daripada permintaan dari GPU karena ada pajak atas komunikasi CPU-ke-GPU saat Anda menggunakan perangkat keras GPU. Namun, GPU umumnya memiliki throughput yang lebih tinggi untuk batch yang lebih besar.

hiperparameter k-NN

Nama Parameter	Deskripsi
`feature_dim`	Jumlah fitur dalam data input. Wajib

Nama Parameter	Deskripsi
	Nilai yang valid: integer positif.
k	Jumlah tetangga terdekat. Wajib Nilai yang valid: bilangan bulat positif
`predictor_type`	Jenis inferensi untuk digunakan pada label data. Wajib Nilai yang benar:Pengklasifikasiuntuk klasifikasi atauregresiuntuk regresi
`sample_size`	Jumlah titik data yang akan diambil sampel dari kumpulan data pelatihan. Wajib Nilai yang valid: bilangan bulat positif
`dimension_reduction_target`	Dimensi target untuk mengurangi ke. Wajibsaat menentukan`dimension_reduction_type` parameter Nilai yang valid: integer positif lebih besar dari 0 dan kurang dari`feature_dim` .
`dimension_reduction_type`	Jenis metode reduksi dimensi. Opsional Nilai yang benar:tandauntuk proyeksi acak ataufjltuntuk transformasi Johnson-Lindenstrauss yang cepat. Nilai default: Tidak ada reduksi dimensi

Nama Parameter	Deskripsi
`faiss_index_ivf_nlists`	Jumlah centroid yang akan dibangun dalam indeks kapan`index_type` adalahFaiss.ivfflatatauFaiss.ivfpq. Opsional Nilai yang valid: bilangan bulat positif Nilai default:auto, yang menyelesaikan`sqrt(sample_size)` .
`faiss_index_pq_m`	Jumlah vektor sub-komponen untuk membangun dalam indeks ketika`index_type` diaturFaiss.ivfpq. Klaster FaceBook AI Similarity Search (FAISS) perpustakaan mengharuskan nilai`faiss_index_pq_m` adalah pembagidimensi data. Jika`faiss_index_pq_m` bukan pembagi dimensi data, kami meningkatkan dimensi data ke bilangan bulat terkecil yang dapat dibagi`faiss_index_pq_m` . Jika tidak ada pengurangan dimensi diterapkan, algoritma menambahkan padding nol. Jika pengurang an dimensi diterapkan, algoritma meningkatkan nilai`dimension _reduction_target` hiper-parameter. Opsional Nilai yang valid: Salah satu bilangan bulat positif berikut: 1, 2, 3, 8, 12, 16, 20, 24, 28, 32, 40, 48, 56, 64, 96
`index_metric`	Metrik untuk mengukur jarak antar titik saat menemukan tetangga terdekat. Saat berlatih dengan`index_type` set`faiss.IVFPQ` , yang`INNER_PRODUCT` jarak`COSINE`kesamaan tidak didukung. Opsional Nilai yang valid: L2untuk Euclidean-jarak,INNER_PRODUCTuntuk jarak produk dalam,KOSINUSuntuk kesamaan kosinus. Nilai default: L2

Nama Parameter	Deskripsi
`index_type`	Jenis indeks. Opsional Nilai yang benar:Faiss.datar,Faiss.ivfflat,Faiss.ivfpq. Nilai default:Faiss.datar
`mini_batch_size`	Jumlah pengamatan per mini-batch untuk iterator data. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5000

Menyetel k-NN

Amazon SageMaker k-terdekat algoritma tetangga adalah algoritma diawasi. Algoritma mengkonsumsi kumpulan data uji dan memancarkan metrik tentang akurasi untuk tugas klasifikasi atau tentang kesalahan kuadrat rata-rata untuk tugas regresi. Metrik akurasi ini membandingkan prediksi model untuk tugas masing-masing dengan kebenaran dasar yang disediakan oleh data uji empiris. Untuk menemukan model terbaik yang melaporkan akurasi tertinggi atau kesalahan terendah pada set data pengujian, jalankan pekerjaan penyetelan hyperparameter untuk k-NN.

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif yang sesuai untuk tugas prediksi algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif. Hyperparameter hanya digunakan untuk membantu memperkirakan parameter model dan tidak digunakan oleh model terlatih untuk membuat prediksi.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma K-nN

Algoritma tetangga k-terdekat menghitung salah satu dari dua metrik dalam tabel berikut selama pelatihan tergantung pada jenis tugas yang ditentukan olehpredictor_typehiper-parameter.

- Pengklasifikasimenentukan tugas klasifikasi dan menghitungtest:accuracy
- regresimenentukan tugas regresi dan menghitungtest:mse.

Pilihpredictor_typenilai yang sesuai untuk jenis tugas yang dilakukan untuk menghitung metrik objektif yang relevan saat menyetel model.

Nama Metrik	Deskripsi	Arah optimalisasi
test:accuracy	Saatpredictor_type diaturPengklasifikasi, K-nn membandingkan label yang diprediksi, berdasarkan rata-rata label tetangga k-terdekat, dengan label kebenaran tanah yang disediakan dalam data saluran uji. Akurasi yang dilaporkan berkisar dari 0,0 (0%) hingga 1,0 (100%).	Maksimalkan
test:mse	Saatpredictor_type diaturregresi, K-nn membandingkan label yang diprediksi, berdasarkan rata-rata label tetangga k-terdekat, dengan label kebenaran tanah yang disediaka n dalam data saluran uji. Kesalahan kuadrat rata-rata dihitung dengan membandingkan dua label.	Minimalkan

Hiperparameter k-NN yang bisa disetel

Menyetel Amazon SageMaker k-model tetangga terdekat dengan hyperparameter berikut.

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan
k	IntegerParameterRanges	MinValue: 1, MaxValue: 1024
`sample_size`	IntegerParameterRanges	MinValue: 256, MaxValue: 20000000

Format Data untuk Input Pelatihan K-nN

Semua Amazon SageMaker algoritma bawaan mematuhi format pelatihan input umum yang dijelaskan dalamFormat Data Umum - Pelatihan. Topik ini berisi daftar format input yang tersedia untuk SageMaker k-nearest-neighbor Algoritme

Format Data CSV

konten-jenis: teks/csv; label_size = 1

```
4,1.2,1.3,9.6,20.3
```

Yang Pertama`label_size`kolom ditafsirkan sebagai vektor label untuk baris itu.

Format Data RECORDIO

konten-jenis: aplikasi/x-recordio-protobuf

```
[
    Record = {
        features = {
            'values': {
                values: [1.2, 1.3, 9.6, 20.3]  # float32
            }
        },
        label = {
            'values': {
                values: [4]  # float32
            }
        }
    }
]
```

```
}
```

Format Permintaan dan Respons K-nN

Semua Amazon SageMaker algoritma bawaan mematuhi format inferensi input umum yang dijelaskan dalamFormat Data Umum - Inferensi. Topik ini berisi daftar format output yang tersedia untuk SageMaker k-nearest-neighbor Algoritme

INPUT: Format Permintaan

konten-jenis: teks/csv

```
1.2,1.3,9.6,20.3
```

Ini menerimalabel_sizeatau parameter pengkodean. Ini mengasumsikanlabel_sizedari 0 dan pengkodean utf-8.

INPUT: Format Permintaan

konten-jenis: aplikasi/json

```
{
  "instances": [
    {"data": {"features": {"values": [-3, -1, -4, 2]}}},
    {"features": [3.0, 0.1, 0.04, 0.002]}]
}
```

INPUT: Format Permintaan JSONLINES

konten-jenis: aplikasi/jsonlines

```
{"features": [1.5, 16.0, 14.0, 23.0]}
{"data": {"features": {"values": [1.5, 16.0, 14.0, 23.0]}}
```

INPUT: Format Permintaan RECORDIO

konten-jenis: aplikasi/x-recordio-protobuf

```
[
    Record = {
```

```
        features = {
            'values': {
                values: [-3, -1, -4, 2]  # float32
            }
        },
        label = {}
    },
    Record = {
        features = {
            'values': {
                values: [3.0, 0.1, 0.04, 0.002]  # float32
            }
        },
        label = {}
    },
]
```

OUTPUT: Format Respons JSON

menerima: aplikasi/json

```
{
  "predictions": [
    {"predicted_label": 0.0},
    {"predicted_label": 2.0}
  ]
}
```

OUTPUT: Format Respons JSONLINES

menerima: aplikasi/jsonlines

```
{"predicted_label": 0.0}
{"predicted_label": 2.0}
```

OUTPUT: Format Respons JSON VERBOSE

Dalam mode verbose, API memberikan hasil pencarian dengan vektor jarak yang diurutkan dari terkecil ke terbesar, dengan elemen yang sesuai dalam vektor label. Dalam contoh ini, k diatur ke 3.

terima: application/json; verbose = true

```
{
```

```
  "predictions": [
    {
        "predicted_label": 0.0,
        "distances": [3.11792408, 3.89746071, 6.32548437],
        "labels": [0.0, 1.0, 0.0]
    },
    {
        "predicted_label": 2.0,
        "distances": [1.08470316, 3.04917915, 5.25393973],
        "labels": [2.0, 2.0, 0.0]
    }
  ]
}
```

OUTPUT: Format Respons RECORDIO-PROTOBUF

konten-jenis: aplikasi/x-recordio-protobuf

```
[
    Record = {
        features = {},
        label = {
            'predicted_label': {
                values: [0.0]  # float32
            }
        }
    },
    Record = {
        features = {},
        label = {
            'predicted_label': {
                values: [2.0]  # float32
            }
        }
    }
]
```

OUTPUT: Format Respons RECORDIO-PROTOBUF VERBOSE

Dalam mode verbose, API memberikan hasil pencarian dengan vektor jarak yang diurutkan dari terkecil ke terbesar, dengan elemen yang sesuai dalam vektor label. Dalam contoh ini, k diatur ke 3.

menerima: aplikasi/x-recordio-protobuf; verbose = true

```
[
    Record = {
        features = {},
        label = {
            'predicted_label': {
                values: [0.0]  # float32
            },
            'distances': {
                values: [3.11792408, 3.89746071, 6.32548437]  # float32
            },
            'labels': {
                values: [0.0, 1.0, 0.0]  # float32
            }
        }
    },
    Record = {
        features = {},
        label = {
            'predicted_label': {
                values: [0.0]  # float32
            },
            'distances': {
                values: [1.08470316, 3.04917915, 5.25393973]  # float32
            },
            'labels': {
                values: [2.0, 2.0, 0.0]  # float32
            }
        }
    }
]
```

OUTPUT SAMPEL untuk Algoritma K-nN

Untuk tugas regresor:

```
[06/08/2018 20:15:33 INFO 140026520049408] #test_score (algo-1) : ('mse',
 0.013333333333333334)
```

Untuk tugas classifier:

```
[06/08/2018 20:15:46 INFO 140285487171328] #test_score (algo-1) : ('accuracy',
 0.98666666666666669)
```

LightGBM

LightGBM adalah implementasi open-source yang populer dan efisien dari algoritma Gradient Boosting Decision Tree (GBDT). GBDT adalah algoritma pembelajaran yang diawasi yang mencoba memprediksi variabel target secara akurat dengan menggabungkan ansambel perkiraan dari serangkaian model yang lebih sederhana dan lebih lemah. LightGBM menggunakan teknik tambahan untuk secara signifikan meningkatkan efisiensi dan skalabilitas GBDT konvensional.

Cara menggunakan SageMaker LightGBM

Anda dapat menggunakan LightGBM sebagai algoritma SageMaker bawaan Amazon. Bagian berikut menjelaskan cara menggunakan LightGBM dengan Python SageMaker SDK. Untuk informasi tentang cara menggunakan LightGBM dari Amazon SageMaker Studio Classic UI, lihat. SageMaker JumpStart

- Gunakan LightGBM sebagai algoritma bawaan

  Gunakan algoritma built-in LightGBM untuk membangun wadah pelatihan LightGBM seperti yang ditunjukkan pada contoh kode berikut. Anda dapat secara otomatis melihat URI image algoritma bawaan LightGBM menggunakan SageMaker `image_uris.retrieve` API (atau `get_image_uri` API jika menggunakan Amazon SageMaker Python SDK versi 2).

  Setelah menentukan URI image LightGBM, Anda dapat menggunakan container LightGBM untuk membuat estimator menggunakan Estimator API dan memulai tugas pelatihan SageMaker . Algoritma bawaan LightGBM berjalan dalam mode skrip, tetapi skrip pelatihan disediakan untuk Anda dan tidak perlu menggantinya. Jika Anda memiliki pengalaman luas menggunakan mode skrip untuk membuat pekerjaan SageMaker pelatihan, maka Anda dapat memasukkan skrip pelatihan LightGBM Anda sendiri.

```
from sagemaker import image_uris, model_uris, script_uris

train_model_id, train_model_version, train_scope = "lightgbm-classification-model",
 "*", "training"
training_instance_type = "ml.m5.xlarge"

# Retrieve the docker image
train_image_uri = image_uris.retrieve(
    region=None,
    framework=None,
    model_id=train_model_id,
    model_version=train_model_version,
```

```
    image_scope=train_scope,
    instance_type=training_instance_type
)

# Retrieve the training script
train_source_uri = script_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 script_scope=train_scope
)

train_model_uri = model_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 model_scope=train_scope
)

# Sample training data is available in this bucket
training_data_bucket = f"jumpstart-cache-prod-{aws_region}"
training_data_prefix = "training-datasets/tabular_multiclass/"

training_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
train"
validation_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
validation"

output_bucket = sess.default_bucket()
output_prefix = "jumpstart-example-tabular-training"

s3_output_location = f"s3://{output_bucket}/{output_prefix}/output"

from sagemaker import hyperparameters

# Retrieve the default hyperparameters for training the model
hyperparameters = hyperparameters.retrieve_default(
    model_id=train_model_id, model_version=train_model_version
)

# [Optional] Override default hyperparameters with custom values
hyperparameters[
    "num_boost_round"
] = "500"
print(hyperparameters)

from sagemaker.estimator import Estimator
from sagemaker.utils import name_from_base
```

```
training_job_name = name_from_base(f"built-in-algo-{train_model_id}-training")

# Create SageMaker Estimator instance
tabular_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1, # for distributed training, specify an instance_count greater
 than 1
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location
)

# Launch a SageMaker Training job by passing the S3 path of the training data
tabular_estimator.fit(
    {
        "train": training_dataset_s3_path,
        "validation": validation_dataset_s3_path,
    }, logs=True, job_name=training_job_name
)
```

Untuk informasi selengkapnya tentang cara mengatur LightGBM sebagai algoritma bawaan, lihat contoh notebook berikut.

- Klasifikasi tabel dengan Amazon SageMaker LightGBM dan algoritma CatBoost
- Regresi tabular dengan Amazon SageMaker LightGBM dan algoritma CatBoost

Antarmuka Input dan Output untuk algoritma LightGBM

Peningkatan gradien beroperasi pada data tabular, dengan baris mewakili pengamatan, satu kolom mewakili variabel target atau label, dan kolom yang tersisa mewakili fitur.

SageMaker Implementasi LightGBM mendukung CSV untuk pelatihan dan inferensi:

- Untuk Pelatihan ContentType, input yang valid harus teks/csv.
- Untuk Inferensi ContentType, input yang valid harus teks/csv.

> **Note**
>
> Untuk pelatihan CSV, algoritme mengasumsikan bahwa variabel target ada di kolom pertama dan CSV tidak memiliki catatan header.
> Untuk inferensi CSV, algoritme mengasumsikan bahwa input CSV tidak memiliki kolom label.

Format input untuk data pelatihan, data validasi, dan fitur kategoris

Perhatikan cara memformat data pelatihan Anda untuk masukan ke model LightGBM. Anda harus menyediakan jalur ke bucket Amazon S3 yang berisi data pelatihan dan validasi Anda. Anda juga dapat menyertakan daftar fitur kategoris. Gunakan saluran `train` dan `validation` saluran untuk memberikan data masukan Anda. Atau, Anda hanya dapat menggunakan `train` saluran.

> **Note**
>
> Keduanya `train` dan `training` merupakan nama saluran yang valid untuk pelatihan LightGBM.

Gunakan kedua **`validation`** saluran **`train`** dan

Anda dapat memberikan data input Anda melalui dua jalur S3, satu untuk `train` saluran dan satu untuk `validation` saluran. Setiap jalur S3 dapat berupa awalan S3 yang menunjuk ke satu atau lebih file CSV atau jalur S3 lengkap yang menunjuk ke satu file CSV tertentu. Variabel target harus berada di kolom pertama file CSV Anda. Variabel prediktor (fitur) harus berada di kolom yang tersisa. Jika beberapa file CSV disediakan untuk `validation` saluran `train` atau, algoritma LightGBM menggabungkan file. Data validasi digunakan untuk menghitung skor validasi di akhir setiap iterasi peningkatan. Penghentian awal diterapkan ketika skor validasi berhenti membaik.

Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan file atau file data pelatihan Anda. Jika Anda menyediakan file JSON untuk fitur kategoris, `train` saluran Anda harus menunjuk ke awalan S3 dan bukan file CSV tertentu. File ini harus berisi kamus Python di mana kuncinya adalah string `"cat_index_list"` dan nilainya adalah daftar bilangan bulat unik. Setiap bilangan bulat dalam daftar nilai harus menunjukkan indeks kolom dari fitur kategoris yang sesuai dalam file CSV data pelatihan Anda. Setiap nilai harus berupa bilangan bulat positif (lebih besar dari nol karena nol mewakili nilai target), kurang dari `Int32.MaxValue` (2147483647), dan kurang dari jumlah kolom. Seharusnya hanya ada satu file JSON indeks kategoris.

Gunakan hanya **`train`** saluran:

Sebagai alternatif, Anda dapat memberikan data input Anda melalui jalur S3 tunggal untuk `train` saluran tersebut. Jalur S3 ini harus menunjuk ke direktori dengan subdirektori bernama `train/` yang berisi satu atau lebih file CSV. Anda dapat secara opsional menyertakan subdirektori lain di lokasi yang sama yang disebut `validation/` yang juga memiliki satu atau lebih file CSV. Jika data validasi tidak disediakan, maka 20% data pelatihan Anda diambil sampelnya secara acak untuk dijadikan data validasi. Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan subdirektori data Anda.

> **Note**
>
> Untuk mode input pelatihan CSV, total memori yang tersedia untuk algoritme (jumlah instance dikalikan dengan memori yang tersedia di`InstanceType`) harus dapat menampung kumpulan data pelatihan.

SageMaker LightGBM menggunakan modul Python Joblib untuk membuat serial atau deserialisasi model, yang dapat digunakan untuk menyimpan atau memuat model.

Untuk menggunakan model yang dilatih dengan SageMaker LightGBM dengan modul JobLib

- Gunakan kode Python berikut:

```
import joblib
import tarfile

t = tarfile.open('model.tar.gz', 'r:gz')
t.extractall()

model = joblib.load(model_file_path)

# prediction with test data
# dtest should be a pandas DataFrame with column names feature_0, feature_1, ...,
 feature_d
pred = model.predict(dtest)
```

Rekomendasi instans Amazon EC2 untuk algoritme LightGBM

SageMaker LightGBM saat ini mendukung pelatihan CPU single-instance dan multi-instance. Untuk pelatihan CPU multi-instance (pelatihan terdistribusi), tentukan `instance_count` lebih besar dari 1 saat Anda menentukan Estimator Anda. Untuk informasi selengkapnya tentang pelatihan terdistribusi dengan LightGBM, lihat Amazon SageMaker LightGBM Distributed training menggunakan Dask.

LightGBM adalah algoritma yang terikat memori (sebagai lawan dari compute-bound). Jadi, instance komputasi tujuan umum (misalnya, M5) adalah pilihan yang lebih baik daripada instance yang dioptimalkan komputasi (misalnya, C5). Selanjutnya, kami menyarankan Anda memiliki memori total yang cukup dalam instance yang dipilih untuk menyimpan data pelatihan.

Notebook sampel LightGBM

Tabel berikut menguraikan berbagai contoh notebook yang membahas kasus penggunaan yang berbeda dari algoritma Amazon SageMaker LightGBM.

Judul Notebook	Deskripsi
Klasifikasi tabel dengan Amazon SageMaker LightGBM dan algoritma CatBoost	Notebook ini menunjukkan penggunaan algoritma Amazon SageMaker LightGBM untuk melatih dan menjadi tuan rumah model klasifikasi tabel.
Regresi tabular dengan Amazon SageMaker LightGBM dan algoritma CatBoost	Notebook ini menunjukkan penggunaan algoritma Amazon SageMaker LightGBM untuk melatih dan menjadi tuan rumah model regresi tabular.
Amazon SageMaker LightGBM Pelatihan terdistribusi menggunakan Dask	Notebook ini menunjukkan pelatihan terdistribusi dengan algoritma Amazon SageMaker LightGBM menggunakan kerangka Dask.

Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Bagaimana LightGBM bekerja

LightGBM mengimplementasikan algoritma Gradient Boosting Decision Tree (GBDT) konvensional dengan penambahan dua teknik baru: Gradient Based One-Side Sampling (GOSS) dan Exclusive Feature Bundling (EFB). Teknik-teknik ini dirancang untuk secara signifikan meningkatkan efisiensi dan skalabilitas GBDT.

Algoritma LightGBM berkinerja baik dalam kompetisi pembelajaran mesin karena penanganannya yang kuat terhadap berbagai tipe data, hubungan, distribusi, dan keragaman hyperparameter yang dapat Anda sesuaikan. Anda dapat menggunakan LightGBM untuk masalah regresi, klasifikasi (biner dan multiclass), dan peringkat.

Untuk informasi lebih lanjut tentang peningkatan gradien, lihat. Bagaimana XGBoost Bekerja Untuk detail mendalam tentang teknik GOSS dan EFB tambahan yang digunakan dalam metode LightGBM, lihat LightGBM: Pohon Keputusan Peningkatan Gradien yang Sangat Efisien.

Hiperparameter LightGBM

Tabel berikut berisi subset hiperparameter yang diperlukan atau paling umum digunakan untuk algoritma Amazon SageMaker LightGBM. Pengguna mengatur parameter ini untuk memfasilitasi estimasi parameter model dari data. Algoritma SageMaker LightGBM adalah implementasi dari paket LightGBM open-source.

> **Note**
>
> Hyperparameter default didasarkan pada contoh kumpulan data di file. Notebook sampel LightGBM

Secara default, algoritma SageMaker LightGBM secara otomatis memilih metrik evaluasi dan fungsi objektif berdasarkan jenis masalah klasifikasi. Algoritma LightGBM mendeteksi jenis masalah klasifikasi berdasarkan jumlah label dalam data Anda. Untuk masalah regresi, metrik evaluasi adalah kesalahan kuadrat rata-rata akar dan fungsi tujuannya adalah kerugian L2. Untuk masalah klasifikasi biner, metrik evaluasi dan fungsi objektif keduanya adalah entropi silang biner. Untuk masalah klasifikasi multikelas, metrik evaluasi adalah entropi silang multikelas dan fungsi tujuannya adalah softmax. Anda dapat menggunakan `metric` hyperparameter untuk mengubah metrik evaluasi default. Lihat tabel berikut untuk informasi selengkapnya tentang hyperparameters LightGBM, termasuk deskripsi, nilai valid, dan nilai default.

Nama Parameter	Deskripsi
num_boost_round	Jumlah maksimum peningkatan iterasi. Catatan: Secara internal, LightGBM membangun num_class * num_boost_round pohon untuk masalah klasifikasi multi-kelas. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat positif. Nilai default:100.
early_stopping_rounds	Pelatihan akan berhenti jika satu metrik dari satu titik data validasi tidak membaik di early_stopping_rounds babak terakhir. Jika early_stopping_rounds kurang dari atau sama dengan nol, hyperparameter ini diabaikan. Nilai yang valid: bilangan bulat. Nilai default:10.
metric	Metrik evaluasi untuk data validasi. Jika metric diatur ke "auto" nilai default, maka algoritma secara otomatis memilih metrik evaluasi berdasarkan jenis masalah klasifikasi: • rmseuntuk regresi • binary_logloss untuk klasifikasi biner • multi_logloss untuk klasifikasi multi-kelas Nilai yang valid: string, salah satu dari berikut ini: ("auto""rmse""l1","l2","huber","fair","binary_logloss" ,"binary_error" ,"auc","average_precision" ,"multi_logloss" ,"multi_error" ,"auc_mu", atau"cross_entropy"). Nilai default:"auto".
learning_rate	Tingkat di mana bobot model diperbarui setelah mengerjakan setiap batch contoh pelatihan. Nilai yang valid: float, range: (0.0,1.0).

Nama Parameter	Deskripsi
	Nilai default:`0.1`.
`num_leaves`	Jumlah maksimum daun dalam satu pohon. Nilai yang valid: integer, range: (`1`,`131072`). Nilai default:`64`.
`feature_fraction`	Subset fitur yang akan dipilih pada setiap iterasi (pohon). Harus kurang dari 1,0. Nilai yang valid: float, range: (`0.0`,`1.0`). Nilai default:`0.9`.
`bagging_fraction`	Subset fitur yang mirip dengan`feature_fraction` , tetapi `bagging_fraction` secara acak memilih bagian dari data tanpa resampling. Nilai yang valid: float, range: (`0.0`,`1.0`]. Nilai default:`0.9`.
`bagging_freq`	Frekuensi untuk melakukan bagging. Pada setiap `bagging_freq` iterasi, LightGBM secara acak memilih persentase data yang akan digunakan untuk iterasi berikutnya. `bagging_freq` Persentase ini ditentukan oleh `bagging_fraction` hyperparameter. Jika `bagging_freq` nol, maka bagging dinonaktifkan. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat non-negatif. Nilai default:`1`.

Nama Parameter	Deskripsi
`max_depth`	Kedalaman maksimum untuk model pohon. Ini digunakan untuk menangani overfitting ketika jumlah data kecil. Jika `max_depth` kurang dari atau sama dengan nol, ini berarti tidak ada batasan untuk kedalaman maksimum. Nilai yang valid: bilangan bulat. Nilai default:`6`.
`min_data_in_leaf`	Jumlah minimum data dalam satu daun. Dapat digunakan untuk menangani overfitting. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat non-negatif. Nilai default:`3`.
`max_delta_step`	Digunakan untuk membatasi output maksimal daun pohon. Jika `max_delta_step` kurang dari atau sama dengan 0, maka tidak ada kendala. Output maksimal akhir daun adalah`learning_rate * max_delta_step` . Nilai yang valid: float. Nilai default:`0.0`.
`lambda_l1`	Regularisasi L1. Nilai yang valid: float, range: Float non-negatif. Nilai default:`0.0`.
`lambda_l2`	Regularisasi L2. Nilai yang valid: float, range: Float non-negatif. Nilai default:`0.0`.

Nama Parameter	Deskripsi
`boosting`	Jenis penguat Nilai yang valid: string, salah satu dari berikut ini: (`"gbdt"`,`"rf"`,`"dart"`, atau`"goss"`). Nilai default:`"gbdt"`.
`min_gain_to_split`	Keuntungan minimum untuk melakukan split. Dapat digunakan untuk mempercepat pelatihan. Nilai yang valid: integer, float: Float non-negatif. Nilai default:`0.0`.
`scale_pos_weight`	Berat label dengan kelas positif. Digunakan hanya untuk tugas klasifikasi biner. `scale_pos_weight` tidak dapat digunakan jika `is_unbalance` disetel ke`"True"`. Nilai yang valid: float, range: Positive float. Nilai default:`1.0`.
`tree_learner`	Jenis pembelajar pohon. Nilai yang valid: string, salah satu dari berikut ini: (`"serial"`,`"feature"` ,`"data"`, atau`"voting"`). Nilai default:`"serial"`.
`feature_fraction_bynode`	Memilih subset fitur acak pada setiap node pohon. Misalnya, jika `feature_fraction_bynode` ya`0.8`, maka 80% fitur dipilih. Dapat digunakan untuk menangani overfitting. Nilai yang valid: integer, range: (`0.0`,`1.0`]. Nilai default:`1.0`.

Nama Parameter	Deskripsi
`is_unbalance`	Setel ke `"True"` jika data pelatihan tidak seimbang. Digunakan hanya untuk tugas klasifikasi biner. `is_unbalance` tidak dapat digunakan dengan`scale_pos_weight` . Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.
`max_bin`	Jumlah maksimum nampan yang digunakan untuk memasukkan nilai fitur. Sejumlah kecil tempat sampah dapat mengurangi akurasi pelatihan, tetapi dapat meningkatkan kinerja umum. Dapat digunakan untuk menangani overfitting. Nilai yang valid: bilangan bulat, rentang: (1, ∞). Nilai default:`255`.
`tweedie_variance_power`	Mengontrol varians distribusi Tweedie. Atur ini lebih dekat `2.0` untuk bergeser ke arah distribusi gamma. Atur ini lebih dekat `1.0` untuk beralih ke distribusi Poisson. Digunakan hanya untuk tugas regresi. Nilai yang valid: float, range: [`1.0`,`2.0`). Nilai default:`1.5`.
`num_threads`	Jumlah thread paralel yang digunakan untuk menjalankan LightGBM. Nilai 0 berarti jumlah default thread di OpenMP. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat non-negatif. Nilai default:`0`.

Nama Parameter	Deskripsi
`verbosity`	Verbositas pesan cetak. Jika `verbosity` kurang dari`0`, maka pesan cetak hanya menunjukkan kesalahan fatal. Jika `verbosity` diatur ke`0`, maka pesan cetak termasuk kesalahan dan peringatan. Jika `verbosity` ya`1`, maka cetak pesan menampilkan informasi lebih lanjut. `verbosity` Lebih besar dari 1 menunjukkan sebagian besar informasi dalam pesan cetak dan dapat digunakan untuk debugging. Nilai yang valid: bilangan bulat. Nilai default:`1`.

Menyetel model LightGBM

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hiperparameter pada kumpulan data pelatihan dan validasi Anda. Penyetelan model berfokus pada hiperparameter berikut:

> **Note**
>
> Fungsi tujuan pembelajaran secara otomatis ditetapkan berdasarkan jenis tugas klasifikasi, yang ditentukan oleh jumlah bilangan bulat unik di kolom label. Untuk informasi selengkapnya, lihat Hiperparameter LightGBM.

- Fungsi tujuan pembelajaran untuk mengoptimalkan selama pelatihan model
- Metrik evaluasi yang digunakan untuk mengevaluasi kinerja model selama validasi
- Satu set hyperparameters dan rentang nilai untuk masing-masing untuk digunakan saat menyetel model secara otomatis

Penyetelan model otomatis mencari hiperparameter yang Anda tentukan untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik evaluasi yang dipilih.

> **Note**
>
> Penyetelan model otomatis untuk LightGBM hanya tersedia dari Amazon SageMaker SDK, bukan dari konsol. SageMaker

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik evaluasi dihitung oleh algoritma LightGBM

Algoritma SageMaker LightGBM menghitung metrik berikut untuk digunakan untuk validasi model. Metrik evaluasi secara otomatis ditetapkan berdasarkan jenis tugas klasifikasi, yang ditentukan oleh jumlah bilangan bulat unik di kolom label.

Nama Metrik	Deskripsi	Arah Optimasi	Pola Regex
`rmse`	kesalahan kuadrat rata-rata akar	memperkecil	`"rmse: ([0-9\\.]+)"`
`l1`	berarti kesalahan absolut	memperkecil	`"l1: ([0-9\\.]+)"`
`l2`	berarti kesalahan kuadrat	memperkecil	`"l2: ([0-9\\.]+)"`
`huber`	kerugian huber	memperkecil	`"huber: ([0-9\\.]+)"`
`fair`	kerugian yang adil	memperkecil	`"fair: ([0-9\\.]+)"`
`binary_logloss`	entropi silang biner	memaksimalkan	`"binary_logloss: ([0-9\\.]+)"`

Nama Metrik	Deskripsi	Arah Optimasi	Pola Regex
`binary_error`	kesalahan biner	memperkecil	`"binary_error: ([0-9\\.]+)"`
`auc`	AUC	memaksimalkan	`"auc: ([0-9\\.]+)"`
`average_precision`	skor presisi rata-rata	memaksimalkan	`"average_precision: ([0-9\\.]+)"`
`multi_logloss`	entropi silang multiclass	memaksimalkan	`"multi_logloss: ([0-9\\.]+)"`
`multi_error`	skor kesalahan multiclass	memperkecil	`"multi_error: ([0-9\\.]+)"`
`auc_mu`	AUC-mu	memaksimalkan	`"auc_mu: ([0-9\\.]+)"`
`cross_entropy`	entropi silang	memperkecil	`"cross_entropy: ([0-9\\.]+)"`

Hiperparameter LightGBM yang dapat disetel

Setel model LightGBM dengan hyperparameter berikut. Hiperparameter yang memiliki efek terbesar dalam mengoptimalkan metrik evaluasi LightGBM adalah:`learning_rate`,,,,,`num_leaves`,

`feature_fraction` dan. `bagging_fraction` `bagging_freq` `max_depth` `min_data_in_leaf` Untuk daftar semua hyperparameters LightGBM, lihat. Hiperparameter LightGBM

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`learning_rate`	ContinuousParameterRanges	MinValue: 0,001, MaxValue: 0,01
`num_leaves`	IntegerParameterRanges	MinValue: 10, MaxValue: 100
`feature_fraction`	ContinuousParameterRanges	MinValue: 0.1, MaxValue: 1.0
`bagging_fraction`	ContinuousParameterRanges	MinValue: 0.1, MaxValue: 1.0
`bagging_freq`	IntegerParameterRanges	MinValue: 0, MaxValue: 10
`max_depth`	IntegerParameterRanges	MinValue: 15, MaxValue: 100
`min_data_in_leaf`	IntegerParameterRanges	MinValue: 10, MaxValue: 200

Algoritma Linear

Model linieradalah algoritma pembelajaran yang diawasi yang digunakan untuk memecahkan masalah klasifikasi atau regresi. Untuk input, Anda memberikan contoh berlabel model (x,y).xadalah vektor berdimensi tinggi danyadalah label numerik. Untuk masalah klasifikasi biner, label harus 0 atau 1. Untuk masalah klasifikasi multiclass, label harus dari 0 hingganum_classes- 1. Untuk masalah regresi,yadalah bilangan real. Algoritma mempelajari fungsi linier, atau, untuk masalah klasifikasi, fungsi ambang linier, dan memetakan vektorxuntuk perkiraan labely.

Amazon SageMaker algoritma pembelajar linier memberikan solusi untuk masalah klasifikasi dan regresi. Dengan SageMaker algoritma, Anda dapat secara bersamaan mengeksplorasi tujuan pelatihan yang berbeda dan memilih solusi terbaik dari set validasi. Anda juga dapat menjelajahi

sejumlah besar model dan memilih yang terbaik. Model terbaik mengoptimalkan salah satu dari berikut:

- Tujuan berkelanjutan, seperti kesalahan kuadrat rata-rata, kehilangan entropi silang, kesalahan absolut.
- Tujuan diskrit cocok untuk klasifikasi, seperti ukuran F1, presisi, penarikan, atau akurasi.

Dibandingkan dengan metode yang memberikan solusi hanya untuk tujuan berkelanjutan, SageMakeralgoritma pembelajar linier memberikan peningkatan kecepatan yang signifikan dibandingkan teknik optimasi hyperparameter naif. Hal ini juga lebih nyaman.

Algoritma pembelajar linier membutuhkan matriks data, dengan baris yang mewakili pengamatan, dan kolom yang mewakili dimensi fitur. Ini juga membutuhkan kolom tambahan yang berisi label yang cocok dengan titik data. Minimal, Amazon SageMaker pelajar linier mengharuskan Anda untuk menentukan lokasi data input dan output, dan tipe tujuan (klasifikasi atau regresi) sebagai argumen. Dimensi fitur juga diperlukan. Untuk informasi selengkapnya, lihat `CreateTrainingJob`. Anda dapat menentukan parameter tambahan dalam`HyperParameters`peta string dari badan permintaan. Parameter ini mengontrol proses pengoptimalan, atau spesifikasi fungsi tujuan yang Anda latih. Misalnya, jumlah zaman, regularisasi, dan jenis kerugian.

Jika Anda menggunakanPelatihan Spot yang Dikelola, algoritma pembelajar linier mendukung penggunaanpos pemeriksaan untuk mengambil snapshot dari keadaan model.

Topik

- Antarmuka Input/Output untuk algoritma pembelajar linier
- Rekomendasi instans EC2 untuk algoritma pembelajar linier
- Buku catatan sampel pelajar linier
- Cara kerja linear learner
- Hiperparameter pelajar linier
- Menyetel model pembelajar linier
- Format respons pelajar linier

Antarmuka Input/Output untuk algoritma pembelajar linier

Amazon SageMaker algoritma pembelajar linier mendukung tiga saluran data: melatih, validasi (opsional), dan tes (opsional). Jika Anda memberikan data

validasi,`S3DataDistributionType`seharusnya`FullyReplicated`. Algoritma mencatat kehilangan validasi di setiap zaman, dan menggunakan sampel data validasi untuk mengkalibrasi dan memilih model terbaik. Jika Anda tidak memberikan data validasi, algoritme menggunakan sampel data pelatihan untuk mengkalibrasi dan memilih model. Jika Anda memberikan data pengujian, log algoritme menyertakan skor tes untuk model akhir.

Untuk pelatihan, algoritma pembelajar linier mendukung keduanya`recordIO-wrapped protobuf`dan`CSV`format. Untuk`application/x-recordio-protobuf`tipe input, hanya tensor Float32 yang didukung. Untuk`text/csv`tipe input, kolom pertama diasumsikan sebagai label, yang merupakan variabel target untuk prediksi. Anda dapat menggunakan mode File atau mode Pipa untuk melatih model pelajar linier pada data yang diformat sebagai`recordIO-wrapped-protobuf`atau sebagai`CSV`.

Untuk inferensi, algoritma pembelajar linier mendukung`application/json`,`application/x-recordio-protobuf`, dan`text/csv`format. Ketika Anda membuat prediksi pada data baru, format respons tergantung pada jenis model. Untuk regresi(`predictor_type='regressor'`),`score`adalah prediksi yang dihasilkan oleh model. Untuk klasifikasi(`predictor_type='binary_classifier'`atau`predictor_type='multiclass_classifi` model mengembalikan`score`dan juga a`predicted_label`. The`predicted_label`adalah kelas yang diprediksi oleh model dan`score`mengukur kekuatan prediksi itu.

- Untuk klasifikasi biner,`predicted_label`adalah`0`atau`1`, dan`score`adalah nomor floating point tunggal yang menunjukkan seberapa kuat algoritma percaya bahwa label harus 1.
- Untuk klasifikasi multiclass,`predicted_class`akan menjadi integer dari`0`kepada`num_classes-1`, dan`score`akan menjadi daftar satu nomor floating point per kelas.

Untuk menafsirkan`score`dalam masalah klasifikasi, Anda harus mempertimbangkan fungsi kerugian yang digunakan. Jika`loss`nilai hyperparameter adalah`logistic`untuk klasifikasi biner atau`softmax_loss`untuk klasifikasi multiclass, maka`score`dapat diartikan sebagai probabilitas kelas yang sesuai. Ini adalah nilai kerugian yang digunakan oleh pelajar linier ketika`loss`Nilai adalah`auto`Nilai default. Tetapi jika kerugian diatur ke`hinge_loss`, maka skor tidak dapat diartikan sebagai probabilitas. Ini karena kehilangan engsel sesuai dengan Support Vector Classifier, yang tidak menghasilkan perkiraan probabilitas.

Untuk informasi selengkapnya tentang format file input dan output, lihat[Format respons pelajar linier](). Untuk informasi lebih lanjut tentang format inferensi, dan[Buku catatan sampel pelajar linier]().

Rekomendasi instans EC2 untuk algoritma pembelajar linier

Algoritma pembelajar linier mendukung instance CPU dan GPU untuk pelatihan dan inferensi. Untuk GPU, algoritma pembelajar linier mendukung keluarga GPU P2, P3, G4dn, dan G5.

Selama pengujian, kami belum menemukan bukti substansional bahwa instans multi-GPU lebih cepat daripada instans GPU tunggal. Hasil dapat bervariasi, tergantung pada kasus penggunaan spesifik Anda.

Buku catatan sampel pelajar linier

Tabel berikut menguraikan berbagai contoh notebook yang membahas berbagai kasus penggunaan Amazon SageMaker algoritma pembelajar linier.

Judul Notebook	Deskripsi
Pengantar dengan dataset MNIST	Menggunakan dataset MNIST, kami melatih pengklasifikasi biner untuk memprediksi satu digit.
Bagaimana Cara Membangun Pengklasifikasi Multiclass?	Menggunakan kumpulan data Covertype UCI, kami mendemonstrasikan cara melatih pengklasifikasi multiclass.
Bagaimana Membangun Pipa Machine Learning (M) untuk Inferensi?	Menggunakan wadah Scikit-learn, kami mendemonstrasikan cara membuat end-to-end Pipa ML.

Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMakerContohtab untuk melihat daftar semua SageMaker sampel. Contoh buku catatan pemodelan topik menggunakan algoritma pembelajaran linier terletak diPengantar algoritma Amazonbagian. Untuk membuka buku catatan, pilihGunakantab dan pilihBuat salinan.

Cara kerja linear learner

Ada tiga langkah yang terlibat dalam implementasi algoritma pembelajar linier: preprocess, train, dan validate.

Langkah 1: Preprocess

Normalisasi, atau penskalaan fitur, adalah langkah pra-pemrosesan penting untuk fungsi kerugian tertentu yang memastikan model yang dilatih pada kumpulan data tidak didominasi oleh bobot satu fitur. Amazon SageMaker Algoritma Linear Learner memiliki opsi normalisasi untuk membantu langkah preprocessing ini. Jika normalisasi dihidupkan, algoritme pertama-tama membahas sampel kecil data untuk mempelajari nilai rata-rata dan standar deviasi untuk setiap fitur dan untuk label. Setiap fitur dalam kumpulan data lengkap kemudian digeser untuk memiliki rata-rata nol dan diskalakan untuk memiliki standar deviasi unit.

> **Note**
>
> Untuk hasil terbaik, pastikan data Anda dikocokkan sebelum pelatihan. Pelatihan dengan data yang tidak diacak dapat menyebabkan pelatihan gagal.

Anda dapat mengonfigurasi apakah algoritme pembelajar linier menormalkan data fitur dan label menggunakan`normalize_data`dan`normalize_label`hiperparameter, masing-masing. Normalisasi diaktifkan secara default untuk fitur dan label untuk regresi. Hanya fitur yang dapat dinormalisasi untuk klasifikasi biner dan ini adalah perilaku default.

Langkah 2: Melatih

Dengan algoritma pembelajar linier, Anda berlatih dengan implementasi terdistribusi dari penurunan gradien stokastik (SGD). Anda dapat mengontrol proses pengoptimalan dengan memilih algoritma pengoptimalan. Misalnya, Anda dapat memilih untuk menggunakan Adam, AdaGrad, penurunan gradien stokastik, atau algoritma optimasi lainnya. Anda juga menentukan hiperparameternya, seperti momentum, tingkat pembelajaran, dan jadwal tingkat pembelajaran. Jika Anda tidak yakin algoritma atau nilai hyperparameter mana yang akan digunakan, pilih default yang berfungsi untuk sebagian besar kumpulan data.

Selama pelatihan, Anda secara bersamaan mengoptimalkan beberapa model, masing-masing dengan tujuan yang sedikit berbeda. Misalnya, Anda memvariasikan regularisasi L1 atau L2 dan mencoba pengaturan pengoptimal yang berbeda.

Langkah 3: Validasi dan atur ambang

Saat melatih beberapa model secara paralel, model dievaluasi terhadap set validasi untuk memilih model yang paling optimal setelah pelatihan selesai. Untuk regresi, model yang paling optimal adalah

model yang mencapai kerugian terbaik pada set validasi. Untuk klasifikasi, sampel set validasi digunakan untuk mengkalibrasi ambang klasifikasi. Model paling optimal yang dipilih adalah model yang mencapai kriteria pemilihan klasifikasi biner terbaik pada set validasi. Contoh kriteria tersebut termasuk ukuran F1, akurasi, dan kehilangan entropi silang.

> **Note**
>
> Jika algoritma tidak disediakan set validasi, maka mengevaluasi dan memilih model yang paling optimal tidak mungkin. Untuk memanfaatkan pelatihan paralel dan pemilihan model, pastikan Anda memberikan set validasi ke algoritme.

Hiperparameter pelajar linier

Tabel berikut berisi kode hiperparameter untuk algoritma pembelajar. Ini adalah parameter yang ditetapkan oleh pengguna untuk memfasilitasi estimasi parameter model dari data. Hyperparameter yang diperlukan yang harus ditetapkan terdaftar terlebih dahulu, dalam urutan abjad. Hyperparameter opsional yang dapat diatur tercantum berikutnya, juga dalam urutan abjad. Ketika hyperparameter disetel ke`auto`, Amazon SageMaker akan secara otomatis menghitung dan menetapkan nilai hyperparameter itu.

Nama Parameter	Deskripsi
`num_classes`	Jumlah kelas untuk variabel respons. Algoritma mengasumsikan bahwa kelas diberi label`0`, ... ,`num_classes - 1` . Diperlukanketika`predictor_type` adalah`multiclass_classifier` . Jika tidak, algoritme mengabaikannya. Nilai yang valid: Bilangan bulat dari 3 hingga 1.000.000
`predictor_type`	Menentukan jenis variabel target sebagai klasifikasi biner, klasifikasi multiclass, atau regresi. Diperlukan Nilai valid: `binary_classifier` , `multiclass_classifier` , atau `regressor`

Nama Parameter	Deskripsi
`accuracy_top_k`	Saat menghitung metrik akurasi top-k untuk klasifikasi multiclas s, nilaik. Jika model menetapkan salah satu skor top-k ke label sebenarnya, sebuah contoh dinilai sebagai benar. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 3
`balance_m ulticlass _weights`	Menentukan apakah akan menggunakan bobot kelas, yang memberikan masing-masing kelas sama pentingnya dalam fungsi kerugian. Digunakan hanya ketika`predictor _type` adalah`multiclass_classifier` . Opsional Nilai valid: `true`, `false` Nilai default: `false`
`beta_1`	Tingkat peluruhan eksponensial untuk perkiraan momen pertama. Berlaku hanya ketika`optimizer` Nilai adalah`adam`. Opsional Nilai valid:`auto`atau nilai floating-point antara 0 dan 1,0 Nilai default: `auto`
`beta_2`	Tingkat peluruhan eksponensial untuk perkiraan momen kedua. Berlaku hanya ketika`optimizer` Nilai adalah`adam`. Opsional Nilai valid:`auto`atau bilangan bulat mengambang antara 0 dan 1,0 Nilai default: `auto`

Nama Parameter	Deskripsi
`bias_lr_mult`	Memungkinkan tingkat pembelajaran yang berbeda untuk istilah bias. Tingkat pembelajaran aktual untuk bias adalah`learning_rate *bias_lr_mult` . Opsional Nilai valid:`auto`atau bilangan bulat floating-point positif Nilai default: `auto`
`bias_wd_mult`	Memungkinkan regularisasi yang berbeda untuk istilah bias. Bobot regularisasi L2 aktual untuk bias adalah`wd*bias_wd_mult` . Secara default, tidak ada regularisasi pada istilah bias. Opsional Nilai valid:`auto`atau bilangan bulat floating-point non-negatif Nilai default: `auto`

Nama Parameter	Deskripsi
`binary_classifier_model_selection_criteria`	Kapan`predictor_type` diatur ke`binary_classifier` , kriteria evaluasi model untuk kumpulan data validasi (atau untuk kumpulan data pelatihan jika Anda tidak memberikan kumpulan data validasi). Kriteria meliputi: • `accuracy`—Model dengan akurasi tertinggi. • `f_beta`—Model dengan skor F1 tertinggi. default adalah F1. • `precision_at_target_recall` —Model dengan presisi tertinggi pada target penarikan yang diberikan. • `recall_at_target_precision` —Model dengan recall tertinggi pada target presisi tertentu. • `loss_function` —Model dengan nilai terendah dari fungsi kerugian yang digunakan dalam pelatihan. Opsional Nilai valid:`accuracy`,`f_beta`,`precision_at_target_recall` ,`recall_at_target_precision` , atau`loss_function` Nilai default: `accuracy`

Nama Parameter	Deskripsi
`early_stopping_patience`	Jika tidak ada perbaikan yang dilakukan dalam metrik yang relevan, jumlah zaman yang harus menunggu sebelum mengakhiri pelatihan. Jika Anda telah memberikan nilai untuk`binary_classifier_model_selection_criteria`. metrik adalah nilai itu. Jika tidak, metrik sama dengan nilai yang ditentukan untuk`loss`hiperparameter. Metrik dievaluasi pada data validasi. Jika Anda belum memberikan data validasi, metrik selalu sama dengan nilai yang ditentukan untuk`loss`hyperparameter dan dievaluasi pada data pelatihan. Untuk menonaktifkan penghentian awal, atur`early_stopping_patience`ke nilai yang lebih besar dari nilai yang ditentukan untuk`epochs`. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 3
`early_stopping_tolerance`	Toleransi relatif untuk mengukur peningkatan kerugian. Jika rasio peningkatan kerugian dibagi dengan kerugian terbaik sebelumnya lebih kecil dari nilai ini, penghentian awal menganggap peningkatan menjadi nol. Opsional Nilai yang valid: Bilangan bulat floating-point positif Nilai default: 0,001
`epochs`	Jumlah maksimum lintasan atas data pelatihan. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 15

Nama Parameter	Deskripsi
`f_beta`	Nilai beta yang digunakan saat menghitung metrik skor F untuk klasifikasi biner atau multiclass. Juga digunakan jika nilai yang ditentukan untuk`binary_classifier_model_selection_criteria` adalah`f_beta`. Opsional Nilai yang valid: Bilangan bulat floating-point positif Nilai default: 1
`feature_dim`	Jumlah fitur dalam data input. Opsional Nilai valid:`auto`atau bilangan bulat positif Nilai default:`auto`
`huber_delta`	Parameter untuk kerugian Huber. Selama pelatihan dan evaluasi metrik, hitung kerugian L2 untuk kesalahan yang lebih kecil dari delta dan kerugian L1 untuk kesalahan yang lebih besar dari delta. Opsional Nilai yang valid: Bilangan bulat floating-point positif Nilai default: 1
`init_bias`	Bobot awal untuk istilah bias. Opsional Nilai yang valid: Integer Floating-point Nilai default: 0

Nama Parameter	Deskripsi
`init_method`	Menetapkan fungsi distribusi awal yang digunakan untuk bobot model. Fungsi meliputi: • `uniform`—Didistribusikan secara seragam antara (-skala, +skala) • `normal`Distribusi normal, dengan rata-rata 0 dan sigma Opsional Nilai yang valid: `uniform` or `normal` Nilai default: `uniform`
`init_scale`	Menimbang distribusi seragam awal untuk bobot model. Berlaku hanya ketika`init_method` hyperparameter diatur ke`uniform`. Opsional Nilai yang valid: Bilangan bulat floating-point positif Nilai default: 0,07
`init_sigma`	Standar deviasi awal untuk distribusi normal. Berlaku hanya ketika`init_method` hyperparameter diatur ke`normal`. Opsional Nilai yang valid: Bilangan bulat floating-point positif Nilai default: 0.01
`l1`	Parameter regularisasi L1. Jika Anda tidak ingin menggunakan regularisasi L1, atur nilainya ke 0. Opsional Nilai valid:`auto`atau pelampung non-negatif Nilai default: `auto`

Nama Parameter	Deskripsi
`learning_rate`	Ukuran langkah yang digunakan oleh pengoptimal untuk pembaruan parameter. Opsional Nilai valid:`auto`atau bilangan bulat floating-point positif Nilai default:`auto`, yang nilainya tergantung pada pengoptimal yang dipilih.
`loss`	Menentukan fungsi kerugian. Fungsi kerugian yang tersedia dan nilai defaultnya tergantung pada nilai`predictor_type` : • Jika`predictor_type` diatur ke`regressor` , opsi yang tersedia adalah`auto`,`squared_loss` ,`absolute_loss` ,`eps_insensitive_squared_loss` ,`eps_insensitive_absolute_loss` ,`quantile_loss` , dan`huber_loss` . Nilai default-nya `auto` is `squared_loss` . • Jika`predictor_type` diatur ke`binary_classifier` , opsi yang tersedia adalah`auto`,`logistic`, dan`hinge_loss` . Nilai default-nya `auto` is `logistic`. • Jika`predictor_type` diatur ke`multiclass_classifier` , opsi yang tersedia adalah`auto`dan`softmax_loss` . Nilai default-nya `auto` is `softmax_loss` . Nilai valid:`auto`,`logistic`,`squared_loss` ,`absolute_loss` ,`hinge_loss` ,`eps_insensitive_squared_loss` ,`eps_insensitive_absolute_loss` ,`quantile_loss` , atau`huber_loss` Opsional Nilai default: `auto`

Nama Parameter	Deskripsi
`loss_insensitivity`	Parameter untuk tipe kerugian epsilon-insensitive. Selama pelatihan dan evaluasi metrik, kesalahan yang lebih kecil dari nilai ini dianggap nol. Opsional Nilai yang valid: Bilangan bulat floating-point positif Nilai default: 0.01
`lr_scheduler_factor`	Untuk setiap`lr_scheduler_step` hyperparameter, tingkat pembelajaran menurun dengan kuantitas ini. Berlaku hanya ketika`use_lr_scheduler` hyperparameter diatur ke`true`. Opsional Nilai valid:`auto`atau bilangan bulat floating-point positif antara 0 dan 1 Nilai default: `auto`
`lr_scheduler_minimum_lr`	Tingkat pembelajaran tidak pernah menurun ke nilai yang lebih rendah dari nilai yang ditetapkan untuk`lr_scheduler_minimum_lr` . Berlaku hanya ketika`use_lr_scheduler` hyperparameter diatur ke`true`. Opsional Nilai valid:`auto`atau bilangan bulat floating-point positif Nilai default:`auto`
`lr_scheduler_step`	Jumlah langkah antara penurunan tingkat pembelajaran. Berlaku hanya ketika`use_lr_scheduler` hyperparameter diatur ke`true`. Opsional Nilai valid:`auto`atau bilangan bulat positif Nilai default: `auto`

Nama Parameter	Deskripsi
`margin`	Margin untuk`hinge_loss` fungsi. Opsional Nilai yang valid: Bilangan bulat floating-point positif Nilai default: 1
`mini_batch_size`	Jumlah pengamatan per mini-batch untuk iterator data. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 1000
`momentum`	Momentum dari`sgd`pengoptimal. Opsional Nilai valid:`auto`atau bilangan bulat mengambang antara 0 dan 1,0 Nilai default: `auto`
`normalize_data`	Menormalkan data fitur sebelum pelatihan. Normalisasi data menggeser data untuk setiap fitur untuk memiliki rata-rata nol dan menskalakannya untuk memiliki standar deviasi unit. Opsional Nilai valid: `auto`, `true`, atau `false` Nilai default: `true`

Nama Parameter	Deskripsi
`normalize_label`	Menormalkan label. Normalisasi label menggeser label untuk memiliki rata-rata nol dan menskalakannya untuk memiliki satuan standar deviasi. The`auto`nilai default menormalkan label untuk masalah regresi tetapi tidak untuk masalah klasifikasi. Jika Anda mengatur`normalize _label` hyperparameter ke`true`untuk masalah klasifikasi, algoritma mengabaikannya. Opsional Nilai valid: `auto`, `true`, atau `false` Nilai default: `auto`
`num_calib ration_samples`	Jumlah pengamatan dari dataset validasi yang akan digunakan untuk kalibrasi model (saat menemukan ambang batas terbaik). Opsional Nilai valid:`auto`atau bilangan bulat positif Nilai default: `auto`
`num_models`	Jumlah model untuk dilatih secara paralel. Untuk default,`auto`, algoritma menentukan jumlah model paralel untuk dilatih. Satu model dilatih sesuai dengan parameter pelatihan yang diberikan (regularisasi, pengoptimal, kehilangan), dan sisanya dengan parameter dekat. Opsional Nilai valid:`auto`atau bilangan bulat positif Nilai default:`auto`

Nama Parameter	Deskripsi
`num_point_for_scaler`	Jumlah titik data yang digunakan untuk menghitung normalisasi atau tidak bias istilah. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 10.000
`optimizer`	Algoritma optimasi untuk digunakan. Opsional Nilai yang valid: • `auto`Nilai default. • `sgd`—Penurunan gradien stokastik. • `adam`—Estimasi momentum adaptif. • `rmsprop`—Teknik optimasi berbasis gradien yang menggunakan rata-rata bergerak gradien kuadrat untuk menormalkan gradien. Nilai default: `auto`. Pengaturan-pengaturan default untuk`auto`adalah`adam`.
`positive_example_weight_mult`	Bobot diberikan pada contoh positif saat melatih pengklasifikasi biner. Bobot contoh negatif ditetapkan pada 1. Jika Anda ingin algoritma memilih bobot sehingga kesalahan dalam mengklasifikasikan negatifvs.contoh positif memiliki dampak yang sama pada kehilangan pelatihan, tentukan`balanced`. Jika Anda ingin algoritme memilih bobot yang mengoptimalkan kinerja, tentukan`auto`. Opsional Nilai valid:`balanced`,`auto`, atau bilangan bulat floating-point positif Nilai default: 1

Nama Parameter	Deskripsi
`quantile`	Kuantil untuk kerugian kuantil. Untuk kuantil q, model mencoba menghasilkan prediksi sehingga nilai`true_label` lebih besar dari prediksi dengan probabilitas q. Opsional Nilai yang valid: Integer Floating-point antara 0 dan 1 Nilai default: 0.5
`target_precision`	Ketepatan target. Jika`binary_classifier_model_selection_criteria` adalah`recall_at_target_precision` , maka presisi dipertahankan pada nilai ini sementara recall dimaksimalkan. Opsional Nilai yang valid: bilangan bulat mengambang antara 0 dan 1,0 Nilai default: 0.8
`target_recall`	Target recall. Jika`binary_classifier_model_selection_criteria` adalah`precision_at_target_recall` , kemudian recall diadakan pada nilai ini sementara presisi dimaksimalkan. Opsional Nilai yang valid: bilangan bulat mengambang antara 0 dan 1,0 Nilai default: 0.8

Nama Parameter	Deskripsi
`unbias_data`	Tidak bias fitur sebelum pelatihan sehingga rata-ratanya adalah 0. Secara default data tidak bias sebagai`use_bias`hyperparameter diatur ke`true`. Opsional Nilai valid: `auto`, `true`, atau `false` Nilai default: `auto`
`unbias_label`	Label tidak bias sebelum pelatihan sehingga rata-ratanya adalah 0. Berlaku untuk regresi hanya jika`use_bias`hyperparameter diatur ke`true`. Opsional Nilai valid: `auto`, `true`, atau `false` Nilai default: `auto`
`use_bias`	Menentukan apakah model harus menyertakan istilah bias, yang merupakan suku intersep dalam persamaan linier. Opsional Nilai yang valid: `true` or `false` Nilai default: `true`
`use_lr_scheduler`	Apakah akan menggunakan penjadwal untuk tingkat pembelajaran. Jika Anda ingin menggunakan penjadwal, tentukan`true`. Opsional Nilai yang valid: `true` or `false` Nilai default: `true`

Nama Parameter	Deskripsi
wd	Parameter peluruhan berat, juga dikenal sebagai parameter regularisasi L2. Jika Anda tidak ingin menggunakan regularisasi L2, atur nilainya ke 0. Opsional Nilai valid:`auto`atau bilangan bulat floating-point non-negatif Nilai default: `auto`

Menyetel model pembelajar linier

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameters yang dapat disetel, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hiperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Algoritma pembelajar linier juga memiliki mekanisme internal untuk menyetel hiperparameter yang terpisah dari fitur penyetelan model otomatis yang dijelaskan di sini. Secara default, algoritme pembelajar linier menyetel hiperparameter dengan melatih beberapa model secara paralel. Saat Anda menggunakan penyetelan model otomatis, mekanisme penyetelan internal pelajar linier dimatikan secara otomatis. Ini menetapkan jumlah model paralel,`num_models`, 1. Algoritma mengabaikan nilai apa pun yang Anda tetapkan`num_models`.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik dihitung dengan algoritma pembelajar linier

Algoritma pembelajar linier melaporkan metrik dalam tabel berikut, yang dihitung selama pelatihan. Pilih salah satunya sebagai metrik objektif. Untuk menghindari overfitting, kami sarankan untuk menyetel model terhadap metrik validasi alih-alih metrik pelatihan.

Nama Metrik	Deskripsi	Arah Optimasi
`test:absolute_loss`	Hilangnya mutlak model akhir pada dataset pengujian. Metrik objektif ini hanya berlaku untuk regresi.	Minimalkan
`test:binary_classification_accuracy`	Keakuratan model akhir pada dataset pengujian. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`test:binary_f_beta`	Skor F-beta dari model akhir pada dataset pengujian. Secara default, ini adalah skor F1, yang merupakan rata-rata harmonik presisi dan ingatan. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`test:dcg`	Keuntungan kumulatif diskon dari model akhir pada dataset pengujian. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`test:macro_f_beta`	Skor F-beta dari model akhir pada dataset pengujian. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`test:macro_precision`	Skor presisi model akhir pada dataset pengujian. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`test:macro_recall`	Skor recall model akhir pada dataset tes. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`test:mse`	Kesalahan kuadrat rata-rata dari model akhir pada dataset pengujian. Metrik objektif ini hanya berlaku untuk regresi.	Minimalkan

Nama Metrik	Deskripsi	Arah Optimasi
`test:multiclass_accuracy`	Keakuratan model akhir pada dataset pengujian. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`test:multiclass_top_k_accuracy`	Akurasi di antara label k teratas yang diprediksi pada kumpulan data pengujian. Jika Anda memilih metrik ini sebagai tujuan, kami sarankan untuk mengatur nilai k menggunakan `accuracy_top_k` hiperparameter. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`test:objective_loss`	Nilai rata-rata dari fungsi kerugian objektif pada dataset uji setelah model dilatih. Secara default, kerugian adalah kerugian logistik untuk klasifikasi biner dan kerugian kuadrat untuk regresi. Untuk mengatur kerugian ke jenis lain, gunakan `loss` hiperparameter.	Minimalkan
`test:precision`	Ketepatan model akhir pada dataset pengujian. Jika Anda memilih metrik ini sebagai tujuan, kami sarankan untuk menetapkan penarikan target dengan menyetel `binary_classifier_model_selection` hyperparameter ke `precision_at_target_recall` dan menetapkan nilai untuk `target_recall` hiperparameter. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan

Nama Metrik	Deskripsi	Arah Optimasi
`test:recall`	Penarikan kembali model akhir pada dataset pengujian. Jika Anda memilih metrik ini sebagai tujuan, kami sarankan untuk menetapkan presisi target dengan menyetel`binary_classifier_model_selection` hyperparameter ke`recall_at_target_precision` dan menetapkan nilai untuk`target_precision` hiperparameter. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`test:roc_auc_score`	Area di bawah kurva karakteristik operasi penerima (kurva ROC) dari model akhir pada kumpulan data pengujian. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`validation:absolute_loss`	Hilangnya mutlak model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk regresi.	Minimalkan
`validation:binary_classification_accuracy`	Keakuratan model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`validation:binary_f_beta`	Skor F-beta dari model akhir pada dataset validasi. Secara default, skor F-beta adalah skor F1, yang merupakan rata-rata harmonik dari`validation:precision` dan`validation:recall` metrik. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`validation:dcg`	Keuntungan kumulatif diskon dari model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan

Nama Metrik	Deskripsi	Arah Optimasi
`validation:macro_f_beta`	Skor F-beta dari model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`validation:macro_precision`	Skor presisi model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`validation:macro_recall`	Skor recall model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`validation:mse`	Kesalahan kuadrat rata-rata dari model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk regresi.	Minimalkan
`validation:multiclass_accuracy`	Keakuratan model akhir pada dataset validasi. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`validation:multiclass_top_k_accuracy`	Akurasi di antara label k teratas yang diprediksi pada kumpulan data validasi. Jika Anda memilih metrik ini sebagai tujuan, kami sarankan untuk mengatur nilai k menggunak an`accuracy_top_k` hiperparameter. Metrik objektif ini hanya berlaku untuk klasifikasi multiclass.	Maksimalkan
`validation:objective_loss`	Nilai rata-rata dari fungsi kerugian objektif pada dataset validasi setiap zaman. Secara default, kerugian adalah kerugian logistik untuk klasifikasi biner dan kerugian kuadrat untuk regresi. Untuk mengatur kerugian ke jenis lain, gunakan`loss`hiperparameter.	Minimalkan

Nama Metrik	Deskripsi	Arah Optimasi
`validation:precision`	Ketepatan model akhir pada dataset validasi. Jika Anda memilih metrik ini sebagai tujuan, kami sarankan untuk menetapkan penarikan target dengan menyetel`binary_classifier_model_selection` hyperparameter ke`precision_at_target_recall` dan menetapkan nilai untuk`target_recall` hiperparameter. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`validation:recall`	Penarikan kembali model akhir pada dataset validasi. Jika Anda memilih metrik ini sebagai tujuan, kami sarankan untuk menetapkan presisi target dengan menyetel`binary_classifier_model_selection` hyperparameter ke`recall_at_target_precision` dan menetapkan nilai untuk`target_precision` hiperparameter. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan
`validation:rmse`	Kesalahan kuadrat rata-rata akar dari model akhir pada kumpulan data validasi. Metrik objektif ini hanya berlaku untuk regresi.	Minimalkan
`validation:roc_auc_score`	Area di bawah kurva karakteristik operasi penerima (kurva ROC) dari model akhir pada kumpulan data validasi. Metrik objektif ini hanya berlaku untuk klasifikasi biner.	Maksimalkan

Menyetel hiperparameter pelajar linier

Anda dapat menyetel model pembelajar linier dengan hyperparameters berikut.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
wd	ContinuousParameterRanges	MinValue: 1e-7, MaxValue: 1
l1	ContinuousParameterRanges	MinValue: 1e-7, MaxValue: 1
learning_rate	ContinuousParameterRanges	MinValue: 1e-5, MaxValue: 1
mini_batch_size	IntegerParameterRanges	MinValue: 100, MaxValue: 5000
use_bias	CategoricalParameterRanges	[True, False]
positive_example_weight_mult	ContinuousParameterRanges	MinValue: 1e-5,MaxValue:1e5

Format respons pelajar linier

Format respons JSON

Semua Amazon SageMaker algoritma bawaan mematuhi format inferensi input umum yang dijelaskan dalamFormat Data Umum - Inferensi. Berikut ini adalah format output yang tersedia untuk SageMaker algoritma pembelajar linier.

Klasifikasi Biner

```
let response =   {
    "predictions":    [
        {
            "score": 0.4,
            "predicted_label": 0
        }
    ]
}
```

Klasifikasi Multiclass

```
let response =   {
    "predictions":    [
        {
            "score": [0.1, 0.2, 0.4, 0.3],
            "predicted_label": 2
        }
    ]
}
```

Regresi

```
let response =   {
    "predictions":    [
        {
            "score": 0.4
        }
    ]
}
```

Format respons JSONLINES

Klasifikasi Biner

```
{"score": 0.4, "predicted_label": 0}
```

Klasifikasi Multiclass

```
{"score": [0.1, 0.2, 0.4, 0.3], "predicted_label": 2}
```

Regresi

```
{"score": 0.4}
```

Format respons RECORDIO

Klasifikasi Biner

```
[
    Record = {
        features = {},
```

```
        label = {
            'score': {
                keys: [],
                values: [0.4]  # float32
            },
            'predicted_label': {
                keys: [],
                values: [0.0]  # float32
            }
        }
    }
]
```

Klasifikasi Multiclass

```
[
    Record = {
    "features": [],
    "label":    {
            "score":  {
                    "values":   [0.1, 0.2, 0.3, 0.4]
            },
            "predicted_label":  {
                    "values":   [3]
            }
       },
    "uid":  "abc123",
    "metadata": "{created_at: '2017-06-03'}"
   }
]
```

Regresi

```
[
    Record = {
        features = {},
        label = {
            'score': {
                keys: [],
                values: [0.4]  # float32
            }
        }
    }
```

```
]
```

TabTransformer

TabTransformeradalah arsitektur pemodelan data tabular mendalam baru untuk pembelajaran yang diawasi. TabTransformer Arsitekturnya dibangun di atas self-attention-based Transformers. Lapisan Transformer mengubah penyematan fitur kategoris menjadi penyematan kontekstual yang kuat untuk mencapai akurasi prediksi yang lebih tinggi. Selain itu, penyematan kontekstual yang dipelajari sangat kuat terhadap fitur data TabTransformer yang hilang dan berisik, dan memberikan interpretasi yang lebih baik.

Cara menggunakan SageMaker TabTransformer

Anda dapat menggunakan TabTransformer sebagai algoritma SageMaker bawaan Amazon. Bagian berikut menjelaskan cara menggunakan TabTransformer dengan SageMaker Python SDK. Untuk informasi tentang cara menggunakan TabTransformer dari Amazon SageMaker Studio Classic UI, lihatSageMaker JumpStart.

- Gunakan TabTransformer sebagai algoritma bawaan

  Gunakan algoritma TabTransformer bawaan untuk membangun wadah TabTransformer pelatihan seperti yang ditunjukkan pada contoh kode berikut. Anda dapat secara otomatis melihat URI image algoritma TabTransformer bawaan menggunakan SageMaker `image_uris.retrieve` API (atau `get_image_uri` API jika menggunakan Amazon SageMaker Python SDK versi 2).

  Setelah menentukan URI TabTransformer image, Anda dapat menggunakan TabTransformer container untuk membuat estimator menggunakan SageMaker Estimator API dan memulai tugas pelatihan. Algoritma TabTransformer bawaan berjalan dalam mode skrip, tetapi skrip pelatihan disediakan untuk Anda dan tidak perlu menggantinya. Jika Anda memiliki pengalaman luas menggunakan mode skrip untuk membuat pekerjaan SageMaker pelatihan, maka Anda dapat memasukkan skrip TabTransformer pelatihan Anda sendiri.

  ```
  from sagemaker import image_uris, model_uris, script_uris

  train_model_id, train_model_version, train_scope = "pytorch-
  tabtransformerclassification-model", "*", "training"
  training_instance_type = "ml.p3.2xlarge"

  # Retrieve the docker image
  train_image_uri = image_uris.retrieve(
      region=None,
  ```

```
    framework=None,
    model_id=train_model_id,
    model_version=train_model_version,
    image_scope=train_scope,
    instance_type=training_instance_type
)

# Retrieve the training script
train_source_uri = script_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 script_scope=train_scope
)

train_model_uri = model_uris.retrieve(
    model_id=train_model_id, model_version=train_model_version,
 model_scope=train_scope
)

# Sample training data is available in this bucket
training_data_bucket = f"jumpstart-cache-prod-{aws_region}"
training_data_prefix = "training-datasets/tabular_binary/"

training_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
train"
validation_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}/
validation"

output_bucket = sess.default_bucket()
output_prefix = "jumpstart-example-tabular-training"

s3_output_location = f"s3://{output_bucket}/{output_prefix}/output"

from sagemaker import hyperparameters

# Retrieve the default hyperparameters for training the model
hyperparameters = hyperparameters.retrieve_default(
    model_id=train_model_id, model_version=train_model_version
)

# [Optional] Override default hyperparameters with custom values
hyperparameters[
    "n_epochs"
] = "50"
print(hyperparameters)
```

```
from sagemaker.estimator import Estimator
from sagemaker.utils import name_from_base

training_job_name = name_from_base(f"built-in-algo-{train_model_id}-training")

# Create SageMaker Estimator instance
tabular_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1,
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location
)

# Launch a SageMaker Training job by passing the S3 path of the training data
tabular_estimator.fit(
    {
        "training": training_dataset_s3_path,
        "validation": validation_dataset_s3_path,
    }, logs=True, job_name=training_job_name
)
```

Untuk informasi selengkapnya tentang cara mengatur TabTransformer sebagai algoritma bawaan, lihat contoh buku catatan berikut.

- Klasifikasi tabel dengan algoritma Amazon SageMaker TabTransformer
- Regresi tabular dengan algoritma Amazon SageMaker TabTransformer

Antarmuka Input dan Output untuk TabTransformer algoritma

TabTransformer beroperasi pada data tabular, dengan baris mewakili pengamatan, satu kolom mewakili variabel target atau label, dan kolom yang tersisa mewakili fitur.

SageMaker Implementasi TabTransformer dukungan CSV untuk pelatihan dan inferensi:

- Untuk Pelatihan ContentType, input yang valid harus teks/csv.

- Untuk Inferensi ContentType, input yang valid harus teks/csv.

> **Note**
>
> Untuk pelatihan CSV, algoritme mengasumsikan bahwa variabel target ada di kolom pertama dan CSV tidak memiliki catatan header.
> Untuk inferensi CSV, algoritme mengasumsikan bahwa input CSV tidak memiliki kolom label.

Format input untuk data pelatihan, data validasi, dan fitur kategoris

Perhatikan cara memformat data pelatihan Anda untuk masukan ke TabTransformer model. Anda harus menyediakan jalur ke bucket Amazon S3 yang berisi data pelatihan dan validasi Anda. Anda juga dapat menyertakan daftar fitur kategoris. Gunakan saluran `training` dan `validation` saluran untuk memberikan data input Anda. Atau, Anda hanya dapat menggunakan `training` saluran.

Gunakan kedua **`validation`** saluran **`training`** dan

Anda dapat memberikan data input Anda melalui dua jalur S3, satu untuk `training` saluran dan satu untuk `validation` saluran. Setiap jalur S3 dapat berupa awalan S3 yang menunjuk ke satu atau lebih file CSV atau jalur S3 lengkap yang menunjuk ke satu file CSV tertentu. Variabel target harus berada di kolom pertama file CSV Anda. Variabel prediktor (fitur) harus berada di kolom yang tersisa. Jika beberapa file CSV disediakan untuk `validation` saluran `training` atau, TabTransformer algoritme menggabungkan file. Data validasi digunakan untuk menghitung skor validasi di akhir setiap iterasi peningkatan. Penghentian awal diterapkan ketika skor validasi berhenti membaik.

Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan file atau file data pelatihan Anda. Jika Anda menyediakan file JSON untuk fitur kategoris, `training` saluran Anda harus menunjuk ke awalan S3 dan bukan file CSV tertentu. File ini harus berisi kamus Python di mana kuncinya adalah string `"cat_index_list"` dan nilainya adalah daftar bilangan bulat unik. Setiap bilangan bulat dalam daftar nilai harus menunjukkan indeks kolom dari fitur kategoris yang sesuai dalam file CSV data pelatihan Anda. Setiap nilai harus berupa bilangan bulat positif (lebih besar dari nol karena nol mewakili nilai target), kurang dari `Int32.MaxValue` (2147483647), dan kurang dari jumlah kolom. Seharusnya hanya ada satu file JSON indeks kategoris.

Gunakan hanya **`training`** saluran:

Sebagai alternatif, Anda dapat memberikan data input Anda melalui jalur S3 tunggal untuk `training` saluran tersebut. Jalur S3 ini harus menunjuk ke direktori dengan subdirektori bernama `training/` yang berisi satu atau lebih file CSV. Anda dapat secara opsional menyertakan subdirektori lain di lokasi yang sama yang disebut `validation/` yang juga memiliki satu atau lebih file CSV. Jika data validasi tidak disediakan, maka 20% data pelatihan Anda diambil sampelnya secara acak untuk dijadikan data validasi. Jika prediktor Anda menyertakan fitur kategoris, Anda dapat memberikan file JSON bernama `categorical_index.json` di lokasi yang sama dengan subdirektori data Anda.

> **Note**
>
> Untuk mode input pelatihan CSV, total memori yang tersedia untuk algoritme (jumlah instance dikalikan dengan memori yang tersedia di`InstanceType`) harus dapat menampung kumpulan data pelatihan.

Rekomendasi instans Amazon EC2 untuk algoritme TabTransformer

SageMaker TabTransformer mendukung pelatihan CPU satu instans dan GPU instans tunggal. Meskipun biaya per instans lebih tinggi, GPU berlatih lebih cepat, membuatnya lebih hemat biaya. Untuk memanfaatkan pelatihan GPU, tentukan jenis instans sebagai salah satu instance GPU (misalnya, P3). SageMaker TabTransformer saat ini tidak mendukung pelatihan multi-GPU.

TabTransformer contoh notebook

Tabel berikut menguraikan berbagai contoh notebook yang membahas berbagai kasus penggunaan algoritma Amazon SageMaker TabTransformer .

Judul Notebook	Deskripsi
Klasifikasi tabel dengan algoritma Amazon SageMaker TabTransformer	Notebook ini menunjukkan penggunaan SageMaker TabTransformer algoritma Amazon untuk melatih dan menjadi tuan rumah model klasifikasi tabel.
Regresi tabular dengan algoritma Amazon SageMaker TabTransformer	Notebook ini menunjukkan penggunaan SageMaker TabTransformer algoritma Amazon untuk melatih dan menjadi tuan rumah model regresi tabular.

Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Bagaimana cara TabTransformer kerja

TabTransformer adalah arsitektur pemodelan data tabular mendalam baru untuk pembelajaran yang diawasi. TabTransformer Ini dibangun di atas Transformers berbasis perhatian diri. Lapisan Transformer mengubah penyematan fitur kategoris menjadi penyematan kontekstual yang kuat untuk mencapai akurasi prediksi yang lebih tinggi. Selain itu, penyematan kontekstual yang dipelajari sangat kuat terhadap fitur data TabTransformer yang hilang dan berisik, dan memberikan interpretasi yang lebih baik.

TabTransformer berkinerja baik dalam kompetisi pembelajaran mesin karena penanganannya yang kuat terhadap berbagai tipe data, hubungan, distribusi, dan keragaman hiperparameter yang dapat Anda sesuaikan. Anda dapat menggunakan TabTransformer untuk regresi, klasifikasi (biner dan multiclass), dan masalah peringkat.

Diagram berikut menggambarkan TabTransformer arsitektur.

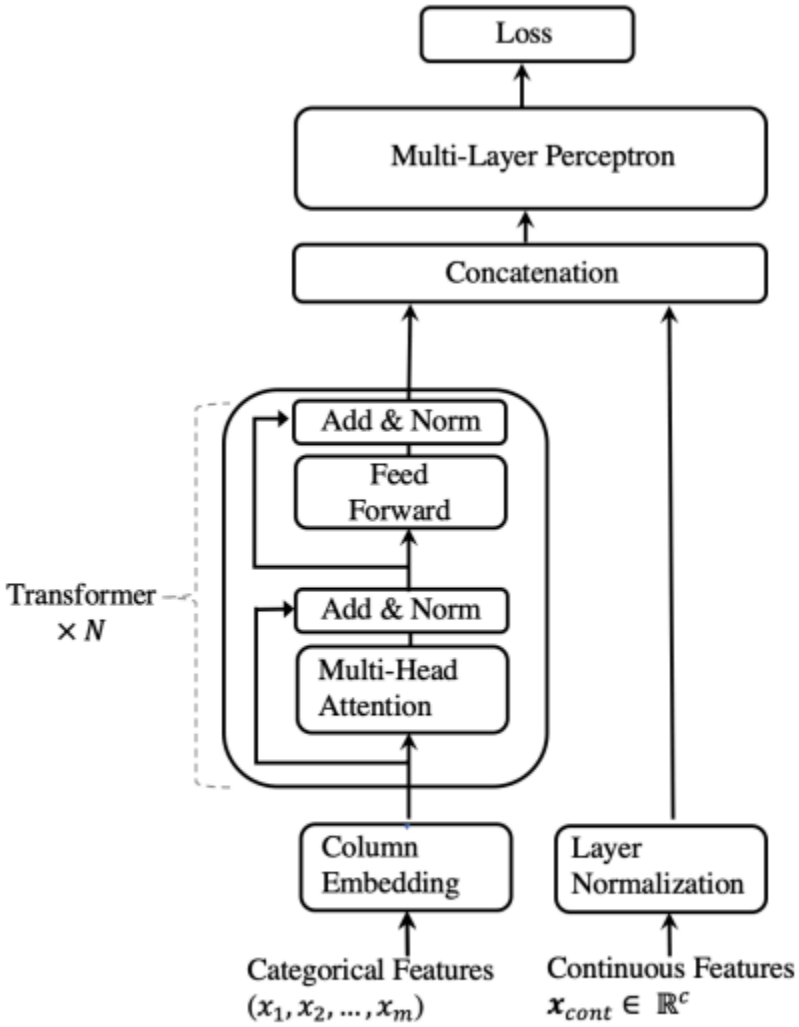


Figure 1: The architecture of TabTransformer.

Untuk informasi lebih lanjut, lihat TabTransformer: Pemodelan Data Tabular Menggunakan Embeddings Kontekstual.

TabTransformer hiperparameter

Tabel berikut berisi subset hiperparameter yang diperlukan atau paling umum digunakan untuk algoritma Amazon SageMaker TabTransformer . Pengguna mengatur parameter ini untuk memfasilitasi estimasi parameter model dari data. SageMaker TabTransformerAlgoritma adalah implementasi dari TabTransformerpaket open-source.

> **Note**
>
> Hyperparameter default didasarkan pada contoh kumpulan data di file. TabTransformer contoh notebook

SageMaker TabTransformer Algoritma secara otomatis memilih metrik evaluasi dan fungsi objektif berdasarkan jenis masalah klasifikasi. TabTransformer Algoritma mendeteksi jenis masalah klasifikasi berdasarkan jumlah label dalam data Anda. Untuk masalah regresi, metrik evaluasi adalah r kuadrat dan fungsi tujuannya adalah kesalahan kuadrat rata-rata. Untuk masalah klasifikasi biner, metrik evaluasi dan fungsi objektif keduanya adalah entropi silang biner. Untuk masalah klasifikasi multikelas, metrik evaluasi dan fungsi objektif keduanya adalah entropi silang multikelas.

> **Note**
>
> Metrik TabTransformer evaluasi dan fungsi objektif saat ini tidak tersedia sebagai hiperparameter. Sebagai gantinya, algoritme SageMaker TabTransformer bawaan secara otomatis mendeteksi jenis tugas klasifikasi (regresi, biner, atau multiclass) berdasarkan jumlah bilangan bulat unik di kolom label dan menetapkan metrik evaluasi dan fungsi objektif.

Nama Parameter	Deskripsi
`n_epochs`	Jumlah zaman untuk melatih jaringan saraf dalam. Nilai yang valid: bilangan bulat, rentang: Bilangan bulat positif. Nilai default:`5`.
`patience`	Pelatihan akan berhenti jika satu metrik dari satu titik data validasi tidak membaik di `patience` babak terakhir. Nilai yang valid: integer, range: (`2`,`60`). Nilai default:`10`.
`learning_rate`	Tingkat di mana bobot model diperbarui setelah mengerjakan setiap batch contoh pelatihan.

Nama Parameter	Deskripsi
	Nilai yang valid: float, range: Nomor floating point positif. Nilai default:`0.001`.
`batch_size`	Jumlah contoh disebarkan melalui jaringan. Nilai yang valid: integer, range: (`1`,`2048`). Nilai default:`256`.
`input_dim`	Dimensi penyematan untuk menyandikan kolom kategoris dan/atau kontinu. Nilai yang valid: string, salah satu dari berikut ini: `"16""32","64",,"128","256"`, atau`"512"`. Nilai default:`"32"`.
`n_blocks`	Jumlah blok encoder Transformer. Nilai yang valid: integer, range: (`1`,`12`). Nilai default:`4`.
`attn_dropout`	Tingkat putus sekolah diterapkan pada lapisan Multi-Head Attention. Nilai yang valid: float, range: (`0`,`1`). Nilai default:`0.2`.
`mlp_dropout`	Tingkat putus sekolah diterapkan ke FeedForward jaringan dalam lapisan encoder dan lapisan MLP akhir di atas encoder Transformer. Nilai yang valid: float, range: (`0`,`1`). Nilai default:`0.1`.

Nama Parameter	Deskripsi
`frac_shared_embed`	Fraksi embeddings dibagi oleh semua kategori yang berbeda untuk satu kolom tertentu. Nilai yang valid: float, range: (`0`,`1`). Nilai default:`0.25`.

Menyetel TabTransformer model

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hiperparameter pada kumpulan data pelatihan dan validasi Anda. Penyetelan model berfokus pada hiperparameter berikut:

> **Note**
>
> Fungsi tujuan pembelajaran dan metrik evaluasi keduanya secara otomatis ditetapkan berdasarkan jenis tugas klasifikasi, yang ditentukan oleh jumlah bilangan bulat unik di kolom label. Untuk informasi selengkapnya, lihat TabTransformer hiperparameter.

- Fungsi tujuan pembelajaran untuk mengoptimalkan selama pelatihan model
- Metrik evaluasi yang digunakan untuk mengevaluasi kinerja model selama validasi
- Satu set hyperparameters dan rentang nilai untuk masing-masing untuk digunakan saat menyetel model secara otomatis

Penyetelan model otomatis mencari hiperparameter pilihan Anda untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik evaluasi yang dipilih.

> **Note**
>
> Penyetelan model otomatis hanya TabTransformer tersedia dari Amazon SageMaker SDK, bukan dari konsol. SageMaker

Untuk informasi lebih lanjut tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik evaluasi dihitung oleh algoritma TabTransformer

SageMaker TabTransformer Algoritma menghitung metrik berikut untuk digunakan untuk validasi model. Metrik evaluasi secara otomatis ditetapkan berdasarkan jenis tugas klasifikasi, yang ditentukan oleh jumlah bilangan bulat unik di kolom label.

Nama Metrik	Deskripsi	Arah Optimasi	Pola Regex
`r2`	r persegi	memaksimalkan	`"metrics={'r2': (\\S+)}"`
`f1_score`	entropi silang biner	memaksimalkan	`"metrics={'f1': (\\S+)}"`
`accuracy_score`	entropi silang multiclass	memaksimalkan	`"metrics={'accuracy': (\\S+)}"`

Hiperparameter yang dapat disetel TabTransformer

Setel TabTransformer model dengan hyperparameters berikut. Hiperparameter yang memiliki efek terbesar dalam mengoptimalkan metrik TabTransformer evaluasi adalah:`learning_rate`,,,, `input_dim` n_blocksattn_dropout, `mlp_dropout` dan. `frac_shared_embed` Untuk daftar semua TabTransformer hyperparameters, lihatTabTransformer hiperparameter.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`learning_rate`	ContinuousParameterRanges	MinValue: 0,001, MaxValue: 0,01

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`input_dim`	CategoricalParameterRanges	[16, 32, 64, 128, 256, 512]
`n_blocks`	IntegerParameterRanges	MinValue: 1, MaxValue: 12
`attn_dropout`	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.8
`mlp_dropout`	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.8
`frac_shared_embed`	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.5

Algoritma XGBoost

XGBoost (Extreme Gradient Boosting) adalah implementasi open-source yang populer dan efisien dari algoritma gradient boosted trees. Peningkatan gradien adalah algoritma pembelajaran yang diawasi yang mencoba memprediksi variabel target secara akurat dengan menggabungkan ansambel perkiraan dari serangkaian model yang lebih sederhana dan lebih lemah. Algoritma XGBoost berkinerja baik dalam kompetisi pembelajaran mesin karena penanganannya yang kuat dari berbagai tipe data, hubungan, distribusi, dan berbagai hiperparameter yang dapat Anda sesuaikan. Anda dapat menggunakan XGBoost untuk regresi, klasifikasi (biner dan multiclass), dan masalah peringkat.

Anda dapat menggunakan rilis baru algoritma XGBoost baik sebagai algoritma SageMaker bawaan Amazon atau sebagai kerangka kerja untuk menjalankan skrip pelatihan di lingkungan lokal Anda. Implementasi ini memiliki jejak memori yang lebih kecil, logging yang lebih baik, validasi hyperparameter yang ditingkatkan, dan serangkaian metrik yang diperluas daripada versi aslinya. Ini menyediakan XGBoost `estimator` yang mengeksekusi skrip pelatihan di lingkungan XGBoost yang dikelola. Rilis SageMaker XGBoost saat ini didasarkan pada versi XGBoost asli 1.0, 1.2, 1.3, 1.5, dan 1.7.

Versi yang didukung

- Mode kerangka kerja (sumber terbuka): 1.0-1, 1.2-1, 1.2-2, 1.3-1, 1.5-1, 1.7-1
- Modus algoritma: 1.0-1, 1.2-1, 1.2-2, 1.3-1, 1.5-1, 1.7-1

Warning

Karena kapasitas komputasi yang diperlukan, SageMaker XGBoost versi 1.7-1 tidak kompatibel dengan instans GPU dari keluarga instans P2 untuk pelatihan atau inferensi.

Important

Saat Anda mengambil URI gambar SageMaker XGBoost, jangan gunakan `:latest` atau `:1` untuk tag URI gambar. Anda harus menentukan salah satu Versi yang didukung untuk memilih wadah XGBoost SageMaker -managed dengan versi paket XGBoost asli yang ingin Anda gunakan. Untuk menemukan versi paket yang dimigrasikan ke kontainer SageMaker XGBoost, lihat Docker Registry Paths dan Example Code, pilihWilayah AWS, dan navigasikan ke bagian XGBoost (algoritma).

Warning

Versi XGBoost 0.90 tidak digunakan lagi. Dukungan untuk pembaruan keamanan atau perbaikan bug untuk XGBoost 0.90 dihentikan. Sangat disarankan untuk memutakhirkan versi XGBoost ke salah satu versi yang lebih baru.

Note

XGBoost v1.1 tidak didukung SageMaker karena XGBoost 1.1 memiliki kemampuan rusak untuk menjalankan prediksi ketika input pengujian memiliki lebih sedikit fitur daripada data pelatihan dalam input LIBSVM. Kemampuan ini telah dipulihkan di XGBoost v1.2. Pertimbangkan untuk menggunakan SageMaker XGBoost 1.2-2 atau yang lebih baru.

Cara Menggunakan SageMaker XGBoost

Dengan SageMaker, Anda dapat menggunakan XGBoost sebagai algoritma atau kerangka kerja bawaan. Dengan menggunakan XGBoost sebagai kerangka kerja, Anda memiliki lebih banyak fleksibilitas dan akses ke skenario yang lebih canggih, seperti validasi silang k-fold, karena Anda dapat menyesuaikan skrip pelatihan Anda sendiri. Bagian berikut menjelaskan cara menggunakan XGBoost dengan Python SageMaker SDK. Untuk informasi tentang cara menggunakan XGBoost dari Amazon SageMaker Studio Classic UI, lihat. SageMaker JumpStart

- Gunakan XGBoost sebagai kerangka kerja

  Gunakan XGBoost sebagai kerangka kerja untuk menjalankan skrip pelatihan khusus Anda yang dapat menggabungkan pemrosesan data tambahan ke dalam pekerjaan pelatihan Anda. Dalam contoh kode berikut, Anda dapat menemukan bagaimana SageMaker Python SDK menyediakan XGBoost API sebagai kerangka kerja dengan cara yang sama menyediakan API kerangka kerja lainnya, seperti, MXNet, TensorFlow dan. PyTorch

```
import boto3
import sagemaker
from sagemaker.xgboost.estimator import XGBoost
from sagemaker.session import Session
from sagemaker.inputs import TrainingInput

# initialize hyperparameters
hyperparameters = {
        "max_depth":"5",
        "eta":"0.2",
        "gamma":"4",
        "min_child_weight":"6",
        "subsample":"0.7",
        "verbosity":"1",
        "objective":"reg:squarederror",
        "num_round":"50"}

# set an output path where the trained model will be saved
bucket = sagemaker.Session().default_bucket()
prefix = 'DEMO-xgboost-as-a-framework'
output_path = 's3://{}/{}/{}/output'.format(bucket, prefix, 'abalone-xgb-framework')

# construct a SageMaker XGBoost estimator
# specify the entry_point to your xgboost training script
estimator = XGBoost(entry_point = "your_xgboost_abalone_script.py",
```

```
                    framework_version='1.7-1',
                    hyperparameters=hyperparameters,
                    role=sagemaker.get_execution_role(),
                    instance_count=1,
                    instance_type='ml.m5.2xlarge',
                    output_path=output_path)

# define the data type and paths to the training and validation datasets
content_type = "libsvm"
train_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix, 'train'),
 content_type=content_type)
validation_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix,
 'validation'), content_type=content_type)

# execute the XGBoost training job
estimator.fit({'train': train_input, 'validation': validation_input})
```

Untuk end-to-end contoh menggunakan SageMaker XGBoost sebagai kerangka kerja, lihat Regresi dengan Amazon XGBoost SageMaker

- Gunakan XGBoost sebagai algoritma bawaan

  Gunakan algoritma bawaan XGBoost untuk membangun wadah pelatihan XGBoost seperti yang ditunjukkan pada contoh kode berikut. Anda dapat secara otomatis melihat URI image algoritma bawaan XGBoost menggunakan SageMaker `image_uris.retrieve` API (atau `get_image_uri` API jika menggunakan Amazon SageMaker Python SDK versi 1). Jika Anda ingin memastikan apakah `image_uris.retrieve` API menemukan URI yang benar, lihat Parameter umum untuk algoritme bawaan dan cari `xgboost` dari daftar lengkap URI gambar algoritme bawaan dan wilayah yang tersedia.

  Setelah menentukan URI gambar XGBoost, Anda dapat menggunakan wadah XGBoost untuk membuat estimator menggunakan Estimator API dan memulai pekerjaan pelatihan SageMaker . Mode algoritma bawaan XGBoost ini tidak menggabungkan skrip pelatihan XGBoost Anda sendiri dan berjalan langsung pada kumpulan data input.

  > **Important**
  >
  > Saat Anda mengambil URI gambar SageMaker XGBoost, jangan gunakan `:latest` atau `:1` untuk tag URI gambar. Anda harus menentukan salah satu Versi yang didukung untuk memilih wadah XGBoost SageMaker -managed dengan versi paket XGBoost asli yang ingin Anda gunakan. Untuk menemukan versi paket yang dimigrasikan ke kontainer

SageMaker XGBoost, lihat Docker Registry Paths dan Example Code, pilihWilayah AWS, dan navigasikan ke bagian XGBoost (algoritma).

```
import sagemaker
import boto3
from sagemaker import image_uris
from sagemaker.session import Session
from sagemaker.inputs import TrainingInput

# initialize hyperparameters
hyperparameters = {
        "max_depth":"5",
        "eta":"0.2",
        "gamma":"4",
        "min_child_weight":"6",
        "subsample":"0.7",
        "objective":"reg:squarederror",
        "num_round":"50"}

# set an output path where the trained model will be saved
bucket = sagemaker.Session().default_bucket()
prefix = 'DEMO-xgboost-as-a-built-in-algo'
output_path = 's3://{}/{}/{}/output'.format(bucket, prefix, 'abalone-xgb-built-in-
algo')

# this line automatically looks for the XGBoost image URI and builds an XGBoost
 container.
# specify the repo_version depending on your preference.
xgboost_container = sagemaker.image_uris.retrieve("xgboost", region, "1.7-1")

# construct a SageMaker estimator that calls the xgboost-container
estimator = sagemaker.estimator.Estimator(image_uri=xgboost_container,
                                          hyperparameters=hyperparameters,
                                          role=sagemaker.get_execution_role(),
                                          instance_count=1,
                                          instance_type='ml.m5.2xlarge',
                                          volume_size=5, # 5 GB
                                          output_path=output_path)

# define the data type and paths to the training and validation datasets
content_type = "libsvm"
```

```
train_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix, 'train'),
 content_type=content_type)
validation_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix,
 'validation'), content_type=content_type)

# execute the XGBoost training job
estimator.fit({'train': train_input, 'validation': validation_input})
```

Untuk informasi selengkapnya tentang cara mengatur XGBoost sebagai algoritma bawaan, lihat contoh notebook berikut.

- Pelatihan Spot Terkelola untuk XGBoost
- Regresi dengan Amazon SageMaker XGBoost (masukan Parket)

Antarmuka Input/Output untuk Algoritma XGBoost

Peningkatan gradien beroperasi pada data tabular, dengan baris mewakili pengamatan, satu kolom mewakili variabel target atau label, dan kolom yang tersisa mewakili fitur.

SageMaker Implementasi XGBoost mendukung format data berikut untuk pelatihan dan inferensi:

- teks/libsvm (default)
- teks/csv
- aplikasi/x-parket
- aplikasi/ x-recordio-protobuf

> **Note**
>
> Ada beberapa pertimbangan yang harus diperhatikan mengenai pelatihan dan input inferensi:
>
> - Untuk pelatihan dengan input kolumnar, algoritma mengasumsikan bahwa variabel target (label) adalah kolom pertama. Untuk inferensi, algoritma mengasumsikan bahwa input tidak memiliki kolom label.
> - Untuk data CSV, input tidak boleh memiliki catatan header.
> - Untuk pelatihan LIBSVM, algoritme mengasumsikan bahwa kolom berikutnya setelah kolom label berisi pasangan nilai indeks berbasis nol untuk fitur. Jadi setiap baris memiliki format:: <label><index0>: <value0><index1>:<value1>.

- Untuk informasi tentang jenis instans dan pelatihan terdistribusi, lihatRekomendasi Instans EC2 untuk Algoritma XGBoost.

Untuk mode input pelatihan CSV, total memori yang tersedia untuk algoritme (Hitungan Instance * memori yang tersedia di`InstanceType`) harus dapat menampung kumpulan data pelatihan. Untuk mode input pelatihan libsvm, itu tidak diperlukan, tetapi kami merekomendasikannya.

Untuk v1.3-1 dan yang lebih baru, SageMaker XGBoost menyimpan model dalam format biner internal XGBoost, menggunakan. `Booster.save_model` Versi sebelumnya menggunakan modul acar Python untuk membuat serialisasi/deserialisasi model.

> **Note**
>
> Perhatikan versi saat menggunakan model XGBoost di SageMaker XGBoost open source. Versi 1.3-1 dan yang lebih baru menggunakan format biner internal XGBoost sementara versi sebelumnya menggunakan modul acar Python.

Untuk menggunakan model yang dilatih dengan SageMaker XGBoost v1.3-1 atau yang lebih baru di XGBoost open source

- Gunakan kode Python berikut:

```
import xgboost as xgb

xgb_model = xgb.Booster()
xgb_model.load_model(model_file_path)
xgb_model.predict(dtest)
```

Untuk menggunakan model yang dilatih dengan versi XGBoost sebelumnya di SageMaker XGBoost open source

- Gunakan kode Python berikut:

```
import pickle as pkl
import tarfile

t = tarfile.open('model.tar.gz', 'r:gz')
```

```
t.extractall()

model = pkl.load(open(model_file_path, 'rb'))

# prediction with test data
pred = model.predict(dtest)
```

Untuk membedakan pentingnya titik data berlabel, gunakan Instance Weight Supports

- SageMaker XGBoost memungkinkan pelanggan untuk membedakan pentingnya titik data berlabel dengan menetapkan setiap instance nilai bobot. Untuk input teks/libsvm, pelanggan dapat menetapkan nilai bobot ke instance data dengan melampirkannya setelah label. Misalnya, `label:weight idx_0:val_0 idx_1:val_1...`. Untuk input teks/csv, pelanggan harus menyalakan `csv_weights` bendera di parameter dan melampirkan nilai bobot di kolom setelah label. Misalnya:`label,weight,val_0,val_1,...`).

Rekomendasi Instans EC2 untuk Algoritma XGBoost

SageMaker XGBoost mendukung pelatihan dan inferensi CPU dan GPU. Rekomendasi instans bergantung pada kebutuhan pelatihan dan inferensi, serta versi algoritma XGBoost. Pilih salah satu opsi berikut untuk informasi lebih lanjut:

- Pelatihan CPU
- Pelatihan GPU
- Pelatihan CPU terdistribusi
- Pelatihan GPU terdistribusi
- Inferensi

Pelatihan

Algoritma SageMaker XGBoost mendukung pelatihan CPU dan GPU.

Pelatihan CPU

SageMaker XGBoost 1.0-1 atau sebelumnya hanya melatih menggunakan CPU. Ini adalah algoritma yang terikat memori (sebagai lawan dari compute-bound). Jadi, instance komputasi tujuan umum (misalnya, M5) adalah pilihan yang lebih baik daripada instance yang dioptimalkan komputasi

(misalnya, C4). Selanjutnya, kami menyarankan Anda memiliki memori total yang cukup dalam instance yang dipilih untuk menyimpan data pelatihan. Meskipun mendukung penggunaan ruang disk untuk menangani data yang tidak sesuai dengan memori utama (out-of-core fitur yang tersedia dengan modus input libsvm), menulis file cache ke disk memperlambat waktu pemrosesan algoritma.

Pelatihan GPU

SageMaker XGBoost versi 1.2-2 atau yang lebih baru mendukung pelatihan GPU. Meskipun biaya per instans lebih tinggi, GPU berlatih lebih cepat, membuatnya lebih hemat biaya.

SageMaker XGBoost versi 1.2-2 atau yang lebih baru mendukung keluarga instans GPU P2, P3, G4dn, dan G5.

SageMaker XGBoost versi 1.7-1 atau yang lebih baru mendukung keluarga instans GPU P3, G4dn, dan G5. Perhatikan bahwa karena persyaratan kapasitas komputasi, versi 1.7-1 atau yang lebih baru tidak mendukung keluarga instans P2.

Untuk memanfaatkan pelatihan GPU, tentukan jenis instance sebagai salah satu instance GPU (misalnya, P3) dan atur `tree_method` hyperparameter ke skrip XGBoost yang `gpu_hist` ada.

Pelatihan terdistribusi

SageMaker XGBoost mendukung instance CPU dan GPU untuk pelatihan terdistribusi.

Pelatihan CPU terdistribusi

Untuk menjalankan pelatihan CPU pada beberapa instance, atur `instance_count` parameter untuk estimator ke nilai yang lebih besar dari satu. Data input harus dibagi antara jumlah total instance.

Bagilah data masukan di seluruh instance

Bagilah data input menggunakan langkah-langkah berikut:

1. Pecah data input menjadi file yang lebih kecil. Jumlah file harus setidaknya sama dengan jumlah instance yang digunakan untuk pelatihan terdistribusi. Menggunakan beberapa file yang lebih kecil sebagai lawan dari satu file besar juga mengurangi waktu pengunduhan data untuk pekerjaan pelatihan.
2. Saat membuat Anda TrainingInput, atur parameter distribusi ke`ShardedByS3Key`. Parameter ini memastikan bahwa setiap instance mendapat sekitar 1/n dari jumlah file di S3 jika ada n instance yang ditentukan dalam pekerjaan pelatihan.

Pelatihan GPU terdistribusi

Anda dapat menggunakan pelatihan terdistribusi dengan instans GPU tunggal atau multi-GPU.

Pelatihan terdistribusi dengan instans GPU tunggal

SageMaker XGBoost versi 1.2-2 hingga 1.3-1 hanya mendukung pelatihan instans GPU tunggal. Ini berarti bahwa bahkan jika Anda memilih instance multi-GPU, hanya satu GPU yang digunakan per instance.

Jika Anda menggunakan XGBoost versi 1.2-2 hingga 1.3-1, atau jika Anda tidak perlu menggunakan instance multi-GPU, maka Anda harus membagi data input Anda di antara jumlah total instance. Untuk informasi selengkapnya, lihat Bagilah data masukan di seluruh instance.

> **Note**
>
> Versi 1.2-2 hingga 1.3-1 dari SageMaker XGBoost hanya menggunakan satu GPU per instans meskipun Anda memilih instans multi-GPU.

Pelatihan terdistribusi dengan instans multi-GPU

Dimulai dengan versi 1.5-1, SageMaker XGBoost menawarkan pelatihan GPU terdistribusi dengan Dask. Dengan Dask Anda dapat menggunakan semua GPU saat menggunakan satu atau lebih instance multi-GPU. Dask juga berfungsi saat menggunakan instance GPU tunggal.

Berlatih dengan Dask menggunakan langkah-langkah berikut:

1. Entah menghilangkan `distribution` parameter di Anda TrainingInputatau mengaturnya ke`FullyReplicated`.
2. Saat mendefinisikan hyperparameters Anda, atur `use_dask_gpu_training` ke. `"true"`

> **Important**
>
> Pelatihan terdistribusi dengan Dask hanya mendukung format input CSV dan Parket. Jika Anda menggunakan format data lain seperti LIBSVM atau PROTOBUF, pekerjaan pelatihan gagal.
> Untuk data Parket, pastikan bahwa nama kolom disimpan sebagai string. Kolom yang memiliki nama tipe data lain akan gagal dimuat.

> **Important**
>
> Pelatihan terdistribusi dengan Dask tidak mendukung mode pipa. Jika mode pipa ditentukan, pekerjaan pelatihan gagal.

Ada beberapa pertimbangan yang harus diperhatikan saat melatih SageMaker XGBoost dengan Dask. Pastikan untuk membagi data Anda menjadi file yang lebih kecil. Dask membaca setiap file Parket sebagai partisi. Ada pekerja Dask untuk setiap GPU, jadi jumlah file harus lebih besar dari jumlah total GPU (jumlah instans* jumlah GPU per instance). Memiliki jumlah file yang sangat besar juga dapat menurunkan kinerja. Untuk informasi selengkapnya, lihat Praktik Terbaik Dask.

Variasi dalam output

`tree_method`Hyperparameter yang ditentukan menentukan algoritma yang digunakan untuk pelatihan XGBoost. Metode pohon`approx`, `hist` dan semuanya `gpu_hist` merupakan metode perkiraan dan menggunakan sketsa untuk perhitungan kuantil. Untuk informasi selengkapnya, lihat Metode Pohon dalam dokumentasi XGBoost. Sketsa adalah algoritma perkiraan. Oleh karena itu, Anda dapat mengharapkan variasi dalam model tergantung pada faktor-faktor seperti jumlah pekerja yang dipilih untuk pelatihan terdistribusi. Signifikansi variasi bergantung pada data.

Inferensi

SageMaker XGBoost mendukung instance CPU dan GPU untuk inferensi. Untuk informasi tentang jenis instance untuk inferensi, lihat Jenis Instance SageMaker Amazon Amazon.

Contoh Notebook XGBoost

Tabel berikut menguraikan berbagai contoh notebook yang membahas kasus penggunaan yang berbeda dari algoritma Amazon SageMaker XGBoost.

Judul Notebook	Deskripsi
Cara Membuat wadah XGBoost Kustom?	Notebook ini menunjukkan cara membuat Wadah XGBoost kustom dengan Amazon SageMaker Batch Transform.
Regresi dengan XGBoost menggunakan Parket	Notebook ini menunjukkan cara menggunak an dataset Abalone di Parquet untuk melatih model XGBoost.

Judul Notebook	Deskripsi
Bagaimana Melatih dan Menyelenggarakan Model Klasifikasi Multiclass?	Notebook ini menunjukkan cara menggunakan dataset MNIST untuk melatih dan meng-host model klasifikasi multiclass.
Bagaimana cara melatih Model untuk Prediksi Churn Pelanggan?	Notebook ini menunjukkan cara melatih model untuk Memprediksi Keberangkatan Pelanggan Seluler dalam upaya mengident ifikasi pelanggan yang tidak bahagia.
Pengantar infrastruktur Spot SageMaker Terkelola Amazon untuk Pelatihan XGBoost	Notebook ini menunjukkan cara menggunakan Instans Spot untuk pelatihan dengan XGBoost Container.
Bagaimana cara menggunakan Amazon SageMaker Debugger untuk men-debug Pekerjaan Pelatihan XGBoost?	Notebook ini menunjukkan cara menggunak an Amazon SageMaker Debugger untuk memantau pekerjaan pelatihan guna mendeteksi inkonsistensi menggunakan aturan debugging bawaan.
Bagaimana cara menggunakan Amazon SageMaker Debugger untuk men-debug Pekerjaan Pelatihan XGBoost secara Real-Time?	Notebook ini menunjukkan kepada Anda cara menggunakan dataset MNIST dan Amazon SageMaker Debugger untuk melakukan analisis real-time dari pekerjaan pelatihan XGBoost saat pekerjaan pelatihan sedang berjalan.

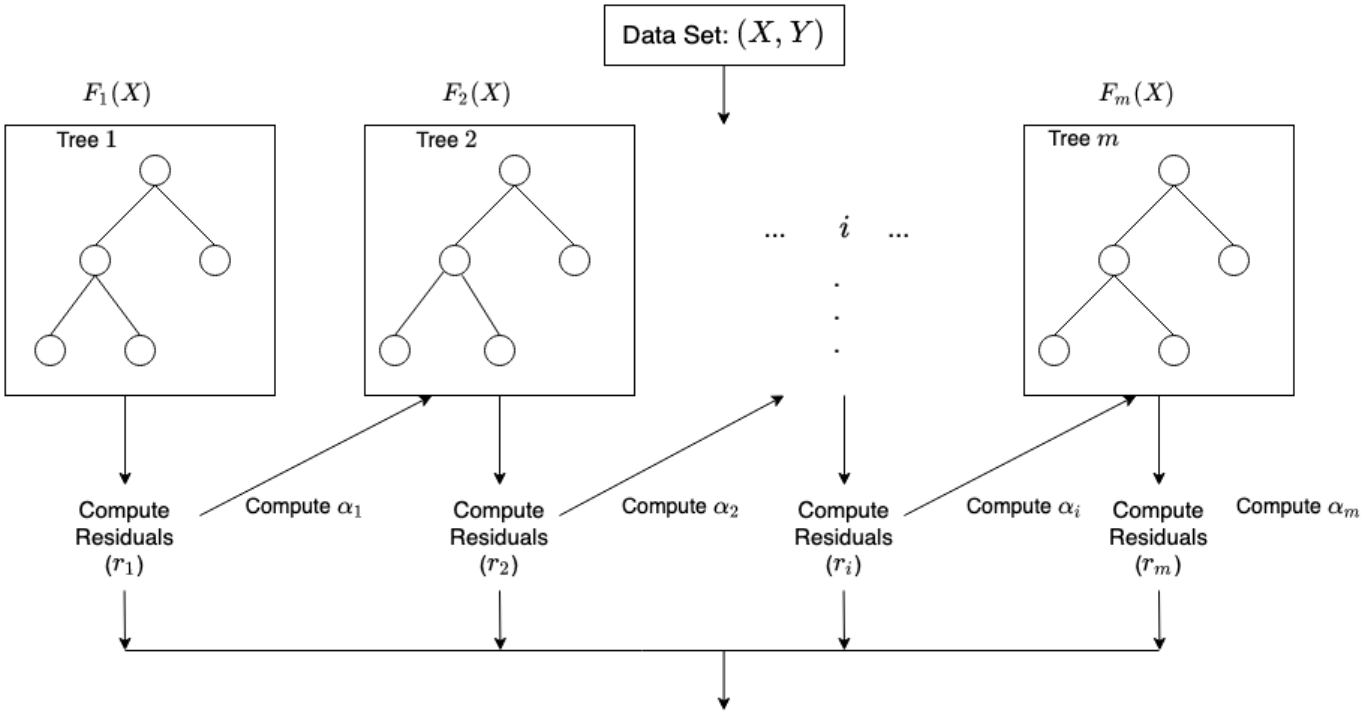
Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Contoh buku catatan pemodelan topik menggunakan algoritme pembelajaran linier terletak di bagian Pengantar Algoritma Amazon. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Bagaimana XGBoost Bekerja

XGBoost adalah implementasi open-source yang populer dan efisien dari algoritma gradient boosted trees. Peningkatan gradien adalah algoritma pembelajaran yang diawasi, yang mencoba memprediksi variabel target secara akurat dengan menggabungkan perkiraan serangkaian model yang lebih sederhana dan lebih lemah.

Saat menggunakan peningkatan gradien untuk regresi, peserta yang lemah adalah pohon regresi, dan setiap pohon regresi memetakan titik data input ke salah satu daunnya yang berisi skor kontinu. XGBoost meminimalkan fungsi objektif regularisasi (L1 dan L2) yang menggabungkan fungsi kerugian cembung (berdasarkan perbedaan antara output yang diprediksi dan target) dan istilah penalti untuk kompleksitas model (dengan kata lain, fungsi pohon regresi). Pelatihan berlangsung secara berulang, menambahkan pohon baru yang memprediksi residu atau kesalahan pohon sebelumnya yang kemudian digabungkan dengan pohon sebelumnya untuk membuat prediksi akhir. Ini disebut peningkatan gradien karena menggunakan algoritma penurunan gradien untuk meminimalkan kerugian saat menambahkan model baru.

Di bawah ini adalah ilustrasi singkat tentang cara kerja peningkatan pohon gradien.



Untuk detail lebih lanjut tentang XGBoost, lihat:

- XGBoost: Sistem Peningkatan Pohon yang Dapat Diskalakan
- Peningkatan Pohon Gradien
- Pengantar Pohon yang Ditingkatkan

XGBoost Hyperparameter

Tabel berikut berisi subset hiperparameter yang diperlukan atau paling umum digunakan untuk algoritma Amazon SageMaker XGBoost. Ini adalah parameter yang ditetapkan oleh pengguna untuk memfasilitasi estimasi parameter model dari data. Hyperparameter yang diperlukan yang harus ditetapkan terdaftar terlebih dahulu, dalam urutan abjad. Hyperparameter opsional yang dapat diatur tercantum berikutnya, juga dalam urutan abjad. Algoritma SageMaker XGBoost adalah implementasi dari paket DMLC XGBoost open-source. Untuk detail tentang set lengkap hyperparameter yang dapat dikonfigurasi untuk versi XGBoost ini, lihat Parameter XGBoost.

Nama Parameter	Deskripsi
num_class	Jumlah kelas. Diperlukan jika *objective* diatur ke multi:softmax atau multi:softprob. Nilai yang valid: Integer.
num_round	Jumlah putaran untuk menjalankan pelatihan. Diperlukan Nilai yang valid: Integer.
alpha	Istilah regularisasi L1 pada bobot. Meningkatkan nilai ini membuat model lebih konservatif. Opsional Nilai yang valid: Float. Nilai default: 0
base_score	Skor prediksi awal dari semua contoh, bias global. Opsional Nilai yang valid: Float. Nilai default: 0,5
booster	Booster mana yang akan digunakan. dartNilai gbtree dan menggunakan model berbasis pohon, sementara gblinear menggunakan fungsi linier. Opsional Nilai yang valid: String. Salah satu"gbtree","gblinear" , atau"dart".

Nama Parameter	Deskripsi
	Nilai default: `"gbtree"`
`colsample_bylevel`	Rasio subsampel kolom untuk setiap split, di setiap level. Opsional Nilai yang valid: Float. Rentang: [0,1]. Nilai default: 1
`colsample_bynode`	Rasio subsampel kolom dari setiap node. Opsional Nilai yang valid: Float. Rentang: (0,1]. Nilai default: 1
`colsample_bytree`	Rasio subsampel kolom saat membangun setiap pohon. Opsional Nilai yang valid: Float. Rentang: [0,1]. Nilai default: 1
`csv_weights`	Saat flag ini diaktifkan, XGBoost membedakan pentingnya instance untuk input csv dengan mengambil kolom kedua (kolom setelah label) dalam data pelatihan sebagai bobot instance. Opsional Nilai yang valid: 0 atau 1 Nilai default: 0

Nama Parameter	Deskripsi
`deterministic_histogram`	Saat flag ini diaktifkan, XGBoost membangun histogram pada GPU secara deterministik. Digunakan hanya jika `tree_method` diatur ke`gpu_hist`. Untuk daftar lengkap input yang valid, silakan lihat Parameter XGBoost. Opsional Nilai yang valid: String. Rentang: `"true"` atau`"false"`. Nilai default: `"true"`
`early_stopping_rounds`	Model berlatih sampai skor validasi berhenti membaik. Kesalahan validasi perlu dikurangi setidaknya setiap `early_stopping_rounds` untuk melanjutkan pelatihan. SageMakerhosting menggunakan model terbaik untuk inferensi. Opsional Nilai yang valid: Integer. Nilai default: -
`eta`	Penyusutan ukuran langkah yang digunakan dalam pembaruan untuk mencegah overfitting. Setelah setiap langkah peningkatan, Anda bisa langsung mendapatkan bobot fitur baru. `eta`Parameter sebenarnya mengecilkan bobot fitur untuk membuat proses peningkatan lebih konservatif. Opsional Nilai yang valid: Float. Rentang: [0,1]. Nilai default: 0,3

Nama Parameter	Deskripsi
`eval_metric`	Metrik evaluasi untuk data validasi. Metrik default ditetapkan sesuai dengan tujuan: • `rmse`: untuk regresi • `error`: untuk klasifikasi • `map`: untuk peringkat Untuk daftar input yang valid, lihat Parameter Tugas Pembelajaran XGBoost. Opsional Nilai yang valid: String. Nilai default: Default sesuai dengan tujuan.
`gamma`	Pengurangan kerugian minimum diperlukan untuk membuat partisi lebih lanjut pada simpul daun pohon. Semakin besar, semakin konservatif algoritmanya. Opsional Nilai yang valid: Float. Rentang: [0, ∞). Nilai default: 0
`grow_policy`	Mengontrol cara node baru ditambahkan ke pohon. Saat ini didukung hanya jika `tree_method` disetel ke`hist`. Opsional Nilai yang valid: String. Baik `"depthwise"` atau `"lossguide"` . Nilai default: `"depthwise"`

Nama Parameter	Deskripsi
`interaction_constraints`	Tentukan kelompok variabel yang diizinkan untuk berinteraksi. Opsional Nilai yang valid: Daftar bilangan bulat bersarang. Setiap bilangan bulat mewakili fitur, dan setiap daftar bersarang berisi fitur yang diizinkan untuk berinteraksi misalnya, [[1,2], [3,4,5]]. Nilai default: Tidak ada
`lambda`	Istilah regularisasi L2 pada bobot. Meningkatkan nilai ini membuat model lebih konservatif. Opsional Nilai yang valid: Float. Nilai default: 1
`lambda_bias`	Istilah regularisasi L2 pada bias. Opsional Nilai yang valid: Float. Rentang: [0.0, 1.0]. Nilai default: 0
`max_bin`	Jumlah maksimum tempat sampah diskrit ke fitur kontinu ember. Digunakan hanya jika `tree_method` diatur ke`hist`. Opsional Nilai yang valid: Integer. Nilai default: 256

Nama Parameter	Deskripsi
`max_delta_step`	Langkah delta maksimum diperbolehkan untuk estimasi berat setiap pohon. Ketika bilangan bulat positif digunakan, ini membantu membuat pembaruan lebih konservatif. Pilihan yang lebih disukai adalah menggunakannya dalam regresi logistik. Setel ke 1-10 untuk membantu mengontrol pembaruan. Opsional Nilai yang valid: Integer. Rentang: [0, ∞). Nilai default: 0
`max_depth`	Kedalaman maksimum pohon. Meningkatkan nilai ini membuat model lebih kompleks dan cenderung overfit. 0 menunjukkan tidak ada batas. Batas diperlukan ketika `grow_policy =depth-wise` . Opsional Nilai yang valid: Integer. Rentang: [0, ∞) Nilai default: 6
`max_leaves`	Jumlah maksimum node yang akan ditambahkan. Relevan hanya jika `grow_policy` disetel ke`lossguide` . Opsional Nilai yang valid: Integer. Nilai default: 0

Nama Parameter	Deskripsi
`min_child_weight`	Jumlah minimum berat badan contoh (hessian) yang dibutuhkan pada anak. Jika langkah partisi pohon menghasilkan simpul daun dengan jumlah bobot instance kurang dari`min_child_weight` , proses pembangunan melepaskan partisi lebih lanjut. Dalam model regresi linier, ini hanya sesuai dengan jumlah minimum instance yang diperlukan di setiap node. Semakin besar algoritme, semakin konservatif itu. Opsional Nilai yang valid: Float. Rentang: [0, ∞). Nilai default: 1
`monotone_constraints`	Menentukan kendala monotonisitas pada fitur apapun. Opsional Nilai yang valid: Tuple of Integers. Bilangan bulat yang valid: -1 (kendala menurun), 0 (tidak ada kendala), 1 (meningkatkan kendala). Misalnya, (0, 1): Tidak ada kendala pada prediktor pertama, dan kendala yang meningkat pada prediktor kedua. (-1, 1): Mengurangi kendala pada prediktor pertama, dan kendala yang meningkat pada prediktor kedua. Nilai default: (0, 0)
`normalize_type`	Jenis algoritma normalisasi. Opsional Nilai yang valid: Baik pohon atau hutan. Nilai default: pohon

Nama Parameter	Deskripsi
`nthread`	Jumlah thread paralel yang digunakan untuk menjalankan xgboost. Opsional Nilai yang valid: Integer. Nilai default: Jumlah utas maksimum.
`objective`	Menentukan tugas pembelajaran dan tujuan pembelaja ran yang sesuai. Contoh:`reg:logistic` ,`multi:sof tmax` ,`reg:squarederror` . Untuk daftar lengkap input yang valid, lihat Parameter Tugas Pembelajaran XGBoost. Opsional Nilai valid: String Nilai default: `"reg:squarederror"`
`one_drop`	Saat flag ini diaktifkan, setidaknya satu pohon selalu dijatuhkan selama putus sekolah. Opsional Nilai yang valid: 0 atau 1 Nilai default: 0
`process_type`	Jenis proses boosting untuk dijalankan. Opsional Nilai yang valid: String. Baik `"default"` atau `"update"`. Nilai default: `"default"`

Nama Parameter	Deskripsi
`rate_drop`	Tingkat putus sekolah yang menentukan fraksi pohon sebelumnya yang akan jatuh selama putus sekolah. Opsional Nilai yang valid: Float. Rentang: [0.0, 1.0]. Nilai default: 0.0
`refresh_leaf`	Ini adalah parameter dari plug-in updater 'refresh'. Ketika diatur ke `true` (1), daun pohon dan statistik simpul pohon diperbaru i. Saat disetel ke `false` (0), hanya statistik simpul pohon yang diperbarui. Opsional Nilai yang valid: 0/1 Nilai default: 1
`sample_type`	Jenis algoritma sampling. Opsional Nilai yang valid: Entah `uniform` atau`weighted`. Nilai default: `uniform`
`scale_pos_weight`	Mengontrol keseimbangan bobot positif dan negatif. Ini berguna untuk kelas yang tidak seimbang. Nilai khas untuk dipertimb angkan:`sum(negative cases) /sum(positive cases)` . Opsional Nilai yang valid: float Nilai default: 1

Nama Parameter	Deskripsi
`seed`	Benih nomor acak. Opsional Nilai yang valid: integer Nilai default: 0
`single_precision_histogram`	Saat flag ini diaktifkan, XGBoost menggunakan presisi tunggal untuk membangun histogram, bukan presisi ganda. Digunakan hanya jika `tree_method` diatur ke `hist` atau `gpu_hist`. Untuk daftar lengkap input yang valid, silakan lihat Parameter XGBoost. Opsional Nilai yang valid: String. Rentang: `"true"` atau `"false"` Nilai default: `"false"`
`sketch_eps`	Digunakan hanya untuk perkiraan algoritma serakah. Ini diterjemahkan menjadi O (1/`sketch_eps`) jumlah tempat sampah. Dibandingkan dengan jumlah tempat sampah yang dipilih secara langsung, ini dilengkapi dengan jaminan teoritis dengan akurasi sketsa. Opsional Nilai yang valid: Float, Range: [0, 1]. Nilai default: 0,03

Nama Parameter	Deskripsi
`skip_drop`	Probabilitas melewatkan prosedur putus sekolah selama iterasi peningkatan. Opsional Nilai yang valid: Float. Rentang: [0.0, 1.0]. Nilai default: 0.0
`subsample`	Rasio subsampel dari contoh pelatihan. Mengaturnya ke 0,5 berarti XGBoost secara acak mengumpulkan setengah dari instance data untuk menumbuhkan pohon. Ini mencegah overfitti ng. Opsional Nilai yang valid: Float. Rentang: [0,1]. Nilai default: 1
`tree_method`	Algoritma konstruksi pohon yang digunakan dalam XGBoost. Opsional Nilai yang valid: Salah satu`auto`,`exact`,`approx`,`hist`, atau`gpu_hist`. Nilai default: `auto`
`tweedie_variance_p ower`	Parameter yang mengontrol varians distribusi Tweedie. Opsional Nilai yang valid: Float. Rentang: (1, 2). Nilai default: 1.5

Nama Parameter	Deskripsi
`updater`	String yang dipisahkan koma yang mendefinisikan urutan pembaru pohon untuk dijalankan. Ini menyediakan cara modular untuk membangun dan memodifikasi pohon. Untuk daftar lengkap input yang valid, silakan lihat Parameter XGBoost. Opsional Nilai yang valid: string dipisahkan koma. Nilai default:`grow_colmaker` , pangkas
`use_dask_gpu_training`	Setel `use_dask_gpu_training` ke `"true"` jika Anda ingin menjalankan pelatihan GPU terdistribusi dengan Dask. Pelatihan GPU Dask hanya didukung untuk versi 1.5-1 dan yang lebih baru. Jangan setel nilai ini `"true"` untuk versi sebelumnya 1.5-1. Untuk informasi selengkapnya, lihat Pelatihan GPU terdistribusi. Opsional Nilai yang valid: String. Rentang: `"true"` atau `"false"` Nilai default: `"false"`
`verbosity`	Verbositas pesan pencetakan. Nilai yang valid: 0 (diam), 1 (peringatan), 2 (info), 3 (debug). Opsional Nilai default: 1

Menyetel Model XGBoost

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hiperparameter pada kumpulan data pelatihan dan validasi Anda. Anda memilih tiga jenis hyperparameters:

- `objective`fungsi pembelajaran untuk mengoptimalkan selama pelatihan model
- an `eval_metric` untuk digunakan untuk mengevaluasi kinerja model selama validasi
- satu set hyperparameters dan rentang nilai untuk masing-masing untuk digunakan saat menyetel model secara otomatis

Anda memilih metrik evaluasi dari kumpulan metrik evaluasi yang dihitung oleh algoritme. Penyetelan model otomatis mencari hiperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik evaluasi.

> **Note**
>
> Penyetelan model otomatis untuk XGBoost 0.90 hanya tersedia dari Amazon SageMaker SDK, bukan dari konsol. SageMaker

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Evaluasi Dihitung oleh Algoritma XGBoost

Algoritma XGBoost menghitung metrik berikut untuk digunakan untuk validasi model. Saat menyetel model, pilih salah satu metrik ini untuk mengevaluasi model. Untuk daftar lengkap `eval_metric` nilai yang valid, lihat Parameter Tugas Pembelajaran XGBoost

Nama Metrik	Deskripsi	Arah Optimasi
`validation:accuracy`	Tingkat klasifikasi, dihitung sebagai # (kanan) / # (semua kasus).	Maksimalkan
`validation:auc`	Area di bawah kurva.	Maksimalkan

Nama Metrik	Deskripsi	Arah Optimasi
`validation:error`	Tingkat kesalahan klasifikasi biner, dihitung sebagai # (kasus salah) /# (semua kasus).	Minimalkan
`validation:f1`	Indikator akurasi klasifikasi, dihitung sebagai rata-rata harmonik presisi dan ingatan.	Maksimalkan
`validation:logloss`	Kemungkinan log negatif.	Minimalkan
`validation:mae`	Berarti kesalahan absolut.	Minimalkan
`validation:map`	Rata-rata presisi rata-rata.	Maksimalkan
`validation:merror`	Tingkat kesalahan klasifikasi multiclass, dihitung sebagai # (kasus salah) /# (semua kasus).	Minimalkan
`validation:mlogloss`	Kemungkinan log negatif untuk klasifikasi multiclass.	Minimalkan
`validation:mse`	Berarti kesalahan kuadrat.	Minimalkan
`validation:ndcg`	Keuntungan kumulatif diskon yang dinormali sasi.	Maksimalkan
`validation:rmse`	Root berarti kesalahan kuadrat.	Minimalkan

Hiperparameter XGBoost yang dapat disetel

Setel model XGBoost dengan hyperparameter berikut. Hyperparameter yang memiliki efek terbesar dalam mengoptimalkan metrik evaluasi XGBoost adalah:`alpha`,,,, `min_child_weight` dan. `subsample` `eta` `num_round`

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
alpha	ContinuousParameterRanges	MinValue: 0, MaxValue: 1000
colsample_bylevel	ContinuousParameterRanges	MinValue: 0,1, MaxValue: 1
colsample_bynode	ContinuousParameterRanges	MinValue: 0,1, MaxValue: 1
colsample_bytree	ContinuousParameterRanges	MinValue: 0,5, MaxValue: 1
eta	ContinuousParameterRanges	MinValue: 0,1, MaxValue: 0,5
gamma	ContinuousParameterRanges	MinValue: 0, MaxValue: 5
lambda	ContinuousParameterRanges	MinValue: 0, MaxValue: 1000
max_delta_step	IntegerParameterRanges	[0, 10]
max_depth	IntegerParameterRanges	[0, 10]
min_child_weight	ContinuousParameterRanges	MinValue: 0, MaxValue: 120
num_round	IntegerParameterRanges	[1, 4000]
subsample	ContinuousParameterRanges	MinValue: 0,5, MaxValue: 1

Versi XGBoost yang tidak digunakan lagi dan Upgrade-nya

Topik ini berisi dokumentasi untuk versi Amazon sebelumnya SageMaker XGBoost yang masih tersedia tetapi tidak berlaku lagi. Ini juga memberikan petunjuk tentang cara meningkatkan versi XGBoost yang tidak digunakan lagi, bila memungkinkan, ke versi yang lebih terkini.

Topik

- Upgrade XGBoost Versi 0.90 ke Versi 1.5
- XGBoost Versi 0.72

Upgrade XGBoost Versi 0.90 ke Versi 1.5

Jika Anda menggunakan SageMaker Python SDK, untuk meng-upgrade XGBoost 0.90 pekerjaan yang ada ke versi 1.5, Anda harus memiliki versi 2.x SDK diinstal dan mengubah XGBoost`version`dan`framework_version`Parameter ke 1,5-1. Jika Anda menggunakan Boto3, Anda perlu memperbarui image Docker, dan beberapa hyperparameter dan tujuan pembelajaran.

Topik

- Meningkatkan SageMaker Python SDK Versi 1.x ke Versi 2.x
- Ubah tag gambar menjadi 1,5-1
- Ubah Gambar Docker untuk Boto3
- Perbarui Hyperparameter dan Tujuan Pembelajaran

Meningkatkan SageMaker Python SDK Versi 1.x ke Versi 2.x

Jika Anda masih menggunakan Versi 1.x SageMaker Python SDK, Anda harus meng-upgrade versi 2.x SageMaker Python SDK. Untuk informasi tentang versi terbaru dari SageMaker Python SDK, lihatGunakan Versi 2.x dari SageMaker Python. Untuk menginstal versi terbaru, jalankan:

```
python -m pip install --upgrade sagemaker
```

Ubah tag gambar menjadi 1,5-1

Jika Anda menggunakan SageMaker Python SDK dan menggunakan algoritma build-in XGBoost, ubah parameter versi di`image_uris.retrive`.

```
from sagemaker import image_uris
```

```
image_uris.retrieve(framework="xgboost", region="us-west-2", version="1.5-1")

estimator = sagemaker.estimator.Estimator(image_uri=xgboost_container,
                                          hyperparameters=hyperparameters,
                                          role=sagemaker.get_execution_role(),
                                          instance_count=1,
                                          instance_type='ml.m5.2xlarge',
                                          volume_size=5, # 5 GB
                                          output_path=output_path)
```

Jika Anda menggunakan SageMaker Python SDK dan menggunakan XGBoost sebagai kerangka kerja untuk menjalankan skrip pelatihan khusus Anda, ubah`framework_version`parameter dalam API XGBoost.

```
estimator = XGBoost(entry_point = "your_xgboost_abalone_script.py",
                    framework_version='1.5-1',
                    hyperparameters=hyperparameters,
                    role=sagemaker.get_execution_role(),
                    instance_count=1,
                    instance_type='ml.m5.2xlarge',
                    output_path=output_path)
```

`sagemaker.session.s3_input`di SageMaker Python SDK versi 1.x telah diubah namanya`sagemaker.inputs.TrainingInput`. Anda harus menggunakan`sagemaker.inputs.TrainingInput`seperti pada contoh berikut.

```
content_type = "libsvm"
train_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix, 'train'),
 content_type=content_type)
validation_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix, 'validation'),
 content_type=content_type)
```

Untuk daftar lengkap SageMaker Perubahan Python SDK versi 2.x, lihatGunakan Versi 2.x dari SageMaker Python.

Ubah Gambar Docker untuk Boto3

Jika Anda menggunakan Boto3 untuk melatih atau menerapkan model Anda, ubah tag gambar docker (1, 0,72, 0,90-1 atau 0,90-2) menjadi 1,5-1.

```
{
```

```
    "AlgorithmSpecification":: {
        "TrainingImage": "746614075791.dkr.ecr.us-west-1.amazonaws.com/sagemaker-
xgboost:1.5-1"
    }
    ...
}
```

Jika Anda menggunakan SageMaker Python SDK untuk mengambil jalur registri, mengubah`version`Parameter dalam`image_uris.retrieve`.

```
from sagemaker import image_uris
image_uris.retrieve(framework="xgboost", region="us-west-2", version="1.5-1")
```

Perbarui Hyperparameter dan Tujuan Pembelajaran

Parameter diam telah usang dan tidak lagi tersedia di XGBoost 1.5 dan versi yang lebih baru. Gunakan `verbosity` sebagai gantinya. Jika Anda menggunakan`reg:linear`tujuan pembelajaran, telah usang juga mendukung `reg:squarederror`. Gunakan `reg:squarederror` sebagai gantinya.

```
hyperparameters = {
    "verbosity": "2",
    "objective": "reg:squarederror",
    "num_round": "50",
    ...
}

estimator = sagemaker.estimator.Estimator(image_uri=xgboost_container,
                                          hyperparameters=hyperparameters,
                                          ...)
```

XGBoost Versi 0.72

> **Important**
>
> XGBoost 0.72 tidak digunakan lagi oleh Amazon SageMaker. Anda masih dapat menggunakan XGBoost versi lama ini (sebagai algoritme bawaan) dengan menarik URI gambarnya seperti yang ditunjukkan pada contoh kode berikut. Untuk XGBoost, URI gambar diakhiri dengan`:1`adalah untuk versi lama.

SageMaker Python SDK v1

```
import boto3
from sagemaker.amazon.amazon_estimator import get_image_uri

xgb_image_uri = get_image_uri(boto3.Session().region_name, "xgboost",
 repo_version="1")
```

SageMaker Python SDK v2

```
import boto3
from sagemaker import image_uris

xgb_image_uri = image_uris.retrieve("xgboost", boto3.Session().region_name,
 "1")
```

Jika Anda ingin menggunakan versi yang lebih baru, Anda harus secara eksplisit menentukan tag URI gambar (lihatVersi yang didukung).

Ini rilis sebelumnya dari Amazon SageMaker algoritma XGBoost didasarkan pada rilis 0,72. XGBoost(eXtreme Gradient Boosting) adalah implementasi sumber terbuka yang populer dan efisien dari algoritma pohon yang ditingkatkan gradien. Peningkatan gradien adalah algoritma pembelajaran yang diawasi yang mencoba memprediksi variabel target secara akurat dengan menggabungkan perkiraan serangkaian model yang lebih sederhana dan lebih lemah. XGBoost telah melakukan sangat baik dalam kompetisi machine learning karena dengan kuat menangani berbagai tipe data, hubungan, dan distribusi, dan karena banyaknya hyperparameter yang dapat di-tweak dan disetel untuk peningkatan kesesuaian. Fleksibilitas ini membuat XGBoost pilihan yang solid untuk masalah dalam regresi, klasifikasi (biner dan multiclass), dan peringkat.

Pelanggan harus mempertimbangkan untuk menggunakan rilis baruAlgoritma XGBoost. Mereka bisa menggunakannya sebagai SageMaker algoritme bawaan atau sebagai kerangka kerja untuk menjalankan skrip di lingkungan lokal mereka seperti biasanya, misalnya, lakukan dengan kerangka kerja deep learning Tensorflow. Implementasi baru memiliki jejak memori yang lebih kecil, pencatatan yang lebih baik, validasi hyperparameter yang ditingkatkan, dan serangkaian metrik yang diperluas. Implementasi XGBoost sebelumnya tetap tersedia untuk pelanggan jika mereka perlu menunda

migrasi ke versi baru. Tetapi implementasi sebelumnya ini akan tetap terkait dengan rilis XGBoost 0.72.

Antarmuka Input/Output untuk Rilis XGBoost 0.72

Peningkatan gradien beroperasi pada data tabular, dengan baris yang mewakili pengamatan, satu kolom mewakili variabel target atau label, dan kolom yang tersisa mewakili fitur.

Parameter SageMaker implementasi XGBoost mendukung format CSV dan libsvm untuk pelatihan dan inferensi:

- Untuk Training ContentType, input yang valid adalahtext/libsvm(default) atautext/csv.
- Untuk Inferensi ContentType, input yang valid adalahtext/libsvmatau (default)text/csv.

> **Note**
>
> Untuk pelatihan CSV, algoritme mengasumsikan bahwa variabel target ada di kolom pertama dan CSV tidak memiliki catatan header. Untuk inferensi CSV, algoritma mengasumsikan bahwa input CSV tidak memiliki kolom label.
> Untuk pelatihan libsvm, algoritma mengasumsikan bahwa label ada di kolom pertama. Kolom berikutnya berisi pasangan nilai indeks berbasis nol untuk fitur. Jadi setiap baris memiliki format: <label><index0>: <value0><index1>:<value1>... Permintaan inferensi untuk libsvm mungkin atau mungkin tidak memiliki label dalam format libsvm.

Ini berbeda dari yang lain SageMaker algoritma, yang menggunakan format input pelatihan protobuf untuk menjaga konsistensi yang lebih besar dengan format data XGBoost standar.

Untuk mode input pelatihan CSV, total memori yang tersedia untuk algoritme (Hitungan Instance * memori yang tersedia di`InstanceType`) harus mampu memegang dataset pelatihan. Untuk mode input pelatihan libsvm, ini tidak diwajibkan, tetapi kami merekomendasikannya.

SageMaker XGBoost menggunakan modul acar Python untuk membuat serialisasi/deserialize model, yang dapat digunakan untuk menyimpan/memuat model.

Untuk menggunakan model yang dilatih dengan SageMaker XGBoost di XGBoost sumber terbuka

- Gunakan kode Phyton berikut:

```
import pickle as pkl
```

```
import tarfile
import xgboost

t = tarfile.open('model.tar.gz', 'r:gz')
t.extractall()

model = pkl.load(open(model_file_path, 'rb'))

# prediction with test data
pred = model.predict(dtest)
```

Untuk membedakan pentingnya titik data berlabel gunakan Instance Weight Supports

- SageMaker XGBoost memungkinkan pelanggan untuk membedakan pentingnya titik data berlabel dengan menetapkan setiap instans nilai bobot. Untuktext/libsvmmasukan, pelanggan dapat menetapkan nilai berat untuk contoh data dengan melampirkan mereka setelah label. Sebagai contoh, `label:weight idx_0:val_0 idx_1:val_1...`. Untuktext/csvmasukan, pelanggan perlu mengaktifkan`csv_weights`bendera dalam parameter dan melampirkan nilai berat di kolom setelah label. Misalnya:`label,weight,val_0,val_1,...`).

Rekomendasi Instans EC2 untuk Rilis XGBoost 0.72

SageMaker XGBoost saat ini hanya melatih menggunakan CPU. Ini adalah algoritma memori-terikat (sebagai lawan dari compute-bound). Jadi, instance komputasi tujuan umum (misalnya, M4) adalah pilihan yang lebih baik daripada instance yang dioptimalkan komputasi (misalnya, C4). Selanjutnya, kami menyarankan Anda memiliki memori total yang cukup dalam instance tertentu untuk menyimpan data pelatihan. Meskipun mendukung penggunaan ruang disk untuk menangani data yang tidak sesuai dengan memori utama (out-of-core fitur yang tersedia dengan modus input libsvm), menulis file cache ke disk memperlambat waktu pemrosesan algoritma.

XGBoost Rilis 0.72 Contoh Notebook

Untuk contoh notebook yang menunjukkan cara menggunakan versi terbaru SageMaker XGBoost sebagai algoritma bawaan untuk melatih dan meng-host model regresi, lihatRegresi dengan Amazon SageMaker Algoritme XGBoost. Untuk menggunakan XGBoost versi 0,72, Anda perlu mengubah versi dalam kode sampel menjadi 0,72. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh di SageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMakerContohtab untuk melihat daftar semua SageMaker Sampel. Contoh pemodelan topik

notebook menggunakan algoritma XGBoost terletak diPengantar algoritma AmazonBagian. Untuk membuka notebook, klikGunakantab dan pilihBuat salinan.

XGBoost Rilis 0.72 Hyperparameter

Tabel berikut berisi hyperparameters untuk algoritma XGBoost. Ini adalah parameter yang ditetapkan oleh pengguna untuk memfasilitasi estimasi parameter model dari data. Hyperparameter yang diperlukan yang harus ditetapkan dicantumkan terlebih dahulu, dalam urutan abjad. Hyperparameter opsional yang dapat diatur tercantum berikutnya, juga dalam urutan abjad. Parameter SageMaker algoritma XGBoost adalah implementasi dari paket XGBoost open-source. SAAT INI SageMaker mendukung versi 0.72. Untuk detail selengkapnya tentang konfigurasi hyperparameter untuk versi XGBoost ini, lihatParameter XGBoost.

Nama Parameter	Deskripsi
`num_class`	Jumlah kelas. Wajibjikaobjective diatur kemulti: softmaxataumulti: softprob. Nilai yang valid: integer
`num_round`	Jumlah putaran untuk menjalankan pelatihan. Wajib Nilai yang valid: integer
`alpha`	Istilah regularisasi L1 pada bobot. Meningkatkan nilai ini membuat model lebih konservatif. Opsional Nilai yang benar: float Nilai default: 0 hari
`base_score`	Skor prediksi awal dari semua contoh,Bias global. Opsional Nilai yang benar: float

Nama Parameter	Deskripsi
	Nilai default: 0,5
`booster`	Booster yang akan digunakan. Parameter`gbtree`dan`dart`nilai menggunakan model berbasis pohon, sementara `gblinear`menggunakan fungsi linier. Opsional Nilai yang valid: String. Salah satu`gbtree`,`gblinear`, atau`dart`. Nilai default: `gbtree`
`colsample_bylevel`	Rasio subsampel kolom untuk setiap split, di setiap level. Opsional Nilai yang valid: Apung. Rentang: [0,1]. Nilai default: 1
`colsample_bytree`	Rasio subsampel kolom saat membangun setiap pohon. Opsional Nilai yang valid: Apung. Rentang: [0,1]. Nilai default: 1
`csv_weights`	Saat flag ini diaktifkan, XGBoost membedakan pentingnya instance untuk input csv dengan mengambil kolom kedua (kolom setelah label) dalam data pelatihan sebagai bobot instans. Opsional Nilai yang benar: 0 atau 1 Nilai default: 0 hari

Nama Parameter	Deskripsi
`early_stopping_rounds`	Model melatih sampai skor validasi berhenti membaik. Kesalahan validasi perlu mengurangi setidaknya setia`early_stopping_rounds` untuk melanjutkan pelatihan. SageMaker hosting menggunakan model terbaik untuk inferensi. Opsional Nilai yang valid: integer Nilai default: -
`eta`	Penyusutan ukuran langkah digunakan dalam pembaruan untuk mencegah overfitting. Setelah setiap langkah meningkatkan, Anda bisa langsung mendapatkan bobot fitur baru. Parameter `eta`parameter benar-benar menyusut bobot fitur untuk membuat proses meningkatkan lebih konservatif. Opsional Nilai yang valid: Apung. Rentang: [0,1]. Nilai default: 0,3

Nama Parameter	Deskripsi
`eval_metric`	Metrik evaluasi untuk data validasi. Metrik default ditetapkan sesuai dengan tujuan: • `rmse`: untuk regresi • `error`: untuk klasifikasi • `map`: untuk peringkat Untuk daftar input yang valid, lihatParameter XGBoost. Opsional Nilai yang valid: String Nilai default: Default sesuai dengan tujuan.
`gamma`	Pengurangan kerugian minimum yang diperlukan untuk membuat partisi lebih lanjut pada simpul daun pohon. Semakin besar, semakin konservatif algoritmanya. Opsional Nilai yang valid: Apung. Rentang: [0, ∞). Nilai default: 0 hari
`grow_policy`	Mengontrol cara node baru ditambahkan ke pohon. Saat ini didukung hanya jika`tree_method` diatur ke`hist`. Opsional Nilai yang valid: String. Baik `depthwise` atau `lossguide` . Nilai default: `depthwise`

Nama Parameter	Deskripsi
`lambda`	Istilah regularisasi L2 pada bobot. Meningkatkan nilai ini membuat model lebih konservatif. Opsional Nilai yang benar: float Nilai default: 1
`lambda_bias`	Istilah regularisasi L2 pada bias. Opsional Nilai yang valid: Apung. Rentang: [0.0, 1.0]. Nilai default: 0 hari
`max_bin`	Jumlah maksimum tempat sampah terpisah ke fitur kontinu bucket. Digunakan hanya jika`tree_method` diatur ke`hist`. Opsional Nilai yang valid: integer Nilai default: 256
`max_delta_step`	Langkah delta maksimum yang diizinkan untuk estimasi berat setiap pohon. Ketika bilangan bulat positif digunakan, ini membantu membuat pembaruan lebih konservatif. Pilihan yang disukai adalah menggunakannya dalam regresi logistik. Atur ke 1-10 untuk membantu mengontrol pembaruan. Opsional Nilai yang valid: Bilangan bulat. Rentang: [0, ∞). Nilai default: 0 hari

Nama Parameter	Deskripsi
`max_depth`	Kedalaman maksimum pohon. Meningkatkan nilai ini membuat model lebih kompleks dan cenderung overfit. 0 menunjukkan tidak ada batas. Batas diperlukan saat`grow_policy =depth-wise` . Opsional Nilai yang valid: Bilangan bulat. Rentang: [0, ∞) Nilai default: 6
`max_leaves`	Jumlah maksimum node yang akan ditambahkan. Relevan hanya jika`grow_policy` diatur ke`lossguide` . Opsional Nilai yang valid: integer Nilai default: 0 hari
`min_child_weight`	Jumlah minimum berat instans (hessian) yang dibutuhkan pada anak. Jika langkah partisi pohon menghasilkan simpul daun dengan jumlah berat instance kurang dari`min_child_weight` , proses pembangunan menyerah partisi lebih lanjut. Dalam model regresi linier, ini hanya sesuai dengan jumlah minimum contoh yang diperlukan di setiap node. Semakin besar algoritmanya, semakin konservatif. Opsional Nilai yang valid: Apung. Rentang: [0, ∞). Nilai default: 1

Nama Parameter	Deskripsi
`normalize_type`	Jenis algoritma normalisasi. Opsional Nilai yang valid: Entahbaganataurimba. Nilai default:bagan
`nthread`	Jumlah benang parallel yang digunakan untuk menjalank anxgboost. Opsional Nilai yang valid: integer Nilai default: Jumlah maksimum utas.
`objective`	Menentukan tugas belajar dan tujuan pembelajaran yang sesuai. Contoh: `reg:logistic` ,`reg:softmax` ,`multi:squarederror` . Untuk daftar lengkap input yang valid, lihatParameter XGBoost. Opsional Nilai yang benar: String Nilai default: `reg:squarederror`
`one_drop`	Ketika bendera ini diaktifkan, setidaknya satu pohon selalu dijatuhkan selama putus sekolah. Opsional Nilai yang benar: 0 atau 1 Nilai default: 0 hari

Nama Parameter	Deskripsi
`process_type`	Jenis proses meningkatkan untuk menjalankan. Opsional Nilai yang valid: String. Baik `default` atau `update`. Nilai default: `default`
`rate_drop`	Tingkat putus sekolah yang menentukan fraksi pohon sebelumnya untuk menjatuhkan selama putus sekolah. Opsional Nilai yang valid: Apung. Rentang: [0.0, 1.0]. Nilai default: 0.0
`refresh_leaf`	Ini adalah parameter dari 'refresh' updater plug-in. Ketika diatur k`etrue`(1), daun pohon dan statistik simpul pohon diperbaru i. Ketika diatur k`efalse`(0), hanya statistik simpul pohon yang diperbarui. Opsional Nilai yang valid: 0/1 Nilai default: 1
`sample_type`	Jenis algoritma sampling. Opsional Nilai yang valid: Baik `uniform` atau `weighted`. Nilai default: `uniform`

Nama Parameter	Deskripsi
`scale_pos_weight`	Mengontrol keseimbangan bobot positif dan negatif. Ini berguna untuk kelas yang tidak seimbang. Nilai khas untuk dipertimb angkan:`sum(negative cases) /sum(positive cases)` . Opsional Nilai yang benar: float Nilai default: 1
`seed`	Benih nomor acak. Opsional Nilai yang valid: integer Nilai default: 0 hari
`silent`	0 berarti mencetak pesan yang berjalan, 1 berarti mode diam. Nilai yang benar: 0 atau 1 Opsional Nilai default: 0 hari
`sketch_eps`	Digunakan hanya untuk algoritma serakah perkiraan. Ini diterjemahkan ke dalam O (1/`sketch_eps`) jumlah tempat sampah. Dibandingkan dengan jumlah tempat sampah yang dipilih secara langsung, ini dilengkapi dengan jaminan teoritis dengan akurasi sketsa. Opsional Nilai yang valid: Mengapung, Rentang: [0, 1]. Nilai default: 0,03

Nama Parameter	Deskripsi
`skip_drop`	Probabilitas melewatkan prosedur putus sekolah selama iterasi meningkatkan. Opsional Nilai yang valid: Apung. Rentang: [0.0, 1.0]. Nilai default: 0.0
`subsample`	Rasio subsampel dari instance pelatihan. Mengaturnya ke 0,5 berarti XGBoost secara acak mengumpulkan setengah dari instance data untuk menanam pohon. Hal ini untuk mencegah overfitting. Opsional Nilai yang valid: Apung. Rentang: [0,1]. Nilai default: 1
`tree_method`	Algoritma konstruksi pohon yang digunakan dalam XGBoost. Opsional Nilai yang valid: Salah satu`auto`,`exact`,`approx`, atau`hist`. Nilai default: `auto`
`tweedie_variance_power`	Parameter yang mengontrol varians distribusi Tweedie. Opsional Nilai yang valid: Apung. Rentang: (1, 2). Nilai default: 1.5

Nama Parameter	Deskripsi
`updater`	Sebuah string dipisahkan koma yang mendefinisikan urutan updaters pohon untuk menjalankan. Ini menyediakan cara modular untuk membangun dan memodifikasi pohon. Untuk daftar lengkap input yang valid, silakan lihatParameter XGBoost. Opsional Nilai yang valid: string yang dipisahkan koma. Nilai default:`grow_colmaker` , prune

Tune XGBoost Rilis 0.72 Model

Penalaan model otomatis, juga dikenal sebagai penyetelan hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada kumpulan data pelatihan dan validasi Anda. Anda memilih tiga jenis hiperparameter:

- pembelajaranobjectiveberfungsi untuk mengoptimalkan selama pelatihan model
- sebuaheval_metricuntuk digunakan untuk mengevaluasi kinerja model selama validasi
- satu set hyperparameters dan berbagai nilai untuk masing-masing untuk digunakan saat menyetel model secara otomatis

Anda memilih metrik evaluasi dari serangkaian metrik evaluasi yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik evaluasi.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma XGBoost Release 0.72

Algoritma XGBoost berdasarkan versi 0.72 menghitung sembilan metrik berikut untuk digunakan untuk validasi model. Saat menyetel model, pilih salah satu metrik ini untuk mengevaluasi model. Untuk daftar lengkap valideval_metricnilai, lihatParameter Tugas Pembelajaran XGBoost

Nama Metrik	Deskripsi	Arah optimalisasi
`validation:auc`	Area di bawah kurva.	Maksimalkan
`validation:error`	Tingkat kesalahan klasifikasi biner, dihitung sebagai # (kasus yang salah) /# (semua kasus).	Minimal
`validation:logloss`	Negatif log-kemungkinan.	Minimal
`validation:mae`	Berarti kesalahan absolut.	Minimal
`validation:map`	Rata-rata presisi rata-rata.	Maksimalkan
`validation:merror`	Tingkat kesalahan klasifikasi multiclass, dihitung sebagai # (kasus yang salah) /# (semua kasus).	Minimal
`validation:mlogloss`	Negatif log-kemungkinan untuk klasifikasi multiclass.	Minimal
`validation:ndcg`	Keuntungan Kumulatif Diskon Normalisasi.	Maksimalkan
`validation:rmse`	Akar berarti kesalahan kuadrat.	Minimal

Rilis XGBoost merdu 0.72 Hyperparameter

Tune model XGBoost dengan hyperparameters berikut. Hyperparameter yang memiliki efek terbesar dalam mengoptimalkan metrik evaluasi XGBoost adalah:`alpha`,`min_child_weight`,`subsample`,`eta`, dan`num_round`.

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan
`alpha`	ContinuousParameterRanges	MinValue: 0, MaxValue: 1000

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan
`colsample_bylevel`	ContinuousParameterRanges	MinValue: 0,1, MaxValue: 1
`colsample_bytree`	ContinuousParameterRanges	MinValue: 0,5, MaxValue: 1
`eta`	ContinuousParameterRanges	MinValue: 0,1, MaxValue: 0,5
`gamma`	ContinuousParameterRanges	MinValue: 0, MaxValue: 5
`lambda`	ContinuousParameterRanges	MinValue: 0, MaxValue: 1000
`max_delta_step`	IntegerParameterRanges	[0, 10]
`max_depth`	IntegerParameterRanges	[0, 10]
`min_child_weight`	ContinuousParameterRanges	MinValue: 0, MaxValue: 120
`num_round`	IntegerParameterRanges	[1, 4000]
`subsample`	ContinuousParameterRanges	MinValue: 0,5, MaxValue: 1

SageMaker Algoritma Bawaan untuk Data Teks

SageMaker menyediakan algoritma yang disesuaikan dengan analisis dokumen tekstual yang digunakan dalam pengolahan bahasa alami, klasifikasi dokumen atau ringkasan, pemodelan topik atau klasifikasi, dan transkripsi bahasa atau terjemahan.

- BlazingTextalgoritma—implementasi yang sangat optimal dari Word2vec dan algoritma klasifikasi teks yang skala ke dataset besar dengan mudah. Hal ini berguna untuk banyak tugas pemrosesan bahasa alami hilir (NLP).

- Laten Dirichlet Alokasi (LDA) Algoritma—algoritma yang cocok untuk menentukan topik dalam satu set dokumen. Ini adalah algoritme tanpa pengawasan, yang berarti tidak menggunakan data contoh dengan jawaban selama pelatihan.
- Algoritma Model Topik Saraf (NTM)—teknik lain yang tidak diawasi untuk menentukan topik dalam satu set dokumen, menggunakan pendekatan jaringan saraf.
- Algoritma Object2Vec—a tujuan umum algoritma embedding saraf yang dapat digunakan untuk sistem rekomendasi, klasifikasi dokumen, dan embeddings kalimat.
- Algoritma Urutan ke Urutan—algoritma yang diawasi yang biasa digunakan untuk terjemahan mesin saraf.
- Klasifikasi Teks - TensorFlow—algoritme yang diawasi yang mendukung pembelajaran transfer dengan model terlatih yang tersedia untuk klasifikasi teks.

Nama algoritma	Nama saluran	Mode input pelatihan	Tipe file	Kelas Instans	Paralleli zable	
BlazingTe xt	melatih	File atau Pipa	File teks (satu kalimat per baris dengan token yang dipisahkan spasi)	GPU (hanya satu instans) atau CPU	Tidak	
LDA	kereta api dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU (hanya contoh tunggal)	Tidak	
Model Topik Saraf	melatih dan (opsional) validasi, tes, atau keduanya	File atau Pipa	Recordio-protobuf atau CSV	GPU atau CPU	Ya	

Nama algoritma	Nama saluran	Mode input pelatihan	Tipe file	Kelas Instans	Paralleli zable	
Object2Ve c	melatih dan (opsional) validasi, tes, atau keduanya	File	Garis JSON	GPU atau CPU (hanya satu instans)	Tidak	
Pemodelan Seq2Seq	melatih, validasi, dan vocab	File	Recordio-protobuf	GPU (hanya contoh tunggal)	Tidak	
Klasifika si Teks - TensorFlo w	pelatihan dan validasi	File	CSV	CPU atau GPU	Ya (hanya di beberapa GPU pada satu instans)	

BlazingTextalgoritma

AmazonSageMaker BlazingTextalgoritma menyediakan implementasi yang sangat optimal dari Word2vec dan algoritma klasifikasi teks. Algoritma Word2vec berguna untuk banyak tugas pemrosesan bahasa alami hilir (NLP), seperti analisis sentimen, pengenalan entitas bernama, terjemahan mesin, dll. Klasifikasi teks adalah tugas penting untuk aplikasi yang melakukan pencarian web, pengambilan informasi, peringkat, dan klasifikasi dokumen.

Algoritma Word2vec memetakan kata-kata ke vektor terdistribusi berkualitas tinggi. Representasi vektor yang dihasilkan dari sebuah kata disebutembedding kata. Kata-kata yang serupa secara semantik sesuai dengan vektor yang berdekatan. Dengan begitu, embeddings kata menangkap hubungan semantik antar kata-kata.

Banyak aplikasi pemrosesan bahasa alami (NLP) mempelajari embeddings kata dengan melatih koleksi dokumen yang besar. Representasi vektor terlatih ini memberikan informasi tentang semantik dan distribusi kata yang biasanya meningkatkan generalisasi model lain yang kemudian dilatih

pada jumlah data yang lebih terbatas. Sebagian besar implementasi algoritma Word2vec tidak dioptimalkan untuk arsitektur CPU multi-core. Hal ini membuat sulit untuk skala ke dataset besar.

DenganBlazingTextalgoritma, Anda dapat skala ke dataset besar dengan mudah. Mirip dengan Word2vec, ia menyediakan Skip-gram dan kontinubag-of-words(CBOW) arsitektur pelatihan. BlazingTextimplementasi algoritma klasifikasi teks multi-kelas multi-label yang diawasi memperluas klasifikasi teks FastText untuk menggunakan akselerasi GPU dengan kustomCUDAkernel. Anda dapat melatih model pada lebih dari satu miliar kata dalam beberapa menit menggunakan CPU multi-core atau GPU. Dan, Anda mencapai kinerja yang setara denganstate-of-the-artalgoritma klasifikasi teks pembelajaran mendalam.

YangBlazingTextalgoritma tidak parallelizable. Untuk informasi lebih lanjut tentang parameter yang terkait dengan pelatihan, lihatJalur Registri Docker untukSageMakerAlgoritma Built-in.

YangSageMaker BlazingTextalgoritma menyediakan fitur-fitur berikut:

- Pelatihan yang dipercepat dari pengklasifikasi teks FastText pada CPU multi-core atau GPU dan Word2Vec pada GPU menggunakan kernel CUDA yang sangat optimal. Untuk informasi lebih lanjut, lihatBlazingText: Penskalaan dan Mempercepat Word2Vec menggunakan Beberapa GPU.
- Vektor Kata yang Diperkaya dengan Informasi Subworddengan mempelajari representasi vektor untuk karakter n-gram. Pendekatan ini memungkinkanBlazingTextuntuk menghasilkan vektor yang bermakna untukout-of-vocabulary(OOV) kata-kata dengan mewakili vektor mereka sebagai jumlah karakter n-gram (subword) vektor.
- SEBUAH`batch_skipgram` `mode`untuk algoritma Word2Vec yang memungkinkan pelatihan lebih cepat dan komputasi terdistribusi di beberapa node CPU. Yang`batch_skipgram` `mode`melakukan mini-batching menggunakan strategi Berbagi Sampel Negatif untuk mengubah operasi BLAS level-1 menjadi operasi BLAS level-3. Ini secara efisien memanfaatkan instruksi multiply-add arsitektur modern. Untuk informasi lebih lanjut, lihatParalelisasi Word2Vec dalam Memori Bersama dan Terdistribusi.

Untuk meringkas, mode berikut didukung olehBlazingTextpada berbagai jenis contoh:

Mode	Word2Vec (Pembelajaran Tanpa Pengawasan)	Klasifikasi Teks (Pembelajaran yang Diawasi)
Contoh CPU tunggal	`cbow`	`supervised`

Mode	Word2Vec (Pembelajaran Tanpa Pengawasan)	Klasifikasi Teks (Pembelajaran yang Diawasi)

	`Skip-gram`
	
	`Batch Skip-gram`	

Instans GPU tunggal (dengan 1 atau lebih GPU)	`cbow`
	
	`Skip-gram`
		`supervised` dengan satu GPU

Beberapa instans CPU	`Batch Skip-gram`	Tidak ada

Untuk informasi lebih lanjut tentang matematika di belakangBlazingText, lihatBlazingText: Penskalaan dan Mempercepat Word2Vec menggunakan Beberapa GPU.

Topik

- Antarmuka Input/Output untukBlazingTextAlgoritma
- Rekomendasi Instans EC2 untukBlazingTextAlgoritma
- BlazingTextContoh Notebook
- BlazingTextHiperparameter
- TuneBlazingTextModel

Antarmuka Input/Output untukBlazingTextAlgoritma

YangBlazingTextalgoritma mengharapkan file teks preprocessed tunggal dengan token spasi-dipisahkan. Setiap baris dalam file harus berisi satu kalimat. Jika Anda perlu melatih beberapa file teks, gabungkan menjadi satu file dan unggah file di saluran masing-masing.

Format Data Pelatihan dan Validasi

Format Data Pelatihan dan Validasi untuk Algoritma Word2Vec

Untuk pelatihan Word2Vec, unggah file di bawahmelatihsaluran. Tidak ada saluran lain yang didukung. File harus berisi kalimat pelatihan per baris.

Format Data Pelatihan dan Validasi untuk Algoritma Klasifikasi Teks

Untuk mode pengawasan, Anda dapat berlatih dengan mode file atau dengan format teks manifes yang ditambah.

Berlatih dengan Mode File

Untuk`supervised`mode, file pelatihan/validasi harus berisi kalimat pelatihan per baris bersama dengan label. Label adalah kata-kata yang diawali oleh string__label__. Berikut adalah contoh file pelatihan/validasi:

```
__label__4  linux ready for prime time , intel says , despite all the linux hype , the
 open-source movement has yet to make a huge splash in the desktop market . that may be
 about to change , thanks to chipmaking giant intel corp .

__label__2  bowled by the slower one again , kolkata , november 14 the past caught up
 with sourav ganguly as the indian skippers return to international cricket was short
 lived .
```

> **Note**
>
> Urutan label dalam kalimat tidak masalah.

Unggah file latihan di bawah saluran kereta api, dan unggah file validasi secara opsional di bawah saluran validasi.

Latih dengan Format Teks Manifes Augmented

Mode yang diawasi untuk instans CPU juga mendukung format manifes yang ditambah, yang memungkinkan Anda melakukan latihan dalam mode pipa tanpa perlu membuat file RecordiO. Saat menggunakan format, file manifes S3 perlu dibuat yang berisi daftar kalimat dan label yang sesuai. Format file manifes harus dalamGaris JSONformat di mana setiap baris mewakili satu sampel. Kalimat ditentukan menggunakan`source`tag dan label dapat ditentukan menggunakan`label`tag. Keduanya`source`dan`label`tag harus disediakan di bawah`AttributeNames`nilai parameter seperti yang ditentukan dalam permintaan.

```
{"source":"linux ready for prime time , intel says , despite all the linux hype",
 "label":1}
```

```
{"source":"bowled by the slower one again , kolkata , november 14 the past caught up
 with sourav ganguly", "label":2}
```

Pelatihan multi-label juga didukung dengan menentukan larik label JSON.

```
{"source":"linux ready for prime time , intel says , despite all the linux hype",
 "label": [1, 3]}
{"source":"bowled by the slower one again , kolkata , november 14 the past caught up
 with sourav ganguly", "label": [2, 4, 5]}
```

Untuk informasi selengkapnya tentang file manifes yang ditambah, lihatMenyediakan Dataset Metadata untuk Training Jobs dengan Augmented Manifest File.

Model Artefak dan Inferensi

Artefak Model untuk Algoritma Word2Vec

Untuk pelatihan Word2Vec, artefak model terdiri darivectors.txt, yang berisiwords-to-vectorspemetaan, danvectors.bin, biner yang digunakan olehBlazingTextuntuk hosting, inferensi, atau keduanya.vectors.txtmenyimpan vektor dalam format yang kompatibel dengan alat lain seperti Gensim dan Spacy. Misalnya, pengguna Gensim dapat menjalankan perintah berikut untuk memuat file vectors.txt:

```
from gensim.models import KeyedVectors
word_vectors = KeyedVectors.load_word2vec_format('vectors.txt', binary=False)
word_vectors.most_similar(positive=['woman', 'king'], negative=['man'])
word_vectors.doesnt_match("breakfast cereal dinner lunch".split())
```

Jika parameter evaluasi diatur ke`True`, file tambahan,eval.json, dibuat. File ini berisi hasil evaluasi kesamaan (menggunakan koefisien korelasi peringkat Spearman) pada dataset WS-353. Jumlah kata dari kumpulan data WS-353 yang tidak ada di korpus pelatihan dilaporkan.

Untuk permintaan inferensi, model menerima file JSON yang berisi daftar string dan mengembalikan daftar vektor. Jika kata tersebut tidak ditemukan dalam kosakata, inferensi mengembalikan vektor nol. Jika subkata diatur ke`True`selama pelatihan, model ini mampu menghasilkan vektor untukout-of-vocabulary(OOV) kata-kata.

Contoh Permintaan JSON

Jenis MIME: `application/json`

```
{
"instances": ["word1", "word2", "word3"]
}
```

Artefak Model untuk Algoritma Klasifikasi Teks

Pelatihan dengan output yang diawasi menciptakanmodel.binfile yang dapat dikonsumsi olehBlazingTexthosting. Untuk kesimpulan,BlazingTextModel menerima file JSON yang berisi daftar kalimat dan mengembalikan daftar label prediksi yang sesuai dan skor probabilitas. Setiap kalimat diharapkan menjadi string dengan token, kata, atau keduanya yang dipisahkan ruang.

Contoh Permintaan JSON

Jenis MIME: `application/json`

```
{
 "instances": ["the movie was excellent", "i did not like the plot ."]
}
```

Secara default, server hanya mengembalikan satu prediksi, yang memiliki probabilitas tertinggi. Untuk mengambil bagian ataskprediksi, Anda dapat mengaturkdalam konfigurasi, sebagai berikut:

```
{
 "instances": ["the movie was excellent", "i did not like the plot ."],
 "configuration": {"k": 2}
}
```

UntukBlazingText, yang `content-type`dan`accept`parameter harus sama. Untuk transformasi batch, keduanya harus`application/jsonlines`. Jika mereka berbeda,`Accept`bidang diabaikan. Format untuk masukan berikut:

```
content-type: application/jsonlines

{"source": "source_0"}
{"source": "source_1"}

if you need to pass the value of k for top-k, then you can do it in the following way:

{"source": "source_0", "k": 2}
```

```
{"source": "source_1", "k": 3}
```

Format untuk output berikut:

```
accept: application/jsonlines


{"prob": [prob_1], "label": ["__label__1"]}
{"prob": [prob_1], "label": ["__label__1"]}

If you have passed the value of k to be more than 1, then response will be in this
 format:

{"prob": [prob_1, prob_2], "label": ["__label__1", "__label__2"]}
{"prob": [prob_1, prob_2], "label": ["__label__1", "__label__2"]}
```

Untuk mode yang diawasi (klasifikasi teks) dan tanpa pengawasan (Word2Vec), biner (*.tempat sampah) diproduksi olehBlazingTextdapat dikonsumsi silang oleh FastText dan sebaliknya. Anda dapat menggunakan biner yang diproduksi olehBlazingTextoleh FastText. Demikian juga, Anda dapat meng-host binari model yang dibuat dengan FastText menggunakanBlazingText.

Berikut adalah contoh cara menggunakan model yang dihasilkan denganBlazingTextdengan FastText:

```

#Download the model artifact from S3
aws s3 cp s3://<YOUR_S3_BUCKET>/<PREFIX>/model.tar.gz model.tar.gz

#Unzip the model archive
tar -xzf model.tar.gz

#Use the model archive with fastText
fasttext predict ./model.bin test.txt
```

Namun, biner hanya didukung ketika pelatihan pada CPU dan GPU tunggal; pelatihan pada multi-GPU tidak akan menghasilkan biner.

Rekomendasi Instans EC2 untukBlazingTextAlgoritma

Untuk`cbow`dan`skipgram`mode,BlazingTextmendukung CPU tunggal dan instans GPU tunggal. Kedua mode ini mendukung pembelajaran`subwords`embeddings. Untuk mencapai kecepatan tertinggi tanpa mengurangi akurasi, kami sarankan Anda menggunakan instans ml.p3.2xlarge.

Untuk`batch_skipgram`modus,BlazingTextmendukung instans CPU tunggal atau beberapa. Saat melatih beberapa instance, tetapkan nilai`S3DataDistributionType`bidang`S3DataSource`objek yang Anda lewati`CreateTrainingJob`kepada`FullyReplicated`. BlazingTextmengurus mendistribusikan data di seluruh mesin.

Untuk mode klasifikasi teks yang diawasi, instans C5 direkomendasikan jika set data pelatihan kurang dari 2 GB. Untuk kumpulan data yang lebih besar, gunakan instance dengan satu GPU. BlazingTextmendukung instans P2, P3, G4dn, dan G5 untuk pelatihan dan inferensi.

BlazingTextContoh Notebook

Untuk contoh notebook yang melatih dan menyebarkanSageMaker BlazingTextalgoritma untuk menghasilkan vektor kata, lihatBelajar Word2Vec Word Representasi menggunakanBlazingText. Untuk petunjuk untuk membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh diSageMaker, lihatInstans SageMaker Notebook Amazon. Setelah membuat dan membuka instance notebook, pilihSageMakerContohtab untuk melihat daftar semuaSageMakercontoh. Notebook contoh pemodelan topik yang menggunakan Teks Berkobar terletak diPengantar algoritma Amazonbagian. Untuk membuka notebook, pilihGunakantab, lalu pilihBuat salinan.

BlazingTextHiperparameter

Ketika Anda memulai pekerjaan pelatihan dengan`CreateTrainingJob`permintaan, Anda menentukan algoritma pelatihan. Anda juga dapat menentukan hyperparameter khusus algoritma sebagaistring-to-stringpeta. Hyperparameter untukBlazingTextalgoritma tergantung pada mode yang Anda gunakan: Word2Vec (tanpa pengawasan) dan Klasifikasi Teks (diawasi).

Word2Vec Hyperparameter

Tabel berikut mencantumkan hyperparameters untukBlazingTextAlgoritma pelatihan Word2Vec disediakan oleh AmazonSageMaker.

Nama Parameter	Deskripsi
`mode`	Arsitektur Word2vec digunakan untuk pelatihan. Diperlukan Nilai valid: `batch_skipgram` , `skipgram`, atau `cbow`

Nama Parameter	Deskripsi
`batch_size`	Ukuran setiap batch saat`mode`diatur ke`batch_skipgram` . Atur ke angka antara 10 dan 20. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 11
`buckets`	Jumlah bucket hash yang digunakan untuk subwords. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 2000000
`epochs`	Jumlah lolos lengkap melalui data pelatihan. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5
`evaluation`	Apakah model terlatih dievaluasi menggunakanWordSimilarity-353 Uji. Opsional Nilai yang valid: (Boolean)`True`atau`False` Nilai default: `True`

Nama Parameter	Deskripsi
`learning_rate`	Ukuran langkah yang digunakan untuk pembaruan parameter. Opsional Nilai yang valid: Pelampung positif Nilai default: 0,05
`min_char`	Jumlah minimum karakter yang digunakan untuk subkata/karakter n-gram. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 3
`min_count`	Kata-kata yang muncul kurang dari`min_count` kali dibuang. Opsional Nilai yang valid: Bilangan bulat non-negatif Nilai default: 5
`max_char`	Jumlah maksimum karakter yang digunakan untuk subkata/karakter n-gram Opsional Nilai yang valid: bilangan bulat positif Nilai default: 6

Nama Parameter	Deskripsi
`negative_samples`	Jumlah sampel negatif untuk strategi berbagi sampel negatif. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5
`sampling_threshold`	Ambang batas untuk terjadinya kata-kata. Kata-kata yang muncul dengan frekuensi yang lebih tinggi dalam data pelatihan diambil sampel secara acak. Opsional Nilai yang valid: Fraksi positif. Rentang yang disarankan adalah (0, 1e-3] Nilai default: 0.0001
`subwords`	Apakah akan belajar subword embeddings pada tidak. Opsional Nilai yang valid: (Boolean)`True`atau`False` Nilai default: `False`
`vector_dim`	Dimensi vektor kata yang dipelajari algoritma. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 100

Nama Parameter	Deskripsi
`window_size`	Ukuran jendela konteks. Jendela konteks adalah jumlah kata yang mengelilingi kata target yang digunakan untuk pelatihan. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5

Hyperparameter Klasifikasi Teks

Tabel berikut mencantumkan hyperparameter untuk algoritma pelatihan Klasifikasi Teks yang disediakan oleh AmazonSageMaker.

> **Note**
>
> Meskipun beberapa parameter umum antara Klasifikasi Teks dan mode Word2Vec, mereka mungkin memiliki arti yang berbeda tergantung pada konteksnya.

Nama Parameter	Deskripsi
`mode`	Modus pelatihan. Diperlukan Nilai yang valid: `supervised`
`buckets`	Jumlah ember hash yang digunakan untuk kata n-gram. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 2000000

Nama Parameter	Deskripsi
`early_stopping`	Apakah akan menghentikan latihan jika akurasi validasi tidak membaik setelah`patience`jumlah zaman. Perhatikan bahwa saluran validasi diperlukan jika berhenti awal digunakan. Opsional Nilai yang valid: (Boolean)`True`atau`False` Nilai default: `False`
`epochs`	Jumlah maksimum lolos lengkap melalui data pelatihan. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5
`learning_rate`	Ukuran langkah yang digunakan untuk pembaruan parameter. Opsional Nilai yang valid: Pelampung positif Nilai default: 0,05
`min_count`	Kata-kata yang muncul kurang dari`min_count` kali dibuang. Opsional Nilai yang valid: Bilangan bulat non-negatif Nilai default: 5

Nama Parameter	Deskripsi
`min_epochs`	Jumlah minimum zaman untuk melatih sebelum logika berhenti awal dipanggil. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5
`patience`	Jumlah zaman untuk menunggu sebelum menerapkan berhenti awal ketika tidak ada kemajuan yang dibuat pada set validasi. Digunakan hanya ketika`early_stopping` adalah`True`. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 4
`vector_dim`	Dimensi lapisan embedding. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 100
`word_ngrams`	Jumlah kata n-gram fitur untuk digunakan. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 2

TuneBlazingTextModel

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada

dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung olehBlazingTextAlgoritma

YangBlazingTextAlgoritma Word2Vec (`skipgram`,`cbow`, dan`batch_skipgram`mode) melaporkan satu metrik selama pelatihan:`train:mean_rho`. Metrik ini dihitungSet data kesamaan kata WS-353. Saat menyetel nilai hyperparameter untuk algoritma Word2Vec, gunakan metrik ini sebagai tujuannya.

YangBlazingTextAlgoritma Klasifikasi Teks (`supervised`mode), juga melaporkan satu metrik selama pelatihan:`validation:accuracy`. Saat menyetel nilai hyperparameter untuk algoritma klasifikasi teks, gunakan metrik ini sebagai tujuan.

Nama Metrik	Deskripsi	Arah Optimasi
`train:mean_rho`	Rho rata-rata (koefisien korelasi peringkat Spearman) padaSet data kesamaan kata WS-353	Maksimalkan
`validation:accuracy`	Akurasi klasifikasi pada set data validasi yang ditentukan pengguna	Maksimalkan

MerduBlazingTextHiperparameter

Hyperparameter merdu untuk Algoritma Word2Vec

Tune di AmazonSageMaker BlazingTextModel Word2Vec dengan hyperparameter berikut. Hyperparameter yang memiliki dampak terbesar pada metrik objektif Word2Vec adalah:`mode`, `learning_rate`,`window_size`,`vector_dim`, dan`negative_samples`.

Nama Parameter	Jenis Parameter	Rentang atau Nilai yang Direkomendasikan
batch_size	IntegerParameterRange	[8-32]
epochs	IntegerParameterRange	[5-15]
learning_rate	ContinuousParameterRange	MinValue: 0,005,MaxValue: 0,01
min_count	IntegerParameterRange	[0-100]
mode	CategoricalParameterRange	['batch_skipgram', 'skipgram', 'cbow']
negative_samples	IntegerParameterRange	[5-25]
sampling_threshold	ContinuousParameterRange	MinValue: 0,0001,MaxValue: 0,001
vector_dim	IntegerParameterRange	[32-300]
window_size	IntegerParameterRange	[1-10]

Hyperparameter merdu untuk algoritma klasifikasi teks

Tune di AmazonSageMaker BlazingTextmodel klasifikasi teks dengan hyperparameter berikut.

Nama Parameter	Jenis Parameter	Rentang atau Nilai yang Direkomendasikan
buckets	IntegerParameterRange	[1000000-10000000]

Nama Parameter	Jenis Parameter	Rentang atau Nilai yang Direkomendasikan
`epochs`	`IntegerParameterRange`	[5-15]
`learning_rate`	`ContinuousParameterRange`	MinValue: 0,005,MaxValue: 0,01
`min_count`	`IntegerParameterRange`	[0-100]
`vector_dim`	`IntegerParameterRange`	[32-300]
`word_ngrams`	`IntegerParameterRange`	[1-3]

Laten Dirichlet Alokasi (LDA) Algoritma

AmazonSageMakerLaten Dirichlet Allocation (LDA) algoritma adalah algoritma pembelajaran tanpa pengawasan yang mencoba untuk menggambarkan satu set pengamatan sebagai campuran dari kategori yang berbeda. LDA paling sering digunakan untuk menemukan sejumlah topik yang ditentukan pengguna yang dibagikan oleh dokumen dalam corpus teks. Di sini setiap pengamatan adalah dokumen, fitur-fiturnya adalah kehadiran (atau jumlah kejadian) dari setiap kata, dan kategorinya adalah topiknya. Karena metode ini tidak diawasi, topik tidak ditentukan di depan, dan tidak dijamin untuk menyelaraskan dengan bagaimana manusia dapat secara alami mengkategorikan dokumen. Topik dipelajari sebagai distribusi probabilitas atas kata-kata yang terjadi di setiap dokumen. Setiap dokumen, pada gilirannya, digambarkan sebagai campuran topik.

Isi yang tepat dari dua dokumen dengan campuran topik serupa tidak akan sama. Tetapi secara keseluruhan, Anda akan mengharapkan dokumen-dokumen ini lebih sering menggunakan subset kata bersama, daripada jika dibandingkan dengan dokumen dari campuran topik yang berbeda. Hal ini memungkinkan LDA untuk menemukan kelompok kata ini dan menggunakannya untuk membentuk topik. Sebagai contoh yang sangat sederhana, diberikan satu set dokumen di mana satu-satunya kata yang terjadi di dalamnya adalah:makan,tidur,pementasan,meong, dankulit, LDA mungkin menghasilkan topik seperti berikut:

Topik	makan	tidur	pementasan	meong	kulit
Topik 1	0.1	0.3	0.2	0.4	0.0

Topik	makan	tidur	pementasan	meong	kulit
Topik 2	0.2	0.1	0.4	0.0	0.3

Anda dapat menyimpulkan bahwa dokumen yang lebih mungkin jatuh ke Topik 1 adalah tentang kucing (yang lebih mungkinmeongdantidur), dan dokumen yang termasuk dalam Topik 2 adalah tentang anjing-anjingnya (yang lebih sukapementasandankulit). Topik-topik ini dapat ditemukan meskipun kata-kata dog dan kucing tidak pernah muncul di salah satu teks.

Topik

- Memilih antara Alokasi Dirichlet Laten (LDA) dan Model Topik Saraf (NTM)
- Antarmuka Input/Output untuk Algoritma LDA
- Rekomendasi Instans EC2 untuk Algoritma LDA
- Notebook Sampel LDA
- Bagaimana LDA Bekerja
- Hyperparameter LDA
- Tune atau Model LDA

Memilih antara Alokasi Dirichlet Laten (LDA) dan Model Topik Saraf (NTM)

Model topik biasanya digunakan untuk menghasilkan topik dari mayat yang (1) secara koheren merangkum makna semantik dan (2) menggambarkan dokumen dengan baik. Dengan demikian, model topik bertujuan untuk meminimalkan kebingungan dan memaksimalkan koherensi topik.

Kebingungan adalah metrik evaluasi pemodelan bahasa intrinsik yang mengukur kebalikan dari rata-rata geometris per kata kemungkinan dalam data pengujian Anda. Skor kebingungan yang lebih rendah menunjukkan kinerja generalisasi yang lebih baik. Penelitian telah menunjukkan bahwa kemungkinan dihitung per kata sering tidak sesuai dengan penilaian manusia, dan dapat sepenuhnya tidak berkorelasi, sehingga koherensi topik telah diperkenalkan. Setiap topik yang disimpulkan dari model Anda terdiri dari kata-kata, dan koherensi topik dihitung ke kata-kata N teratas untuk topik tertentu dari model Anda. Ini sering didefinisikan sebagai rata-rata atau median dari skor kesamaan kata berpasangan dari kata-kata dalam topik itu misalnya, Pointwise Mutual Information (PMI). Model yang menjanjikan menghasilkan topik atau topik yang koheren dengan skor koherensi topik tinggi.

Sementara tujuannya adalah untuk melatih model topik yang meminimalkan kebingungan dan memaksimalkan koherensi topik, seringkali ada tradeoff dengan LDA dan NTM. Penelitian terbaru

oleh Amazon, Dinget et al., 2018 telah menunjukkan bahwa NTM menjanjikan untuk mencapai koherensi topik tinggi tetapi LDA dilatih dengan sampling Gibbs runtuh mencapai kebingungan yang lebih baik. Ada tradeoff antara kebingungan dan koherensi topik. Dari sudut pandang kepraktisan mengenai perangkat keras dan daya komputasi,SageMakerPerangkat keras NTM lebih fleksibel daripada LDA dan dapat menskalakan lebih baik karena NTM dapat berjalan pada CPU dan GPU dan dapat diparalelkan di beberapa instance GPU, sedangkan LDA hanya mendukung pelatihan CPU satu contoh.

Topik

Antarmuka Input/Output untuk Algoritma LDA

LDA mengharapkan data disediakan di saluran kereta api, dan secara opsional mendukung saluran uji, yang dicetak oleh model akhir. LDA mendukung keduanya`recordIO-wrapped-protobuf`(padat dan jarang) dan`CSV`format file. Untuk`CSV`, data harus padat dan memiliki dimensi yang sama denganjumlah catatan* ukuran kosakata. LDA dapat dilatih dalam mode File atau Pipa saat menggunakan protobuf yang dibungkus Recordio, tetapi hanya dalam mode File untuk`CSV`format.

Untuk kesimpulan,`text/csv`,`application/json`, dan`application/x-recordio-protobuf`jenis konten yang didukung. Data jarang juga dapat dilewatkan`application/json`dan`application/x-recordio-protobuf`. LDA inferensi kembali`application/json`atau`application/x-recordio-protobuf` prediksi, yang meliputi`topic_mixture`vektor untuk setiap pengamatan.

Silakan lihatNotebook Sampel LDAuntuk detail lebih lanjut tentang format pelatihan dan inferensi.

Rekomendasi Instans EC2 untuk Algoritma LDA

LDA saat ini hanya mendukung pelatihan CPU satu contoh. Instans CPU direkomendasikan untuk hosting/inferensi.

Notebook Sampel LDA

Untuk contoh notebook yang menunjukkan cara melatihSageMakerAlgoritma Dirichlet Alokasi laten pada dataset dan kemudian bagaimana menerapkan model terlatih untuk melakukan kesimpulan tentang campuran topik dalam dokumen masukan, lihatPengantarSageMakerLDA. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contohSageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMakerContohtab untuk melihat daftar semuaSageMakersampel. Contoh pemodelan topik notebook menggunakan algoritma NTM terletak diPengantar algoritma Amazonbagian. Untuk membuka notebook, klikGunakantab dan pilihBuat salinan.

Bagaimana LDA Bekerja

AmazonSageMakerLDA adalah algoritma pembelajaran tanpa pengawasan yang mencoba untuk menggambarkan serangkaian pengamatan sebagai campuran dari kategori yang berbeda. Kategori-kategori ini sendiri distribusi probabilitas atas fitur. LDA adalah model probabilitas generatif, yang berarti ia mencoba untuk menyediakan model untuk distribusi output dan input berdasarkan variabel laten. Ini bertentangan dengan model diskriminatif, yang mencoba mempelajari bagaimana input memetakan ke output.

Anda dapat menggunakan LDA untuk berbagai tugas, mulai dari pengelompokan pelanggan berdasarkan pembelian produk hingga analisis harmonik otomatis dalam musik. Namun, ini paling sering dikaitkan dengan pemodelan topik dalam korpus teks. Pengamatan disebut sebagai dokumen. Set fitur disebut sebagai kosakata. Sebuah fitur disebut sebagai sebuah kata. Dan kategori yang dihasilkan disebut sebagai topik.

> **Note**
>
> Lemmatisasi secara signifikan meningkatkan kinerja dan akurasi algoritma. Pertimbangkan pra-pemrosesan data teks masukan apa pun. Untuk informasi lebih lanjut, lihatStemming dan lemmatisasi.

Model LDA didefinisikan oleh dua parameter:

- α—Perkiraan sebelumnya tentang probabilitas topik (dengan kata lain, frekuensi rata-rata setiap topik dalam dokumen tertentu terjadi).

- β—kumpulan topik k di mana setiap topik diberi distribusi probabilitas atas kosakata yang digunakan dalam korpus dokumen, juga disebut “distribusi topik-kata.”

LDA adalah”bag-of-words“model, yang berarti bahwa urutan kata-kata tidak masalah. LDA adalah model generatif di mana setiap dokumen dihasilkanword-by-worddengan memilih campuran topik θ Dirichlet (α).

Untuk setiap kata dalam dokumen:

- Pilih topik z Multinomial (θ)
- Pilih distribusi topik-kata yang sesuai β_z.
- Gambarlah sebuah kata w Multinomial (β_z).

Saat melatih model, tujuannya adalah untuk menemukan parameter α dan β, yang memaksimalkan probabilitas bahwa corpus teks dihasilkan oleh model.

Metode yang paling populer untuk memperkirakan model LDA menggunakan Gibbs sampling atau Expectation Maximization (EM) teknik. AmazonSageMakerLDA menggunakan dekomposisi spektral tensor. Ini memberikan beberapa keuntungan:

- Jaminan teoritis pada hasil. Metode EM standar dijamin hanya bertemu dengan optima lokal, yang seringkali berkualitas buruk.
- Parallelizable memalukan. Pekerjaan dapat dibagi secara sepele atas dokumen masukan baik dalam pelatihan maupun inferensi. Pendekatan EM-metode dan Gibbs Sampling dapat diparalelkan, tetapi tidak semudah.
- Cepat. Meskipun metode EM memiliki biaya iterasi rendah, metode ini rentan terhadap tingkat konvergensi yang lambat. Gibbs Sampling juga tunduk pada tingkat konvergensi yang lambat dan juga membutuhkan sejumlah besar sampel.

Pada tingkat tinggi, algoritma dekomposisi tensor mengikuti proses ini:

1. Tujuannya adalah untuk menghitung dekomposisi spektralVxVxVtensor, yang merangkum saat-saat dokumen di korpus kami. Vadalah ukuran kosakata (dengan kata lain, jumlah kata yang berbeda dalam semua dokumen). Komponen spektral tensor ini adalah parameter LDA α dan β, yang memaksimalkan kemungkinan keseluruhan korpus dokumen. Namun, karena ukuran kosakata cenderung besar, iniVxVxVtensor sangat besar untuk disimpan dalam memori.

2. Sebaliknya, ia menggunakan aVxVmatriks momen, yang merupakan analog dua dimensi tensor dari langkah 1, untuk menemukan matriks pemutih dimensiVxk. Matriks ini dapat digunakan untuk mengkonversiVxVsaat matriks menjadikxkmatriks identitas.kadalah jumlah topik dalam model.
3. Matriks pemutih yang sama ini kemudian dapat digunakan untuk menemukan yang lebih kecilkxkxktensor. Saat didekomposisi secara spektral, tensor ini memiliki komponen yang memiliki hubungan sederhana dengan komponenVxVxVtensor.
4. Kotak Paling Sedikit Bergantiandigunakan untuk menguraikan yang lebih kecilkxkxktensor. Ini memberikan peningkatan substansif dalam konsumsi dan kecepatan memori. Parameter α dan β dapat ditemukan dengan “unwhitening” output ini dalam dekomposisi spektral.

Setelah parameter model LDA ditemukan, Anda dapat menemukan campuran topik untuk setiap dokumen. Anda menggunakan turunan gradien stokastik untuk memaksimalkan fungsi kemungkinan mengamati campuran topik tertentu yang sesuai dengan data ini.

Kualitas topik dapat ditingkatkan dengan meningkatkan jumlah topik yang harus dicari dalam pelatihan dan kemudian menyaring topik yang berkualitas buruk. Hal ini sebenarnya dilakukan secara otomatis diSageMakerLDA: 25% lebih banyak topik dihitung dan hanya yang dengan prior Dirichlet terkait terbesar dikembalikan. Untuk melakukan pemfilteran dan analisis topik lebih lanjut, Anda dapat meningkatkan jumlah topik dan memodifikasi model LDA yang dihasilkan sebagai berikut:

```
> import mxnet as mx
> alpha, beta = mx.ndarray.load('model.tar.gz')
> # modify alpha and beta
> mx.nd.save('new_model.tar.gz', [new_alpha, new_beta])
> # upload to S3 and create new SageMaker model using the console
```

Untuk informasi lebih lanjut tentang algoritma untuk LDA danSageMakerimplementasi, lihat yang berikut:

- Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, dan Matus Telgarsky. Dekomposisi Tensor untuk Mempelajari Model Variabel Laten, Jurnal Penelitian Pembelajaran Mesin, 15:2773 —2832, 2014.
- David M Bleu, Andrew Y Ng, dan Michael I Jordan. Alokasi Dirichlet laten. Jurnal Penelitian Pembelajaran Mesin, 3 (Jan) :993—1022, 2003.
- Thomas L Griffiths dan Mark Steyvers. Menemukan Topik Ilmiah. Prosiding National Academy of Sciences, 101 (suppl 1): 5228—5235, 2004.

- Tamara G Kolda dan Brett W Bader. Dekomposisi Tensor dan Aplikasi. Ulasan SIAM, 51 (3): 455-500, 2009.

Hyperparameter LDA

Di dalam`CreateTrainingJob`permintaan, Anda menentukan algoritma pelatihan. Anda juga dapat menentukan hyperparameter khusus algoritma sebagaistring-to-stringpeta. Tabel berikut mencantumkan hyperparameter untuk algoritma pelatihan LDA yang disediakan oleh AmazonSageMaker. Untuk informasi selengkapnya, lihat Bagaimana LDA Bekerja.

Nama Parameter	Deskripsi
`num_topics`	Jumlah topik untuk LDA untuk menemukan dalam data. Diperlukan Nilai yang valid: bilangan bulat positif
`feature_dim`	Ukuran kosakata korpus dokumen masukan. Diperlukan Nilai yang valid: bilangan bulat positif
`mini_batch_size`	Jumlah total dokumen dalam korpus dokumen masukan. Diperlukan Nilai yang valid: bilangan bulat positif
`alpha0`	Tebak awal untuk parameter konsentrasi: jumlah elemen Dirichlet sebelumnya. Nilai kecil lebih cenderung menghasilkan campuran topik yang jarang dan nilai besar (lebih besar dari 1,0) menghasilkan campuran yang lebih seragam. Opsional Nilai yang valid: Pelampung positif Nilai default: 1.0

Nama Parameter	Deskripsi
`max_restarts`	Jumlah restart untuk melakukan selama Bolak-balik Least Squares (ALS) fase dekomposisi spektral algoritma. Dapat digunakan untuk menemukan minima lokal berkualitas lebih baik dengan mengorbankan perhitungan tambahan, tetapi biasanya tidak boleh disesuaikan. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 10
`max_iterations`	Jumlah maksimum iterasi untuk melakukan selama fase ALS algoritma. Dapat digunakan untuk menemukan minima kualitas yang lebih baik dengan mengorbankan perhitungan tambahan, tetapi biasanya tidak boleh disesuaikan. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 1000
`tol`	Target toleransi kesalahan untuk fase ALS dari algoritma. Dapat digunakan untuk menemukan minima kualitas yang lebih baik dengan mengorbankan perhitungan tambahan, tetapi biasanya tidak boleh disesuaikan. Opsional Nilai yang valid: Pelampung positif Nilai default: 1e-8

Tune atau Model LDA

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada

dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

LDA adalah algoritma pemodelan topik tanpa pengawasan yang mencoba untuk menggambarkan serangkaian pengamatan (dokumen) sebagai campuran dari berbagai kategori (topik). Metrik "per kata log-probabilitas" (PWLL) mengukur kemungkinan bahwa serangkaian topik yang dipelajari (model LDA) secara akurat menggambarkan kumpulan data dokumen pengujian. Nilai PWLL yang lebih besar menunjukkan bahwa data uji lebih mungkin dijelaskan oleh model LDA.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma LDA

Algoritma LDA melaporkan satu metrik selama pelatihan:`test:pwll`. Saat menyetel model, pilih metrik ini sebagai metrik objektif.

Nama Metrik	Deskripsi	Arah Optimasi
`test:pwll`	Per-kata log-kemungkinan pada dataset pengujian. Kemungkinan bahwa dataset uji dijelaskan secara akurat oleh model LDA yang dipelajari.	Maksimalkan

Hyperparameter LDA merdu

Anda dapat menyetel hyperparameter berikut untuk algoritma LDA. Kedua hiperparameter,`alpha0`dan`num_topics`, dapat mempengaruhi metrik objektif LDA (`test:pwll`). Jika Anda belum mengetahui nilai optimal untuk hyperparameter ini, yang memaksimalkan kemungkinan log per kata dan menghasilkan model LDA yang akurat, penyetelan model otomatis dapat membantu menemukannya.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`alpha0`	ContinuousParameterRanges	MinValue: 0,1,MaxValue: 10
`num_topics`	IntegerParameterRanges	MinValue: 1,MaxValue: 150

Algoritma Model Topik Saraf (NTM)

Amazon SageMaker NTM adalah algoritma pembelajaran tanpa pengawasan yang digunakan untuk mengatur korpus dokumen ketopikyang berisi pengelompokan kata berdasarkan distribusi statistiknya. Dokumen yang sering berisi kata-kata seperti “sepeda”, “mobil”, “kereta”, “jarak tempuh”, dan “kecepatan” cenderung berbagi topik tentang “transportasi” misalnya. Pemodelan topik dapat digunakan untuk mengklasifikasikan atau meringkas dokumen berdasarkan topik yang terdeteksi atau untuk mengambil informasi atau merekomendasikan konten berdasarkan kesamaan topik. Topik dari dokumen yang dipelajari NTM dicirikan sebagai arepresentasi latenkarena topik disimpulkan dari distribusi kata yang diamati di korpus. Semantik topik biasanya disimpulkan dengan memeriksa kata-kata peringkat teratas yang dikandungnya. Karena metode ini tidak diawasi, hanya jumlah topik, bukan topik itu sendiri, yang ditentukan sebelumnya. Selain itu, topik tidak dijamin untuk menyelaraskan dengan bagaimana manusia mungkin secara alami mengkategorikan dokumen.

Pemodelan topik menyediakan cara untuk memvisualisasikan isi korpus dokumen besar dalam hal topik yang dipelajari. Dokumen yang relevan dengan setiap topik dapat diindeks atau dicari berdasarkan label topik lunak mereka. Representasi laten dokumen juga dapat digunakan untuk menemukan dokumen serupa di ruang topik. Anda juga dapat menggunakan representasi laten dokumen yang dipelajari model topik untuk masukan ke algoritma lain yang diawasi seperti pengklasifikasi dokumen. Karena representasi laten dokumen diharapkan untuk menangkap semantik dokumen yang mendasarinya, algoritma yang sebagian didasarkan pada representasi ini diharapkan berkinerja lebih baik daripada yang didasarkan pada fitur leksikal saja.

Meskipun Anda dapat menggunakan kedua Amazon SageMaker NTM dan LDA algoritma untuk pemodelan topik, mereka algoritma yang berbeda dan dapat diharapkan untuk menghasilkan hasil yang berbeda pada data input yang sama.

Untuk informasi lebih lanjut tentang matematika di belakang NTM, lihatNeural Variational Inference untuk Pemrosesan Teks.

Topik

Antarmuka Input/Output untuk Algoritma NTM

Amazon SageMaker Neural Topic Model mendukung empat saluran data: kereta api, validasi, tes, dan tambahan. Saluran data validasi, pengujian, dan tambahan bersifat opsional. Jika Anda menentukan salah satu saluran opsional ini, tetapkan nilai`S3DataDistributionType`parameter bagi mereka untuk`FullyReplicated`. Jika Anda memberikan data validasi, kehilangan data ini dicatat pada setiap zaman, dan model menghentikan pelatihan segera setelah mendeteksi bahwa kerugian validasi tidak membaik. Jika Anda tidak memberikan data validasi, algoritme berhenti lebih awal berdasarkan data pelatihan, tetapi ini bisa kurang efisien. Jika Anda memberikan data uji, algoritma melaporkan kehilangan tes dari model akhir.

Kereta, validasi, dan saluran data tes untuk NTM mendukung keduanya`recordIO-wrapped-protobuf`(padat dan jarang) dan`CSV`Format file. Untuk`CSV`Format, setiap baris harus diwakili padat dengan jumlah nol untuk kata-kata yang tidak ada dalam dokumen yang sesuai, dan memiliki dimensi sama dengan: (jumlah catatan) * (ukuran kosakata). Anda dapat menggunakan mode File atau mode Pipa untuk melatih model pada data yang diformat sebagai`recordIO-wrapped-protobuf`atau sebagai`CSV`. Saluran bantu digunakan untuk memasok file teks yang berisi kosakata. Dengan memasok file kosakata, pengguna dapat melihat kata-kata teratas untuk setiap topik yang dicetak dalam log alih-alih ID bilangan bulat mereka. Memiliki file kosakata juga memungkinkan NTM untuk menghitung skor Word Embedding Topic Coherence (WETC), metrik baru yang ditampilkan dalam log yang menangkap kesamaan di antara kata-kata teratas di setiap topik secara efektif. Klaster`ContentType`untuk saluran tambahan adalah`text/plain`, dengan setiap baris berisi satu kata, dalam urutan yang sesuai dengan ID integer yang disediakan dalam data. File kosakata harus diberi nama`vocab.txt`dan saat ini hanya pengodean UTF-8 yang didukung.

Untuk inferensi,`text/csv`,`application/json`,`application/jsonlines`, dan`application/x-recordio-protobuf`jenis konten yang didukung. Data jarang juga dapat dilewatkan`application/json`dan`application/x-recordio-protobuf`. Inferensi

NTM kembali`application/json`atau`application/x-recordio-protobuf` prediksi, yang meliputi`topic_weights`vektor untuk setiap pengamatan.

LihatPosting blogdan pendampingNotebooksuntuk detail lebih lanjut tentang penggunaan saluran bantu dan skor WETC. Untuk informasi lebih lanjut tentang cara menghitung skor WETC, lihatPemodelan Topik Saraf Sadar Koherensi-Sadar. Kami menggunakan WETC berpasangan yang dijelaskan dalam paper ini untuk Amazon SageMakerModel Topik Saraf.

Untuk informasi lebih lanjut tentang format file input dan output, lihatFormat Respons NTMuntuk kesimpulan danNotebooks NTM.

Rekomendasi Instans EC2 untuk Algoritma NTM

Pelatihan NTM mendukung jenis instans GPU dan CPU. Kami merekomendasikan instans GPU, tetapi untuk beban kerja tertentu, instans CPU dapat mengakibatkan biaya pelatihan yang lebih rendah. Instans CPU harus cukup untuk inferensi. Pelatihan NTM mendukung keluarga instan GPU P2, P3, G4dn, dan G5 untuk pelatihan dan inferensi.

Notebooks NTM

Untuk contoh notebook yang menggunakan SageMaker Algoritma NTM untuk mengungkap topik dalam dokumen dari sumber data sintetis di mana distribusi topik diketahui, lihatPengantar Fungsi Dasar NTM. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMaker Contohtab untuk melihat daftar semua SageMaker Sampel. Contoh pemodelan topik notebook menggunakan algoritma NTM terletak diPengantar algoritme AmazonBagian. Untuk membuka notebook, klikGunakantab dan pilihBuat salinan.

Hyperparameter NTM

Nama Parameter	Deskripsi
`feature_dim`	Ukuran kosakata dari set data. Wajib Nilai yang valid: Bilangan positif (min: 1, maks: 1.000.000)
`num_topics`	Jumlah topik yang diperlukan.

Nama Parameter	Deskripsi
	Wajib Nilai yang valid: Bilangan bulat positif (min: 2, maks: 1000)
`batch_norm`	Apakah akan menggunakan normalisasi batch selama pelatihan. Opsional Nilai yang benar:benarataupalsu Nilai default:palsu
`clip_gradient`	Besarnya maksimum untuk setiap komponen gradien. Opsional Nilai yang valid: Mengapung (min: 1e-3) Nilai default: Infinity
`encoder_layers`	Jumlah lapisan dalam encoder dan ukuran output setiap lapisan. Ketika diatur keauto, algoritma menggunakan dua lapisan ukuran 3 x`num_topics` dan 2 x`num_topics` masing-masing. Opsional Nilai yang valid: Daftar bilangan bulat positif yang dipisahkan dengan komaauto Nilai default:auto

Nama Parameter	Deskripsi
`encoder_layers_activation`	Fungsi aktivasi untuk digunakan dalam lapisan encoder. Opsional Nilai yang valid: • `sigmoid`:Fungsi sigmoid • `tanh`:Tangen hiperbolik • `relu`:Unit linear yang diperbaiki Nilai default: `sigmoid`
`epochs`	Jumlah maksimum melewati data pelatihan. Opsional Nilai yang valid: Bilangan bulat positif (min: 1) Nilai default: 50
`learning_rate`	Tingkat pembelajaran untuk pengoptimal. Opsional Nilai yang valid: Mengapung (min: 1e-6, maks: 1.0) Nilai default: 0.001
`mini_batch_size`	Jumlah contoh di setiap batch mini. Opsional Nilai yang valid: Bilangan bulat positif (min: 1, maks: 10000) Nilai default: 256

Nama Parameter	Deskripsi
`num_patience_epochs`	Jumlah zaman berturut-turut di mana kriteria penghentian awal dievaluasi. Awal berhenti dipicu ketika perubahan fungsi kerugian turun di bawah yang ditentukan`tolerance` dalam yang terakhir`num_patience_epochs` jumlah jangka waktu Untuk menonaktifkan penghentian awal, atur`num_patience_epochs` ke nilai yang lebih besar dari`epochs`. Opsional Nilai yang valid: Bilangan bulat positif (min: 1) Nilai default: 3
`optimizer`	Pengoptimalan yang digunakan untuk pelatihan. Opsional Nilai yang valid: • `sgd`:Keturunan gradien stokastik • `adam`:Estimasi momentum adaptif • `adagrad`:Algoritme gradien adaptif • `adadelta`:Algoritma tingkat pembelajaran adaptif • `rmsprop`:Akar berarti propagasi kuadrat Nilai default: `adadelta`
`rescale_gradient`	Faktor rescale untuk gradien. Opsional Nilai yang valid: float (min: 1e-3, max: 1.0) Nilai default: 1.0

Nama Parameter	Deskripsi
`sub_sample`	Fraksi data pelatihan untuk sampel untuk pelatihan per zaman. Opsional Nilai yang valid: Float (min: 0.0, maks: 1.0) Nilai default: 1.0
`tolerance`	Perubahan relatif maksimum dalam fungsi kerugian. Awal berhenti dipicu ketika perubahan dalam fungsi kerugian turun di bawah nilai ini dalam terakhir`num_patience_epochs` jumlah jangka waktu Opsional Nilai yang valid: Mengapung (min: 1e-6, maks: 0,1) Nilai default: 0.001
`weight_decay`	Koefisien peluruhan berat. Menambahkan regularisasi L2. Opsional Nilai yang valid: Float (min: 0.0, maks: 1.0) Nilai default: 0.0

Tune Model NTM

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Amazon SageMaker NTM adalah algoritma pembelajaran tanpa pengawasan yang mempelajari representasi laten dari koleksi besar data diskrit, seperti korpus dokumen. Representasi laten

menggunakan variabel disimpulkan yang tidak langsung diukur untuk memodelkan pengamatan dalam dataset. Penyetelan model otomatis pada NTM membantu Anda menemukan model yang meminimalkan kerugian atas data pelatihan atau validasi. Kehilangan pelatihanmengukur seberapa baik model sesuai dengan data pelatihan. Kehilangan validasimengukur seberapa baik model dapat menggeneralisasi data yang tidak dilatih. Kehilangan pelatihan yang rendah menunjukkan bahwa model cocok dengan data pelatihan. Kehilangan validasi yang rendah menunjukkan bahwa model belum terlalu sesuai dengan data pelatihan dan karenanya harus dapat memodelkan dokumen dengan sukses yang belum dilatih. Biasanya, itu lebih baik untuk memiliki kedua kerugian menjadi kecil. Namun, meminimalkan kehilangan pelatihan terlalu banyak dapat mengakibatkan overfitting dan meningkatkan kerugian validasi, yang akan mengurangi generalitas model.

Untuk informasi lebih lanjut tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma NTM

Algoritma NTM melaporkan metrik tunggal yang dihitung selama pelatihan: `validation:total_loss`. Kerugian total adalah jumlah dari kerugian rekonstruksi dan divergensi Kullback-Leibler. Saat menyetel nilai hyperparameter, pilih metrik ini sebagai tujuannya.

Nama Metrik	Deskripsi	Arah optimalisasi
`validation:total_loss`	Total Rugi pada set validasi	Meminimalkan

Hyperparameter NTM yang bisa disetel

Anda dapat menyetel hyperparameters berikut untuk algoritma NTM. Biasanya pengaturan rendah`mini_batch_size`dan kecil`learning_rate`nilai menghasilkan kerugian validasi yang lebih rendah, meskipun mungkin butuh waktu lebih lama untuk berlatih. Kerugian validasi yang rendah tidak selalu menghasilkan topik yang lebih koheren seperti yang ditafsirkan oleh manusia. Efek hyperparameter lain pada pelatihan dan kehilangan validasi dapat bervariasi dari set data ke dataset. Untuk melihat nilai mana yang kompatibel, lihatHyperparameter NTM.

Nama Parameter	Tipe Parameter	Rentang yang disarankan
`encoder_layers_activation`	CategoricalParameterRanges	['sigmoid', 'tanh', 'relu']
`learning_rate`	ContinuousParameterRange	MinValue: 1e-4, MaxValue: 0,1
`mini_batch_size`	IntegerParameterRanges	MinValue: 16, MaxValue:2048
`optimizer`	CategoricalParameterRanges	['sgd', 'adam', 'adadelta']
`rescale_gradient`	ContinuousParameterRange	MinValue: 0,1, MaxValue: 1.0
`weight_decay`	ContinuousParameterRange	MinValue: 0,0, MaxValue: 1.0

Format Respons NTM

Semua Amazon SageMaker algoritma bawaan mematuhi format inferensi input umum yang dijelaskan dalamFormat Data Umum - Inferensi. Topik ini berisi daftar format output yang tersedia untuk SageMaker Algoritme NTM.

Format Respons JSON

```
{
    "predictions":    [
        {"topic_weights": [0.02, 0.1, 0,...]},
        {"topic_weights": [0.25, 0.067, 0,...]}
    ]
}
```

Format Respons JSONLINES

```
{"topic_weights": [0.02, 0.1, 0,...]}
{"topic_weights": [0.25, 0.067, 0,...]}
```

Format Respons RECORDIO

```
[
    Record = {
        features = {},
        label = {
            'topic_weights': {
                keys: [],
                values: [0.25, 0.067, 0, ...]  # float32
            }
        }
    },
    Record = {
        features = {},
        label = {
            'topic_weights': {
                keys: [],
                values: [0.25, 0.067, 0, ...]  # float32
            }
        }
    }
]
```

Algoritma Object2Vec

Amazon SageMaker Algoritma Object2Vec adalah algoritma penyematan saraf tujuan umum yang sangat dapat disesuaikan. Ia dapat mempelajari penyematan padat dimensi rendah dari objek dimensi tinggi. Embeddings dipelajari dengan cara yang mempertahankan semantik hubungan antara pasangan objek di ruang asli di ruang embedding. Anda dapat menggunakan embeddings yang dipelajari untuk secara efisien menghitung tetangga terdekat objek dan untuk memvisualisasikan kelompok alami objek terkait dalam ruang dimensi rendah, misalnya. Anda juga dapat menggunakan embeddings sebagai fitur dari objek yang sesuai dalam tugas yang diawasi hilir, seperti klasifikasi atau regresi.

Object2Vec menggeneralisasi teknik penyematan Word2Vec yang terkenal untuk kata-kata yang dioptimalkan di SageMaker BlazingTextalgoritma. Untuk posting blog yang membahas

cara menerapkan Object2Vec ke beberapa kasus penggunaan praktis, lihatPengantar Amazon SageMakerObject2Vec.

Topik

Antarmuka I/O untuk Algoritma Object2Vec

Anda dapat menggunakan Object2Vec pada banyak tipe data input, termasuk contoh berikut.

Tipe Data	Contoh
Pasangan kalimat-kalimat	“Pertandingan sepak bola dengan banyak pria bermain.” dan “Beberapa pria sedang bermain olahraga.”
Label - pasangan urutan	Tag genre film “Titanic”, seperti “Romance” dan “Drama”, dan deskripsi singkatnya: “Titanic karya James Cameron adalah romansa epik dan penuh aksi yang bertentangan dengan pelayaran perdananya yang naas dari R.M.S. Titanic. Dia adalah kapal paling mewah di zamannya, sebuah kapal impian, yang akhirnya membawa lebih dari 1.500 orang sampai mati di perairan es dingin Atlantik Utara pada dini hari tanggal 15 April 1912.
Pasangan pelanggan-pelanggan	ID pelanggan Jane dan ID pelanggan Jackie.
Pasangan produk-produk	ID produk sepak bola dan ID produk bola basket.

Tipe Data	Contoh
Ulasan item pasangan item pengguna	ID pengguna dan barang yang dia beli, seperti apel, pir, dan jeruk.

Untuk mengubah data input ke dalam format yang didukung, Anda harus memprosesnya terlebih dahulu. Saat ini, Object2Vec secara native mendukung dua jenis input:

- Token diskrit, yang direpresentasikan sebagai daftar tunggal`integer-id`. Sebagai contoh, `[10]`.
- Urutan token diskrit, yang direpresentasikan sebagai daftar`integer-ids`. Sebagai contoh, `[0,12,10,13]`.

Objek di setiap pasangan bisa asimetris. Misalnya, pasangan dapat berupa (token, urutan) atau (token, token) atau (urutan, urutan). Untuk input token, algoritme mendukung penyematan sederhana sebagai encoder yang kompatibel. Untuk urutan vektor token, algoritme mendukung yang berikut ini sebagai encoder:

- Penyematan gabungan rata-rata
- Jaringan saraf konvolusional hierarkis (CNN),
- Memori jangka pendek panjang dua arah berlapis-lapis (BILSTMS)

Label masukan untuk setiap pasangan dapat berupa salah satu dari yang berikut:

- Label kategoris yang mengekspresikan hubungan antara objek dalam pasangan
- Skor yang mengekspresikan kekuatan kesamaan antara dua objek

Untuk label kategoris yang digunakan dalam klasifikasi, algoritme mendukung fungsi kehilangan entropi silang. Untuk label berbasis rating/skor yang digunakan dalam regresi, algoritme mendukung fungsi kerugian mean squared error (MSE). Tentukan fungsi kerugian ini dengan`output_layer`hyperparameter saat Anda membuat pekerjaan pelatihan model.

Rekomendasi Instans EC2 untuk Algoritma Object2Vec

Jenis instans Amazon Elastic Compute Cloud (Amazon EC2) yang Anda gunakan tergantung apakah Anda berlatih atau menjalankan inferensi.

Saat melatih model menggunakan algoritma Object2Vec pada CPU, mulailah dengan instance ml.m5.2xlarge. Untuk pelatihan tentang GPU, mulailah dengan instance ml.p2.xlarge. Jika pelatihan memakan waktu terlalu lama pada contoh ini, Anda dapat menggunakan instance yang lebih besar. Saat ini, algoritma Object2Vec hanya dapat melatih pada satu mesin. Namun, ia menawarkan dukungan untuk beberapa GPU. Object2Vec mendukung keluarga instance GPU P2, P3, G4dn, dan G5 untuk pelatihan dan inferensi.

Untuk inferensi dengan model Object2Vec terlatih yang memiliki jaringan saraf dalam, sebaiknya gunakan instance GPU ml.p3.2xlarge. Karena kelangkaan memori GPU,`INFERENCE_PREFERRED_MODE`variabel lingkungan dapat ditentukan untuk mengoptimalkan apakahthe section called “Optimasi GPU: Klasifikasi atau Regresi”atauthe section called “Optimasi GPU: Embeddings Encoder”jaringan inferensi dimuat ke dalam GPU.

Notebook Contoh Object2Vec

- Menggunakan Object2Vec untuk Mengkodekan Kalimat menjadi Embeddings Panjang Tetap

> **Note**
>
> Untuk menjalankan notebook pada instance notebook, lihatContoh Notebook. Untuk menjalankan notebook di Studio, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic.

Bagaimana Object2Vec Bekerja

Saat menggunakan Amazon SageMaker Algoritma Object2Vec, Anda mengikuti alur kerja standar: memproses data, melatih model, dan menghasilkan kesimpulan.

Topik

- Langkah 1: Memproses
- Langkah 2: Latih Model
- Langkah 3: Menghasilkan Inferensi

Langkah 1: Memproses

Selama preprocessing, konversikan data keGaris JSONformat file teks yang ditentukan dalamFormat Data untuk Pelatihan Object2Vec. Untuk mendapatkan akurasi tertinggi selama

pelatihan, acak juga data secara acak sebelum memasukkannya ke dalam model. Bagaimana Anda menghasilkan permutasi acak tergantung pada bahasa. Untuk python, Anda dapat menggunakan`np.random.shuffle`; untuk Unix,`shuf`.

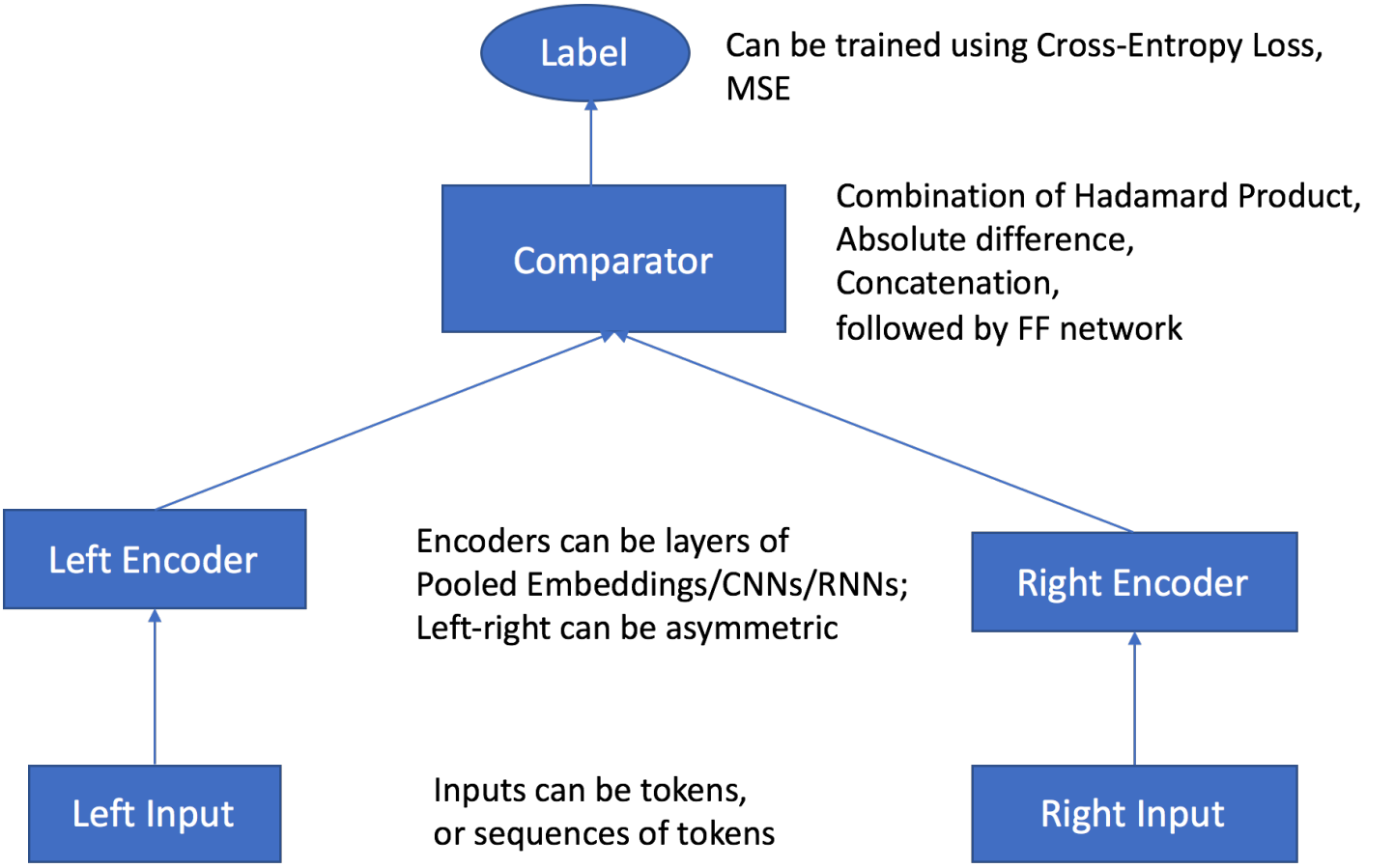
Langkah 2: Latih Model

The SageMaker Algoritma Object2Vec memiliki komponen utama berikut:

- Dua saluran input— Saluran input mengambil sepasang objek dari jenis yang sama atau berbeda sebagai input, dan meneruskannya ke encoder independen dan dapat disesuaikan.
- Dua encoder— Dua encoder, enc0 dan enc1, mengubah setiap objek menjadi vektor embedding dengan panjang tetap. Embeddings yang dikodekan dari benda-benda dalam pasangan kemudian diteruskan ke komparator.
- Komparator— Komparator membandingkan embeddings dengan cara yang berbeda dan menghasilkan skor yang menunjukkan kekuatan hubungan antara objek yang dipasangkan. Dalam skor output untuk pasangan kalimat. Misalnya, 1 menunjukkan hubungan yang kuat antara pasangan kalimat, dan 0 mewakili hubungan yang lemah.

Selama pelatihan, algoritme menerima pasangan objek dan label atau skor hubungannya sebagai input. Objek di setiap pasangan dapat dari jenis yang berbeda, seperti yang dijelaskan sebelumnya. Jika input ke kedua encoder terdiri dari unit tingkat token yang sama, Anda dapat menggunakan lapisan penyematan token bersama dengan menyetel`tied_token_embedding_weight`hyperparameter ke`True`ketika Anda membuat pekerjaan pelatihan. Ini dimungkinkan, misalnya, ketika membandingkan kalimat yang keduanya memiliki satuan tingkat token kata. Untuk menghasilkan sampel negatif pada tingkat tertentu, atur`negative_sampling_rate`hiperparameter dengan rasio sampel negatif dan positif yang diinginkan. Hiperparameter ini mempercepat pembelajaran bagaimana membedakan antara sampel positif yang diamati dalam data pelatihan dan sampel negatif yang tidak mungkin diamati.

Pasangan objek dilewatkan melalui encoder independen dan dapat disesuaikan yang kompatibel dengan jenis input objek yang sesuai. Encoder mengubah setiap objek berpasangan menjadi vektor penyematan dengan panjang tetap dengan panjang yang sama. Sepasang vektor diteruskan ke operator komparator, yang merakit vektor menjadi satu vektor menggunakan nilai yang ditentukan dalam he`comparator_list`hiperparameter. Vektor yang dirakit kemudian melewati lapisan multilayer perceptron (MLP), yang menghasilkan output yang dibandingkan fungsi kerugian dengan label yang Anda berikan. Perbandingan ini mengevaluasi kekuatan hubungan antara objek dalam pasangan seperti yang diprediksi oleh model. Gambar berikut menunjukkan alur kerja ini.



Arsitektur Algoritma Object2Vec dari Input Data ke Skor

Langkah 3: Menghasilkan Inferensi

Setelah model dilatih, Anda dapat menggunakan encoder terlatih untuk memproses objek input atau untuk melakukan dua jenis inferensi:

- Untuk mengubah objek masukan tunggal menjadi embeddings dengan panjang tetap menggunakan encoder yang sesuai
- Untuk memprediksi label hubungan atau skor antara sepasang objek masukan

Server inferensi secara otomatis mencari tahu jenis mana yang diminta berdasarkan data input. Untuk mendapatkan embeddings sebagai output, berikan hanya satu input. Untuk memprediksi label hubungan atau skor, berikan kedua input dalam pasangan.

Hiperparameter Object2Vec

Di`CreateTrainingJob`permintaan, Anda menentukan algoritma pelatihan. Anda juga dapat menentukan hyperparameters spesifik algoritme sebagai string-to-string peta. Tabel berikut mencantumkan hyperparameters untuk algoritma pelatihan Object2Vec.

Nama Parameter	Deskripsi
`enc0_max_seq_len`	Panjang urutan maksimum untuk encoder enc0. Diperlukan Nilai yang valid: 1 ≤ integer ≤ 5000
`enc0_vocab_size`	Ukuran kosakata token enc0. Diperlukan Nilai yang valid: 2 ≤ integer ≤ 3000000
`bucket_width`	Perbedaan yang diizinkan antara panjang urutan data saat bucketing diaktifkan. Untuk mengaktifkan bucketing, tentukan nilai bukan nol untuk parameter ini. Opsional Nilai yang valid: 0 ≤ integer Nilai default: 0 (tidak ada ember)
`comparator_list`	Daftar yang digunakan untuk menyesuaikan cara di mana dua embeddings dibandingkan. Lapisan operator komparato r Object2Vec mengambil pengkodean dari kedua encoder sebagai input dan output satu vektor. Vektor ini adalah rangkaian subvektor. Nilai string diteruskan ke`comparato r_list` dan urutan di mana mereka lulus menentuka n bagaimana subvektor ini dirakit. Sebagai contoh, jika`comparator_list="hadamard, concat"` , kemudian operator komparator membangun vektor dengan menggabun gkan produk Hadamard dari dua pengkodean dan penggabun

Nama Parameter	Deskripsi
	gan dua pengkodean. Jika, di sisi lain,`comparator_list="hadamard"` , maka operator komparator membangun vektor sebagai produk hadamard dari hanya dua pengkodean. Opsional Nilai yang valid: Sebuah string yang berisi kombinasi dari nama-nama dari tiga operator biner:`hadamard`,`concat`, atau`abs_diff`. Algoritma Object2Vec saat ini mensyaratkan bahwa dua pengkodean vektor memiliki dimensi yang sama. Operator ini menghasilkan subvektor sebagai berikut: • `hadamard`: Membangun vektor sebagaiProduk Hadamard (dari segi elemen)dari dua pengkodean. • `concat`: Membangun vektor sebagai rangkaian dari dua pengkodean. • `abs_diff`: Membangun vektor sebagai perbedaan mutlak antara dua pengkodean. Nilai default: `"hadamard, concat, abs_diff"`
`dropout`	Probabilitas putus sekolah untuk lapisan jaringan. Putus sekolahadalah bentuk regularisasi yang digunakan dalam jaringan saraf yang mengurangi overfitting dengan memangkas neuron kodependen. Opsional Nilai yang valid: 0.0 ≤ float ≤ 1.0 Default: 0.0

Nama Parameter	Deskripsi
`early_stopping_patience`	Jumlah zaman berturut-turut tanpa perbaikan diperbolehkan sebelum penghentian awal diterapkan. Perbaikan didefinisikan oleh dengan`early_stopping_tolerance` hiperparameter. Opsional Nilai yang valid: 1 ≤ integer Default: 3
`early_stopping_tolerance`	Pengurangan fungsi kerugian yang harus dicapai algoritma antara zaman berturut-turut untuk menghindari penghentian awal setelah jumlah zaman berturut-turut yang ditentukan dalam`early_stopping_patience` hyperparameter menyimpulkan. Opsional Nilai yang valid: 0,000001 ≤ float ≤ 0,1 Default: 0.01
`enc_dim`	Dimensi output dari lapisan embedding. Opsional Nilai yang valid: 4 ≤ bilangan bulat ≤ 10000 Default: 4096

Nama Parameter	Deskripsi
`enc0_network`	Model jaringan untuk encoder enc0. Opsional Nilai valid: `hcnn`, `bilstm`, atau `pooled_embedding` • `hcnn`: Jaringan saraf konvolusional hierarkis. • `bilstm`: Jaringan memori jangka pendek dua arah (BilsTM), di mana sinyal merambat mundur dan maju dalam waktu. Ini adalah arsitektur jaringan saraf berulang (RNN) yang sesuai untuk tugas pembelajaran berurutan. • `pooled_embedding` : Rata-rata penyematan semua token dalam input. Nilai default: `hcnn`
`enc0_cnn_filter_width`	Lebar filter encoder convolutional neural network (CNN) enc0. Bersyarat Nilai yang valid: 1 ≤ integer ≤ 9 Default: 3
`enc0_freeze_pretrained_embedding`	Apakah akan membekukan bobot penyematan enc0 yang telah dilatih sebelumnya. Bersyarat Nilai yang valid: `True` or `False` Nilai default: `True`

Nama Parameter	Deskripsi
`enc0_layers`	Jumlah lapisan dalam encoder enc0. Bersyarat Nilai yang valid:`auto`atau 1 ≤ bilangan bulat ≤ 4 • Untuk`hcnn`,`auto`berarti 4. • Untuk`bilstm`,`auto`berarti 1. • Untuk`pooled_embedding` ,`auto`mengabaikan jumlah lapisan. Nilai default: `auto`
`enc0_pretrained_embedding_file`	Nama file file embedding token enc0 yang telah dilatih sebelumnya di saluran data tambahan. Bersyarat Nilai yang valid: String dengan karakter alfanumerik, garis bawah, atau titik. [A-Z0-9\. Nilai default: “” (string kosong)
`enc0_token_embedding_dim`	Dimensi output dari layer embedding token enc0. Bersyarat Nilai yang valid: 2 ≤ bilangan bulat ≤ 1000 Default: 300

Nama Parameter	Deskripsi
`enc0_vocab_file`	File kosakata untuk memetakan vektor penyematan token enc0 yang telah dilatih sebelumnya ke ID kosakata numerik. Bersyarat Nilai yang valid: String dengan karakter alfanumerik, garis bawah, atau titik. [A-Z0-9\. Nilai default: “” (string kosong)

Nama Parameter	Deskripsi
`enc1_network`	Model jaringan untuk encoder enc1. Jika Anda ingin encoder enc1 menggunakan model jaringan yang sama dengan enc0, termasuk nilai hyperparameter, tetapkan nilainya ke`enc0`. **Note** Bahkan ketika jaringan encoder enc0 dan enc1 memiliki arsitektur simetris, Anda tidak dapat berbagi nilai parameter untuk jaringan ini. Opsional Nilai yang benar: `enc0`, `hcnn`, `bilstm`, atau `pooled_embedding` • `enc0`: Model jaringan untuk encoder enc0. • `hcnn`: Jaringan saraf konvolusional hierarkis. • `bilstm`: LSTM dua arah, di mana sinyal merambat mundur dan maju dalam waktu. Ini adalah arsitektur jaringan saraf berulang (RNN) yang sesuai untuk tugas pembelajaran berurutan. • `pooled_embedding` : Rata-rata penyematan semua token dalam input. Nilai default: `enc0`
`enc1_cnn_filter_width`	Lebar filter encoder CNN enc1. Bersyarat Nilai yang valid: 1 ≤ integer ≤ 9 Default: 3

Nama Parameter	Deskripsi
`enc1_freeze_pretrained_embedding`	Apakah akan membekukan bobot penyematan enc1 yang telah dilatih sebelumnya. Bersyarat Nilai yang valid: `True` or `False` Nilai default: `True`
`enc1_layers`	Jumlah lapisan dalam encoder enc1. Bersyarat Nilai yang valid:`auto`atau 1 ≤ bilangan bulat ≤ 4 • Untuk`hcnn`,`auto`berarti 4. • Untuk`bilstm`,`auto`berarti 1. • Untuk`pooled_embedding` ,`auto`mengabaikan jumlah lapisan. Nilai default: `auto`
`enc1_max_seq_len`	Panjang urutan maksimum untuk encoder enc1. Bersyarat Nilai yang valid: 1 ≤ integer ≤ 5000
`enc1_pretrained_embedding_file`	Nama file embedding token enc1 yang telah dilatih sebelumnya di saluran data tambahan. Bersyarat Nilai yang valid: String dengan karakter alfanumerik, garis bawah, atau titik. [A-Z0-9\. Nilai default: “” (string kosong)

Nama Parameter	Deskripsi
`enc1_token_embedding_dim`	Dimensi output dari layer embedding token enc1. Bersyarat Nilai yang valid: 2 ≤ bilangan bulat ≤ 1000 Default: 300
`enc1_vocab_file`	File kosakata untuk memetakan embeddings token enc1 yang telah dilatih sebelumnya ke ID kosakata. Bersyarat Nilai yang valid: String dengan karakter alfanumerik, garis bawah, atau titik. [A-Z0-9\. Nilai default: “” (string kosong)
`enc1_vocab_size`	Ukuran kosakata token enc0. Bersyarat Nilai yang valid: 2 ≤ integer ≤ 3000000
`epochs`	Jumlah zaman yang harus dijalankan untuk pelatihan. Opsional Nilai yang valid: 1 ≤ integer Default: 30
`learning_rate`	Tingkat pembelajaran untuk pelatihan. Opsional Nilai yang valid: 1.0E-6 ≤ float ≤ 1.0 Default: 0.0004

Nama Parameter	Deskripsi
`mini_batch_size`	Ukuran batch yang dibagi menjadi kumpulan data untuk`optimizer` selama pelatihan. Opsional Nilai yang valid: 1 ≤ integer ≤ 10000 Default: 32
`mlp_activation`	Jenis fungsi aktivasi untuk lapisan multilayer perceptron (MLP). Opsional Nilai valid: `tanh`, `relu`, atau `linear` • `tanh`: Tangen hiperbolik • `relu`: Unit linier yang diperbaiki (ReLU) • `linear`: Fungsi Lefault Nilai default: `linear`
`mlp_dim`	Dimensi output dari lapisan MLP. Opsional Nilai yang valid: 2 ≤ bilangan bulat ≤ 10000 Default: 512
`mlp_layers`	Jumlah lapisan MLP dalam jaringan. Opsional Nilai yang valid: 0 ≤ integer Nilai default: 2

Nama Parameter	Deskripsi
`negative_sampling_rate`	Rasio sampel negatif, yang dihasilkan untuk membantu melatih algoritme, dengan sampel positif yang disediakan oleh pengguna. Sampel negatif mewakili data yang tidak mungkin terjadi dalam kenyataan dan diberi label negatif untuk pelatihan . Mereka memfasilitasi pelatihan model untuk membedakan antara sampel positif yang diamati dan sampel negatif yang tidak. Untuk menentukan rasio sampel negatif terhadap sampel positif yang digunakan untuk pelatihan, tetapkan nilainya ke bilangan bulat positif. Misalnya, jika Anda melatih algoritme pada data input di mana semua sampel positif dan disetel`negative_sampling_rate` ke 2, algoritma Object2Vec secara internal menghasilkan dua sampel negatif per sampel positif. Jika Anda tidak ingin menghasilkan atau menggunakan sampel negatif selama pelatihan, atur nilainya ke 0. Opsional Nilai yang valid: 0 ≤ integer Default: 0 (off)
`num_classes`	Jumlah kelas untuk pelatihan klasifikasi. Amazon SageMaker mengabaikan hyperparameter ini untuk masalah regresi. Opsional Nilai yang valid: 2 ≤ bilangan bulat ≤ 30 Nilai default: 2

Nama Parameter	Deskripsi
`optimizer`	Jenis pengoptimal. Opsional Nilai yang valid:`adadelta`,`adagrad`,`adam`,`sgd`, atau`rmsprop`. • `adadelta`: SEBUAHmetode tingkat pembelajaran per dimensi untuk penurunan gradien • `adagrad`:algoritma gradien adaptif • `adam`:algoritma estimasi momen adaptif • `sgd`:Penurunan gradien stokastik • `rmsprop`:Perbanyakan kuadrat rata-rata akar Nilai default: `adam`
`output_layer`	Jenis lapisan keluaran tempat Anda menentukan bahwa tugasnya adalah regresi atau klasifikasi. Opsional Nilai yang valid: `softmax` atau `mean_squared_error` • `softmax`:Fungsi Softmaxdigunakan untuk klasifikasi. • `mean_squared_error` :MSEdigunakan untuk regresi. Nilai default: `softmax`

Nama Parameter	Deskripsi
`tied_token_embeddi ng_weight`	Apakah akan menggunakan layer embedding bersama untuk kedua encoder. Jika input ke kedua encoder menggunakan unit tingkat token yang sama, gunakan layer embedding token bersama. Misalnya, untuk kumpulan dokumen, jika satu encoder mengkodekan kalimat dan yang lain mengkodekan seluruh dokumen, Anda dapat menggunakan lapisan penyematan token bersama. Itu karena kalimat dan dokumen terdiri dari token kata dari kosakata yang sama. Opsional Nilai yang valid: `True` or `False` Nilai default: `False`

Nama Parameter	Deskripsi
`token_embedding_storage_type`	Mode pembaruan gradien yang digunakan selama pelatihan: ketika`dense`mode digunakan, pengoptimal menghitung matriks gradien penuh untuk lapisan penyematan token bahkan jika sebagian besar baris gradien bernilai nol. Kapan`sparse`mode digunakan, pengoptimal hanya menyimpan baris gradien yang sebenarnya digunakan dalam mini-batch. Jika Anda ingin algoritme melakukan pembaruan gradien malas, yang menghitung gradien hanya di baris bukan nol dan yang mempercepat pelatihan, tentukan`row_sparse` . Mengatur nilai ke`row_sparse` membatasi nilai yang tersedia untuk hiperpara meter lainnya, sebagai berikut: • The`optimizer` hyperparameter harus diatur ke`adam`,`adagrad`, atau`sgd`. Jika tidak, algoritme melempar a`CustomerValueError` . • Algoritma secara otomatis menonaktifkan bucketing, mengatur`bucket_width` hyperparameter ke 0. Opsional Nilai yang valid: `dense` or `row_sparse` Nilai default: `dense`
`weight_decay`	Parameter peluruhan berat yang digunakan untuk optimasi. Opsional Nilai yang valid: 0 ≤ float ≤ 10000 Nilai default: 0 (tidak ada kerusakan)

Menyetel Model Object2Vec

Tuning model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada

dataset Anda. Anda memilih hyperparameters yang dapat disetel, rentang nilai untuk masing-masing, dan metrik objektif. Untuk metrik objektif, Anda menggunakan salah satu metrik yang dihitung oleh algoritme. Penyetelan model otomatis mencari hiperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi lebih lanjut tentang tuning model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma Object2Vec

Algoritma Object2Vec memiliki metrik klasifikasi dan regresi. The`output_layer`jenis menentukan metrik mana yang dapat Anda gunakan untuk penyetelan model otomatis.

Metrik Regressor Dihitung oleh Algoritma Object2Vec

Algoritma melaporkan metrik regressor kesalahan kuadrat rata-rata, yang dihitung selama pengujian dan validasi. Saat menyetel model untuk tugas regresi, pilih metrik ini sebagai tujuannya.

Nama Metrik	Deskripsi	Arah Optimasi
`test:mean_squared_error`	Kesalahan Persegi Rata-Rata	Minimalkan
`validation:mean_squared_error`	Kesalahan Persegi Rata-Rata	Minimalkan

Metrik Klasifikasi Dihitung oleh Algoritma Object2Vec

Algoritma Object2Vec melaporkan akurasi dan metrik klasifikasi lintas entropi, yang dihitung selama pengujian dan validasi. Saat menyetel model untuk tugas klasifikasi, pilih salah satunya sebagai tujuan.

Nama Metrik	Deskripsi	Arah Optimasi
`test:accuracy`	Akurasi	Maksimalkan
`test:cross_entropy`	Default	Minimalkan

Nama Metrik	Deskripsi	Arah Optimasi
validation:accuracy	Akurasi	Maksimalkan
validation:cross_entropy	Default	Minimalkan

Hiperparameter Object2Vec yang Dapat Disetel

Anda dapat menyetel hyperparameters berikut untuk algoritma Object2Vec.

Nama Hyperparameter	Jenis Hyperparameter	Rentang dan Nilai yang Direkomendasikan	
dropout	ContinuousParameterRange	MinValue: 0,0, MaxValue:	
early_stopping_patience	IntegerParameterRange	MinValue: 1, MaxValue: 5	
early_stopping_tolerance	ContinuousParameterRange	MinValue: 0.001, MaxValue: 0.1	
enc_dim	IntegerParameterRange	MinValue: 4, MaxValue: 4096	
enc0_cnn_filter_width	IntegerParameterRange	MinValue: 1, MaxValue: 5	
enc0_layers	IntegerParameterRange	MinValue: 1, MaxValue: 4	

Nama Hyperparameter	Jenis Hyperparameter	Rentang dan Nilai yang Direkomendasikan	
`enc0_token_embedding_dim`	IntegerParameterRange	MinValue: 5, MaxValue: 300	
`enc1_cnn_filter_width`	IntegerParameterRange	MinValue: 1, MaxValue: 5	
`enc1_layers`	IntegerParameterRange	MinValue: 1, MaxValue: 4	
`enc1_token_embedding_dim`	IntegerParameterRange	MinValue: 5, MaxValue: 300	
`epochs`	IntegerParameterRange	MinValue: 4, MaxValue: 20	
`learning_rate`	ContinuousParameterRange	MinValue: 1e-6, MaxValue:	
`mini_batch_size`	IntegerParameterRange	MinValue: 1, MaxValue: 8192	
`mlp_activation`	CategoricalParameterRanges	[`tanh`, `relu`, `linear`]	
`mlp_dim`	IntegerParameterRange	MinValue: 16, MaxValue: 1024	
`mlp_layers`	IntegerParameterRange	MinValue: 1, MaxValue: 4	

Nama Hyperparameter	Jenis Hyperparameter	Rentang dan Nilai yang Direkomendasikan	
`optimizer`	CategoricalParameterRanges	[`adagrad`, `adam`, `rmsprop`, `sgd`, `adadelta`]	
`weight_decay`	ContinuousParameterRange	MinValue: 0,0, MaxValue:	

Format Data untuk Pelatihan Object2Vec

Masukan: Format Permintaan Garis JSON

Tipe konten: aplikasi/jsonlines

```
{"label": 0, "in0": [6, 17, 606, 19, 53, 67, 52, 12, 5, 10, 15, 10178, 7, 33, 652, 80,
 15, 69, 821, 4], "in1": [16, 21, 13, 45, 14, 9, 80, 59, 164, 4]}
{"label": 1, "in0": [22, 1016, 32, 13, 25, 11, 5, 64, 573, 45, 5, 80, 15, 67, 21, 7, 9,
 107, 4], "in1": [22, 32, 13, 25, 1016, 573, 3252, 4]}
{"label": 1, "in0": [774, 14, 21, 206], "in1": [21, 366, 125]}
```

“in0" dan “in1" adalah input untuk encoder0 dan encoder1, masing-masing. Format yang sama berlaku untuk masalah klasifikasi dan regresi. Untuk regresi, lapangan`"label"`dapat menerima input bernilai nyata.

Format Data untuk Inferensi Object2Vec

Optimasi GPU: Klasifikasi atau Regresi

Karena kelangkaan memori GPU,`INFERENCE_PREFERRED_MODE`variabel lingkungan dapat ditentukan untuk mengoptimalkan apakah klasifikasi/regresi atauthe section called “Keluaran: Embeddings Encoder”jaringan inferensi dimuat ke dalam GPU. Jika sebagian besar inferensi Anda adalah untuk klasifikasi atau regresi, tentukan`INFERENCE_PREFERRED_MODE=classification`. Berikut ini adalah contoh Transformasi Batch menggunakan 4 instance p3.2xlarge yang mengoptimalkan untuk inferensi klasifikasi/regresi:

```
transformer = o2v.transformer(instance_count=4,
                              instance_type="ml.p2.xlarge",
```

```
                            max_concurrent_transforms=2,
                            max_payload=1,  # 1MB
                            strategy='MultiRecord',
                            env={'INFERENCE_PREFERRED_MODE': 'classification'},  #
 only useful with GPU
                            output_path=output_s3_path)
```

Masukan: Format Permintaan Klasifikasi atau Regresi

Tipe konten: aplikasi/json

```
{
  "instances" : [
    {"in0": [6, 17, 606, 19, 53, 67, 52, 12, 5, 10, 15, 10178, 7, 33, 652, 80, 15, 69,
 821, 4], "in1": [16, 21, 13, 45, 14, 9, 80, 59, 164, 4]},
    {"in0": [22, 1016, 32, 13, 25, 11, 5, 64, 573, 45, 5, 80, 15, 67, 21, 7, 9, 107,
 4], "in1": [22, 32, 13, 25, 1016, 573, 3252, 4]},
    {"in0": [774, 14, 21, 206], "in1": [21, 366, 125]}
  ]
}
```

Tipe konten: aplikasi/jsonlines

```
{"in0": [6, 17, 606, 19, 53, 67, 52, 12, 5, 10, 15, 10178, 7, 33, 652, 80, 15, 69, 821,
 4], "in1": [16, 21, 13, 45, 14, 9, 80, 59, 164, 4]}
{"in0": [22, 1016, 32, 13, 25, 11, 5, 64, 573, 45, 5, 80, 15, 67, 21, 7, 9, 107, 4],
 "in1": [22, 32, 13, 25, 1016, 573, 3252, 4]}
{"in0": [774, 14, 21, 206], "in1": [21, 366, 125]}
```

Untuk masalah klasifikasi, panjang vektor skor sesuai dengan`num_classes`. Untuk masalah regresi, panjangnya adalah 1.

Keluaran: Klasifikasi atau Format Respons Regresi

Terima: aplikasi/json

```
{
    "predictions": [
        {
            "scores": [
                0.6533935070037842,
                0.07582679390907288,
```

```
                0.2707797586917877
            ]
        },
        {
            "scores": [
                0.026291321963071823,
                0.6577019095420837,
                0.31600672006607056
            ]
        }
    ]
}
```

Terima: aplikasi/jsonlines

```
{"scores":[0.195667684078216,0.395351558923721,0.408980727195739]}
{"scores":[0.251988261938095,0.258233487606048,0.489778339862823]}
{"scores":[0.280087798833847,0.368331134319305,0.3515810966491 69]}
```

Dalam format klasifikasi dan regresi, skor berlaku untuk masing-masing label.

Embeddings Encoder untuk Object2Vec

Optimasi GPU: Embeddings Encoder

Embedding adalah pemetaan dari objek diskrit, seperti kata-kata, ke vektor bilangan real.

Karena kelangkaan memori GPU,`INFERENCE_PREFERRED_MODE`variabel lingkungan dapat ditentukan untuk mengoptimalkan apakahthe section called “Format Inferensi: Penilaian”atau jaringan inferensi penyematan encoder dimuat ke dalam GPU. Jika sebagian besar inferensi Anda adalah untuk penyematan encoder, tentukan`INFERENCE_PREFERRED_MODE=embedding`. Berikut ini adalah contoh Transformasi Batch menggunakan 4 instance p3.2xlarge yang mengoptimalkan inferensi penyematan encoder:

```
transformer = o2v.transformer(instance_count=4,
                              instance_type="ml.p2.xlarge",
                              max_concurrent_transforms=2,
                              max_payload=1,  # 1MB
                              strategy='MultiRecord',
                              env={'INFERENCE_PREFERRED_MODE': 'embedding'},  # only
 useful with GPU
                              output_path=output_s3_path)
```

Masukan: Embeddings Encoder

<FWD-LENGTH>Tipe konten: aplikasi/json; infer_max_seqlens=, <BCK-LENGTH>

Dimana <FWD-LENGTH>dan <BCK-LENGTH>merupakan bilangan bulat dalam rentang [1.5000] dan tentukan panjang urutan maksimum untuk encoder maju dan mundur.

```
{
  "instances" : [
    {"in0": [6, 17, 606, 19, 53, 67, 52, 12, 5, 10, 15, 10178, 7, 33, 652, 80, 15, 69,
 821, 4]},
    {"in0": [22, 1016, 32, 13, 25, 11, 5, 64, 573, 45, 5, 80, 15, 67, 21, 7, 9, 107,
 4]},
    {"in0": [774, 14, 21, 206]}
  ]
}
```

<FWD-LENGTH>Tipe konten: aplikasi/jsonlines; infer_max_seqlens=, <BCK-LENGTH>

Dimana <FWD-LENGTH>dan <BCK-LENGTH>merupakan bilangan bulat dalam rentang [1.5000] dan tentukan panjang urutan maksimum untuk encoder maju dan mundur.

```
{"in0": [6, 17, 606, 19, 53, 67, 52, 12, 5, 10, 15, 10178, 7, 33, 652, 80, 15, 69, 821,
 4]}
{"in0": [22, 1016, 32, 13, 25, 11, 5, 64, 573, 45, 5, 80, 15, 67, 21, 7, 9, 107, 4]}
{"in0": [774, 14, 21, 206]}
```

Dalam kedua format ini, Anda hanya menentukan satu jenis input:“`in0`”atau“`in1.`”Layanan inferensi kemudian memanggil encoder yang sesuai dan mengeluarkan embeddings untuk setiap instance.

Keluaran: Embeddings Encoder

Tipe konten: aplikasi/json

```
{
  "predictions": [
    {"embeddings":
[0.057368703186511,0.030703511089086,0.099890425801277,0.063688032329082,0.026327300816774,0.00
    {"embeddings":
[0.150190666317939,0.05145975202322,0.098204270005226,0.064249359071254,0.056249320507049,0.015
  ]
```

```
}
```

Tipe konten: aplikasi/jsonlines

```
{"embeddings":
[0.057368703186511,0.030703511089086,0.099890425801277,0.063688032329082,0.026327300816774,0.00
{"embeddings":
[0.150190666317939,0.05145975202322,0.098204270005226,0.064249359071254,0.056249320507049,0.015
```

Panjang vektor output embeddings oleh layanan inferensi sama dengan nilai salah satu hyperparameter berikut yang Anda tentukan pada waktu pelatihan:`enc0_token_embedding_dim`,`enc1_token_embedding_dim`, atau`enc_dim`.

Algoritma Urutan ke Urutan

Amazon SageMaker Sequence to Sequence adalah algoritma pembelajaran yang diawasi di mana input adalah urutan token (misalnya, teks, audio) dan output yang dihasilkan adalah urutan token lainnya. Contoh aplikasi meliputi: terjemahan mesin (masukan kalimat dari satu bahasa dan prediksi apa kalimat itu dalam bahasa lain), ringkasan teks (masukan string kata yang lebih panjang dan prediksi string kata yang lebih pendek yang merupakan ringkasan), speech-to-text (klip audio diubah menjadi kalimat keluaran dalam token). Baru-baru ini, masalah dalam domain ini telah berhasil dimodelkan dengan jaringan saraf dalam yang menunjukkan peningkatan kinerja yang signifikan dibandingkan metodologi sebelumnya. Amazon SageMaker seq2seq menggunakan model Recurrent Neural Networks (RNNs) dan Convolutional Neural Network (CNN) dengan perhatian sebagai arsitektur encoder-decoder.

Topik

- Antarmuka Input/Output untuk Algoritma Urutan ke Urutan
- Rekomendasi Instans EC2 untuk Algoritma Urutan ke Urutan
- Notebook Sampel Urutan-ke-Urutan
- Cara Kerja Urutan-ke-Urutan
- Urutan-ke-urutan Hyperparameter
- Setel Model Urutan-ke-Urutan

Antarmuka Input/Output untuk Algoritma Urutan ke Urutan

Pelatihan

SageMaker seq2seq mengharapkan data dalam format Recordio-protobuf. Namun, token diharapkan sebagai bilangan bulat, bukan sebagai floating point, seperti biasanya.

Script untuk mengkonversi data dari file teks tokenized ke format protobuf disertakan dalamcontoh notebook seq2seq. Secara umum, ini mengemas data ke dalam tensor integer 32-bit dan menghasilkan file kosakata yang diperlukan, yang diperlukan untuk perhitungan metrik dan inferensi.

Setelah preprocessing dilakukan, algoritma dapat dipanggil untuk pelatihan. Algoritma mengharapkan tiga saluran:

- `train`: Ini harus berisi data pelatihan (misalnya,`train.rec`file yang dihasilkan oleh script preprocessing).
- `validation`: Ini harus berisi data validasi (misalnya,`val.rec`file yang dihasilkan oleh script preprocessing).
- `vocab`: Ini harus berisi dua file kosakata (`vocab.src.json`dan`vocab.trg.json`)

Jika algoritme tidak menemukan data di salah satu dari tiga saluran ini, latihan akan menghasilkan kesalahan.

Inferensi

Untuk endpoint yang di-host, inferensi mendukung dua format data. Untuk melakukan inferensi menggunakan token teks terpisah spasi, gunakan`application/json`format. Jika tidak, gunakan`recordio-protobuf`format untuk bekerja dengan integer dikodekan data. Kedua mode mendukung batching data input.`application/json`Format juga memungkinkan Anda untuk memvisualisasikan matriks perhatian.

- `application/json`: Mengharapkan input dalam format JSON dan mengembalikan output dalam format JSON. Kedua konten dan menerima jenis harus`application/json`. Setiap urutan diharapkan menjadi string dengan token terpisah spasi. Format ini direkomendasikan ketika jumlah urutan sumber dalam batch kecil. Ini juga mendukung opsi konfigurasi tambahan berikut:

  `configuration`: {`attention_matrix`:`true`}: Mengembalikan matriks perhatian untuk urutan masukan tertentu.
- `application/x-recordio-protobuf`: Mengharapkan masukan di`recordio-protobuf`memformat dan mengembalikan output dalam`recordio-protobuf format`. Kedua konten dan menerima jenis harus`applications/x-recordio-protobuf`. Untuk format ini,

urutan sumber harus diubah menjadi daftar bilangan bulat untuk pengkodean protobuf berikutnya. Format ini direkomendasikan untuk inferensi massal.

Untuk transformasi batch, inferensi mendukung format JSON Lines. Batch transform mengharapkan input dalam format JSON Lines dan mengembalikan output dalam format JSON Lines. Kedua konten dan menerima jenis harus`application/jsonlines`. Format untuk input adalah sebagai berikut:

```
content-type: application/jsonlines

{"source": "source_sequence_0"}
{"source": "source_sequence_1"}
```

Format respons adalah sebagai berikut:

```
accept: application/jsonlines

{"target": "predicted_sequence_0"}
{"target": "predicted_sequence_1"}
```

Untuk detail tambahan tentang cara membuat serial dan deserialisasi input dan output ke format tertentu untuk inferensi, lihatNotebook Sampel Urutan-ke-Urutan.

Rekomendasi Instans EC2 untuk Algoritma Urutan ke Urutan

Amazon SageMaker algoritma seq2seq hanya mendukung pada jenis instans GPU dan hanya dapat melatih pada satu mesin. Namun, Anda dapat menggunakan contoh dengan beberapa GPU. Algoritma seq2seq mendukung keluarga instan GPU P2, P3, G4dn, dan G5.

Notebook Sampel Urutan-ke-Urutan

Untuk contoh notebook yang menunjukkan cara menggunakan SageMaker Urutan algoritma Urutan untuk melatih model terjemahan Inggris-Jerman, lihatMesin Terjemahan Bahasa Inggris-Jerman Contoh Menggunakan SageMaker Seq2Seq. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, lihatlahInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMakerContohtab untuk melihat daftar semua SageMaker Sampel. Contoh pemodelan topik notebook menggunakan algoritma NTM terletak diPengantar algoritme AmazonBagian. Untuk membuka notebook, klik nyaGunakantab dan pilihBuat salinan.

Cara Kerja Urutan-ke-Urutan

Biasanya, jaringan saraf untuk sequence-to-sequence pemodelan terdiri dari beberapa lapisan, termasuk:

- Sesilapisan embedding. Dalam lapisan ini, matriks input, yang merupakan token input yang dikodekan dengan cara yang jarang (misalnya, satu-panas dikodekan) dipetakan ke lapisan fitur padat. Ini diperlukan karena vektor fitur dimensi tinggi lebih mampu mengkodekan informasi mengenai token tertentu (kata untuk teks corpora) daripada yang sederhana one-hot-encoded vektor. Ini juga merupakan praktik standar untuk menginisialisasi lapisan embedding ini dengan vektor kata pra-terlatih sepertiFastTextatauSarung Tanganatau untuk menginisialisasi secara acak dan mempelajari parameter selama pelatihan.
- SesiLapisan encoder. Setelah token input dipetakan ke dalam ruang fitur dimensi tinggi, urutan dilewatkan melalui lapisan encoder untuk mengompres semua informasi dari lapisan penyematan input (dari seluruh urutan) ke dalam vektor fitur panjang tetap. Biasanya, encoder terbuat dari jaringan tipe RNN seperti memori jangka pendek panjang (LSTM) atau unit rekuren terjaga keamanannya (GRU). (Blog Colahmenjelaskan LSTM dengan sangat rinci.)
- SEBUAHLapisan decoder. Lapisan decoder mengambil vektor fitur yang dikodekan ini dan menghasilkan urutan keluaran token. Lapisan ini juga biasanya dibangun dengan arsitektur RNN (LSTM dan GRU).

Seluruh model dilatih bersama untuk memaksimalkan probabilitas urutan target yang diberikan urutan sumber. Model ini pertama kali diperkenalkan olehSutskever dkk.pada tahun 2014.

Mekanisme perhatian. Kerugian dari kerangka encoder-decoder adalah bahwa kinerja model menurun ketika dan ketika panjang urutan sumber meningkat karena batas berapa banyak informasi yang dapat dikandung vektor fitur yang dikodekan dengan panjang tetap. Untuk mengatasi masalah ini, pada 2015, Bahdanau et al. mengusulkanMekanisme perhatian. Dalam mekanisme perhatian, decoder mencoba menemukan lokasi dalam urutan encoder di mana informasi yang paling penting dapat ditemukan dan menggunakan informasi itu dan kata-kata yang sebelumnya diterjemahkan untuk memprediksi token berikutnya dalam urutan.

Untuk detailnya, lihat whitepaperPendekatan Efektif untuk Terjemahan Mesin Neural Berbasis Perhatianoleh Luong, et al. yang menjelaskan dan menyederhanakan perhitungan untuk berbagai mekanisme perhatian. Selain itu, whitepaperSistem Terjemahan Mesin Neural Google: Menjembatani Kesenjangan antara Terjemahan Manusia dan Mesinoleh Wu, et al. menjelaskan arsitektur Google untuk terjemahan mesin, yang menggunakan koneksi lewati antara lapisan encoder dan decoder.

Urutan-ke-urutan Hyperparameter

Nama Parameter	Deskripsi
batch_size	Ukuran batch mini untuk turunan gradien. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 64
beam_size	Panjang balok untuk pencarian balok. Digunakan selama pelatihan untuk komputasi`bleu`dan digunakan selama inferensi. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5
bleu_sample_size	Jumlah instance untuk memilih dari dataset validasi untuk memecahkan kode dan menghitung`bleu`skor selama pelatihan. Set ke -1 untuk menggunakan set validasi penuh (jika`bleu`dipilih sebagai`optimized _metric`). Opsional Nilai yang benar: Integer Nilai default: 0
bucket_width	Pengembalian (sumber, target) ember hingga (`max_seq_len_source` ,`max_seq_len_target`). Sisi yang lebih panjang dari data menggunakan langkah-langkah`bucket_width` sedangkan sisi yang lebih pendek menggunakan langkah-langkah yang diperkecil oleh rasio panjang target/sumber rata-rata.

Nama Parameter	Deskripsi
	Jika satu sisi mencapai panjang maksimumnya sebelum yang lain, lebar ember ekstra di sisi itu dipasang ke sisi itu`max_len`. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 10
`bucketing_enabled`	set ke`false`untuk menonaktifkan bucketing, membuka gulungan ke panjang maksimum. Opsional Nilai yang valid: `true` or `false` Nilai default: `true`
`checkpoint_frequency_num_batches`	Checkpoint dan mengevaluasi setiap x batch. Hyperparameter checkpointing ini dilewatkan ke SageMaker Algoritma seq2seq untuk menghentikan awal dan mengambil model terbaik. Checkpointing algoritma berjalan secara lokal dalam wadah pelatihan algoritma dan tidak kompatibel dengan SageMaker Checkpointing. Algoritma untuk sementara menyimpan pos pemeriksaan ke jalur lokal dan menyimpan artefak model terbaik ke jalur keluaran model di S3 setelah pekerjaan pelatihan berhenti. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 1000

Nama Parameter	Deskripsi
`checkpoint_threshold`	Jumlah maksimum model pos pemeriksaan diperbolehkan untuk tidak meningkatkan`optimized _metric` pada set data validasi sebelum pelatihan dihentikan. Hyperparameter checkpointing ini dilewatkan ke SageMakerAlgoritma seq2seq untuk menghentikan awal dan mengambil model terbaik. Checkpointing algoritma berjalan secara lokal dalam wadah pelatihan algoritma dan tidak kompatibel dengan SageMaker Checkpointing. Algoritma untuk sementara menyimpan pos pemeriksaan ke jalur lokal dan menyimpan artefak model terbaik ke jalur keluaran model di S3 setelah pekerjaan pelatihan berhenti. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 3
`clip_gradient`	Klip nilai gradien absolut lebih besar dari ini. Atur ke negatif untuk menonaktifkan. Opsional Nilai yang benar: float Nilai default: 1
`cnn_activation_type`	Klastercnnjenis aktivasi yang akan digunakan. Opsional Nilai yang valid: String. Salah satu`glu`,`relu`,`softrelu`,`sigmoid`, atau`tanh`. Nilai default: `glu`

Nama Parameter	Deskripsi
`cnn_hidden_dropout`	Probabilitas putus sekolah untuk putus sekolah di antara lapisan konvolusional. Opsional Nilai yang valid: FLOAT Rentang dalam [0,1]. Nilai default: 0
`cnn_kernel_width_decoder`	Lebar kernel untuk`cnn`dekoder. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5
`cnn_kernel_width_encoder`	Lebar kernel untuk`cnn`pembuat kode. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 3
`cnn_num_hidden`	Jumlah`cnn`unit tersembunyi untuk encoder dan decoder. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 512
`decoder_type`	Jenis decoder. Opsional Nilai yang valid: String. Baik `rnn` atau `cnn`. Nilai default:`rnn`

Nama Parameter	Deskripsi
`embed_dropout_source`	Probabilitas putus sekolah untuk embeddings sisi sumber. Opsional Nilai yang valid: FLOAT Rentang dalam [0,1]. Nilai default: 0
`embed_dropout_target`	Probabilitas putus sekolah untuk embeddings sisi target. Opsional Nilai yang valid: FLOAT Rentang dalam [0,1]. Nilai default: 0
`encoder_type`	Jenis encoder. Klaster`rnn`narsitektur didasarkan pada mekanisme perhatian oleh Bahdanau et al. dancnnarsitektur didasarkan pada Gehring et al. Opsional Nilai yang valid: String. Baik `rnn` atau `cnn`. Nilai default: `rnn`
`fixed_rate_lr_half_life`	Setengah hidup untuk tingkat pembelajaran dalam hal jumlah pos pemeriksaan untuk`fixed_rate_ *` Penjadwal. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 10

Nama Parameter	Deskripsi
`learning_rate`	Tingkat pembelajaran awal. Opsional Nilai yang benar: float Nilai default: 0.0003
`loss_type`	Fungsi kerugian untuk pelatihan. Opsional Nilai yang valid: String.`cross-entropy` Nilai default: `cross-entropy`
`lr_scheduler_type`	Jenis penjadwal tingkat pembelajaran.`plateau_reduce` berarti mengurangi tingkat pembelajaran kapan punoptimized_metric di atas`validation_accuracy` dataran tinggi.`inv_t`adalah pembusukan waktu terbalik.`learning_rate` /(1+`decay_rate` *t) Opsional Nilai yang valid: String. Salah satu`plateau_reduce` ,`fixed_rate_inv_t` , atau`fixed_rate_inv_sqrt_t` . Nilai default: `plateau_reduce`
`max_num_batches`	Jumlah maksimum update/batch untuk diproses. -1 untuk tak terbatas. Opsional Nilai yang benar: Integer Nilai default: -1

Nama Parameter	Deskripsi
`max_num_epochs`	Jumlah maksimum zaman untuk melewati data pelatihan sebelum pemasangan dihentikan. Pelatihan berlanjut hingga jumlah zaman ini bahkan jika akurasi validasi tidak membaik jika parameter ini dilewatkan. Diabaikan jika tidak dilewati. Opsional Nilai yang valid: Bilangan bulat positif dan kurang dari atau sama dengan max_num_epochs. Nilai default: tidak ada.
`max_seq_len_source`	Panjang maksimum untuk panjang urutan sumber. Urutan yang lebih panjang dari panjang ini dipotong hingga panjang ini. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 100
`max_seq_len_target`	Panjang maksimum untuk panjang urutan target. Urutan yang lebih panjang dari panjang ini dipotong hingga panjang ini. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 100

Nama Parameter	Deskripsi
`min_num_epochs`	Jumlah minimum zaman pelatihan harus dijalankan sebelum dihentikan melalui`early_stopping` syarat. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 0
`momentum`	Konstanta momentum digunakan untuk`sgd`. Jangan lewati parameter ini jika Anda menggunak an`adam`ataurmsprop. Opsional Nilai yang benar: float Nilai default: tidak ada.
`num_embed_source`	Menyematkan ukuran untuk token sumber. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 512
`num_embed_target`	Embedding ukuran untuk token target. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 512

Nama Parameter	Deskripsi
num_layers_decoder	Jumlah lapisan untuk Decoderrnnataucnn. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 1
num_layers_encoder	Jumlah lapisan untuk Encoderrnnataucnn. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 1
optimized_metric	Metrik untuk dioptimalkan dengan berhenti lebih awal. Opsional Nilai yang valid: String. Salah satuperplexit y ,accuracy, ataubleu. Nilai default: perplexity
optimizer_type	Optimizer untuk memilih dari. Opsional Nilai yang valid: String. Salah satuadam,sgd, ataurmsprop. Nilai default: adam

Nama Parameter	Deskripsi
`plateau_reduce_lr_factor`	Faktor untuk melipatgandakan tingkat pembelajaran dengan (untuk`plateau_reduce`). Opsional Nilai yang benar: float Nilai default: 0,5
`plateau_reduce_lr_threshold`	Untuk`plateau_reduce` scheduler, kalikan tingkat pembelajaran dengan mengurangi faktor jika`optimized_metric` Tidak membaik untuk banyak pos pemeriksaan ini. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 3
`rnn_attention_in_upper_layers`	Lulus perhatian ke lapisan atas`rnn`, LIKEpaper Google NMT. Hanya berlaku jika lebih dari satu lapisan digunakan. Opsional Nilai yang benar: boolean (`true`atau`false`) Nilai default: `true`
`rnn_attention_num_hidden`	Jumlah unit tersembunyi untuk lapisan perhatian. default`rnn_num_hidden` . Opsional Nilai yang valid: bilangan bulat positif Nilai default: `rnn_num_hidden`

Nama Parameter	Deskripsi
`rnn_attention_type`	Model perhatian untuk encoders.`mlp`mengacu pada concat dan bilinear mengacu pada umum dari Luong et al. paper. Opsional Nilai yang valid: String. Salah satu`dot`,`fixed`,`mlp`, atau`bilinear`. Nilai default: `mlp`
`rnn_cell_type`	Jenis spesifik`rnn`narsitektur. Opsional Nilai yang valid: String. Baik `lstm` atau `gru`. Nilai default: `lstm`
`rnn_decoder_state_init`	Cara menginisialisasi`rnn`decoder menyatakan dari encoders. Opsional Nilai yang valid: String. Salah satu`last`,`avg`, atau`zero`. Nilai default: `last`
`rnn_first_residual_layer`	FIRST`rnn`lapisan untuk memiliki koneksi sisa, hanya berlaku jika jumlah lapisan dalam encoder atau decoder lebih dari 1. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 2

Nama Parameter	Deskripsi
`rnn_num_hidden`	Jumlahrnnunit tersembunyi untuk encoder dan decoder. Ini harus berupa kelipatan 2 karena algoritme menggunakan Memori Jangka Panjang Jangka Panjang (LSTM) dua arah secara default. Opsional Nilai yang valid: bilangan bulat genap positif Nilai default: 1024
`rnn_residual_connections`	Tambahkan koneksi sisa ke ditumpukrnn. Jumlah lapisan harus lebih dari 1. Opsional Nilai yang benar: boolean (`true`ataufalse`)  Nilai default: `false`
`rnn_decoder_hidden_dropout`	Probabilitas putus sekolah untuk keadaan tersembun yi yang menggabungkan konteks denganrnnkeadaan tersembunyi di decoder. Opsional Nilai yang valid: FLOAT Rentang dalam [0,1]. Nilai default: 0
`training_metric`	Metrik untuk melacak pelatihan tentang data validasi. Opsional Nilai yang valid: String. Baik `perplexity` atau `accuracy`. Nilai default: `perplexity`

Nama Parameter	Deskripsi
`weight_decay`	Berat pembusukan konstan. Opsional Nilai yang benar: float Nilai default: 0
`weight_init_scale`	Skala inisialisasi berat (untuk`uniform`dan`xavier`inisialisasi). Opsional Nilai yang benar: float Nilai default: 2.34
`weight_init_type`	Jenis inisialisasi berat. Opsional Nilai yang valid: String. Baik `uniform` atau `xavier`. Nilai default: `xavier`
`xavier_factor_type`	Tipe faktor Xavier. Opsional Nilai yang valid: String. Salah satu`in`,`out`, atau`avg`. Nilai default: `in`

Setel Model Urutan-ke-Urutan

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis

mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi lebih lanjut tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma Urutan-ke-Urutan

Algoritma urutan ke urutan melaporkan tiga metrik yang dihitung selama pelatihan. Pilih salah satunya sebagai tujuan untuk dioptimalkan saat menyetel nilai hyperparameter.

Nama Metrik	Deskripsi	Arah optimalisasi
`validation:accuracy`	Akurasi dihitung pada dataset validasi.	Maksimalkan
`validation:bleu`	Bleuskor dihitung pada dataset validasi. Karena perhitungan BLEU mahal, Anda dapat memilih untuk menghitung BLEU pada subsampel acak dari set data validasi untuk mempercep at proses pelatihan secara keseluruhan. Gunakan`bleu_sample_size` parameter untuk menentukan subsampel.	Maksimalkan
`validation:perplexity`	Kebingungan, adalah fungsi kerugian dihitung pada dataset validasi. Kebingungan mengukur entropi silang antara sampel empiris dan distribusi yang diprediksi oleh model sehingga memberikan ukuran seberapa baik model memprediksi nilai sampel, Model yang pandai memprediksi sampel memiliki kebingungan yang rendah.	Minimalkan

Hyperparameter Urutan-ke-Urutan Merdu

Anda dapat menyetel hyperparameter berikut untuk SageMaker Urutan algoritma Urutan. Hyperparameter yang memiliki dampak terbesar pada urutan urutan metrik objektif

adalah:`batch_size`,`optimizer_type`,`learning_rate`,`num_layers_encoder`, dan`num_layers_decoder`.

Nama Parameter	Tipe Parameter	Rentang Direkomendasikan
`num_layers_encoder`	IntegerParameterRange	[1-10]
`num_layers_decoder`	IntegerParameterRange	[1-10]
`batch_size`	CategoricalParameterRange	[16,32,64,128,256,512,1024,2048]
`optimizer_type`	CategoricalParameterRange	['adam', 'sgd', 'rmsprop']
`weight_init_type`	CategoricalParameterRange	['xavier', 'seragam']
`weight_init_scale`	ContinuousParameterRange	Untuk tipe xavier: MinValue: 2.0, MaxValue: 3.0 Untuk tipe seragam: MinValue: -1.0, MaxValue: 1,0
`learning_rate`	ContinuousParameterRange	MinValue: 0.00005, MaxValue: 0.2
`weight_decay`	ContinuousParameterRange	MinValue: 0,0, MaxValue: 0,1
`momentum`	ContinuousParameterRange	MinValue: 0,5, MaxValue: 0,9
`clip_gradient`	ContinuousParameterRange	MinValue: 1.0, MaxValue: 5.0

Nama Parameter	Tipe Parameter	Rentang Direkomendasikan
rnn_num_hidden	CategoricalParameterRange	Berlaku hanya untuk jaringan saraf berulang (RNN). [128,256,512,1024, 2048]
cnn_num_hidden	CategoricalParameterRange	Berlaku hanya untuk jaringan saraf konvolusional (CNN). [128,256,512,1024, 2048]
num_embed_source	IntegerParameterRange	[256-512]
num_embed_target	IntegerParameterRange	[256-512]
embed_dropout_source	ContinuousParameterRange	MinValue: 0,0, MaxValue: 0,5
embed_dropout_target	ContinuousParameterRange	MinValue: 0,0, MaxValue: 0,5
rnn_decoder_hidden_dropout	ContinuousParameterRange	MinValue: 0,0, MaxValue: 0,5
cnn_hidden_dropout	ContinuousParameterRange	MinValue: 0,0, MaxValue: 0,5
lr_scheduler_type	CategoricalParameterRange	['plateau_reduce', 'fixed_rate_inv_t', 'fixed_rate_inv_sqrt_t']

Nama Parameter	Tipe Parameter	Rentang Direkomendasikan
`plateau_reduce_lr_factor`	ContinuousParameterRange	MinValue: 0,1, MaxValue: 0,5
`plateau_reduce_lr_threshold`	IntegerParameterRange	[1-5]
`fixed_rate_lr_half_life`	IntegerParameterRange	[10-30]

Klasifikasi Teks - TensorFlow

Klasifikasi SageMaker Teks Amazon - TensorFlow algoritma adalah algoritma pembelajaran yang diawasi yang mendukung pembelajaran transfer dengan banyak model yang telah dilatih sebelumnya dari Hub. TensorFlow Gunakan pembelajaran transfer untuk menyempurnakan salah satu model terlatih yang tersedia pada kumpulan data Anda sendiri, meskipun sejumlah besar data teks tidak tersedia. Algoritma klasifikasi teks mengambil string teks sebagai input dan menghasilkan probabilitas untuk masing-masing label kelas. Kumpulan data pelatihan harus dalam format CSV.

Topik

- Cara menggunakan Klasifikasi SageMaker Teks - TensorFlow Algoritma
- Antarmuka input dan output untuk Klasifikasi Teks - TensorFlow algoritma
- Rekomendasi instans Amazon EC2 untuk Klasifikasi Teks - algoritma TensorFlow
- Klasifikasi Teks - TensorFlow contoh buku catatan
- Bagaimana Klasifikasi Teks - TensorFlow Bekerja
- TensorFlow Model Hub
- Klasifikasi Teks - TensorFlow Hyperparameters
- Menyetel Klasifikasi Teks - TensorFlow model

Cara menggunakan Klasifikasi SageMaker Teks - TensorFlow Algoritma

Anda dapat menggunakan Klasifikasi Teks - TensorFlow sebagai algoritma SageMaker bawaan Amazon. Bagian berikut menjelaskan cara menggunakan Klasifikasi Teks - TensorFlow dengan SageMaker Python SDK. Untuk informasi tentang cara menggunakan Klasifikasi Teks - TensorFlow dari Amazon SageMaker Studio Classic UI, lihatSageMaker JumpStart.

Klasifikasi Teks - TensorFlow algoritma mendukung pembelajaran transfer menggunakan salah satu TensorFlow model pra-terlatih yang kompatibel. Untuk daftar semua model terlatih yang tersedia, lihatTensorFlow Model Hub. Setiap model yang telah dilatih sebelumnya memiliki keunikan`model_id`. Contoh berikut menggunakan BERT Base Uncased (`model_id`:`tensorflow-tc-bert-en-uncased-L-12-H-768-A-12-2`) untuk menyempurnakan dataset kustom. Model yang telah dilatih sebelumnya semuanya telah diunduh sebelumnya dari TensorFlow Hub dan disimpan dalam bucket Amazon S3 sehingga pekerjaan pelatihan dapat berjalan dalam isolasi jaringan. Gunakan artefak pelatihan model yang dibuat sebelumnya ini untuk membangun Estimator. SageMaker

Pertama, ambil URI image Docker, URI skrip pelatihan, dan URI model yang telah dilatih sebelumnya. Kemudian, ubah hyperparameters sesuai keinginan Anda. Anda dapat melihat kamus Python dari semua hyperparameters yang tersedia dan nilai defaultnya dengan. `hyperparameters.retrieve_default` Untuk informasi selengkapnya, lihat Klasifikasi Teks - TensorFlow Hyperparameters. Gunakan nilai-nilai ini untuk membangun SageMaker Estimator.

> **Note**
>
> Nilai hyperparameter default berbeda untuk model yang berbeda. Misalnya, untuk model yang lebih besar, ukuran batch default lebih kecil.

Contoh ini menggunakan SST2kumpulan data, yang berisi ulasan film positif dan negatif. Kami mengunduh dataset sebelumnya dan membuatnya tersedia dengan Amazon S3. Untuk menyempurnakan model Anda, hubungi `.fit` menggunakan lokasi Amazon S3 dari kumpulan data pelatihan Anda. Bucket S3 apa pun yang digunakan dalam notebook harus berada di AWS Region yang sama dengan instance notebook yang mengaksesnya.

```
from sagemaker import image_uris, model_uris, script_uris, hyperparameters
from sagemaker.estimator import Estimator

model_id, model_version = "tensorflow-tc-bert-en-uncased-L-12-H-768-A-12-2", "*"
```

```
training_instance_type = "ml.p3.2xlarge"

# Retrieve the Docker image
train_image_uri =
 image_uris.retrieve(model_id=model_id,model_version=model_version,image_scope="training",insta

# Retrieve the training script
train_source_uri = script_uris.retrieve(model_id=model_id, model_version=model_version,
 script_scope="training")

# Retrieve the pretrained model tarball for transfer learning
train_model_uri = model_uris.retrieve(model_id=model_id, model_version=model_version,
 model_scope="training")

# Retrieve the default hyperparameters for fine-tuning the model
hyperparameters = hyperparameters.retrieve_default(model_id=model_id,
 model_version=model_version)

# [Optional] Override default hyperparameters with custom values
hyperparameters["epochs"] = "5"

# Sample training data is available in this bucket
training_data_bucket = f"jumpstart-cache-prod-{aws_region}"
training_data_prefix = "training-datasets/SST2/"

training_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}"

output_bucket = sess.default_bucket()
output_prefix = "jumpstart-example-tc-training"
s3_output_location = f"s3://{output_bucket}/{output_prefix}/output"

# Create an Estimator instance
tf_tc_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1,
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location,
)
```

```
# Launch a training job
tf_tc_estimator.fit({"training": training_dataset_s3_path}, logs=True)
```

Untuk informasi selengkapnya tentang cara menggunakan TensorFlow algoritma Klasifikasi SageMaker Teks - untuk pembelajaran transfer pada kumpulan data kustom, lihat buku catatan Pengantar JumpStart - Klasifikasi Teks.

Antarmuka input dan output untuk Klasifikasi Teks - TensorFlow algoritma

Masing-masing model terlatih yang tercantum dalam Model TensorFlow Hub dapat disetel dengan baik ke kumpulan data apa pun yang terdiri dari kalimat teks dengan sejumlah kelas. Model yang telah dilatih sebelumnya melampirkan lapisan klasifikasi ke model Text Embedding dan menginisialisasi parameter lapisan ke nilai acak. Dimensi output dari lapisan klasifikasi ditentukan berdasarkan jumlah kelas yang terdeteksi dalam data input.

Perhatikan cara memformat data pelatihan Anda untuk masukan ke TensorFlow model Klasifikasi Teks.

- Format input data pelatihan: Direktori yang berisi `data.csv` file. Setiap baris kolom pertama harus memiliki label kelas integer antara 0 dan jumlah kelas. Setiap baris kolom kedua harus memiliki data teks yang sesuai.

Berikut ini adalah contoh file CSV input. Perhatikan bahwa file tersebut seharusnya tidak memiliki header apa pun. File harus di-host di bucket Amazon S3 dengan jalur yang mirip dengan berikut ini:. `s3://`*`bucket_name`*`/`*`input_directory`*`/` Perhatikan bahwa trailing `/` diperlukan.

```
|   |  |
|---|---|
|0 |hide new secretions from the parental units|
|0 |contains no wit , only labored gags|
|1 |that loves its characters and communicates something rather beautiful about human
 nature|
|...|...|
```

Pelatihan tambahan

Anda dapat menyemai pelatihan model baru dengan artefak dari model yang Anda latih sebelumnya. SageMaker Pelatihan tambahan menghemat waktu pelatihan ketika Anda ingin melatih model baru dengan data yang sama atau serupa.

> **Note**
>
> Anda hanya dapat menyemai model Klasifikasi SageMaker Teks - TensorFlow model dengan Klasifikasi Teks lain - TensorFlow model yang dilatih SageMaker.

Anda dapat menggunakan kumpulan data apa pun untuk pelatihan tambahan, selama kumpulan kelas tetap sama. Langkah pelatihan inkremental mirip dengan langkah fine-tuning, tetapi alih-alih memulai dengan model yang telah dilatih sebelumnya, Anda mulai dengan model fine-tuning yang ada.

Untuk informasi selengkapnya tentang penggunaan pelatihan tambahan dengan TensorFlow algoritma Klasifikasi SageMaker Teks -, lihat buku catatan contoh Pengantar JumpStart - Klasifikasi Teks.

Inferensi dengan Klasifikasi Teks - algoritma TensorFlow

Anda dapat meng-host model yang disetel dengan baik yang dihasilkan dari pelatihan Klasifikasi TensorFlow Teks Anda untuk inferensi. Setiap format teks mentah untuk inferensi harus tipe `application/x-text` konten.

Menjalankan inferensi menghasilkan nilai probabilitas, label kelas untuk semua kelas, dan label prediksi yang sesuai dengan indeks kelas dengan probabilitas tertinggi yang dikodekan dalam format JSON. Klasifikasi Teks - TensorFlow model memproses satu string per permintaan dan hanya menghasilkan satu baris. Berikut ini adalah contoh respons format JSON:

```
accept: application/json;verbose

{"probabilities": [prob_0, prob_1, prob_2, ...],
"labels": [label_0, label_1, label_2, ...],
"predicted_label": predicted_label}
```

Jika `accept` diatur ke`application/json`, maka model hanya menghasilkan probabilitas.

Rekomendasi instans Amazon EC2 untuk Klasifikasi Teks - algoritma TensorFlow

Klasifikasi Teks - TensorFlow algoritma mendukung semua instans CPU dan GPU untuk pelatihan, termasuk:

- `ml.p2.xlarge`

- `ml.p2.16xlarge`
- `ml.p3.2xlarge`
- `ml.p3.16xlarge`
- `ml.g4dn.xlarge`
- `ml.g4dn.16.xlarge`
- `ml.g5.xlarge`
- `ml.g5.48xlarge`

Kami merekomendasikan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Instance CPU (seperti M5) dan GPU (P2, P3, G4dn, atau G5) dapat digunakan untuk inferensi. Untuk daftar lengkap instans SageMaker pelatihan dan inferensi di seluruh AWS Wilayah, lihat Harga Amazon SageMaker .

Klasifikasi Teks - TensorFlow contoh buku catatan

Untuk informasi selengkapnya tentang cara menggunakan TensorFlow algoritma Klasifikasi SageMaker Teks - untuk pembelajaran transfer pada kumpulan data kustom, lihat buku catatan Pengantar JumpStart - Klasifikasi Teks.

Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Bagaimana Klasifikasi Teks - TensorFlow Bekerja

Klasifikasi Teks - TensorFlow algoritma mengambil teks sebagai mengklasifikasikannya ke dalam salah satu label kelas output. Jaringan pembelajaran mendalam seperti BERT sangat akurat untuk klasifikasi teks. Ada juga jaringan pembelajaran mendalam yang dilatih pada kumpulan data teks besar, seperti TextNet, yang memiliki lebih dari 11 juta teks dengan sekitar 11.000 kategori. Setelah jaringan dilatih dengan TextNet data, Anda kemudian dapat menyempurnakan jaringan pada kumpulan data dengan fokus khusus untuk melakukan tugas klasifikasi teks yang lebih spesifik. Klasifikasi SageMaker Teks Amazon - TensorFlow algoritma mendukung pembelajaran transfer pada banyak model yang telah dilatih sebelumnya yang tersedia di TensorFlow Hub.

Menurut jumlah label kelas dalam data pelatihan Anda, lapisan klasifikasi teks dilampirkan ke TensorFlow model pilihan Anda yang telah dilatih sebelumnya. Lapisan klasifikasi terdiri dari lapisan putus sekolah, lapisan padat, dan lapisan yang terhubung penuh dengan regularisasi 2-norma, dan

diinisialisasi dengan bobot acak. Anda dapat mengubah nilai hyperparameter untuk tingkat putus sekolah dari lapisan putus sekolah dan faktor regularisasi L2 untuk lapisan padat.

Anda dapat menyempurnakan seluruh jaringan (termasuk model yang telah dilatih sebelumnya) atau hanya lapisan klasifikasi teratas pada data pelatihan baru. Dengan metode pembelajaran transfer ini, pelatihan dengan kumpulan data yang lebih kecil dimungkinkan.

TensorFlow Model Hub

Model pra-terlatih berikut tersedia untuk digunakan untuk pembelajaran transfer dengan TensorFlow algoritma Klasifikasi Teks.

Model berikut bervariasi secara signifikan dalam ukuran, jumlah parameter model, waktu pelatihan, dan latensi inferensi untuk setiap kumpulan data tertentu. Model terbaik untuk kasus penggunaan Anda bergantung pada kompleksitas kumpulan data fine-tuning Anda dan persyaratan apa pun yang Anda miliki pada waktu pelatihan, latensi inferensi, atau akurasi model.

Nama Model	**`model_id`**	Sumber
Basis BERT Tidak Terkisi	`tensorflow-tc-bert-en-uncased-L-12-H-768-A-12-2`	TensorFlow Tautan hub
Bert Base Cased	`tensorflow-tc-bert-en-cased-L-12-H-768-A-12-2`	TensorFlow Tautan hub
Bert Basis Multilingual Cased	`tensorflow-tc-bert-multi-cased-L-12-H-768-A-12-2`	TensorFlow Tautan hub
BERT L-2_H-128_A-2 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-2-H-128-A-2`	TensorFlow Tautan hub
BERT L-2_H-256_A-4 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-2-H-256-A-4`	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
BERT L-2_H-512_A-8 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-2-H-512-A-8	TensorFlow Tautan hub
BERT L-2_H-768_A-12 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-2-H-768-A-12	TensorFlow Tautan hub
BERT L-4_H-128_A-2 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-4-H-128-A-2	TensorFlow Tautan hub
BERT L-4_H-256_A-4 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-4-H-256-A-4	TensorFlow Tautan hub
BERT L-4_H-512_A-8 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-4-H-512-A-8	TensorFlow Tautan hub
BERT L-4_H-768_A-12 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-4-H-768-A-12	TensorFlow Tautan hub
BERT L-6_H-128_A-2 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-6-H-128-A-2	TensorFlow Tautan hub
BERT L-6_H-256_A-4 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-6-H-256-A-4	TensorFlow Tautan hub
BERT L-6_H-512_A-8 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-6-H-512-A-8	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
BERT L-6_H-768_A-12 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-6-H-768-A-12`	TensorFlow Tautan hub
BERT L-8_H-128_A-2 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-8-H-128-A-2`	TensorFlow Tautan hub
BERT L-8_H-256_A-4 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-8-H-256-A-4`	TensorFlow Tautan hub
BERT L-8_H-512_A-8 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-8-H-512-A-8`	TensorFlow Tautan hub
BERT L-8_H-768_A-12 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-8-H-768-A-12`	TensorFlow Tautan hub
BERT L-10_H-128_A-2 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-10-H-128-A-2`	TensorFlow Tautan hub
BERT L-10_H-256_A-4 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-10-H-256-A-4`	TensorFlow Tautan hub
BERT L-10_H-512_A-8 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-10-H-512-A-8`	TensorFlow Tautan hub
BERT L-10_H-768_A-12 kecil	`tensorflow-tc-small-bert-bert-en-uncased-L-10-H-768-A-12`	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
BERT L-12_H-128_A-2 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-12-H-128-A-2	TensorFlow Tautan hub
BERT L-12_H-256_A-4 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-12-H-256-A-4	TensorFlow Tautan hub
BERT L-12_H-512_A-8 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-12-H-512-A-8	TensorFlow Tautan hub
BERT L-12_H-768_A-12 kecil	tensorflow-tc-small-bert-bert-en-uncased-L-12-H-768-A-12	TensorFlow Tautan hub
BERT Besar Tanpa Alas	tensorflow-tc-bert-en-uncased-L-24-H-1024-A-16-2	TensorFlow Tautan hub
BERT Sangkar Besar	tensorflow-tc-bert-en-cased-L-24-H-1024-A-16-2	TensorFlow Tautan hub
BERT Masking Kata Utuh Besar yang Tidak Dihapus	tensorflow-tc-bert-en-wwm-uncased-L-24-H-1024-A-16-2	TensorFlow Tautan hub
BERT Masking Kata Utuh Berseukuran Besar	tensorflow-tc-bert-en-wwm-cased-L-24-H-1024-A-16-2	TensorFlow Tautan hub
Pangkalan ALBERT	tensorflow-tc-albert-en-base	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
ELECTRA Kecil++	`tensorflow-tc-electra-small-1`	TensorFlow Tautan hub
Basis ELECTRA	`tensorflow-tc-electra-base-1`	TensorFlow Tautan hub
Bert Basis Wikipedia dan BooksCorpus	`tensorflow-tc-experts-bert-wiki-books-1`	TensorFlow Tautan hub
MEDLINE Basis BERT/ PubMed	`tensorflow-tc-experts-bert-pubmed-1`	TensorFlow Tautan hub
Basis Kepala Berbicara	`tensorflow-tc-talking-heads-base`	TensorFlow Tautan hub
Talking Heads Besar	`tensorflow-tc-talking-heads-large`	TensorFlow Tautan hub

Klasifikasi Teks - TensorFlow Hyperparameters

Hyperparameters adalah parameter yang ditetapkan sebelum model pembelajaran mesin mulai belajar. Hyperparameter berikut didukung oleh TensorFlow algoritma Deteksi Objek SageMaker bawaan Amazon. Lihat Menyetel Klasifikasi Teks - TensorFlow model untuk informasi tentang tuning hyperparameter.

Nama Parameter	Deskripsi
`batch_size`	Ukuran batch untuk pelatihan. Untuk pelatihan tentang instance dengan beberapa GPU, ukuran batch ini digunakan di seluruh GPU. Nilai yang valid: bilangan bulat positif. Nilai default:`32`.

Nama Parameter	Deskripsi
`beta_1`	Beta1 untuk `"adam"` dan `"adamw"` pengoptimal. Merupakan tingkat peluruhan eksponensial untuk perkiraan momen pertama. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.9`.
`beta_2`	Beta2 untuk `"adam"` dan `"adamw"` pengoptimal. Merupakan tingkat peluruhan eksponensial untuk perkiraan momen kedua. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.999`.
`dropout_rate`	Tingkat putus sekolah untuk lapisan putus sekolah di lapisan klasifikasi atas. Digunakan hanya ketika `reinitialize_top_layer` diatur ke`"True"`. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default: `0.2`
`early_stopping`	Atur `"True"` untuk menggunakan logika penghentian awal selama pelatihan. Jika`"False"`, berhenti dini tidak digunakan. Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.

Nama Parameter	Deskripsi
`early_stopping_min_delta`	Perubahan minimum yang diperlukan untuk memenuhi syarat sebagai perbaikan. Perubahan absolut kurang dari nilai `early_stopping_min_delta` tidak memenuhi syarat sebagai perbaikan. Digunakan hanya ketika `early_stopping` diatur ke`"True"`. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.0`.
`early_stopping_patience`	Jumlah zaman untuk melanjutkan pelatihan tanpa perbaikan. Digunakan hanya ketika `early_stopping` diatur ke`"True"`. Nilai yang valid: bilangan bulat positif. Nilai default:`5`.
`epochs`	Jumlah zaman pelatihan. Nilai yang valid: bilangan bulat positif. Nilai default:`10`.
`epsilon`	Epsilon untuk`"adam"`,, `"rmsprop"` `"adadelta"` , dan `"adagrad"` pengoptimal. Biasanya diatur ke nilai kecil untuk menghindari pembagian dengan 0. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`1e-7`.
`initial_accumulator_value`	Nilai awal untuk akumulator, atau nilai momentum per parameter , untuk pengoptimal. `"adagrad"` Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.0001`.

Nama Parameter	Deskripsi
`learning_rate`	Tingkat pembelajaran pengoptimal. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.001`.
`momentum`	Momentum untuk `"sgd"` dan `"nesterov"` pengoptimal. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.9`.
`optimizer`	Jenis pengoptimal. Untuk informasi selengkapnya, lihat Pengoptimal dalam dokumentasi. TensorFlow Nilai yang valid: string, salah satu dari berikut ini: (`"adamw"``"adam"`,`"sgd"`,`"nesterov"` ,`"rmsprop"` , `"adagrad"` ,`"adadelta"` ).  Nilai default:`"adam"`.
`regularizers_l2`	Faktor regularisasi L2 untuk lapisan padat di lapisan klasifikasi. Digunakan hanya ketika `reinitialize_top_layer` diatur ke`"True"`. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.0001`.

Nama Parameter	Deskripsi
`reinitialize_top_layer`	Jika disetel ke`"Auto"`, parameter lapisan klasifikasi atas diinisialisasi ulang selama fine-tuning. Untuk pelatihan tambahan, parameter lapisan klasifikasi teratas tidak diinisialisasi ulang kecuali disetel ke. `"True"` Nilai yang valid: string, salah satu dari berikut ini: (`"Auto"`, `"True"` atau`"False"`). Nilai default:`"Auto"`.
`rho`	Faktor diskon untuk gradien `"adadelta"` dan `"rmsprop"` pengoptimal. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.95`.
`train_only_on_top_layer`	Jika`"True"`, hanya parameter lapisan klasifikasi teratas yang disetel dengan baik. Jika`"False"`, semua parameter model disetel dengan baik. Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.
`validation_split_ratio`	Fraksi data pelatihan untuk dibagi secara acak untuk membuat data validasi. Hanya digunakan jika data validasi tidak disediakan melalui `validation` saluran. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.2`.

Nama Parameter	Deskripsi
`warmup_steps_fraction`	Fraksi dari jumlah total langkah pembaruan gradien, di mana tingkat pembelajaran meningkat dari 0 ke tingkat pembelajaran awal sebagai pemanasan. Hanya digunakan dengan `adamw` pengoptimal. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.1`.

Menyetel Klasifikasi Teks - TensorFlow model

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hiperparameter pada kumpulan data Anda. Anda memilih hyperparameters yang dapat disetel, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hiperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi lebih lanjut tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik dihitung oleh Klasifikasi Teks - algoritma TensorFlow

Lihat bagan berikut untuk menemukan metrik mana yang dihitung oleh algoritma Klasifikasi Teks. TensorFlow

Nama Metrik	Deskripsi	Arah Optimasi	Pola Regex
`validation:accuracy`	Rasio jumlah prediksi yang benar dengan jumlah prediksi yang dibuat.	Maksimalkan	`val_accuracy=([0-9\.]+)`

Klasifikasi Teks yang Dapat Disetel - hyperparameters TensorFlow

Tune model klasifikasi teks dengan hyperparameter berikut. Hiperparameter yang memiliki dampak terbesar pada metrik objektif klasifikasi teks adalah:`batch_size`,`learning_rate`, dan.

`optimizer` Setel hiperparameter terkait pengoptimal, seperti,,`momentum`, `regularizers_l2` `beta_1``beta_2`, dan `eps` berdasarkan yang dipilih. `optimizer` Misalnya, gunakan `beta_1` dan `beta_2` hanya ketika `adamw` atau `adam` adalah`optimizer`.

Untuk informasi lebih lanjut tentang hiperparameter mana yang digunakan untuk masing-masing`optimizer`, lihatKlasifikasi Teks - TensorFlow Hyperparameters.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`batch_size`	IntegerParameterRanges	MinValue: 4, MaxValue: 128
`beta_1`	ContinuousParameterRanges	MinValue: 1e-6,: 0.999 MaxValue
`beta_2`	ContinuousParameterRanges	MinValue: 1e-6,: 0.999 MaxValue
`eps`	ContinuousParameterRanges	MinValue: 1e-8, MaxValue: 1.0
`learning_rate`	ContinuousParameterRanges	MinValue: 1e-6, MaxValue: 0,5
`momentum`	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.999
`optimizer`	CategoricalParameterRanges	['adamw', 'adam', 'sgd', 'rmsprop', 'nesterov', 'adagrad', 'adadelta']
`regularizers_l2`	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.999
`train_onl y_on_top_layer`	CategoricalParameterRanges	['Benar', 'Salah']

Built-inSageMakerAlgoritma untuk Data Time-Series

SageMakermenyediakan algoritma yang disesuaikan dengan analisis data deret waktu untuk memperkirakan permintaan produk, beban server, permintaan halaman web, dan banyak lagi.

- Algoritma Peramalan Deepar—algoritma pembelajaran yang diawasi untuk meramalkan rangkaian waktu skalar (satu dimensi) menggunakan jaringan saraf berulang (RNN).

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas Instans	Paralleli zable	
Peramalan Deepar	kereta api dan (opsional) tes	File	Garis JSON atau Parket	GPU atau CPU	Ya	

Algoritma Peramalan Deepar

AmazonSageMakerAlgoritma peramalan Deepar adalah algoritma pembelajaran yang diawasi untuk meramalkan rangkaian waktu skalar (satu dimensi) menggunakan jaringan saraf berulang (RNN). Metode peramalan klasik, seperti autoregressive integrated moving average (ARIMA) atau exponential smoothing (ETS), sesuai dengan model tunggal untuk setiap rangkaian waktu individu. Mereka kemudian menggunakan model itu untuk mengekstrapolasi deret waktu ke masa depan.

Namun, dalam banyak aplikasi, Anda memiliki banyak rangkaian waktu serupa di satu set unit penampang melintang. Misalnya, Anda mungkin memiliki pengelompokan deret waktu untuk permintaan untuk berbagai produk, beban server, dan permintaan untuk halaman web. Untuk jenis aplikasi ini, Anda bisa mendapatkan keuntungan dari melatih satu model bersama di semua rangkaian waktu. Deepar mengambil pendekatan ini. Ketika dataset Anda berisi ratusan rangkaian waktu terkait, DeepAR mengungguli metode ARIMA dan ETS standar. Anda juga dapat menggunakan model terlatih untuk menghasilkan perkiraan untuk deret waktu baru yang mirip dengan yang telah dilatih.

Input pelatihan untuk algoritma DeepAR adalah satu atau, lebih disukai, lebih`target`time series yang telah dihasilkan oleh proses yang sama atau proses serupa. Berdasarkan dataset input ini, algoritma melatih model yang mempelajari perkiraan proses/proses ini dan menggunakannya

untuk memprediksi bagaimana rangkaian waktu target berkembang. Setiap rangkaian waktu target dapat dikaitkan secara opsional dengan vektor fitur kategoris statis (independen waktu) yang disediakan oleh`cat`bidang dan vektor dinamis (tergantung waktu) deret waktu yang disediakan oleh`dynamic_feat`bidang. SageMakermelatih model DeepAR dengan mengambil sampel contoh pelatihan secara acak dari setiap rangkaian waktu target dalam kumpulan data pelatihan. Setiap contoh pelatihan terdiri dari sepasang konteks dan jendela prediksi yang berdekatan dengan panjang yang telah ditentukan sebelumnya. Untuk mengontrol seberapa jauh di masa lalu jaringan dapat melihat, gunakan`context_length`hiperparameter. Untuk mengontrol seberapa jauh prediksi masa depan dapat dibuat, gunakan`prediction_length`hiperparameter. Untuk informasi selengkapnya, lihat Cara Kerja Algoritma DeepAR.

Topik

- Antarmuka Input/Output untuk Algoritma DeepAR
- Praktik Terbaik untuk Menggunakan Algoritma DeepAR
- Rekomendasi Instans EC2 untuk Algoritma DeepAR
- Notebook Contoh Deepar
- Cara Kerja Algoritma DeepAR
- Hiperparameter Deepar
- Tune Model Deepar
- Format Inferensi Deepar

Antarmuka Input/Output untuk Algoritma DeepAR

DeepAR mendukung dua saluran data. Diperlukan`train`channel menjelaskan kumpulan data pelatihan. Opsional`test`channel menjelaskan dataset yang digunakan algoritma untuk mengevaluasi akurasi model setelah pelatihan. Anda dapat menyediakan set data pelatihan dan pengujian diGaris JSONformat. File bisa di gzip atauParketformat file.

Saat menentukan jalur untuk data pelatihan dan pengujian, Anda dapat menentukan satu file atau direktori yang berisi beberapa file, yang dapat disimpan dalam subdirektori. Jika Anda menentukan direktori, DeepAR menggunakan semua file dalam direktori sebagai input untuk saluran yang sesuai, kecuali yang dimulai dengan periode (.) dan yang bernama_SUKSES. Ini memastikan bahwa Anda dapat langsung menggunakan folder keluaran yang diproduksi oleh pekerjaan Spark sebagai saluran input untuk pekerjaan pelatihan DeepAR Anda.

Secara default, model DeepAR menentukan format input dari ekstensi file (`.json`,`.json.gz`, atau`.parquet`) di jalur masukan yang ditentukan. Jika jalur tidak berakhir di salah satu ekstensi ini, Anda harus secara eksplisit menentukan format dalam SDK untuk Python. Gunakan`content_type`parameter dari`s3_masukan`kelas.

Catatan dalam file masukan Anda harus berisi bidang-bidang berikut:

- `start`—Sebuah string dengan format`YYYY-MM-DD HH:MM:SS`. Stempel waktu awal tidak dapat berisi informasi zona waktu.
- `target`—Array nilai floating-point atau bilangan bulat yang mewakili deret waktu. Anda dapat menyandikan nilai yang hilang sebagai`null`literal, atau sebagai`"NaN"`string di JSON, atau sebagai`nan`nilai floating-point di Parket.
- `dynamic_feat`(opsional) —Array array nilai floating-point atau bilangan bulat yang mewakili vektor dari rangkaian waktu fitur khusus (fitur dinamis). Jika Anda mengatur bidang ini, semua catatan harus memiliki jumlah array dalam yang sama (jumlah seri waktu fitur yang sama). Selain itu, setiap array batin harus memiliki panjang yang sama dengan yang terkait`target`nilai plus`prediction_length`. Nilai yang hilang tidak didukung dalam fitur. Misalnya, jika target waktu seri mewakili permintaan produk yang berbeda, yang terkait`dynamic_feat`mungkin berupa deret waktu boolean yang menunjukkan apakah promosi diterapkan (1) ke produk tertentu atau tidak (0):

  ```
  {"start": ..., "target": [1, 5, 10, 2], "dynamic_feat": [[0, 1, 1, 0]]}
  ```

- `cat`(opsional) —Sebuah array fitur kategoris yang dapat digunakan untuk mengkodekan kelompok yang catatan milik. Fitur kategoris harus dikodekan sebagai urutan bilangan bulat positif berbasis 0. Misalnya, domain kategoris {R, G, B} dapat dikodekan sebagai {0, 1, 2}. Semua nilai dari setiap domain kategoris harus direpresentasikan dalam kumpulan data pelatihan. Itu karena algoritma DeepAR hanya dapat meramalkan untuk kategori yang telah diamati selama pelatihan. Dan, setiap fitur kategoris tertanam dalam ruang dimensi rendah yang dimensionalitasnya dikendalikan oleh`embedding_dimension`hiperparameter. Untuk informasi selengkapnya, lihat Hiperparameter Deepar.

Jika Anda menggunakan file JSON, itu harus diGaris JSONformat. Misalnya:

```
{"start": "2009-11-01 00:00:00", "target": [4.3, "NaN", 5.1, ...], "cat": [0, 1],
 "dynamic_feat": [[1.1, 1.2, 0.5, ...]]}
{"start": "2012-01-30 00:00:00", "target": [1.0, -5.0, ...], "cat": [2, 3],
 "dynamic_feat": [[1.1, 2.05, ...]]}
```

```
{"start": "1999-01-30 00:00:00", "target": [2.0, 1.0], "cat": [1, 4], "dynamic_feat":
 [[1.3, 0.4]]}
```

Dalam contoh ini, setiap rangkaian waktu memiliki dua fitur kategoris terkait dan fitur satu seri waktu.

Untuk Parket, Anda menggunakan tiga bidang yang sama dengan kolom. Selain itu,`"start"`bisa menjadi`datetime`ketik. Anda dapat mengompres file Parket menggunakan gzip (`gzip`) atau perpustakaan kompresi Snappy (`snappy`).

Jika algoritma dilatih tanpa`cat`dan`dynamic_feat`bidang, ia belajar model “global”, yang merupakan model yang agnostik untuk identitas spesifik dari seri waktu target pada waktu inferensi dan dikondisikan hanya pada bentuknya.

Jika model dikondisikan pada`cat`dan`dynamic_feat`data fitur yang disediakan untuk setiap rangkaian waktu, prediksi mungkin akan dipengaruhi oleh karakter deret waktu dengan yang sesuai`cat`fitur. Misalnya, jika`target`time series mewakili permintaan item pakaian, Anda dapat mengasosiasikan dua dimensi`cat`vektor yang mengkodekan jenis item (misalnya 0 = sepatu, 1 = dress) dalam komponen pertama dan warna item (misalnya 0 = merah, 1 = biru) di komponen kedua. Masukan sampel akan terlihat sebagai berikut:

```
{ "start": ..., "target": ..., "cat": [0, 0], ... } # red shoes
{ "start": ..., "target": ..., "cat": [1, 1], ... } # blue dress
```

Pada saat inferensi, Anda dapat meminta prediksi untuk target`cat`yang merupakan kombinasi dari`cat`diamati dalam data pelatihan, misalnya:

```
{ "start": ..., "target": ..., "cat": [0, 1], ... } # blue shoes
{ "start": ..., "target": ..., "cat": [1, 0], ... } # red dress
```

Pedoman berikut berlaku untuk data pelatihan:

- Waktu mulai dan lamanya deret waktu bisa berbeda. Misalnya, dalam pemasaran, produk sering memasuki katalog ritel pada tanggal yang berbeda, sehingga tanggal mulai mereka secara alami berbeda. Tetapi semua seri harus memiliki frekuensi yang sama, jumlah fitur kategoris, dan jumlah fitur dinamis.
- Kocokkan file pelatihan sehubungan dengan posisi deret waktu dalam file. Dengan kata lain, deret waktu harus terjadi dalam urutan acak dalam file.
- Pastikan untuk mengatur`start`bidang dengan benar. Algoritma menggunakan`start`timestamp untuk mendapatkan fitur internal.

- Jika Anda menggunakan fitur kategoris (`cat`), semua rangkaian waktu harus memiliki jumlah fitur kategoris yang sama. Jika dataset berisi`cat`lapangan, algoritma menggunakannya dan ekstrak kardinalitas kelompok dari dataset. Secara default,`cardinality`adalah`"auto"`. Jika dataset berisi`cat`bidang, tetapi Anda tidak ingin menggunakannya, Anda dapat menonaktifkannya dengan mengatur`cardinality`kepada`""`. Jika model dilatih menggunakan`cat`fitur, Anda harus memasukkannya untuk kesimpulan.
- Jika dataset Anda berisi`dynamic_feat`lapangan, algoritma menggunakannya secara otomatis. Semua rangkaian waktu harus memiliki jumlah seri waktu fitur yang sama. Poin waktu di masing-masing rangkaian waktu fitur sesuai one-to-one`target`ke titik waktu di target. Selain itu, entri di`dynamic_feat`harus memiliki panjang yang sama dengan`target`. Jika dataset berisi`dynamic_feat`bidang, tetapi Anda tidak ingin menggunakannya, menonaktifkannya dengan pengaturan (`num_dynamic_feat`kepada`""`). Jika model dilatih dengan`dynamic_feat`bidang, Anda harus menyediakan bidang ini untuk inferensi. Selain itu, masing-masing fitur harus memiliki panjang target yang disediakan ditambah`prediction_length`. Dengan kata lain, Anda harus memberikan nilai fitur di masa mendatang.

Jika Anda menentukan data saluran pengujian opsional, algoritme DeepAR mengevaluasi model terlatih dengan metrik akurasi yang berbeda. Algoritma menghitung root mean square error (RMSE) di atas data uji sebagai berikut:

$$\text{RMSE} = \sqrt{\frac{1}{nT}\sum_{i,t}\left(\hat{y}_{i,t} - y_{i,t}\right)^2}$$

$y_{i,t}$adalah nilai sebenarnya dari deret waktusayapada saat itut.$\hat{y}_{i,t}$adalah prediksi rata-rata. Jumlahnya adalah atas semuanderet waktu dalam set pengujian dan selama titik waktu terakhir untuk setiap rangkaian waktu, di mana sesuai dengan cakrawala perkiraan. Anda menentukan panjang cakrawala perkiraan dengan menetapkan`prediction_length`hiperparameter. Untuk informasi selengkapnya, lihat Hiperparameter Deepar.

Selain itu, algoritma mengevaluasi keakuratan distribusi perkiraan menggunakan kehilangan kuantil tertimbang. Untuk kuantil dalam kisaran [0, 1], kehilangan kuantil tertimbang didefinisikan sebagai berikut:

$$\text{wQuantileLoss}[\tau] = 2\frac{\sum_{i,t} Q_{i,t}^{(\tau)}}{\sum_{i,t}|y_{i,t}|}, \quad \text{with} \quad Q_{i,t}^{(\tau)} = \begin{cases} (1-\tau)|q_{i,t}^{(\tau)} - y_{i,t}| & \text{if } q_{i,t}^{(\tau)} > y_{i,t} \\ \tau|q_{i,t}^{(\tau)} - y_{i,t}| & \text{otherwise} \end{cases}$$

$q_{i,t}^{(\tau)}$adalah τ-quantile dari distribusi yang diprediksi model. Untuk menentukan kuantil mana yang akan dihitung kerugian, aturtest_quantileshiperparameter. Selain itu, rata-rata kerugian kuantil yang ditentukan dilaporkan sebagai bagian dari log pelatihan. Untuk informasi, lihat Hiperparameter Deepar.

Untuk inferensi, DeepAR menerima format JSON dan bidang-bidang berikut:

- "instances", yang mencakup satu atau lebih seri waktu dalam format Garis JSON
- Sebuah nama dari"configuration", yang mencakup parameter untuk menghasilkan perkiraan

Untuk informasi selengkapnya, lihat Format Inferensi Deepar.

Praktik Terbaik untuk Menggunakan Algoritma DeepAR

Saat menyiapkan data deret waktu Anda, ikuti praktik terbaik ini untuk mencapai hasil terbaik:

- Kecuali saat membagi kumpulan data Anda untuk pelatihan dan pengujian, selalu berikan seluruh rangkaian waktu untuk pelatihan, pengujian, dan saat memanggil model untuk inferensi. Terlepas dari bagaimana Anda mengaturcontext_length, Jangan memecah deret waktu atau hanya menyediakan sebagian saja. Model ini menggunakan titik data lebih jauh dari nilai yang ditetapkancontext_lengthuntuk fitur nilai yang tertinggal.
- Saat menyetel model DeepAR, Anda dapat membagi kumpulan data untuk membuat set data pelatihan dan kumpulan data pengujian. Dalam evaluasi tipikal, Anda akan menguji model pada rangkaian waktu yang sama yang digunakan untuk pelatihan, tetapi di masa depanprediction_lengthpoin waktu yang mengikuti segera setelah titik waktu terakhir terlihat selama pelatihan. Anda dapat membuat set data pelatihan dan pengujian yang memenuhi kriteria ini dengan menggunakan seluruh kumpulan data (panjang penuh dari semua rangkaian waktu yang tersedia) sebagai set pengujian dan menghapus yang terakhirprediction_lengthpoin dari setiap rangkaian waktu untuk pelatihan. Selama pelatihan, model tidak melihat nilai target untuk titik waktu yang dievaluasi selama pengujian. Selama pengujian, algoritma menahan yang terakhirprediction_lengthpoin dari setiap rangkaian waktu dalam set tes dan menghasilkan prediksi. Kemudian membandingkan perkiraan dengan nilai yang ditahan. Anda dapat membuat evaluasi yang lebih kompleks dengan mengulangi deret waktu beberapa kali dalam set pengujian, tetapi memotongnya pada titik akhir yang berbeda. Dengan pendekatan ini, metrik akurasi dirata-ratakan selama beberapa prakiraan dari titik waktu yang berbeda. Untuk informasi selengkapnya, lihat Tune Model Deepar.

- Hindari menggunakan nilai yang sangat besar (> 400) untuk`prediction_length`karena membuat model lambat dan kurang akurat. Jika Anda ingin meramalkan lebih jauh ke masa depan, pertimbangkan untuk menggabungkan data Anda pada frekuensi yang lebih rendah. Misalnya, gunakan `5min` sebagai ganti dari `1min`.
- Karena kelambatan digunakan, model dapat melihat lebih jauh ke belakang dalam deret waktu daripada nilai yang ditentukan`context_length`. Oleh karena itu, Anda tidak perlu mengatur parameter ini ke nilai yang besar. Kami sarankan untuk memulai dengan nilai yang Anda gunakan`prediction_length`.
- Kami merekomendasikan melatih model DeepAR pada rangkaian waktu sebanyak yang tersedia. Meskipun model DeepAR yang dilatih pada rangkaian waktu tunggal mungkin bekerja dengan baik, algoritma peramalan standar, seperti ARIMA atau ETS, mungkin memberikan hasil yang lebih akurat. Algoritma DeepAR mulai mengungguli metode standar ketika kumpulan data Anda berisi ratusan rangkaian waktu terkait. Saat ini, DeepAR mensyaratkan bahwa jumlah pengamatan yang tersedia di semua rangkaian waktu pelatihan setidaknya 300.

Rekomendasi Instans EC2 untuk Algoritma DeepAR

Anda dapat melatih DeepAR pada instans GPU dan CPU dan dalam pengaturan tunggal dan multi-mesin. Sebaiknya mulai dengan satu instance CPU (misalnya, ml.c4.2xlarge atau ml.c4.4xlarge), dan beralih ke instans GPU dan beberapa mesin hanya bila diperlukan. Menggunakan GPU dan beberapa mesin meningkatkan throughput hanya untuk model yang lebih besar (dengan banyak sel per lapisan dan banyak lapisan) dan untuk ukuran mini-batch besar (misalnya, lebih besar dari 512).

Untuk inferensi, DeepAR hanya mendukung instance CPU.

Menentukan nilai besar untuk`context_length`,`prediction_length`,`num_cells`,`num_layers`, atau`mini_batch_size`dapat membuat model yang terlalu besar untuk instance kecil. Dalam hal ini, gunakan jenis instance yang lebih besar atau kurangi nilai untuk parameter ini. Masalah ini juga sering terjadi saat menjalankan pekerjaan penyetelan hyperparameter. Dalam hal ini, gunakan tipe instans yang cukup besar untuk pekerjaan penyetelan model dan pertimbangkan untuk membatasi nilai atas parameter kritis untuk menghindari kegagalan pekerjaan.

Notebook Contoh Deepar

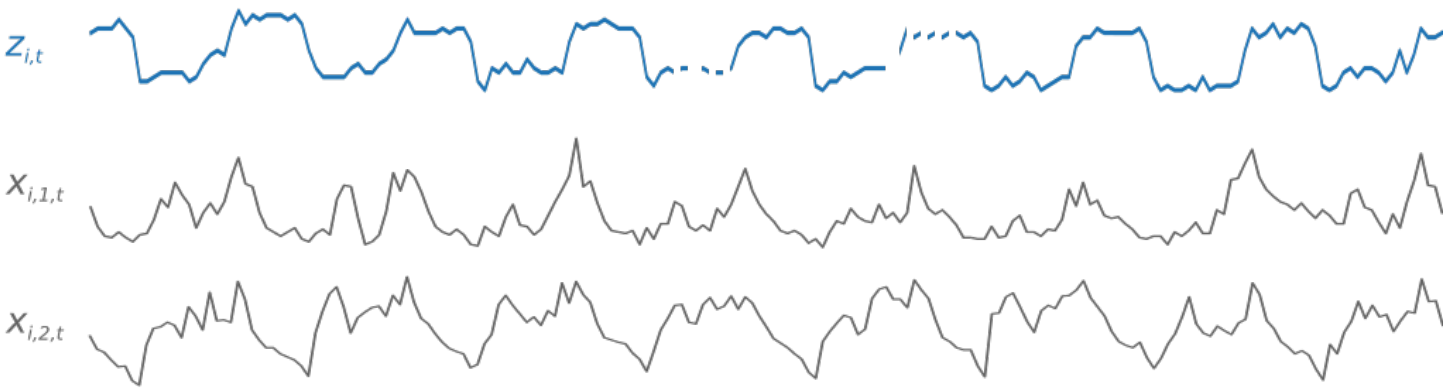
Untuk contoh buku catatan yang menunjukkan cara menyiapkan kumpulan data deret waktu untuk melatihSageMakerAlgoritma DeepAR dan cara menerapkan model terlatih untuk melakukan kesimpulan, lihatDemo DeepAR pada dataset listrik, yang menggambarkan fitur-fitur

canggih DeepAR pada dataset dunia nyata. Untuk petunjuk tentang membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contohSageMaker, lihatInstans SageMaker Notebook Amazon. Setelah membuat dan membuka instance notebook, pilihSageMakerContohtab untuk melihat daftar semuaSageMakercontoh. Untuk membuka notebook, pilihGunakantab, dan pilihBuat salinan.

Cara Kerja Algoritma DeepAR

Selama pelatihan, DeepAR menerima kumpulan data pelatihan dan kumpulan data pengujian opsional. Menggunakan dataset tes untuk mengevaluasi model terlatih. Secara umum, kumpulan data tidak harus berisi rangkaian waktu yang sama. Anda dapat menggunakan model yang dilatih pada set pelatihan tertentu untuk menghasilkan perkiraan untuk masa depan rangkaian waktu dalam rangkaian pelatihan, dan untuk rangkaian waktu lainnya. Baik pelatihan dan kumpulan data pengujian terdiri dari satu atau, lebih disukai, lebih banyak seri waktu target. Setiap rangkaian waktu target secara opsional dapat dikaitkan dengan vektor rangkaian waktu fitur dan vektor fitur kategoris. Untuk informasi selengkapnya, lihat Antarmuka Input/Output untuk Algoritma DeepAR.

Misalnya, berikut ini adalah elemen dari set pelatihan yang diindeks olehsayayang terdiri dari rangkaian waktu target,$Z_{i,t}$, dan dua rangkaian waktu fitur terkait,$X_{i,1,t}$dan$X_{i,2,t}$:



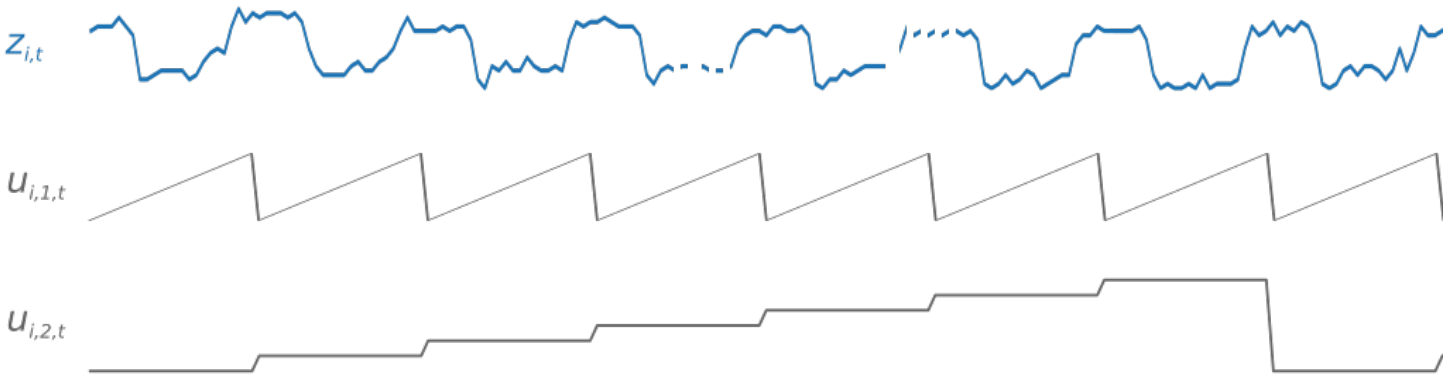
Target time series mungkin mengandung nilai yang hilang, yang diwakili oleh jeda baris dalam deret waktu. DeepAR hanya mendukung fitur time series yang dikenal di masa depan. Hal ini memungkinkan Anda untuk menjalankan “bagaimana jika?” skenario. Apa yang terjadi, misalnya, jika saya mengubah harga suatu produk dengan cara tertentu?

Setiap rangkaian waktu target juga dapat dikaitkan dengan sejumlah fitur kategoris. Anda dapat menggunakan fitur ini untuk menyandikan pengelompokan yang dimiliki oleh deret waktu. Fitur kategoris memungkinkan model untuk mempelajari perilaku khas untuk kelompok, yang dapat digunakan untuk meningkatkan akurasi model. DeepAR mengimplementasikan ini dengan

mempelajari vektor embedding untuk setiap kelompok yang menangkap properti umum dari semua rangkaian waktu dalam grup.

Cara Kerja Seri Waktu Fitur dalam Algoritma DeepAR

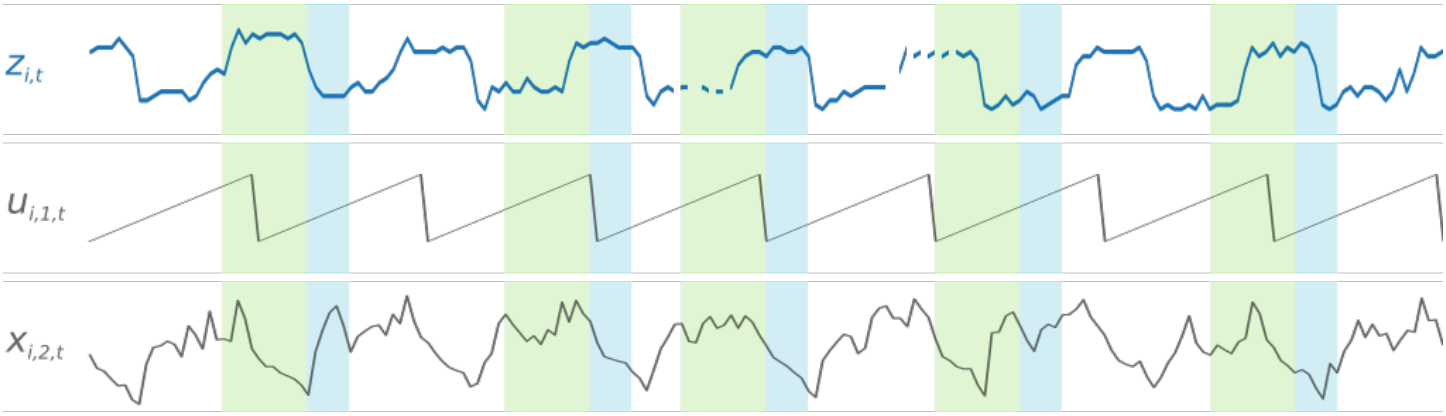
Untuk memfasilitasi pola yang bergantung pada waktu pembelajaran, seperti lonjakan selama akhir pekan, DeepAR secara otomatis membuat rangkaian waktu fitur berdasarkan frekuensi deret waktu target. Ini menggunakan rangkaian waktu fitur turunan ini dengan deret waktu fitur khusus yang Anda berikan selama pelatihan dan inferensi. Gambar berikut menunjukkan dua fitur deret waktu turunan ini:$u_{i,1,t}$mewakili jam hari danu$_{i,2,t}$hari dalam seminggu.



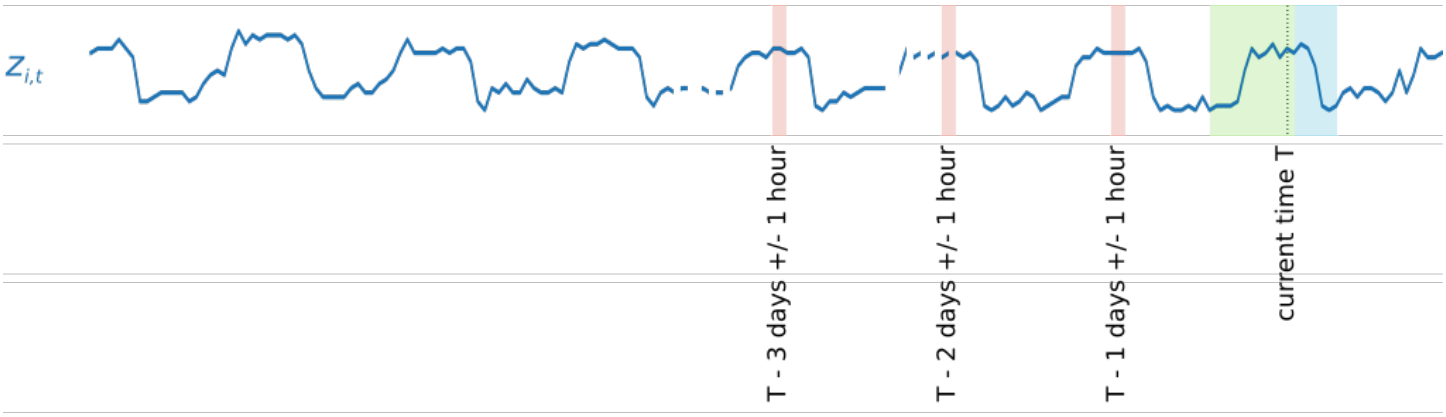
Algoritma DeepAR secara otomatis menghasilkan rangkaian waktu fitur ini. Tabel berikut mencantumkan fitur turunan untuk frekuensi waktu dasar yang didukung.

Frekuensi Seri Waktu	Fitur Berasal
`Minute`	`minute-of-hour` , `hour-of-day` , `day-of-week` , `day-of-month` , `day-of-year`
`Hour`	`hour-of-day` , `day-of-week` , `day-of-month` , `day-of-year`
`Day`	`day-of-week` , `day-of-month` , `day-of-year`
`Week`	`day-of-month` , `week-of-year`
`Month`	month-of-year

DeepAR melatih model dengan mengambil sampel secara acak beberapa contoh pelatihan dari masing-masing rangkaian waktu dalam kumpulan data pelatihan. Setiap contoh pelatihan terdiri dari sepasang konteks dan jendela prediksi yang berdekatan dengan panjang yang telah ditentukan sebelumnya. Yang`context_length`hyperparameter mengontrol seberapa jauh di masa lalu jaringan dapat melihat, dan`prediction_length`hyperparameter mengontrol seberapa jauh dalam prediksi masa depan dapat dibuat. Selama pelatihan, algoritme mengabaikan elemen set pelatihan yang berisi deret waktu yang lebih pendek dari panjang prediksi yang ditentukan. Gambar berikut mewakili lima sampel dengan panjang konteks 12 jam dan panjang prediksi 6 jam diambil dari elemensaya. Singkatnya, kami telah menghilangkan rangkaian waktu fitur$x_{i,1,t}$dan$u_{i,2,t}$.



Untuk menangkap pola musiman, DeepAR juga secara otomatis memberi makan nilai yang tertinggal dari deret waktu target. Dalam contoh dengan frekuensi per jam, untuk setiap indeks waktu,t = T, model mengeksposz$_{i,t}$nilai, yang terjadi sekitar satu, dua, dan tiga hari di masa lalu.



Untuk inferensi, model terlatih mengambil sebagai rangkaian waktu target input, yang mungkin atau mungkin tidak digunakan selama pelatihan, dan memperkirakan distribusi probabilitas untuk selanjutnya`prediction_length`nilai-nilai. Karena DeepAR dilatih di seluruh dataset, perkiraan memperhitungkan pola akun yang dipelajari dari rangkaian waktu yang sama.

Untuk informasi tentang matematika di balik Deepar, lihatDeepAR: Peramalan Probabilistik dengan Jaringan Berulang Autoregresif.

Hiperparameter Deepar

Nama Parameter	Deskripsi
`context_length`	Jumlah titik-waktu yang dapat dilihat model sebelum membuat prediksi. Nilai untuk parameter ini harus hampir sama dengan`prediction_length` . Model ini juga menerima input yang tertinggal dari target, jadi`context_length` bisa jauh lebih kecil dari musiman khas. Misalnya, rangkaian waktu harian dapat memiliki musiman tahunan. Model ini secara otomatis mencakup jeda satu tahun, sehingga panjang konteksnya bisa lebih pendek dari setahun. Nilai lag yang diambil model bergantung pada frekuensi deret waktu. Misalnya, nilai lag untuk frekuensi harian adalah minggu sebelumnya, 2 minggu, 3 minggu, 4 minggu, dan tahun. Diperlukan Nilai yang valid: bilangan bulat positif
`epochs`	Jumlah maksimum melewati data pelatihan. Nilai optimal tergantung pada ukuran data dan tingkat pembelajaran Anda. Lihat juga `early_stopping_patience` . Nilai tipikal berkisar antara 10 hingga 1000. Diperlukan Nilai yang valid: bilangan bulat positif
`prediction_length`	Jumlah waktu-langkah yang model dilatih untuk memprediksi, juga disebut cakrawala perkiraan. Model terlatih selalu menghasilkan perkiraan dengan panjang ini. Itu tidak dapat menghasilkan perkiraan yang lebih lama. Yang`prediction_length` diperbaiki ketika model dilatih dan tidak dapat diubah nanti.

Nama Parameter	Deskripsi
	Diperlukan Nilai yang valid: bilangan bulat positif
`time_freq`	Granularitas dari deret waktu dalam dataset. Gunakan`time_freq` untuk memilih fitur tanggal yang sesuai dan tertinggal. Model ini mendukung frekuensi dasar berikut. Ini juga mendukung kelipatan frekuensi dasar ini. Sebagai contoh,`5min`menentukan frekuensi 5 menit. • M: bulanan • W: mingguan • D: setiap hari • H: per jam • min: setiap menit Diperlukan Nilai yang valid: Sebuah integer diikuti oleh`M`,`W`,`D`,`H`, atau`min`. Sebagai contoh,`5min`.

Nama Parameter	Deskripsi
`cardinality`	Saat menggunakan fitur kategoris (`cat`),`cardinality` adalah array menentukan jumlah kategori (kelompok) per fitur kategoris . Atur ini ke`auto`untuk menyimpulkan kardinalitas dari data. Yang`auto`modus juga bekerja ketika tidak ada fitur kategoris yang digunakan dalam dataset. Ini adalah pengaturan yang disarankan untuk parameter. Atur kardinalitas ke`ignore`untuk memaksa DeepAR untuk tidak menggunakan fitur kategoris, bahkan itu mereka hadir dalam data. Untuk melakukan validasi data tambahan, dimungkinkan untuk secara eksplisit mengatur parameter ini ke nilai aktual. Misalnya, jika dua fitur kategoris disediakan di mana yang pertama memiliki 2 dan yang lainnya memiliki 3 nilai yang mungkin, atur ini ke [2, 3]. Untuk informasi lebih lanjut tentang cara menggunakan fitur kategoris, lihat bagian data pada halaman dokumentasi utama DeepAR. Opsional Nilai yang valid:`auto`,`ignore`, array bilangan bulat positif, string kosong, atau Nilai default: `auto`

Nama Parameter	Deskripsi
`dropout_rate`	Tingkat putus sekolah untuk digunakan selama pelatihan. Model ini menggunakan regularisasi zoneout. Untuk setiap iterasi, subset acak neuron tersembunyi tidak diperbarui. Nilai tipikal kurang dari 0,2. Opsional Nilai yang valid: float Nilai default: 0,1
`early_stopping_patience`	Jika parameter ini disetel, pelatihan akan berhenti jika tidak ada kemajuan yang dibuat dalam jumlah yang ditentukan`epochs`. Model yang memiliki kerugian terendah dikembalikan sebagai model akhir. Opsional Nilai yang valid: bilangan bulat

Nama Parameter	Deskripsi
`embedding_dimension`	Ukuran vektor embedding dipelajari per fitur kategoris (nilai yang sama digunakan untuk semua fitur kategoris). Model DeepAR dapat mempelajari pola deret waktu tingkat grup ketika fitur pengelompokan kategoris disediakan. Untuk melakukan ini, model mempelajari vektor embedding ukuran`embedding_dimension` untuk setiap kelompok, menangkap sifat umum dari semua rangkaian waktu dalam grup. Yang lebih besar`embedding_dimension` memungkin kan model untuk menangkap pola yang lebih kompleks. Namun, karena meningkatkan`embedding_dimension` meningkat kan jumlah parameter dalam model, lebih banyak data pelatihan diperlukan untuk mempelajari parameter ini secara akurat. Nilai khas untuk parameter ini adalah antara 10-100. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 10
`learning_rate`	Tingkat pembelajaran yang digunakan dalam pelatihan. Nilai tipikal berkisar dari 1e-4 hingga 1e-1. Opsional Nilai yang valid: float Nilai default: 1e-3

Nama Parameter	Deskripsi
`likelihood`	Model menghasilkan perkiraan probabilistik, dan dapat memberikan kuantil distribusi dan kembali sampel. Tergantun g pada data Anda, pilih kemungkinan yang sesuai (model kebisingan) yang digunakan untuk perkiraan ketidakpastian. Kemungkinan-kemungkinan berikut dapat dipilih: • gaussian: Gunakan untuk data bernilai nyata. • beta: Gunakan untuk target bernilai nyata antara 0 dan 1 inklusif. • negatif-binomial: Gunakan untuk menghitung data (bilangan bulat non-negatif). • Mahasiswa-T: Alternatif untuk data bernilai nyata yang bekerja dengan baik untuk data yang meledak. • Deterministik-L1: Fungsi kerugian yang tidak memperkirakan ketidakpastian dan hanya mempelajari perkiraan poin. Opsional Nilai yang valid: Salah satugaussian,beta,negatif-b inomial,Mahasiswa-T, atauDeterministik-L1. Nilai default: `student-T`
`mini_batch_size`	Ukuran mini-batch yang digunakan selama pelatihan. Nilai tipikal berkisar antara 32 hingga 512. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 128

Nama Parameter	Deskripsi
num_cells	Jumlah sel yang digunakan di setiap lapisan tersembunyi RNN. Nilai tipikal berkisar antara 30 hingga 100. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 40
num_dynamic_feat	Jumlahdynamic_feat disediakan dalam data. Atur ini keautountuk menyimpulkan jumlah fitur dinamis dari data. Yangautomodus juga bekerja ketika tidak ada fitur dinamis yang digunakan dalam dataset. Ini adalah pengaturan yang disarankan untuk parameter. Untuk memaksa DeepAR untuk tidak menggunakan fitur dinamis, bahkan itu mereka hadir dalam data, setnum_dynamic_feat kepadaignore. Untuk melakukan validasi data tambahan, dimungkinkan untuk secara eksplisit mengatur parameter ini ke nilai integer aktual. Misalnya, jika dua fitur dinamis disediakan, atur ini ke 2. Opsional Nilai yang valid:auto,ignore, bilangan bulat positif, atau string kosong Nilai default: auto

Nama Parameter	Deskripsi
num_eval_samples	Jumlah sampel yang digunakan per deret waktu saat menghitung metrik akurasi pengujian. Parameter ini tidak memiliki pengaruh pada pelatihan atau model akhir. Secara khusus, model dapat ditanyakan dengan jumlah sampel yang berbeda. Parameter ini hanya memengaruhi skor akurasi yang dilaporkan pada saluran uji setelah pelatihan. Nilai yang lebih kecil menghasilkan evaluasi yang lebih cepat, tetapi kemudian skor evaluasi biasanya lebih buruk dan lebih tidak pasti. Saat mengevaluasi dengan kuantil yang lebih tinggi, misalnya 0,95, mungkin penting untuk meningkatkan jumlah sampel evaluasi. Opsional Nilai yang valid: bilangan bulat Nilai default: 100
num_layers	Jumlah lapisan tersembunyi di RNN. Nilai tipikal berkisar dari 1 hingga 4. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 2
test_quantiles	Kuantil untuk menghitung kerugian kuantil pada saluran uji. Opsional Nilai yang valid: array mengapung Nilai default: [0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9]

Tune Model Deepar

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada

dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma DeepAR

Algoritma DeepAR melaporkan tiga metrik, yang dihitung selama pelatihan. Saat menyetel model, pilih salah satunya sebagai tujuannya. Untuk tujuan tersebut, gunakan akurasi perkiraan pada saluran pengujian yang disediakan (disarankan) atau kehilangan pelatihan. Untuk rekomendasi untuk pelatihan/test split untuk algoritma DeepAR, lihat Praktik Terbaik untuk Menggunakan Algoritma DeepAR.

Nama Metrik	Deskripsi	Arah Optimasi
`test:RMSE`	Akar berarti kesalahan kuadrat antara perkiraan dan target aktual dihitung pada set pengujian.	Meminimalkan
`test:mean_wQuantileLoss`	Rata-rata kerugian kuantil keseluruhan dihitung pada set tes. Untuk mengontrol kuantil mana yang digunakan, atur `test_quantiles` hiperparameter.	Meminimalkan
`train:final_loss`	Kerugian log-kemungkinan negatif pelatihan rata-rata selama periode pelatihan terakhir untuk model.	Minimalkan

Hyperparameter merdu untuk Algoritma DeepAR

Tune model DeepAR dengan hyperparameters berikut. Hyperparameter yang memiliki dampak terbesar, tercantum dalam urutan dari yang paling tidak berdampak, pada metrik objektif DeepAR adalah:`epochs`,`context_length`,`mini_batch_size`,`learning_rate`, dan`num_cells`.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
epochs	IntegerParameterRanges	MinValue: 1,MaxValue: 1000
context_length	IntegerParameterRanges	MinValue: 1,MaxValue: 200
mini_batch_size	IntegerParameterRanges	MinValue: 32,MaxValue: 1028
learning_rate	ContinuousParameterRange	MinValue: 1e-5MaxValue: 1e-1
num_cells	IntegerParameterRanges	MinValue: 30,MaxValue: 200
num_layers	IntegerParameterRanges	MinValue: 1,MaxValue: 8
dropout_rate	ContinuousParameterRange	MinValue: 0,00,MaxValue: 0.2
embedding_dimension	IntegerParameterRanges	MinValue: 1,MaxValue: 50

Format Inferensi Deepar

Format Permintaan Deepar JSON

Query model terlatih dengan menggunakan endpoint model. Endpoint mengambil format permintaan JSON berikut.

Dalam permintaan,`instances`bidang sesuai dengan deret waktu yang harus diramalkan oleh model.

Jika model dilatih dengan kategori, Anda harus memberikan`cat`untuk setiap contoh. Jika model dilatih tanpa`cat`bidang, itu harus dihilangkan.

Jika model dilatih dengan rangkaian waktu fitur khusus (`dynamic_feat`), Anda harus memberikan jumlah yang sama`dynamic_feat`nilai untuk setiap contoh. Masing-masing harus memiliki panjang yang diberikan oleh`length(target) + prediction_length`, dimana yang terakhir`prediction_length`nilai-nilai sesuai dengan titik waktu di masa depan yang akan diprediksi. Jika model dilatih tanpa seri waktu fitur khusus, bidang tidak boleh disertakan dalam permintaan.

```
{
    "instances": [
        {
            "start": "2009-11-01 00:00:00",
            "target": [4.0, 10.0, "NaN", 100.0, 113.0],
            "cat": [0, 1],
            "dynamic_feat": [[1.0, 1.1, 2.1, 0.5, 3.1, 4.1, 1.2, 5.0, ...]]
        },
        {
            "start": "2012-01-30",
            "target": [1.0],
            "cat": [2, 1],
            "dynamic_feat": [[2.0, 3.1, 4.5, 1.5, 1.8, 3.2, 0.1, 3.0, ...]]
        },
        {
            "start": "1999-01-30",
            "target": [2.0, 1.0],
            "cat": [1, 3],
            "dynamic_feat": [[1.0, 0.1, -2.5, 0.3, 2.0, -1.2, -0.1, -3.0, ...]]
        }
    ],
    "configuration": {
         "num_samples": 50,
         "output_types": ["mean", "quantiles", "samples"],
         "quantiles": ["0.5", "0.9"]
    }
}
```

Yang`configuration`lapangan adalah opsional.`configuration.num_samples`menetapkan jumlah jalur sampel yang dihasilkan model untuk memperkirakan mean dan quantiles.`configuration.output_types`menjelaskan informasi yang akan dikembalikan dalam permintaan. Nilai yang valid adalah`"mean"` `"quantiles"`dan`"samples"`. Jika Anda menentukan`"quantiles"`, masing-masing nilai kuantil di`configuration.quantiles`dikembalikan sebagai deret waktu. Jika Anda

menentukan`"samples"`, model juga mengembalikan sampel mentah yang digunakan untuk menghitung output lainnya.

Format Respons Deepar JSON

Berikut ini adalah format respons, di mana`[...]`adalah array angka:

```
{
    "predictions": [
        {
            "quantiles": {
                "0.9": [...],
                "0.5": [...]
            },
            "samples": [...],
            "mean": [...]
        },
        {
            "quantiles": {
                "0.9": [...],
                "0.5": [...]
            },
            "samples": [...],
            "mean": [...]
        },
        {
            "quantiles": {
                "0.9": [...],
                "0.5": [...]
            },
            "samples": [...],
            "mean": [...]
        }
    ]
}
```

Deepar memiliki batas waktu respons 60 detik. Saat melewati beberapa deret waktu dalam satu permintaan, prakiraan dihasilkan secara berurutan. Karena perkiraan untuk setiap rangkaian waktu biasanya memakan waktu sekitar 300 hingga 1000 milidetik atau lebih lama, tergantung pada ukuran model, melewati terlalu banyak deret waktu dalam satu permintaan dapat menyebabkan waktu habis. Lebih baik mengirim lebih sedikit deret waktu per permintaan dan mengirim lebih banyak permintaan.

Karena algoritma DeepAR menggunakan beberapa pekerja per instans, Anda dapat mencapai throughput yang jauh lebih tinggi dengan mengirimkan beberapa permintaan secara paralel.

Secara default, DeepAR menggunakan satu pekerja per CPU untuk inferensi, jika ada cukup memori per CPU. Jika modelnya besar dan tidak ada cukup memori untuk menjalankan model pada setiap CPU, jumlah pekerja akan berkurang. Jumlah pekerja yang digunakan untuk inferensi dapat ditimpa menggunakan variabel lingkunganMODEL_SERVER_WORKERSMisalnya, dengan menetapkanMODEL_SERVER_WORKERS=1) saat memanggilSageMaker CreateModelAPI.

Batch Transform dengan Algoritma DeepAR

Peramalan Deepar mendukung mendapatkan kesimpulan dengan menggunakan batch transform dari data menggunakan format JSON Lines. Dalam format ini, setiap catatan diwakili pada satu baris sebagai objek JSON, dan garis dipisahkan oleh karakter baris baru. Formatnya identik dengan format Garis JSON yang digunakan untuk pelatihan model. Untuk informasi, lihat Antarmuka Input/Output untuk Algoritma DeepAR. Misalnya:

```
{"start": "2009-11-01 00:00:00", "target": [4.3, "NaN", 5.1, ...], "cat": [0, 1],
 "dynamic_feat": [[1.1, 1.2, 0.5, ..]]}
{"start": "2012-01-30 00:00:00", "target": [1.0, -5.0, ...], "cat": [2, 3],
 "dynamic_feat": [[1.1, 2.05, ...]]}
{"start": "1999-01-30 00:00:00", "target": [2.0, 1.0], "cat": [1, 4], "dynamic_feat":
 [[1.3, 0.4]]}
```

> **Note**
>
> Saat membuat pekerjaan transformasi denganCreateTransformJob, mengaturBatchStrategynilai keSingleRecorddan tetaplahSplitTypenilai dalamTransformInputkonfigurasi keLine, sebagai nilai default saat ini menyebabkan kegagalan runtime.

Mirip dengan format permintaan inferensi titik akhir yang di-host,catdandynamic_featbidang untuk setiap instance diperlukan jika kedua hal berikut ini benar:

- Model ini dilatih pada dataset yang berisi keduacatdandynamic_featbidang.
- Yang sesuaicardinalitydannum_dynamic_featnilai yang digunakan dalam pekerjaan pelatihan tidak diatur ke"".

Tidak seperti inferensi endpoint yang di-host, bidang konfigurasi disetel satu kali untuk seluruh tugas inferensi batch menggunakan variabel lingkungan bernamaDEEPAR_INFERENCE_CONFIG. Nilai dariDEEPAR_INFERENCE_CONFIGdapat dilewatkan ketika model dibuat dengan memanggilCreateTransformJobAPI. JikaDEEPAR_INFERENCE_CONFIGhilang dalam lingkungan kontainer, wadah inferensi menggunakan default berikut:

```
{
    "num_samples": 100,
    "output_types": ["mean", "quantiles"],
    "quantiles": ["0.1", "0.2", "0.3", "0.4", "0.5", "0.6", "0.7", "0.8", "0.9"]
}
```

Outputnya juga dalam format Garis JSON, dengan satu baris per prediksi, dalam urutan yang identik dengan urutan instance dalam file input yang sesuai. Prediksi dikodekan sebagai objek yang identik dengan yang dikembalikan oleh respons dalam mode inferensi online. Misalnya:

```
{ "quantiles": { "0.1": [...], "0.2": [...] }, "samples": [...], "mean": [...] }
```

Perhatikan bahwa diTransformInputkonfigurasiSageMaker CreateTransformJobmeminta klien harus secara eksplisit mengaturAssembleWithnilai keLine, sebagai nilai defaultNonemenggabungkan semua objek JSON pada baris yang sama.

Misalnya, di sini adalahSageMaker CreateTransformJobmeminta pekerjaan DeepAR dengan customDEEPAR_INFERENCE_CONFIG:

```
{
   "BatchStrategy": "SingleRecord",
   "Environment": {
      "DEEPAR_INFERENCE_CONFIG" : "{ \"num_samples\": 200, \"output_types\": [\"mean
\"] }",
      ...
   },
   "TransformInput": {
      "SplitType": "Line",
      ...
   },
   "TransformOutput": {
      "AssembleWith": "Line",
      ...
   },
```

```
    ...
}
```

Built-in tanpa pengawasanSageMakerAlgoritma

AmazonSageMakermenyediakan beberapa algoritma bawaan yang dapat digunakan untuk berbagai tugas pembelajaran tanpa pengawasan seperti pengelompokan, pengurangan dimensi, pengenalan pola, dan deteksi anomali.

- Wawasan IP—learns pola penggunaan untuk alamat IPv4. Ini dirancang untuk menangkap asosiasi antara alamat IPv4 dan berbagai entitas, seperti ID pengguna atau nomor akun.
- Algoritma K-Means—finds pengelompokan diskrit dalam data, di mana anggota kelompok yang sama mungkin satu sama lain dan berbeda mungkin dari anggota kelompok lain.
- Algoritma Analisis Komponen Utama (PCA)—reduce dimensi (jumlah fitur) dalam dataset dengan memproyeksikan titik data ke beberapa komponen utama pertama. Tujuannya adalah untuk menyimpan sebanyak mungkin informasi atau variasi. Bagi ahli matematika, komponen utama adalah vektor vektor matriks kovarians data.
- Algoritma Acak Cut Forest (RCF)—detects anomali titik data dalam kumpulan data yang menyimpang dari jika tidak terstruktur dengan baik atau data berpola.

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas Instans	Paralleli zable	
Wawasan IP	melatih dan (opsional) validasi	File	CSV	CPU atau GPU	Ya	
K-Berarti	kereta api dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU atau GPUCommo (perangka t GPU tunggal pada satu atau	Tidak	

Nama algoritma	Nama saluran	Mode masukan pelatihan	Tipe file	Kelas Instans	Paralleli zable	
				beberapa instance)		
PCA	kereta api dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	GPU atau CPU	Ya	
Forest Pemotonga n Acak	kereta api dan (opsional) tes	File atau Pipa	Recordio-protobuf atau CSV	CPU	Ya	

Wawasan IP

Amazon SageMaker IP Insights adalah algoritma pembelajaran tanpa pengawasan yang mempelajari pola penggunaan untuk alamat IPv4. Ini dirancang untuk menangkap asosiasi antara alamat IPv4 dan berbagai entitas, seperti ID pengguna atau nomor akun. Anda dapat menggunakannya untuk mengidentifikasi pengguna yang mencoba masuk ke layanan web dari alamat IP anomali, misalnya. Atau Anda dapat menggunakannya untuk mengidentifikasi akun yang mencoba membuat sumber daya komputasi dari alamat IP yang tidak biasa. Model IP Insight yang terlatih dapat dihosting di titik akhir untuk membuat prediksi waktu nyata atau digunakan untuk memproses transformasi batch.

SageMaker Wawasan IP menelan data historis sebagai pasangan (entitas, Alamat IPv4) dan mempelajari pola penggunaan IP dari setiap entitas. Ketika ditanya dengan acara (entitas, Alamat IPv4), a SageMaker Model IP Insights mengembalikan skor yang menyimpulkan seberapa anomali pola acara tersebut. Misalnya, ketika pengguna mencoba masuk dari alamat IP, jika skor IP Insights cukup tinggi, server login web mungkin memutuskan untuk memicu sistem otentikasi multi-faktor. Dalam solusi yang lebih canggih, Anda dapat memasukkan skor IP Insights ke model pembelajaran mesin lainnya. Misalnya, Anda dapat menggabungkan skor IP Insight dengan fitur lain untuk menentukan peringkat temuan sistem keamanan lain, seperti yang dariAmazon GuardDuty.

Klaster SageMaker Algoritma IP Insights juga dapat mempelajari representasi vektor alamat IP, yang dikenal sebagaiembeddings. Anda dapat menggunakan embeddings yang dikodekan vektor

sebagai fitur dalam tugas machine learning hilir yang menggunakan informasi yang diamati di alamat IP. Misalnya, Anda dapat menggunakannya dalam tugas-tugas seperti mengukur kesamaan antara alamat IP dalam tugas pengelompokan dan visualisasi.

Topik

Antarmuka Input/Output untuk Algoritma Wawasan IP

Pelatihan dan Validasi

Klaster SageMaker Algoritma IP Insights mendukung saluran data pelatihan dan validasi. Menggunakan saluran validasi opsional untuk menghitung area-under-curve (AUC) skor pada strategi pengambilan sampel negatif yang telah ditetapkan. Metrik AUC memvalidasi seberapa baik model membedakan antara sampel positif dan negatif. Pelatihan dan validasi jenis konten data harus di`text/csv`format. Kolom pertama dari data CSV adalah string buram yang menyediakan pengenal unik untuk entitas. Kolom kedua adalah alamat IPv4 dalam notasi desimal-dot. IP Insights saat ini hanya mendukung mode File. Untuk informasi lebih lanjut dan beberapa contoh, lihatFormat Data Pelatihan Wawasan IP.

Kesimpulan inferensi

Untuk inferensi, IP Insights mendukung`text/csv`,`application/json`, dan`application/jsonlines`jenis konten data. Untuk informasi lebih lanjut tentang format data umum untuk inferensi yang disediakan oleh SageMaker, LihatFormat Data Umum untuk Inferensi. Inferensi IP Insights mengembalikan output yang diformat sebagai baik`application/json`atau`application/jsonlines`. Setiap catatan dalam data output berisi yang sesuai`dot_product`(atau skor kompatibilitas) untuk setiap titik data input. Untuk informasi lebih lanjut dan beberapa contoh, lihatFormat Data Inferensi Wawasan IP.

Rekomendasi Instans EC2 untuk Algoritma Wawasan IP

Klaster SageMaker Algoritma IP Insights dapat berjalan pada instans GPU dan CPU. Untuk pekerjaan pelatihan, sebaiknya gunakan instans GPU. Namun, untuk beban kerja tertentu dengan set data pelatihan yang besar, instans CPU terdistribusi dapat mengurangi biaya pelatihan. Untuk inferensi, sebaiknya gunakan instans CPU. IP Insights mendukung keluarga GPU P2, P3, G4dn, dan G5.

Instans GPU untuk Algoritma Wawasan IP

IP Insights mendukung semua GPU yang tersedia. Jika Anda perlu mempercepat pelatihan, sebaiknya mulai dengan instans GPU tunggal, seperti ml.p3.2xlarge, lalu pindah ke lingkungan multi-GPU, seperti ml.p3.8xlarge dan ml.p3.16xlarge. Multi-GPU secara otomatis membagi batch mini data pelatihan di seluruh diri mereka sendiri. Jika Anda beralih dari satu GPU ke beberapa GPU,`mini_batch_size`dibagi rata ke dalam jumlah GPU yang digunakan. Anda mungkin ingin meningkatkan nilai`mini_batch_size`untuk mengimbangi ini.

Instans CPU untuk Algoritma Wawasan IP

Jenis instance CPU yang kami rekomendasikan sangat bergantung pada memori instans yang tersedia dan ukuran model. Ukuran model ditentukan oleh dua hiperparameter:`vector_dim`dan`num_entity_vectors`. Ukuran model maksimum yang didukung adalah 8 GB. Tabel berikut mencantumkan tipe instans EC2 khas yang akan Anda terapkan berdasarkan parameter input ini untuk berbagai ukuran model. Pada Tabel 1, nilai untuk`vector_dim`dalam rentang kolom pertama dari 32 hingga 2048 dan nilai untuk`num_entity_vectors`di baris pertama berkisar dari 10.000 hingga 50.000.000.

vector_dim \ num_entity_vectors .	10.000	50.000	100.000	500.000	1.000.000	5.000.000	10.000.000	50.000.000
32	db.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.xlarge	ml.m5.2xlarge	ml.m5.4xlarge
64	db.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.2xlarge	ml.m5.2xlarge	

vector_dim \ num_entry_vectors	10.000	50.000	100.000	500.000	1.000.000	5.000.000	10.000.000	50.000.000
128	db.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.2xlarge	ml.m5.4xlarge	
256	db.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.xlarge	ml.m5.4xlarge		
512	db.m5.large	ml.m5.large	ml.m5.large	ml.m5.large	ml.m5.2xlarge			
1024	db.m5.large	ml.m5.large	ml.m5.large	ml.m5.xlarge	ml.m5.4xlarge			
2048	db.m5.large	ml.m5.large	ml.m5.xlarge	ml.m5.xlarge				

Nilainya untuk`mini_batch_size`,`num_ip_encoder_layers`,`random_negative_sampling_rate`, dan`shuffled_negative_sampling_rate`hyperparameters juga mempengaruhi jumlah memori yang dibutuhkan. Jika nilai ini besar, Anda mungkin perlu menggunakan tipe instans yang lebih besar dari biasanya.

Notebook Sampel IP Insights

Untuk contoh notebook yang menunjukkan cara melatih SageMaker Algoritma IP Insights dan melakukan kesimpulan dengannya, lihatPengantar SageMakerAlgoritma Wawasan IP. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, LihatInstans SageMaker Notebook Amazon. Setelah membuat instance notebook, pilihSageMaker Contohtab untuk melihat daftar semua SageMaker contoh. Untuk membuka notebook, pilihGunakantab dan pilihBuat salinan.

Cara Kerja IP Insights

Amazon SageMaker IP Insights adalah algoritma tanpa pengawasan yang mengkonsumsi data yang diamati dalam bentuk pasangan (entitas, alamat IPv4) yang mengaitkan entitas dengan alamat IP. IP Insights menentukan seberapa besar kemungkinan suatu entitas akan menggunakan alamat IP tertentu dengan mempelajari representasi vektor laten untuk entitas dan alamat IP. Jarak antara kedua representasi ini kemudian dapat berfungsi sebagai proxy untuk seberapa besar kemungkinan asosiasi ini.

Algoritma IP Insights menggunakan jaringan saraf untuk mempelajari representasi vektor laten untuk entitas dan alamat IP. Entitas pertama kali di-hash ke ruang hash besar tapi tetap dan kemudian dikodekan oleh lapisan embedding sederhana. String karakter seperti nama pengguna atau ID akun dapat dimasukkan langsung ke IP Insights saat ditampilkan dalam file log. Anda tidak perlu memproses data untuk pengidentifikasi entitas. Anda dapat memberikan entitas sebagai nilai string arbitrer selama pelatihan dan inferensi. Ukuran hash harus dikonfigurasi dengan nilai yang cukup tinggi untuk memastikan bahwa jumlahtabrakan, yang terjadi ketika entitas yang berbeda dipetakan ke vektor laten yang sama, tetap tidak signifikan. Untuk informasi lebih lanjut tentang cara memilih ukuran hash yang sesuai, lihatFitur Hashing untuk Pembelajaran Multitask Skala Besar. Untuk mewakili alamat IP, di sisi lain, IP Insights menggunakan jaringan encoder yang dirancang khusus untuk secara unik mewakili setiap alamat IPv4 yang mungkin dengan memanfaatkan struktur awalan alamat IP.

Selama pelatihan, IP Insights secara otomatis menghasilkan sampel negatif dengan memasangkan entitas dan alamat IP secara acak. Sampel negatif ini mewakili data yang cenderung tidak terjadi pada kenyataannya. Model ini dilatih untuk membedakan antara sampel positif yang diamati dalam data pelatihan dan sampel negatif yang dihasilkan ini. Lebih khusus lagi, model ini dilatih untuk meminimalkanlintas entropi, juga dikenal sebagaiKehilangan log, didefinisikan sebagai berikut:

$$L = \frac{1}{N} \sum_{n} \left[y_n \log p_n + (1 - y_n) \log (1 - p_n) \right]$$

$y_n$adalah label yang menunjukkan apakah sampel berasal dari distribusi nyata yang mengatur data yang diamati (y_n= 1) atau dari distribusi menghasilkan sampel negatif (y_n=0). hal_nadalah probabilitas bahwa sampel berasal dari distribusi nyata, seperti yang diprediksi oleh model.

Menghasilkan sampel negatif adalah proses penting yang digunakan untuk mencapai model akurat dari data yang diamati. Jika sampel negatif sangat tidak mungkin, misalnya, jika semua

alamat IP dalam sampel negatif adalah 10.0.0.0, maka model sepele belajar untuk membedakan sampel negatif dan gagal untuk secara akurat mengkarakterisasi dataset diamati aktual. Untuk menjaga sampel negatif lebih realistis, IP Insights menghasilkan sampel negatif baik dengan menghasilkan alamat IP secara acak dan memilih alamat IP secara acak dari data pelatihan. Anda dapat mengkonfigurasi jenis sampling negatif dan tingkat di mana sampel negatif dihasilkan dengan`random_negative_sampling_rate`dan`shuffled_negative_sampling_rate`hiperparameter.

Mengingat n (entitas, pasangan alamat IP), model IP Insights output askor, S_n, yang menunjukkan seberapa kompatibel entitas dengan alamat IP. Skor ini sesuai dengan rasio log odds untuk (entitas, alamat IP) tertentu dari pasangan yang berasal dari distribusi nyata dibandingkan dengan yang berasal dari distribusi negatif. Ini didefinisikan sebagai berikut:

$$S_n = \log\left(\frac{P_{real}(n)}{P_{neg}(n)}\right)$$

Skor pada dasarnya adalah ukuran kesamaan antara representasi vektor entitas ke-n dan alamat IP. Hal ini dapat ditafsirkan sebagai seberapa besar kemungkinan untuk mengamati peristiwa ini dalam kenyataan daripada dalam dataset yang dihasilkan secara acak. Selama pelatihan, algoritma menggunakan skor ini untuk menghitung perkiraan probabilitas sampel yang berasal dari distribusi nyata, hal_n, untuk digunakan dalam minimisasi entropi lintas, di mana:

$$p_n = \frac{1}{1 + e^{-S_n}}$$

Hyperparameters Wawasan IP

Di`CreateTransformJob`permintaan, Anda menentukan algoritma pelatihan. Anda juga dapat menentukan hyperparameters khusus algoritme sebagai string-to-string peta. Tabel berikut menjelaskan hyperparameters untuk Amazon SageMaker Algoritma Wawasan IP.

Nama Parameter	Deskripsi
`num_entity_vectors`	Jumlah representasi vektor entitas (entitas embedding vektor) untuk melatih. Setiap entitas dalam set pelatihan ditugaskan secara acak ke salah satu vektor ini menggunakan fungsi hash. Karena tabrakan hash,

Nama Parameter	Deskripsi
	mungkin ada beberapa entitas yang ditugaskan ke vektor yang sama. Hal ini akan menyebabkan vektor yang sama untuk mewakili beberapa entitas. Ini umumnya memiliki efek yang dapat diabaikan pada kinerja model, selama tingkat tabrakan tidak terlalu parah. Untuk menjaga tingkat tabrakan tetap rendah, tetapkan nilai ini setinggi mungkin. Namun, ukuran model, dan, oleh karena itu, persyaratan memori, untuk pelatihan dan inferensi, skala secara linear dengan hiperparameter ini. Sebaiknya tetapkan nilai ini menjadi dua kali jumlah pengenal entitas unik. Wajib Nilai yang valid: 1 ≤ bilangan bulat positif ≤ 250.000.000
`vector_dim`	Ukuran vektor embedding untuk mewakili entitas dan alamat IP. Semakin besar nilainya, semakin banyak informasi yang dapat dikodekan menggunak an representasi ini. Dalam praktiknya, ukuran model skala linier dengan parameter ini dan membatasi seberapa besar dimensinya. Selain itu, menggunakan representasi vektor yang terlalu besar dapat menyebabk an model terlalu pas, terutama untuk kumpulan data pelatihan kecil. Overfitting terjadi ketika model tidak mempelajari pola apa pun dalam data tetapi secara efektif menghafal data pelatihan dan, oleh karena itu, tidak dapat menggeneralisasi dengan baik dan berkinerj a buruk selama inferensi. Nilai yang disarankan adalah 128. Wajib Nilai yang valid: 4 ≤ bilangan bulat positif ≤ 4096

Nama Parameter	Deskripsi
`batch_metrics_publish_interval`	Interval (setiap batch X) di mana fungsi Apache MXNet Speedometer mencetak kecepatan pelatihan jaringan (sampel/detik). Opsional Nilai yang valid: bilangan bulat positif ≥ 1 Nilai default: 1.000
`epochs`	Jumlah melewati data pelatihan. Nilai optimal tergantung pada ukuran data dan tingkat pembelajaran Anda. Nilai tipikal berkisar dari 5 hingga 100. Opsional Nilai yang valid: bilangan bulat positif ≥ 1 Nilai default: 10
`learning_rate`	Tingkat pembelajaran untuk pengoptimal. IP Insights menggunakan gradient-descent-based Adam optimizer . Tingkat pembelajaran secara efektif mengontrol ukuran langkah untuk memperbarui parameter model pada setiap iterasi. Tingkat pembelajaran yang terlalu besar dapat menyebabkan model menyimpang karena pelatihan cenderung melampaui minima. Di sisi lain, tingkat pembelajaran yang terlalu kecil memperlambat konvergensi. Nilai tipikal berkisar dari 1e-4 hingga 1e-1. Opsional Nilai yang valid: 1e-6 ≤ float ≤ 10.0 Nilai default: 0.001

Nama Parameter	Deskripsi
`mini_batch_size`	Jumlah contoh di setiap batch mini. Prosedur pelatihan memproses data dalam batch mini. Nilai optimal tergantung pada jumlah pengidentifikasi akun unik dalam kumpulan data. Secara umum, semakin besar`mini_batch_size` , semakin cepat pelatihan dan semakin besar jumlah yang mungkin shuffled-negative-sample kombinasi. Namun, dengan`mini_batch_size` , pelatihan ini lebih cenderung bertemu dengan minimum lokal yang buruk dan tampil relatif lebih buruk untuk kesimpulan. Opsional Nilai yang valid: 1 ≤ bilangan bulat positif ≤ 500000 Nilai default: 10.000
`num_ip_encoder_layers`	Jumlah lapisan yang terhubung sepenuhnya digunakan untuk menyandikan embedding alamat IP. Semakin besar jumlah layer, semakin besar kapasitas model untuk menangkap pola di antara alamat IP. Namun, menggunakan sejumlah besar lapisan meningkatkan kemungkinan overfitting. Opsional Nilai yang valid: 0 ≤ bilangan bulat positif ≤ 100 Nilai default: 1

Nama Parameter	Deskripsi
`random_negative_sampling_rate`	Jumlah sampel negatif acak, R, untuk menghasilkan per contoh masukan. Prosedur pelatihan bergantung pada sampel negatif untuk mencegah representasi vektor model runtuh ke satu titik. Pengambilan sampel negatif acak menghasilkan R alamat IP acak untuk setiap akun input dalam batch mini. Jumlah dari `random_negative_sampling_rate` (R) dan `shuffled_negative_sampling_rate` (S) harus dalam interval: 1 ≤ R+S ≤ 500. Opsional Nilai yang valid: 0 ≤ bilangan bulat positif ≤ 500 Nilai default: 1
`shuffled_negative_sampling_rate`	Jumlah sampel negatif dikocokkan, S, untuk menghasilkan per contoh masukan. Dalam beberapa kasus, ada baiknya menggunakan sampel negatif yang lebih realistis yang diambil secara acak dari data pelatihan itu sendiri. Sampling negatif semacam ini dicapai dengan menyeret data dalam batch mini. Sampling negatif yang dikocokkan menghasilkan alamat IP negatif S dengan menyeret alamat IP dan pasangan akun dalam batch mini. Jumlah dari `random_negative_sampling_rate` (R) dan `shuffled_negative_sampling_rate` (S) harus dalam interval: 1 ≤ R+S ≤ 500. Opsional Nilai yang valid: 0 ≤ bilangan bulat positif ≤ 500 Nilai default: 1

Nama Parameter	Deskripsi
`weight_decay`	Koefisien peluruhan berat. Parameter ini menambahkan faktor regularisasi L2 yang diperlukan untuk mencegah model dari overfitting data pelatihan. Opsional Nilai yang valid: 0,0 ≤ float ≤ 10.0 Nilai default: 0.00001

Setel Model Wawasan IP

Penalaan model otomatis, juga disebut hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi lebih lanjut tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma IP Insights

Amazon SageMaker Algoritma IP Insights adalah algoritma pembelajaran tanpa pengawasan yang mempelajari hubungan antara alamat IP dan entitas. Algoritma melatih model diskriminator, yang belajar memisahkan titik data yang diamati (Sampel positif) dari titik data yang dihasilkan secara acak (Sampel negatif). Penyetelan model otomatis pada IP Insights membantu Anda menemukan model yang paling akurat membedakan antara data validasi yang tidak berlabel dan sampel negatif yang dihasilkan secara otomatis. Akurasi model pada dataset validasi diukur dengan area di bawah kurva karakteristik operasi penerima. Ini`validation:discriminator_auc`metrik dapat mengambil nilai antara 0,0 dan 1,0, di mana 1,0 menunjukkan akurasi sempurna.

Algoritma IP Insights menghitung`validation:discriminator_auc`metrik selama validasi, nilai yang digunakan sebagai fungsi obyektif untuk mengoptimalkan penyetelan hyperparameter.

Nama Metrik	Deskripsi	Arah optimalisasi
`validation:discriminator_auc`	Area di bawah kurva karakteristik operasi penerima pada dataset validasi. Dataset validasi tidak diberi label. Area Under the Curve (AUC) adalah metrik yang menggambarkan kemampuan model untuk membedakan titik data validasi dari titik data yang dihasilkan secara acak.	Maksimalkan

Hyperparameter Wawasan IP Merdu

Anda dapat menyetel hyperparameter berikut untuk SageMaker Algoritma Wawasan IP.

Nama Parameter	Tipe Parameter	Rentang Direkomendasikan
`epochs`	IntegerParameterRange	MinValue: 1, MaxValue: 100
`learning_rate`	ContinuousParameterRange	MinValue: 1e-4, MaxValue: 0,1
`mini_batch_size`	IntegerParameterRanges	MinValue: 100, MaxValue: 50000
`num_entity_vectors`	IntegerParameterRanges	MinValue: 10000, MaxValue: 1000000
`num_ip_encoder_layers`	IntegerParameterRanges	MinValue: 1, MaxValue: 10
`random_negative_sampling_rate`	IntegerParameterRanges	MinValue: 0, MaxValue: 10

Nama Parameter	Tipe Parameter	Rentang Direkomendasikan
`shuffled_negative_sampling_rate`	IntegerParameterRanges	MinValue: 0, MaxValue: 10
`vector_dim`	IntegerParameterRanges	MinValue: 8, MaxValue: 256
`weight_decay`	ContinuousParameterRange	MinValue: 0,0, MaxValue: 1.0

Format Data Wawasan IP

Bagian ini memberikan contoh format data input dan output yang tersedia yang digunakan oleh algoritma IP Insights selama pelatihan dan inferensi.

Topik

- Format Data Pelatihan Wawasan IP
- Format Data Inferensi Wawasan IP

Format Data Pelatihan Wawasan IP

Berikut ini adalah format input data yang tersedia untuk algoritma IP Insights. Amazon SageMaker algoritma bawaan mematuhi format pelatihan input umum yang dijelaskan dalamFormat Data Umum untuk Pelatihan. Namun, SageMakerAlgoritma IP Insights saat ini hanya mendukung format input data CSV.

Format Input Data Pelatihan Wawasan IP

INPUT: CSV

File CSV harus memiliki dua kolom. Kolom pertama adalah string buram yang sesuai dengan identifier unik entitas. Kolom kedua adalah alamat IPv4 acara akses entitas dalam notasi desimal-dot.

konten-jenis: teks/csv

```
entity_id_1, 192.168.1.2
```

```
entity_id_2, 10.10.1.2
```

Format Data Inferensi Wawasan IP

Berikut ini adalah format input dan output yang tersedia untuk algoritma IP Insights. Amazon SageMaker algoritma bawaan mematuhi format inferensi input umum yang dijelaskan dalamFormat Data Umum untuk Inferensi. Namun, SageMaker Algoritma IP Insights saat ini tidak mendukung format RecordiO.

Format Permintaan Input Wawasan IP

INPUT: Format CSV

File CSV harus memiliki dua kolom. Kolom pertama adalah string buram yang sesuai dengan identifier unik entitas. Kolom kedua adalah alamat IPv4 acara akses entitas dalam notasi desimal-dot.

konten-jenis: teks/csv

```
entity_id_1, 192.168.1.2
entity_id_2, 10.10.1.2
```

INPUT: Format JSON

Data JSON dapat diberikan dalam format yang berbeda. IP Insights mengikuti umum SageMaker format format. Untuk informasi lebih lanjut tentang format inferensi, lihatFormat Data Umum untuk Inferensi.

konten-jenis: aplikasi/json

```
{
  "instances": [
    {"data": {"features": {"values": ["entity_id_1", "192.168.1.2"]}}},
    {"features": ["entity_id_2", "10.10.1.2"]}
  ]
}
```

INPUT: Format JSONLINES

Jenis konten JSON Lines berguna untuk menjalankan pekerjaan transformasi batch. Untuk informasi lebih lanjut tentang SageMaker format inferensi, lihatFormat Data Umum untuk Inferensi. Untuk informasi lebih lanjut tentang menjalankan tugas transformasi batch, lihatGunakan Batch Transform.

konten-jenis: aplikasi/jsonlines

```
{"data": {"features": {"values": ["entity_id_1", "192.168.1.2"]}}},
{"features": ["entity_id_2", "10.10.1.2"]}]
```

Format Respons Output Wawasan IP

OUTPUT OUTPUT: Format Respons JSON

Output default-nya SageMaker Algoritma IP Insights adalah`dot_product`antara entitas input dan alamat IP. The dot_product menandakan seberapa kompatibel model mempertimbangkan entitas dan alamat IP. Klaster`dot_product`Tidak terbatas. Untuk membuat prediksi tentang apakah suatu peristiwa anomali, Anda perlu menetapkan ambang batas berdasarkan distribusi yang ditentukan. Untuk informasi tentang cara menggunakan`dot_product`untuk deteksi anomali, lihatPengantar SageMakerAlgoritma Wawasan IP.

menerima: aplikasi/json

```
{
  "predictions": [
    {"dot_product": 0.0},
    {"dot_product": 2.0}
  ]
}
```

Pengguna tingkat lanjut dapat mengakses entitas yang dipelajari model dan embeddings IP dengan menyediakan parameter tipe konten tambahan`verbose=True`ke judul Terima. Anda dapat menggunakan`entity_embedding`dan`ip_embedding`untuk debugging, visualisasi, dan pemahaman model. Selain itu, Anda dapat menggunakan embeddings ini dalam teknik pembelajaran mesin lainnya, seperti klasifikasi atau pengelompokan.

menerima: application/json; Verbose = true

```
{
  "predictions": [
    {
        "dot_product": 0.0,
        "entity_embedding": [1.0, 0.0, 0.0],
        "ip_embedding": [0.0, 1.0, 0.0]
    },
```

```
    {
        "dot_product": 2.0,
        "entity_embedding": [1.0, 0.0, 1.0],
        "ip_embedding": [1.0, 0.0, 1.0]
    }
  ]
}
```

OUTPUT OUTPUT: Format Respons JSONLINES

menerima: aplikasi/jsonlines

```
{"dot_product": 0.0}
{"dot_product": 2.0}
```

menerima: aplikasi/jsonlines; Verbose = true

```
{"dot_product": 0.0, "entity_embedding": [1.0, 0.0, 0.0], "ip_embedding": [0.0, 1.0,
 0.0]}
{"dot_product": 2.0, "entity_embedding": [1.0, 0.0, 1.0], "ip_embedding": [1.0, 0.0,
 1.0]}
```

Algoritma K-Means

K-Means adalah algoritma pembelajaran tanpa pengawasan. Ini mencoba untuk menemukan pengelompokan diskrit dalam data, di mana anggota kelompok semirip mungkin satu sama lain dan berbeda mungkin dari anggota kelompok lain. Anda menentukan atribut yang Anda inginkan algoritma untuk digunakan untuk menentukan kesamaan.

Amazon SageMaker menggunakan versi modifikasi dari algoritma pengelompokan k-means skala web. Dibandingkan dengan versi asli dari algoritma, versi yang digunakan oleh Amazon SageMaker lebih akurat. Seperti algoritme asli, algoritme ini menskalakan ke kumpulan data besar dan memberikan peningkatan dalam waktu pelatihan. Untuk melakukan ini, versi yang digunakan oleh Amazon SageMaker mengalirkan mini-batch (subset kecil dan acak) dari data pelatihan. Untuk informasi selengkapnya tentang mini-batch k-means, lihatSkala web k-berarti Clustering.

Algoritma k-mean mengharapkan data tabular, di mana baris mewakili pengamatan yang ingin Anda kelompokkan, dan kolom mewakili atribut pengamatan. Thenatribut di setiap baris mewakili titik din-ruang dimensi. Jarak Euclidean antara titik-titik ini mewakili kesamaan pengamatan yang sesuai. Algoritma mengelompokkan pengamatan dengan nilai atribut yang serupa (titik-titik yang sesuai

dengan pengamatan ini lebih dekat satu sama lain). Untuk informasi lebih lanjut tentang cara kerja k-means di Amazon SageMaker, lihatBagaimana K-Means Clustering Bekerja.

Topik

Antarmuka Input/Output untuk Algoritma K-Means

Untuk pelatihan, algoritma k-means mengharapkan data disediakan dimelatihsaluran (direkomendasikan`S3DataDistributionType=ShardedByS3Key`), dengan opsionalujisaluran (direkomendasikan`S3DataDistributionType=FullyReplicated`) untuk mencetak data pada. Keduanya`recordIO-wrapped-protobuf`dan`CSV`format didukung untuk pelatihan. Anda dapat menggunakan mode File atau mode Pipa untuk melatih model pada data yang diformat sebagai`recordIO-wrapped-protobuf`atau sebagai`CSV`.

Untuk inferensi,`text/csv`,`application/json`, dan`application/x-recordio-protobuf`didukung. k-means mengembalikan a`closest_cluster`label dan`distance_to_cluster`untuk setiap observasi.

Untuk informasi selengkapnya tentang format file input dan output, lihatFormat Respons K-Meansuntuk kesimpulan danNotebook Contoh K-Means. Algoritma k-means tidak mendukung pembelajaran instance ganda, di mana set pelatihan terdiri dari “tas” berlabel, yang masing-masing merupakan kumpulan instance yang tidak berlabel.

Rekomendasi Instans EC2 untuk Algoritma K-Means

Kami merekomendasikan pelatihan k-means pada instance CPU. Anda dapat melatih instans GPU, tetapi harus membatasi pelatihan GPU ke instans GPU tunggal (seperti ml.g4dn.xlarge) karena hanya satu GPU yang digunakan per instance. Algoritma k-means mendukung instance P2, P3, G4dn, dan G5 untuk pelatihan dan inferensi.

Notebook Contoh K-Means

Untuk contoh notebook yang menggunakan SageMaker Algoritma K-means untuk mengelompokkan populasi kabupaten di Amerika Serikat berdasarkan atribut yang diidentifikasi menggunakan analisis komponen prinsip, lihatAnalisis data sensus AS untuk segmentasi populasi menggunakan Amazon SageMaker. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMakerContohtab untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, klikGunakantab dan pilihBuat salinan.

Bagaimana K-Means Clustering Bekerja

K-means adalah algoritma yang melatih model yang mengelompokkan objek serupa bersama-sama. Algoritma k-mean menyelesaikan ini dengan memetakan setiap pengamatan dalam dataset input ke suatu titik din-ruang dimensi (di mananadalah jumlah atribut pengamatan). Misalnya, kumpulan data Anda mungkin berisi pengamatan suhu dan kelembaban di lokasi tertentu, yang dipetakan ke titik (t, h) dalam ruang 2 dimensi.

> **Note**
>
> Algoritma pengelompokan tidak diawasi. Dalam pembelajaran tanpa pengawasan, label yang mungkin terkait dengan objek dalam kumpulan data pelatihan tidak digunakan. Untuk informasi selengkapnya, lihat Pembelajaran tanpa pengawasan.

Dalam k-means clustering, setiap cluster memiliki pusat. Selama pelatihan model, algoritma k-mean menggunakan jarak titik yang sesuai dengan setiap pengamatan dalam kumpulan data ke pusat cluster sebagai dasar pengelompokan. Anda memilih jumlah cluster (k) untuk membuat.

Misalnya, anggaplah Anda ingin membuat model untuk mengenali digit tulisan tangan dan Anda memilih kumpulan data MNIST untuk pelatihan. DATASET menyediakan ribuan gambar dari digit tulisan tangan (0 hingga 9). Dalam contoh ini, Anda dapat memilih untuk membuat 10 cluster, satu untuk setiap digit (0, 1,..., 9). Sebagai bagian dari pelatihan model, algoritma k-means mengelompokkan gambar input ke dalam 10 cluster.

Setiap gambar dalam dataset MNIST adalah gambar 28x28-piksel, dengan total 784 piksel. Setiap gambar sesuai dengan titik dalam ruang 784 dimensi, mirip dengan titik dalam ruang 2 dimensi (x, y). Untuk menemukan kluster di mana suatu titik berada, algoritma k-means menemukan jarak titik itu

dari semua pusat cluster. Kemudian memilih cluster dengan pusat terdekat sebagai cluster tempat gambar berada.

> **Note**
>
> Amazon SageMaker menggunakan versi algoritma yang disesuaikan di mana, alih-alih menentukan bahwa algoritme membuatkcluster, Anda dapat memilih untuk meningkatkan akurasi model dengan menentukan pusat cluster tambahan(K = k*x). Namun, algoritme pada akhirnya mengurangi ini menjadikkluster.

Di SageMaker, Anda menentukan jumlah cluster saat membuat pekerjaan pelatihan. Untuk informasi selengkapnya, lihat `CreateTrainingJob`. Di badan permintaan, Anda menambahkan`HyperParameters`peta string untuk menentukan`k`dan`extra_center_factor`string.

Berikut ini adalah ringkasan tentang bagaimana k-means bekerja untuk pelatihan model di SageMaker:

1. Ini menentukan awal`K`pusat cluster.

   > **Note**
   >
   > Dalam topik-topik berikut,`K`cluster mengacu pada`k * x`, di mana Anda menentukan`k`dan`x`saat membuat pekerjaan pelatihan model.

2. Ini mengulangi data pelatihan input dan menghitung ulang pusat cluster.
3. Ini mengurangi cluster yang dihasilkan menjadik(jika ilmuwan data menentukan penciptaan`k*x`cluster dalam permintaan).

Bagian berikut juga menjelaskan beberapa parameter yang mungkin ditentukan oleh ilmuwan data untuk mengonfigurasi pekerjaan pelatihan model sebagai bagian dari`HyperParameters`peta string.

Topik

- Langkah 1: Tentukan Pusat klasternya
- Langkah 2: Ulangi Dataset Pelatihan dan Hitung Pusat Cluster
- Langkah 3: Kurangi Cluster dari`K`kepada`k`

Langkah 1: Tentukan Pusat klasternya

Saat menggunakan k-means di SageMaker, pusat cluster awal dipilih dari pengamatan dalam batch kecil sampel acak. Pilih salah satu strategi berikut untuk menentukan bagaimana pusat klaster awal ini dipilih:

- Pendekatan acak — Pilih secara acakKpengamatan dalam dataset input Anda sebagai pusat cluster. Misalnya, Anda dapat memilih pusat cluster yang menunjuk ke ruang 784 dimensi yang sesuai dengan 10 gambar dalam kumpulan data pelatihan MNIST.
- Pendekatan k-means++, yang berfungsi sebagai berikut:
  1. Mulailah dengan satu cluster dan tentukan pusatnya. Anda secara acak memilih observasi dari dataset pelatihan Anda dan menggunakan titik yang sesuai dengan pengamatan sebagai pusat cluster. Misalnya, dalam kumpulan data MNIST, pilih gambar digit tulisan tangan secara acak. Kemudian pilih titik dalam ruang 784 dimensi yang sesuai dengan gambar sebagai pusat cluster Anda. Ini adalah cluster center 1.
  2. Tentukan pusat untuk cluster 2. Dari pengamatan yang tersisa dalam dataset pelatihan, pilih pengamatan secara acak. Pilih salah satu yang berbeda dari yang Anda pilih sebelumnya. Pengamatan ini sesuai dengan titik yang jauh dari pusat cluster 1. Menggunakan dataset MNIST sebagai contoh, Anda melakukan hal berikut:
     - Untuk setiap gambar yang tersisa, cari jarak titik yang sesuai dari pusat cluster 1. Kuadratkan jarak dan tetapkan probabilitas yang sebanding dengan kuadrat jarak. Dengan begitu, gambar yang berbeda dari yang Anda pilih sebelumnya memiliki probabilitas lebih tinggi untuk dipilih sebagai pusat cluster 2.
     - Pilih salah satu gambar secara acak, berdasarkan probabilitas yang ditetapkan pada langkah sebelumnya. Titik yang sesuai dengan gambar adalah pusat cluster 2.
  3. Ulangi Langkah 2 untuk menemukan pusat cluster 3. Kali ini, cari jarak gambar yang tersisa dari pusat cluster 2.
  4. Ulangi proses ini sampai Anda memilikiKpusat cluster.

Untuk melatih model di SageMaker, Anda membuat pekerjaan pelatihan. Dalam permintaan, Anda memberikan informasi konfigurasi dengan menentukan hal berikut`HyperParameters`peta string:

- Untuk menentukan jumlah cluster yang akan dibuat, tambahkanktali.
- Untuk akurasi yang lebih besar, tambahkan opsional`extra_center_factor`tali.

- Untuk menentukan strategi yang ingin Anda gunakan untuk menentukan pusat cluster awal, tambahkan`init_method`string dan tetapkan nilainya ke`random`atau`k-means++`.

Untuk informasi lebih lanjut tentang SageMaker k-means estimator, lihatK-berartidiAmazon SageMaker Python SDKdokumentasi.

Anda sekarang memiliki satu set awal pusat cluster.

Langkah 2: Ulangi Dataset Pelatihan dan Hitung Pusat Cluster

Pusat cluster yang Anda buat pada langkah sebelumnya sebagian besar acak, dengan beberapa pertimbangan untuk dataset pelatihan. Pada langkah ini, Anda menggunakan kumpulan data pelatihan untuk memindahkan pusat-pusat ini menuju pusat cluster yang sebenarnya. Algoritma mengulangi dataset pelatihan, dan menghitung ulang`K`pusat cluster.

1. Baca sejumlah kecil pengamatan (subset kecil yang dipilih secara acak dari semua catatan) dari kumpulan data pelatihan dan lakukan hal berikut.

   > **Note**
   >
   > Saat membuat pekerjaan pelatihan model, Anda menentukan ukuran batch dimini_batch_sizestring di`HyperParameters`peta string.

   a. Tetapkan semua pengamatan dalam batch mini ke salah satu cluster dengan pusat cluster terdekat.

   b. Hitung jumlah pengamatan yang ditugaskan untuk setiap cluster. Kemudian, hitung proporsi poin baru yang ditetapkan per cluster.

      Misalnya, pertimbangkan klaster-klasternya:

      Cluster c1 = 100 poin yang ditetapkan sebelumnya. Anda menambahkan 25 poin dari mini-batch di langkah ini.

      Cluster c2 = 150 poin yang ditetapkan sebelumnya. Anda menambahkan 40 poin dari mini-batch di langkah ini.

      Cluster c3 = 450 poin yang ditetapkan sebelumnya. Anda menambahkan 5 poin dari mini-batch di langkah ini.

Hitung proporsi poin baru yang ditetapkan untuk masing-masing cluster sebagai berikut:

```
p1 = proportion of points assigned to c1 = 25/(100+25)
p2 = proportion of points assigned to c2 = 40/(150+40)
p3 = proportion of points assigned to c3 = 5/(450+5)
```

c. Hitung pusat poin baru yang ditambahkan ke setiap cluster:

```
d1 = center of the new points added to cluster 1
d2 = center of the new points added to cluster 2
d3 = center of the new points added to cluster 3
```

d. Hitung rata-rata tertimbang untuk menemukan pusat cluster yang diperbarui sebagai berikut:

```
Center of cluster 1 = ((1 - p1) * center of cluster 1) + (p1 * d1)
Center of cluster 2 = ((1 - p2) * center of cluster 2) + (p2 * d2)
Center of cluster 3 = ((1 - p3) * center of cluster 3) + (p3 * d3)
```

2. Baca mini-batch berikutnya, dan ulangi Langkah 1 untuk menghitung ulang pusat cluster.
3. Untuk informasi lebih lanjut tentang mini-batchk-berarti, lihatSkala web k-berarti Clustering).

Langkah 3: Kurangi Cluster dariKkepadak

Jika algoritma dibuatKkluster—(K = k*x)di manaxlebih besar dari 1 — maka mengurangiKcluster untukkkluster. (Untuk informasi lebih lanjut, lihat`extra_center_factor`dalam diskusi sebelumnya.) Hal ini dilakukan dengan menerapkan metode Lloyd dengan`k-means++`inisialisasi keKpusat cluster. Untuk informasi lebih lanjut tentang metode Lloyd, lihatk-berarti pengelompokan.

K-Berarti Hyperparameter

Di`CreateTrainingJob`permintaan, Anda menentukan algoritma pelatihan yang ingin Anda gunakan. Anda juga dapat menentukan hiperparameter khusus algoritme sebagai string-to-string peta. Tabel berikut mencantumkan hyperparameters untuk algoritma pelatihan k-means yang disediakan oleh Amazon SageMaker. Untuk informasi lebih lanjut tentang cara kerja klasterlah, lihatBagaimana K-Means Clustering Bekerja.

Nama Parameter	Deskripsi
`feature_dim`	Jumlah fitur dalam data input.

Nama Parameter	Deskripsi
	Diperlukan Nilai yang valid: Bilangan bulat positif
k	Jumlah cluster yang dibutuhkan. Diperlukan Nilai yang valid: Bilangan bulat positif
epochs	Jumlah pass yang dilakukan atas data pelatihan. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 1
eval_metrics	Daftar tipe metrik JSON yang digunakan untuk melaporkan skor untuk model. Nilai yang diizinkan adalah`msd`untuk Means Square Deviation dan`ssd`untuk Jumlah Jarak Persegi. Jika data pengujian disediakan, skor dilaporkan untuk setiap metrik yang diminta. Opsional Nilai yang valid: Baik`[\"msd\"]` atau`[\"ssd\"]` atau`[\"msd\",\"ssd\"]` . Nilai default: `[\"msd\"]`
extra_center_factor	Algoritma menciptakan K center =`num_clusters *extra_center_factor` saat berjalan dan mengurangi jumlah pusat dari K keksaat menyelesaikan model. Opsional Nilai yang valid: Entah bilangan bulat positif atau`auto`. Nilai default: `auto`

Nama Parameter	Deskripsi
`half_life_time_size`	Digunakan untuk menentukan bobot yang diberikan untuk pengamatan saat menghitung rata-rata cluster. Bobot ini meluruh secara eksponensial karena lebih banyak titik diamati. Ketika suatu titik pertama kali diamati, itu diberi bobot 1 saat menghitung rata-rata cluster. Konstanta peluruhan untuk fungsi peluruhan eksponensial dipilih sehingga setelah diamati`half_life_time_size` poin, beratnya 1/2. Jika diatur ke 0, tidak ada pembusukan. Opsional Nilai yang valid: Bilangan bulat non-negatif Nilai default: 0
`init_method`	Metode dimana algoritma memilih pusat cluster awal. Pendekatan k-means standar memilihnya secara acak. Metode alternatif k-means++ memilih pusat cluster pertama secara acak. Kemudian menyebar posisi cluster awal yang tersisa dengan menimbang pemilihan pusat dengan distribusi probabilitas yang sebanding dengan kuadrat jarak titik data yang tersisa dari pusat yang ada. Opsional Nilai yang valid: Baik`random`atau`kmeans++`. Nilai default: `random`
`local_lloyd_init_method`	Metode inisialisasi untuk prosedur ekspektasi-maksimisasi (EM) Lloyd yang digunakan untuk membangun model akhir yang berisikpusat. Opsional Nilai yang valid: Baik`random`atau`kmeans++`. Nilai default: `kmeans++`

Nama Parameter	Deskripsi
`local_lloyd_max_iter`	Jumlah maksimum iterasi untuk prosedur ekspektasi-maksima lisasi (EM) Lloyd yang digunakan untuk membangun model akhir yang berisikpusat. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 300
`local_lloyd_num_trials`	Berapa kali prosedur ekspektasi-maksimisasi (EM) Lloyd dengan kerugian paling sedikit dijalankan saat membangun model akhir yang berisikpusat. Opsional Nilai yang valid: Entah bilangan bulat positif atau`auto`. Nilai default: `auto`
`local_lloyd_tol`	Toleransi untuk perubahan kerugian untuk penghentian dini prosedur ekspektasi-maksimisasi (EM) Lloyd yang digunakan untuk membangun model akhir yang mengandungkpusat. Opsional Nilai yang valid: Float. Rentang dalam [0, 1]. Nilai default: 0,0001
`mini_batch_size`	Jumlah pengamatan per mini-batch untuk iterator data. Opsional Nilai yang valid: Bilangan bulat positif Nilai default: 5000

Menyetel Model K-Means

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameters yang dapat disetel, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hiperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Amazon SageMaker Algoritma k-means adalah algoritma tanpa pengawasan yang mengelompokkan data ke dalam cluster yang anggotanya semirip mungkin. Karena tidak diawasi, ia tidak menggunakan kumpulan data validasi yang dapat dioptimalkan oleh hiperparameter. Tetapi dibutuhkan kumpulan data pengujian dan memancarkan metrik yang bergantung pada jarak kuadrat antara titik data dan centroid cluster terakhir di akhir setiap latihan. Untuk menemukan model yang melaporkan cluster terketat pada kumpulan data pengujian, Anda dapat menggunakan pekerjaan penyetelan hiperparameter. Cluster mengoptimalkan kesamaan anggotanya.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik yang Dihitung oleh Algoritma K-Means

Algoritma k-means menghitung metrik berikut selama pelatihan. Saat menyetel model, pilih salah satu metrik ini sebagai metrik objektif.

Nama Metrik	Deskripsi	Arah Optimasi
`test:msd`	Jarak kuadrat rata-rata antara setiap catatan dalam set uji dan pusat terdekat model.	Minimalkan
`test:ssd`	Jumlah jarak kuadrat antara setiap catatan dalam set uji dan pusat terdekat model.	Minimalkan

Hiperparameter K-Means yang Dapat Disetel

Setel Amazon SageMaker k-means model dengan hyperparameter berikut. Hiperparameter yang memiliki dampak terbesar pada metrik objektif k-means adalah:`mini_batch_size`,`extra_center_factor`, dan`init_method`. Menyetel hiperparameter`epochs`umumnya menghasilkan perbaikan kecil.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`epochs`	IntegerParameterRanges	MinValue: 1, MaxValue:10
`extra_center_factor`	IntegerParameterRanges	MinValue: 4, MaxValue:10
`init_method`	CategoricalParameterRanges	['kmeans++', 'acak']
`mini_batch_size`	IntegerParameterRanges	MinValue: 3000, MaxValue:15000

Format Respons K-Means

Semua SageMaker algoritma bawaan mematuhi format inferensi input umum yang dijelaskan dalamFormat Data Umum - Inferensi. Topik ini berisi daftar format keluaran yang tersedia untuk SageMaker algoritma k-means.

Format Respons JSON

```
{
    "predictions": [
        {
            "closest_cluster": 1.0,
            "distance_to_cluster": 3.0,
        },
        {
            "closest_cluster": 2.0,
            "distance_to_cluster": 5.0,
        },

        ....
    ]
}
```

Format Respons JSONLINES

```
{"closest_cluster": 1.0, "distance_to_cluster": 3.0}
```

```
{"closest_cluster": 2.0, "distance_to_cluster": 5.0}
```

Format Respons RECORDIO

```
[
    Record = {
        features = {},
        label = {
            'closest_cluster': {
                keys: [],
                values: [1.0, 2.0]  # float32
            },
            'distance_to_cluster': {
                keys: [],
                values: [3.0, 5.0]  # float32
            },
        }
    }
]
```

Format Respons CSV

Nilai pertama di setiap baris sesuai dengan`closest_cluster`.

Nilai kedua di setiap baris sesuai dengan`distance_to_cluster`.

```
1.0,3.0
2.0,5.0
```

Algoritma Analisis Komponen Utama (PCA)

PCA adalah algoritma pembelajaran mesin tanpa pengawasan yang mencoba mengurangi dimensi (jumlah fitur) dalam kumpulan data sambil tetap mempertahankan informasi sebanyak mungkin. Hal ini dilakukan dengan menemukan serangkaian fitur baru yang disebutkomponen, yang merupakan komposit dari fitur asli yang tidak berkorelasi satu sama lain. Mereka juga dibatasi sehingga komponen pertama menyumbang variabilitas terbesar dalam data, komponen kedua variabilitas paling kedua, dan sebagainya.

Di Amazon SageMaker, PCA beroperasi dalam dua mode, tergantung pada skenario:

- teratur: Untuk dataset dengan data yang jarang dan jumlah pengamatan dan fitur yang moderat.

- acak: Untuk dataset dengan sejumlah besar pengamatan dan fitur. Mode ini menggunakan algoritma aproksimasi.

PCA menggunakan data tabular.

Baris mewakili pengamatan yang ingin Anda sematkan dalam ruang dimensi yang lebih rendah. Kolom mewakili fitur yang ingin Anda temukan perkiraan yang dikurangi. Algoritma menghitung matriks kovarians (atau perkiraan daripadanya dengan cara terdistribusi), dan kemudian melakukan dekomposisi nilai tunggal pada ringkasan ini untuk menghasilkan komponen utama.

Topik

Antarmuka Input/Output untuk Algoritma PCA

Untuk pelatihan, PCA mengharapkan data yang disediakan di saluran kereta api, dan secara opsional mendukung kumpulan data yang diteruskan ke set data pengujian, yang dinilai oleh algoritme akhir. Keduanya`recordIO-wrapped-protobuf`dan`CSV`Format yang didukung untuk pelatihan. Anda dapat menggunakan mode File atau mode Pipa untuk melatih model pada data yang diformat sebagai`recordIO-wrapped-protobuf`atau sebagai`CSV`.

Untuk inferensi, dukungan PCA`text/csv`,`application/json`, dan`application/x-recordio-protobuf`. Hasil dikembalikan di salah satu`application/json`atau`application/x-recordio-protobuf`format dengan vektor “proyeksi.”

Untuk informasi lebih lanjut tentang format file input dan output, lihatFormat Responsuntuk kesimpulan danNotebooks PCA.

Rekomendasi Instans EC2 untuk Algoritma PCA

PCA mendukung instans CPU dan GPU untuk pelatihan dan inferensi. Jenis instans mana yang paling berkinerja sangat bergantung pada spesifikasi data input. Untuk instans GPU, PCA mendukung P2, P3, G4dn, dan G5.

Notebooks PCA

Untuk contoh notebook yang menunjukkan cara menggunakan SageMaker Algoritma Analisis Komponen Utama untuk menganalisis gambar digit tulisan tangan dari nol hingga sembilan dalam kumpulan data MNIST, lihatPengantar PCA dengan MNIST. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, LihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMaker Contohtab untuk melihat daftar semua SageMaker Sampel. Contoh pemodelan topik notebook menggunakan algoritma NTM terletak diPengantar algoritmeBagian. Untuk membuka notebook, klik nyaGunakantab dan pilihBuat salinan.

Cara PCA Berfungsi

Principal Component Analysis (PCA) adalah algoritma pembelajaran yang mengurangi dimensi (jumlah fitur) dalam dataset sambil tetap mempertahankan informasi sebanyak mungkin.

PCA mengurangi dimensi dengan menemukan serangkaian fitur baru yang disebutkomponen, yang merupakan komposit dari fitur asli, tetapi tidak berkorelasi satu sama lain. Komponen pertama menyumbang variabilitas terbesar dalam data, komponen kedua variabilitas paling kedua, dan seterusnya.

Ini adalah algoritma pengurangan dimensi tanpa pengawasan. Dalam pembelajaran tanpa pengawasan, label yang mungkin terkait dengan objek dalam kumpulan data pelatihan tidak digunakan.

Mengingat masukan dari matriks dengan
baris$x_1, \ldots, x_n$ m
masing dimensi1 * d, data dipartisi menjadi batch mini baris dan didistribusikan di antara node pelatihan (pekerja). Setiap pekerja kemudian menghitung ringkasan datanya. Ringkasan pekerja yang berbeda kemudian disatukan menjadi satu solusi pada akhir perhitungan.

Mode

Amazon SageMaker algoritma PCA menggunakan salah satu dari dua mode untuk menghitung ringkasan ini, tergantung pada situasi:

- teratur: untuk kumpulan data dengan data yang jarang dan jumlah pengamatan dan fitur yang moderat.
- acak: untuk kumpulan data dengan sejumlah besar pengamatan dan fitur. Mode ini menggunakan algoritma aproksimasi.

Sebagai langkah terakhir algoritma, ia melakukan dekomposisi nilai tunggal pada solusi terpadu, dari mana komponen utama kemudian diturunkan.

Mode 1: Reguler

Para pekerja bersama-sama menghitung keduanya$\sum x_i^T x_i$

> **Note**
>
> Karenax_i ac
> * dvektor
> baris,$x_i^T x_i$ ada
> matriks (bukan skalar). Menggunakan vektor baris dalam kode memungkinkan kita untuk mendapatkan caching efisien.

Matriks kovarians dihitung sebagai$\sum x_i^T x_i - (1/n)(\sum x_i)^T \sum x_i$ dan puncaknya`num_components`vektor tunggal membentuk model.

> **Note**
>
> Jika`subtract_mean`adalah`False`, kami menghindari komputasi dan mengurangi$\sum x_i$

Gunakan algoritma ini ketika dimensiddari vektor cukup kecil sehinggad^2 muat dalam memori.

Mode 2: Acak

Ketika jumlah fitur dalam kumpulan data input besar, kami menggunakan metode untuk memperkirakan metrik kovarians. Untuk setiap batch miniX_t din
* d, kita secara acak menginisialisasi`(num_components + extra_components) *` bmatriks yang kita kalikan dengan masing-masing mini-batch, untuk membuat`(num_components`

`+ extra_components) *` dmatriks. Jumlah matriks ini dihitung oleh pekerja, dan server melakukan SVD pada final`(num_components + extra_components) *` dmatriks. Kanan atas`num_components`vektor tunggal itu adalah perkiraan vektor tunggal atas matriks masukan.

Biarkanℓ
`= num_components + extra_components`. Diberikan mini-
batchX_t d
`*` d, pekerja menggambar matriks
acakH_t di
Tergantung pada apakah lingkungan menggunakan GPU atau CPU dan ukuran dimensi, matriks adalah matriks tanda acak di mana setiap entri`+-1`atauFJLT(transformasi Johnson Lindenstrauss cepat; untuk informasi, lihatTransformasi FJLTdan kertas tindak lanjut). Pekerja kemudian menghitung$H_t X_t$
memelihara$B = \sum H_t X_t$
Pekerja juga
mempertahankanh^T
jumlah
kolom$H_1, \ldots, H_T$ (
jumlah total mini-batch), dan`s`, jumlah semua baris masukan. Setelah memproses seluruh pecahan data, pekerja mengirimkan serverB,h,`s`, dann(jumlah baris input).

Menunjukkan input yang berbeda ke server
sebagaiB^1, h^1, s^1, n^1
menghitungB,h,`s`,njumlah input masing-masing. Kemudian
menghitung$C = B - (1/n)h^T s$
dan menemukan dekomposisi nilai tunggalnya. Vektor tunggal kanan atas dan nilai tunggalCdigunakan sebagai solusi perkiraan untuk masalah ini.

Hiperparameter PCA

Di`CreateTrainingJob`permintaan, Anda menentukan algoritma pelatihan. Anda juga dapat menentukan algoritme HyperParameters sebagai string-to-string peta. Tabel berikut mencantumkan hiperparameter untuk algoritme pelatihan PCA yang disediakan oleh Amazon SageMaker. Untuk informasi selengkapnya tentang cara kerja PCA, lihatCara PCA Berfungsi.

Nama Parameter	Deskripsi
`feature_dim`	Dimensi input.

Nama Parameter	Deskripsi
	Wajib Nilai yang valid: bilangan bulat positif
`mini_batch_size`	Jumlah baris dalam batch mini. Wajib Nilai yang valid: bilangan bulat positif
`num_components`	Jumlah komponen utama untuk menghitung. Wajib Nilai yang valid: bilangan bulat positif
`algorithm_mode`	Mode untuk menghitung komponen utama. Opsional Nilai yang benar:teraturatauacak Nilai default:teratur
`extra_components`	Ketika nilai meningkat, solusinya menjadi lebih akurat tetapi runtime dan konsumsi memori meningkat secara linear. Default, -1, berarti maksimum 10 dan`num_components` . Berlaku untukacakMode saja. Opsional Nilai yang valid: Bilangan bulat non-negatif atau -1 Nilai default: -1

Nama Parameter	Deskripsi
`subtract_mean`	Menunjukkan apakah data harus berisi baik selama pelatihan dan pada kesimpulan. Opsional Nilai yang valid: Salah satubenarataupalsu Nilai default:benar

Format Respons

Semua Amazon SageMaker algoritma bawaan mematuhi format inferensi input umum yang dijelaskan dalamFormat Data Umum - Inferensi. Topik ini berisi daftar format keluaran yang tersedia untuk SageMaker Algoritme PCA.

Format Respons

Terima — aplikasi/JSON

```
{
    "projections": [
        {
            "projection": [1.0, 2.0, 3.0, 4.0, 5.0]
        },
        {
            "projection": [6.0, 7.0, 8.0, 9.0, 0.0]
        },
        ....
    ]
}
```

Format Respons

Terima — aplikasi/JSONLines

```
{ "projection": [1.0, 2.0, 3.0, 4.0, 5.0] }
{ "projection": [6.0, 7.0, 8.0, 9.0, 0.0] }
```

Format Respons

Terima — aplikasi/x-recordio-protobuf

```
[
    Record = {
        features = {},
        label = {
            'projection': {
                keys: [],
                values: [1.0, 2.0, 3.0, 4.0, 5.0]
            }
        }
    },
    Record = {
        features = {},
        label = {
            'projection': {
                keys: [],
                values: [1.0, 2.0, 3.0, 4.0, 5.0]
            }
        }
    }
]
```

Algoritma Acak Cut Forest (RCF)

AmazonSageMakerRandom Cut Forest (RCF) adalah algoritma tanpa pengawasan untuk mendeteksi titik data anomali dalam kumpulan data. Ini adalah pengamatan yang menyimpang dari data yang terstruktur dengan baik atau berpola. Anomali dapat bermanifestasi sebagai lonjakan tak terduga dalam data deret waktu, jeda periodisitas, atau titik data yang tidak dapat diklasifikasikan. Mereka mudah dijelaskan dalam hal itu, jika dilihat dalam plot, mereka sering mudah dibedakan dari data “biasa”. Termasuk anomali ini dalam kumpulan data dapat secara drastis meningkatkan kompleksitas tugas pembelajaran mesin karena data “reguler” sering dapat digambarkan dengan model sederhana.

Dengan setiap titik data, RCF mengaitkan skor anomali. Nilai skor rendah menunjukkan bahwa titik data dianggap “normal.” Nilai tinggi menunjukkan adanya anomali dalam data. Definisi “rendah” dan “tinggi” bergantung pada aplikasi tetapi praktik umum menunjukkan bahwa skor di luar tiga penyimpangan standar dari skor rata-rata dianggap anomali.

Meskipun ada banyak aplikasi algoritma deteksi anomali untuk data deret waktu satu dimensi seperti analisis volume lalu lintas atau deteksi lonjakan volume suara, RCF dirancang untuk bekerja dengan input dimensi sewenang-wenang. AmazonSageMakerRCF menskalakan dengan baik sehubungan dengan jumlah fitur, ukuran kumpulan data, dan jumlah instance.

Topik

Antarmuka Input/Output untuk Algoritma RCF

AmazonSageMakerAcak Cut Forest mendukung`train`dan`test`saluran data. Saluran uji opsional digunakan untuk menghitung akurasi, presisi, penarikan, dan metrik skor F1 pada data berlabel. Melatih dan menguji jenis konten data dapat berupa`application/x-recordio-protobuf`atau`text/csv`format. Untuk data pengujian, saat menggunakan format teks/csv, konten harus ditentukan sebagai text/csv; label_size=1 di mana kolom pertama dari setiap baris mewakili label anomali: “1" untuk titik data anomali dan “0" untuk titik data normal. Anda dapat menggunakan mode File atau mode Pipa untuk melatih model RCF pada data yang diformat sebagai`recordIO-wrapped-protobuf`atau sebagai`CSV`

Saluran kereta hanya mendukung`S3DataDistributionType=ShardedByS3Key`dan saluran uji hanya mendukung`S3DataDistributionType=FullyReplicated`. Contoh berikut menentukan jenis distribusi S3 untuk saluran kereta menggunakanAmazonSageMakerPython.

> **Note**
>
> Yang`sagemaker.inputs.s3_input`berganti nama menjadi`sagemaker.inputs.TrainingInput`diSageMakerPython.

```
import sagemaker
```

```
# specify Random Cut Forest training job information and hyperparameters
rcf = sagemaker.estimator.Estimator(...)

# explicitly specify "ShardedByS3Key" distribution type
train_data = sagemaker.inputs.TrainingInput(
     s3_data=s3_training_data_location,
     content_type='text/csv;label_size=0',
     distribution='ShardedByS3Key')

# run the training job on input data stored in S3
rcf.fit({'train': train_data})
```

Untuk menghindari kesalahan umum di sekitar peran eksekusi, pastikan bahwa Anda memiliki peran eksekusi yang diperlukan,`AmazonSageMakerFullAccess`dan`AmazonEC2ContainerRegistryFullAccess`. Untuk menghindari kesalahan umum di sekitar gambar Anda yang tidak ada atau izinnya salah, pastikan bahwa gambar ECR Anda tidak lebih besar daripada ruang disk yang dialokasikan pada instance pelatihan. Untuk menghindari hal ini, jalankan pekerjaan pelatihan Anda pada instance yang memiliki ruang disk yang cukup. Selain itu, jika gambar ECR Anda berasal dari yang berbedaAWSrepositori Elastic Container Service (ECS) akun, dan Anda tidak mengatur izin repositori untuk memberikan akses, ini akan mengakibatkan kesalahan. LihatIzin repositori ECRuntuk informasi lebih lanjut tentang pengaturan pernyataan kebijakan repositori.

Lihat`S3DataSource`untuk informasi lebih lanjut tentang menyesuaikan atribut sumber data S3. Akhirnya, untuk memanfaatkan pelatihan multi-instance, data pelatihan harus dipartisi menjadi setidaknya sebanyak file sebagai contoh.

Untuk inferensi, RCF mendukung`application/x-recordio-protobuf`,`text/csv`dan`application/json`masukan jenis konten data. LihatFormat Data Umum untuk Algoritma Bawaandokumentasi untuk informasi lebih lanjut. RCF inferensi kembali`application/x-recordio-protobuf`atau`application/json`output diformat. Setiap catatan dalam data keluaran ini berisi skor anomali yang sesuai untuk setiap titik data input. LihatFormat Data Umum--Inferensiuntuk informasi lebih lanjut.

Untuk informasi lebih lanjut tentang format file input dan output, lihatFormat Respons RCFuntuk kesimpulan danNotebook Sampel RCF.

Rekomendasi Instans untuk Algoritma RCF

Untuk pelatihan, kami merekomendasikan`ml.m4`,`ml.c4`, dan`ml.c5`contoh keluarga. Untuk kesimpulan, kami sarankan menggunakan`ml.c5.xl`jenis instance khususnya, untuk kinerja maksimum serta meminimalkan biaya per jam penggunaan. Meskipun algoritma secara teknis dapat berjalan pada jenis instans GPU itu tidak mengambil keuntungan dari perangkat keras GPU.

Notebook Sampel RCF

Untuk contoh cara melatih model RCF dan melakukan kesimpulan dengannya, lihatPengantarSageMakerHutan Potong Acakbuku catatan. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contohSageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMakerContohtab untuk melihat daftar semuaSageMakersampel. Untuk membuka notebook, klik pada nyaGunakantab dan pilihBuat salinan.

Bagaimana RCF Bekerja

AmazonSageMakerRandom Cut Forest (RCF) adalah algoritma tanpa pengawasan untuk mendeteksi titik data anomali dalam dataset. Ini adalah pengamatan yang menyimpang dari data yang terstruktur dengan baik atau berpola. Anomali dapat bermanifestasi sebagai lonjakan tak terduga dalam data deret waktu, jeda periodisitas, atau titik data yang tidak dapat diklasifikasikan. Mereka mudah dijelaskan dalam hal itu, jika dilihat dalam plot, mereka sering mudah dibedakan dari data “biasa”. Termasuk anomali ini dalam dataset dapat secara drastis meningkatkan kompleksitas tugas pembelajaran mesin karena data “reguler” sering dapat digambarkan dengan model sederhana.

Ide utama di balik algoritma RCF adalah membuat hutan pohon di mana setiap pohon diperoleh dengan menggunakan partisi sampel data pelatihan. Misalnya, sampel acak dari data input pertama kali ditentukan. Sampel acak kemudian dipartisi sesuai dengan jumlah pohon di hutan. Setiap pohon diberi partisi seperti itu dan mengatur bagian titik itu menjadi pohon k-d. Skor anomali yang ditetapkan ke titik data oleh pohon didefinisikan sebagai perubahan yang diharapkan dalam kompleksitas pohon sebagai hasil menambahkan titik itu ke pohon; yang, dalam perkiraan, berbanding terbalik dengan kedalaman titik yang dihasilkan di pohon. Hutan potong acak memberikan skor anomali dengan menghitung skor rata-rata dari setiap pohon konstituen dan skala hasil sehubungan dengan ukuran sampel. Algoritma RCF didasarkan pada yang dijelaskan dalam referensi [1].

Data Sampel Secara Acak

Langkah pertama dalam algoritma RCF adalah untuk mendapatkan sampel acak dari data pelatihan. Secara khusus, misalkan kita ingin sampel ukuran K poin data. Jika data pelatihan cukup kecil, seluruh dataset dapat digunakan, dan kita bisa menggambar secara acak K el dari set ini. Namun, seringkali data pelatihan terlalu besar untuk muat sekaligus, dan pendekatan ini tidak layak dilakukan. Sebaliknya, kita menggunakan teknik yang disebut reservoir sampling.

Sampling waduk adalah algoritma untuk secara efisien menggambar sampel acak dari dataset $S = \{S_1, \ldots, S_N\}$ elemen-elemen dalam dataset hanya dapat diamati satu per satu atau dalam batch. Bahkan, sampling waduk bekerja bahkan ketika N t diketahui apriori. Jika hanya satu sampel yang diminta, seperti kapan $K = 1$, algoritmanya seperti ini:

Algoritma: Sampling Reservoir

- Input: dataset atau aliran data $S = \{S_1, \ldots, S_N\}$
- Inisialisasi sampel acak $X = S_1$
- Untuk setiap sampel yang diamati $S_n, n = 2, \ldots, N$
  - Pilih nomor acak yang seragam $\xi \in [0, 1]$
  - Jika $\xi < 1/n$
    - Set $X = S_n$
- Kembali X

Algoritma ini memilih sampel acak sehingga $P(X = S_n) = 1/N$

semua$n = 1, \ldots, N$
Kapan$K > 1$
lebih rumit. Selain itu, perbedaan harus dibuat antara pengambilan sampel acak yang dengan dan tanpa penggantian. RCF melakukan pengambilan sampel reservoir ditambah tanpa penggantian pada data pelatihan berdasarkan algoritma yang dijelaskan dalam [2].

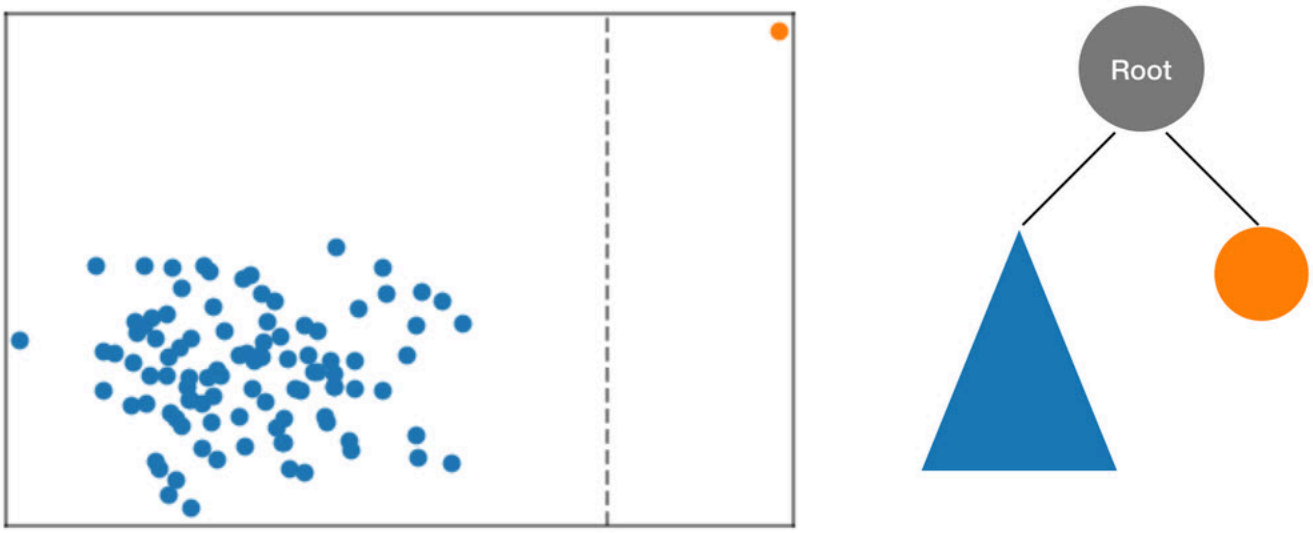
Melatih Model RCF dan Menghasilkan Inferensi

Langkah selanjutnya dalam RCF adalah membangun hutan potong acak menggunakan sampel data acak. Pertama, sampel dipartisi menjadi sejumlah partisi berukuran sama dengan jumlah pohon di hutan. Kemudian, setiap partisi dikirim ke pohon individu. Pohon secara rekursif mengatur partisinya menjadi pohon biner dengan mempartisi domain data ke dalam kotak pembatas.

Prosedur ini paling baik diilustrasikan dengan sebuah contoh. Misalkan pohon diberikan dataset dua dimensi berikut. Pohon yang sesuai diinisialisasi ke simpul akar:

Sebuah dataset dua dimensi di mana mayoritas data terletak pada cluster (biru) kecuali untuk satu titik data anomali (oranye). Pohon diinisialisasi dengan simpul akar.

Algoritma RCF mengatur data ini di pohon dengan terlebih dahulu menghitung kotak pembatas data, memilih dimensi acak (memberikan bobot lebih pada dimensi dengan “varians” yang lebih tinggi), dan kemudian secara acak menentukan posisi “potong” hyperplane melalui dimensi itu. Dua subruang yang dihasilkan mendefinisikan sub pohon mereka sendiri. Dalam contoh ini, pemotongan terjadi untuk memisahkan satu titik dari sisa sampel. Tingkat pertama dari pohon biner yang dihasilkan terdiri dari dua node, satu yang akan terdiri dari subtree poin di sebelah kiri potongan awal dan yang lainnya mewakili titik tunggal di sebelah kanan.



Potongan acak yang mempartisi dataset dua dimensi. Titik data anomali lebih cenderung diisolasi dalam kotak pembatas pada kedalaman pohon yang lebih kecil daripada titik lainnya.

Kotak pembatas kemudian dihitung untuk bagian kiri dan kanan data dan proses diulang sampai setiap daun pohon mewakili titik data tunggal dari sampel. Perhatikan bahwa jika titik tunggal cukup jauh maka kemungkinan besar potongan acak akan menghasilkan isolasi titik. Pengamatan ini memberikan intuisi bahwa kedalaman pohon, secara longgar, berbanding terbalik dengan skor anomali.

Ketika melakukan inferensi menggunakan model RCF terlatih skor anomali akhir dilaporkan sebagai rata-rata di seluruh skor yang dilaporkan oleh masing-masing pohon. Perhatikan bahwa sering terjadi bahwa titik data baru belum berada di pohon. Untuk menentukan skor yang terkait dengan titik baru, titik data dimasukkan ke dalam pohon yang diberikan dan pohon secara efisien (dan sementara) dipasang kembali dengan cara yang setara dengan proses pelatihan yang dijelaskan di atas. Artinya, pohon yang dihasilkan seolah-olah titik data input adalah anggota sampel yang digunakan untuk membangun pohon di tempat pertama. Skor yang dilaporkan berbanding terbalik dengan kedalaman titik input di dalam pohon.

Pilih Hyperparameters

Hyperparameter utama yang digunakan untuk menyetel model RCF adalah`num_trees`dan`num_samples_per_tree`. Meningkat`num_trees`memiliki efek mengurangi kebisingan yang diamati pada skor anomali karena skor akhir adalah rata-rata skor yang dilaporkan oleh masing-masing pohon. Sementara nilai optimal tergantung pada aplikasi, kami sarankan

untuk menggunakan 100 pohon untuk memulai sebagai keseimbangan antara noise skor dan kompleksitas model. Perhatikan bahwa waktu inferensi sebanding dengan jumlah pohon. Meskipun waktu pelatihan juga terpengaruh itu didominasi oleh algoritma sampling reservoir dijelaskan di atas.

Parameternyanum_samples_per_treeterkait dengan kepadatan anomali yang diharapkan dalam kumpulan data. Secara khusus,num_samples_per_treeharus dipilih sedemikian rupa sehingga1/num_samples_per_treemendekati rasio data anomali terhadap data normal. Misalnya, jika 256 sampel digunakan di setiap pohon maka kami mengharapkan data kami mengandung anomali 1/256 atau sekitar 0,4% dari waktu. Sekali lagi, nilai optimal untuk hyperparameter ini tergantung pada aplikasi.

References

1. Sudipto Guha, Nina Mishra, Gourav Roy, dan Okke Schrijvers. “Deteksi anomali berbasis hutan secara acak yang kuat di sungai.” DalamKonferensi Internasional tentang Pembelajaran Mesin, hlm. 2712-2721. 2016.
2. Taman Byung-Hoon, George Ostrouchov, Nagiza F. Samatova, dan Al Geist. “Sampling acak berbasis reservoir dengan penggantian dari aliran data.” DalamProsiding Konferensi Internasional SIAM 2004 tentang Data Mining, hlm. 492-496. Masyarakat untuk Matematika Industri dan Terapan, 2004.

Hiperparameter RCF

DalamCreateTrainingJobpermintaan, Anda menentukan algoritma pelatihan. Anda juga dapat menentukan hyperparameter khusus algoritma sebagaistring-to-stringpeta. Tabel berikut mencantumkan hyperparameter untuk AmazonSageMakerAlgoritma RCF. Untuk informasi lebih lanjut, termasuk rekomendasi tentang cara memilih hyperparameter, lihatBagaimana RCF Bekerja.

Nama Parameter	Deskripsi
feature_dim	Jumlah fitur dalam kumpulan data. (Jika Anda menggunakanHutan Potong Acakestimator, nilai ini dihitung untuk Anda dan tidak perlu ditentukan.) Diperlukan Nilai yang valid: bilangan bulat positif (min: 1, maks: 10000)

Nama Parameter	Deskripsi
`eval_metrics`	Daftar metrik yang digunakan untuk mencetak kumpulan data pengujian berlabel. Metrik berikut dapat dipilih untuk output: • `accuracy`- mengembalikan sebagian kecil dari prediksi yang benar. • `precision_recall_fscore` - mengembalikan presisi positif dan negatif, recall, dan F1-skor. Opsional Nilai yang valid: daftar dengan nilai yang mungkin diambil dari`accuracy`atau`precision_recall_fscore` . Nilai default: Keduanya`accuracy`,`precision_recall_fscore` dihitung.
`num_samples_per_tree`	Jumlah sampel acak yang diberikan ke setiap pohon dari kumpulan data pelatihan. Opsional Nilai yang valid: bilangan bulat positif (min: 1, maks: 2048) Nilai default: 256
`num_trees`	Jumlah pohon di hutan. Opsional Nilai yang valid: bilangan bulat positif (min: 50, maks: 1000) Nilai default: 100

Tune atau Model RCF

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning atau hyperparameter optimization, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang

nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

AmazonSageMakeralgoritma RCF adalah algoritma anomali-deteksi tanpa pengawasan yang membutuhkan dataset uji berlabel untuk optimasi hyperparameter. RCF menghitung skor anomali untuk titik data uji dan kemudian memberi label titik data sebagai anomali jika skor mereka melampaui tiga penyimpangan standar dari skor rata-rata. Ini dikenal sebagai heuristik batas tiga-sigma. Skor F1 didasarkan pada perbedaan antara label yang dihitung dan label aktual. Pekerjaan penyetelan hyperparameter menemukan model yang memaksimalkan skor itu. Keberhasilan optimasi hyperparameter tergantung pada penerapan heuristik batas tiga-sigma ke set data pengujian.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma RCF

Algoritma RCF menghitung metrik berikut selama pelatihan. Saat menyetel model, pilih metrik ini sebagai metrik objektif.

Nama Metrik	Deskripsi	Arah Optimasi
`test:f1`	Skor F1 pada kumpulan data pengujian, berdasarkan perbedaan antara label yang dihitung dan label aktual.	Maksimalkan

Hyperparameter RCF merdu

Anda dapat menyetel model RCF dengan hyperparameter berikut.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`num_samples_per_tree`	IntegerParameterRanges	MinValue: 1,MaxValue:2048
`num_trees`	IntegerParameterRanges	MinValue: 50,MaxValue:1000

Format Respons RCF

Semua AmazonSageMakeralgoritma bawaan mematuhi format inferensi input umum yang dijelaskan dalamFormat Data Umum - Inferensi. Perhatikan bahwaSageMakerRandom Cut Forest mendukung format JSON dan RecordO yang padat dan jarang. Topik ini berisi daftar format keluaran yang tersedia untukSageMakerAlgoritma RCF.

Format Respons JSON

TERIMA: aplikasi/json.

```
    {


        "scores":    [


            {"score": 0.02},


            {"score": 0.25}


        ]


    }
```

Format Respon JSONLINES

TERIMA: aplikasi/jsonlines.

```
{"score": 0.02},
{"score": 0.25}
```

Format Respons RECORDIO

MENERIMA: aplikasi/x-recordio-protobuf.

```
[

    Record = {

        features = {},

        label = {

            'score': {

                keys: [],

                values: [0.25]  # float32

            }

        }

    },

    Record = {
```

```
            features = {},

            label = {

                'score': {

                    keys: [],

                    values: [0.23]  # float32

                }

            }

        }

    ]
```

SageMaker Algoritma Built-in untuk Visi Komputer

SageMaker menyediakan algoritma pengolahan gambar yang digunakan untuk klasifikasi gambar, deteksi objek, dan visi komputer.

- Klasifikasi Gambar - MXNet—uses contoh data dengan jawaban (disebut sebagai algoritma diawasi). Gunakan algoritma ini untuk mengklasifikasikan gambar.

- Klasifikasi Gambar - TensorFlow—menggunakan model TensorFlow Hub terlatih untuk menyempurnakan tugas tertentu (disebut sebagai algoritma yang diawasi). Gunakan algoritma ini untuk mengklasifikasikan gambar.
- Deteksi—detects dan mengklasifikasikan objek dalam gambar menggunakan jaringan saraf dalam tunggal. Ini adalah algoritma pembelajaran yang diawasi yang mengambil gambar sebagai input dan mengidentifikasi semua contoh objek dalam adegan gambar.
- Deteksi Objek - TensorFlow—detects kotak pembatas dan label objek dalam gambar. Ini adalah algoritma pembelajaran yang diawasi yang mendukung pembelajaran transfer dengan TensorFlow model terlatih yang tersedia.
- Algoritme segmentasi semantik—menyediakan pendekatan tingkat piksel halus untuk mengembangkan aplikasi visi komputer.

Nama algoritma	Nama saluran	Mode input pelatihan	Tipe file	Kelas Instans	Paralleli zable
Klasifikasi Citra - MXNet	melatih dan validasi, (opsional) train_lst, validation_lst, dan model	File atau Pipa	RecordO atau file gambar (.jpg atau .png)	GPU	Ya
Klasifikasi Citra - TensorFlow	pelatihan dan validasi	File	file gambar (.jpg, .jpeg, atau.png)	CPU atau GPU	Ya (hanya di beberapa GPU pada satu instans)
Deteksi Objek	melatih dan validasi, (opsional) train_ann	File atau Pipa	RecordO atau file gambar (.jpg atau .png)	GPU	Ya

Nama algoritma	Nama saluran	Mode input pelatihan	Tipe file	Kelas Instans	Paralleli zable	
	otation, validatio n_annotat ion, dan model					
Deteksi Objek - TensorFlo w	pelatihan dan validasi	File	file gambar (.jpg, .jpeg, atau.png)	GPU	Ya (hanya di beberapa GPU pada satu instans)	
Segmentas i semantik	melatih dan validasi, train_ann otation, validatio n_annotat ion, dan (opsional) label_map dan model	File atau Pipa	File gambar	GPU (hanya contoh tunggal)	Tidak	

Klasifikasi Gambar - MXNet

AmazonSageMakeralgoritma klasifikasi gambar adalah algoritma pembelajaran diawasi yang mendukung klasifikasi multi-label. Dibutuhkan gambar sebagai input dan output satu atau lebih label ditugaskan untuk gambar itu. Ini menggunakan jaringan saraf konvolusional yang dapat dilatih dari awal atau dilatih menggunakan pembelajaran transfer ketika sejumlah besar gambar pelatihan tidak tersedia

Format input yang direkomendasikan untuk AmazonSageMakeralgoritma klasifikasi gambar adalah Apache MXNetRecordiO. Namun, Anda juga dapat menggunakan gambar mentah dalam format.jpg

atau .png. Merujuk kediskusi iniuntuk gambaran luas tentang persiapan dan pemuatan data yang efisien untuk sistem pembelajaran mesin.

> **Note**
>
> Untuk mempertahankan interoperabilitas yang lebih baik dengan kerangka kerja deep learning yang ada, ini berbeda dari format data protobuf yang biasa digunakan oleh Amazon lainnyaSageMakeralgoritma.

Untuk informasi lebih lanjut tentang jaringan konvolusional, lihat:

- Pembelajaran residual mendalam untuk pengenalan gambarKaiming He, et al., Konferensi IEEE 2016 tentang Visi Komputer dan Pengenalan Pola
- ImageNetdatabase gambar
- Klasifikasi gambar dengan Gluon-CV dan MXNet

Topik

- Antarmuka Input/Output untuk Algoritma Klasifikasi Gambar
- Rekomendasi Instans EC2 untuk Algoritma Klasifikasi Gambar
- Notebook Contoh Klasifikasi Gambar
- Bagaimana Klasifikasi Gambar Bekerja
- Hyperparameter Klasifikasi Gambar
- Tune Model Klasifikasi Gambar

Antarmuka Input/Output untuk Algoritma Klasifikasi Gambar

YangSageMakerAlgoritma Klasifikasi Gambar mendukung kedua RecordO (`application/x-recordio`) dan gambar (`image/png`,`image/jpeg`, dan`application/x-image`) jenis konten untuk pelatihan dalam mode file, dan mendukung recordO (`application/x-recordio`) jenis konten untuk pelatihan dalam mode pipa. Namun, Anda juga dapat berlatih dalam mode pipa menggunakan file gambar (`image/png`,`image/jpeg`, dan`application/x-image`), tanpa membuat file Recordio, dengan menggunakan format manifes augmented.

Pelatihan terdistribusi didukung untuk mode file dan mode pipa. Saat menggunakan jenis konten Recordio dalam mode pipa, Anda harus

mengaturS3DataDistributionTypedariS3DataSourcekepadaFullyReplicated. Algoritma ini mendukung model yang sepenuhnya direplikasi di mana data Anda disalin ke setiap mesin.

Algoritma ini mendukungimage/png,image/jpeg, danapplication/x-imageuntuk kesimpulan.

Berlatih dengan Format Recordio

Jika Anda menggunakan format RecordO untuk pelatihan, tentukan keduanyatraindanvalidationsaluran sebagai nilai untukInputDataConfigparameter dariCreateTrainingJobpermintaan. Tentukan satu recordO (.rec) file ditrainsaluran dan satu file recordO divalidationsaluran. Mengatur jenis konten untuk kedua saluran untukapplication/x-recordio.

Berlatih dengan Format Gambar

Jika Anda menggunakan format Gambar untuk pelatihan, tentukantrain,validation,train_lst, danvalidation_lstsaluran sebagai nilai untukInputDataConfigparameter dariCreateTrainingJobpermintaan. Tentukan data gambar individu (.jpgatau.pngfile) untuktraindanvalidationsaluran. Tentukan satu.lstfile di masing-masingtrain_lstdanvalidation_lstsaluran. Atur jenis konten untuk keempat saluran keapplication/x-image.

> **Note**
>
> SageMakermembaca data pelatihan dan validasi secara terpisah dari saluran yang berbeda, jadi Anda harus menyimpan data pelatihan dan validasi di folder yang berbeda.

SEBUAH.lstfile adalah file tab yang dipisahkan dengan tiga kolom yang berisi daftar file gambar. Kolom pertama menentukan indeks gambar, kolom kedua menentukan indeks label kelas untuk gambar, dan kolom ketiga menentukan jalur relatif dari file gambar. Indeks gambar di kolom pertama harus unik di semua gambar. Kumpulan indeks label kelas diberi nomor berturut-turut dan penomoran harus dimulai dengan 0. Misalnya, 0 untuk kelas kucing, 1 untuk kelas dog, dan seterusnya untuk kelas tambahan.

Berikut ini adalah contoh dari.lstberkas:

```
5       1   your_image_directory/train_img_dog1.jpg
1000    0   your_image_directory/train_img_cat1.jpg
22      1   your_image_directory/train_img_dog2.jpg
```

Misalnya, jika gambar latihan Anda disimpans3://<your_bucket>/train/class_dog,s3://<your_bucket>/train/class_cat, dan seterusnya, tentukan jalur untuktrainsaluran sebagais3://<your_bucket>/train, yang merupakan direktori tingkat atas untuk data Anda. Dalam.lstfile, tentukan path relatif untuk file individual bernamatrain_image_dog1.jpgdi dalamclass_dogdirektori kelas sebagaiclass_dog/train_image_dog1.jpg. Anda juga dapat menyimpan semua file gambar Anda di bawah satu subdirektori di dalamtraindirektori. Dalam hal ini, gunakan subdirektori itu untuk jalur relatif. Sebagai contoh, s3://<your_bucket>/train/your_image_directory.

Latih dengan Format Gambar Manifes Augmented

Format manifes yang ditambah memungkinkan Anda melakukan pelatihan dalam mode Pipe menggunakan file gambar tanpa perlu membuat file RecordiO. Anda perlu menentukan saluran kereta dan validasi sebagai nilai untukInputDataConfigparameter dariCreateTrainingJobpermintaan. Saat menggunakan format, file manifes S3 perlu dibuat yang berisi daftar gambar dan anotasi yang sesuai. Format file manifes harus diGaris JSONformat di mana setiap baris mewakili satu sampel. Gambar yang ditentukan menggunakan'source-ref'tag yang menunjuk ke lokasi S3 gambar. Anotasi disediakan di bawah"AttributeNames"nilai parameter seperti yang ditentukan dalamCreateTrainingJobpermintaan. Hal ini juga dapat berisi metadata tambahan di bawahmetadatatag, tapi ini diabaikan oleh algoritma. Pada contoh berikut,"AttributeNames"terkandung dalam daftar referensi gambar dan anotasi["source-ref", "class"]. Nilai label yang sesuai adalah"0"untuk gambar pertama dan“1”untuk gambar kedua:

```
{"source-ref":"s3://image/filename1.jpg", "class":"0"}
{"source-ref":"s3://image/filename2.jpg", "class":"1", "class-metadata": {"class-name":
 "cat", "type" : "groundtruth/image-classification"}}
```

Urutan"AttributeNames"dalam file masukan penting saat melatihImageClassificationalgoritma. Ia menerima data pipa dalam urutan tertentu, denganimagepertama, diikuti olehlabel. Jadi”AttributeNames“dalam contoh ini disediakan dengan"source-ref"pertama, diikuti oleh"class". Saat menggunakanImageClassificationalgoritma dengan Augmented Manifest, nilaiRecordWrapperTypeparameter harus"RecordIO".

Pelatihan multi-label juga didukung dengan menentukan larik nilai JSON. Yangnum_classeshyperparameter harus diatur untuk mencocokkan jumlah kelas. Ada dua format label yang valid: multi-hot dan kelas-id.

Dalam format multi-hot, setiap label adalah vektor multi-hot dikodekan dari semua kelas, di mana setiap kelas mengambil nilai 0 atau 1. Pada contoh berikut, ada tiga kelas. Gambar pertama diberi label dengan kelas 0 dan 2, sedangkan gambar kedua diberi label dengan kelas 2 saja:

```
{"image-ref": "s3://mybucket/sample01/image1.jpg", "class": "[1, 0, 1]"}
{"image-ref": "s3://mybucket/sample02/image2.jpg", "class": "[0, 0, 1]"}
```

Dalam format kelas-id, setiap label adalah daftar id kelas, dari [0,`num_classes`), yang berlaku untuk titik data. Contoh sebelumnya akan terlihat seperti ini:

```
{"image-ref": "s3://mybucket/sample01/image1.jpg", "class": "[0, 2]"}
{"image-ref": "s3://mybucket/sample02/image2.jpg", "class": "[2]"}
```

Format multi-hot adalah default, tetapi dapat secara eksplisit diatur dalam jenis konten dengan`label-format`parameter:`"application/x-recordio; label-format=multi-hot"`.Format kelas-id, yang merupakan format yang dikeluarkan olehGroundTruth, harus diatur secara eksplisit:`"application/x-recordio; label-format=class-id"`.

Untuk informasi selengkapnya tentang file manifes tambahan, lihatMenyediakan Dataset Metadata untuk Training Jobs dengan Augmented Manifest File.

Pelatihan Inkremental

Anda juga dapat menyemai pelatihan model baru dengan artefak dari model yang Anda latih sebelumnyaSageMaker. Pelatihan tambahan menghemat waktu pelatihan saat Anda ingin melatih model baru dengan data yang sama atau serupa. SageMakermodel klasifikasi gambar dapat diunggulkan hanya dengan model klasifikasi gambar built-in lainnya yang dilatihSageMaker.

Untuk menggunakan model yang telah dilatih sebelumnya, di`CreateTrainingJob`permintaan, tentukan`ChannelName`sebagai “model” di`InputDataConfig`parameter. Mengatur`ContentType`untuk saluran model`application/x-sagemaker-model`. Hyperparameter input dari model baru dan model terlatih yang Anda unggah ke saluran model harus memiliki pengaturan yang sama untuk`num_layers`,`image_shape`dan`num_classes`parameter masukan. Parameter ini menentukan arsitektur jaringan. Untuk file model yang telah dilatih sebelumnya, gunakan artefak model terkompresi (dalam format.tar.gz) keluaran olehSageMaker. Anda dapat menggunakan format RecordiO atau gambar untuk input data.

Inferensi dengan Algoritma Klasifikasi Gambar

Model yang dihasilkan dapat di-host untuk inferensi dan dukungan dikodekan `.jpg`dan `.png`format gambar sebagai`image/png, image/jpeg`, dan`application/x-image`konten-jenis. Gambar input diubah ukurannya secara otomatis. Outputnya adalah nilai probabilitas untuk semua kelas yang dikodekan dalam format JSON, atau diFormat teks JSON Linesuntuk batch transform. Model klasifikasi gambar memproses satu gambar per permintaan sehingga hanya menghasilkan satu baris dalam format Garis JSON atau JSON. Berikut ini adalah contoh respons dalam format Garis JSON:

```
accept: application/jsonlines

 {"prediction": [prob_0, prob_1, prob_2, prob_3, ...]}
```

Untuk detail selengkapnya tentang pelatihan dan inferensi, lihat contoh contoh notebook klasifikasi gambar yang direferensikan dalam pendahuluan.

Rekomendasi Instans EC2 untuk Algoritma Klasifikasi Gambar

Untuk klasifikasi gambar, kami mendukung instans P2, P3, G4dn, dan G5. Sebaiknya gunakan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Anda juga dapat menjalankan algoritme pada pengaturan multi-GPU dan multi-mesin untuk pelatihan terdistribusi. Instans CPU (seperti C4) dan GPU (P2, P3, G4dn, atau G5) dapat digunakan untuk inferensi.

Notebook Contoh Klasifikasi Gambar

Untuk contoh notebook yang menggunakanSageMakeralgoritma klasifikasi gambar, lihatBuat dan Daftarkan Model Klasifikasi Gambar MXNet melaluiSageMakerSaluran pipa. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contohSageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilihSageMakerContohtab untuk melihat daftar semuaSageMakersampel. Contoh notebook klasifikasi gambar terletak diPengantar algoritma Amazonbagian. Untuk membuka notebook, klik pada nyaGunakantab dan pilihBuat salinan.

Bagaimana Klasifikasi Gambar Bekerja

Algoritma klasifikasi gambar mengambil gambar sebagai input dan mengklasifikasikannya ke dalam salah satu kategori keluaran. Pembelajaran mendalam telah merevolusi domain klasifikasi gambar dan telah mencapai kinerja yang luar biasa. Berbagai jaringan pembelajaran mendalam

seperti ResNet, DenseNet, Awal, dan seterusnya, telah dikembangkan untuk menjadi sangat akurat untuk klasifikasi gambar. Pada saat yang sama, ada upaya untuk mengumpulkan data gambar berlabel yang penting untuk melatih jaringan ini. ImageNet adalah salah satu kumpulan data besar yang memiliki lebih dari 11 juta gambar dengan sekitar 11.000 kategori. Setelah jaringan dilatih dengan ImageNet data, kemudian dapat digunakan untuk menggeneralisasi dengan dataset lain juga, dengan penyesuaian ulang sederhana atau fine-tuning. Dalam pendekatan pembelajaran transfer ini, jaringan diinisialisasi dengan bobot (dalam contoh ini, dilatih ImageNet), yang nantinya dapat disetel dengan baik untuk tugas klasifikasi gambar dalam kumpulan data yang berbeda.

Klasifikasi gambar di Amazon SageMaker dapat dijalankan dalam dua mode: pelatihan penuh dan pembelajaran transfer. Dalam mode pelatihan penuh, jaringan diinisialisasi dengan bobot acak dan dilatih pada data pengguna dari awal. Dalam mode pembelajaran transfer, jaringan diinisialisasi dengan bobot pra-terlatih dan hanya lapisan atas yang terhubung sepenuhnya diinisialisasi dengan bobot acak. Kemudian, seluruh jaringan disetel dengan data baru. Dalam mode ini, pelatihan dapat dicapai bahkan dengan kumpulan data yang lebih kecil. Ini karena jaringan sudah dilatih dan oleh karena itu dapat digunakan dalam kasus tanpa data pelatihan yang memadai.

Hyperparameter Klasifikasi Gambar

Hyperparameter adalah parameter yang ditetapkan sebelum model pembelajaran mesin mulai belajar. Hyperparameter berikut didukung oleh Amazon SageMaker algoritma Klasifikasi Gambar bawaan. Lihat Tune Model Klasifikasi Gambar untuk informasi tentang klasifikasi gambar hyperparameter tuning.

Nama Parameter	Deskripsi
`num_classes`	Jumlah kelas output. Parameter ini mendefinisikan dimensi output jaringan dan biasanya diatur ke jumlah kelas dalam dataset. Selain klasifikasi multi-kelas, klasifikasi multi-label juga didukung. Silakan merujuk ke Antarmuka Input/Output untuk Algoritma Klasifikasi Gambar untuk detail tentang cara bekerja dengan klasifikasi multi-label dengan file manifes yang ditambah. Diperlukan Nilai yang valid: bilangan bulat positif

Nama Parameter	Deskripsi
num_training_samples	Jumlah contoh pelatihan dalam kumpulan data masukan. Jika ada ketidakcocokan antara nilai ini dan jumlah sampel dalam set pelatihan, maka perilakulr_schedu ler_step parameter tidak terdefinisi dan akurasi pelatihan terdistribusi mungkin terpengaruh. Diperlukan Nilai yang valid: bilangan bulat positif
augmentation_type	Jenis augmentasi data. Gambar input dapat ditambah dengan berbagai cara seperti yang ditentukan di bawah ini. • crop: Pangkas gambar secara acak dan balikkan gambar secara horizontal • crop_color : Selain 'crop', tiga nilai acak dalam kisaran [-36, 36], [-50, 50], dan [-50, 50] ditambahkan ke saluran Hue-Saturation-Lightness yang sesuai masing-masing • crop_color_transform : Selaincrop_color , transformasi acak, termasuk variasi rotasi, geser, dan rasio aspek diterapkan pada gambar. Sudut rotasi maksimum adalah 10 derajat, rasio geser maksimum adalah 0,1, dan rasio perubahan aspek maksimum adalah 0,25. Opsional Nilai yang valid:crop,crop_color , ataucrop_colo r_transform . Nilai default: tidak ada nilai default

Nama Parameter	Deskripsi
`beta_1`	Beta1 untuk`adam`, yaitu tingkat peluruhan eksponensial untuk perkiraan momen pertama. Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 0,9
`beta_2`	Beta2 untuk`adam`, yaitu tingkat peluruhan eksponensial untuk perkiraan momen kedua. Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 0.999
`checkpoint_frequency`	Periode untuk menyimpan parameter model (dalam jumlah zaman). Perhatikan bahwa semua file pos pemeriksaan disimpan sebagai bagian dari file model akhir "model.tar.gz" dan diunggah ke S3 ke lokasi model yang ditentukan. Ini meningkatkan ukuran file model secara proporsional dengan jumlah pos pemeriksaan yang disimpan selama pelatihan. Opsional Nilai yang valid: bilangan bulat positif tidak lebih besar dari`epochs`. Nilai default: tidak ada nilai default (Simpan pos pemeriksaan pada zaman yang memiliki akurasi validasi terbaik)

Nama Parameter	Deskripsi
`early_stopping`	`True`untuk menggunakan logika penghentian awal selama pelatihan.`False`tidak menggunakannya. Opsional Nilai yang valid: `True` or `False` Nilai default: `False`
`early_stopping_min_epochs`	Jumlah minimum zaman yang harus dijalankan sebelum logika berhenti awal dapat dipanggil. Ini hanya digunakan bila`early_stopping =True`. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 10
`early_stopping_patience`	Jumlah zaman yang harus menunggu sebelum mengakhiri pelatihan jika tidak ada perbaikan yang dilakukan dalam metrik yang relevan. Ini hanya digunakan bila`early_stopping =True`. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5

Nama Parameter	Deskripsi
`early_stopping_tolerance`	Toleransi relatif untuk mengukur peningkatan metrik validasi akurasi. Jika rasio peningkatan akurasi dibagi dengan akurasi terbaik sebelumnya lebih kecil dari`early_stopping_tolerance` set nilai, berhenti awal menganggap tidak ada perbaikan. Ini hanya digunakan bila`early_stopping =True`. Opsional Nilai yang valid: 0 ≤ float ≤ 1 Nilai default: 0.0
`epochs`	Jumlah zaman pelatihan. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 30
`eps`	Epsilon untuk`adam`dan`rmsprop`. Biasanya diatur ke nilai kecil untuk menghindari pembagian dengan 0. Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 1e-8
`gamma`	Gamma untuk`rmsprop`, faktor peluruhan untuk rata-rata bergerak dari gradien kuadrat. Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 0,9

Nama Parameter	Deskripsi
image_shape	Dimensi gambar input, yang ukurannya sama dengan lapisan input jaringan. Format didefinisikan sebagai 'num_channels , tinggi, lebar '. Dimensi gambar dapat mengambil nilai apa pun karena jaringan dapat menangani dimensi input yang bervariasi. Namun, mungkin ada kendala memori jika dimensi gambar yang lebih besar digunakan. Model terlatih hanya dapat menggunak an ukuran gambar 224 x 224 yang tetap. Dimensi gambar tipikal untuk klasifikasi gambar adalah '3,224,224'. Ini mirip denganIma geNetset data. Untuk pelatihan, jika ada gambar input yang lebih kecil dari parameter ini dalam dimensi apa pun, pelatihan gagal. Jika gambar lebih besar, sebagian gambar dipotong, dengan area yang dipotong ditentukan oleh parameter ini. Jika hyperpara meteraugmentation_type diatur, tanaman acak diambil; jika tidak, tanaman sentral diambil. Pada inferensi, gambar masukan diubah ukurannya keimage_shape yang digunakan selama pelatihan. Rasio aspek tidak dipertahankan, dan gambar tidak dipotong. Opsional Nilai yang valid: string Nilai default: '3,224.224'

Nama Parameter	Deskripsi
`kv_store`	Mode sinkronisasi pembaruan berat selama pelatihan terdistri busi. Pembaruan bobot dapat diperbarui baik secara sinkron maupun asinkron di seluruh mesin. Pembaruan sinkron biasanya memberikan akurasi yang lebih baik daripada pembaruan asinkron tetapi bisa lebih lambat. Lihat pelatihan terdistribusi di MXNet untuk lebih jelasnya. Parameter ini tidak berlaku untuk pelatihan mesin tunggal. • `dist_sync` : Gradien disinkronkan setelah setiap batch dengan semua pekerja. Dengan`dist_sync` , ukuran batch sekarang berarti ukuran batch yang digunakan pada setiap mesin. Jadi jika ada n mesin dan kami menggunakan ukuran batch b, maka`dist_sync` berperilaku seperti lokal dengan ukuran batch n* b • `dist_async` : Melakukan pembaruan asinkron. Bobot diperbarui setiap kali gradien diterima dari mesin mana pun dan pembaruan bobotnya bersifat atom. Namun, pesanan tidak dijamin. Opsional Nilai yang valid: `dist_sync` atau `dist_async` Nilai default: tidak ada nilai default
`learning_rate`	Tingkat pembelajaran awal. Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 0,1

Nama Parameter	Deskripsi
`lr_scheduler_factor`	Rasio untuk mengurangi tingkat pembelajaran yang digunakan bersamaan dengan`lr_scheduler_step` parameter, didefinisikan sebagai`lr_new=lr_old*lr_scheduler_factor` . Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 0,1
`lr_scheduler_step`	Epochs di mana untuk mengurangi tingkat pembelajaran. Seperti yang dijelaskan dalam`lr_scheduler_factor` parameter, tingkat pembelajaran dikurangi dengan`lr_scheduler_factor` pada zaman ini. Misalnya, jika nilainya diatur ke “10, 20", maka tingkat pembelajaran dikurangi`lr_scheduler_factor` setelah zaman ke-10 dan lagi oleh`lr_scheduler_factor` setelah zaman ke-20. Epochs dibatasi oleh “,”. Opsional Nilai yang valid: string Nilai default: tidak ada nilai default
`mini_batch_size`	Ukuran batch untuk pelatihan. Dalam pengaturan multi-GPU mesin tunggal, masing-masing GPU menangani`mini_batch_size` /num_gpu sampel pelatihan. Untuk pelatihan multi-mesin dalam mode dist_sync, ukuran batch sebenarnya adalah`mini_batch_size` * jumlah mesin. Lihat dokumen MXNet untuk detail selengkapnya. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 32

Nama Parameter	Deskripsi
momentum	Momentum untuksgddannag, diabaikan untuk pengoptimal lainnya. Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 0,9
multi_label	Tandai digunakan untuk klasifikasi multi-label di mana setiap sampel dapat diberi beberapa label. Akurasi rata-rata di semua kelas dicatat. Opsional Nilai yang valid: 0 atau 1 Nilai default: 0
num_layers	Jumlah lapisan untuk jaringan. Untuk data dengan ukuran gambar besar (misalnya, 224x224 - sepertiImageNet), kami sarankan memilih jumlah lapisan dari set [18, 34, 50, 101, 152, 200]. Untuk data dengan ukuran gambar kecil (misalnya, 28x28 - seperti CIFAR), kami sarankan memilih jumlah lapisan dari set [20, 32, 44, 56, 110]. Jumlah lapisan di setiap set didasarkan padaResNetkertas. Untuk pembelajaran transfer, jumlah lapisan mendefinisikan arsitektur jaringan dasar dan karenanya hanya dapat dipilih dari himpunan [18, 34, 50, 101, 152, 200]. Opsional Nilai yang valid: bilangan bulat positif dalam [18, 34, 50, 101, 152, 200] atau [20, 32, 44, 56, 110] Nilai default: 152

Nama Parameter	Deskripsi
`optimizer`	Jenis optimizer. Untuk detail lebih lanjut tentang parameter untuk pengoptimal, silakan merujuk ke API MXNet. Opsional Nilai yang valid: Salah satu`sgd`,`adam`,`rmsprop`, atau`nag`. • `sgd`:Keturunan gradien stokastik • `adam`:Estimasi momentum adaptif • `rmsprop`:Akar berarti propagasi kuadrat • `nag`:Nesterov dipercepat gradien Nilai default: `sgd`
`precision_dtype`	Ketepatan bobot yang digunakan untuk pelatihan. Algoritma dapat menggunakan presisi tunggal (`float32`) atau setengah presisi (`float16`) untuk bobot. Menggunakan setengah presisi untuk bobot menghasilkan pengurangan konsumsi memori. Opsional Nilai yang valid: `float32` or `float16` Nilai default: `float32`

Nama Parameter	Deskripsi
`resize`	Jumlah piksel di sisi terpendek gambar setelah mengubah ukurannya untuk pelatihan. Jika parameter tidak diatur, maka data pelatihan digunakan tanpa mengubah ukuran. Parameter harus lebih besar dari komponen lebar dan tinggi`image_shape` untuk mencegah kegagalan pelatihan. Diperlukansaat menggunakan jenis konten gambar Opsionalsaat menggunakan tipe konten Recordio Nilai yang valid: bilangan bulat positif Nilai default: tidak ada nilai default
`top_k`	Melaporkan akurasi top-k selama pelatihan. Parameter ini harus lebih besar dari 1, karena akurasi pelatihan top-1 sama dengan akurasi pelatihan reguler yang telah dilaporkan. Opsional Nilai yang valid: bilangan bulat positif lebih besar dari 1. Nilai default: tidak ada nilai default
`use_pretrained_model`	Tandai untuk menggunakan model pra-terlatih untuk pelatihan. Jika diatur ke 1, maka model yang telah dilatih sebelumnya dengan jumlah lapisan yang sesuai dimuat dan digunakan untuk pelatihan. Hanya lapisan FC atas yang diinisialisasi ulang dengan bobot acak. Jika tidak, jaringan dilatih dari awal. Opsional Nilai yang valid: 0 atau 1 Nilai default: 0

Nama Parameter	Deskripsi
`use_weighted_loss`	Bendera untuk menggunakan kehilangan entropi silang tertimbang untuk klasifikasi multi-label (hanya digunakan saat`multi_label = 1`), di mana bobot dihitung berdasarkan distribusi kelas. Opsional Nilai yang valid: 0 atau 1 Nilai default: 0
`weight_decay`	Koefisien pembusukan berat untuk`sgd`dan`nag`, diabaikan untuk pengoptimal lainnya. Opsional Nilai yang valid: float. Rentang dalam [0, 1]. Nilai default: 0.0001

Tune Model Klasifikasi Gambar

Penyetelan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi selengkapnya tentang penyetelan model, lihat[Lakukan Penyetelan Model Otomatis dengan SageMaker](#).

Metrik Dihitung oleh Algoritma Klasifikasi Gambar

Algoritma klasifikasi gambar adalah algoritma yang diawasi. Ini melaporkan metrik akurasi yang dihitung selama pelatihan. Saat menyetel model, pilih metrik ini sebagai metrik objektif.

Nama Metrik	Deskripsi	Arah Optimasi
`validation:accuracy`	Rasio jumlah prediksi yang benar dengan jumlah prediksi yang dibuat.	Maksimalkan

Hyperparameter klasifikasi gambar merdu

Tune model klasifikasi gambar dengan hyperparameters berikut. Hyperparameter yang memiliki dampak terbesar pada metrik objektif klasifikasi gambar adalah:`mini_batch_size`,`learning_rate`, dan`optimizer`. Tune hyperparameter terkait pengoptimalan, seperti`momentum`,`weight_decay`,`beta_1`,`beta_2`,`eps`, dan`gamma`, berdasarkan yang dipilih`optimizer`. Misalnya, gunakan`beta_1`dan`beta_2`hanya ketika`adam`adalah`optimizer`.

Untuk informasi lebih lanjut tentang hyperparameter mana yang digunakan di setiap pengoptimal, lihatHyperparameter Klasifikasi Gambar.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`beta_1`	ContinuousParameterRanges	MinValue: 1e-6MaxValue: 0,999
`beta_2`	ContinuousParameterRanges	MinValue: 1e-6MaxValue: 0,999
`eps`	ContinuousParameterRanges	MinValue: 1e-8MaxValue: 1.0
`gamma`	ContinuousParameterRanges	MinValue: 1e-8MaxValue: 0,999
`learning_rate`	ContinuousParameterRanges	MinValue: 1e-6MaxValue: 0,5
`mini_batch_size`	IntegerParameterRanges	MinValue: 8,MaxValue: 512

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
momentum	ContinuousParameterRanges	MinValue: 0,0,MaxValue: 0,999
optimizer	CategoricalParameterRanges	['sgd', 'adam', 'rmsprop', 'mengomel']
weight_decay	ContinuousParameterRanges	MinValue: 0,0,MaxValue: 0,999

Klasifikasi Gambar - TensorFlow

Algoritma Amazon SageMaker Image Classification - adalah TensorFlow algoritma pembelajaran yang diawasi yang mendukung pembelajaran transfer dengan banyak model yang telah dilatih sebelumnya dari Hub. TensorFlow Gunakan pembelajaran transfer untuk menyempurnakan salah satu model terlatih yang tersedia pada kumpulan data Anda sendiri, meskipun sejumlah besar data gambar tidak tersedia. Algoritma klasifikasi gambar mengambil gambar sebagai input dan menghasilkan probabilitas untuk setiap label kelas yang disediakan. Kumpulan data pelatihan harus terdiri dari gambar dalam format.jpg, .jpeg, atau .png.

Topik

- Cara menggunakan Klasifikasi SageMaker Gambar - TensorFlow algoritma
- Antarmuka input dan output untuk Klasifikasi Gambar - TensorFlow algoritma
- Rekomendasi instans Amazon EC2 untuk Klasifikasi Gambar - algoritma TensorFlow
- Klasifikasi Gambar - TensorFlow contoh buku catatan
- Bagaimana Klasifikasi Gambar - TensorFlow Bekerja
- TensorFlow Model Hub
- Klasifikasi Gambar - TensorFlow Hyperparameters
- Menyetel Klasifikasi Gambar - TensorFlow model

Cara menggunakan Klasifikasi SageMaker Gambar - TensorFlow algoritma

Anda dapat menggunakan Klasifikasi Gambar - TensorFlow sebagai algoritma SageMaker bawaan Amazon. Bagian berikut menjelaskan cara menggunakan Klasifikasi Gambar - TensorFlow

dengan SageMaker Python SDK. Untuk informasi tentang cara menggunakan Klasifikasi Gambar - TensorFlow dari UI Amazon SageMaker Studio Classic, lihatSageMaker JumpStart.

TensorFlow Algoritma Klasifikasi Gambar - mendukung pembelajaran transfer menggunakan salah satu model TensorFlow Hub terlatih sebelumnya yang kompatibel. Untuk daftar semua model terlatih yang tersedia, lihatTensorFlow Model Hub. Setiap model yang telah dilatih sebelumnya memiliki keunikan`model_id`. Contoh berikut menggunakan MobileNet V2 1.00 224 (`model_id`:`tensorflow-ic-imagenet-mobilenet-v2-100-224-classification-4`) untuk menyempurnakan dataset kustom. Model yang telah dilatih sebelumnya semuanya telah diunduh sebelumnya dari TensorFlow Hub dan disimpan dalam bucket Amazon S3 sehingga pekerjaan pelatihan dapat berjalan dalam isolasi jaringan. Gunakan artefak pelatihan model yang dibuat sebelumnya ini untuk membangun Estimator. SageMaker

Pertama, ambil URI image Docker, URI skrip pelatihan, dan URI model yang telah dilatih sebelumnya. Kemudian, ubah hyperparameters sesuai keinginan Anda. Anda dapat melihat kamus Python dari semua hyperparameters yang tersedia dan nilai defaultnya dengan. `hyperparameters.retrieve_default` Untuk informasi selengkapnya, lihat Klasifikasi Gambar - TensorFlow Hyperparameters. Gunakan nilai-nilai ini untuk membangun SageMaker Estimator.

> **Note**
>
> Nilai hyperparameter default berbeda untuk model yang berbeda. Untuk model yang lebih besar, ukuran batch default lebih kecil dan `train_only_top_layer` hyperparameter diatur ke`"True"`.

Contoh ini menggunakan `tf_flowers`dataset, yang berisi lima kelas gambar bunga. Kami mengunduh dataset sebelumnya dari TensorFlow bawah lisensi Apache 2.0 dan membuatnya tersedia dengan Amazon S3. Untuk menyempurnakan model Anda, hubungi `.fit` menggunakan lokasi Amazon S3 dari kumpulan data pelatihan Anda.

```
from sagemaker import image_uris, model_uris, script_uris, hyperparameters
from sagemaker.estimator import Estimator

model_id, model_version = "tensorflow-ic-imagenet-mobilenet-v2-100-224-
classification-4", "*"
training_instance_type = "ml.p3.2xlarge"

# Retrieve the Docker image
```

```
train_image_uri =
 image_uris.retrieve(model_id=model_id,model_version=model_version,image_scope="training",insta

# Retrieve the training script
train_source_uri = script_uris.retrieve(model_id=model_id, model_version=model_version,
 script_scope="training")

# Retrieve the pretrained model tarball for transfer learning
train_model_uri = model_uris.retrieve(model_id=model_id, model_version=model_version,
 model_scope="training")

# Retrieve the default hyper-parameters for fine-tuning the model
hyperparameters = hyperparameters.retrieve_default(model_id=model_id,
 model_version=model_version)

# [Optional] Override default hyperparameters with custom values
hyperparameters["epochs"] = "5"

# The sample training data is available in the following S3 bucket
training_data_bucket = f"jumpstart-cache-prod-{aws_region}"
training_data_prefix = "training-datasets/tf_flowers/"

training_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}"

output_bucket = sess.default_bucket()
output_prefix = "jumpstart-example-ic-training"
s3_output_location = f"s3://{output_bucket}/{output_prefix}/output"

# Create SageMaker Estimator instance
tf_ic_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1,
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location,
)

# Use S3 path of the training data to launch SageMaker TrainingJob
```

```
tf_ic_estimator.fit({"training": training_dataset_s3_path}, logs=True)
```

Antarmuka input dan output untuk Klasifikasi Gambar - TensorFlow algoritma

Setiap model terlatih yang tercantum dalam Model TensorFlow Hub dapat disetel dengan baik ke kumpulan data apa pun dengan sejumlah kelas gambar. Perhatikan cara memformat data pelatihan Anda untuk masukan ke TensorFlow model Klasifikasi Gambar.

- Format input data pelatihan: Data pelatihan Anda harus berupa direktori dengan subdirektori sebanyak jumlah kelas. Setiap subdirektori harus berisi gambar milik kelas itu dalam format.jpg, .jpeg, atau .png.

Berikut ini adalah contoh struktur direktori input. Contoh dataset ini memiliki dua kelas: `roses` dan`dandelion`. File gambar di setiap folder kelas dapat memiliki nama apa pun. Direktori input harus di-host di bucket Amazon S3 dengan jalur yang mirip dengan berikut ini:. `s3://`*`bucket_name`*`/`*`input_directory`*`/` Perhatikan bahwa trailing `/` diperlukan.

```
input_directory
    |--roses
        |--abc.jpg
        |--def.jpg
    |--dandelion
        |--ghi.jpg
        |--jkl.jpg
```

Model terlatih mengeluarkan file pemetaan label yang memetakan nama folder kelas ke indeks dalam daftar probabilitas kelas keluaran. Pemetaan ini dalam urutan abjad. Misalnya, pada contoh sebelumnya, kelas dandelion adalah indeks 0 dan kelas mawar adalah indeks 1.

Setelah pelatihan, Anda memiliki model yang disetel dengan baik yang dapat Anda latih lebih lanjut menggunakan pelatihan tambahan atau diterapkan untuk inferensi. TensorFlow Algoritma Klasifikasi Gambar secara otomatis menambahkan tanda tangan pra-pemrosesan dan pasca-pemrosesan ke model yang disetel dengan baik sehingga dapat mengambil gambar sebagai probabilitas kelas input dan pengembalian. Indeks kelas pemetaan file ke label kelas disimpan bersama dengan model.

Pelatihan tambahan

Anda dapat menyemai pelatihan model baru dengan artefak dari model yang Anda latih sebelumnya. SageMaker Pelatihan tambahan menghemat waktu pelatihan ketika Anda ingin melatih model baru dengan data yang sama atau serupa.

> **Note**
>
> Anda hanya dapat menyemai Klasifikasi SageMaker Gambar - TensorFlow model dengan Klasifikasi Gambar lain - TensorFlow model yang dilatih SageMaker.

Anda dapat menggunakan kumpulan data apa pun untuk pelatihan tambahan, selama kumpulan kelas tetap sama. Langkah pelatihan inkremental mirip dengan langkah fine-tuning, tetapi alih-alih memulai dengan model yang telah dilatih sebelumnya, Anda mulai dengan model fine-tuning yang ada. Untuk contoh pelatihan tambahan dengan TensorFlow algoritma Klasifikasi SageMaker Gambar -, lihat buku catatan contoh Pengantar SageMaker TensorFlow - Klasifikasi Gambar.

Inferensi dengan Klasifikasi Gambar - algoritma TensorFlow

Anda dapat meng-host model yang disetel dengan baik yang dihasilkan dari pelatihan Klasifikasi TensorFlow Gambar Anda untuk inferensi. Setiap gambar input untuk inferensi harus dalam `.jpg`,. `jpeg`, atau `.png` format dan menjadi tipe konten`application/x-image`. Klasifikasi Gambar - TensorFlow algoritma mengubah ukuran gambar input secara otomatis.

Menjalankan inferensi menghasilkan nilai probabilitas, label kelas untuk semua kelas, dan label prediksi yang sesuai dengan indeks kelas dengan probabilitas tertinggi yang dikodekan dalam format JSON. Klasifikasi Gambar - TensorFlow model memproses satu gambar per permintaan dan hanya menghasilkan satu baris. Berikut ini adalah contoh respons format JSON:

```
accept: application/json;verbose

 {"probabilities": [prob_0, prob_1, prob_2, ...],
  "labels":        [label_0, label_1, label_2, ...],
  "predicted_label": predicted_label}
```

Jika `accept` diatur ke`application/json`, maka model hanya menghasilkan probabilitas. Untuk informasi lebih lanjut tentang pelatihan dan inferensi dengan TensorFlow algoritma Klasifikasi Gambar -, lihat buku catatan contoh Pengantar SageMaker TensorFlow - Klasifikasi Gambar.

Rekomendasi instans Amazon EC2 untuk Klasifikasi Gambar - algoritma TensorFlow

Klasifikasi Gambar - TensorFlow algoritma mendukung semua instans CPU dan GPU untuk pelatihan, termasuk:

- `ml.p2.xlarge`

- `ml.p2.16xlarge`
- `ml.p3.2xlarge`
- `ml.p3.16xlarge`
- `ml.g4dn.xlarge`
- `ml.g4dn.16.xlarge`
- `ml.g5.xlarge`
- `ml.g5.48xlarge`

Kami merekomendasikan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Instance CPU (seperti M5) dan GPU (P2, P3, G4dn, atau G5) dapat digunakan untuk inferensi.

Klasifikasi Gambar - TensorFlow contoh buku catatan

Untuk informasi selengkapnya tentang cara menggunakan TensorFlow algoritma Klasifikasi SageMaker Gambar - untuk pembelajaran transfer pada kumpulan data khusus, lihat buku catatan Pengantar SageMaker TensorFlow - Klasifikasi Gambar.

Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Bagaimana Klasifikasi Gambar - TensorFlow Bekerja

Klasifikasi Gambar - TensorFlow algoritma mengambil gambar sebagai input dan mengklasifikasikannya ke dalam salah satu label kelas output. Berbagai jaringan pembelajaran mendalam seperti MobileNet, ResNet, Inception, dan EfficientNet sangat akurat untuk klasifikasi gambar. Ada juga jaringan pembelajaran mendalam yang dilatih pada kumpulan data gambar besar, seperti ImageNet, yang memiliki lebih dari 11 juta gambar dan hampir 11.000 kelas. Setelah jaringan dilatih dengan ImageNet data, Anda kemudian dapat menyempurnakan jaringan pada kumpulan data dengan fokus khusus untuk melakukan tugas klasifikasi yang lebih spesifik. TensorFlow Algoritma Klasifikasi SageMaker Gambar Amazon mendukung pembelajaran transfer pada banyak model yang telah dilatih sebelumnya yang tersedia di TensorFlow Hub.

Menurut jumlah label kelas dalam data pelatihan Anda, lapisan klasifikasi dilampirkan ke model TensorFlow Hub yang telah dilatih sebelumnya pilihan Anda. Lapisan klasifikasi terdiri dari lapisan

putus sekolah, lapisan padat, dan lapisan yang terhubung sepenuhnya dengan regularizer 2-norma yang diinisialisasi dengan bobot acak. Model ini memiliki hiperparameter untuk tingkat putus sekolah dari lapisan putus sekolah dan faktor regularisasi L2 untuk lapisan padat. Anda kemudian dapat menyempurnakan seluruh jaringan (termasuk model yang telah dilatih sebelumnya) atau hanya lapisan klasifikasi teratas pada data pelatihan baru. Dengan metode pembelajaran transfer ini, pelatihan dengan kumpulan data yang lebih kecil dimungkinkan.

TensorFlow Model Hub

Model terlatih berikut tersedia untuk digunakan untuk pembelajaran transfer dengan TensorFlow algoritma Klasifikasi Gambar -.

Model berikut bervariasi secara signifikan dalam ukuran, jumlah parameter model, waktu pelatihan, dan latensi inferensi untuk setiap kumpulan data tertentu. Model terbaik untuk kasus penggunaan Anda bergantung pada kompleksitas kumpulan data fine-tuning Anda dan persyaratan apa pun yang Anda miliki pada waktu pelatihan, latensi inferensi, atau akurasi model.

Nama Model	**`model_id`**	Sumber
MobileNet V2 1,00 224	`tensorflow-ic-imagenet-mobilenet-v2-100-224-classification-4`	TensorFlow Tautan hub
MobileNet V2 0,75 224	`tensorflow-ic-imagenet-mobilenet-v2-075-224-classification-4`	TensorFlow Tautan hub
MobileNet V2 0,50 224	`tensorflow-ic-imagenet-mobilenet-v2-050-224-classification-4`	TensorFlow Tautan hub
MobileNet V2 0,35 224	`tensorflow-ic-imagenet-mobilenet-v2-035-224-classification-4`	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
MobileNet V2 1,40 224	tensorflow-ic-imagenet-mobilenet-v2-140-224-classification-4	TensorFlow Tautan hub
MobileNet V2 1,30 224	tensorflow-ic-imagenet-mobilenet-v2-130-224-classification-4	TensorFlow Tautan hub
MobileNet V2	tensorflow-ic-tf2-preview-mobilenet-v2-classification-4	TensorFlow Tautan hub
Asalnya V3	tensorflow-ic-imagenet-inception-v3-classification-4	TensorFlow Tautan hub
Inception V2	tensorflow-ic-imagenet-inception-v2-classification-4	TensorFlow Tautan hub
Asalnya V1	tensorflow-ic-imagenet-inception-v1-classification-4	TensorFlow Tautan hub
Pratinjau V3 Inception	tensorflow-ic-tf2-preview-inception-v3-classification-4	TensorFlow Tautan hub
Inception V2 ResNet	tensorflow-ic-imagenet-inception-resnet-v2-classification-4	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
ResNet V2 50	tensorflow-ic-imagenet-resnet-v2-50-classification-4	TensorFlow Tautan hub
ResNet V2 101	tensorflow-ic-imagenet-resnet-v2-101-classification-4	TensorFlow Tautan hub
ResNet V2 152	tensorflow-ic-imagenet-resnet-v2-152-classification-4	TensorFlow Tautan hub
ResNet V1 50	tensorflow-ic-imagenet-resnet-v1-50-classification-4	TensorFlow Tautan hub
ResNet V1 101	tensorflow-ic-imagenet-resnet-v1-101-classification-4	TensorFlow Tautan hub
ResNet V1 152	tensorflow-ic-imagenet-resnet-v1-152-classification-4	TensorFlow Tautan hub
ResNet 50	tensorflow-ic-imagenet-resnet-50-classification-4	TensorFlow Tautan hub
EfficientNet B0	tensorflow-ic-efficientnet-b0-classification-1	TensorFlow Tautan hub
EfficientNet B1	tensorflow-ic-efficientnet-b1-classification-1	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
EfficientNet B2	tensorflow-ic-efficientnet-b2-classification-1	TensorFlow Tautan hub
EfficientNet B3	tensorflow-ic-efficientnet-b3-classification-1	TensorFlow Tautan hub
EfficientNet B4	tensorflow-ic-efficientnet-b4-classification-1	TensorFlow Tautan hub
EfficientNet B5	tensorflow-ic-efficientnet-b5-classification-1	TensorFlow Tautan hub
EfficientNet B6	tensorflow-ic-efficientnet-b6-classification-1	TensorFlow Tautan hub
EfficientNet B7	tensorflow-ic-efficientnet-b7-classification-1	TensorFlow Tautan hub
EfficientNet B0 Lite	tensorflow-ic-efficientnet-lite0-classification-2	TensorFlow Tautan hub
EfficientNet B1 Lite	tensorflow-ic-efficientnet-lite1-classification-2	TensorFlow Tautan hub
EfficientNet B2 Lite	tensorflow-ic-efficientnet-lite2-classification-2	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
EfficientNet B3 Lite	`tensorflow-ic-efficientnet-lite3-classification-2`	TensorFlow Tautan hub
EfficientNet B4 Lite	`tensorflow-ic-efficientnet-lite4-classification-2`	TensorFlow Tautan hub
MobileNet V1 1,00 224	`tensorflow-ic-imagenet-mobilenet-v1-100-224-classification-4`	TensorFlow Tautan hub
MobileNet V1 1,00 192	`tensorflow-ic-imagenet-mobilenet-v1-100-192-classification-4`	TensorFlow Tautan hub
MobileNet V1 1,00 160	`tensorflow-ic-imagenet-mobilenet-v1-100-160-classification-4`	TensorFlow Tautan hub
MobileNet V1 1,00 128	`tensorflow-ic-imagenet-mobilenet-v1-100-128-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,75 224	`tensorflow-ic-imagenet-mobilenet-v1-075-224-classification-4`	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
MobileNet V1 0,75 192	`tensorflow-ic-imagenet-mobilenet-v1-075-192-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,75 160	`tensorflow-ic-imagenet-mobilenet-v1-075-160-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,75 128	`tensorflow-ic-imagenet-mobilenet-v1-075-128-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,50 224	`tensorflow-ic-imagenet-mobilenet-v1-050-224-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,50 192	`tensorflow-ic-imagenet-mobilenet-v1-050-192-classification-4`	TensorFlow Tautan hub
MobileNet V1 1,00 160	`tensorflow-ic-imagenet-mobilenet-v1-050-160-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,50 128	`tensorflow-ic-imagenet-mobilenet-v1-050-128-classification-4`	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
MobileNet V1 0,25 224	`tensorflow-ic-imagenet-mobilenet-v1-025-224-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,25 192	`tensorflow-ic-imagenet-mobilenet-v1-025-192-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,25 160	`tensorflow-ic-imagenet-mobilenet-v1-025-160-classification-4`	TensorFlow Tautan hub
MobileNet V1 0,25 128	`tensorflow-ic-imagenet-mobilenet-v1-025-128-classification-4`	TensorFlow Tautan hub
Bit-S R50x1	`tensorflow-ic-bit-s-r50x1-ilsvrc2012-classification-1`	TensorFlow Tautan hub
Bit-S R50x3	`tensorflow-ic-bit-s-r50x3-ilsvrc2012-classification-1`	TensorFlow Tautan hub
Bit-s R101x1	`tensorflow-ic-bit-s-r101x1-ilsvrc2012-classification-1`	TensorFlow Tautan hub
Bit-s R101x3	`tensorflow-ic-bit-s-r101x3-ilsvrc2012-classification-1`	TensorFlow Tautan hub

Nama Model	**model_id**	Sumber
Bit-M R50x1	`tensorflow-ic-bit-m-r50x1-ilsvrc2012-classification-1`	TensorFlow Tautan hub
Bit-M R50x3	`tensorflow-ic-bit-m-r50x3-ilsvrc2012-classification-1`	TensorFlow Tautan hub
Bit-M R101x1	`tensorflow-ic-bit-m-r101x1-ilsvrc2012-classification-1`	TensorFlow Tautan hub
Bit-M R101x3	`tensorflow-ic-bit-m-r101x3-ilsvrc2012-classification-1`	TensorFlow Tautan hub
Bit-M R50x1 -21k ImageNet	`tensorflow-ic-bit-m-r50x1-imagenet21k-classification-1`	TensorFlow Tautan hub
Bit-M R50x3 -21k ImageNet	`tensorflow-ic-bit-m-r50x3-imagenet21k-classification-1`	TensorFlow Tautan hub
Bit-M R101x1 -21k ImageNet	`tensorflow-ic-bit-m-r101x1-imagenet21k-classification-1`	TensorFlow Tautan hub
Bit-M R101x3 -21k ImageNet	`tensorflow-ic-bit-m-r101x3-imagenet21k-classification-1`	TensorFlow Tautan hub

Klasifikasi Gambar - TensorFlow Hyperparameters

Hyperparameters adalah parameter yang ditetapkan sebelum model pembelajaran mesin mulai belajar. Hyperparameter berikut didukung oleh TensorFlow algoritma Klasifikasi Gambar SageMaker

bawaan Amazon. Lihat Menyetel Klasifikasi Gambar - TensorFlow model untuk informasi tentang tuning hyperparameter.

Nama Parameter	Deskripsi
`augmentation`	Setel `"True"` untuk menerapkan`augmentation_rando m_flip` ,`augmentation_random_rotation` , dan `augmentation_random_zoom` ke data pelatihan. Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.
`augmentation_rando m_flip`	Menunjukkan mode flip mana yang akan digunakan untuk augmentasi data saat `augmentation` disetel ke`"True"`. Untuk informasi lebih lanjut, lihat RandomFlipdi TensorFlow dokumentasi. Nilai yang valid: string, salah satu dari berikut ini: (`"horizont al_and_vertical"` ,`"vertical"` , atau`"None"`). Nilai default:`"horizontal_and_vertical"` .
`augmentation_rando m_rotation`	Menunjukkan berapa banyak rotasi yang akan digunakan untuk augmentasi data saat `augmentation` disetel ke`"True"`. Nilai mewakili sebagian kecil dari 2π. Nilai positif berputar berlawana n arah jarum jam sementara nilai negatif berputar searah jarum jam. `0`berarti tidak ada rotasi. Untuk informasi lebih lanjut, lihat RandomRotationdi TensorFlow dokumentasi. Nilai yang valid: float, range: [`-1.0`,`1.0`]. Nilai default:`0.2`.
`augmentation_rando m_zoom`	Menunjukkan berapa banyak zoom vertikal yang akan digunakan untuk augmentasi data saat `augmentation` disetel ke`"True"`. Nilai positif diperkecil sementara nilai negatif diperbesar. `0`berarti tidak ada zoom. Untuk informasi lebih lanjut, lihat RandomZoomdi TensorFlow dokumentasi.

Nama Parameter	Deskripsi
	Nilai yang valid: float, range: [-1.0,1.0]. Nilai default:0.1.
batch_size	Ukuran batch untuk pelatihan. Untuk pelatihan tentang instance dengan beberapa GPU, ukuran batch ini digunakan di seluruh GPU. Nilai yang valid: bilangan bulat positif. Nilai default:32.
beta_1	Beta1 untuk pengoptimal. "adam" Merupakan tingkat peluruhan eksponensial untuk perkiraan momen pertama. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [0.0,1.0]. Nilai default:0.9.
beta_2	Beta2 untuk pengoptimal. "adam" Merupakan tingkat peluruhan eksponensial untuk perkiraan momen kedua. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [0.0,1.0]. Nilai default:0.999.
binary_mode	Ketika binary_mode diatur ke"True", model mengembal ikan nomor probabilitas tunggal untuk kelas positif dan dapat menggunakan eval_metric opsi tambahan. Gunakan hanya untuk masalah klasifikasi biner. Nilai yang valid: string, baik: ("True"atau"False"). Nilai default:"False".

Nama Parameter	Deskripsi
`dropout_rate`	Tingkat putus sekolah untuk lapisan putus sekolah di lapisan klasifikasi atas. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default: `0.2`
`early_stopping`	Atur `"True"` untuk menggunakan logika penghentian awal selama pelatihan. Jika`"False"`, berhenti dini tidak digunakan. Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.
`early_stopping_min_delta`	Perubahan minimum yang diperlukan untuk memenuhi syarat sebagai perbaikan. Perubahan absolut kurang dari nilai `early_stopping_min_delta` tidak memenuhi syarat sebagai perbaikan. Digunakan hanya ketika `early_stopping` diatur ke`"True"`. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.0`.
`early_stopping_patience`	Jumlah zaman untuk melanjutkan pelatihan tanpa perbaikan. Digunakan hanya ketika `early_stopping` diatur ke`"True"`. Nilai yang valid: bilangan bulat positif. Nilai default:`5`.
`epochs`	Jumlah zaman pelatihan. Nilai yang valid: bilangan bulat positif. Nilai default:`3`.

Nama Parameter	Deskripsi
epsilon	Epsilon untuk"adam",, "rmsprop" "adadelta" , dan "adagrad" pengoptimal. Biasanya diatur ke nilai kecil untuk menghindari pembagian dengan 0. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [0.0,1.0]. Nilai default:1e-7.
eval_metric	Jika binary_mode diatur ke"False", hanya eval_metric bisa"accuracy" . Jika binary_mode ya"True", pilih salah satu nilai yang valid. Untuk informasi selengkapnya, lihat Metrik dalam TensorFlow dokumentasi. Nilai yang valid: string, salah satu dari berikut ini: ("accuracy " "precision" ,,"recall","auc", atau"prc"). Nilai default:"accuracy" .
image_resize_interpolation	Menunjukkan metode interpolasi yang digunakan saat mengubah ukuran gambar. Untuk informasi selengkapnya, lihat image.resize dalam dokumentasi. TensorFlow Nilai yang valid: string, salah satu dari berikut ini: ("bilinear " "nearest" ,"bicubic" ,"area", "lanczos3 " ,"lanczos5" ,"gaussian" , atau"mitchellcubic"). Nilai default:"bilinear" .
initial_accumulator_value	Nilai awal untuk akumulator, atau nilai momentum per parameter , untuk pengoptimal. "adagrad" Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [0.0,1.0]. Nilai default:0.0001.

Nama Parameter	Deskripsi
`label_smoothing`	Menunjukkan seberapa banyak untuk merilekskan kepercaya an pada nilai label. Misalnya, jika `label_smoothing` ya`0.1`, maka label non-target adalah `0.1/num_classes` dan label target adalah`0.9+0.1/num_classes` . Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.1`.
`learning_rate`	Tingkat pembelajaran pengoptimal. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.001`.
`momentum`	Momentum untuk`"sgd"`,`"nesterov"` , dan `"rmsprop"` pengoptimal. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.9`.
`optimizer`	Jenis pengoptimal. Untuk informasi selengkapnya, lihat Pengoptimal dalam dokumentasi. TensorFlow Nilai yang valid: string, salah satu dari berikut ini: (`"adam"``"sgd"`,,`"nesterov"` ,`"rmsprop"` , `"adagrad"` ,`"adadelta"` ).  Nilai default:`"adam"`.
`regularizers_l2`	Faktor regularisasi L2 untuk lapisan padat di lapisan klasifikasi. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`.0001`.

Nama Parameter	Deskripsi
`reinitialize_top_layer`	Jika disetel ke`"Auto"`, parameter lapisan klasifikasi atas diinisialisasi ulang selama fine-tuning. Untuk pelatihan tambahan, parameter lapisan klasifikasi teratas tidak diinisialisasi ulang kecuali disetel ke. `"True"` Nilai yang valid: string, salah satu dari berikut ini: (`"Auto"`, `"True"` atau`"False"`). Nilai default:`"Auto"`.
`rho`	Faktor diskon untuk gradien `"adadelta"` dan `"rmsprop"` pengoptimal. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.95`.
`train_only_top_layer`	Jika`"True"`, hanya parameter lapisan klasifikasi teratas yang disetel dengan baik. Jika`"False"`, semua parameter model disetel dengan baik. Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.

Menyetel Klasifikasi Gambar - TensorFlow model

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hiperparameter pada kumpulan data Anda. Anda memilih hyperparameters yang dapat disetel, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hiperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik dihitung oleh Klasifikasi Gambar - algoritma TensorFlow

Algoritma klasifikasi gambar adalah algoritma yang diawasi. Ini melaporkan metrik akurasi yang dihitung selama pelatihan. Saat menyetel model, pilih metrik ini sebagai metrik objektif.

Nama Metrik	Deskripsi	Arah Optimasi
`validation:accuracy`	Rasio jumlah prediksi yang benar dengan jumlah prediksi yang dibuat.	Maksimalkan

Klasifikasi Gambar yang Dapat Disetel - hyperparameters TensorFlow

Sesuaikan model klasifikasi gambar dengan hyperparameter berikut. Hiperparameter yang memiliki dampak terbesar pada metrik objektif klasifikasi gambar adalah:`batch_size`,`learning_rate`, dan. `optimizer` Setel hiperparameter terkait pengoptimal, seperti,,`momentum`,, `regularizers_l2` `beta_1``beta_2`, dan `eps` berdasarkan yang dipilih. `optimizer` Misalnya, gunakan `beta_1` dan `beta_2` `adam` hanya kapan`optimizer`.

Untuk informasi lebih lanjut tentang hiperparameter mana yang digunakan untuk masing-masing`optimizer`, lihatKlasifikasi Gambar - TensorFlow Hyperparameters.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`batch_size`	IntegerParameterRanges	MinValue: 8, MaxValue: 512
`beta_1`	ContinuousParameterRanges	MinValue: 1e-6,: 0.999 MaxValue
`beta_2`	ContinuousParameterRanges	MinValue: 1e-6,: 0.999 MaxValue
`eps`	ContinuousParameterRanges	MinValue: 1e-8, MaxValue: 1.0
`learning_rate`	ContinuousParameterRanges	MinValue: 1e-6, MaxValue: 0,5

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
momentum	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.999
optimizer	CategoricalParameterRanges	['sgd', 'adam', 'rmsprop', 'nesterov', 'adagrad', 'adadelta']
regularizers_l2	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.999
train_only_top_layer	ContinuousParameterRanges	['Benar', 'Salah']

Deteksi

Algoritma Amazon SageMaker Object Detection - MXNet mendeteksi dan mengklasifikasikan objek dalam gambar menggunakan jaringan neural dalam tunggal. Ini adalah algoritma pembelajaran yang diawasi yang mengambil gambar sebagai input dan mengidentifikasi semua contoh objek dalam adegan gambar. Objek dikategorikan ke dalam salah satu kelas dalam koleksi tertentu dengan skor kepercayaan bahwa itu milik kelas. Lokasi dan skalanya pada gambar ditunjukkan oleh kotak pembatas persegi panjang. Ini menggunakan kerangka kerja Single Shot multibox Detector (SSD) dan mendukung dua jaringan dasar: VGG dan ResNet. Jaringan dapat dilatih dari awal, atau dilatih dengan model yang telah dilatih sebelumnya pada ImageNetdataset.

Topik

- Antarmuka Input/Output untuk Algoritma Deteksi Objek
- Rekomendasi Instans EC2 untuk Algoritma Deteksi Objek
- Notebook
- Cara Kerja
- Hyperparameter
- Tune Model Deteksi Objek
- Format

Antarmuka Input/Output untuk Algoritma Deteksi Objek

Algoritma SageMaker Object Detection mendukung tipe konten RecordiO (`application/x-recordio`) dan image (`image/pngimage/jpeg`,, dan`application/x-image`) untuk pelatihan dalam mode file dan mendukung RecordIO (`application/x-recordio`) untuk pelatihan dalam mode pipa. Namun Anda juga dapat melatih dalam mode pipa menggunakan file gambar (`image/png`,, dan`application/x-image`)`image/jpeg`, tanpa membuat file RecordiO, dengan menggunakan format manifes yang ditambah. Format input yang direkomendasikan untuk algoritma deteksi SageMaker objek Amazon adalah Apache MXNet RecordIO. Namun, Anda juga dapat menggunakan citra mentah dalam format .jpg atau .png. Algoritma hanya mendukung`application/x-image` untuk inferensi.

> **Note**
>
> Untuk mempertahankan interoperabilitas yang lebih baik dengan kerangka kerja deep learning yang ada, ini berbeda dari format data protobuf yang biasa digunakan oleh SageMaker algoritme Amazon lainnya.

Lihat detailNotebook lebih lanjut tentang format data.

Berlatih dengan Format RecordIO

Jika Anda menggunakan format RecordIo untuk pelatihan, tentukan saluran kereta dan validasi sebagai nilai untuk`InputDataConfig` parameter `CreateTrainingJob`permintaan. Tentukan satu RecordIO RecordiO (.rec) di saluran kereta dan satu file RecordiO di saluran validasi. Atur jenis konten untuk kedua saluran`application/x-recordio`. Contoh cara menghasilkan file RecordIO dapat ditemukan di notebook sampel deteksi objek. Anda juga dapat menggunakan alat dari GluonCV MXNet untuk menghasilkan file RecordIO untuk dataset populer seperti PASCAL Visual Object Classes dan Common Objects in Context (COCO).

Berlatih dengan Format Gambar

Jika Anda menggunakan format gambar untuk pelatihan, tentukan`train`,`validation`,`train_annotation`,, dan`validation_annotation` saluran sebagai nilai untuk`InputDataConfig` parameter `CreateTrainingJob`permintaan. Tentukan file data gambar individu (.jpg atau .png) untuk saluran kereta dan validasi. Untuk data anotasi, Anda dapat menggunakan format JSON. Tentukan file.json yang sesuai di`validation_annotation`

salurantrain_annotation dan. Atur jenis konten untuk keempat saluran keimage/png atauimage/jpeg berdasarkan jenis gambar. Anda juga dapat menggunakan jenis kontenapplication/x-image ketika dataset Anda berisi gambar.jpg dan .png. Berikut ini adalah contoh file.json.

```
{
   "file": "your_image_directory/sample_image1.jpg",
   "image_size": [
      {
         "width": 500,
         "height": 400,
         "depth": 3
      }
   ],
   "annotations": [
      {
         "class_id": 0,
         "left": 111,
         "top": 134,
         "width": 61,
         "height": 128
      },
      {
         "class_id": 0,
         "left": 161,
         "top": 250,
         "width": 79,
         "height": 143
      },
      {
         "class_id": 1,
         "left": 101,
         "top": 185,
         "width": 42,
         "height": 130
      }
   ],
   "categories": [
      {
         "class_id": 0,
         "name": "dog"
      },
      {
```

```
         "class_id": 1,
         "name": "cat"
      }
   ]
}
```

Setiap gambar membutuhkan file.json untuk anotasi, dan file.json harus memiliki nama yang sama dengan gambar yang sesuai. Nama file.json di atas harus “sample_image1.json”. Ada empat properti dalam anotasi file.json. Properti “file” menentukan path relatif dari file gambar. Misalnya, jika gambar latihan dan file.json yang sesuai disimpan dalam s3://*your_bucket* /train/sample_image dan s3://*your_bucket* /train_annotation, tentukan jalur untuk saluran kereta dan train_annotation Anda sebagai s3://*your_bucket* /train dan s3://*your_bucket* /train_annotation, masing-masing.

Dalam file.json, path relatif untuk gambar bernama sample_image1.jpg harus sample_image/sample_image1.jpg. `"image_size"`Properti menentukan dimensi gambar secara keseluruhan. Algoritma deteksi SageMaker objek saat ini hanya mendukung gambar 3-channel. `"annotations"`Properti menentukan kategori dan kotak pembatas untuk objek dalam gambar. Setiap objek dijelaskan oleh`"class_id"` indeks dan oleh empat koordinat kotak pembatas (`"left"`,,`"top"``"width"`,`"height"`). Nilai`"left"` (koordinat x) dan`"top"` (koordinat-y) mewakili sudut kiri kotak batas. Nilai`"width"` (koordinat x) dan`"height"` (koordinat-y) mewakili dimensi kotak batas. Origin (0, 0) adalah sudut kiri atas seluruh citra. Jika Anda memiliki beberapa objek dalam satu gambar, semua anotasi harus disertakan dalam satu file.json. `"categories"`Properti menyimpan pemetaan antara indeks kelas dan nama kelas. Indeks kelas harus diberi nomor berturut-turut dan penomoran harus dimulai dengan 0. `"categories"`Properti ini opsional untuk anotasi berkas.json

Latih dengan Format Gambar Manifes Augmented

Format manifes yang ditambah memungkinkan Anda melakukan pelatihan dalam mode pipa menggunakan file gambar tanpa perlu membuat file RecordiO. Anda perlu menentukan saluran kereta dan validasi sebagai nilai untuk`InputDataConfig` parameter `CreateTrainingJob`permintaan. Saat menggunakan format, file manifes S3 perlu dibuat yang berisi daftar gambar dan anotasi yang sesuai. Format file manifes harus dalam format Garis JSON di mana setiap baris mewakili satu sampel. Gambar ditentukan menggunakan`'source-ref'` tag yang menunjuk ke lokasi S3 gambar. Anotasi disediakan di bawah nilai`"AttributeNames"` parameter sebagaimana ditentukan dalam `CreateTrainingJob`permintaan. Hal ini juga dapat berisi metadata tambahan di bawah`metadata` tag, tetapi ini diabaikan oleh algoritma. Pada contoh berikut,`"AttributeNames` yang terkandung dalam daftar`["source-ref", "bounding-box"]`:

```
{"source-ref": "s3://your_bucket/image1.jpg", "bounding-box":{"image_size":[{ "width":
 500, "height": 400, "depth":3}], "annotations":[{"class_id": 0, "left": 111, "top":
 134, "width": 61, "height": 128}, {"class_id": 5, "left": 161, "top": 250, "width":
 80, "height": 50}]}, "bounding-box-metadata":{"class-map":{"0": "dog", "5": "horse"},
 "type": "groundtruth/object-detection"}}
{"source-ref": "s3://your_bucket/image2.jpg", "bounding-box":{"image_size":[{ "width":
 400, "height": 300, "depth":3}], "annotations":[{"class_id": 1, "left": 100, "top":
 120, "width": 43, "height": 78}]}, "bounding-box-metadata":{"class-map":{"1": "cat"},
 "type": "groundtruth/object-detection"}}
```

Urutan"`AttributeNames`" dalam file input penting saat melatih algoritma Object Detection. Ia menerima data pipa dalam urutan tertentu, dengan`image` pertama, diikuti oleh`annotations`. Jadi "AttributeNames" dalam contoh ini disediakan dengan"`source-ref`" pertama, diikuti oleh"`bounding-box`". Saat menggunakan Object Detection dengan Augmented Manifest, nilai parameter`RecordWrapperType` harus ditetapkan sebagai"`RecordIO`".

Untuk informasi selengkapnya tentang file manifes tambahan, lihatMenyediakan Dataset Metadata untuk Training Jobs dengan Augmented Manifest File.

Pelatihan

Anda juga dapat benih pelatihan model baru dengan artefak dari model yang Anda dilatih sebelumnya dengan SageMaker. Pelatihan tambahan menghemat waktu pelatihan saat Anda ingin melatih model baru dengan data yang sama atau serupa. SageMaker model deteksi objek dapat diunggulkan hanya dengan model deteksi objek built-in lainnya yang dilatih SageMaker.

Untuk menggunakan model yang telah dilatih sebelumnya, dalam `CreateTrainingJob`permintaan, tentukan`ChannelName` sebagai “model” dalam`InputDataConfig` parameter. Atur`ContentType` untuk saluran model ke`application/x-sagemaker-model`. Hyperparameter input dari model baru dan model terlatih yang Anda unggah ke saluran model harus memiliki pengaturan yang sama untuk parameter`base_network` dan`num_classes` input. Parameter ini menentukan arsitektur jaringan. Untuk file model yang telah dilatih sebelumnya, gunakan artefak model terkompresi (dalam format.tar.gz) keluaran oleh SageMaker. Anda dapat menggunakan format RecordiO atau gambar untuk input data.

Untuk informasi lebih lanjut tentang pelatihan tambahan dan untuk petunjuk tentang cara menggunakannya, lihatMenggunakan Pelatihan Inkremental di Amazon SageMaker.

Rekomendasi Instans EC2 untuk Algoritma Deteksi Objek

Algoritma deteksi objek mendukung keluarga instans GPU P2, P3, G4dn, dan G5. Sebaiknya gunakan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Anda dapat menjalankan algoritma deteksi objek pada pengaturan multi-GPU dan mult-machine untuk pelatihan terdistribusi.

Anda dapat menggunakan instans CPU (seperti C5 dan M5) dan GPU (seperti P3 dan G4dn) untuk inferensi.

Notebook

Untuk contoh notebook yang menunjukkan cara menggunakan algoritma SageMaker Object Detection untuk melatih dan meng-host model pada

Caltech Birds (CUB 200 2011) dataset menggunakan algoritma Single Shot multibox Detector, lihat Amazon SageMaker Object Detection for Bird Species. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMaker Examples untuk melihat daftar semua SageMaker sampel. Notebook contoh deteksi objek menggunakan algoritma Object Detection terletak di bagian Introduction to Amazon Algorithms. Untuk membuka notebook, klik tab Gunakan dan pilih Buat salinan.

Cara Kerja

Algoritma deteksi objek mengidentifikasi dan menempatkan semua contoh objek dalam gambar dari koleksi dikenal kategori objek. Algoritma mengambil gambar sebagai input dan output kategori yang objek milik, bersama dengan skor kepercayaan bahwa itu milik kategori. Algoritma ini juga memprediksi lokasi objek dan skala dengan kotak pembatas persegi panjang. Amazon SageMaker Object Detection menggunakan algoritma Single Shot multibox Detector (SSD) yang menggunakan jaringan saraf konvolusional (CNN) yang telah dilatih sebelumnya untuk tugas klasifikasi sebagai jaringan dasar. SSD menggunakan output lapisan menengah sebagai fitur untuk deteksi.

Berbagai CNN seperti VGG dan ResNettelah mencapai kinerja yang luar biasa pada tugas klasifikasi gambar. Deteksi objek di Amazon SageMaker mendukung VGG-16 dan ResNet -50 sebagai jaringan dasar untuk SSD. Algoritma dapat dilatih dalam mode pelatihan penuh atau dalam mode pembelajaran transfer. Dalam mode latihan penuh, jaringan dasar diinisialisasi dengan bobot acak dan kemudian dilatih pada data pengguna. Dalam mode pembelajaran transfer, bobot jaringan dasar dimuat dari model yang telah dilatih sebelumnya.

Algoritma deteksi objek menggunakan operasi augmentasi data standar, seperti flip, rescale, dan jitter, dengan cepat secara internal untuk membantu menghindari overfitting.

Hyperparameter

Dalam `CreateTrainingJob`permintaan, Anda menentukan algoritma pelatihan yang ingin Anda gunakan. Anda juga dapat menentukan hyperparameter khusus algoritma yang digunakan untuk membantu memperkirakan parameter model dari kumpulan data pelatihan. Tabel berikut mencantumkan hyperparameter yang disediakan oleh Amazon SageMaker untuk melatih algoritma deteksi objek. Untuk informasi selengkapnya tentang cara kerja pelatihan, lihatCara Kerja.

Nama Parameter	Deskripsi
`num_classes`	Jumlah kelas output. Parameter ini mendefinisikan dimensi output jaringan dan biasanya diatur ke jumlah kelas dalam dataset. Diperlukan Nilai yang valid: bilangan bulat positif
`num_training_samples`	Jumlah contoh pelatihan dalam kumpulan data masukan. **Note** Jika ada ketidakcocokan antara nilai ini dan jumlah sampel dalam set pelatihan, maka perilaku`lr_schedu ler_step` parameter akan tidak terdefinisi dan akurasi pelatihan terdistribusi dapat terpengaruh. Diperlukan Nilai yang valid: bilangan bulat positif
`base_network`	Arsitektur jaringan dasar untuk digunakan. Opsional Nilai yang valid: 'vgg-16' atau 'resnet-50'

Nama Parameter	Deskripsi
	Nilai default: 'vgg-16'
`early_stopping`	`True`untuk menggunakan logika penghentian awal selama pelatihan. `False`tidak menggunakannya. Opsional Nilai yang valid: `True` or `False` Nilai default: `False`
`early_stopping_min_epochs`	Jumlah minimum zaman yang harus dijalankan sebelum logika berhenti awal dapat dipanggil. Hal ini digunakan hanya ketika`early_stopping =True`. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 10
`early_stopping_patience`	Jumlah zaman yang harus menunggu sebelum mengakhiri pelatihan jika tidak ada perbaikan, seperti yang didefinisikan oleh`early_stopping_tolerance` hyperparameter, dibuat dalam metrik yang relevan. Hal ini digunakan hanya ketika`early_stopping =True`. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 5

Nama Parameter	Deskripsi
`early_stopping_tolerance`	Nilai toleransi bahwa perbaikan relatif dalam`validation:mAP` , rata-rata presisi rata-rata (mAP), diperlukan untuk melebihi untuk menghindari berhenti dini. Jika rasio perubahan dalam MAP dibagi dengan MAP terbaik sebelumnya lebih kecil dari`early_stopping_tolerance` nilai yang ditetapkan, penghentian awal menganggap bahwa tidak ada perbaikan. Hal ini digunakan hanya ketika`early_stopping =True`. Opsional Nilai yang valid: 0 ≤ 1 Nilai default: 0,0
`image_shape`	Ukuran gambar untuk gambar masukan. Kami rescale gambar input ke gambar persegi dengan ukuran ini. Kami merekomendasikan penggunaan 300 dan 512 untuk kinerja yang lebih baik. Opsional Nilai yang valid: bilangan bulat positif ≥300 Default: 300
`epochs`	Jumlah zaman pelatihan. Opsional Nilai yang valid: bilangan bulat positif Default: 30

Nama Parameter	Deskripsi
`freeze_layer_pattern`	Ekspresi reguler (regex) untuk pembekuan lapisan dalam jaringan dasar. Misalnya, jika kita menetapkan`freeze_layer_pattern ="^(conv1_|conv2_).*"` , maka setiap lapisan dengan nama yang berisi`"conv1_"` atau`"conv2_"` dibekukan, yang berarti bahwa bobot untuk lapisan ini tidak diperbarui selama pelatihan. Nama layer dapat ditemukan di file simbol jaringan vgg16-symbol.json dan resnet-50-symbol.json. Pembekuan lapisan berarti bobotnya tidak dapat dimodifikasi lebih lanjut. Hal ini dapat mengurangi waktu pelatihan secara signifikan dengan imbalan kerugian sederhana dalam akurasi. Teknik ini umumnya digunakan dalam pembelajaran transfer di mana lapisan bawah dalam jaringan dasar tidak perlu dilatih ulang. Opsional Nilai yang valid: String Default: Tidak ada lapisan beku.

Nama Parameter	Deskripsi
`kv_store`	Mode sinkronisasi pembaruan berat yang digunakan untuk pelatihan terdistribusi. Bobot dapat diperbarui baik secara sinkron atau asinkron di seluruh mesin. Pembaruan sinkron biasanya memberikan akurasi yang lebih baik daripada pembaruan asinkron tetapi bisa lebih lambat. Lihat tutorial Pelatihan Terdistribusi MXNet untuk detailnya. **Note** Parameter ini tidak berlaku untuk pelatihan mesin tunggal. Opsional Nilai yang valid: `'dist_sync'` atau `'dist_async'` • `'dist_sync'` : Gradien disinkronkan setelah setiap batch dengan semua pekerja. Dengan `'dist_sync'` , ukuran batch sekarang berarti ukuran batch yang digunakan pada setiap mesin. Jadi jika ada n mesin dan kita menggunakan ukuran batch b, maka dist_sync berperilaku seperti mesin tunggal dengan ukuran batch n* b. • `'dist_async'` : Melakukan pembaruan asinkron. Bobot diperbarui setiap kali gradien diterima dari mesin mana pun dan pembaruan bobotnya bersifat atom. Namun, pesanan tersebut tidak dijamin. Default: -

Nama Parameter	Deskripsi
`label_width`	Lebar label padding gaya yang digunakan untuk menyinkronkan seluruh data pelatihan dan validasi. Misalnya, jika satu gambar dalam data berisi paling banyak 10 objek, dan setiap anotasi objek ditentukan dengan 5 angka, [class_id, kiri, atas, lebar, tinggi], maka`label_width` harus tidak lebih kecil dari (10* 5 + panjang informasi header). Panjang informasi header biasanya 2. Sebaiknya gunakan sedikit lebih besar`label_width` untuk pelatihan, seperti 60 untuk contoh ini. Opsional Nilai yang valid: Bilangan bulat positif cukup besar untuk mengakomodasi panjang informasi anotasi terbesar dalam data. Standar: 350
`learning_rate`	Tingkat pembelajaran awal. Opsional Nilai yang valid: mengambang di (0, 1] Default: 0.001
`lr_scheduler_factor`	Rasio untuk mengurangi tingkat pembelajaran. Digunakan bersama dengan`lr_scheduler_step` parameter yang didefinisikan sebagai`lr_new = lr_old * lr_schedu ler_factor` . Opsional Nilai yang valid: mengambang di (0, 1) Default: 0,1

Nama Parameter	Deskripsi
`lr_scheduler_step`	Epochs di mana untuk mengurangi tingkat pembelajaran. Tingkat pembelajaran dikurangi`lr_scheduler_factor` pada zaman yang tercantum dalam string yang dibatasi koma: “epoch1, epoch2,...”. Misalnya, jika nilai diatur ke “10, 20" dan`lr_scheduler_factor` diatur ke 1/2, maka tingkat pembelajaran dibelah dua setelah zaman ke-10 dan kemudian dibelah dua lagi setelah zaman ke-20. Opsional Nilai yang valid: String Default: String
`mini_batch_size`	Ukuran batch untuk pelatihan. Dalam pengaturan multi-gpu mesin tunggal, setiap GPU`mini_batch_size` menangani /sampel`num_gpu` pelatihan. Untuk pelatihan multi-mes in dalam`dist_sync` mode, ukuran batch sebenarny a`mini_batch_size` adalah* jumlah mesin. Sebuah besar`mini_batch_size` biasanya mengarah ke pelatihan lebih cepat, tetapi dapat menyebabkan keluar dari masalah memori. Penggunaan memori terkait dengan`mini_batc h_size` ,`image_shape` , dan`base_network` arsitektu r. Misalnya, pada instans p3.2xlarge tunggal, kesalahan terbesar`mini_batch_size` tanpa memori keluar adalah 32 dengan base_network diatur ke “resnet-50" dan 300.`image_sha pe` Dengan contoh yang sama, Anda dapat menggunakan 64 sebagai`mini_batch_size` dengan jaringan dasar`vgg-16` dan 300.`image_shape` Opsional Nilai yang valid: bilangan bulat positif Default: 32

Nama Parameter	Deskripsi
`momentum`	Momentum untuk`sgd`. Diabaikan untuk pengoptimal lainnya. Opsional Nilai yang valid: mengambang di (0, 1] Bawaan: 0,9
`nms_threshold`	Ambang batas penindasan non-maksimum. Opsional Nilai yang valid: mengambang di (0, 1] Default: 0,45
`optimizer`	Jenis optimizer. Untuk detail tentang nilai pengoptimal, lihat API MXNet. Opsional Nilai yang valid: ['sgd', 'adam', 'rmsprop', 'adadelta'] Bawaan: 'sgd'
`overlap_threshold`	Evaluasi tumpang tindih ambang batas. Opsional Nilai yang valid: mengambang di (0, 1] Default: 0,5

Nama Parameter	Deskripsi
`use_pretrained_model`	Menunjukkan apakah akan menggunakan model pra-terla tih untuk pelatihan. Jika diatur ke 1, maka model pra-terlatih dengan arsitektur yang sesuai dimuat dan digunakan untuk pelatihan. Jika tidak, jaringan dilatih dari awal. Opsional Nilai yang valid: 0 atau 1 Default: 1
`weight_decay`	Koefisien peluruhan berat untuk`sgd` dan`rmsprop`. Diabaikan untuk pengoptimal lainnya. Opsional Nilai yang valid: mengambang di (0, 1) Default: 0,0005

Tune Model Deteksi Objek

Penyetelan model otomatis, juga dikenal sebagai penyetelan hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada kumpulan data Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi selengkapnya tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik Dihitung oleh Algoritma Deteksi Objek

Algoritma deteksi objek melaporkan satu metrik selama pelatihan:`validation:mAP`. Saat menyetel model, pilih metrik ini sebagai metrik objektif.

Nama Metrik	Deskripsi	Arah Optimasi
`validation:mAP`	Mean Average Precision (mAP) dihitung pada set validasi.	Maksimalkan

Hyperparameter Deteksi

Tune model deteksi SageMaker objek Amazon dengan hyperparameter berikut. Hyperparameter yang memiliki dampak terbesar pada metrik objektif deteksi objek adalah:`mini_batch_size`,`learning_rate`, dan`optimizer`.

Nama Parameter	Tipe	Rentang yang Direkomendasikan
`learning_rate`	ContinuousParameterRange	MinValue: 1e-6, MaxValue: 0,5
`mini_batch_size`	IntegerParameterRanges	MinValue: 8, MaxValue: 64
`momentum`	ContinuousParameterRange	MinValue: 0.0, MaxValue: 0.999
`optimizer`	CategoricalParameterRanges	['sgd', 'adam', 'rmsprop', 'adadelta']
`weight_decay`	ContinuousParameterRange	MinValue: 0.0, MaxValue: 0.999

Format

Format Permintaan

Query model terlatih dengan menggunakan endpoint model. Endpoint mengambil format gambar.jpg dan .png dengan`image/jpeg` dan`image/png` jenis konten.

Format Respons

Responsnya adalah indeks kelas dengan skor kepercayaan dan koordinat kotak pembatas untuk semua objek dalam gambar yang dikodekan dalam format JSON. Berikut ini adalah contoh file .json:


```
{"prediction":[
  [4.0, 0.86419455409049988, 0.3088374733924866, 0.07030484080314636,
 0.7110607028007507, 0.9345266819000244],
  [0.0, 0.73376623392105103, 0.5714187026023865, 0.40427327156066895,
 0.827075183391571, 0.9712159633636475],
  [4.0, 0.32643985450267792, 0.3677481412887573, 0.034883320331573486,
 0.6318609714508057, 0.5967587828636169],
  [8.0, 0.22552496790885925, 0.6152569651603699, 0.5722782611846924, 0.882301390171051,
 0.8985623121261597],
  [3.0, 0.42260299175977707, 0.019305512309074402, 0.08386176824569702,
 0.39093565940856934, 0.9574796557426453]
]}
```


Setiap baris dalam file.json ini berisi array yang mewakili objek yang terdeteksi. Masing-masing array objek terdiri dari daftar enam angka. Angka pertama adalah label kelas yang diprediksi. Angka kedua adalah skor kepercayaan terkait untuk deteksi. Empat angka terakhir mewakili koordinat kotak pembatas [xmin, ymin, xmax, ymax]. Indeks sudut kotak pembatas keluaran ini dinormalisasi oleh ukuran gambar keseluruhan. Perhatikan bahwa pengkodean ini berbeda dari yang digunakan oleh format.json masukan. Misalnya, pada entri pertama hasil deteksi, 0.3088374733924866 adalah koordinat kiri (koordinat x sudut kiri atas) dari kotak pembatas sebagai rasio lebar gambar keseluruhan, 0,07030484080314636 adalah koordinat atas (koordinat y dari sudut kiri atas) dari kotak pembatas sebagai rasio tinggi gambar keseluruhan, 0.7110607028007507 adalah koordinat yang tepat (koordinat x sudut kanan bawah) dari kotak pembatas sebagai rasio lebar gambar keseluruhan, dan 0,9345266819000244 adalah koordinat bawah (koordinat y dari sudut kanan bawah) dari batas kotak sebagai rasio tinggi citra secara keseluruhan.

Untuk menghindari hasil deteksi yang tidak dapat diandalkan, Anda mungkin ingin menyaring hasil deteksi dengan skor kepercayaan rendah. Dalam notebook sampel deteksi objek, kami memberikan contoh skrip yang menggunakan ambang batas untuk menghapus deteksi kepercayaan rendah dan untuk merencanakan kotak pembatas pada gambar asli.

Untuk transformasi batch, responsnya dalam format JSON, di mana formatnya identik dengan format JSON yang dijelaskan di atas. Hasil deteksi setiap gambar direpresentasikan sebagai file JSON. Misalnya:

```
{"prediction": [[label_id, confidence_score, xmin, ymin, xmax, ymax], [label_id,
 confidence_score, xmin, ymin, xmax, ymax]]}
```

Untuk detail lebih lanjut tentang pelatihan dan inferensi, lihatNotebook.

KELUARAN: Format Respons JSON

menerima: aplikasi/json; anotasi = 1

```
{
   "image_size": [
      {
         "width": 500,
         "height": 400,
         "depth": 3
      }
   ],
   "annotations": [
      {
         "class_id": 0,
         "score": 0.943,
         "left": 111,
         "top": 134,
         "width": 61,
         "height": 128
      },
      {
         "class_id": 0,
         "score": 0.0013,
         "left": 161,
         "top": 250,
         "width": 79,
         "height": 143
      },
      {
         "class_id": 1,
         "score": 0.0133,
         "left": 101,
         "top": 185,
         "width": 42,
         "height": 130
      }
   ]
```

```
}
```

Deteksi Objek - TensorFlow

Deteksi SageMaker Objek Amazon - TensorFlow algoritma adalah algoritma pembelajaran yang diawasi yang mendukung pembelajaran transfer dengan banyak model terlatih dari TensorFlow Model Garden. Gunakan pembelajaran transfer untuk menyempurnakan salah satu model terlatih yang tersedia pada kumpulan data Anda sendiri, meskipun sejumlah besar data gambar tidak tersedia. Algoritma deteksi objek mengambil gambar sebagai input dan output daftar kotak pembatas. Kumpulan data pelatihan harus terdiri dari gambar di. `jpg`,`.jpeg`, atau `.png` format.

Topik

- Cara menggunakan Deteksi SageMaker Objek - TensorFlow algoritma
- Antarmuka input dan output untuk Deteksi Objek - TensorFlow algoritma
- Rekomendasi instans Amazon EC2 untuk Deteksi Objek - algoritma TensorFlow
- Deteksi Objek - TensorFlow contoh buku catatan
- Bagaimana Deteksi Objek - TensorFlow Bekerja
- TensorFlow Model
- Deteksi Objek - TensorFlow Hyperparameters
- Menyetel Deteksi Objek - TensorFlow model

Cara menggunakan Deteksi SageMaker Objek - TensorFlow algoritma

Anda dapat menggunakan Deteksi Objek - TensorFlow sebagai algoritma SageMaker bawaan Amazon. Bagian berikut menjelaskan cara menggunakan Object Detection - TensorFlow dengan SageMaker Python SDK. Untuk informasi tentang cara menggunakan Deteksi Objek - TensorFlow dari UI Amazon SageMaker Studio Classic, lihatSageMaker JumpStart.

Deteksi Objek - TensorFlow algoritma mendukung pembelajaran transfer menggunakan salah satu TensorFlow model pra-terlatih yang kompatibel. Untuk daftar semua model terlatih yang tersedia, lihatTensorFlow Model. Setiap model yang telah dilatih sebelumnya memiliki keunikan`model_id`. Contoh berikut menggunakan ResNet 50 (`model_id`:`tensorflow-od1-ssd-resnet50-v1-fpn-640x640-coco17-tpu-8`) untuk menyempurnakan dataset kustom. Model yang telah dilatih sebelumnya semuanya telah diunduh sebelumnya dari TensorFlow Hub dan disimpan dalam bucket Amazon S3 sehingga pekerjaan pelatihan dapat berjalan dalam isolasi jaringan. Gunakan artefak pelatihan model yang dibuat sebelumnya ini untuk membangun Estimator. SageMaker

Pertama, ambil URI image Docker, URI skrip pelatihan, dan URI model yang telah dilatih sebelumnya. Kemudian, ubah hyperparameters sesuai keinginan Anda. Anda dapat melihat kamus Python dari semua hyperparameters yang tersedia dan nilai defaultnya dengan. `hyperparameters.retrieve_default` Untuk informasi selengkapnya, lihat Deteksi Objek - TensorFlow Hyperparameters. Gunakan nilai-nilai ini untuk membangun SageMaker Estimator.

> **Note**
>
> Nilai hyperparameter default berbeda untuk model yang berbeda. Misalnya, untuk model yang lebih besar, jumlah epoch default lebih kecil.

Contoh ini menggunakan PennFudanPeddataset, yang berisi gambar pejalan kaki di jalan. Kami mengunduh dataset sebelumnya dan membuatnya tersedia dengan Amazon S3. Untuk menyempurnakan model Anda, hubungi `.fit` menggunakan lokasi Amazon S3 dari kumpulan data pelatihan Anda.

```
from sagemaker import image_uris, model_uris, script_uris, hyperparameters
from sagemaker.estimator import Estimator

model_id, model_version = "tensorflow-od1-ssd-resnet50-v1-fpn-640x640-coco17-tpu-8",
 "*"
training_instance_type = "ml.p3.2xlarge"

# Retrieve the Docker image
train_image_uri =
 image_uris.retrieve(model_id=model_id,model_version=model_version,image_scope="training",insta

# Retrieve the training script
train_source_uri = script_uris.retrieve(model_id=model_id, model_version=model_version,
 script_scope="training")

# Retrieve the pretrained model tarball for transfer learning
train_model_uri = model_uris.retrieve(model_id=model_id, model_version=model_version,
 model_scope="training")

# Retrieve the default hyperparameters for fine-tuning the model
hyperparameters = hyperparameters.retrieve_default(model_id=model_id,
 model_version=model_version)

# [Optional] Override default hyperparameters with custom values
```

```
hyperparameters["epochs"] = "5"

# Sample training data is available in this bucket
training_data_bucket = f"jumpstart-cache-prod-{aws_region}"
training_data_prefix = "training-datasets/PennFudanPed_COCO_format/"

training_dataset_s3_path = f"s3://{training_data_bucket}/{training_data_prefix}"

output_bucket = sess.default_bucket()
output_prefix = "jumpstart-example-od-training"
s3_output_location = f"s3://{output_bucket}/{output_prefix}/output"

# Create an Estimator instance
tf_od_estimator = Estimator(
    role=aws_role,
    image_uri=train_image_uri,
    source_dir=train_source_uri,
    model_uri=train_model_uri,
    entry_point="transfer_learning.py",
    instance_count=1,
    instance_type=training_instance_type,
    max_run=360000,
    hyperparameters=hyperparameters,
    output_path=s3_output_location,
)

# Launch a training job
tf_od_estimator.fit({"training": training_dataset_s3_path}, logs=True)
```

Untuk informasi selengkapnya tentang cara menggunakan TensorFlow algoritma Deteksi SageMaker Objek - untuk pembelajaran transfer pada kumpulan data kustom, lihat buku catatan Pengantar SageMaker TensorFlow - Deteksi Objek.

Antarmuka input dan output untuk Deteksi Objek - TensorFlow algoritma

Masing-masing model terlatih yang tercantum dalam TensorFlow Model dapat disetel dengan baik ke kumpulan data apa pun dengan sejumlah kelas gambar. Perhatikan cara memformat data pelatihan Anda untuk masukan ke Deteksi Objek - TensorFlow model.

- Format input data pelatihan: Data pelatihan Anda harus berupa direktori dengan `images` subdirektori dan `annotations.json` file.

Berikut ini adalah contoh struktur direktori input. Direktori input harus di-host di bucket Amazon S3 dengan jalur yang mirip dengan berikut ini:. `s3://bucket_name/input_directory/` Perhatikan bahwa trailing / diperlukan.

```
input_directory
    |--images
        |--abc.png
        |--def.png
    |--annotations.json
```

`annotations.json`File harus berisi informasi untuk kotak pembatas dan label kelas mereka dalam bentuk kamus `"images"` dan `"annotations"` kunci. Nilai untuk `"images"` kunci harus berupa daftar kamus. Harus ada satu kamus untuk setiap gambar dengan informasi berikut:`{"file_name": image_name, "height": height, "width": width, "id": image_id}`. Nilai untuk `"annotations"` kunci juga harus berupa daftar kamus. Harus ada satu kamus untuk setiap kotak pembatas dengan informasi berikut:`{"image_id": image_id, "bbox": [xmin, ymin, xmax, ymax], "category_id": bbox_label}`.

Setelah pelatihan, file pemetaan label dan model terlatih disimpan ke bucket Amazon S3 Anda.

Pelatihan tambahan

Anda dapat menyemai pelatihan model baru dengan artefak dari model yang Anda latih sebelumnya. SageMaker Pelatihan tambahan menghemat waktu pelatihan ketika Anda ingin melatih model baru dengan data yang sama atau serupa.

> **Note**
>
> Anda hanya dapat menyemai Deteksi SageMaker Objek - TensorFlow model dengan Deteksi Objek lain - TensorFlow model yang dilatih SageMaker.

Anda dapat menggunakan kumpulan data apa pun untuk pelatihan tambahan, selama kumpulan kelas tetap sama. Langkah pelatihan inkremental mirip dengan langkah fine-tuning, tetapi alih-alih memulai dengan model yang telah dilatih sebelumnya, Anda mulai dengan model fine-tuning yang ada. Untuk informasi selengkapnya tentang cara menggunakan pelatihan tambahan dengan Deteksi SageMaker Objek - TensorFlow, lihat buku catatan Pengantar SageMaker TensorFlow - Deteksi Objek.

Inferensi dengan Deteksi Objek - algoritma TensorFlow

Anda dapat meng-host model yang disetel dengan baik yang dihasilkan dari pelatihan Deteksi TensorFlow Objek Anda untuk inferensi. Setiap gambar input untuk inferensi harus dalam `.jpg`,. `jpeg`, atau `.png` format dan menjadi tipe konten`application/x-image`. Deteksi Objek - TensorFlow algoritma mengubah ukuran gambar input secara otomatis.

Menjalankan inferensi menghasilkan kotak pembatas, kelas yang diprediksi, dan skor setiap prediksi yang dikodekan dalam format JSON. Deteksi Objek - TensorFlow model memproses satu gambar per permintaan dan hanya menghasilkan satu baris. Berikut ini adalah contoh respons format JSON:

```
accept: application/json;verbose

{"normalized_boxes":[[xmin1, xmax1, ymin1, ymax1],....],
    "classes":[classidx1, class_idx2,...],
    "scores":[score_1, score_2,...],
    "labels": [label1, label2, ...],
    "tensorflow_model_output":<original output of the model>}
```

Jika `accept` diatur ke`application/json`, maka model hanya mengeluarkan kotak, kelas, dan skor yang dinormalisasi.

Rekomendasi instans Amazon EC2 untuk Deteksi Objek - algoritma TensorFlow

Deteksi Objek - TensorFlow algoritma mendukung semua instans GPU untuk pelatihan, termasuk:

- `ml.p2.xlarge`
- `ml.p2.16xlarge`
- `ml.p3.2xlarge`
- `ml.p3.16xlarge`

Kami merekomendasikan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Instance CPU (seperti M5) dan GPU (P2 atau P3) dapat digunakan untuk inferensi. Untuk daftar lengkap instans SageMaker pelatihan dan inferensi di seluruh AWS Wilayah, lihat Harga Amazon SageMaker .

Deteksi Objek - TensorFlow contoh buku catatan

Untuk informasi selengkapnya tentang cara menggunakan TensorFlow algoritma Deteksi SageMaker Objek - untuk pembelajaran transfer pada kumpulan data kustom, lihat buku catatan Pengantar SageMaker TensorFlow - Deteksi Objek.

Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMakerContoh untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Use dan pilih Create copy.

Bagaimana Deteksi Objek - TensorFlow Bekerja

Deteksi Objek - TensorFlow algoritma mengambil gambar sebagai input dan memprediksi kotak pembatas dan label objek. Berbagai jaringan pembelajaran mendalam seperti MobileNet, ResNet, Inception, dan EfficientNet sangat akurat untuk deteksi objek. Ada juga jaringan pembelajaran mendalam yang dilatih pada kumpulan data gambar besar, seperti Common Objects in Context (COCO), yang memiliki 328.000 gambar. Setelah jaringan dilatih dengan data COCO, Anda kemudian dapat menyempurnakan jaringan pada kumpulan data dengan fokus tertentu untuk melakukan tugas deteksi objek yang lebih spesifik. Deteksi SageMaker Objek Amazon - TensorFlow algoritma mendukung pembelajaran transfer pada banyak model yang telah dilatih sebelumnya yang tersedia di TensorFlow Model Garden.

Menurut jumlah label kelas dalam data pelatihan Anda, lapisan deteksi objek dilampirkan ke TensorFlow model pilihan Anda yang telah dilatih sebelumnya. Anda kemudian dapat menyempurnakan seluruh jaringan (termasuk model yang telah dilatih sebelumnya) atau hanya lapisan klasifikasi teratas pada data pelatihan baru. Dengan metode pembelajaran transfer ini, pelatihan dengan kumpulan data yang lebih kecil dimungkinkan.

TensorFlow Model

Model terlatih berikut tersedia untuk digunakan untuk pembelajaran transfer dengan TensorFlow algoritma Deteksi Objek.

Model berikut bervariasi secara signifikan dalam ukuran, jumlah parameter model, waktu pelatihan, dan latensi inferensi untuk setiap kumpulan data tertentu. Model terbaik untuk kasus penggunaan Anda bergantung pada kompleksitas kumpulan data fine-tuning Anda dan persyaratan apa pun yang Anda miliki pada waktu pelatihan, latensi inferensi, atau akurasi model.

Nama Model	**model_id**	Sumber
ResNet50 V1 FPN 640	`tensorflow-od1-ssd-resnet50-v1-fpn-640x640-coco17-tpu-8`	TensorFlow Tautan Model Taman
EfficientDet D0 512	`tensorflow-od1-ssd-efficientdet-d0-512x512-coco17-tpu-8`	TensorFlow Tautan Model Taman
EfficientDet D1 640	`tensorflow-od1-ssd-efficientdet-d1-640x640-coco17-tpu-8`	TensorFlow Tautan Model Taman
EfficientDet D2 768	`tensorflow-od1-ssd-efficientdet-d2-768x768-coco17-tpu-8`	TensorFlow Tautan Model Taman
EfficientDet D3 896	`tensorflow-od1-ssd-efficientdet-d3-896x896-coco17-tpu-32`	TensorFlow Tautan Model Taman
MobileNet V1 FPN 640	`tensorflow-od1-ssd-mobilenet-v1-fpn-640x640-coco17-tpu-8`	TensorFlow Tautan Model Taman
MobileNet V2 FPNLite 320	`tensorflow-od1-ssd-mobilenet-v2-fpnlite-320x320-coco17-tpu-8`	TensorFlow Tautan Model Taman
MobileNet V2 FPNLite 640	`tensorflow-od1-ssd-mobilenet-v2-fpnlite-640x640-coco17-tpu-8`	TensorFlow Tautan Model Taman

Nama Model	**model_id**	Sumber
ResNet50 V1 FPN 1024	`tensorflow-od1-ssd-resnet50-v1-fpn-1024x1024-coco17-tpu-8`	TensorFlow Tautan Model Taman
ResNet101 V1 FPN 640	`tensorflow-od1-ssd-resnet101-v1-fpn-640x640-coco17-tpu-8`	TensorFlow Tautan Model Taman
ResNet101 V1 FPN 1024	`tensorflow-od1-ssd-resnet101-v1-fpn-1024x1024-coco17-tpu-8`	TensorFlow Tautan Model Taman
ResNet152 V1 FPN 640	`tensorflow-od1-ssd-resnet152-v1-fpn-640x640-coco17-tpu-8`	TensorFlow Tautan Model Taman
ResNet152 V1 FPN 1024	`tensorflow-od1-ssd-resnet152-v1-fpn-1024x1024-coco17-tpu-8`	TensorFlow Tautan Model Taman

Deteksi Objek - TensorFlow Hyperparameters

Hyperparameters adalah parameter yang ditetapkan sebelum model pembelajaran mesin mulai belajar. Hyperparameter berikut didukung oleh TensorFlow algoritma Deteksi Objek SageMaker bawaan Amazon. Lihat Menyetel Deteksi Objek - TensorFlow model untuk informasi tentang tuning hyperparameter.

Nama Parameter	Deskripsi
`batch_size`	Ukuran batch untuk pelatihan.

Nama Parameter	Deskripsi
	Nilai yang valid: bilangan bulat positif. Nilai default:`3`.
`beta_1`	Beta1 untuk pengoptimal. `"adam"` Merupakan tingkat peluruhan eksponensial untuk perkiraan momen pertama. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.9`.
`beta_2`	Beta2 untuk pengoptimal. `"adam"` Merupakan tingkat peluruhan eksponensial untuk perkiraan momen kedua. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.999`.
`early_stopping`	Atur `"True"` untuk menggunakan logika penghentian awal selama pelatihan. Jika`"False"`, berhenti dini tidak digunakan. Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.
`early_stopping_min_delta`	Perubahan minimum yang diperlukan untuk memenuhi syarat sebagai perbaikan. Perubahan absolut kurang dari nilai `early_stopping_min_delta` tidak memenuhi syarat sebagai perbaikan. Digunakan hanya ketika `early_stopping` diatur ke`"True"`. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.0`.

Nama Parameter	Deskripsi
`early_stopping_patience`	Jumlah zaman untuk melanjutkan pelatihan tanpa perbaikan. Digunakan hanya ketika `early_stopping` diatur ke`"True"`. Nilai yang valid: bilangan bulat positif. Nilai default:`5`.
`epochs`	Jumlah zaman pelatihan. Nilai yang valid: bilangan bulat positif. Nilai default: 5 untuk model yang lebih kecil, 1 untuk model yang lebih besar.
`epsilon`	Epsilon untuk`"adam"`,, `"rmsprop"` `"adadelta"` , dan `"adagrad"` pengoptimal. Biasanya diatur ke nilai kecil untuk menghindari pembagian dengan 0. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`1e-7`.
`initial_accumulator_value`	Nilai awal untuk akumulator, atau nilai momentum per parameter , untuk pengoptimal. `"adagrad"` Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.1`.
`learning_rate`	Tingkat pembelajaran pengoptimal. Nilai yang valid: float, range: [`0.0`,`1.0`]. Nilai default:`0.001`.

Nama Parameter	Deskripsi
momentum	Momentum untuk "sgd" dan "nesterov" pengoptimal. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [0.0,1.0]. Nilai default:0.9.
optimizer	Jenis pengoptimal. Untuk informasi selengkapnya, lihat Pengoptimal dalam dokumentasi. TensorFlow Nilai yang valid: string, salah satu dari berikut ini: ("adam""sgd",,"nesterov" ,"rmsprop" , "adagrad" ,"adadelta"). Nilai default:"adam".
reinitialize_top_layer	Jika disetel ke"Auto", parameter lapisan klasifikasi atas diinisialisasi ulang selama fine-tuning. Untuk pelatihan tambahan, parameter lapisan klasifikasi teratas tidak diinisialisasi ulang kecuali disetel ke. "True" Nilai yang valid: string, salah satu dari berikut ini: ("Auto", "True" atau"False"). Nilai default:"Auto".
rho	Faktor diskon untuk gradien "adadelta" dan "rmsprop" pengoptimal. Diabaikan untuk pengoptimal lainnya. Nilai yang valid: float, range: [0.0,1.0]. Nilai default:0.95.

Nama Parameter	Deskripsi
`train_only_on_top_layer`	Jika`"True"`, hanya parameter lapisan klasifikasi teratas yang disetel dengan baik. Jika`"False"`, semua parameter model disetel dengan baik. Nilai yang valid: string, baik: (`"True"`atau`"False"`). Nilai default:`"False"`.

Menyetel Deteksi Objek - TensorFlow model

Penyetelan model otomatis, juga dikenal sebagai tuning hyperparameter, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hiperparameter pada kumpulan data Anda. Anda memilih hyperparameters yang dapat disetel, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hiperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Untuk informasi lebih lanjut tentang penyetelan model, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Metrik dihitung oleh algoritma Deteksi Objek TensorFlow

Lihat bagan berikut untuk menemukan metrik mana yang dihitung oleh algoritma Deteksi Objek. TensorFlow

Nama Metrik	Deskripsi	Arah Optimasi	Pola Regex
`validation:localization_loss`	Kerugian lokalisasi untuk prediksi kotak.	Minimalkan	`Val_localization=([0-9\\.]+)`

Deteksi Objek yang Dapat Disetel - hyperparameters TensorFlow

Setel model deteksi objek dengan hyperparameter berikut. Hiperparameter yang memiliki dampak terbesar pada metrik tujuan deteksi objek adalah:`batch_size`,`learning_rate`, dan. `optimizer`

Setel hiperparameter terkait pengoptimal, seperti,,`momentum`,, `regularizers_l2` `beta_1``beta_2`, dan `eps` berdasarkan yang dipilih. `optimizer` Misalnya, gunakan `beta_1` dan `beta_2` `adam` hanya kapan`optimizer`.

Untuk informasi lebih lanjut tentang hiperparameter mana yang digunakan untuk masing-masing`optimizer`, lihatDeteksi Objek - TensorFlow Hyperparameters.

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`batch_size`	IntegerParameterRanges	MinValue: 8, MaxValue: 512
`beta_1`	ContinuousParameterRanges	MinValue: 1e-6,: 0.999 MaxValue
`beta_2`	ContinuousParameterRanges	MinValue: 1e-6,: 0.999 MaxValue
`eps`	ContinuousParameterRanges	MinValue: 1e-8, MaxValue: 1.0
`learning_rate`	ContinuousParameterRanges	MinValue: 1e-6, MaxValue: 0,5
`momentum`	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.999
`optimizer`	CategoricalParameterRanges	['sgd', 'adam', 'rmsprop', 'nesterov', 'adagrad', 'adadelta']
`regularizers_l2`	ContinuousParameterRanges	MinValue: 0.0, MaxValue: 0.999
`train_onl y_on_top_layer`	CategoricalParameterRanges	['Benar', 'Salah']

Nama Parameter	Jenis Parameter	Rentang yang Direkomendasikan
`initial_accumulator_value`	CategoricalParameterRanges	MinValue: 0.0, MaxValue: 0.999

Algoritme segmentasi semantik

Klaster SageMaker algoritma segmentasi semantik memberikan pendekatan tingkat piksel halus untuk mengembangkan aplikasi visi komputer. Ini tag setiap pixel dalam gambar dengan label kelas dari set yang telah ditetapkan kelas. Penandaan sangat penting untuk memahami adegan, yang sangat penting untuk peningkatan jumlah aplikasi penglihatan komputer, seperti kendaraan self-driving, diagnostik pencitraan medis, dan penginderaan robot.

Sebagai perbandingan, SageMaker Klasifikasi Gambar - MXNetadalah algoritma pembelajaran yang diawasi yang hanya menganalisis seluruh gambar, mengklasifikasikannya ke dalam salah satu dari beberapa kategori keluaran. KlasterDeteksiadalah algoritma pembelajaran yang diawasi yang mendeteksi dan mengklasifikasikan semua contoh objek dalam gambar. Ini menunjukkan lokasi dan skala setiap objek dalam gambar dengan kotak pembatas persegi panjang.

Karena algoritma segmentasi semantik mengklasifikasikan setiap piksel dalam gambar, ia juga memberikan informasi tentang bentuk objek yang terkandung dalam gambar. Output segmentasi direpresentasikan sebagai gambar grayscale, yang disebutmasker segmentasi. Masker segmentasi adalah gambar grayscale dengan bentuk yang sama dengan gambar input.

Klaster SageMaker algoritma segmentasi semantik dibangun menggunakanKerangka kerja MXNet Gluon dan toolkit Gluon CV. Ini memberi Anda pilihan tiga algoritma bawaan untuk melatih jaringan saraf dalam. Anda dapat menggunakanAlgoritma Fully-Convolutional Network (FCN),Algoritma Pyramid Scene Parsing (PSP), atauDeepLabV3.

Masing-masing dari tiga algoritma memiliki dua komponen yang berbeda:

- Klastertulang punggung(atauencoder) —Jaringan yang menghasilkan peta aktivasi fitur yang andal.
- Klasterdekoder—Jaringan yang membangun topeng segmentasi dari peta aktivasi yang dikodekan.

Anda juga memiliki pilihan tulang punggung untuk FCN, PSP, dan DeepLabAlgoritme V3: ResNet50 atau ResNet101. Tulang punggung ini termasuk artefak terlatih yang awalnya dilatih padaImageNettugas klasifikasi. Anda dapat menyempurnakan tulang punggung ini untuk segmentasi menggunakan data Anda sendiri. Atau, Anda dapat menginisialisasi dan melatih jaringan ini dari awal hanya menggunakan data Anda sendiri. Dekoder tidak pernah dilatih sebelumnya.

Untuk menerapkan model terlatih untuk inferensi, gunakan SageMaker layanan hosting. Selama inferensi, Anda dapat meminta masker segmentasi baik sebagai gambar PNG atau sebagai serangkaian probabilitas untuk setiap kelas untuk setiap piksel. Anda dapat menggunakan masker ini sebagai bagian dari pipeline yang lebih besar yang mencakup pemrosesan gambar hilir tambahan atau aplikasi lain.

Topik

- Notebook Sampel Segmentasi Semantik
- Antarmuka Input/Output untuk Algoritma Segmentasi Semantik
- Rekomendasi Instans EC2 untuk Algoritma Segmentasi Semantik
- Hyperparameters segmentasi semantik
- Menyetel Model Segmentasi Semantik

Notebook Sampel Segmentasi Semantik

Untuk contoh notebook Jupyter yang menggunakan SageMaker algoritma segmentasi semantik untuk melatih model dan menerapkannya untuk melakukan kesimpulan, lihatContoh Segmentasi semantik. Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, lihatInstans SageMaker Notebook Amazon.

Untuk melihat daftar semua SageMaker sampel, membuat dan membuka contoh notebook, dan memilihSageMaker ContohTab. Contoh notebook segmentasi semantik terletak di bawahPengantar algoritme Amazon. Untuk membuka notebook, pilihGunakantab, dan pilihBuat salinan.

Antarmuka Input/Output untuk Algoritma Segmentasi Semantik

SageMaker segmentasi semantik mengharapkan dataset pelatihan pelanggan untuk aktifAmazon Simple Storage Service (Amazon S3). Setelah dilatih, ia menghasilkan artefak model yang dihasilkan di Amazon S3. Format antarmuka input untuk SageMaker segmentasi semantik mirip dengan yang paling standar semantik segmentasi benchmarking dataset. Dataset di Amazon S3 diharapkan akan disajikan dalam dua saluran, satu untuk`train`dan satu untuk`validation`menggunakan empat

direktori, dua untuk gambar dan dua untuk anotasi. Anotasi diharapkan menjadi gambar PNG yang tidak terkompresi. Dataset mungkin juga memiliki peta label yang menjelaskan bagaimana pemetaan anotasi dibuat. Jika tidak, algoritma menggunakan default. Ini juga mendukung format gambar manifes yang ditambah (`application/x-image`) untuk pelatihan dalam mode input Pipe langsung dari Amazon S3. Untuk inferensi, titik akhir menerima gambar dengan`image/jpeg`jenis konten.

Cara Kerja Pelatihan

Data pelatihan dibagi menjadi empat direktori:`train`,`train_annotation`,`validation`, dan`validation_annotation`. Ada saluran untuk masing-masing direktori ini. Dataset juga diharapkan memiliki satu`label_map.json`file per saluran untuk`train_annotation`dan`validation_annotation`masing-masing. Jika Anda tidak menyediakan file JSON ini, SageMaker menyediakan peta label set default.

Dataset yang menentukan file-file ini akan terlihat seperti contoh berikut:

```
s3://bucket_name
    |
    |- train
                |
                | - 0000.jpg
                | - coffee.jpg
    |- validation
                |
                | - 00a0.jpg
                | - bananna.jpg
    |- train_annotation
                |
                | - 0000.png
                | - coffee.png
    |- validation_annotation
                |
                | - 00a0.png
                | - bananna.png
    |- label_map
                | - train_label_map.json
                | - validation_label_map.json
```

Setiap gambar JPG di direktori kereta dan validasi memiliki gambar label PNG yang sesuai dengan nama yang sama di`train_annotation`dan`validation_annotation`direktori. Konvensi penamaan ini membantu algoritma untuk mengaitkan label dengan gambar yang sesuai selama

pelatihan. Klastertrain,train_annotation,validation, danvalidation_annotationsaluran wajib dilakukan. Anotasi adalah gambar PNG saluran tunggal. Format bekerja selama metadata (mode) dalam gambar membantu algoritma membaca gambar anotasi ke dalam satu saluran 8-bit unsigned integer. Untuk informasi lebih lanjut tentang dukungan mode kami, lihatDokumentasi Perpustakaan Gambar Python. Sebaiknya gunakan 8-bit pixel, true colorPmodus.

Gambar yang dikodekan adalah bilangan bulat 8-bit sederhana saat menggunakan mode. Untuk mendapatkan dari pemetaan ini ke peta label, algoritma menggunakan satu file pemetaan per saluran, yang disebutpeta label. Peta label digunakan untuk memetakan nilai-nilai dalam gambar dengan indeks label yang sebenarnya. Di peta label default, yang disediakan secara default jika Anda tidak menyediakannya, nilai piksel dalam matriks anotasi (gambar) langsung mengindeks label. Gambar-gambar ini dapat berupa file PNG grayscale atau file PNG 8-bit yang diindeks. File peta label untuk kasus default unscaled adalah sebagai berikut:

```
{
  "scale": "1"
}
```

Untuk memberikan beberapa kontras untuk dilihat, beberapa perangkat lunak anotasi skala gambar label dengan jumlah yang konstan. Untuk mendukung ini, SageMaker algoritma segmentasi semantik menyediakan opsi rescaling untuk menurunkan nilai nilai label yang sebenarnya. Ketika scaling down tidak mengkonversi nilai ke integer yang sesuai, algoritma default ke bilangan bulat terbesar kurang dari atau sama dengan nilai skala. Kode berikut menunjukkan cara mengatur nilai skala untuk menskalakan kembali nilai label:

```
{
  "scale": "3"
}
```

Contoh berikut menunjukkan cara kerjanya"scale"digunakan untuk rescaleencoded_labelnilai dari gambar anotasi masukan ketika mereka dipetakan kemapped_labelnilai yang akan digunakan dalam pelatihan. Nilai label dalam gambar anotasi input adalah 0, 3, 6, dengan skala 3, sehingga dipetakan ke 0, 1, 2 untuk pelatihan:

```
encoded_label = [0, 3, 6]
mapped_label = [0, 1, 2]
```

Dalam beberapa kasus, Anda mungkin perlu menentukan pemetaan warna tertentu untuk setiap kelas. Gunakan opsi peta dalam pemetaan label seperti yang ditunjukkan pada contoh berikut`label_map`berkas:

```
{
    "map": {
        "0": 5,
        "1": 0,
        "2": 2
    }
}
```

Pemetaan label untuk contoh ini adalah:

```
encoded_label = [0, 5, 2]
mapped_label = [1, 0, 2]
```

Dengan pemetaan label, Anda dapat menggunakan sistem anotasi dan perangkat lunak anotasi yang berbeda untuk mendapatkan data tanpa banyak preprocessing. Anda dapat memberikan satu peta label per saluran. File untuk peta label di`label_map`channel harus mengikuti konvensi penamaan untuk struktur empat direktori. Jika Anda tidak memberikan peta label, algoritme mengasumsikan skala 1 (default).

Pelatihan dengan Format Manifes Augmented

Format manifes yang ditambah memungkinkan Anda melakukan pelatihan dalam mode Pipe menggunakan file gambar tanpa perlu membuat file RecordiO. File manifes augmented berisi objek data dan harus diGaris JSONformat, seperti yang dijelaskan dalam`CreateTrainingJob`request. Setiap baris dalam manifes adalah entri yang berisi URI Amazon S3 untuk gambar dan URI untuk gambar anotasi.

Setiap objek JSON dalam file manifes harus berisi`source-ref`kunci. Kelas`source-ref`kunci harus berisi nilai URI Amazon S3 ke gambar. Label disediakan di bawah`AttributeNames`nilai parameter seperti yang ditentukan dalam`CreateTrainingJob`request. Hal ini juga dapat berisi metadata tambahan di bawah tag metadata, tetapi ini diabaikan oleh algoritma. Pada contoh di bawah ini,`AttributeNames`terkandung dalam daftar referensi gambar dan anotasi`["source-ref", "city-streets-ref"]`. Nama-nama ini harus memiliki`-ref`ditambahkan ke mereka. Saat menggunakan algoritma segmentasi semantik dengan Augmented Manifest,

nilaiRecordWrapperTypeparameter harus"RecordIO"dan nilai dariContentTypeparameter harusapplication/x-recordio.

```
{"source-ref": "S3 bucket location", "city-streets-ref": "S3 bucket location", "city-
streets-metadata": {"job-name": "label-city-streets", }}
```

Untuk informasi selengkapnya tentang file manifes yang diperbesar, lihatMenyediakan Dataset Metadata untuk Training Jobs dengan Augmented Manifest File.

Pelatihan tambahan

Anda juga dapat benih pelatihan model baru dengan model yang Anda dilatih sebelumnya menggunakan SageMaker. Pelatihan tambahan ini menghemat waktu pelatihan saat Anda ingin melatih model baru dengan data yang sama atau serupa. Saat ini, pelatihan tambahan hanya didukung untuk model yang dilatih dengan built-in SageMaker Segmentasi semantik.

Untuk menggunakan model pra-terlatih Anda sendiri, tentukanChannelNamesebagai “model” diInputDataConfiguntukCreateTrainingJobrequest. MengaturContentTypeuntuk saluran model untukapplication/x-sagemaker-model. Klasterbackbone,algorithm,crop_size, dannum_classesparameter input yang menentukan arsitektur jaringan harus ditentukan secara konsisten dalam hiperparameter input model baru dan model pra-terlatih yang Anda unggah ke saluran model. Untuk file model yang telah dilatih sebelumnya, Anda dapat menggunakan artefak terkompresi (.tar.gz) dari SageMaker output. Anda hanya dapat menggunakan format Gambar untuk input data. Untuk informasi lebih lanjut tentang pelatihan tambahan dan untuk petunjuk tentang cara menggunakannya, lihatMenggunakan Pelatihan Inkremental di Amazon SageMaker.

Menghasilkan inferensi

Untuk menanyakan model terlatih yang diterapkan ke titik akhir, Anda perlu memberikan gambar danAcceptTypeyang menunjukkan jenis output yang dibutuhkan. Endpoint mengambil gambar JPEG denganimage/jpegjenis konten. Jika Anda memintaAcceptTypedariimage/png, algoritma mengeluarkan file PNG dengan masker segmentasi dalam format yang sama dengan label itu sendiri. Jika Anda meminta jenis menerimaapplication/x-recordio-protobuf, algoritma mengembalikan probabilitas kelas dikodekan dalam format recordio-protobuf. Format terakhir mengeluarkan tensor 3D di mana dimensi ketiga berukuran sama dengan jumlah kelas. Komponen ini menunjukkan probabilitas setiap label kelas untuk setiap piksel.

Rekomendasi Instans EC2 untuk Algoritma Segmentasi Semantik

Klaster SageMaker algoritma segmentasi semantik hanya mendukung instans GPU untuk pelatihan, dan sebaiknya gunakan instans GPU dengan lebih banyak memori untuk pelatihan dengan ukuran batch besar. Algoritma dapat dilatih menggunakan instans P2, P3, G4dn, atau G5 dalam konfigurasi mesin tunggal.

Untuk inferensi, Anda dapat menggunakan instans CPU (seperti C5 dan M5) dan instans GPU (seperti P3 dan G4dn) atau keduanya. Untuk informasi tentang jenis instans yang menyediakan berbagai kombinasi CPU, GPU, memori, dan kapasitas jaringan untuk inferensi, lihatAmazon SageMaker Jenis Instans.

Hyperparameters segmentasi semantik

Tabel berikut ini mencantumkan hiperparameter yang didukung oleh Amazon SageMaker algoritma segmentasi semantik untuk arsitektur jaringan, input data, dan pelatihan. Anda menentukan Segmentasi Semantik untuk pelatihan di`AlgorithmName`dari`CreateTrainingJob`request.

Hyperparameters Arsitektur Jaringan

Nama Parameter	Deskripsi
`backbone`	Tulang punggung yang digunakan untuk komponen encoder algoritma. Opsional Nilai valid: `resnet-50` , `resnet-101` Nilai default: `resnet-50`
`use_pretrained_model`	Apakah model yang telah dilatih sebelumnya akan digunakan untuk tulang punggung. Opsional Nilai valid: `True`, `False` Nilai default: `True`
`algorithm`	Algoritma yang digunakan untuk segmentasi semantik. Opsional

Nama Parameter	Deskripsi
	Nilai yang valid: • `fcn`:Algoritma Fully-Convolutional Network (FCN) • `psp`:Algoritma Pyramid Scene Parsing (PSP) • `deeplab`:DeepLab Algoritme V3 Nilai default: `fcn`

Hyperparameter Data

Nama Parameter	Deskripsi
`num_classes`	Jumlah kelas untuk segmen. Wajib Nilai yang valid: 2 ≤ bilangan bulat positif ≤ 254
`num_training_samples`	Jumlah sampel dalam data pelatihan. Algoritma menggunakan nilai ini untuk mengatur penjadwal tingkat pembelajaran. Wajib Nilai yang valid: bilangan bulat positif
`base_size`	Mendefinisikan bagaimana gambar yang rescaled sebelum memotong. Gambar yang rescaled sedemikian rupa sehingga panjang ukuran panjang diatur ke`base_size` dikalikan dengan angka acak 0,5-2,0, dan ukuran pendek dihitung untuk mempertahankan rasio aspek. Opsional Nilai yang valid: bilangan bulat positif> 16 Nilai default: 520

Nama Parameter	Deskripsi
`crop_size`	Ukuran gambar untuk input selama pelatihan. Kami secara acak rescale gambar input berdasarkan`base_size` , dan kemudian mengambil tanaman persegi acak dengan panjang sisi sama dengan`crop_size` . Klaster`crop_size` akan secara otomatis dibulatkan ke kelipatan 8. Opsional Nilai yang valid: bilangan bulat positif> 16 Nilai default: 240

Hyperparameters pelatihan

Nama Parameter	Deskripsi
`early_stopping`	Apakah akan menggunakan logika berhenti awal selama pelatihan. Opsional Nilai valid: `True`, `False` Nilai default: `False`
`early_sto pping_min_epochs`	Jumlah minimum zaman yang harus dijalankan. Opsional Nilai yang valid: integer Nilai default: 5
`early_sto pping_patience`	Jumlah zaman yang memenuhi toleransi untuk kinerja yang lebih rendah sebelum algoritma memberlakukan pemberhentian awal. Opsional Nilai yang valid: integer

Nama Parameter	Deskripsi
	Nilai default: 4
`early_stopping_tolerance`	Jika peningkatan relatif dari skor pekerjaan pelatihan, mIOU, lebih kecil dari nilai ini, penghentian awal menganggap zaman tidak membaik. Ini hanya digunakan bila`early_stopping =True`. Opsional Nilai yang valid: 0 ≤ float ≤ 1 Nilai default: 0.0
`epochs`	Jumlah zaman yang dapat digunakan untuk melatih. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 10
`gamma1`	Faktor peluruhan untuk rata-rata bergerak dari gradien kuadrat untuk`rmsprop`. Digunakan hanya untuk`rmsprop`. Opsional Nilai yang valid: 0 ≤ float ≤ 1 Nilai default: 0,9
`gamma2`	Faktor momentum untuk`rmsprop`. Opsional Nilai yang valid: 0 ≤ float ≤ 1 Nilai default: 0,9

Nama Parameter	Deskripsi
`learning_rate`	Tingkat pembelajaran awal. Opsional Nilai yang valid: 0 <float ≤ 1 Nilai default: 0.001
`lr_scheduler`	Bentuk jadwal belajar yang mengontrol penurunannya dari waktu ke waktu. Opsional Nilai yang valid: • `step`: Peluruhan bertahap, di mana tingkat pembelajaran dikurangi (dikalikan) dengan`lr_scheduler_factor` setelah zaman ditentukan oleh`lr_scheduler_step` . • `poly`: Peluruhan halus menggunakan fungsi polinomial. • `cosine`: Peluruhan halus menggunakan fungsi kosinus. Nilai default: `poly`
`lr_schedu ler_factor`	Jika`lr_scheduler` diatur ke`step`, rasio yang digunakan untuk mengurangi (multipy)`learning_rate` setelah masing-masing zaman yang ditentukan oleh`lr_scheduler_step` . Jika tidak, diabaikan. Opsional Nilai yang valid: 0 ≤ float ≤ 1 Nilai default: 0,1

Nama Parameter	Deskripsi
`lr_scheduler_step`	Sebuah koma dibatasi daftar zaman setelah`learning_rate` dikurangi (dikalikan) dengan`lr_scheduler_factor` . Misalnya, jika nilai diatur ke`"10, 20"`, maka`learning-rate` dikurangi dengan`lr_scheduler_factor` setelah zaman ke-10 dan lagi oleh faktor ini setelah zaman ke-20. Diperlukan secara kondisionaljika`lr_scheduler` diatur ke`step`. Jika tidak, diabaikan. Nilai yang benar: String Nilai default: (Tidak ada default, karena nilainya diperlukan saat digunakan.)
`mini_batch_size`	Ukuran batch untuk pelatihan. Menggunakan besar`mini_batch_size` biasanya menghasilkan pelatihan yang lebih cepat, tetapi mungkin menyebabkan Anda kehabisan memori. Penggunaan memori dipengaruhi oleh nilai-nilai`mini_batch_size` dan`image_shape` parameter, dan arsitektur tulang punggung. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 16.
`momentum`	Momentum untuk`sgd`optimalisasi. Saat Anda menggunakan pengoptimal lain, algoritma segmentasi semantik mengabaikan parameter ini. Opsional Nilai yang valid: 0 <float ≤ 1 Nilai default: 0,9

Nama Parameter	Deskripsi
`optimizer`	Jenis optimizer. Untuk informasi selengkapnya tentang pengoptimal, pilih tautan yang sesuai: • `adam`:Estimasi momentum adaptif • `adagrad`:Penurunan gradien adaptif • `nag`:Gradien terakselerasi Nesterov • `rmsprop`:Akar berarti propagasi kuadrat • `sgd`:Keturunan gradien stokastik Opsional Nilai yang benar:`adam`,`adagrad`,`nag`,`rmsprop`,`sgd` Nilai default: `sgd`
`syncbn`	Jika set ke`True`, mean normalisasi batch dan varians dihitung atas semua sampel yang diproses di seluruh GPU. Opsional Nilai valid: `True`, `False` Nilai default: `False`

Nama Parameter	Deskripsi
`validation_mini_batch_size`	Ukuran batch untuk validasi. Besar`mini_batch_size` biasanya menghasilkan pelatihan yang lebih cepat, tetapi mungkin menyebabkan Anda kehabisan memori. Penggunaan memori dipengaruhi oleh nilai-nilai`mini_batch_size` dan`image_shape` parameter, dan arsitektur tulang punggung. • Untuk mencetak validasi pada seluruh gambar tanpa memotong gambar, atur parameter ini ke 1. Gunakan opsi ini jika Anda ingin mengukur kinerja pada keseluruhan gambar secara keseluruhan. **Note** Mengatur`validation_mini_batch_size` parameter ke 1 menyebabkan algoritma untuk membuat model jaringan baru untuk setiap gambar. Ini mungkin memperlambat validasi dan pelatihan. • Untuk memotong gambar dengan ukuran yang ditentukan dalam`crop_size` parameter, bahkan selama evaluasi, atur parameter ini menjadi nilai yang lebih besar dari 1. Opsional Nilai yang valid: bilangan bulat positif Nilai default: 16.
`weight_decay`	Koefisien peluruhan berat untuk`sgd`optimalisasi. Saat Anda menggunakan pengoptimal lain, algoritme mengabaikan parameter ini. Opsional Nilai yang valid: 0 <float <1 Nilai default: 0.0001

Menyetel Model Segmentasi Semantik

Penalaan model otomatis, juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan yang menguji berbagai hyperparameter pada dataset Anda. Anda memilih hyperparameter merdu, rentang nilai untuk masing-masing, dan metrik objektif. Anda memilih metrik objektif dari metrik yang dihitung algoritme. Penyetelan model otomatis mencari hyperparameter yang dipilih untuk menemukan kombinasi nilai yang menghasilkan model yang mengoptimalkan metrik objektif.

Metrik Dihitung oleh Algoritma Segmentasi Semantik

Algoritma segmentasi semantik melaporkan dua metrik validasi. Saat menyetel nilai hyperparameter, pilih salah satu metrik ini sebagai tujuannya.

Nama Metrik	Deskripsi	Arah Pengoptimalan
`validation:mIOU`	Area persimpangan segmentasi yang diprediksi dan kebenaran dasar dibagi dengan area persatuan di antara mereka untuk gambar dalam set validasi. Juga dikenal sebagai Indeks Jaccard.	Maksimalkan
`validation:pixel_accuracy`	Persentase piksel yang diklasifikasikan dengan benar dalam gambar dari set validasi.	Maksimalkan

Hiperparameter segmentasi semantik merdu

Anda dapat menyetel hyperparameter berikut untuk algoritma segmentasi semantik.

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan
`learning_rate`	ContinuousParameterRange	MinValue: 1e-4, MaxValue: 1e-1
`mini_batch_size`	IntegerParameterRanges	MinValue: 1, MaxValue: 128

Nama Parameter	Tipe Parameter	Rentang yang direkomendasikan
momentum	ContinuousParameterRange	MinValue: 0,9, MaxValue: 0,999
optimzer	CategoricalParameterRanges	['sgd', 'adam', 'adadelta']
weight_decay	ContinuousParameterRange	MinValue: 1e-5, MaxValue: 1e-3

Gunakan Pembelajaran Penguatan dengan Amazon SageMaker

Pembelajaran penguatan (RL) menggabungkan bidang-bidang seperti ilmu komputer, ilmu saraf, dan psikologi untuk menentukan bagaimana memetakan situasi ke tindakan untuk memaksimalkan sinyal hadiah numerik. Gagasan tentang sinyal hadiah di RL ini berasal dari penelitian ilmu saraf tentang bagaimana otak manusia membuat keputusan tentang tindakan mana yang memaksimalkan hadiah dan meminimalkan hukuman. Dalam kebanyakan situasi, manusia tidak diberi instruksi eksplisit tentang tindakan mana yang harus diambil, tetapi sebaliknya harus mempelajari tindakan mana yang menghasilkan imbalan paling langsung, dan bagaimana tindakan tersebut memengaruhi situasi dan konsekuensi masa depan.

Masalah RL diformalkan menggunakan proses keputusan Markov (MDPs) yang berasal dari teori sistem dinamis. MDP bertujuan untuk menangkap detail tingkat tinggi dari masalah nyata yang dihadapi agen pembelajaran selama beberapa periode waktu dalam upaya mencapai beberapa tujuan akhir. Agen pembelajaran harus dapat menentukan keadaan lingkungannya saat ini dan mengidentifikasi kemungkinan tindakan yang mempengaruhi keadaan agen pembelajaran saat ini. Selanjutnya, tujuan agen pembelajaran harus berkorelasi kuat dengan keadaan lingkungan. Solusi untuk masalah yang dirumuskan dengan cara ini dikenal sebagai metode pembelajaran penguatan.

Apa perbedaan antara paradigma pembelajaran penguatan, pengawasan, dan tanpa pengawasan?

Pembelajaran mesin dapat dibagi menjadi tiga paradigma pembelajaran yang berbeda: diawasi, tidak diawasi, dan penguatan.

Dalam pembelajaran yang diawasi, supervisor eksternal memberikan serangkaian pelatihan contoh berlabel. Setiap contoh berisi informasi tentang suatu situasi, termasuk dalam kategori, dan memiliki label yang mengidentifikasi kategori yang menjadi miliknya. Tujuan dari pembelajaran yang diawasi adalah untuk menggeneralisasi untuk memprediksi dengan benar dalam situasi yang tidak ada dalam data pelatihan.

Sebaliknya, RL menangani masalah interaktif, sehingga tidak mungkin untuk mengumpulkan semua contoh situasi yang mungkin dengan label yang benar yang mungkin dihadapi agen. Jenis pembelajaran ini paling menjanjikan ketika seorang agen dapat secara akurat belajar dari pengalamannya sendiri dan menyesuaikannya.

Dalam pembelajaran tanpa pengawasan, seorang agen belajar dengan mengungkap struktur dalam data yang tidak berlabel. Sementara agen RL mungkin mendapat manfaat dari mengungkap struktur berdasarkan pengalamannya, satu-satunya tujuan RL adalah untuk memaksimalkan sinyal hadiah.

Topik

Mengapa Pembelajaran Penguatan Penting?

RL sangat cocok untuk memecahkan masalah besar dan kompleks, seperti manajemen rantai pasokan, sistem HVAC, robotika industri, kecerdasan buatan game, sistem dialog, dan kendaraan otonom. Karena model RL belajar dengan proses terus menerus menerima penghargaan dan hukuman untuk setiap tindakan yang diambil oleh agen, adalah mungkin untuk melatih sistem untuk membuat keputusan di bawah ketidakpastian dan dalam lingkungan yang dinamis.

Proses Keputusan Markov (MDP)

RL didasarkan pada model yang disebut Markov Decision Processes (MDPs). MDP terdiri dari serangkaian langkah waktu. Setiap langkah waktu terdiri dari berikut:

Environment

Mendefinisikan ruang di mana model RL beroperasi. Ini bisa berupa lingkungan dunia nyata atau simulator. Misalnya, jika Anda melatih kendaraan otonom fisik di jalan fisik, itu akan menjadi lingkungan dunia nyata. Jika Anda melatih program komputer yang memodelkan kendaraan otonom yang mengemudi di jalan, itu akan menjadi simulator.

Status

Menentukan semua informasi tentang lingkungan dan langkah-langkah masa lalu yang relevan dengan masa depan. Misalnya, dalam model RL di mana robot dapat bergerak ke segala arah kapan saja, posisi robot pada langkah waktu saat ini adalah keadaan, karena jika kita tahu di mana robot itu berada, tidak perlu mengetahui langkah-langkah yang diambil untuk sampai ke sana.

Action

Apa yang dilakukan agen. Misalnya, robot mengambil langkah maju.

Hadiah

Angka yang mewakili nilai negara yang dihasilkan dari tindakan terakhir yang diambil agen. Misalnya, jika tujuannya adalah agar robot menemukan harta karun, hadiah untuk menemukan harta karun mungkin 5, dan hadiah karena tidak menemukan harta karun mungkin 0. Model RL mencoba menemukan strategi yang mengoptimalkan imbalan kumulatif dalam jangka panjang. Strategi ini disebutkebijakan.

Observasi

Informasi tentang keadaan lingkungan yang tersedia untuk agen di setiap langkah. Ini mungkin seluruh negara bagian, atau mungkin hanya bagian dari negara. Misalnya, agen dalam model bermain catur akan dapat mengamati seluruh keadaan papan pada langkah apa pun, tetapi robot dalam labirin mungkin hanya dapat mengamati sebagian kecil labirin yang saat ini ditempati.

Biasanya, pelatihan di RL terdiri dari banyakepisode. Episode terdiri dari semua langkah waktu dalam MDP dari keadaan awal hingga lingkungan mencapai status terminal.

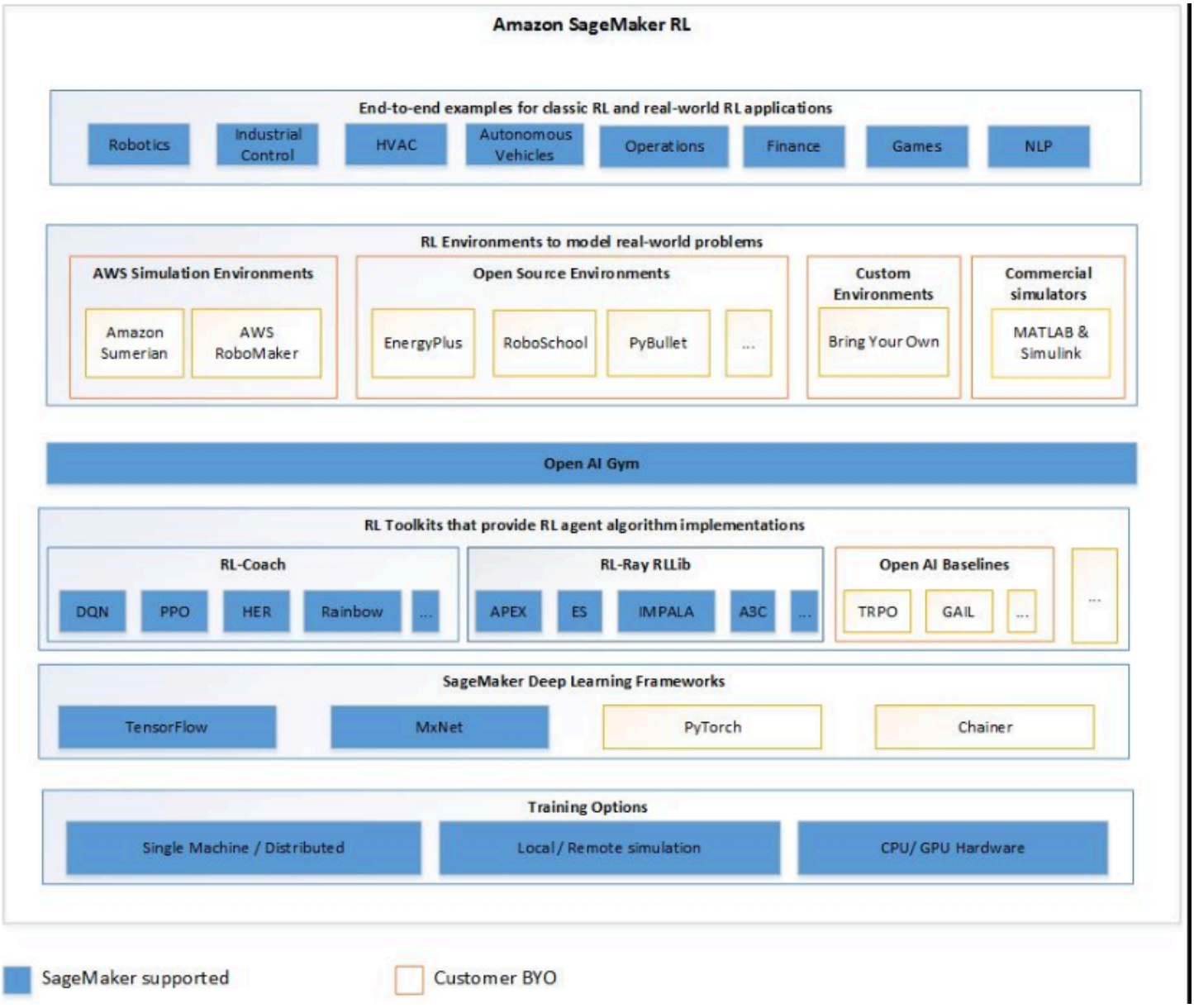
Fitur Utama dari Amazon SageMaker RL

Untuk melatih model RL di SageMaker RL, gunakan komponen-komponen berikut:

- Kerangka pembelajaran mendalam (DL). Saat ini, SageMaker mendukung RL di TensorFlow dan Apache MxNet.

- Toolkit RL. Toolkit RL mengelola interaksi antara agen dan lingkungan dan menyediakan berbagai pilihan algoritma RL canggih. SageMaker mendukung toolkit Intel Coach dan Ray RLLib. Untuk informasi tentang Intel Coach, lihathttps://nervanasystems.github.io/coach/. Untuk informasi tentang Ray RLLib, lihathttps://ray.readthedocs.io/en/latest/rllib.html.
- Lingkungan RL. Anda dapat menggunakan lingkungan khusus, lingkungan sumber terbuka, atau lingkungan komersial. Untuk informasi, lihat Lingkungan RL di Amazon SageMaker.

Diagram berikut menunjukkan komponen RL yang didukung di SageMaker RL.



Notebook Contoh Pembelajaran Penguatan

Untuk contoh kode lengkap, lihatbuku catatan sampel pembelajaran penguatandi SageMaker Contoh repositori.

Contoh Alur Kerja RL Menggunakan Amazon SageMaker RL

Contoh berikut menjelaskan langkah-langkah untuk mengembangkan model RL menggunakan Amazon SageMakerRL.

1. Merumuskan masalah RLPertama, merumuskan masalah bisnis menjadi masalah RL. Misalnya, penskalaan otomatis memungkinkan layanan meningkatkan atau mengurangi kapasitas secara dinamis tergantung pada kondisi yang Anda tentukan. Saat ini, ini memerlukan pengaturan alarm, kebijakan penskalaan, ambang batas, dan langkah manual lainnya. Untuk mengatasi ini dengan RL, kami mendefinisikan komponen Proses Keputusan Markov:

   a. Tujuan—Skalakan kapasitas instans sehingga cocok dengan profil beban yang diinginkan.
   b. Lingkungan—Lingkungan khusus yang mencakup profil beban. Ini menghasilkan beban simulasi dengan variasi harian dan mingguan dan lonjakan sesekali. Sistem simulasi memiliki penundaan antara ketika sumber daya baru diminta dan ketika mereka tersedia untuk melayani permintaan.
   c. negara bagian—Beban saat ini, jumlah pekerjaan yang gagal, dan jumlah mesin aktif.
   d. Aksi—Hapus, tambahkan, atau simpan jumlah instance yang sama.
   e. HadiahHadiah positif untuk transaksi yang berhasil dan penalti tinggi untuk transaksi yang gagal di luar ambang batas yang ditentukan.
2. Tentukan lingkungan RL—Lingkungan RL dapat menjadi dunia nyata di mana agen RL berinteraksi atau simulasi dunia nyata. Anda dapat menghubungkan open source dan lingkungan kustom yang dikembangkan menggunakan antarmuka Gym dan lingkungan simulasi komersial seperti MATLAB dan Simulink.
3. Tentukan preset—Preset mengkonfigurasi pekerjaan pelatihan RL dan menentukan hiperparameter untuk algoritma RL.
4. Tulis kode pelatihan—Tulis kode pelatihan sebagai script Python dan berikan script ke SageMaker pekerjaan pelatihan. Dalam kode pelatihan Anda, impor file lingkungan dan file preset, lalu tentukan`main()`fungsi.
5. Latih Model RL—Gunakan SageMaker `RLEstimator`diAmazon SageMaker Python SDKuntuk memulai pekerjaan pelatihan RL. Jika Anda menggunakan mode lokal, pekerjaan pelatihan berjalan pada instance notebook. Saat Anda menggunakan SageMaker untuk pelatihan, Anda

dapat memilih instance GPU atau CPU. Simpan output dari pekerjaan pelatihan di direktori lokal jika Anda berlatih dalam mode lokal, atau di Amazon S3 jika Anda menggunakan SageMaker pelatihan.

The`RLEstimator`membutuhkan informasi berikut sebagai parameter.

a. Direktori sumber tempat lingkungan, preset, dan kode pelatihan diunggah.
b. Jalan menuju skrip pelatihan.
c. Toolkit RL dan kerangka pembelajaran mendalam yang ingin Anda gunakan. Ini secara otomatis menyelesaikan jalur Amazon ECR untuk wadah RL.
d. Parameter pelatihan, seperti jumlah instance, nama pekerjaan, dan jalur S3 untuk output.
e. Definisi metrik yang ingin Anda tangkap di log Anda. Ini juga dapat divisualisasikan di CloudWatch dan di SageMaker buku catatan.

6. Visualisasikan metrik dan output pelatihan—Setelah pekerjaan pelatihan yang menggunakan model RL selesai, Anda dapat melihat metrik yang Anda tentukan dalam pekerjaan pelatihan di CloudWatch,. Anda juga dapat memplot metrik di buku catatan dengan menggunakanAmazon SageMaker Python SDKperpustakaan analitik. Memvisualisasikan metrik membantu Anda memahami bagaimana kinerja model yang diukur dengan hadiah meningkat dari waktu ke waktu.

> **Note**
>
> Jika Anda berlatih dalam mode lokal, Anda tidak dapat memvisualisasikan metrik CloudWatch.

7. Evaluasi model—Data yang diperiksa dari model yang dilatih sebelumnya dapat diteruskan untuk evaluasi dan inferensi di saluran pos pemeriksaan. Dalam mode lokal, gunakan direktori lokal. Di SageMakermode pelatihan, Anda perlu mengunggah data ke S3 terlebih dahulu.
8. Menyebarkan model RL—Akhirnya, terapkan model terlatih pada titik akhir yang dihosting SageMaker kontainer atau pada perangkat tepi dengan menggunakanAWS IoT Greengrass.

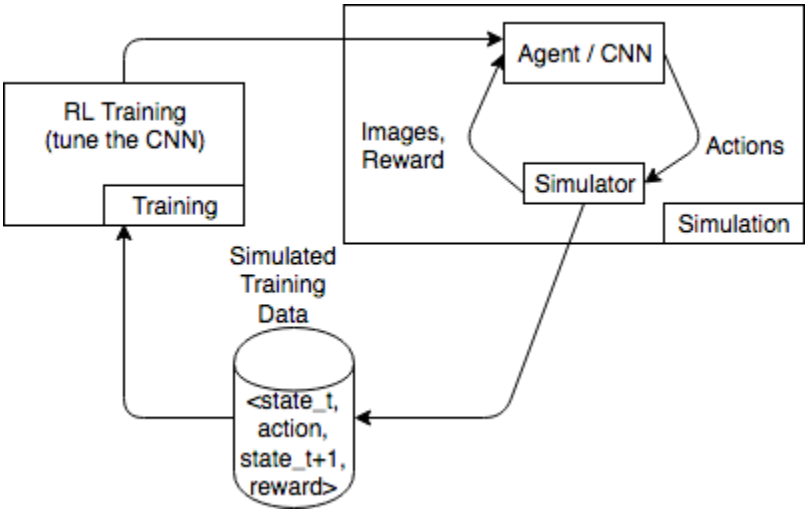
Untuk informasi lebih lanjut tentang RL dengan SageMaker, lihatMenggunakan RL dengan SageMaker Python SDK.

Lingkungan RL di Amazon SageMaker

Amazon SageMaker RL menggunakan lingkungan untuk meniru skenario dunia nyata. Mengingat keadaan lingkungan saat ini dan tindakan yang diambil oleh agen atau agen, simulator memproses

dampak tindakan, dan mengembalikan keadaan berikutnya dan hadiah. Simulator berguna dalam kasus di mana tidak aman untuk melatih agen di dunia nyata (misalnya, menerbangkan drone) atau jika algoritma RL membutuhkan waktu lama untuk bertemu (misalnya, saat bermain catur).

Diagram berikut menunjukkan contoh dari interaksi dengan simulator untuk game balap mobil.



Lingkungan simulasi terdiri dari agen dan simulator. Di sini, jaringan saraf konvolusional (CNN) mengkonsumsi gambar dari simulator dan menghasilkan tindakan untuk mengontrol pengontrol game. Dengan beberapa simulasi, lingkungan ini menghasilkan data pelatihan formulir`state_t`,`action`,`state_t+1`, dan`reward_t+1`. Mendefinisikan hadiah tidak sepele dan berdampak pada kualitas model RL. Kami ingin memberikan beberapa contoh fungsi hadiah, tetapi ingin membuatnya dapat dikonfigurasi pengguna.

Topik

- Gunakan Antarmuka OpenAI Gym untuk Lingkungan di SageMaker RL
- Gunakan Lingkungan Sumber Terbuka
- Gunakan Lingkungan Komersil

Gunakan Antarmuka OpenAI Gym untuk Lingkungan di SageMaker RL

Untuk menggunakan lingkungan OpenAI Gym di SageMaker RL, gunakan elemen API berikut. Untuk informasi lebih lanjut tentang OpenAI Gym, lihatDokumentasi Gym.

- `env.action_space`Mendefinisikan tindakan yang dapat diambil agen, menentukan apakah setiap tindakan kontinu atau diskrit, dan menentukan minimum dan maksimum jika tindakan itu kontinu.

- `env.observation_space`—Mendefinisikan pengamatan yang diterima agen dari lingkungan, serta minimum dan maksimum untuk pengamatan berkelanjutan.
- `env.reset()`—Menginisialisasi episode pelatihan. The`reset()`fungsi mengembalikan keadaan awal lingkungan, dan agen menggunakan keadaan awal untuk mengambil tindakan pertama. Tindakan tersebut kemudian dikirim ke`step()`berulang kali sampai episode mencapai status terminal. Kapan`step()`pulang`done = True`, episode berakhir. Toolkit RL menginisialisasi ulang lingkungan dengan memanggil`reset()`.
- `step()`—Mengambil tindakan agen sebagai masukan dan output keadaan lingkungan berikutnya, hadiah, apakah episode telah berakhir, dan`info`kamus untuk mengkomunikasikan informasi debugging. Merupakan tanggung jawab lingkungan untuk memvalidasi input.
- `env.render()`Digunakan untuk lingkungan yang memiliki visualisasi. Toolkit RL memanggil fungsi ini untuk menangkap visualisasi lingkungan setelah setiap panggilan ke`step()`fungsi.

Gunakan Lingkungan Sumber Terbuka

Anda dapat menggunakan lingkungan sumber terbuka, seperti EnergyPlus dan RoboSchool, di SageMakerRL dengan membangun wadah Anda sendiri. Untuk informasi lebih lanjut tentang EnergyPlus, lihathttps://energyplus.net/. Untuk informasi lebih lanjut tentang RoboSchool, lihathttps://github.com/openai/roboschool. HVAC dan RoboSchool contoh diSageMaker contoh repositoritunjukkan cara membuat wadah khusus untuk digunakan SageMaker RL:

Gunakan Lingkungan Komersil

Anda dapat menggunakan lingkungan komersial, seperti MATLAB dan Simulink, di SageMaker RL dengan membangun wadah Anda sendiri. Anda perlu mengelola lisensi Anda sendiri.

Pelatihan Terdistribusi dengan Amazon SageMaker RL

Amazon SageMaker RL mendukung pelatihan terdistribusi multi-core dan multi-instance. Tergantung pada kasus penggunaan Anda, pelatihan dan/atau peluncuran lingkungan dapat didistribusikan. Sebagai contoh, SageMaker RL berfungsi untuk skenario terdistribusi berikut:

- Instance training dan multiple rollout dari tipe yang sama. Sebagai contoh, lihat contoh Kompresi Jaringan Saraf diSageMaker contoh repositori.
- Instance pelatih tunggal dan beberapa instance peluncuran, di mana jenis instans yang berbeda untuk pelatihan dan peluncuran. Sebagai contoh, lihatAWS DeepRacer /AWS RoboMaker contoh diSageMaker contoh repositori.

- Instance pelatih tunggal yang menggunakan beberapa inti untuk peluncuran. Sebagai contoh, lihat contoh Roboschool diSageMaker contoh repositori. Ini berguna jika lingkungan simulasi ringan dan dapat berjalan pada satu utas.
- Beberapa contoh untuk pelatihan dan peluncuran. Sebagai contoh, lihat contoh Roboschool diSageMaker contoh repositori.

Tuning Hyperparameter dengan Amazon SageMaker RL

Anda dapat menjalankan pekerjaan tuning hyperparameter untuk mengoptimalkan hyperparameters untuk Amazon SageMaker RL. Contoh Roboschool dalam contoh buku catatan diSageMaker contoh repositorimenunjukkan bagaimana Anda dapat melakukan ini dengan RL Coach. Skrip peluncur menunjukkan bagaimana Anda dapat mengabstraksi parameter dari file preset Coach dan mengoptimalkannya.

Jalankan kode lokal Anda sebagai pekerjaan SageMaker pelatihan

Anda dapat menjalankan kode Python machine learning (ML) lokal Anda sebagai pekerjaan pelatihan SageMaker Amazon single-node yang besar atau sebagai beberapa pekerjaan paralel. Anda dapat melakukan ini dengan membuat anotasi kode Anda dengan dekorator @remote, seperti yang ditunjukkan pada contoh kode berikut. Pelatihan terdistribusi (di beberapa instance) tidak didukung dengan fungsi jarak jauh.

```
@remote(**settings)
def divide(x, y):
    return x / y
```

SDK SageMaker Python akan secara otomatis menerjemahkan lingkungan ruang kerja Anda yang ada dan kode pemrosesan data serta kumpulan data terkait ke dalam pekerjaan SageMaker pelatihan yang berjalan di platform pelatihan. SageMaker Anda juga dapat mengaktifkan fitur cache persisten, yang selanjutnya akan mengurangi latensi awal pekerjaan dengan menyimpan paket ketergantungan yang diunduh sebelumnya. Pengurangan latensi pekerjaan ini lebih besar daripada pengurangan latensi dari penggunaan kolam hangat yang SageMaker dikelola saja. Untuk informasi selengkapnya, lihat Menggunakan cache persisten.

> **Note**
>
> Pekerjaan pelatihan terdistribusi tidak didukung oleh fungsi jarak jauh.

Bagian berikut menunjukkan cara membuat anotasi kode HTML lokal Anda dengan dekorator @remote dan menyesuaikan pengalaman Anda untuk kasus penggunaan Anda. Ini termasuk menyesuaikan lingkungan Anda dan mengintegrasikan dengan SageMaker Eksperimen.

Topik

Siapkan lingkungan Anda

Pilih salah satu dari tiga opsi berikut untuk mengatur lingkungan Anda.

Jalankan kode Anda dari Amazon SageMaker Studio Classic

Anda dapat membuat anotasi dan menjalankan kode HTML lokal Anda dari SageMaker Studio Classic dengan membuat SageMaker Notebook dan melampirkan gambar apa pun yang tersedia pada gambar SageMaker Studio Classic. Petunjuk berikut membantu Anda membuat SageMaker Notebook, menginstal SageMaker Python SDK, dan membubuhi keterangan kode Anda dengan dekorator.

1. Buat SageMaker Notebook dan lampirkan gambar di SageMaker Studio Classic sebagai berikut:
   a. Ikuti petunjuk di Luncurkan Amazon SageMaker Studio Classic di Panduan SageMaker Pengembang Amazon.
   b. Pilih Studio dari panel navigasi kiri. Ini membuka jendela baru.
   c. Di kotak dialog Memulai, pilih profil pengguna dari panah bawah. Ini membuka jendela baru.
   d. Pilih Open Studio Classic.
   e. Pilih Buka Peluncur dari area kerja utama. Ini membuka halaman baru.

f. Pilih Buat buku catatan dari area kerja utama.

g. Pilih Base Python 3.0 dari panah bawah di sebelah Gambar di kotak dialog Ubah lingkungan.

Dekorator @remote secara otomatis mendeteksi gambar yang dilampirkan ke notebook SageMaker Studio Classic dan menggunakannya untuk menjalankan tugas SageMaker pelatihan. Jika `image_uri` ditentukan baik sebagai argumen di dekorator atau dalam file konfigurasi, maka nilai yang ditentukan dalam `image_uri` akan digunakan sebagai pengganti gambar yang terdeteksi.

Untuk informasi selengkapnya tentang cara membuat buku catatan di SageMaker Studio Classic, lihat bagian Membuat Buku Catatan dari Menu File di Membuat atau Membuka Notebook Amazon SageMaker Studio Classic.

Untuk daftar gambar yang tersedia, lihat Gambar Docker yang didukung.

2. Instal SDK SageMaker Python.

Untuk membuat anotasi kode Anda dengan fungsi @remote di dalam Notebook SageMaker Studio Classic, Anda harus menginstal SageMaker Python SDK. Instal SageMaker Python SDK, seperti yang ditunjukkan pada contoh kode berikut.

```
!pip install sagemaker
```

3. Gunakan dekorator @remote untuk menjalankan fungsi dalam pekerjaan SageMaker pelatihan.

Untuk menjalankan kode HTML lokal Anda, pertama-tama buat file dependensi untuk menginstruksikan SageMaker di mana menemukan kode lokal Anda. Untuk melakukannya, ikuti langkah-langkah berikut:

a. Dari area kerja utama SageMaker Studio Classic Launcher, di Utilitas dan file, pilih File teks. Ini membuka tab baru dengan file teks yang disebut `untitled.txt`.

Untuk informasi selengkapnya tentang antarmuka pengguna (UI) SageMaker Studio Classic, lihat Ikhtisar UI Amazon SageMaker Studio Classic.

b. Ganti nama `untitled.txt` ke `requirements.txt`.

c. Tambahkan semua dependensi yang diperlukan untuk kode bersama dengan SageMaker pustaka ke. `requirements.txt`

Contoh kode minimal `requirements.txt` untuk `divide` fungsi contoh disediakan di bagian berikut, sebagai berikut.

```
sagemaker
```

d. Jalankan kode Anda dengan dekorator jarak jauh dengan meneruskan file dependensi, sebagai berikut.

```
from sagemaker.remote_function import remote

@remote(instance_type="ml.m5.xlarge", dependencies='./requirements.txt')
def divide(x, y):
    return x / y

divide(2, 3.0)
```

Untuk contoh kode tambahan, lihat contoh notebook quick_start.ipynb.

Jika Anda sudah menjalankan notebook SageMaker Studio Classic, dan Anda menginstal Python SDK seperti yang diinstruksikan dalam 2. Instal SageMaker Python SDK, Anda harus me-restart kernel Anda. Untuk informasi selengkapnya, lihat Menggunakan Toolbar Notebook SageMaker Studio Classic di Panduan SageMaker Pengembang Amazon.

Jalankan kode Anda dari SageMaker notebook Amazon

Anda dapat membuat anotasi kode HTML lokal Anda dari instance SageMaker notebook. Petunjuk berikut menunjukkan cara membuat instance notebook dengan kernel kustom, menginstal SageMaker Python SDK, dan membubuhi keterangan kode Anda dengan dekorator.

1. Buat instance notebook dengan `conda` kernel kustom.

   Anda dapat membuat anotasi kode HTML lokal Anda dengan dekorator @remote untuk digunakan di dalam pekerjaan pelatihan. SageMaker Pertama, Anda harus membuat dan menyesuaikan instance SageMaker notebook untuk menggunakan kernel dengan Python versi 3.7 atau lebih tinggi, hingga 3.10.x. Untuk melakukannya, ikuti langkah-langkah berikut:

   a. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.

   b. Di panel navigasi kiri, pilih Notebook untuk memperluas opsinya.

   c. Pilih Instans Notebook dari opsi yang diperluas.

   d. Pilih tombol Create Notebook Instance. Ini membuka halaman baru.

e. Untuk nama instance Notebook, masukkan nama dengan maksimal 63 karakter dan tanpa spasi. Karakter yang valid: A-Z, a-z, 0-9, dan. : + = @ _% - (tanda hubung).

f. Di kotak dialog Pengaturan instance Notebook, perluas panah kanan di samping Konfigurasi Tambahan.

g. Di bawah Konfigurasi Siklus Hidup - opsional, perluas panah bawah dan pilih Buat konfigurasi siklus hidup baru. Ini membuka kotak dialog baru.

h. Di bawah Nama, masukkan nama untuk pengaturan konfigurasi Anda.

i. Di kotak dialog Skrip, di tab Mulai buku catatan, ganti isi kotak teks yang ada dengan skrip berikut.

```
#!/bin/bash

set -e

sudo -u ec2-user -i <<'EOF'
unset SUDO_UID
WORKING_DIR=/home/ec2-user/SageMaker/custom-miniconda/
source "$WORKING_DIR/miniconda/bin/activate"
for env in $WORKING_DIR/miniconda/envs/*; do
    BASENAME=$(basename "$env")
    source activate "$BASENAME"
    python -m ipykernel install --user --name "$BASENAME" --display-name "Custom
 ($BASENAME)"
done
EOF

echo "Restarting the Jupyter server.."
# restart command is dependent on current running Amazon Linux and JupyterLab
CURR_VERSION_AL=$(cat /etc/system-release)
CURR_VERSION_JS=$(jupyter --version)

if [[ $CURR_VERSION_JS == *$"jupyter_core     : 4.9.1"* ]] && [[ $CURR_VERSION_AL
 == *$" release 2018"* ]]; then
 sudo initctl restart jupyter-server --no-wait
else
 sudo systemctl --no-block restart jupyter-server.service
fi
```

j. Di kotak dialog Skrip, di tab Buat buku catatan, ganti isi kotak teks yang ada dengan skrip berikut.

```
#!/bin/bash

set -e

sudo -u ec2-user -i <<'EOF'
unset SUDO_UID
# Install a separate conda installation via Miniconda
WORKING_DIR=/home/ec2-user/SageMaker/custom-miniconda
mkdir -p "$WORKING_DIR"
wget https://repo.anaconda.com/miniconda/Miniconda3-4.6.14-Linux-x86_64.sh -O
 "$WORKING_DIR/miniconda.sh"
bash "$WORKING_DIR/miniconda.sh" -b -u -p "$WORKING_DIR/miniconda"
rm -rf "$WORKING_DIR/miniconda.sh"
# Create a custom conda environment
source "$WORKING_DIR/miniconda/bin/activate"
KERNEL_NAME="custom_python310"
PYTHON="3.10"
conda create --yes --name "$KERNEL_NAME" python="$PYTHON" pip
conda activate "$KERNEL_NAME"
pip install --quiet ipykernel
# Customize these lines as necessary to install the required packages
EOF
```

k. Pilih tombol Buat konfigurasi di kanan bawah jendela.

l. Pilih tombol Create notebook instance di kanan bawah jendela.

m. Tunggu Status instance notebook berubah dari Pending ke InService.

2. Buat notebook Jupyter di instance notebook.

   Petunjuk berikut menunjukkan cara membuat notebook Jupyter menggunakan Python 3.10 di instance yang baru Anda buat. SageMaker

   a. Setelah status instance notebook dari langkah sebelumnya adalah InService, lakukan hal berikut:

      i. Pilih Buka Jupyter di bawah Tindakan di baris yang berisi Nama instance notebook yang baru dibuat. Ini membuka server Jupyter baru.

   b. Di server Jupyter, pilih Baru dari menu kanan atas.

   c. Dari panah bawah, pilih conda_custom_python310. Ini menciptakan notebook Jupyter baru yang menggunakan kernel Python 3.10. Notebook Jupyter baru ini sekarang dapat digunakan mirip dengan notebook Jupyter lokal.

3. Instal SDK SageMaker Python.

   Setelah lingkungan virtual Anda berjalan, instal SDK SageMaker Python dengan menggunakan contoh kode berikut.

   ```
   !pip install sagemaker
   ```

4. Gunakan dekorator @remote untuk menjalankan fungsi dalam pekerjaan SageMaker pelatihan.

   Saat Anda membuat anotasi kode HTML lokal Anda dengan dekorator @remote di dalam SageMaker buku catatan, SageMaker pelatihan akan secara otomatis menafsirkan fungsi kode Anda dan menjalankannya sebagai pekerjaan pelatihan. SageMaker Siapkan buku catatan Anda dengan melakukan hal berikut:

   a. Pilih nama kernel di menu notebook dari instance SageMaker notebook yang Anda buat di langkah 1, Buat instance SageMaker Notebook dengan kernel kustom.

      Untuk informasi selengkapnya, lihat Mengubah Gambar atau Kernel.

   b. Dari panah bawah, pilih `conda` kernel kustom yang menggunakan versi Python yang 3.7 atau lebih tinggi.

      Sebagai contoh, memilih `conda_custom_python310` memilih kernel untuk Python 3.10.

   c. Pilih Pilih.

   d. Tunggu status kernel ditampilkan sebagai idle, yang menunjukkan bahwa kernel telah dimulai.

   e. Di Jupyter Server Home, pilih New dari menu kanan atas.

   f. Di sebelah panah bawah, pilih File teks. Ini membuat file teks baru yang disebut `untitled.txt`.

   g. Ganti nama `untitled.txt` menjadi `requirements.txt` dan tambahkan dependensi apa pun yang diperlukan untuk kode bersama. `sagemaker`

   h. Jalankan kode Anda dengan dekorator jarak jauh dengan meneruskan file dependensi seperti yang ditunjukkan di bawah ini.

      ```
      from sagemaker.remote_function import remote

      @remote(instance_type="ml.m5.xlarge", dependencies='./requirements.txt')
      def divide(x, y):
          return x / y
      ```

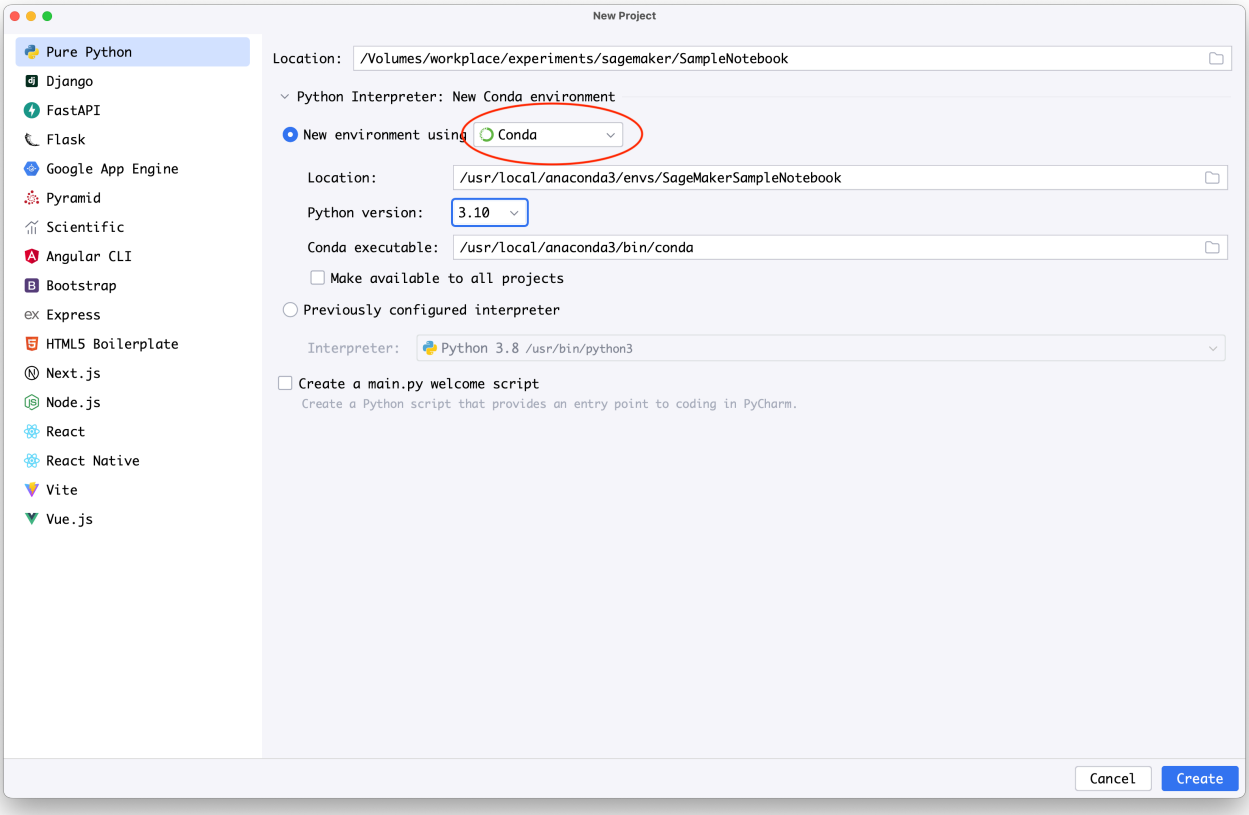
```
divide(2, 3.0)
```

Lihat contoh notebook quick_start.ipnyb untuk contoh kode tambahan.

Jalankan kode Anda dari dalam IDE lokal Anda

Anda dapat membuat anotasi kode HTML lokal Anda dengan dekorator @remote di dalam IDE lokal pilihan Anda. Langkah-langkah berikut menunjukkan prasyarat yang diperlukan, cara menginstal SDK Python, dan cara membuat anotasi kode Anda dengan dekorator @remote.

1. Instal prasyarat dengan menyiapkan AWS Command Line Interface (AWS CLI) dan membuat peran, sebagai berikut:
   - Onboard ke SageMaker domain mengikuti petunjuk di bagian AWS CLIPrasyarat Mengatur Prasyarat Amazon. SageMaker
   - Buat peran IAM mengikuti bagian Buat peran eksekusi SageMakerPeran.
2. Buat lingkungan virtual dengan menggunakan salah satu PyCharm atau `conda` dan menggunakan Python versi 3.7 atau lebih tinggi, hingga 3.10.x.
   - Siapkan lingkungan virtual menggunakan PyCharm sebagai berikut:
     a. Pilih File dari menu utama.
     b. Pilih Proyek Baru.
     c. Pilih Conda dari panah bawah di bawah Lingkungan baru menggunakan.
     d. Di bidang untuk versi Python gunakan panah bawah untuk memilih versi Python yang 3.7 atau lebih tinggi. Anda dapat naik ke 3.10.x dari daftar.



- Jika Anda telah menginstal Anaconda, Anda dapat mengatur lingkungan virtual menggunakan`conda`, sebagai berikut:
  - Buka antarmuka terminal prompt Anaconda.
  - Buat dan aktifkan `conda` lingkungan baru menggunakan versi Python 3.7 atau lebih tinggi, hingga 3.10x. Contoh kode berikut menunjukkan cara membuat `conda` lingkungan menggunakan Python versi 3.10.

```
conda create -n sagemaker_jobs_quick_start python=3.10 pip
conda activate sagemaker_jobs_quick_start
```

3. Instal SDK SageMaker Python.

   Untuk mengemas kode Anda dari IDE pilihan Anda, Anda harus memiliki lingkungan virtual yang diatur menggunakan Python 3.7 atau lebih tinggi, hingga 3.10x. Anda juga memerlukan gambar kontainer yang kompatibel. Instal SDK SageMaker Python menggunakan contoh kode berikut.

```
pip install sagemaker
```

4. Bungkus kode Anda di dalam dekorator @remote. SageMaker Python SDK akan secara otomatis menafsirkan fungsi kode Anda dan menjalankannya sebagai pekerjaan pelatihan. SageMaker Contoh kode berikut menunjukkan cara mengimpor pustaka yang diperlukan, menyiapkan SageMaker sesi, dan membuat anotasi fungsi dengan dekorator @remote.

   Anda dapat menjalankan kode Anda dengan menyediakan dependensi yang dibutuhkan secara langsung, atau dengan menggunakan dependensi dari lingkungan aktif. `conda`

   - Untuk memberikan dependensi secara langsung, lakukan hal berikut:
     - Buat `requirements.txt` file di direktori kerja tempat kode berada.
     - Tambahkan semua dependensi yang diperlukan untuk kode bersama dengan perpustakaan. SageMaker Bagian berikut memberikan contoh kode minimal `requirements.txt` untuk `divide` fungsi contoh.

       ```
       sagemaker
       ```

     - Jalankan kode Anda dengan dekorator @remote dengan meneruskan file dependensi. Dalam contoh kode berikut, ganti `The IAM role name` dengan ARN peran AWS Identity and Access Management (IAM) yang SageMaker ingin Anda gunakan untuk menjalankan pekerjaan Anda.

       ```
       import boto3
       import sagemaker
       from sagemaker.remote_function import remote

       sm_session =
        sagemaker.Session(boto_session=boto3.session.Session(region_name="us-west-2"))
       settings = dict(
           sagemaker_session=sm_session,
           role=<The IAM role name>,
           instance_type="ml.m5.xlarge",
           dependencies='./requirements.txt'
       )

       @remote(**settings)
       def divide(x, y):
           return x / y


       if __name__ == "__main__":
           print(divide(2, 3.0))
       ```

- Untuk menggunakan dependensi dari `conda` lingkungan aktif, gunakan nilai `auto_capture` untuk `dependencies` parameter, seperti yang ditunjukkan pada berikut ini.

```
import boto3
import sagemaker
from sagemaker.remote_function import remote

sm_session = sagemaker.Session(boto_session=boto3.session.Session(region_name="us-west-2"))
settings = dict(
    sagemaker_session=sm_session,
    role=<The IAM role name>,
    instance_type="ml.m5.xlarge",
    dependencies="auto_capture"
)

@remote(**settings)
def divide(x, y):
    return x / y


if __name__ == "__main__":
    print(divide(2, 3.0))
```

> **Note**
>
> Anda juga dapat menerapkan kode sebelumnya di dalam notebook Jupyter. PyCharm Edisi Profesional mendukung Jupyter secara asli. Untuk panduan selengkapnya, lihat Dukungan notebook Jupyter dalam dokumentasi PyCharm.

Memanggil fungsi

Untuk memanggil fungsi di dalam dekorator @remote, gunakan salah satu metode berikut:

- Gunakan dekorator @remote untuk menjalankan fungsi.
- Gunakan `RemoteExecutor` API untuk menjalankan fungsi.

Jika Anda menggunakan metode dekorator @remote untuk menjalankan fungsi, pekerjaan pelatihan akan menunggu fungsi selesai sebelum memulai tugas baru. Namun, jika Anda menggunakan `RemoteExecutor` API, Anda dapat menjalankan lebih dari satu pekerjaan secara paralel. Bagian berikut menunjukkan kedua cara menjalankan fungsi.

Gunakan dekorator @remote untuk menjalankan fungsi

Anda dapat menggunakan dekorator @remote untuk membubuhi keterangan suatu fungsi. SageMaker akan mengubah kode di dalam dekorator menjadi pekerjaan SageMaker pelatihan. Pekerjaan pelatihan kemudian akan memanggil fungsi di dalam dekorator dan menunggu pekerjaan selesai. Contoh kode berikut menunjukkan cara mengimpor pustaka yang diperlukan, memulai SageMaker instance, dan membubuhi keterangan perkalian matriks dengan dekorator @remote.

```
from sagemaker.remote_function import remote
import numpy as np

@remote(instance_type="ml.m5.large")
def matrix_multiply(a, b):
    return np.matmul(a, b)

a = np.array([[1, 0],
              [0, 1]])
b = np.array([1, 2])

assert (matrix_multiply(a, b) == np.array([1,2])).all()
```

Dekorator didefinisikan sebagai berikut.

```
def remote(
    *,
    **kwarg):
        ...
```

Saat Anda menjalankan fungsi yang didekorasi, SageMaker Python SDK memuat pengecualian apa pun yang dimunculkan oleh kesalahan ke memori lokal. Dalam contoh kode berikut, panggilan pertama ke fungsi membagi selesai dengan sukses dan hasilnya dimuat ke memori lokal. Dalam panggilan kedua ke fungsi membagi, kode mengembalikan kesalahan dan kesalahan ini dimuat ke memori lokal.

```
from sagemaker.remote_function import remote
import pytest
```

```
@remote()
def divide(a, b):
    return a/b

# the underlying job is completed successfully
# and the function return is loaded
assert divide(10, 5) == 2

# the underlying job fails with "AlgorithmError"
# and the function exception is loaded into local memory
with pytest.raises(ZeroDivisionError):
    divide(10, 0)
```

> **Note**
>
> Fungsi yang didekorasi dijalankan sebagai pekerjaan jarak jauh. Jika utas terputus, pekerjaan yang mendasarinya tidak akan dihentikan.

Cara mengubah nilai variabel lokal

Fungsi dekorator dijalankan pada mesin jarak jauh. Mengubah variabel non-lokal atau argumen masukan di dalam fungsi yang didekorasi tidak akan mengubah nilai lokal.

Dalam contoh kode berikut, daftar dan dict ditambahkan di dalam fungsi dekorator. Ini tidak berubah ketika fungsi dekorator dipanggil.

```
a = []

@remote
def func():
    a.append(1)

# when func is invoked, a in the local memory is not modified
func()
func()

# a stays as []

a = {}
@remote
```

```
def func(a):
    # append new values to the input dictionary
    a["key-2"] = "value-2"

a = {"key": "value"}
func(a)

# a stays as {"key": "value"}
```

Untuk mengubah nilai variabel lokal yang dideklarasikan di dalam fungsi dekorator, kembalikan variabel dari fungsi. Contoh kode berikut menunjukkan bahwa nilai variabel lokal berubah ketika dikembalikan dari fungsi.

```
a = {"key-1": "value-1"}

@remote
def func(a):
    a["key-2"] = "value-2"
    return a

a = func(a)

-> {"key-1": "value-1", "key-2": "value-2"}
```

Serialisasi data dan deserialisasi

Saat Anda menjalankan fungsi jarak jauh, SageMaker secara otomatis membuat serial argumen fungsi Anda selama tahap input dan output. Argumen dan pengembalian fungsi diserialisasikan menggunakan cloudpickle. SageMaker mendukung serialisasi objek dan fungsi Python berikut.

- Objek Python bawaan termasuk dict, list, float, int, string, nilai boolean, dan tupel
- Array Numpy
- Dataframes Panda
- Scikit-learn dataset dan estimator
- PyTorch model
- TensorFlow model
- Kelas Booster untuk XGBoost

Berikut ini dapat digunakan dengan beberapa batasan.

- Dask DataFrames
- Kelas XGBoost Dmatrix
- TensorFlow dataset dan subclass
- PyTorch model

Bagian berikut berisi praktik terbaik untuk menggunakan kelas Python sebelumnya dengan beberapa batasan dalam fungsi jarak jauh Anda, informasi tentang tempat SageMaker menyimpan data serial Anda dan cara mengelola akses ke sana.

Praktik terbaik untuk kelas Python dengan dukungan terbatas untuk serialisasi data jarak jauh

Anda dapat menggunakan kelas Python yang tercantum di bagian ini dengan batasan. Bagian selanjutnya membahas praktik terbaik untuk cara menggunakan kelas Python berikut.

- Dask DataFrames
- Kelas XGBoost DMatric
- TensorFlow dataset dan subclass
- PyTorch model

Praktik terbaik untuk Dask

Dask adalah perpustakaan sumber terbuka yang digunakan untuk komputasi paralel dengan Python. Bagian ini menunjukkan yang berikut ini.

- Cara meneruskan Dask DataFrame ke fungsi jarak jauh Anda
- Bagaimana mengubah statistik ringkasan dari Dask DataFrame menjadi Panda DataFrame

Cara meneruskan Dask DataFrame ke fungsi jarak jauh Anda

Dask sering DataFrames digunakan untuk memproses dataset besar karena mereka dapat menyimpan dataset yang membutuhkan lebih banyak memori daripada yang tersedia. Ini karena Dask DataFrame tidak memuat data lokal Anda ke dalam memori. Jika Anda meneruskan Dask DataFrame sebagai argumen fungsi ke fungsi jarak jauh Anda, Dask dapat meneruskan referensi ke data di disk lokal atau penyimpanan cloud Anda, bukan data itu sendiri. Kode berikut menunjukkan contoh melewati Dask DataFrame di dalam fungsi remote Anda yang akan beroperasi pada kosong DataFrame.

```
#Do not pass a Dask DataFrame  to your remote function as follows
def clean(df: dask.DataFrame ):
    cleaned = df[] \ ...
```

Dask akan memuat data dari Dask DataFrame ke dalam memori hanya ketika Anda menggunakan file. DataFrame Jika Anda ingin menggunakan Dask DataFrame di dalam fungsi jarak jauh, berikan jalur ke data. Kemudian Dask akan membaca dataset langsung dari jalur data yang Anda tentukan saat kode berjalan.

Contoh kode berikut menunjukkan cara menggunakan Dask DataFrame di dalam fungsi `clean` remote. Dalam contoh kode, `raw_data_path` diteruskan ke clean bukan Dask DataFrame. Saat kode berjalan, kumpulan data dibaca langsung dari lokasi bucket Amazon S3 yang ditentukan. `raw_data_path` Kemudian `persist` fungsi menyimpan kumpulan data dalam memori untuk memfasilitasi `random_split` fungsi selanjutnya dan ditulis kembali ke jalur data keluaran dalam bucket S3 menggunakan fungsi API Dask DataFrame .

```
import dask.dataframe as dd

@remote(
   instance_type='ml.m5.24xlarge',
   volume_size=300,
   keep_alive_period_in_seconds=600)
#pass the data path to your remote function rather than the Dask DataFrame  itself
def clean(raw_data_path: str, output_data_path: str: split_ratio: list[float]):
    df = dd.read_parquet(raw_data_path) #pass the path to your DataFrame
    cleaned = df[(df.column_a >= 1) & (df.column_a < 5)]\
        .drop(['column_b', 'column_c'], axis=1)\
        .persist() #keep the data in memory to facilitate the following random_split
 operation

    train_df, test_df = cleaned.random_split(split_ratio, random_state=10)

    train_df.to_parquet(os.path.join(output_data_path, 'train')
    test_df.to_parquet(os.path.join(output_data_path, 'test'))

clean("s3://my-bucket/raw/", "s3://my-bucket/cleaned/", split_ratio=[0.7, 0.3])
```

Bagaimana mengubah statistik ringkasan dari Dask DataFrame menjadi Panda DataFrame

Ringkasan statistik dari Dask DataFrame dapat dikonversi menjadi Pandas DataFrame dengan menggunakan `compute` metode seperti yang ditunjukkan dalam contoh kode berikut. Dalam contoh, bucket S3 berisi Dask besar DataFrame yang tidak dapat masuk ke dalam memori atau ke dalam kerangka data Pandas. Dalam contoh berikut, fungsi jarak jauh memindai kumpulan data dan mengembalikan Dask yang DataFrame berisi statistik keluaran dari `describe` ke Pandas. DataFrame

```
executor = RemoteExecutor(
    instance_type='ml.m5.24xlarge',
    volume_size=300,
    keep_alive_period_in_seconds=600)

future = executor.submit(lambda: dd.read_parquet("s3://my-bucket/
raw/").describe().compute())

future.result()
```

Praktik terbaik untuk kelas XGBoost DMatric

DMatrix adalah struktur data internal yang digunakan oleh XGBoost untuk memuat data. Objek DMatrix tidak dapat diasinkan untuk bergerak dengan mudah di antara sesi komputasi. Melewati instance DMatrix secara langsung akan gagal dengan a. `SerializationError`

Cara meneruskan objek data ke fungsi jarak jauh Anda dan berlatih dengan XGBoost

Untuk mengonversi Pandas DataFrame menjadi instance DMatrix dan menggunakannya untuk pelatihan dalam fungsi jarak jauh Anda, teruskan langsung ke fungsi jarak jauh seperti yang ditunjukkan pada contoh kode berikut.

```
import xgboost as xgb

@remote
def train(df, params):
    #Convert a pandas dataframe into a DMatrix DataFrame and use it for training
    dtrain = DMatrix(df)
    return xgb.train(dtrain, params)
```

Praktik terbaik untuk TensorFlow kumpulan data dan sub-kelas

TensorFlow dataset dan subclass adalah objek internal yang digunakan oleh TensorFlow untuk memuat data selama pelatihan. TensorFlow kumpulan data dan subkelas tidak dapat diasinkan untuk berpindah dengan mudah di antara sesi komputasi. Melewati kumpulan data atau subkelas Tensorflow secara langsung akan gagal dengan file. `SerializationError` Gunakan Tensorflow I/O API untuk memuat data dari penyimpanan, seperti yang ditunjukkan pada contoh kode berikut.

```
import tensorflow as tf
import tensorflow_io as tfio

@remote
def train(data_path: str, params):

    dataset = tf.data.TextLineDataset(tf.data.Dataset.list_files(f"{data_path}/*.txt"))
    ...

train("s3://my-bucket/data", {})
```

Praktik terbaik untuk PyTorch model

PyTorch model dapat diserialkan dan dapat diteruskan antara lingkungan lokal Anda dan fungsi jarak jauh. Jika lingkungan lokal dan lingkungan jarak jauh Anda memiliki jenis perangkat yang berbeda, seperti (GPU dan CPU), Anda tidak dapat mengembalikan model terlatih ke lingkungan lokal Anda. Misalnya, jika kode berikut dikembangkan di lingkungan lokal tanpa GPU tetapi dijalankan dalam instance dengan GPU, mengembalikan model terlatih secara langsung akan mengarah ke file. `DeserializationError`

```
# Do not return a model trained on GPUs to a CPU-only environment as follows

@remote(instance_type='ml.g4dn.xlarge')
def train(...):
    if torch.cuda.is_available():
        device = torch.device("cuda")
    else:
        device = torch.device("cpu") # a device without GPU capabilities

    model = Net().to(device)

    # train the model
    ...
```

```
    return model

model = train(...) #returns a DeserializationError if run on a device with GPU
```

Untuk mengembalikan model yang dilatih dalam lingkungan GPU ke model yang hanya berisi kemampuan CPU, gunakan PyTorch model I/O API secara langsung seperti yang ditunjukkan pada contoh kode di bawah ini.

```
import s3fs

model_path = "s3://my-bucket/folder/"

@remote(instance_type='ml.g4dn.xlarge')
def train(...):
    if torch.cuda.is_available():
        device = torch.device("cuda")
    else:
        device = torch.device("cpu")

    model = Net().to(device)

    # train the model
    ...

    fs = s3fs.FileSystem()
    with fs.open(os.path.join(model_path, 'model.pt'), 'wb') as file:
        torch.save(model.state_dict(), file) #this writes the model in a device-
agnostic way (CPU vs GPU)

train(...) #use the model to train on either CPUs or GPUs

model = Net()
fs = s3fs.FileSystem()with fs.open(os.path.join(model_path, 'model.pt'), 'rb') as file:
    model.load_state_dict(torch.load(file, map_location=torch.device('cpu')))
```

Tempat SageMaker menyimpan data serial Anda

Saat Anda memanggil fungsi jarak jauh, SageMaker secara otomatis membuat serial argumen fungsi Anda dan mengembalikan nilai selama tahap input dan output. Data serial ini disimpan di bawah direktori root di bucket S3 Anda. Anda menentukan direktori root,`<s3_root_uri>`, dalam file konfigurasi. Parameter `job_name` dibuat secara otomatis untuk Anda.

Di bawah direktori root, SageMaker buat <job_name> folder, yang menyimpan direktori kerja Anda saat ini, fungsi serial, argumen untuk fungsi serial Anda, hasil, dan pengecualian apa pun yang muncul dari menjalankan fungsi serial.

Di bawah<job_name>, direktori workdir berisi arsip zip dari direktori kerja Anda saat ini. Arsip zip mencakup file Python apa pun di direktori kerja Anda dan requirements.txt file, yang menentukan dependensi apa pun yang diperlukan untuk menjalankan fungsi jarak jauh Anda.

Berikut ini adalah contoh struktur folder di bawah bucket S3 yang Anda tentukan dalam file konfigurasi Anda.

```
<s3_root_uri>/ # specified by s3_root_uri or S3RootUri
    <job_name>/ #automatically generated for you
        workdir/workspace.zip # archive of the current working directory (workdir)
        function/ # serialized function
        arguments/ # serialized function arguments
        results/ # returned output from the serialized function including the model
        exception/ # any exceptions from invoking the serialized function
```

Direktori root yang Anda tentukan di bucket S3 Anda tidak dimaksudkan untuk penyimpanan jangka panjang. Data serial terkait erat dengan versi Python dan versi kerangka pembelajaran mesin (ML) yang digunakan selama serialisasi. Jika Anda meng-upgrade versi Python atau kerangka kerja ML, Anda mungkin tidak dapat menggunakan data serial Anda. Sebaliknya, lakukan hal berikut.

- Simpan artefak model dan model Anda dalam format yang agnostik ke versi Python dan kerangka kerja MLmu.
- Jika Anda memutakhirkan kerangka kerja Python atau HTML Anda, akses hasil model Anda dari penyimpanan jangka panjang Anda.

> **Important**
>
> Untuk menghapus data serial Anda setelah jangka waktu tertentu, tetapkan konfigurasi seumur hidup pada bucket S3 Anda.

> **Note**
>
> File yang diserialkan dengan modul acar Python bisa kurang portabel dibandingkan format data lainnya termasuk CSV, Parquet dan JSON. Berhati-hatilah saat memuat file acar dari sumber yang tidak dikenal.

Untuk informasi selengkapnya tentang apa yang harus disertakan dalam file konfigurasi untuk fungsi jarak jauh, lihat File Konfigurasi.

Akses ke data serial Anda

Administrator dapat menyediakan pengaturan untuk data serial Anda, termasuk lokasinya dan pengaturan enkripsi apa pun dalam file konfigurasi. Secara default, data serial dienkripsi dengan Kunci AWS Key Management Service ()AWS KMS. Administrator juga dapat membatasi akses ke direktori root yang Anda tentukan dalam file konfigurasi dengan kebijakan bucket. File konfigurasi dapat dibagikan dan digunakan di seluruh proyek dan pekerjaan. Untuk informasi selengkapnya, lihat File Konfigurasi.

Gunakan `RemoteExecutor` API untuk menjalankan fungsi

Anda dapat menggunakan `RemoteExecutor` API untuk menjalankan fungsi. SageMaker Python SDK akan mengubah kode di dalam `RemoteExecutor` panggilan menjadi pekerjaan pelatihan. SageMaker Pekerjaan pelatihan kemudian akan memanggil fungsi sebagai operasi asinkron dan mengembalikan masa depan. Jika Anda menggunakan `RemoteExecutor` API, Anda dapat menjalankan lebih dari satu pekerjaan pelatihan secara paralel. Untuk informasi lebih lanjut tentang futures di Python, lihat Futures.

Contoh kode berikut menunjukkan cara mengimpor pustaka yang diperlukan, mendefinisikan fungsi, memulai SageMaker instance, dan menggunakan API untuk mengirimkan permintaan untuk menjalankan 2 pekerjaan secara paralel.

```
from sagemaker.remote_function import RemoteExecutor

def matrix_multiply(a, b):
    return np.matmul(a, b)


a = np.array([[1, 0],
```

```
              [0, 1]])
b = np.array([1, 2])

with RemoteExecutor(max_parallel_job=2, instance_type="ml.m5.large") as e:
    future = e.submit(matrix_multiply, a, b)

assert (future.result() == np.array([1,2])).all()
```

`RemoteExecutor`Kelas adalah implementasi dari pustaka Concurrent.futures.Executor.

Contoh kode berikut menunjukkan bagaimana mendefinisikan fungsi dan memanggilnya menggunakan`RemoteExecutor`API. Dalam contoh ini, `RemoteExecutor` akan mengirimkan 4 pekerjaan secara total, tetapi hanya secara 2 paralel. Dua pekerjaan terakhir akan menggunakan kembali cluster dengan overhead minimal.

```
from sagemaker.remote_function.client import RemoteExecutor

def divide(a, b):
    return a/b

with RemoteExecutor(max_parallel_job=2, keep_alive_period_in_seconds=60) as e:
    futures = [e.submit(divide, a, 2) for a in [3, 5, 7, 9]]

for future in futures:
    print(future.result())
```

`max_parallel_job`Parameter hanya berfungsi sebagai mekanisme pembatas laju tanpa mengoptimalkan alokasi sumber daya komputasi. Dalam contoh kode sebelumnya, `RemoteExecutor` tidak mencadangkan sumber daya komputasi untuk dua pekerjaan paralel sebelum pekerjaan apa pun dikirimkan. Untuk informasi selengkapnya tentang `max_parallel_job` atau parameter lain untuk dekorator @remote, lihat Kelas fungsi jarak jauh dan spesifikasi metode.

Kelas masa depan untuk **`RemoteExecutor`** API

Kelas future adalah kelas publik yang mewakili fungsi pengembalian dari pekerjaan pelatihan ketika dipanggil secara asinkron. Kelas future mengimplementasikan class concurrent.futures.future. Kelas ini dapat digunakan untuk melakukan operasi pada pekerjaan yang mendasarinya dan memuat data ke dalam memori.

File konfigurasi

Amazon SageMaker Python SDK mendukung pengaturan nilai default untuk tipe primitif AWS infrastruktur. Setelah administrator mengonfigurasi default ini, mereka secara otomatis diteruskan ketika SageMaker Python SDK memanggil API yang didukung. Argumen untuk fungsi dekorator dapat dimasukkan ke dalam file konfigurasi. Ini agar Anda dapat memisahkan pengaturan yang terkait dengan infrastruktur dari basis kode. Untuk informasi selengkapnya tentang parameter dan argumen untuk fungsi dan metode jarak jauh, lihat Kelas fungsi jarak jauh dan spesifikasi metode.

Anda dapat mengatur pengaturan infrastruktur untuk konfigurasi jaringan, peran IAM, folder Amazon S3 untuk input, data output, dan tag di dalam file konfigurasi. File konfigurasi dapat digunakan saat menjalankan fungsi menggunakan dekorator @remote atau API. `RemoteExecutor`

Contoh file konfigurasi yang mendefinisikan dependensi, sumber daya, dan argumen lainnya berikut. Contoh file konfigurasi ini digunakan untuk memanggil fungsi yang dimulai baik menggunakan dekorator @remote atau API. RemoteExecutor

```
SchemaVersion: '1.0'
SageMaker:
  PythonSDK:
    Modules:
      RemoteFunction:
        Dependencies: 'path/to/requirements.txt'
        EnableInterContainerTrafficEncryption: true
        EnvironmentVariables: {'EnvVarKey': 'EnvVarValue'}
        ImageUri: '366666666666.dkr.ecr.us-west-2.amazonaws.com/my-image:latest'
        IncludeLocalWorkDir: true
        CustomFileFilter:
          IgnoreNamePatterns:
          - "*.ipynb"
          - "data"
        InstanceType: 'ml.m5.large'
        JobCondaEnvironment: 'your_conda_env'
        PreExecutionCommands:
            - 'command_1'
            - 'command_2'
        PreExecutionScript: 'path/to/script.sh'
        RoleArn: 'arn:aws:iam::366666666666:role/MyRole'
        S3KmsKeyId: 'yourkmskeyid'
        S3RootUri: 's3://my-bucket/my-project'
        VpcConfig:
            SecurityGroupIds:
```

```
            - 'sg123'
            Subnets:
            - 'subnet-1234'
        Tags: [{'Key': 'yourTagKey', 'Value':'yourTagValue'}]
        VolumeKmsKeyId: 'yourkmskeyid'
```

Dekorator @remote dan `RemoteExecutor` akan mencari `Dependencies` dalam file konfigurasi berikut:

- File konfigurasi yang ditentukan admin.
- File konfigurasi yang ditentukan pengguna.

Lokasi default untuk file konfigurasi ini bergantung pada, dan relatif terhadap, lingkungan Anda. Contoh kode berikut mengembalikan lokasi default admin dan file konfigurasi pengguna Anda. Perintah ini harus dijalankan di lingkungan yang sama di mana Anda menggunakan SageMaker Python SDK.

```
import os
from platformdirs import site_config_dir, user_config_dir

#Prints the location of the admin config file
print(os.path.join(site_config_dir("sagemaker"), "config.yaml"))

#Prints the location of the user config file
print(os.path.join(user_config_dir("sagemaker"), "config.yaml"))
```

Anda dapat mengganti lokasi default file ini dengan menyetel variabel `SAGEMAKER_ADMIN_CONFIG_OVERRIDE` dan `SAGEMAKER_USER_CONFIG_OVERRIDE` lingkungan untuk jalur file konfigurasi yang ditentukan admin dan yang ditentukan pengguna, masing-masing.

Jika kunci ada di file konfigurasi yang ditentukan admin dan yang ditentukan pengguna, nilai dalam file yang ditentukan pengguna akan digunakan.

Sesuaikan lingkungan runtime Anda

Anda dapat menyesuaikan lingkungan runtime untuk menggunakan lingkungan pengembangan terintegrasi lokal (IDE), SageMaker notebook, atau notebook SageMaker Studio Classic pilihan Anda untuk menulis kode ML Anda. SageMaker akan membantu mengemas dan mengirimkan fungsi Anda dan dependensinya sebagai pekerjaan SageMaker pelatihan. Ini memungkinkan Anda mengakses kapasitas server SageMaker pelatihan untuk menjalankan pekerjaan pelatihan Anda.

Baik dekorator jarak jauh maupun `RemoteExecutor` metode untuk menjalankan fungsi memungkinkan pengguna untuk menentukan dan menyesuaikan lingkungan runtime mereka. Anda dapat menggunakan `requirements.txt` file atau file YAMM lingkungan conda.

Untuk menyesuaikan lingkungan runtime menggunakan file YAMB lingkungan conda dan `requirements.txt` file, lihat contoh kode berikut.

```
# specify a conda environment inside a yaml file
@remote(instance_type="ml.m5.large",
        image_uri = "my_base_python:latest",
        dependencies = "./environment.yml")
def matrix_multiply(a, b):
    return np.matmul(a, b)

# use a requirements.txt file to import dependencies
@remote(instance_type="ml.m5.large",
        image_uri = "my_base_python:latest",
        dependencies = './requirements.txt')
def matrix_multiply(a, b):
    return np.matmul(a, b)
```

Atau, Anda dapat mengatur `dependencies` `auto_capture` agar SDK SageMaker Python menangkap dependensi yang diinstal di lingkungan conda aktif. Berikut ini diperlukan `auto_capture` untuk bekerja dengan andal:

- Anda harus memiliki lingkungan conda yang aktif. Kami menyarankan untuk tidak menggunakan lingkungan `base` conda untuk pekerjaan jarak jauh sehingga Anda dapat mengurangi potensi konflik ketergantungan. Tidak menggunakan lingkungan `base` conda juga memungkinkan pengaturan lingkungan yang lebih cepat dalam pekerjaan jarak jauh.
- Anda tidak boleh memiliki dependensi yang diinstal menggunakan pip dengan nilai untuk parameter. `--extra-index-url`
- Anda tidak boleh memiliki konflik ketergantungan antara paket yang diinstal dengan conda dan paket yang diinstal dengan pip di lingkungan pengembangan lokal.
- Lingkungan pengembangan lokal Anda tidak boleh berisi dependensi khusus sistem operasi yang tidak kompatibel dengan Linux.

Jika `auto_capture` tidak berhasil, kami sarankan Anda meneruskan dependensi Anda sebagai file requirement.txt atau conda environment.yaml, seperti yang dijelaskan dalam contoh pengkodean pertama di bagian ini.

Kompatibilitas gambar kontainer

Tabel berikut menunjukkan daftar gambar SageMaker pelatihan yang kompatibel dengan dekorator @remote.

Nama	Versi Python	URI Gambar - CPU	URI Gambar - GPU
Ilmu Data	3.7 (py37)	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel SageMaker Studio Classic Notebook.	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel SageMaker Studio Classic Notebook.
Ilmu Data 2.0	3.8 (py38)	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel SageMaker Studio Classic Notebook.	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel SageMaker Studio Classic Notebook.
Ilmu Data 3.0	3.10 (py310)	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel

Nama	Versi Python	URI Gambar - CPU	URI Gambar - GPU
		SageMaker Studio Classic Notebook.	SageMaker Studio Classic Notebook.
Dasar Python 2.0	3.8 (py38)	Python SDK memilih gambar ini ketika mendeteksi bahwa lingkungan pengembangan menggunakan runtime Python 3.8. Jika tidak, Python SDK secara otomatis memilih gambar ini saat digunakan sebagai gambar kernel SageMaker Studio Classic Notebook	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel SageMaker Studio Classic Notebook.
Dasar Python 3.0	3.10 (py310)	Python SDK memilih gambar ini ketika mendeteksi bahwa lingkungan pengembangan menggunakan runtime Python 3.8. Jika tidak, Python SDK secara otomatis memilih gambar ini saat digunakan sebagai gambar kernel SageMaker Studio Classic Notebook	Hanya untuk SageMaker Studio Classic Notebook. Python SDK secara otomatis memilih URI gambar saat digunakan sebagai image kernel Studio Classic Notebook.

Nama	Versi Python	URI Gambar - CPU	URI Gambar - GPU
DLC- TensorFlow 2.12.0 untuk Pelatihan SageMaker	3.10 (py310)	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.12.0-cpu-py310-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.12.0-gpu-py310-cu118-ubuntu20.04-sagemaker
DLC-Tensorflow 2.11.0 untuk pelatihan SageMaker	3.9 (py39)	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.11.0-cpu-py39-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.11.0-gpu-py39-cu112-ubuntu20.04-sagemaker
DLC- TensorFlow 2.10.1 untuk pelatihan SageMaker	3.9 (py39)	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.10.1-cpu-py39-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.10.1-gpu-py39-cu112-ubuntu20.04-sagemaker
DLC- TensorFlow 2.9.2 untuk pelatihan SageMaker	3.9 (py39)	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.9.2-cpu-py39-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.9.2-gpu-py39-cu112-ubuntu20.04-sagemaker

Nama	Versi Python	URI Gambar - CPU	URI Gambar - GPU
DLC- TensorFlow 2.8.3 untuk pelatihan SageMaker	3.9 (py39)	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.8.3-cpu-py39-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/ tensorflow-training:2.8.3-gpu-py39-cu112-ubuntu20.04-sagemaker
DLC- PyTorch 2.0.0 untuk pelatihan SageMaker	3.10 (py310)	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:2.0.0-cpu-py310-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:2.0.0-gpu-py310-cu118-ubuntu20.04-sagemaker
DLC- PyTorch 1.13.1 untuk pelatihan SageMaker	3.9 (py39)	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:1.13.1-cpu-py39-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:1.13.1-gpu-py39-cu117-ubuntu20.04-sagemaker
DLC- PyTorch 1.12.1 untuk pelatihan SageMaker	3.8 (py38)	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:1.12.1-cpu-py38-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:1.12.1-gpu-py38-cu113-ubuntu20.04-sagemaker

Nama	Versi Python	URI Gambar - CPU	URI Gambar - GPU
DLC- PyTorch 1.11.0 untuk pelatihan SageMaker	3.8 (py38)	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:1.11.0-cpu-py38-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/pytorch-training:1.11.0-gpu-py38-cu113-ubuntu20.04-sagemaker
DLC-MXNet 1.9.0 untuk pelatihan SageMaker	3.8 (py38)	763104351884.dkr.ecr. <region>. amazonaws.com/mxnet-training:1.9.0-cpu-py38-ubuntu20.04-sagemaker	763104351884.dkr.ecr. <region>. amazonaws.com/mxnet-training:1.9.0-gpu-py38-cu112-ubuntu20.04-sagemaker

Note

Untuk menjalankan pekerjaan secara lokal menggunakan gambar AWS Deep Learning Containers (DLC), gunakan URI gambar yang ditemukan dalam dokumentasi DLC. Gambar DLC tidak mendukung `auto_capture` nilai untuk dependensi.
Pekerjaan dengan SageMakerDistribusi di SageMaker Studio berjalan dalam wadah sebagai pengguna non-root bernama`sagemaker-user`. Pengguna ini membutuhkan izin penuh untuk mengakses `/opt/ml` dan`/tmp`. Berikan izin ini dengan menambahkan `sudo chmod -R 777 /opt/ml /tmp` ke `pre_execution_commands` daftar, seperti yang ditunjukkan pada cuplikan berikut:

```
@remote(pre_execution_commands=["sudo chmod -R 777 /opt/ml /tmp"])
def func():
    pass
```

Anda juga dapat menjalankan fungsi jarak jauh dengan gambar khusus Anda. Untuk kompatibilitas dengan fungsi jarak jauh, gambar khusus harus dibuat dengan Python versi 3.7.x-3.10.x. Berikut

ini adalah contoh Dockerfile minimal yang menunjukkan cara menggunakan image Docker dengan Python 3.10.

```
FROM python:3.10

#... Rest of the Dockerfile
```

Untuk membuat `conda` lingkungan dalam gambar Anda dan menggunakannya untuk menjalankan pekerjaan, atur variabel lingkungan `SAGEMAKER_JOB_CONDA_ENV` ke nama `conda` lingkungan. Jika gambar Anda memiliki `SAGEMAKER_JOB_CONDA_ENV` nilai yang ditetapkan, fungsi jarak jauh tidak dapat membuat lingkungan conda baru selama runtime pekerjaan pelatihan. Lihat contoh Dockerfile berikut yang menggunakan `conda` lingkungan dengan Python versi 3.10.

```
FROM continuumio/miniconda3:4.12.0

ENV SHELL=/bin/bash \
    CONDA_DIR=/opt/conda \
    SAGEMAKER_JOB_CONDA_ENV=sagemaker-job-env

RUN conda create -n $SAGEMAKER_JOB_CONDA_ENV \
   && conda install -n $SAGEMAKER_JOB_CONDA_ENV python=3.10 -y \
   && conda clean --all -f -y \
```

SageMaker Untuk menggunakan mamba untuk mengelola lingkungan virtual Python Anda di image container, instal toolkit mamba dari miniforge. Untuk menggunakan mamba, tambahkan contoh kode berikut ke Dockerfile Anda. Kemudian, SageMaker akan mendeteksi `mamba` ketersediaan saat runtime dan menggunakannya sebagai gantinya. `conda`

```
#Mamba Installation
RUN curl -L -O "https://github.com/conda-forge/miniforge/releases/latest/download/
Mambaforge-Linux-x86_64.sh" \
    && bash Mambaforge-Linux-x86_64.sh -b -p "/opt/conda"  \
    && /opt/conda/bin/conda init bash
```

Menggunakan saluran conda khusus pada bucket Amazon S3 tidak kompatibel dengan mamba saat menggunakan fungsi jarak jauh. Jika Anda memilih untuk menggunakan mamba, pastikan Anda tidak menggunakan saluran conda khusus di Amazon S3. Untuk informasi selengkapnya, lihat bagian Prasyarat di bawah Repositori conda kustom menggunakan Amazon S3.

Berikut ini adalah contoh Dockerfile lengkap yang menunjukkan cara membuat image Docker yang kompatibel.

```
FROM python:3.10

RUN apt-get update -y \
    # Needed for awscli to work
    # See: https://github.com/aws/aws-cli/issues/1957#issuecomment-687455928
    && apt-get install -y groff unzip curl \
    && pip install --upgrade \
        'boto3>1.0<2' \
        'awscli>1.0<2' \
        'ipykernel>6.0.0<7.0.0' \
#Use ipykernel with --sys-prefix flag, so that the absolute path to
    #/usr/local/share/jupyter/kernels/python3/kernel.json python is used
    # in kernelspec.json file
    && python -m ipykernel install --sys-prefix

#Install Mamba
RUN curl -L -O "https://github.com/conda-forge/miniforge/releases/latest/download/
Mambaforge-Linux-x86_64.sh" \
    && bash Mambaforge-Linux-x86_64.sh -b -p "/opt/conda"  \
    && /opt/conda/bin/conda init bash

#cleanup
RUN apt-get clean \
    && rm -rf /var/lib/apt/lists/* \
    && rm -rf ${HOME}/.cache/pip \
    && rm Mambaforge-Linux-x86_64.sh

ENV SHELL=/bin/bash \
    PATH=$PATH:/opt/conda/bin
```

Gambar yang dihasilkan dari menjalankan contoh Dockerfile sebelumnya juga dapat digunakan sebagai image kernel SageMaker Studio Classic.

Mencatat parameter dan metrik dengan Amazon Experiments SageMaker

Panduan ini menunjukkan cara mencatat parameter dan metrik dengan SageMaker Eksperimen Amazon. SageMakerEksperimen terdiri dari run, dan setiap run terdiri dari semua input, parameter, konfigurasi, dan hasil untuk interaksi pelatihan model tunggal.

Anda dapat mencatat parameter dan metrik dari fungsi jarak jauh menggunakan dekorator @remote atau API. `RemoteExecutor`

Untuk mencatat parameter dan metrik dari fungsi jarak jauh, pilih salah satu metode berikut:

- Buat instance SageMaker eksperimen yang dijalankan di dalam fungsi jarak jauh menggunakan `Run` dari pustaka SageMaker Eksperimen. Untuk informasi selengkapnya, lihat Membuat SageMaker Eksperimen Amazon.
- Gunakan `load_run` fungsi di dalam fungsi jarak jauh dari pustaka SageMaker Eksperimen. Ini akan memuat `Run` instance yang dideklarasikan di luar fungsi jarak jauh.

Bagian berikut menunjukkan cara membuat dan melacak garis keturunan dengan SageMaker eksperimen yang dijalankan dengan menggunakan metode yang tercantum sebelumnya. Bagian tersebut juga menjelaskan kasus-kasus yang tidak didukung oleh SageMaker pelatihan.

Gunakan dekorator @remote untuk berintegrasi dengan Eksperimen SageMaker

Anda dapat membuat instance eksperimen SageMaker, atau memuat SageMaker eksperimen saat ini dari dalam fungsi jarak jauh. Bagian berikut menunjukkan Anda menunjukkan untuk menggunakan salah satu metode.

Buat eksperimen dengan SageMaker Eksperimen

Anda dapat membuat eksperimen yang dijalankan dalam SageMaker eksperimen. Untuk melakukan ini, Anda meneruskan nama eksperimen, menjalankan nama, dan parameter lainnya ke dalam fungsi jarak jauh Anda.

Contoh kode berikut mengimpor nama eksperimen Anda, nama run, dan parameter yang akan dicatat selama setiap proses. Parameter `param_1` dan `param_2` dicatat dari waktu ke waktu di dalam loop pelatihan. Parameter umum mungkin termasuk ukuran batch atau zaman. Dalam contoh ini, metrik `metric_a` dan `metric_b` dicatat untuk menjalankan dari waktu ke waktu di dalam loop pelatihan. Metrik umum lainnya mungkin termasuk `accuracy` atau`loss`.

```
from sagemaker.remote_function import remote
from sagemaker.experiments.run import Run

# Define your remote function
@remote
def train(value_1, value_2, exp_name, run_name):
    ...
```

```
    ...
    #Creates the experiment
    with Run(
        experiment_name=exp_name,
        run_name=run_name,
    ) as run:
        ...
        #Define values for the parameters to log
        run.log_parameter("param_1", value_1)
        run.log_parameter("param_2", value_2)
        ...
        #Define metrics to log
        run.log_metric("metric_a", 0.5)
        run.log_metric("metric_b", 0.1)


# Invoke your remote function
train(1.0, 2.0, "my-exp-name", "my-run-name")
```

Muat SageMaker Eksperimen saat ini dengan pekerjaan yang diprakarsai oleh dekorator @remote

Gunakan `load_run()` fungsi dari pustaka SageMaker Eksperimen untuk memuat objek run saat ini dari konteks run. Anda juga dapat menggunakan `load_run()` fungsi dalam fungsi jarak jauh Anda. Muat objek run diinisialisasi lokal oleh `with` pernyataan pada objek run seperti yang ditunjukkan pada contoh kode berikut.

```
from sagemaker.experiments.run import Run, load_run

# Define your remote function
@remote
def train(value_1, value_2):
    ...
    ...
    with load_run() as run:
        run.log_metric("metric_a", value_1)
        run.log_metric("metric_b", value_2)


# Invoke your remote function
with Run(
    experiment_name="my-exp-name",
    run_name="my-run-name",
) as run:
```

```
    train(0.5, 1.0)
```

Memuat eksperimen saat ini yang dijalankan dalam pekerjaan yang dimulai dengan API **`RemoteExecutor`**

Anda juga dapat memuat SageMaker eksperimen saat ini jika pekerjaan Anda dimulai dengan `RemoteExecutor` API. Contoh kode berikut menunjukkan cara menggunakan `RemoteExecutor` API dengan `load_run` fungsi SageMaker Eksperimen. Anda melakukan ini untuk memuat SageMaker eksperimen yang dijalankan saat ini dan menangkap metrik dalam pekerjaan yang dikirimkan oleh`RemoteExecutor`.

```
from sagemaker.experiments.run import Run, load_run

def square(x):
    with load_run() as run:
        result = x * x
        run.log_metric("result", result)
    return result


with RemoteExecutor(
    max_parallel_job=2,
    instance_type="ml.m5.large"
) as e:
    with Run(
        experiment_name="my-exp-name",
        run_name="my-run-name",
    ):
        future_1 = e.submit(square, 2)
```

Penggunaan SageMaker Eksperimen yang tidak didukung saat membuat anotasi kode Anda dengan dekorator @remote

SageMaker tidak mendukung meneruskan objek `Run` tipe ke fungsi @remote atau menggunakan `Run` objek global. Contoh berikut menunjukkan kode yang akan melempar`SerializationError`.

Contoh kode berikut mencoba untuk meneruskan objek `Run` tipe ke dekorator @remote, dan itu menghasilkan kesalahan.

```
@remote
```

```
def func(run: Run):
    run.log_metrics("metric_a", 1.0)

with Run(...) as run:
    func(run) ---> SerializationError caused by NotImplementedError
```

Contoh kode berikut mencoba menggunakan `run` objek global yang dipakai di luar fungsi jarak jauh. Dalam contoh kode, `train()` fungsi didefinisikan di dalam `with Run` konteks, merujuk objek run global dari dalam. Ketika `train()` dipanggil, itu menghasilkan kesalahan.

```
with Run(...) as run:
    @remote
    def train(metric_1, value_1, metric_2, value_2):
        run.log_parameter(metric_1, value_1)
        run.log_parameter(metric_2, value_2)

    train("p1", 1.0, "p2", 0.5) ---> SerializationError caused by NotImplementedError
```

Menggunakan kode modular dengan dekorator @remote

Anda dapat mengatur kode Anda ke dalam modul untuk kemudahan manajemen ruang kerja selama pengembangan dan masih menggunakan fungsi @remote untuk menjalankan fungsi. Anda juga dapat mereplikasi modul lokal dari lingkungan pengembangan Anda ke lingkungan kerja jarak jauh. Untuk melakukannya, atur parameter `include_local_workdir` ke`True`, seperti yang ditunjukkan pada contoh kode berikut.

```
@remote(
  include_local_workdir=True,
)
```

> **Note**
>
> Dekorator dan parameter @remote harus muncul di file utama, bukan di file dependen mana pun.

Ketika `include_local_workdir` diatur ke`True`, SageMaker paket semua skrip Python sambil mempertahankan struktur direktori dalam direktori proses saat ini. Itu juga membuat dependensi tersedia di direktori kerja pekerjaan.

Misalnya, skrip Python Anda yang memproses dataset MNIST dibagi menjadi skrip dan `main.py` skrip dependen. `pytorch_mnist.py` `main.py`memanggil skrip dependen. Juga, `main.py` skrip berisi kode untuk mengimpor ketergantungan seperti yang ditunjukkan.

```
from mnist_impl.pytorch_mnist import ...
```

`main.py`File juga harus berisi `@remote` dekorator, dan harus mengatur `include_local_workdir` parameter ke`True`.

`include_local_workdir`Parameter secara default mencakup semua skrip Python di direktori. Anda dapat menyesuaikan file mana yang ingin Anda unggah ke pekerjaan dengan menggunakan parameter ini bersama dengan `custom_file_filter` parameter. Anda dapat meneruskan fungsi yang memfilter dependensi pekerjaan untuk diunggah ke S3, atau `CustomFileFilter` objek yang menentukan direktori dan file lokal untuk diabaikan dalam fungsi jarak jauh. Anda `custom_file_filter` hanya dapat menggunakan jika `include_local_workdir` disetel ke `True` —jika tidak parameter diabaikan.

Contoh berikut digunakan `CustomFileFilter` untuk mengabaikan semua file notebook dan folder atau file bernama `data` saat mengunggah file ke S3.

```
@remote(
   include_local_workdir=True,
   custom_file_filter=CustomFileFilter(
      ignore_pattern_names=[ # files or directories to ignore
        "*.ipynb", # all notebook files
        "data", # folter or file named data
      ]
   )
)
```

Contoh berikut menunjukkan bagaimana Anda dapat mengemas seluruh ruang kerja.

```
@remote(
   include_local_workdir=True,
   custom_file_filter=CustomFileFilter(
      ignore_pattern_names=[] # package whole workspace
   )
)
```

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan fungsi untuk memfilter file.

```
import os

def my_filter(path: str, files: List[str]) -> List[str]:
    to_ignore = []
   for file in files:
       if file.endswith(".txt") or file.endswith(".ipynb"):
           to_ignore.append(file)
   return to_ignore

@remote(
   include_local_workdir=True,
   custom_file_filter=my_filter
)
```

Praktik terbaik dalam menyusun direktori kerja Anda

Praktik terbaik berikut menyarankan bagaimana Anda dapat mengatur struktur direktori Anda saat menggunakan `@remote` dekorator dalam kode modular Anda.

- Letakkan dekorator @remote dalam file yang berada di direktori tingkat root ruang kerja.
- Struktur modul lokal di tingkat root.

Contoh gambar berikut menunjukkan struktur direktori direkomendasikan. Dalam struktur contoh ini, `main.py` skrip terletak di direktori tingkat root.

```
.
### config.yaml
### data/
### main.py <----------------- @remote used here
### mnist_impl
# ### __pycache__/
# # ### pytorch_mnist.cpython-310.pyc
# ### pytorch_mnist.py <-------- dependency of main.py
### requirements.txt
```

Contoh gambar berikut menunjukkan struktur direktori yang akan menghasilkan perilaku yang tidak konsisten ketika digunakan untuk membubuhi keterangan kode Anda dengan dekorator @remote.

Dalam struktur contoh ini, `main.py` skrip yang berisi dekorator @remote tidak terletak di direktori tingkat root. Struktur berikut TIDAK direkomendasikan.

```
.
### config.yaml
### entrypoint
# ### data
# ### main.py <----------------- @remote used here
### mnist_impl
# ### __pycache__
# # ### pytorch_mnist.cpython-310.pyc
# ### pytorch_mnist.py <-------- dependency of main.py
### requirements.txt
```

Repositori pribadi untuk dependensi runtime

Anda dapat menggunakan perintah atau skrip pra-eksekusi untuk mengonfigurasi manajer ketergantungan seperti pip atau conda di lingkungan pekerjaan Anda. Untuk mencapai isolasi jaringan, gunakan salah satu opsi ini untuk mengarahkan manajer ketergantungan Anda untuk mengakses repositori pribadi Anda dan menjalankan fungsi jarak jauh dalam VPC. Perintah atau skrip pra-eksekusi akan berjalan sebelum fungsi jarak jauh Anda berjalan. Anda dapat mendefinisikannya dengan dekorator @remote, `RemoteExecutor` API, atau dalam file konfigurasi.

Bagian berikut menunjukkan cara mengakses repositori Python Package Index (PyPI) pribadi yang dikelola dengan. AWS CodeArtifact Bagian ini juga menunjukkan cara mengakses saluran conda khusus yang dihosting di Amazon Simple Storage Service (Amazon S3).

Cara menggunakan repositori PyPI khusus yang dikelola dengan AWS CodeArtifact

Untuk menggunakan CodeArtifact untuk mengelola repositori PyPI kustom, prasyarat berikut diperlukan:

- Repositori PyPI pribadi Anda seharusnya sudah dibuat. Anda dapat memanfaatkan AWS CodeArtifact untuk membuat dan mengelola repositori paket pribadi Anda. Untuk mempelajari selengkapnya CodeArtifact, lihat Panduan CodeArtifact Pengguna.
- VPC Anda harus memiliki akses ke repositori Anda CodeArtifact . Untuk mengizinkan koneksi dari VPC Anda ke CodeArtifact repositori Anda, Anda harus melakukan hal berikut:
  - Buat titik akhir VPC untuk. CodeArtifact
  - Buat titik akhir gateway Amazon S3 untuk VPC Anda, yang memungkinkan CodeArtifact untuk menyimpan aset paket.

Contoh perintah pra-eksekusi berikut menunjukkan cara mengkonfigurasi pip dalam tugas SageMaker pelatihan untuk menunjuk ke repositori Anda CodeArtifact . Untuk informasi selengkapnya, lihat Mengkonfigurasi dan menggunakan pip dengan CodeArtifact.

```
# use a requirements.txt file to import dependencies
@remote(
    instance_type="ml.m5.large"
    image_uri = "my_base_python:latest",
    dependencies = './requirements.txt',
    pre_execution_commands=[
        "aws codeartifact login --tool pip --domain my-org --domain-owner
 <000000000000> --repository my-codeartifact-python-repo --endpoint-url https://vpce-
xxxxx.api.codeartifact.us-east-1.vpce.amazonaws.com"
    ]
)
def matrix_multiply(a, b):
    return np.matmul(a, b)
```

Cara menggunakan saluran conda khusus yang dihosting di Amazon S3

Untuk menggunakan Amazon S3 untuk mengelola repositori conda kustom, diperlukan prasyarat berikut:

- Saluran conda pribadi Anda harus sudah diatur di bucket Amazon S3 Anda, dan semua paket dependen harus diindeks dan diunggah ke bucket Amazon S3 Anda. Untuk petunjuk tentang cara mengindeks paket conda Anda, lihat Membuat saluran khusus.
- VPC Anda harus memiliki akses ke bucket Amazon S3. Untuk informasi selengkapnya, lihat Titik Akhir untuk Amazon S3.
- Lingkungan conda dasar dalam gambar pekerjaan Anda seharusnya sudah `boto3` diinstal. Untuk memeriksa lingkungan Anda, masukkan yang berikut ini di prompt Anaconda Anda untuk memeriksa yang `boto3` muncul di daftar yang dihasilkan.

  ```
  conda list -n base
  ```

- Gambar pekerjaan Anda harus diinstal dengan conda, bukan mamba. Untuk memeriksa lingkungan Anda, pastikan bahwa prompt kode sebelumnya tidak kembali`mamba`.

Contoh perintah pra-eksekusi berikut menunjukkan cara mengonfigurasi conda dalam tugas SageMaker pelatihan untuk menunjuk ke saluran pribadi Anda di Amazon S3 Perintah pra-eksekusi menghapus saluran default dan menambahkan saluran khusus ke file konfigurasi conda. `.condarc`

```
# specify your dependencies inside a conda yaml file
@remote(
    instance_type="ml.m5.large"
    image_uri = "my_base_python:latest",
    dependencies = "./environment.yml",
    pre_execution_commands=[
        "conda config --remove channels 'defaults'"
        "conda config --add channels 's3://my_bucket/my-conda-repository/conda-
forge/'",
        "conda config --add channels 's3://my_bucket/my-conda-repository/main/'"
    ]
)
def matrix_multiply(a, b):
    return np.matmul(a, b)
```

Notebook contoh

Anda dapat mengubah kode pelatihan di lingkungan ruang kerja yang ada dan kode pemrosesan data serta kumpulan data terkait menjadi SageMaker pekerjaan pelatihan. Notebook berikut menunjukkan cara menyesuaikan lingkungan, pengaturan pekerjaan, dan lainnya untuk masalah klasifikasi gambar, menggunakan algoritma XGBoost dan Hugging Face.

Notebook quick_start berisi contoh kode berikut:

- Cara menyesuaikan pengaturan pekerjaan Anda dengan file konfigurasi.
- Cara memanggil fungsi Python sebagai pekerjaan, secara asinkron.
- Cara menyesuaikan lingkungan runtime pekerjaan dengan membawa dependensi tambahan.
- Cara menggunakan dependensi lokal dengan metode fungsi @remote.

Notebook berikut memberikan contoh kode tambahan untuk berbagai jenis dan implementasi masalah ML.

- Untuk melihat contoh kode untuk menggunakan dekorator @remote untuk masalah klasifikasi gambar, buka buku catatan pytorch_mnist.ipynb. Masalah klasifikasi ini mengenali digit tulisan

tangan menggunakan kumpulan data sampel Modified National Institute of Standards and Technology (MNIST).

- Untuk melihat contoh kode untuk menggunakan dekorator @remote untuk masalah klasifikasi gambar sebelumnya dengan skrip, lihat skrip contoh Pytorch MNIST, train.py.
- Untuk melihat bagaimana algoritma XGBoost diimplementasikan dengan dekorator @remote: Buka notebook xgboost_abalone.ipynb.
- Untuk melihat bagaimana Hugging Face terintegrasi dengan dekorator @remote: Buka notebook huggingface.ipynb.

Kelola Machine Learning dengan Amazon SageMaker Experiments

Amazon SageMaker Experiments adalah kemampuan Amazon SageMaker yang memungkinkan Anda membuat, mengelola, menganalisis, dan membandingkan eksperimen pembelajaran mesin Anda.

Eksperimen dalam pembelajaran mesin

Pembelajaran mesin adalah proses berulang. Anda perlu bereksperimen dengan beberapa kombinasi data, algoritme, dan parameter, sambil mengamati dampak perubahan tambahan pada akurasi model. Seiring waktu, eksperimen berulang ini dapat menghasilkan ribuan pelatihan model dan versi model. Ini membuat sulit untuk melacak model berkinerja terbaik dan konfigurasi inputnya. Juga sulit untuk membandingkan eksperimen aktif dengan eksperimen sebelumnya untuk mengidentifikasi peluang untuk peningkatan tambahan lebih lanjut. Gunakan SageMaker Eksperimen untuk mengatur, melihat, menganalisis, dan membandingkan eksperimen ML berulang untuk mendapatkan wawasan komparatif dan melacak model berkinerja terbaik Anda.

Kelola eksperimen ML dengan SageMaker Eksperimen

SageMaker Eksperimen secara otomatis melacak input, parameter, konfigurasi, dan hasil iterasi Anda saat berjalan. Anda dapat menetapkan, mengelompokkan, dan mengatur proses ini ke dalam eksperimen. SageMakerEksperimen terintegrasi dengan Amazon SageMaker Studio Classic, menyediakan antarmuka visual untuk menelusuri eksperimen aktif dan sebelumnya, membandingkan proses pada metrik kinerja utama, dan mengidentifikasi model berkinerja terbaik. SageMaker Eksperimen melacak semua langkah dan artefak yang digunakan untuk membuat model, dan Anda dapat dengan cepat meninjau kembali asal model saat Anda memecahkan masalah dalam produksi, atau mengaudit model Anda untuk verifikasi kepatuhan.

Gunakan SageMaker Eksperimen untuk melihat, mengelola, menganalisis, dan membandingkan kedua eksperimen kustom yang Anda buat secara terprogram dan eksperimen yang dibuat secara otomatis dari SageMaker pekerjaan.

Wilayah AWS yang Didukung

SageMaker Eksperimen umumnya tersedia di semua Wilayah AWS komersial di mana Amazon SageMaker Studio Classic tersedia, kecuali Wilayah China.

Topik

Buat SageMaker Eksperimen Amazon

Buat SageMaker eksperimen Amazon untuk melacak alur kerja machine learning (ML) Anda dengan beberapa baris kode dari lingkungan pengembangan pilihan Anda. Anda kemudian dapat menelusuri eksperimen Anda, membuat visualisasi untuk analisis, dan menemukan model berkinerja terbaik. Anda juga dapat mengintegrasikan SageMaker Eksperimen ke dalam skrip SageMaker pelatihan Anda menggunakan SageMaker Python SDK.

Ikhtisar

Komponen-komponen berikut membentuk blok bangunan percobaan di Amazon SageMaker.

- `experiment`: Eksperimen adalah kumpulan lari. Saat Anda menginisialisasi lari di loop pelatihan Anda, Anda menyertakan nama eksperimen yang menjadi milik run tersebut. Nama eksperimen harus unik dalam AWS akun Anda.

- Run: Lari terdiri dari semua input, parameter, konfigurasi, dan hasil untuk satu interaksi pelatihan model. Inisialisasi percobaan untuk melacak pekerjaan pelatihan dengan`Run.init()`.

> **Note**
>
> Kami menyarankan Anda menginisialisasi Run objek di Notebook Jupyter, dan membuat SageMaker pekerjaan untuk eksperimen Anda dalam konteks inisialisasi objek iniRun. Untuk merujuk ke Run objek ini dalam mode skrip, gunakan `load_run()` operasi. Sebagai contoh, lihat Contoh notebook untuk Eksperimen Amazon SageMaker .

> **Note**
>
> SageMaker Python SDK secara otomatis mengubah nama eksperimen dan menjalankan nama menjadi huruf kecil.

- `load_run`: Untuk menjalankan eksperimen Anda dalam mode skrip, lihat Run objek yang diinisialisasi dengan`load_run()`. Jika eksperimen untuk menjalankan ada, `load_run` mengembalikan konteks eksperimen. Umumnya, Anda menggunakan `load_run` tanpa argumen untuk melacak metrik, parameter, dan artefak dalam skrip pekerjaan SageMaker pelatihan atau pemrosesan.

```
# Load run from a local script passing experiment and run names
with load_run(experiment_name=experiment_name, run_name=run_name) as run:
   run.log_parameter("param1", "value1")
```

```
# Load run within a training or processing Job (automated context sharing)
with load_run() as run:
    run.log_parameter("param1", "value1")
```

- `log_parameter`: Log parameter untuk menjalankan, seperti ukuran batch atau epoch, dari waktu ke waktu dalam loop pelatihan dengan. `run.log_parameter()` `log_parameter`merekam pasangan nama-nilai tunggal dalam proses. Anda dapat menggunakan `run.log_parameters()` untuk mencatat beberapa parameter. Jika dipanggil beberapa kali dalam menjalankan parameter dengan nama yang sama, `log_parameter` menimpa nilai sebelumnya. Nama harus berupa string dan nilainya harus berupa string, integer, atau float.

```
# Log a single parameter
```

```
run.log_parameter("param1", "value1")
```

```
# Log multiple parameters
run.log_parameters({
    "param2": "value2",
    "param3": "value3"
})
```

- `log_metric`: Log metrik untuk menjalankan, seperti akurasi atau kehilangan, dari waktu ke waktu dalam loop pelatihan dengan`run.log_metric()`. `log_metric`mencatat pasangan nama-nilai di mana nama adalah string dan nilainya adalah integer atau float. Untuk mendeklarasikan frekuensi logging selama proses berjalan, tentukan nilai. `step` Anda kemudian dapat memvisualisasikan metrik ini di UI Eksperimen Klasik Studio. Untuk informasi selengkapnya, lihat Lihat, cari, dan bandingkan percobaan yang dijalankan.

```
# Log a metric over the course of a run
run.log_metric(name="Final_loss", value=finalloss)
```

```
# Log a metric over the course of a run at each epoch
run.log_metric(name="test:loss", value=loss, step=epoch)
```

- `log_artifact`: Log artefak input atau output apa pun yang terkait dengan run with`run.log_artifact()`. Artefak log seperti URI S3, kumpulan data, model, dan lainnya untuk eksperimen Anda guna membantu Anda melacak artefak di beberapa proses. `is_output`adalah secara `True` default. Untuk merekam artefak sebagai artefak masukan alih-alih artefak keluaran, atur ke. `is_output` `False`

```
# Track a string value as an input or output artifact
run.log_artifact(name="training_data", value="data.csv" is_output=False)
```

- `log_file`: Log file input atau output apa pun yang terkait dengan proses, seperti data pelatihan atau pengujian, dan simpan di Amazon S3 dengan. `run.log_file()` `is_output`adalah secara `True` default. Untuk merekam file sebagai artefak input alih-alih artefak keluaran, atur `is_output` ke. `False`

```
# Upload a local file to S3 and track it as an input or output artifact
run.log_file("training_data.csv", name="training_data", is_output=False)
```

Untuk informasi selengkapnya tentang menginisialisasi Run objek, lihat Eksperimen dalam dokumentasi SageMaker Python SDK. Untuk informasi tentang memvisualisasikan data eksperimen yang dicatat dan pencatatan otomatis, lihatLihat, cari, dan bandingkan percobaan yang dijalankan.

Buat eksperimen dengan SageMaker Python SDK

Bagian berikut menunjukkan cara membuat SageMaker Eksperimen Amazon menggunakan SageMaker Python SDK. Contoh ini menggunakan Run kelas untuk melacak model Keras di lingkungan notebook. CallbackKelas Keras menyediakan operasi on_epoch_end yang memancarkan metrik pada akhir setiap zaman. Pertama, tentukan Callback kelas.

```
class ExperimentCallback(keras.callbacks.Callback):
    """ """

    def __init__(self, run, model, x_test, y_test):
        """Save params in constructor"""
        self.run = run
        self.model = model
        self.x_test = x_test
        self.y_test = y_test

    def on_epoch_end(self, epoch, logs=None):
        """ """
        keys = list(logs.keys())
        for key in keys:
            run.log_metric(name=key, value=logs[key], step=epoch)
            print("Epoch: {}\n{} -> {}".format(epoch, key, logs[key]))
```

Selanjutnya, latih model Keras di lingkungan notebook dan lacak sebagai percobaan.

> **Note**
>
> Contoh ini melakukan pekerjaan secara berurutan. Untuk menjalankan SageMaker pekerjaan secara asinkron, Anda mungkin perlu meningkatkan batas sumber daya Anda.

```
from sagemaker.experiments import Run

# The run name is an optional argument to `run.init()`
with Run(experiment_name = 'my-experiment') as run:
```

```
    # Define values for the parameters to log
    run.log_parameter("batch_size", batch_size)
    run.log_parameter("epochs", epochs)
    run.log_parameter("dropout", 0.5)

    # Define input artifacts
    run.log_file('datasets/input_train.npy', is_output = False)
    run.log_file('datasets/input_test.npy', is_output = False)
    run.log_file('datasets/input_train_labels.npy', is_output = False)
    run.log_file('datasets/input_test_labels.npy', is_output = False)

    # Train locally
    model.fit(
        x_train,
        y_train,
        batch_size=batch_size,
        epochs=epochs,
        validation_split=0.1,
        callbacks = [ExperimentCallback(run, model, x_test, y_test)]
    )

    score = model.evaluate(x_test, y_test, verbose=0)
    print("Test loss:", score[0])
    print("Test accuracy:", score[1])

    # Define metrics to log
    run.log_metric(name = "Final Test Loss", value = score[0])
    run.log_metric(name = "Final Test Accuracy", value = score[1])
```

Untuk lebih banyak contoh kode dan contoh buku catatan, lihatContoh notebook untuk Eksperimen Amazon SageMaker .

Buat eksperimen menggunakan mode SageMaker skrip

Anda dapat menggunakan mode SageMaker skrip untuk menulis kode Anda sendiri untuk melatih model dan melacaknya sebagai percobaan. Saat membuat eksperimen dengan mode skrip, gunakan`load_run()`.

```
# Make sure that you have the latest version of the SageMaker Python SDK
import os
os.system("pip install -U sagemaker")
```

```
# Import additional requirements
import boto3
from sagemaker.session import Session
from sagemaker.experiments.run import load_run

# Define training script
if __name__ == "__main__":
    session = Session(boto3.session.Session(region_name=args.region))
    with load_run(sagemaker_session=session) as run:
        # Define values for the parameters to log
        run.log_parameters({
            "batch_size": batch_size,
            "epochs": epochs,
            "dropout": 0.5
        })
        # Define input artifacts
        run.log_file('datasets/input_train.npy', is_output = False)
        run.log_file('datasets/input_test.npy', is_output = False)
        run.log_file('datasets/input_train_labels.npy', is_output = False)
        run.log_file('datasets/input_test_labels.npy', is_output = False)

        # Train the model
        model.fit(
            x_train,
            y_train,
            batch_size=batch_size,
            epochs=epochs,
            validation_split=0.1,
            callbacks = [ExperimentCallback(run, model, x_test, y_test)]
        )

        score = model.evaluate(x_test, y_test, verbose=0)
        print("Test loss:", score[0])
        print("Test accuracy:", score[1])

        # Define metrics to log
        run.log_metric(name = "Final Test Loss", value = score[0])
        run.log_metric(name = "Final Test Accuracy", value = score[1])
```

Untuk contoh kode lainnya dan contoh buku catatan tentang penggunaan SageMaker Eksperimen Amazon dalam mode SageMaker skrip, lihat[Lacak eksperimen untuk pekerjaan SageMaker pelatihan menggunakan mode skrip](#).

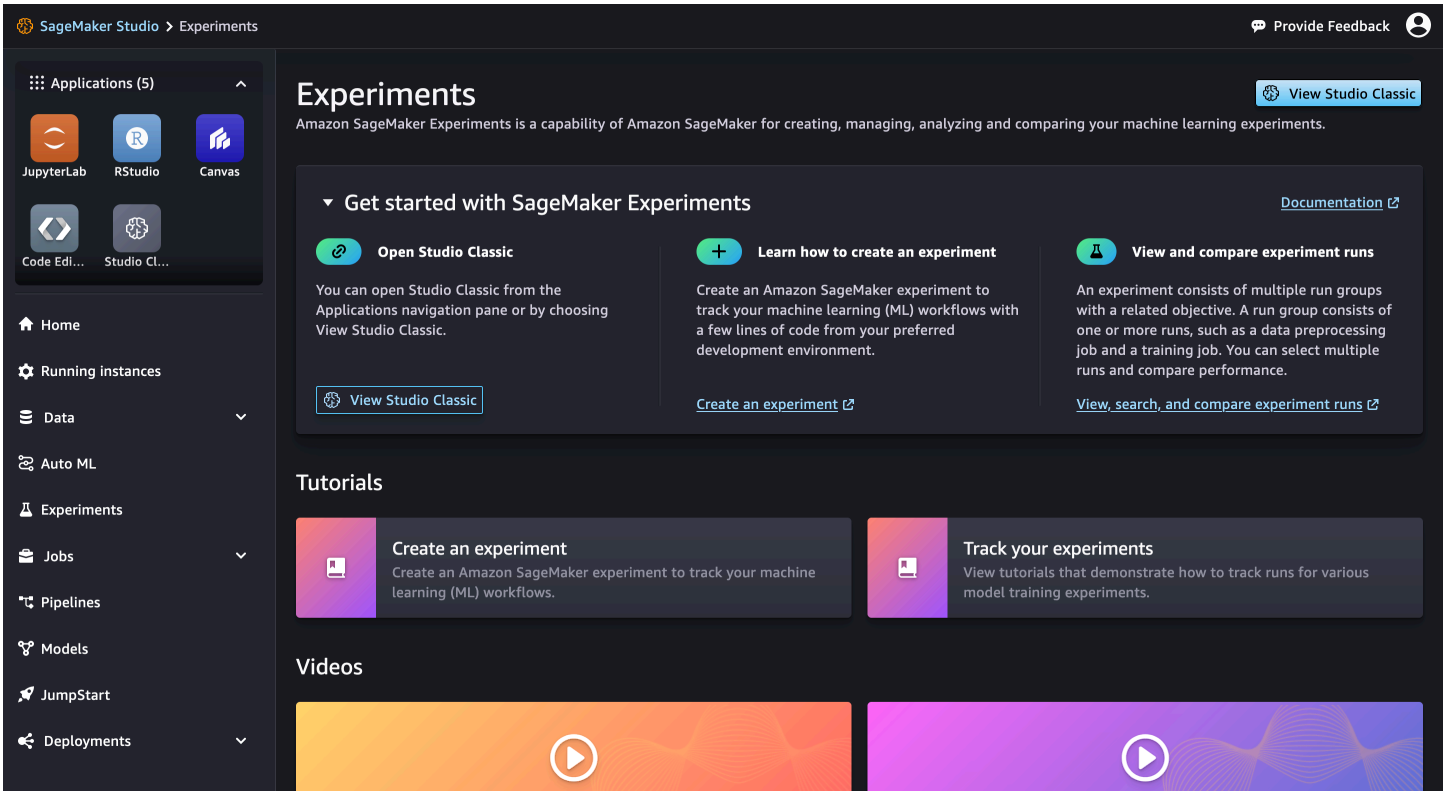
Untuk informasi selengkapnya tentang mode skrip, lihat Menggunakan mode skrip dalam kerangka kerja yang didukung. Anda juga dapat menentukan metrik kustom dalam mode skrip dengan menentukan nama dan ekspresi reguler untuk setiap metrik yang dipantau oleh pekerjaan penyetelan. Lihat Menggunakan algoritma kustom untuk pelatihan untuk informasi selengkapnya.

Lihat eksperimen Anda di Studio

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Untuk melihat eksperimen Anda, Anda harus melakukannya melalui Studio Classic, yang dapat Anda akses melalui pengalaman Studio yang diperbarui.

Di dalam Studio, pilih Eksperimen di panel navigasi sebelah kiri. Kemudian, pilih View Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik. Untuk informasi selengkapnya tentang memulai pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

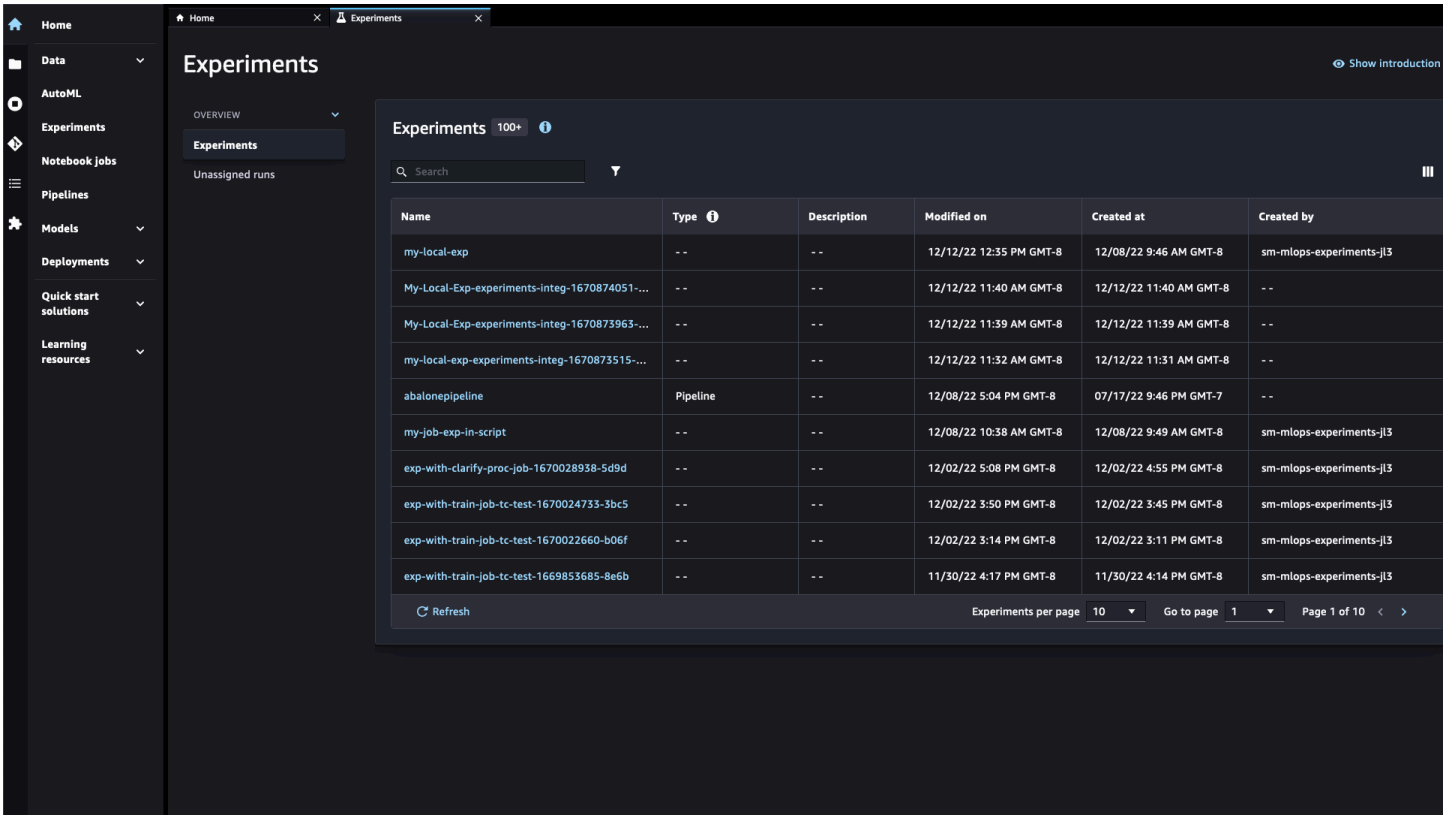


Lihat eksperimen Anda di Studio Classic

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Untuk melihat eksperimen di Studio Classic, di bilah sisi kiri, pilih Eksperimen.

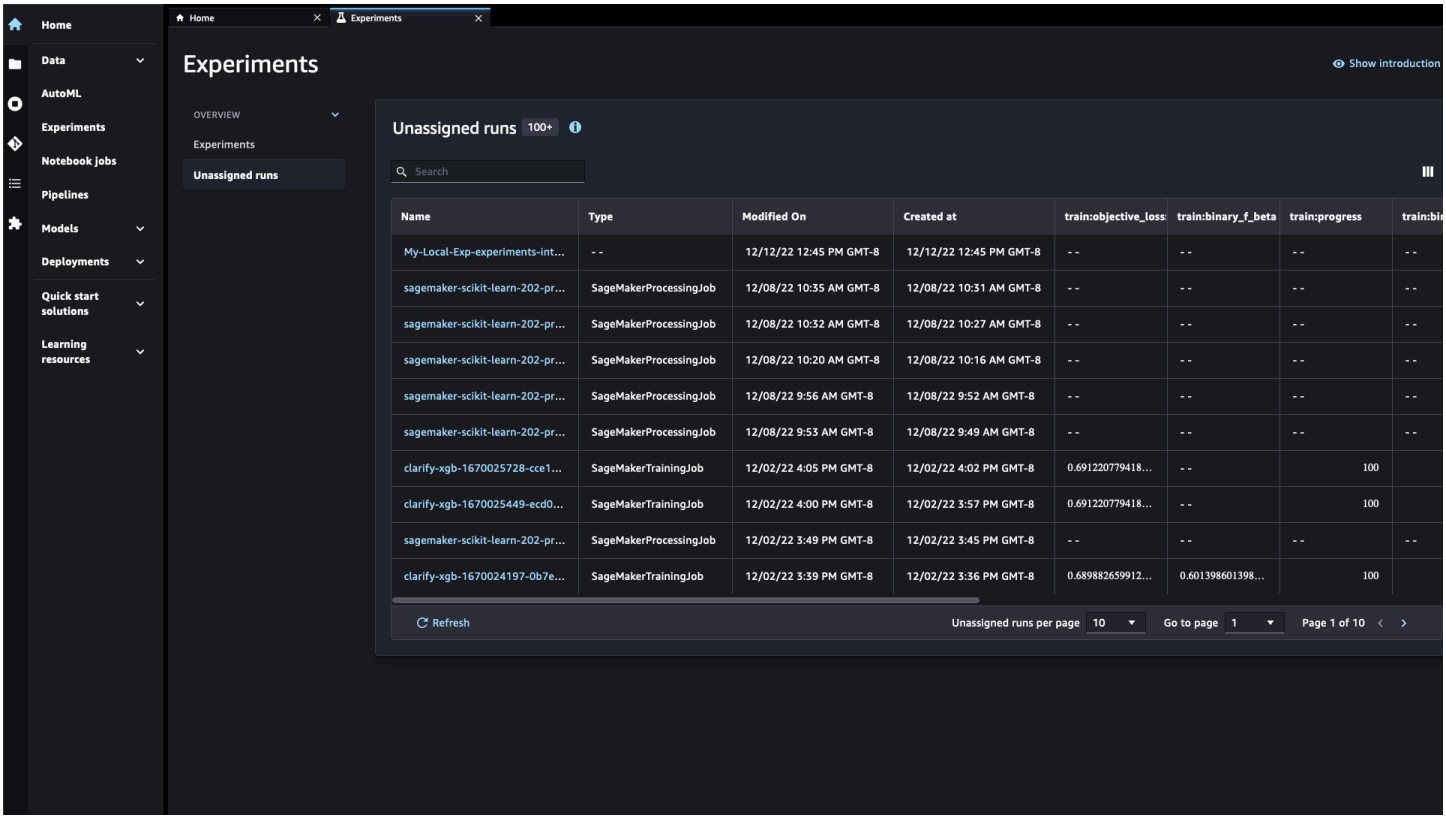
Pilih nama eksperimen untuk melihat semua proses terkait. Mungkin perlu beberapa saat bagi daftar untuk menyegarkan dan menampilkan eksperimen atau eksperimen baru yang dijalankan. Anda dapat mengklik Refresh untuk memperbarui halaman. Daftar eksperimen Anda akan terlihat mirip dengan yang berikut ini:



Untuk melihat proses yang membentuk eksperimen Anda, pilih nama eksperimen. Untuk informasi selengkapnya, lihat Lihat, cari, dan bandingkan percobaan yang dijalankan.

Lihat proses yang tidak ditetapkan

Semua SageMaker pekerjaan, termasuk pekerjaan pelatihan, pekerjaan pemrosesan, dan pekerjaan transformasi, sesuai dengan menjalankan dan membuat Run objek secara default. Jika Anda meluncurkan pekerjaan ini tanpa secara eksplisit mengaitkannya dengan eksperimen, proses yang dihasilkan tidak ditetapkan dan dapat dilihat di bagian Unassigned run pada UI Studio Classic Experiments.



Untuk membersihkan sumber daya yang Anda buat, lihatBersihkan Sumber Daya SageMaker Eksperimen Amazon.

Lihat, cari, dan bandingkan percobaan yang dijalankan

SageMaker Eksperimen Amazon terdiri dari beberapa grup lari dengan tujuan terkait. Grup lari terdiri dari satu atau lebih lari, seperti pekerjaan pra-pemrosesan data dan pekerjaan pelatihan.

Untuk melihat eksperimen Anda, Anda harus melakukannya melalui Studio Classic. Untuk gambaran umum tentang antarmuka pengguna Studio Classic, lihatIkhtisar UI Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Lihat eksperimen dan lari

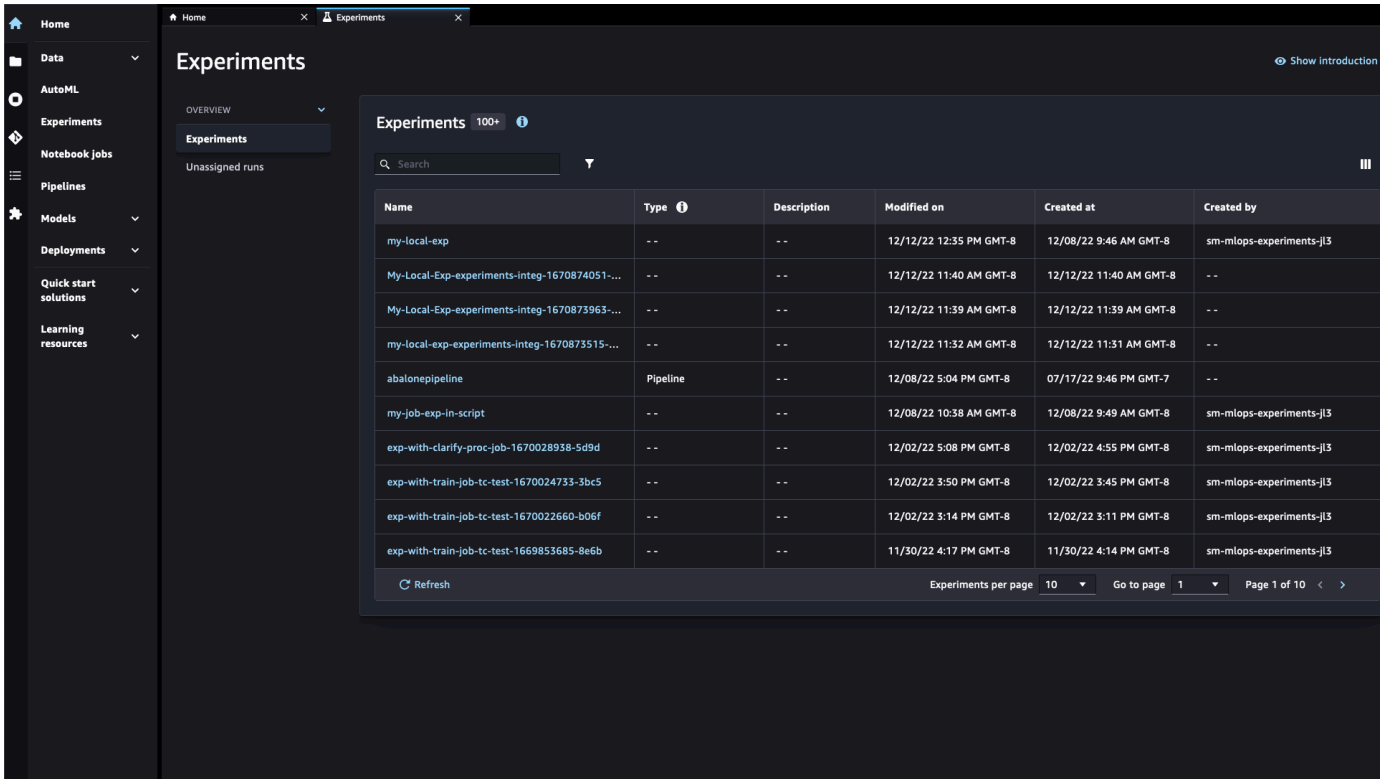
Amazon SageMaker Studio Classic menyediakan browser eksperimen yang dapat Anda gunakan untuk melihat daftar eksperimen dan proses. Anda dapat memilih salah satu entitas ini untuk melihat informasi terperinci tentang entitas atau memilih beberapa entitas untuk perbandingan. Anda dapat memfilter daftar eksperimen berdasarkan nama entitas, jenis, dan tag.

Untuk melihat eksperimen dan menjalankan

1. Untuk melihat eksperimen di Studio Classic, di bilah sisi kiri, pilih Eksperimen.

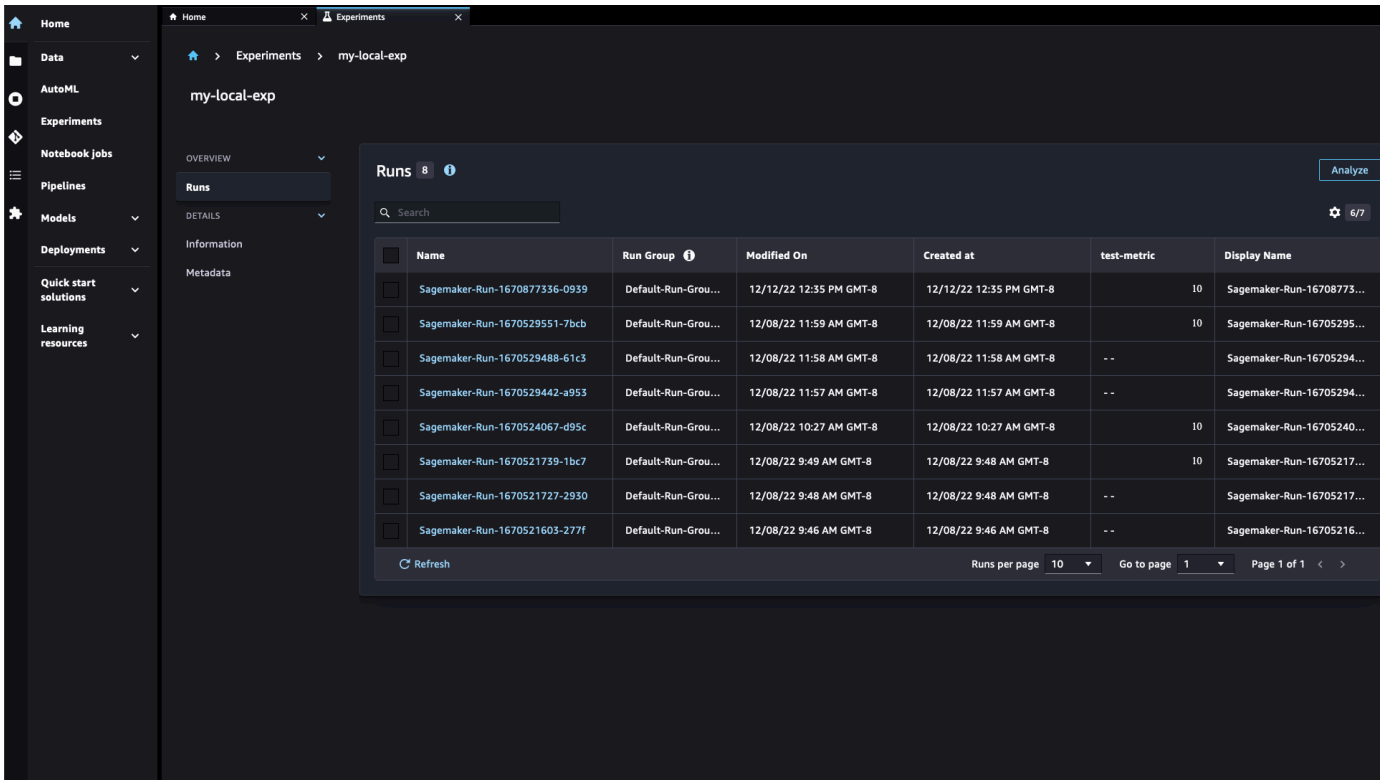
   Pilih nama eksperimen untuk melihat semua proses terkait. Anda dapat mencari eksperimen dengan mengetik langsung ke bilah Pencarian atau memfilter jenis eksperimen. Anda juga dapat memilih kolom mana yang akan ditampilkan dalam daftar percobaan atau jalankan.

   Mungkin perlu beberapa saat bagi daftar untuk menyegarkan dan menampilkan eksperimen atau eksperimen baru yang dijalankan. Anda dapat mengklik Refresh untuk memperbarui halaman. Daftar eksperimen Anda akan terlihat mirip dengan yang berikut ini:



2. Dalam daftar eksperimen, klik dua kali eksperimen untuk menampilkan daftar proses dalam eksperimen.

> **Note**
>
> Eksperimen berjalan yang secara otomatis dibuat oleh SageMaker lowongan dan container akan terlihat di Experiments Studio Classic UI secara default. Untuk menyembunyikan proses yang dibuat oleh SageMaker pekerjaan untuk eksperimen tertentu, pilih ikon pengaturan ( ) dan alihkan Tampilkan pekerjaan.



3. Klik dua kali run untuk menampilkan informasi tentang proses tertentu.

   Di panel Ikhtisar, pilih salah satu judul berikut untuk melihat informasi yang tersedia tentang setiap proses:

   - Metrik — Metrik yang dicatat selama menjalankan.
   - Charts — Bangun bagan Anda sendiri untuk membandingkan proses.
   - Artefak keluaran — Artefak apa pun yang dihasilkan dari percobaan dijalankan dan lokasi artefak di Amazon S3.

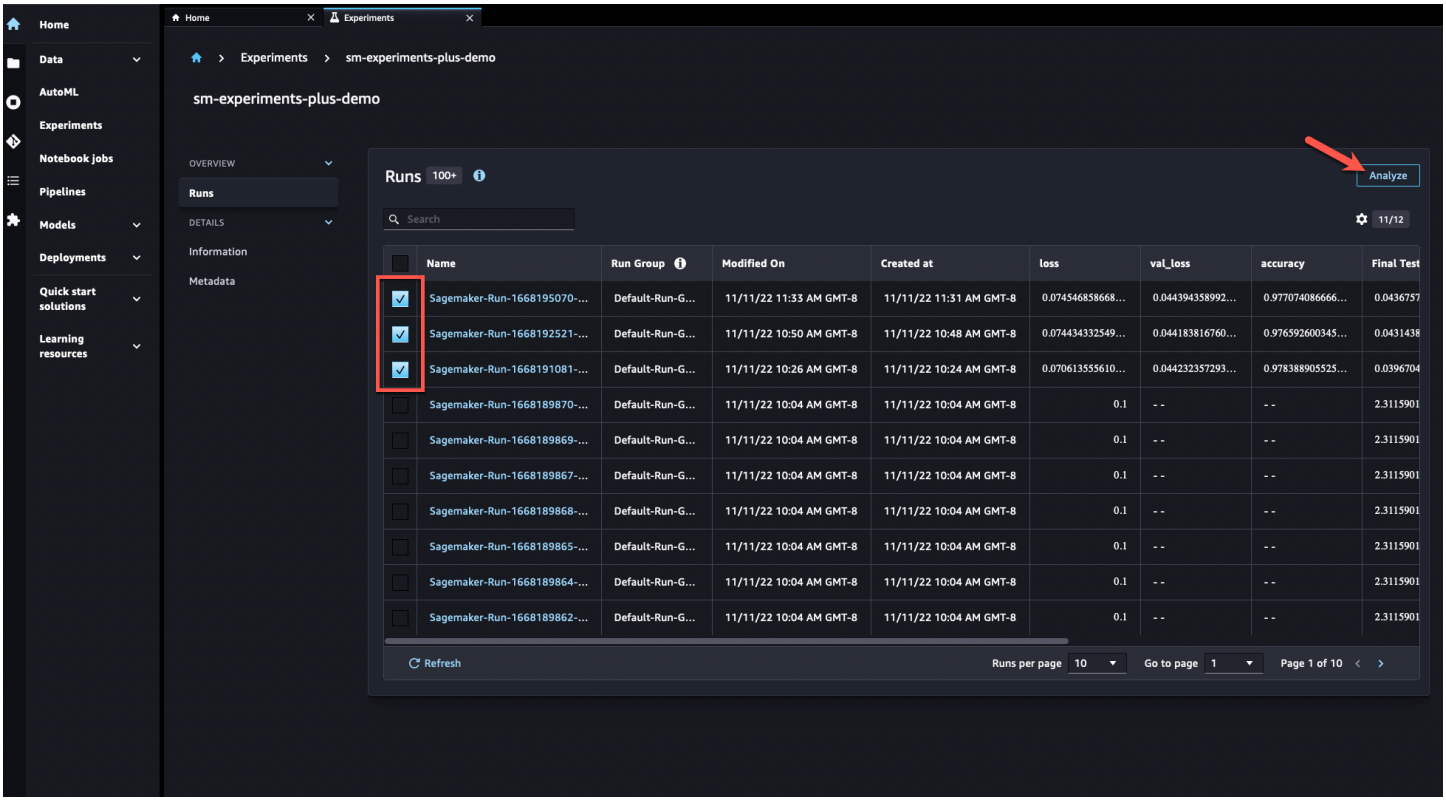
- Laporan bias — Laporan bias pra-pelatihan atau pasca-pelatihan yang dihasilkan menggunakan Clarify.
- Keterjelasan — Laporan penjelasan yang dihasilkan menggunakan Clarify.
- Debug — Daftar aturan debugger dan masalah apa pun yang ditemukan.

Bandingkan dan analisis proses

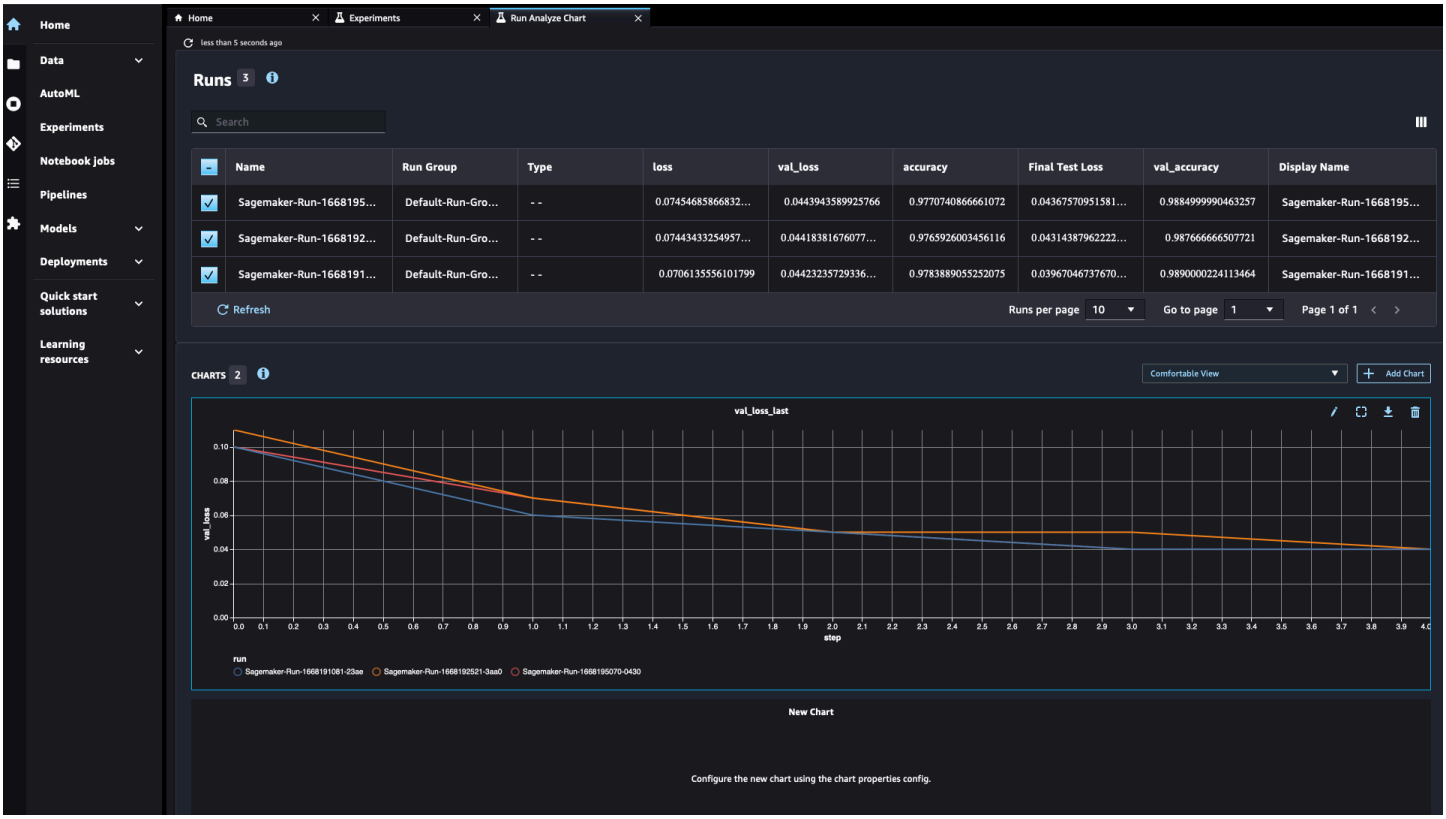
Untuk menganalisis proses eksperimen, pilih eksperimen pilihan Anda di Amazon SageMaker Studio Classic Experiments UI, lalu pilih proses yang ingin Anda bandingkan. Anda harus memilih antara 1 dan 20 run. Setelah Anda memilih run, pilih Analyze di sudut kanan atas.

Untuk membandingkan percobaan berjalan:

1. Setelah menavigasi ke eksperimen pilihan Anda, pilih semua proses yang ingin Anda bandingkan. Anda harus memilih lebih dari 1 dan kurang dari 20 run untuk dianalisis.
2. Pilih Analisis di sudut kanan atas.
3. Visualisasikan metrik komparatif dari beberapa percobaan yang berjalan dalam histogram, diagram garis, plot sebar, atau bagan batang. Untuk menambahkan bagan, pilih Tambahkan Bagan, pilih nilai untuk sumbu bagan Anda, dan pilih Buat.



Anda dapat memperbarui, mengunduh, atau menghapus bagan yang ada.



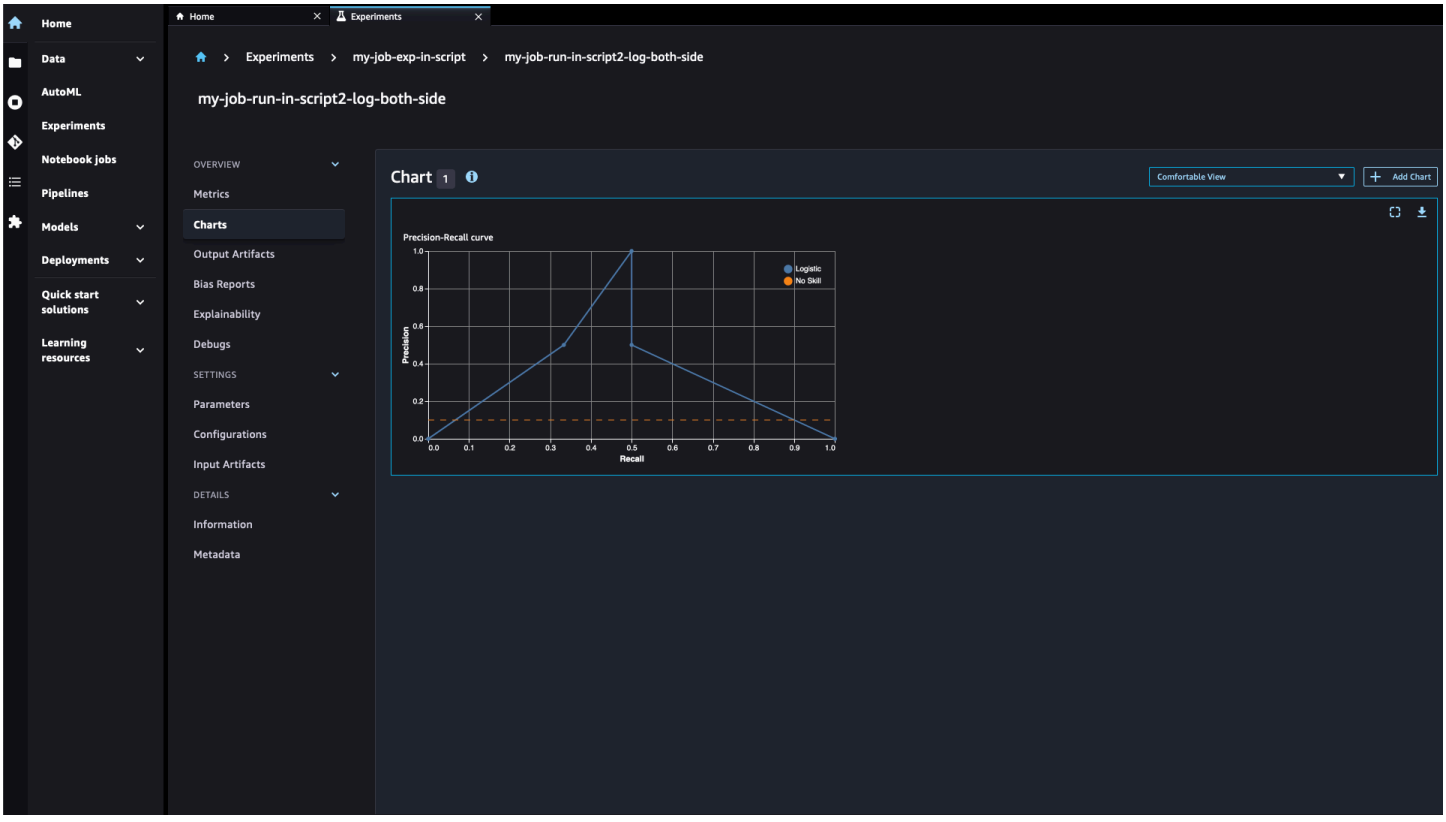
Bagan log

Bagan pencatatan dan visualisasi tersedia untuk model klasifikasi. Anda dapat mencatat matriks kebingungan, karakteristik operasi penerima, atau grafik presisi dan ingatan.

Log dan visualisasikan metrik dengan metode Python SDK berikut:

- `log_confusion_matrix`: Merekam artefak matriks kebingungan yang dapat Anda lihat di bagian Charts dari Run Overview in Studio Classic.
- `log_roc_curve`: Merekam artefak karakteristik operasi penerima yang dapat Anda lihat di bagian Charts pada Run Overview in Studio Classic.
- `log_precision_recall`: Merekam grafik recall presisi yang dapat Anda lihat di bagian Charts pada Run Overview in Studio Classic.

Catatan penarikan presisi yang dicatat secara otomatis membuat bagan yang mirip dengan berikut ini:



SageMaker integrasi

Amazon SageMaker Experiments terintegrasi dengan sejumlah SageMaker fitur. SageMaker Pekerjaan tertentu secara otomatis membuat eksperimen. Anda dapat melihat dan mengelola laporan bias SageMaker Clarify atau tensor keluaran SageMaker Debugger untuk eksperimen tertentu yang dijalankan langsung di UI Eksperimen Klasik Studio.

- Pembuatan eksperimen otomatis
- Laporan bias dan penjelasan
- Debugging

Pembuatan eksperimen otomatis

Amazon SageMaker secara otomatis membuat eksperimen saat menjalankan pekerjaan Autopilot, pekerjaan pengoptimalan hiperparameter (HPO), atau eksekusi Pipeline. Anda dapat melihat eksperimen ini di UI Studio Classic Experiments.

Autopilot

Amazon SageMaker Experiments terintegrasi dengan Amazon SageMaker Autopilot. Saat Anda melakukan pekerjaan Autopilot, SageMaker Eksperimen membuat eksperimen untuk pekerjaan itu serta berjalan untuk setiap kombinasi berbeda dari komponen, parameter, dan artefak run yang tersedia. Anda dapat menemukan proses ini di UI SageMaker Eksperimen dengan memfilter untuk jenis jalankan Autopilot. Untuk informasi selengkapnya, lihat Mengotomatiskan pengembangan model dengan Amazon SageMaker Autopilot.

HPO

Amazon SageMaker Experiments terintegrasi dengan pekerjaan HPO. Pekerjaan HPO secara otomatis membuat SageMaker eksperimen, proses, dan komponen Amazon untuk setiap pekerjaan pelatihan yang diselesaikan. Anda dapat menemukan proses ini di UI SageMaker Eksperimen dengan memfilter untuk HPO tipe run. Untuk informasi selengkapnya, lihat Menyetel Beberapa Algoritma dengan Optimasi Hyperparameter untuk Menemukan Model Terbaik.

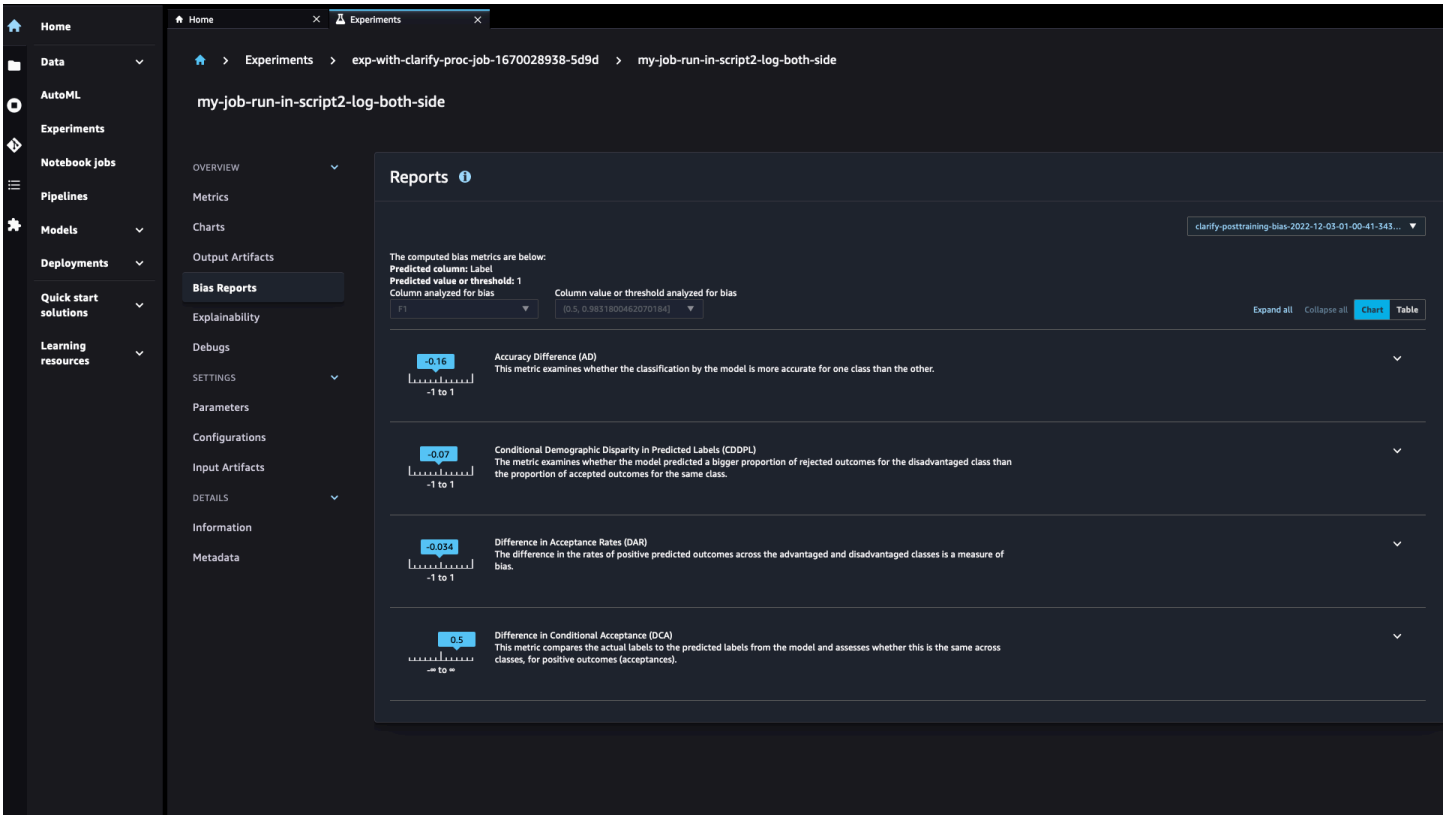
Alur

Amazon SageMaker Model Building Pipelines terintegrasi erat dengan Amazon SageMaker Experiments. Secara default, ketika SageMaker Pipelines membuat dan mengeksekusi pipeline,

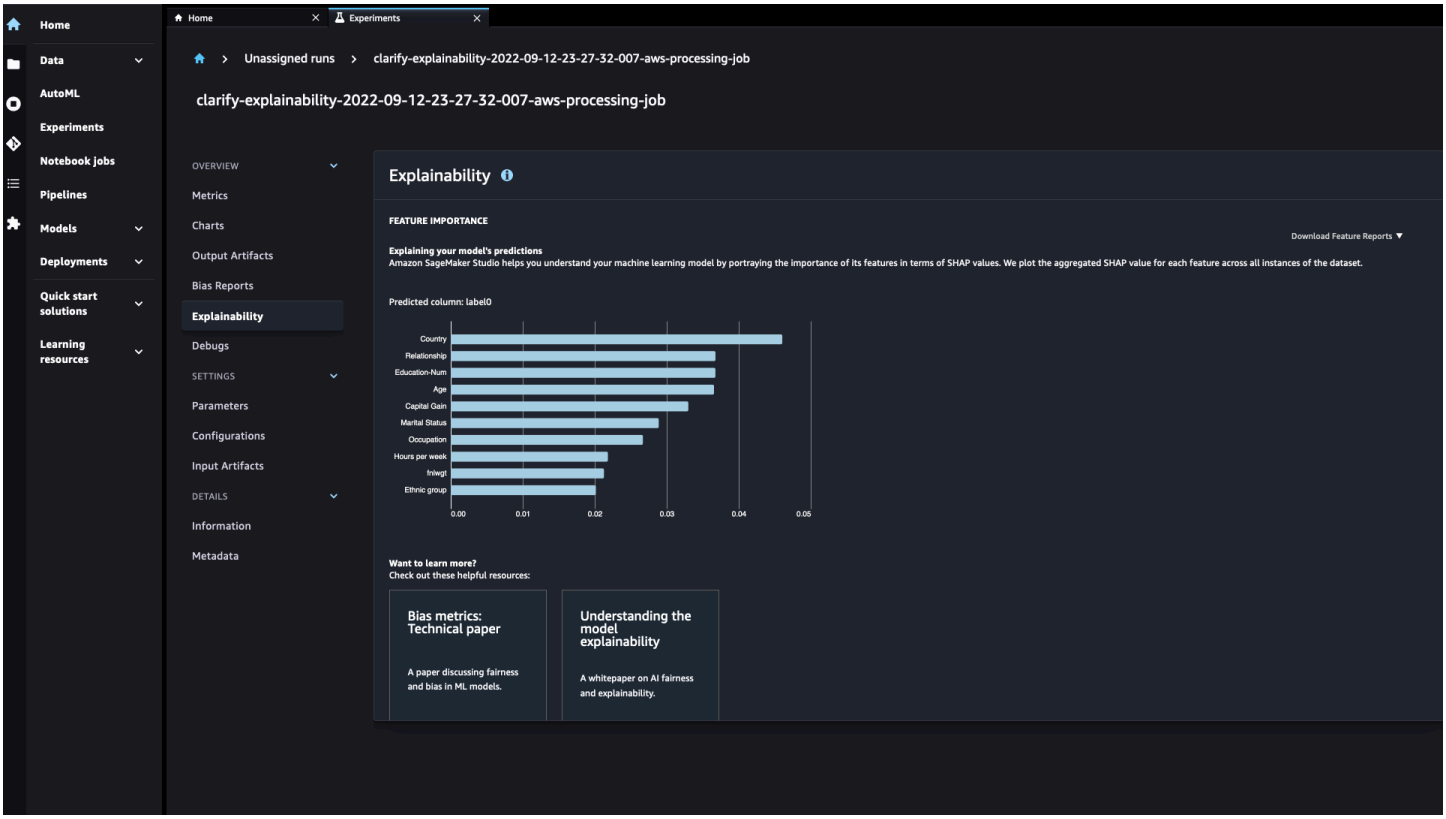
eksperimen, run, dan komponen dibuat jika belum ada. Anda dapat menemukan proses ini di UI SageMaker Eksperimen dengan memfilter untuk Pipelines tipe run. Untuk informasi selengkapnya, lihat Integrasi SageMaker Eksperimen Amazon.

Laporan bias dan penjelasan

Kelola laporan bias SageMaker Clarify dan penjelasan untuk eksperimen yang dijalankan langsung melalui Studio Classic. Untuk melihat laporan, cari dan pilih nama percobaan pilihan Anda di Studio Classic. Pilih laporan Bias untuk melihat laporan bias Klarifikasi apa pun yang terkait dengan percobaan yang dijalankan.



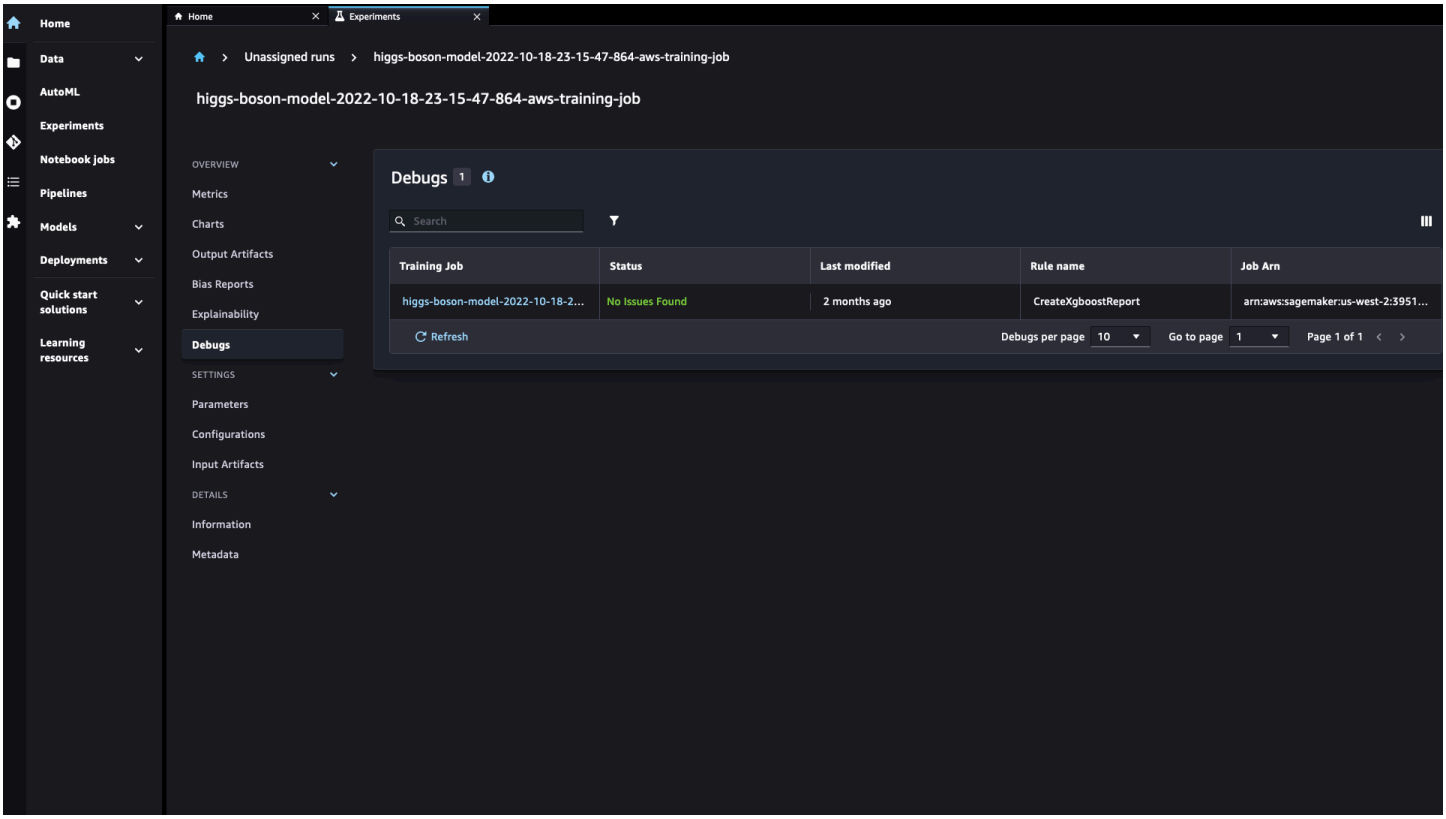
Pilih Penjelasan untuk melihat laporan klarifikasi apa pun yang terkait dengan eksperimen yang dijalankan.



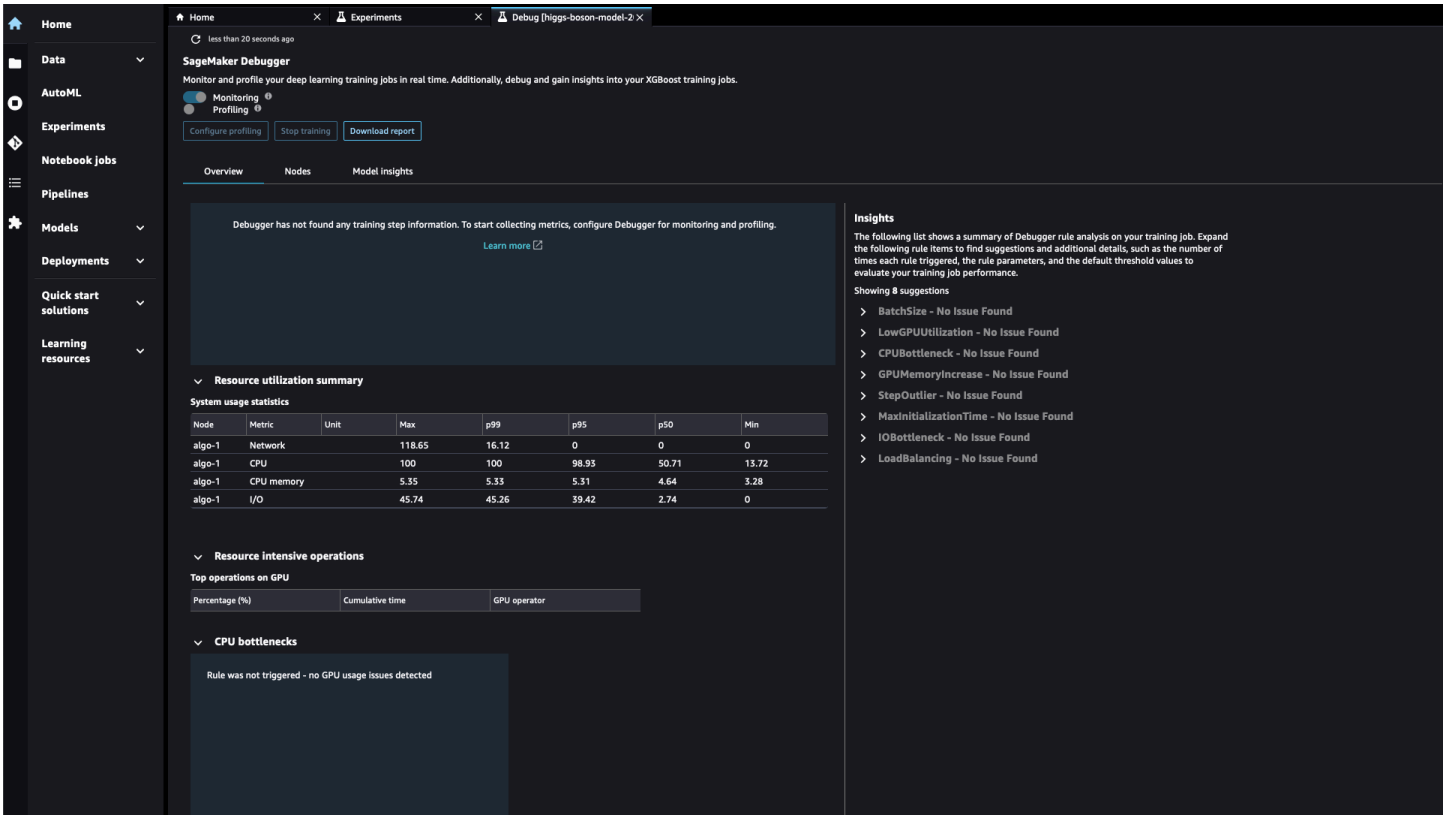
Anda dapat menghasilkan laporan bias pra-pelatihan atau pasca-pelatihan yang menganalisis bias dalam kumpulan data atau prediksi model menggunakan label dan metrik bias dengan Clarify. SageMaker Anda juga dapat menggunakan SageMaker Clarify untuk menghasilkan laporan penjelasan yang mendokumentasikan perilaku model untuk sampel data global atau lokal. Untuk informasi selengkapnya, lihat Amazon SageMaker Clarify Detection Bias and Model Explainability.

Debugging

Anda dapat men-debug progres pelatihan model dengan Amazon SageMaker Debugger dan melihat tensor keluaran debug di UI Studio Classic Experiments. Pilih nama run yang terkait dengan laporan Debugger dan pilih Debugger.



Kemudian, pilih nama pekerjaan pelatihan untuk melihat dasbor Amazon SageMaker Debugger terkait.



Untuk informasi selengkapnya, lihat Mendebug Pekerjaan Pelatihan Menggunakan Amazon SageMaker Debugger.

Contoh notebook untuk Eksperimen Amazon SageMaker

Tutorial berikut menunjukkan cara melacak lari untuk berbagai eksperimen pelatihan model. Anda dapat melihat eksperimen yang dihasilkan di Studio Classic setelah menjalankan notebook. Untuk membersihkan sumber daya yang dibuat oleh notebook, lihatBersihkan Sumber Daya SageMaker Eksperimen Amazon. Untuk tutorial yang menampilkan fitur tambahan Studio, lihatTur Klasik Amazon SageMaker Studio.

Lacak eksperimen di lingkungan notebook

Untuk mempelajari lebih lanjut tentang melacak eksperimen di lingkungan buku catatan, lihat contoh buku catatan berikut:

- Lacak eksperimen sambil melatih model Keras secara lokal
- Lacak eksperimen saat melatih model Pytorch secara lokal atau di buku catatan Anda

Lacak bias dan penjelasan untuk eksperimen Anda dengan Clarify SageMaker

Untuk step-by-step panduan tentang melacak bias dan penjelasan untuk eksperimen Anda, lihat contoh buku catatan berikut:

- Keadilan dan Keterjelasan dengan Clarify SageMaker

Lacak eksperimen untuk pekerjaan SageMaker pelatihan menggunakan mode skrip

Untuk informasi selengkapnya tentang melacak eksperimen untuk pekerjaan SageMaker pelatihan, lihat contoh buku catatan berikut:

- Jalankan SageMaker Eksperimen dengan Paralel Data Terdistribusi Pytorch - Klasifikasi Digit Tulisan Tangan MNIST
- Lacak eksperimen sambil melatih model Pytorch dengan Training Job SageMaker
- Latih TensorFlow model dengan pekerjaan SageMaker pelatihan dan lacak menggunakan SageMaker Eksperimen

Pantau metrik pelatihan eksperimen dengan AWS CloudTrail

Metrik pelatihan untuk SageMaker Eksperimen Amazon terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan. CloudTrail menangkap semua panggilan API untuk `BatchPutMetrics` sebagai peristiwa. SageMaker secara otomatis memanggil `BatchPutMetrics` saat Anda membuat eksperimen yang dijalankan menggunakan SageMaker SDK untuk Python. AWS CloudTrailmenangkap data yang terkait dengan panggilan untuk jenis `AWS::SageMaker::ExperimentTrialComponent` sumber daya.

> **Note**
>
> Dalam UI Eksperimen Klasik Studio, uji coba disebut sebagai grup lari dan komponen uji coba disebut sebagai run.

Saat membuat eksperimen dijalankan, Anda juga dapat mengonfigurasi pengiriman CloudTrail peristiwa yang berkelanjutan ke bucket Amazon S3. Gunakan CloudTrail untuk memantau semua metrik pelatihan yang dicerna untuk menjalankan eksperimen, termasuk informasi seperti nama

metrik, langkah pelatihan metrik yang direkam, stempel waktu, dan nilai metrik. CloudTrail peristiwa juga mencakup eksperimen menjalankan ARN, ID akun yang membuat run, dan jenis sumber daya, yang seharusnya. `AWS::SageMaker::ExperimentTrialComponent`

Untuk memantau panggilan `BatchPutMetrics` API sebagai CloudTrail peristiwa, Anda harus terlebih dahulu menyiapkan pencatatan aktivitas API bidang data di CloudTrail. Lihat Pencatatan log peristiwa data untuk jejak untuk informasi selengkapnya. Untuk kontrol terperinci atas panggilan API mana yang ingin Anda log dan bayar secara selektif, Anda dapat memfilter CloudTrail peristiwa berdasarkan jenis sumber daya. Tentukan `AWS::SageMaker::ExperimentTrialComponent` sebagai tipe sumber daya untuk memantau panggilan ke `BatchPutMetrics` API. Untuk informasi selengkapnya, lihat DataResourcedi referensi AWS CloudTrail API. Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Untuk penjelasan mendalam tentang cara SageMaker kerja AmazonAWS CloudTrail, lihatLog Panggilan SageMaker API Amazon dengan AWS CloudTrail.

Berikut ini adalah contoh CloudTrail peristiwa untuk metrik pelatihan dalam percobaan dijalankan:

```
{
  ...
  "eventTime": "2022-12-14T21:53:41Z",
  "eventSource": "metrics-sagemaker.amazonaws.com",
  "eventName": "BatchPutMetrics",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-cli/2.7.25 Python/3.9.11 Linux/5.4.214-134.408.amzn2int.x86_64 exe/
x86_64.amzn.2 prompt/off command/sm-metrics.batch-put-metrics",
  "requestParameters": {
    "trialComponentName": "trial-component-name",
    "metricData": [
      {
        "metricName": "foo",
        "timestamp": 1670366870000,
        "step": 101,
        "value": 0.9
      }
    ]
  },
  ...
  "resources": [
    {
      "accountId": "abcdef01234567890",
```

```
      "type": "AWS::SageMaker::ExperimentTrialComponent",
      "ARN": "arn:aws:sagemaker:us-east-1:1234567890abcdef0:experiment-trial-component/
trial-component-name"
    }
  ],
  ...
}
```

Bersihkan Sumber Daya SageMaker Eksperimen Amazon

Untuk menghindari biaya yang tidak perlu, hapus sumber daya SageMaker Eksperimen Amazon yang tidak lagi Anda perlukan. Anda tidak dapat menghapus sumber daya Eksperimen melalui SageMaker Management Console atau Amazon SageMaker Studio Classic UI. Topik ini menunjukkan cara membersihkan sumber daya ini menggunakan SageMaker Python SDK, Boto3, dan Experiments SDK.

Topik

- Bersihkan Menggunakan SageMaker Python SDK (Disarankan)
- Bersihkan Menggunakan Python SDK (Boto3)
- Bersihkan Menggunakan SDK Eksperimen

Bersihkan Menggunakan SageMaker Python SDK (Disarankan)

Untuk membersihkan menggunakan SageMaker Python SDK

```
from sagemaker.experiments.experiment import Experiment

exp = Experiment.load(experiment_name=experiment_name, sagemaker_session=sm_session)
exp._delete_all(action="--force")
```

Bersihkan Menggunakan Python SDK (Boto3)

Untuk membersihkan menggunakan Boto 3

```
import boto3
sm = boto3.Session().client('sagemaker')
```

Tentukan cleanup_boto3

```
def cleanup_boto3(experiment_name):
    trials = sm.list_trials(ExperimentName=experiment_name)['TrialSummaries']
    print('TrialNames:')
    for trial in trials:
        trial_name = trial['TrialName']
        print(f"\n{trial_name}")

        components_in_trial = sm.list_trial_components(TrialName=trial_name)
        print('\tTrialComponentNames:')
        for component in components_in_trial['TrialComponentSummaries']:
            component_name = component['TrialComponentName']
            print(f"\t{component_name}")
            sm.disassociate_trial_component(TrialComponentName=component_name,
 TrialName=trial_name)
            try:
                # comment out to keep trial components
                sm.delete_trial_component(TrialComponentName=component_name)
            except:
                # component is associated with another trial
                continue
            # to prevent throttling
            time.sleep(.5)
        sm.delete_trial(TrialName=trial_name)
    sm.delete_experiment(ExperimentName=experiment_name)
    print(f"\nExperiment {experiment_name} deleted")
```

Panggil cleanup_boto3

```
# Use experiment name not display name
experiment_name = "experiment-name"
cleanup_boto3(experiment_name)
```

Bersihkan Menggunakan SDK Eksperimen

Untuk membersihkan menggunakan SDK Eksperimen

```
import sys
!{sys.executable} -m pip install sagemaker-experiments
```

```
import time
```

```
from smexperiments.experiment import Experiment
from smexperiments.trial import Trial
from smexperiments.trial_component import TrialComponent
```

Tentukan cleanup_sme_sdk

```
def cleanup_sme_sdk(experiment):
    for trial_summary in experiment.list_trials():
        trial = Trial.load(trial_name=trial_summary.trial_name)
        for trial_component_summary in trial.list_trial_components():
            tc = TrialComponent.load(
                trial_component_name=trial_component_summary.trial_component_name)
            trial.remove_trial_component(tc)
            try:
                # comment out to keep trial components
                tc.delete()
            except:
                # tc is associated with another trial
                continue
            # to prevent throttling
            time.sleep(.5)
        trial.delete()
        experiment_name = experiment.experiment_name
    experiment.delete()
    print(f"\nExperiment {experiment_name} deleted")
```

Panggil cleanup_sme_sdk

```
experiment_to_cleanup = Experiment.load(
    # Use experiment name not display name
    experiment_name="experiment-name")

cleanup_sme_sdk(experiment_to_cleanup)
```

SDK tambahan yang didukung

Important

Mulai v2.123.0, SageMaker Eksperimen sekarang sepenuhnya terintegrasi dengan SageMaker Python SDK dan Anda tidak perlu lagi menggunakan SDK Eksperimen terpisah.

SageMaker Sebaiknya buat eksperimen dengan `sagemaker.experiments.run` bukan `smexperiments` modul berikut.

Bagian berikut menjelaskan cara membuat SageMaker Eksperimen dengan SDK SageMaker Eksperimen.

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Membuat SageMaker Eksperimen Amazon dengan SDK SageMaker Eksperimen

Buat SageMaker eksperimen Amazon untuk melacak SageMaker pelatihan, pemrosesan, dan transformasi pekerjaan Anda.

Prosedur berikut menunjukkan kepada Anda cara membuat SageMaker eksperimen untuk SageMaker pelatihan, pemrosesan, atau transformasi pekerjaan. Langkah-langkah yang diberi label sebagai (Studio Classic) menjelaskan cara melihat eksperimen di Amazon SageMaker Studio Classic. Anda tidak perlu menjalankan eksperimen di Studio Classic untuk melihat eksperimen di Studio Classic.

1. Impor `sys` modul untuk menginstal SDK.

   ```
   import sys
   ```

2. (Opsional) Amazon SageMaker Python SDK, sudah diinstal sebelumnya di Amazon SageMaker Studio Classic. Jika Anda berencana untuk menjalankan kode Anda di luar Studio Classic, instal SageMaker Python SDK.

   ```
   !{sys.executable} -m pip install sagemaker
   ```

3. Instal SDK Python SageMaker Eksperimen.

   ```
   !{sys.executable} -m pip install sagemaker-experiments
   ```

4. Impor modul.

```
import time
from time import strftime

import sagemaker

from smexperiments.experiment import Experiment
from smexperiments.trial import Trial
from smexperiments.trial_component import TrialComponent
from smexperiments.tracker import Tracker
```

5. Dapatkan peran eksekusi dan buat SageMaker sesi.

```
role = sagemaker.get_execution_role()
sm_sess = sagemaker.session.Session()
```

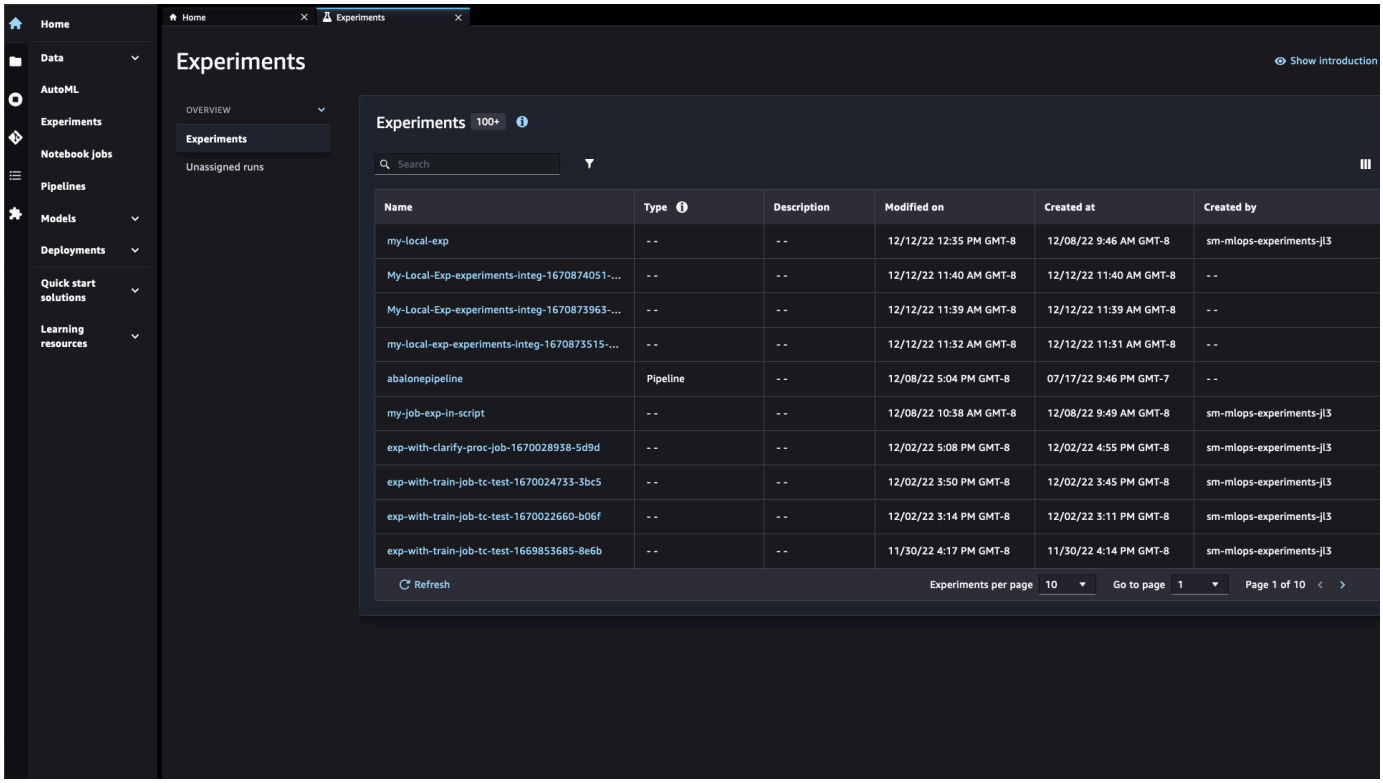
6. Buat SageMaker eksperimen. Nama eksperimen harus unik di akun Anda.

> **Note**
>
> Parameter `tags` bersifat opsional. Anda dapat mencari tag menggunakan Studio Classic, SageMaker konsol, dan SDK. Tag juga dapat diterapkan pada komponen uji coba dan uji coba.

```
create_date = strftime("%Y-%m-%d-%H-%M-%S")
demo_experiment = Experiment.create(experiment_name = "DEMO-
{}".format(create_date),
                                    description = "Demo experiment",
                                    tags = [{'Key': 'demo-experiments', 'Value':
 'demo1'}])
```

7. (Studio Classic) Untuk melihat eksperimen di Amazon SageMaker Studio Classic, di bilah sisi kiri, pilih Eksperimen.

   Setelah kode berjalan, daftar eksperimen berisi eksperimen baru. Mungkin perlu beberapa saat bagi daftar untuk menyegarkan dan menampilkan eksperimen. Filter pada tag percobaan juga ditampilkan. Hanya eksperimen yang memiliki tag yang cocok yang ditampilkan. Daftar Anda akan terlihat mirip dengan yang berikut ini:



8. Buat percobaan untuk percobaan. Nama uji coba harus unik di akun Anda.

```
demo_trial = Trial.create(trial_name = "DEMO-{}".format(create_date),
                          experiment_name = demo_experiment.experiment_name,
                          tags = [{'Key': 'demo-trials', 'Value': 'demo1'}])
```

9. Buat komponen uji coba sebagai bagian dari uji coba. Komponen percobaan adalah SageMaker pekerjaan.

   Tambahkan ExperimentConfigparameter ke metode yang sesuai. SageMaker Pekerjaan yang tercantum dalam tabel berikut didukung.

Pekerjaan	SageMaker Metode Python SDK	Metode Boto3
Pelatihan	Estimator.fit	CreateTrainingJob
Pemrosesan	Prosesor.run	CreateProcessingJob
Transformasi	Transformer.Transform	CreateTransformJob

Contoh berikut adalah untuk pekerjaan pelatihan. `Tags`Parameter menambahkan tag ke komponen percobaan. `ExperimentName`tidak ditentukan karena uji coba dikaitkan dengan eksperimen saat uji coba dibuat pada langkah sebelumnya.

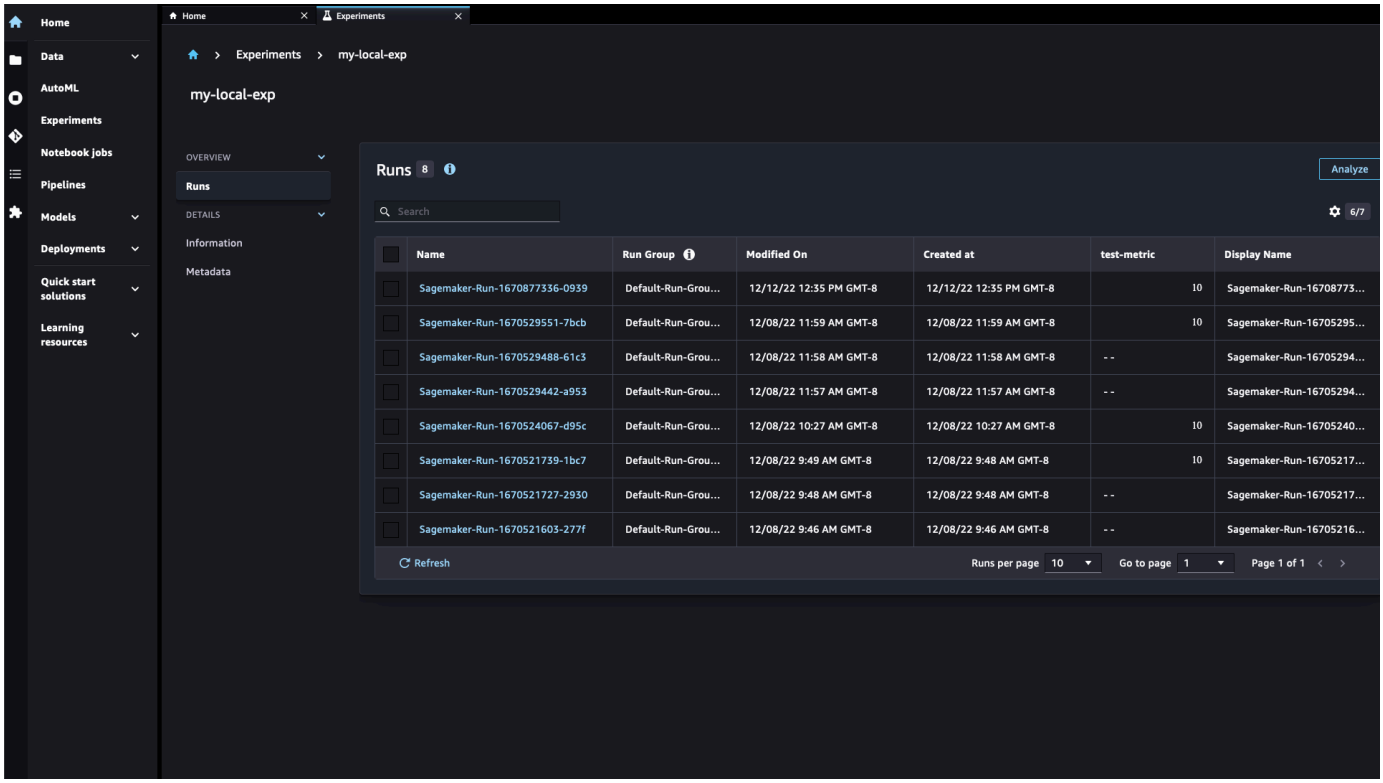
Menggunakan SageMaker Python SDK

```
sagemaker.estimator.Estimator(
    ...,
    sagemaker_session = sm_sess,
    tags = [{'Key': 'demo-jobs', 'Value': 'demo2'}])

estimator.fit(
    ...,
    experiment_config = {
        # "ExperimentName"
        "TrialName" : demo_trial.trial_name,
        "TrialComponentDisplayName" : "TrainingJob",
    })
```

Menggunakan Boto3

```
create_training_job(
    ...,
    "ExperimentConfig": {
        # "ExperimentName"
        "TrialName" : demo_trial.trial_name,
        "TrialComponentDisplayName" : "TrainingJob",
    },
    "Tags": [{'Key': 'demo-jobs', 'Value': 'demo2'}])
```

10. (Studio Klasik) Dalam daftar eksperimen, klik dua kali eksperimen untuk menampilkan daftar uji coba dalam percobaan. Di UI Studio Classic, uji coba disebut sebagai grup lari dan komponen uji coba disebut sebagai run. Daftar Anda akan terlihat mirip dengan yang berikut ini:



11. (Studio Classic) Untuk melihat informasi tentang eksperimen, uji coba, dan pekerjaan (komponen uji coba), lihatLihat, cari, dan bandingkan percobaan yang dijalankan.

Untuk membersihkan sumber daya yang Anda buat, lihatBersihkan Sumber Daya SageMaker Eksperimen Amazon.

FAQ Eksperimen

Lihat item FAQ berikut untuk jawaban atas pertanyaan umum tentang SageMaker Eksperimen.

T. Apa metode yang direkomendasikan untuk membuat eksperimen?

J: Eksperimen adalah kumpulan lari yang bertujuan menemukan model terbaik untuk memecahkan masalah. Untuk menginisialisasi proses dalam eksperimen, gunakan kelas SageMaker Python `Run` SDK. Untuk contoh lainnya, lihat Buat SageMaker Eksperimen Amazon.

T. Dapatkah saya membuat eksperimen menggunakan mode SageMaker skrip?

Ya. Anda dapat membuat eksperimen menggunakan mode SageMaker skrip. Di notebook Jupyter atau file Python yang Anda gunakan untuk menentukan estimator Anda, inisialisasi proses menggunakan kelas. `Run` Dalam proses, luncurkan estimator dengan skrip titik masuk kustom Anda. Dalam skrip titik masuk itu, gunakan `load_run` metode untuk menginisialisasi proses yang

Anda tentukan dalam skrip titik masuk dan catat metrik Anda. Untuk contoh mendalam, lihatLacak eksperimen untuk pekerjaan SageMaker pelatihan menggunakan mode skrip.

T. SageMaker Pekerjaan apa yang secara otomatis membuat eksperimen?

SageMaker Pekerjaan Hyperparameter Optimzation (HPO) (juga dikenal sebagai pekerjaan tuning) secara otomatis membuat eksperimen untuk melacak semua pekerjaan pelatihan yang diluncurkan selama pencarian hyperparameter. Semua SageMaker pekerjaan lain membuat proses yang tidak ditetapkan kecuali diluncurkan dari dalam eksperimen.

T. SageMaker Pekerjaan seperti apa yang bisa saya buat eksperimen?

Anda dapat menggunakan SageMaker Eksperimen untuk melacak metrik dari pekerjaan pelatihan, memproses pekerjaan, dan mengubah pekerjaan.

T. Mengapa saya melihat eksperimen dan berjalan di UI Experiments Studio Classic yang tidak saya buat menggunakan SageMaker Python SDK?

Eksperimen berjalan yang secara otomatis dibuat oleh SageMaker lowongan dan container akan terlihat di Experiments Studio Classic UI secara default. Untuk menyembunyikan proses yang dibuat oleh SageMaker pekerjaan untuk eksperimen tertentu, pilih ikon pengaturan (⚙) dan alihkan Tampilkan pekerjaan.

Q. Apakah SDK SageMaker Eksperimen masih didukung?

Ya, SDK SageMaker Eksperimen masih didukung. Namun, pada v2.123.0, SageMaker Eksperimen sepenuhnya terintegrasi dengan Python SageMaker SDK. Sebaiknya gunakan SageMaker Python SDK untuk membuat eksperimen dan menjalankan. Untuk informasi selengkapnya, lihat Buat SageMaker Eksperimen Amazon.

T. Dapatkah saya menggunakan pelatihan terdistribusi dengan eksperimen saya?

J: Ya. Namun, metrik untuk pelatihan terdistribusi hanya dapat dicatat pada tingkat epoch. Pastikan Anda hanya mencatat metrik yang dihasilkan oleh node pemimpin, seperti yang ditunjukkan pada contoh berikut:

```
...
if rank == 0:
        test_loss, correct, target, pred = test(model, test_loader, device, tracker)
        logger.info(
```

```
            "Test Average loss: {:.4f}, Test Accuracy: {:.0f}%;\n".format(
                test_loss, test_accuracy)
            )
        )
        run.log_metric(name = "train_loss", value = loss.item(), step = epoch)
        run.log_metric(name = "test_loss", value = test_loss, step = epoch)
        run.log_metric(name = "test_accuracy", value = test_accuracy, step = epoch)
        ...
```

Untuk informasi selengkapnya, lihat buku catatan contoh Jalankan SageMaker Eksperimen dengan Pytorch Distributed Data Parallel - MNIST Handwritten Digit Classification.

T. Apa yang dimaksud dengan unassigned run?

J: Semua pekerjaan di SageMaker (pekerjaan pelatihan, pekerjaan pemrosesan, transformasi pekerjaan) sesuai dengan lari. Saat meluncurkan pekerjaan ini, `TrialComponents` dibuat secara default. `TrialComponents`memetakan langsung ke run. Jika pekerjaan ini diluncurkan tanpa secara eksplisit dikaitkan dengan eksperimen atau proses, pekerjaan tersebut dibuat sebagai proses yang tidak ditetapkan.

T. Apakah saya harus meneruskan konteks percobaan lari ke skrip pelatihan saat menjalankan pekerjaan SageMaker pelatihan?

J: Ya. Anda perlu memuat konteks run ke dalam skrip pelatihan, bersama dengan informasi SageMaker sesi.

```
from sagemaker.session import Session
from sagemaker.experiments.run import load_run

session = Session(boto3.session.Session(region_name=args.region))

with load_run(sagemaker_session=session) as run:
    run.log_parameters(
        {"num_train_samples": len(train_set.data), "num_test_samples":
 len(test_set.data)}
    )
```

T. Bagaimana cara menambahkan proses baru ke analisis eksperimen?

J: Jika Anda sudah membuat perbandingan untuk eksperimen dan ingin menambahkan proses baru untuk dianalisis, pilih semua proses dari analisis sebelumnya serta proses baru dan pilih Analisis. Jika Anda tidak melihat proses baru Anda di halaman analisis yang dihasilkan, segarkan browser

Studio Classic. Perhatikan bahwa menyegarkan browser Studio Classic dapat memengaruhi tab terbuka lainnya.

Cari Menggunakan SageMaker Konsol Amazon dan API

Mengembangkan model pembelajaran mesin biasanya membutuhkan eksperimen ekstensif dengan kumpulan data, algoritma, dan nilai hyperparameter yang berbeda. Untuk mengelola hingga ribuan eksperimen model pembelajaran mesin, gunakan kemampuan pencarian di SageMaker.

Anda dapat menggunakan SageMaker pencarian untuk:

- Atur, temukan, dan evaluasi pekerjaan pelatihan menggunakan properti, hiperparameter, metrik kinerja, atau metadata apa pun.
- Temukan model berkinerja terbaik dengan meninjau pekerjaan pelatihan dan metrik model, seperti kehilangan pelatihan atau akurasi validasi.
- Lacak garis keturunan model ke pekerjaan pelatihan dan sumber daya terkaitnya, seperti kumpulan data pelatihan.

Topik ini mencakup pencarian dari SageMaker konsol dan SageMaker API.

Topik

Mengatur, Menemukan, dan Mengevaluasi Pekerjaan Pelatihan (Konsol)

Untuk mengatur pekerjaan pelatihan, tetapkan satu atau lebih tag untuk mereka.

Untuk menemukan pekerjaan, model, atau sumber daya pelatihan tertentu, gunakan pelacakan model untuk mencari kata kunci yang ditetapkan untuk item yang dapat dicari. Item yang dapat dicari termasuk pekerjaan pelatihan, model, hiperparameter, metadata, tag, dan URL. Untuk menyempurnakan hasil pelacakan, Anda dapat mencari menggunakan beberapa kriteria.

Untuk memilih model terbaik untuk penerapan, evaluasi bagaimana semua model dilakukan terhadap satu atau beberapa metrik. Anda dapat menggunakan hasil pelacakan model untuk membuat daftar, mengurutkan, dan mengevaluasi kinerja model dalam eksperimen Anda.

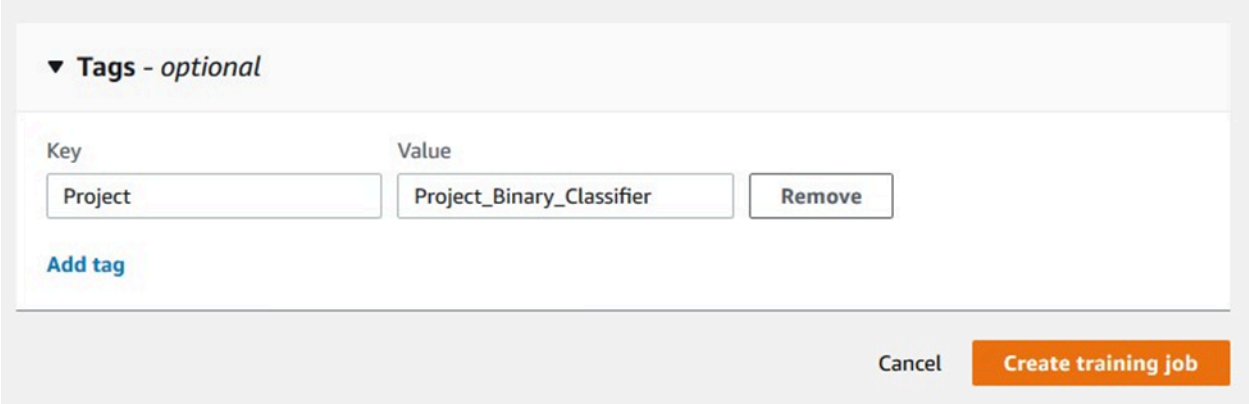
Topik

Menggunakan Tag untuk Melacak Pekerjaan Pelatihan (Konsol)

Untuk mengelompokkan pekerjaan pelatihan, buat tag dengan kunci deskriptif dan nilai. Misalnya, buat kunci tag untuk: proyek, pemilik, pelanggan, dan industri.

Tambahkan tag ke pekerjaan pelatihan (konsol)

1. Buka SageMaker konsol Amazon.
2. Di panel navigasi, pilih Pekerjaan pelatihan dan Buat pekerjaan pelatihan.
3. Gulir ke bagian bawah halaman dan masukkan kunci dan nilai untuk tag.



4. Untuk menambahkan tag lain, pilih Tambah tag, dan tambahkan pasangan nilai kunci lainnya.

Temukan Pekerjaan Pelatihan (Konsol)

Anda dapat mencari pekerjaan pelatihan menggunakan berbagai atribut pekerjaan. Perhatikan bahwa beberapa parameter pencarian hanya muncul jika Anda telah membuat pekerjaan pelatihan dengan atribut tersebut. Misalnya, Tag hanya muncul jika Anda telah menambahkan tag untuk pekerjaan pelatihan.

Untuk menemukan pekerjaan pelatihan (konsol)

1. Buka SageMaker konsol Amazon.
2. Di panel navigasi, pilih Cari.

3. Tambahkan Parameter.

   a. Di kotak pencarian, masukkan parameter dan pilih jenis parameter, misalnya TrainingJobName.

   b. Pilih operasi bersyarat. Untuk nilai numerik, gunakan operator seperti is sama dengan, lebih kecil dari, atau atau lebih besar dari. Untuk nilai berbasis teks, gunakan operator seperti sama dengan atau berisi.

   c. Masukkan nilai untuk parameter.

4. (Opsional) Untuk mempersempit pencarian Anda, tambahkan kriteria pencarian tambahan. Pilih Tambah baris dan masukkan nilai parameter.

5. Pilih Cari.

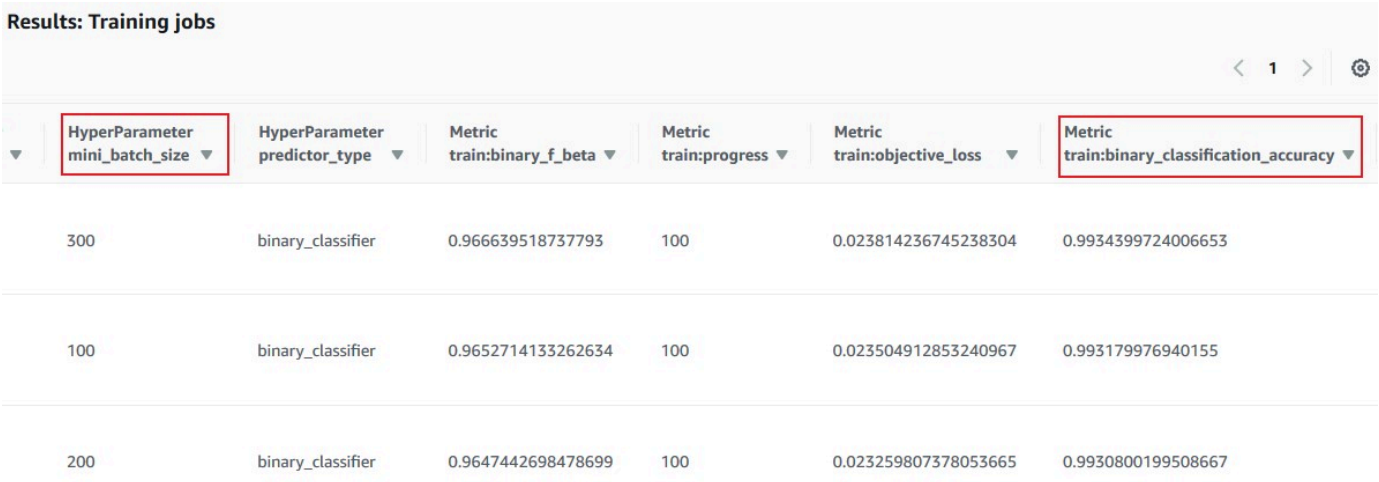
Evaluasi Model (Konsol)

Untuk mengevaluasi kinerja model, tinjau metadata, hiperparameter, dan metriknya. Untuk menyorot metrik, sesuaikan tampilan agar hanya menampilkan metrik dan hiperparameter penting.

Untuk mengevaluasi model (konsol)

1. Buka SageMaker konsol Amazon.

2. Di panel navigasi, pilih Cari dan cari pekerjaan pelatihan dengan menentukan parameter yang relevan. Hasilnya ditampilkan dalam tabel.



3. Buka jendela preferensi dengan memilih ikon pengaturan di tabel hasil pencarian.

4. Untuk menampilkan atau menyembunyikan hyperparameter atau metrik, aktifkan atau nonaktifkan dengan memilih Hyperparameter atau Metric.

5. Buat perubahan yang diperlukan, lalu pilih Perbarui tampilan.
6. Setelah melihat metrik dan hiperparameter penting, Anda dapat membandingkan dan membedakan hasilnya. Kemudian, Anda dapat memilih model terbaik untuk menjadi tuan rumah atau menyelidiki model yang berkinerja buruk.

Temukan dan Evaluasi Pekerjaan Pelatihan (API)

Untuk menemukan dan mengevaluasi pekerjaan pelatihan atau untuk mendapatkan saran untuk item yang digunakan dalam eksperimen yang dapat dicari, Anda dapat menggunakan API. `Search`

Topik

- Temukan Pekerjaan Pelatihan (API)
- Evaluasi Model (API)
- Dapatkan Saran untuk Penelusuran (API)

Temukan Pekerjaan Pelatihan (API)

Untuk menemukan pekerjaan pelatihan, buat parameter pencarian menggunakan `search_params` parameter. Kemudian gunakan fungsi pencarian di `smclient` subproses di AWS SDK for Python (Boto3).

Contoh berikut menunjukkan cara menggunakan `Search`API untuk menemukan pekerjaan pelatihan.

```
import boto3

search_params={
   "MaxResults": 10,
   "Resource": "TrainingJob",
   "SearchExpression": {
      "Filters": [{
            "Name": "Tags.Project",
            "Operator": "Equals",
            "Value": "Project_Binary_Classifier"
        }]},
  "SortBy": "Metrics.train:binary_classification_accuracy",
  "SortOrder": "Descending"
}

smclient = boto3.client(service_name='sagemaker')
```

```
results = smclient.search(**search_params)
```

Evaluasi Model (API)

Untuk mengevaluasi model, jalankan pencarian seperti yang dijelaskan dalamTemukan Pekerjaan Pelatihan (API), tinjau metrik model, lalu, gunakan AWS SDK for Python (Boto3) untuk membuat tabel dan memplot.

Contoh berikut menunjukkan bagaimana untuk mengevaluasi model dan untuk menampilkan hasil dalam tabel.

```
import pandas

headers=["Training Job Name", "Training Job Status", "Batch Size", "Binary
 Classification Accuracy"]
rows=[]
for result in results['Results']:
    trainingJob = result['TrainingJob']
    metrics = trainingJob['FinalMetricDataList']
    rows.append([trainingJob['TrainingJobName'],
     trainingJob['TrainingJobStatus'],
     trainingJob['HyperParameters']['mini_batch_size'],
     metrics[[x['MetricName'] for x in
     metrics].index('train:binary_classification_accuracy')]['Value']
    ])

df = pandas.DataFrame(data=rows,columns=headers)

from IPython.display import display, HTMLdisplay(HTML(df.to_html()))
```

Dapatkan Saran untuk Penelusuran (API)

Untuk mendapatkan saran untuk pencarian, gunakan `GetSearchSuggestions`API.

Contoh berikut untuk AWS SDK for Python (Boto3) SDK for Python (Boto3) adalah permintaan untuk item yang berisi`get_search_suggestions`. `linear`

```
search_suggestion_params={
   "Resource": "TrainingJob",
   "SuggestionQuery": {
      "PropertyNameQuery": {
         "PropertyNameHint": "linear"
```

```
      }
   }
}
```

Berikut ini adalah contoh respons untuk `get_search_suggestions` permintaan.

```
{
        'PropertyNameSuggestions': [{'PropertyName':
 'hyperparameters.linear_init_method'},
                    {'PropertyName': 'hyperparameters.linear_init_value'},
                    {'PropertyName': 'hyperparameters.linear_init_sigma'},
                    {'PropertyName': 'hyperparameters.linear_lr'},
                    {'PropertyName': 'hyperparameters.linear_wd'}]
}
```

Setelah mendapatkan saran pencarian, Anda dapat menggunakan salah satu nama properti dalam pencarian.

Verifikasi Dataset yang Digunakan oleh Pekerjaan Pelatihan Anda

Anda dapat menggunakan kemampuan pelacakan model untuk memverifikasi kumpulan data mana yang digunakan dalam pelatihan, tempat kumpulan data penahanan digunakan, dan detail lainnya tentang pekerjaan pelatihan. Misalnya, gunakan kemampuan pelacakan model untuk memverifikasi bahwa kumpulan data tertentu digunakan dalam pekerjaan pelatihan untuk audit atau untuk memverifikasi kepatuhan.

Untuk memeriksa apakah kumpulan data tertentu digunakan dalam pekerjaan pelatihan, Anda mencari URL ke lokasinya di Amazon Simple Storage Service (Amazon S3). Kemampuan pelacakan model mengembalikan pekerjaan pelatihan yang menggunakan kumpulan data yang Anda tentukan. Jika pencarian Anda tidak mengembalikan kumpulan data (hasilnya kosong), kumpulan data tidak digunakan dalam pekerjaan pelatihan. Hasil kosong mengonfirmasi, misalnya, bahwa kumpulan data penahanan tidak digunakan.

Garis Silsilah Model Jejak

Anda dapat menggunakan kemampuan pelacakan model untuk mendapatkan informasi tentang garis keturunan pekerjaan pelatihan dan sumber daya model yang digunakan untuk mereka, termasuk kumpulan data, algoritme, hiperparameter, dan metrik. Misalnya, jika Anda menemukan bahwa kinerja model yang dihosting telah menurun, Anda dapat meninjau pekerjaan pelatihannya dan sumber daya yang digunakan untuk menentukan penyebab masalah.

Topik

- Lacak Model Silsilah (Konsol)
- Garis Silsilah Model Jejak (API)

Lacak Model Silsilah (Konsol)

Untuk melacak garis keturunan model (konsol)

1. Buka SageMaker konsol Amazon.
2. Di panel navigasi, pilih Endpoints, dan pilih endpoint yang relevan.
3. Gulir ke bagian Pengaturan konfigurasi titik akhir. Bagian ini mencantumkan semua versi model yang diterapkan di titik akhir, dengan hyperlink ke pekerjaan pelatihan yang dibuat masing-masing.

Garis Silsilah Model Jejak (API)

Untuk melacak garis keturunan model, dapatkan nama model, lalu gunakan untuk mencari pekerjaan pelatihan.

Contoh berikut menunjukkan cara melacak garis keturunan model menggunakan API.

```
# Get the name of model deployed at endpoint
endpoint_config = smclient.describe_endpoint_config(EndpointConfigName=endpointName)
model_name = endpoint_config['ProductionVariants'][0]['ModelName']

# Get the model's name
model = smclient.describe_model(ModelName=model_name)

# Search the training job by the location of model artifacts in Amazon S3
search_params={
   "MaxResults": 1,
   "Resource": "TrainingJob",
   "SearchExpression": {
      "Filters": [
         {
            "Name": "ModelArtifacts.S3ModelArtifacts",
            "Operator": "Equals",
            "Value": model['PrimaryContainer']['ModelDataUrl']
         }]},
```

```
}
results = smclient.search(**search_params)
```

Setelah menemukan pekerjaan pelatihan, Anda dapat meninjau sumber daya yang digunakan untuk melatih model.

Lakukan Penyetelan Model Otomatis dengan SageMaker

Amazon SageMaker automatic model tuning (AMT), juga dikenal sebagai hyperparameter tuning, menemukan versi terbaik dari sebuah model dengan menjalankan banyak pekerjaan pelatihan pada dataset Anda. Untuk melakukan ini, AMT menggunakan algoritma dan rentang hyperparameters yang Anda tentukan. Kemudian memilih nilai hyperparameter yang menciptakan model yang berkinerja terbaik, yang diukur dengan metrik yang Anda pilih.

Misalnya, Anda ingin memecahkan masalah klasifikasi biner pada dataset pemasaran. Tujuan Anda adalah memaksimalkan metrik area di bawah kurva (AUC) algoritme dengan melatih Algoritma XGBoost model. Anda ingin menemukan nilai mana untuk`eta`,, `alpha`min_child_weight, dan `max_depth` hyperparameters yang akan melatih model terbaik. Tentukan rentang nilai untuk hiperparameter ini. Kemudian, pencarian tuning SageMaker hyperparameter dalam rentang ini untuk menemukan kombinasi nilai yang menciptakan pekerjaan pelatihan yang menciptakan model dengan AUC tertinggi. Untuk menghemat sumber daya atau memenuhi harapan kualitas model tertentu, Anda juga dapat mengatur kriteria penyelesaian untuk menghentikan penyetelan setelah kriteria terpenuhi.

Anda dapat menggunakan SageMaker AMT dengan algoritme bawaan, algoritme khusus, atau wadah SageMaker pra-bangun untuk kerangka kerja pembelajaran mesin.

SageMaker AMT dapat menggunakan instans Amazon EC2 Spot untuk mengoptimalkan biaya saat menjalankan pekerjaan pelatihan. Untuk informasi selengkapnya, lihat Gunakan Pelatihan Spot Terkelola di Amazon SageMaker.

Sebelum Anda mulai menggunakan tuning hyperparameter, Anda harus memiliki masalah pembelajaran mesin yang terdefinisi dengan baik, termasuk yang berikut ini:

- Dataset
- Pemahaman tentang jenis algoritma yang perlu Anda latih
- Pemahaman yang jelas tentang bagaimana Anda mengukur kesuksesan

Siapkan dataset dan algoritme Anda sehingga mereka bekerja SageMaker dan berhasil menjalankan pekerjaan pelatihan setidaknya sekali. Untuk informasi tentang menyiapkan dan menjalankan pekerjaan pelatihan, lihatMemulai.

Topik

Bagaimana Hyperparameter Tuning Bekerja

Ketika Anda membangun sistem pembelajaran mesin yang kompleks seperti jaringan saraf pembelajaran mendalam, menjelajahi semua kombinasi yang mungkin tidak praktis. Penyetelan hyperparameter dapat mempercepat produktivitas Anda dengan mencoba banyak variasi model. Ini mencari model terbaik secara otomatis dengan berfokus pada kombinasi nilai hyperparameter yang paling menjanjikan dalam rentang yang Anda tentukan. Untuk mendapatkan hasil yang baik, Anda harus memilih rentang yang tepat untuk dijelajahi.

Gunakan panduan referensi API untuk memahami cara berinteraksi dengan tuning hyperparameter. Contoh pada halaman ini dapat ditemukan di HyperbandStrategyConfigAPI HyperParameterTuningJobConfigdan.

> **Note**
>
> Karena algoritme itu sendiri adalah stokastik, ada kemungkinan bahwa model tuning hyperparameter akan gagal untuk bertemu pada jawaban terbaik. Ini dapat terjadi bahkan jika kombinasi nilai terbaik ada dalam rentang yang Anda pilih.

Pencarian Grid

Saat menggunakan pencarian kisi, penyetelan hyperparameter memilih kombinasi nilai dari rentang nilai kategoris yang Anda tentukan saat Anda membuat pekerjaan. Hanya parameter kategoris yang didukung saat menggunakan strategi pencarian kisi. Anda tidak perlu menentukan`MaxNumberOfTrainingJobs`. Jumlah pekerjaan pelatihan yang dibuat oleh pekerjaan penyetelan akan secara otomatis dihitung menjadi jumlah total kombinasi kategoris yang berbeda yang mungkin. Jika ditentukan, nilai `MaxNumberOfTrainingJobs` harus sama dengan jumlah total kombinasi kategoris yang berbeda mungkin.

Pencarian Acak

Saat menggunakan pencarian acak, penyetelan hyperparameter memilih kombinasi nilai acak dari dalam rentang yang Anda tentukan untuk hiperparameter untuk setiap pekerjaan pelatihan yang diluncurkan. Karena pilihan nilai hyperparameter tidak bergantung pada hasil pekerjaan pelatihan sebelumnya, Anda dapat menjalankan jumlah maksimum pekerjaan pelatihan bersamaan tanpa mempengaruhi kinerja penyetelan.

Untuk contoh notebook yang menggunakan pencarian acak, lihat Pencarian acak dan penskalaan hyperparameter dengan notebook SageMaker XGBoost dan Automatic Model Tuning.

Optimasi Bayesian

Optimasi Bayesian memperlakukan penyetelan hyperparameter seperti masalah regresi. Mengingat serangkaian fitur input (hyperparameters), tuning hyperparameter mengoptimalkan model untuk metrik yang Anda pilih. Untuk mengatasi masalah regresi, penyetelan hyperparameter membuat tebakan tentang kombinasi hyperparameter mana yang cenderung mendapatkan hasil terbaik, dan menjalankan pekerjaan pelatihan untuk menguji nilai-nilai ini. Setelah menguji satu set nilai hyperparameter, tuning hyperparameter menggunakan regresi untuk memilih set nilai hyperparameter berikutnya untuk diuji.

Penyetelan hyperparameter menggunakan SageMaker implementasi Amazon dari optimasi Bayesian.

Saat memilih hiperparameter terbaik untuk pekerjaan pelatihan berikutnya, penyetelan hyperparameter mempertimbangkan semua yang diketahui tentang masalah ini sejauh ini. Terkadang ia memilih kombinasi nilai hiperparameter yang dekat dengan kombinasi yang menghasilkan pekerjaan pelatihan terbaik sebelumnya untuk meningkatkan kinerja secara bertahap. Hal ini memungkinkan tuning hyperparameter untuk mengeksploitasi hasil yang paling terkenal. Di

lain waktu, ia memilih satu set nilai hyperparameter yang jauh dari yang telah dicoba. Hal ini memungkinkannya untuk mengeksplorasi kisaran nilai hyperparameter untuk mencoba menemukan area baru yang belum dipahami dengan baik. Explore/exploit trade-off adalah hal biasa dalam banyak masalah pembelajaran mesin.

Untuk informasi selengkapnya tentang optimasi Bayesian, lihat berikut ini:

Topik Dasar tentang Optimasi Bayesian

- Tutorial tentang Optimalisasi Bayesian dari Fungsi Biaya Mahal, dengan Aplikasi untuk Pemodelan Pengguna Aktif dan Pembelajaran Penguatan Hirarkis
- Optimalisasi Bayesian Praktis dari Algoritma Machine Learning
- Mengambil Manusia Keluar dari Lingkaran: Tinjauan Optimasi Bayesian

Mempercepat Optimasi Bayesian

- Google Wazir: Layanan untuk Optimasi Black-Box
- Prediksi Kurva Pembelajaran dengan Jaringan Saraf Bayesian
- Mempercepat optimalisasi hiperparameter otomatis jaringan saraf dalam dengan ekstrapolasi kurva pembelajaran

Pemodelan Tingkat Lanjut dan Pembelajaran Transfer

- Pembelajaran Transfer Hyperparameter yang Dapat Diskalakan
- Optimasi Bayesian dengan Dependensi Terstruktur Pohon
- Optimasi Bayesian dengan Jaringan Saraf Bayesian yang Kuat
- Optimasi Bayesian yang Dapat Diskalakan Menggunakan Jaringan Neural Dalam
- Input Warping untuk Optimalisasi Bayesian dari Fungsi Non-stasioner

Hyperband

Hyperband adalah strategi tuning berbasis multi-fidelity yang secara dinamis merealokasi sumber daya. Hyperband menggunakan hasil menengah dan akhir dari pekerjaan pelatihan untuk mengalokasikan kembali zaman ke konfigurasi hyperparameter yang digunakan dengan baik dan secara otomatis menghentikan yang berkinerja buruk. Ini juga menskalakan dengan mulus untuk

menggunakan banyak pekerjaan pelatihan paralel. Fitur-fitur ini dapat secara signifikan mempercepat penyetelan hyperparameter melalui pencarian acak dan strategi optimasi Bayesian.

Hyperband seharusnya hanya digunakan untuk menyetel algoritme berulang yang mempublikasikan hasil pada tingkat sumber daya yang berbeda. Misalnya, Hyperband dapat digunakan untuk menyetel jaringan saraf untuk klasifikasi gambar yang menerbitkan metrik akurasi setelah setiap zaman.

Untuk informasi lebih lanjut tentang Hyperband, lihat tautan berikut:

- Hyperband: Pendekatan Berbasis Bandit Baru untuk Optimasi Hyperparameter
- Penyetelan Hyperparameter Paralel Secara Masif
- BOHB: Optimasi Hyperparameter yang Kuat dan Efisien pada Skala
- Pencarian Hyperparameter Asinkron dan Arsitektur Saraf Berbasis Model

Hyperband dengan berhenti lebih awal

Pekerjaan pelatihan dapat dihentikan lebih awal ketika mereka tidak mungkin meningkatkan metrik objektif dari pekerjaan penyetelan hiperparameter. Ini dapat membantu mengurangi waktu komputasi dan menghindari overfitting model Anda. Hyperband menggunakan mekanisme internal canggih untuk menerapkan penghentian dini. Dengan demikian, parameter `TrainingJobEarlyStoppingType` dalam `HyperParameterTuningJobConfig` API harus disetel ke `OFF` saat menggunakan fitur penghentian awal internal Hyperband.

> **Note**
>
> Penyetelan hyperparameter mungkin tidak meningkatkan model Anda. Ini adalah alat canggih untuk membangun solusi mesin. Dengan demikian, harus dianggap sebagai bagian dari proses pengembangan ilmiah.

Tentukan metrik dan variabel lingkungan

Pekerjaan penyetelan mengoptimalkan hiperparameter untuk pekerjaan pelatihan yang diluncurkan dengan menggunakan metrik untuk mengevaluasi kinerja. Panduan ini menunjukkan cara menentukan metrik sehingga Anda dapat menggunakan algoritme khusus untuk pelatihan, atau menggunakan algoritme bawaan dari Amazon SageMaker. Panduan ini juga menunjukkan cara menentukan variabel lingkungan selama pekerjaan Automatic model tuning (AMT).

Tentukan metrik

Penyetelan SageMaker hiperparameter Amazon mem-parsing algoritme pembelajaran mesin `stdout` dan `stderr` aliran Anda untuk menemukan metrik, seperti kehilangan atau akurasi validasi. Metrik menunjukkan seberapa baik kinerja model pada dataset.

Bagian berikut menjelaskan cara menggunakan dua jenis algoritma untuk pelatihan: built-in dan custom.

Gunakan algoritma bawaan untuk pelatihan

Jika Anda menggunakan salah satu algoritma SageMaker bawaan, metrik sudah ditentukan untuk Anda. Selain itu, algoritme bawaan secara otomatis mengirim metrik ke penyetelan hyperparameter untuk pengoptimalan. Metrik ini juga ditulis ke CloudWatch log Amazon. Untuk informasi selengkapnya, lihat Log SageMaker Acara Amazon dengan Amazon CloudWatch.

Untuk metrik objektif untuk pekerjaan penyetelan, pilih salah satu metrik yang dipancarkan oleh algoritme bawaan. Untuk daftar metrik yang tersedia, lihat bagian penyetelan model untuk algoritme yang sesuai di Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih.

Anda dapat memilih hingga 40 metrik untuk dipantau dalam pekerjaan penyetelan Anda. Pilih salah satu metrik tersebut untuk menjadi metrik objektif. Pekerjaan tuning hyperparameter mengembalikan pekerjaan pelatihan yang berkinerja terbaik terhadap metrik objektif.

> **Note**
>
> Penyetelan hyperparameter secara otomatis mengirimkan hyperparameter tambahan `_tuning_objective_metric` untuk meneruskan metrik objektif Anda ke pekerjaan penyetelan untuk digunakan selama pelatihan.

Gunakan algoritma khusus untuk pelatihan

Bagian ini menunjukkan cara menentukan metrik Anda sendiri untuk menggunakan algoritme kustom Anda sendiri untuk pelatihan. Saat melakukannya, pastikan algoritme Anda menulis setidaknya satu metrik ke `stderr` atau`stdout`. Penyetelan hyperparameter mem-parsing aliran ini untuk menemukan metrik algoritme yang menunjukkan seberapa baik kinerja model pada kumpulan data.

Anda dapat menentukan metrik kustom dengan menentukan nama dan ekspresi reguler untuk setiap metrik yang dipantau oleh pekerjaan penyetelan Anda. Kemudian, berikan definisi metrik

ini CreateHyperParameterTuningJobke API di TrainingJobDefinition parameter di MetricDefinitions bidangAlgorithmSpecification.

Berikut ini menunjukkan contoh output dari log yang ditulis ke stderr atau stdout oleh algoritma pelatihan.

```
GAN_loss=0.138318;  Scaled_reg=2.654134; disc:[-0.017371,0.102429] real 93.3% gen 0.0%
 disc-combined=0.000000; disc_train_loss=1.374587;  Loss = 16.020744;  Iteration 0 took
 0.704s;  Elapsed=0s
```

Contoh kode berikut menunjukkan bagaimana menggunakan ekspresi reguler di Python (regex). Ini digunakan untuk mencari output log sampel dan menangkap nilai numerik dari empat metrik yang berbeda.

```
[
    {
        "Name": "ganloss",
        "Regex": "GAN_loss=(.*?);",
    },
    {
        "Name": "disc-combined",
        "Regex": "disc-combined=(.*?);",
    },
    {
        "Name": "discloss",
        "Regex": "disc_train_loss=(.*?);",
    },
    {
        "Name": "loss",
        "Regex": "Loss = (.*?);",
    },
]
```

Dalam ekspresi reguler, tanda kurung ( ) digunakan untuk mengelompokkan bagian-bagian dari ekspresi reguler bersama-sama.

- Untuk loss metrik yang didefinisikan dalam contoh kode, ekspresi (.*?); menangkap karakter apa pun antara teks yang tepat "Loss=" dan karakter titik koma () pertama. ;
- Karakter . menginstruksikan ekspresi reguler untuk mencocokkan karakter apa pun.
- Karakter * berarti mencocokkan nol atau lebih karakter.

- Karakter `?` berarti menangkap hanya sampai contoh pertama dari `;` karakter.

Metrik kerugian yang ditentukan dalam sampel kode akan menangkap `Loss = 16.020744` dari output sampel.

Pilih salah satu metrik yang Anda definisikan sebagai metrik objektif untuk pekerjaan penyetelan. Jika Anda menggunakan SageMaker API, tentukan nilai `name` kunci di `HyperParameterTuningJobObjective` bidang `HyperParameterTuningJobConfig` parameter yang Anda kirim ke `CreateHyperParameterTuningJob`operasi.

Tentukan variabel lingkungan

SageMaker AMT mengoptimalkan hiperparameter dalam pekerjaan penyetelan untuk menemukan parameter terbaik untuk kinerja model. Anda dapat menggunakan variabel lingkungan untuk mengonfigurasi pekerjaan penyetelan Anda untuk mengubah perilakunya. Anda juga dapat menggunakan variabel lingkungan yang Anda gunakan selama pelatihan di dalam pekerjaan penyetelan Anda.

Jika Anda ingin menggunakan variabel lingkungan dari pekerjaan penyetelan atau menentukan variabel lingkungan baru, masukkan nilai string untuk `Environment` dalam SageMaker HyperParameterTrainingJobDefinitionAPI. Lulus definisi pekerjaan pelatihan ini ke CreateHyperParameterTuningJobAPI.

Misalnya, variabel lingkungan `SM_LOG_LEVEL` dapat diatur ke nilai-nilai berikut untuk menyesuaikan output dari wadah Python.

```
NOTSET=0
DEBUG=10
INFO=20
WARN=30
ERROR=40
CRITICAL=50
```

Sebagai contoh, untuk mengatur tingkat log `10` untuk men-debug log kontainer Anda, atur variabel lingkungan di dalam HyperParameterTrainingJobDefinition, sebagai berikut.

```
{
   "HyperParameterTuningJobConfig": {
   ...,
```

```
    }
    "TrainingJobDefinition": {
        ...,
        "Environment" : [
            {
              "SM_LOG_LEVEL": 10
            }
        ],
        ...,
    },
    ...,
}
```

Tentukan Rentang Hyperparameter

Panduan ini menunjukkan cara menggunakan SageMaker API untuk menentukan rentang hyperparameter. Ini juga menyediakan daftar jenis penskalaan hyperparameter yang dapat Anda gunakan.

Memilih hiperparameter dan rentang secara signifikan memengaruhi kinerja pekerjaan penyetelan Anda. Penyetelan hyperparameter menemukan nilai hyperparameter terbaik untuk model Anda dengan mencari pada rentang nilai yang Anda tentukan untuk setiap hyperparameter yang dapat disetel. Anda juga dapat menentukan hingga 100 hiperparameter statis yang tidak berubah selama pekerjaan penyetelan. Anda dapat menggunakan hingga 100 hyperparameters secara total (statis +tunable). Untuk panduan memilih hiperparameter dan rentang, lihatPraktik Terbaik untuk Tuning Hyperparameter. Anda juga dapat menggunakan autotune untuk menemukan pengaturan pekerjaan penyetelan yang optimal. Untuk informasi lebih lanjut, lihat bagian Autotune berikut.

> **Note**
>
> SageMaker Automatic Model Tuning (AMT) dapat menambahkan hiperparameter tambahan yang berkontribusi pada batas 100 hiperparameter total. Saat ini, untuk meneruskan metrik tujuan Anda ke pekerjaan penyetelan untuk digunakan selama pelatihan, SageMaker tambahkan `_tuning_objective_metric` secara otomatis.

Hiperparameter statis

Gunakan hyperparameters statis untuk kasus berikut: Misalnya, Anda dapat menggunakan AMT untuk menyetel model Anda menggunakan `param1` (parameter yang dapat disetel) dan `param2`

(parameter statis). Jika Anda melakukannya, maka gunakan ruang pencarian untuk `param1` yang terletak di antara dua nilai, dan lulus `param2` sebagai hyperparameter statis, sebagai berikut.

```
param1: ["range_min","range_max"]
param2: "static_value"
```

Hiperparameter statis memiliki struktur berikut:

```
"StaticHyperParameters": {
    "objective" : "reg:squarederror",
    "dropout_rate": "0.3"
}
```

Anda dapat menggunakan Amazon SageMaker API untuk menentukan pasangan nilai kunci di StaticHyperParametersbidang `HyperParameterTrainingJobDefinition` parameter yang Anda teruskan ke CreateHyperParameterTuningJoboperasi.

Hiperparameter dinamis

Anda dapat menggunakan SageMaker API untuk menentukan rentang hyperparameter. Tentukan nama hyperparameters dan rentang nilai di `ParameterRanges` bidang `HyperParameterTuningJobConfig` parameter yang Anda berikan ke `CreateHyperParameterTuningJob`operasi.

`ParameterRanges`Bidang ini memiliki tiga subbidang: kategoris, bilangan bulat, dan kontinu. Anda dapat menentukan hingga 30 total (kategorik+integer+kontinu) hiperparameter yang dapat disetel untuk dicari.

> **Note**
>
> Setiap hiperparameter kategoris dapat memiliki paling banyak 30 nilai yang berbeda.

Hyperparameter dinamis memiliki struktur berikut:

```
"ParameterRanges": {
    "CategoricalParameterRanges": [
        {
            "Name": "tree_method",
            "Values": ["auto", "exact", "approx", "hist"]
        }
```

```
    ],
    "ContinuousParameterRanges": [
        {
            "Name": "eta",
            "MaxValue" : "0.5",
            "MinValue": "0",
            "ScalingType": "Auto"
        }
    ],
    "IntegerParameterRanges": [
        {
            "Name": "max_depth",
            "MaxValue": "10",
            "MinValue": "1",
            "ScalingType": "Auto"
        }
    ]
}
```

Jika Anda membuat pekerjaan penyetelan dengan `Grid` strategi, Anda hanya dapat menentukan nilai kategoris. Anda tidak perlu menyediakan`MaxNumberofTrainingJobs`. Nilai ini disimpulkan dari jumlah total konfigurasi yang dapat dihasilkan dari parameter kategoris Anda. Jika ditentukan, nilai `MaxNumberOfTrainingJobs` harus sama dengan jumlah total kombinasi kategoris yang berbeda mungkin.

Autotune

Untuk menghemat waktu dan sumber daya mencari rentang hiperparameter, sumber daya, atau metrik objektif, autotune dapat secara otomatis menebak nilai optimal untuk beberapa bidang hyperparameter. Gunakan autotune untuk menemukan nilai optimal untuk bidang berikut:

- ParameterRanges— Nama dan rentang hiperparameter yang dapat dioptimalkan oleh pekerjaan penyetelan.
- ResourceLimits— Sumber daya maksimum yang akan digunakan dalam pekerjaan penyetelan. Sumber daya ini dapat mencakup jumlah maksimum pekerjaan pelatihan, runtime maksimum dari pekerjaan tuning, dan jumlah maksimum pekerjaan pelatihan yang dapat dijalankan pada saat yang sama.
- TrainingJobEarlyStoppingTypeBendera yang menghentikan pekerjaan pelatihan jika suatu pekerjaan tidak membaik secara signifikan terhadap metrik objektif. Default untuk diaktifkan. Untuk informasi selengkapnya, lihat Hentikan Pekerjaan Pelatihan Lebih Awal.

- RetryStrategy— Berapa kali untuk mencoba kembali pekerjaan pelatihan. Nilai bukan nol untuk `RetryStrategy` dapat meningkatkan kemungkinan bahwa pekerjaan Anda akan berhasil diselesaikan.
- Strategi - Menentukan bagaimana tuning hyperparameter memilih kombinasi nilai hyperparameter yang akan digunakan untuk pekerjaan pelatihan yang diluncurkan.
- ConvergenceDetected— Bendera untuk menunjukkan bahwa Automatic Model Tuning (AMT) telah mendeteksi konvergensi model.

Untuk menggunakan autotune, lakukan hal berikut:

1. Tentukan hyperparameter dan nilai contoh di `AutoParameters` bidang ParameterRangesAPI.
2. Aktifkan autotune.

AMT akan menentukan apakah hyperparameters dan nilai contoh Anda memenuhi syarat untuk autotune. Hyperparameter yang dapat digunakan dalam autotune secara otomatis ditetapkan ke jenis rentang parameter yang sesuai. Kemudian, AMT menggunakan `ValueHint` untuk memilih rentang optimal untuk Anda. Anda dapat menggunakan `DescribeHyperParameterTrainingJob` API untuk melihat rentang ini.

Contoh berikut menunjukkan cara mengkonfigurasi pekerjaan tuning yang menggunakan autotune. Dalam contoh konfigurasi, hyperparameter `max_depth` telah `ValueHint` berisi nilai contoh dari. `4`

```
config = {
    'Autotune': {'Mode': 'Enabled'},
    'HyperParameterTuningJobName':'my-autotune-job',
    'HyperParameterTuningJobConfig': {
        'HyperParameterTuningJobObjective': {'Type': 'Minimize', 'MetricName':
 'validation:rmse'},
        'ResourceLimits': {'MaxNumberOfTrainingJobs': 5, 'MaxParallelTrainingJobs': 1},
        'ParameterRanges': {
            'AutoParameters': [
                {'Name': 'max_depth', 'ValueHint': '4'}
            ]
        }
    },
    'TrainingJobDefinition': {
    .... }
```

Melanjutkan contoh sebelumnya, pekerjaan penyetelan dibuat setelah konfigurasi sebelumnya disertakan dalam panggilan ke `CreateHyperParameterTuningJob` API. Kemudian, autotune mengubah max_depth hyperparameter menjadi hyperparameter. AutoParameters `IntegerParameterRanges` Respons berikut dari `DescribeHyperParameterTrainingJob` API menunjukkan bahwa yang optimal `IntegerParameterRanges` untuk `max_depth` adalah antara 2 dan8.

```
{
    'HyperParameterTuningJobName':'my_job',
    'HyperParameterTuningJobConfig': {
        'ParameterRanges': {
            'IntegerParameterRanges': [
                {'Name': 'max_depth', 'MinValue': '2', 'MaxValue': '8'},
            ],
        }
    },
    'TrainingJobDefinition': {
        ...
    },
    'Autotune': {'Mode': 'Enabled'}

}
```

Jenis penskalaan hyperparameter

Untuk rentang hiperparameter integer dan kontinu, Anda dapat memilih skala yang ingin digunakan oleh penyetelan hyperparameter. Misalnya, untuk mencari rentang nilai, Anda dapat menentukan nilai untuk `ScalingType` bidang rentang hyperparameter. Anda dapat memilih dari jenis penskalaan hyperparameter berikut:

Otomatis

SageMaker tuning hyperparameter memilih skala terbaik untuk hyperparameter.

Linear

Tuning hyperparameter mencari nilai dalam rentang hyperparameter dengan menggunakan skala linier. Biasanya, Anda memilih ini jika rentang semua nilai dari terendah ke tertinggi relatif kecil (dalam satu urutan besarnya). Nilai pencarian seragam dari rentang memberikan eksplorasi yang masuk akal dari seluruh rentang.

Logaritmik

Tuning hyperparameter mencari nilai dalam rentang hyperparameter dengan menggunakan skala logaritmik.

Penskalaan logaritmik hanya berfungsi untuk rentang yang memiliki nilai lebih besar dari 0.

Pilih penskalaan logaritmik saat Anda mencari rentang yang mencakup beberapa urutan besarnya.

Misalnya, jika Anda menyetel Menyetel model pembelajar linier model, dan Anda menentukan rentang nilai antara .0001 dan 1.0 untuk `learning_rate` hiperparameter, pertimbangkan hal berikut: Mencari secara seragam pada skala logaritmik memberi Anda sampel yang lebih baik dari seluruh rentang daripada mencari pada skala linier. Ini karena pencarian dalam skala linier rata-rata akan mencurahkan 90 persen anggaran pelatihan Anda hanya untuk nilai antara 0,1 dan 1,0. Akibatnya, itu hanya menyisakan 10 persen dari anggaran pelatihan Anda untuk nilai antara .0001 dan .1.

`ReverseLogarithmic`

Tuning hyperparameter mencari nilai dalam rentang hyperparameter dengan menggunakan skala logaritmik terbalik. Penskalaan logaritmik terbalik hanya didukung untuk rentang hyperparameter kontinu. Hal ini tidak didukung untuk rentang hyperparameter integer.

Pilih penskalaan logaritmik terbalik saat Anda mencari rentang yang sangat sensitif terhadap perubahan kecil yang sangat dekat dengan 1.

Penskalaan logaritmik terbalik hanya berfungsi untuk rentang yang seluruhnya berada dalam kisaran 0 <=x <1.0.

Untuk contoh notebook yang menggunakan penskalaan hyperparameter, lihat contoh SageMaker hyperparameter Amazon ini di. GitHub

Lacak dan tetapkan kriteria penyelesaian untuk pekerjaan penyetelan Anda

Anda dapat menggunakan kriteria penyelesaian untuk menginstruksikan Penyetelan model otomatis (AMT) untuk menghentikan pekerjaan penyetelan Anda jika kondisi tertentu terpenuhi. Dengan kondisi ini, Anda dapat menetapkan kinerja model minimum atau jumlah maksimum pekerjaan pelatihan yang tidak membaik ketika dievaluasi terhadap metrik objektif. Anda juga dapat melacak kemajuan pekerjaan penyetelan Anda dan memutuskan untuk membiarkannya berlanjut atau

menghentikannya secara manual. Panduan ini menunjukkan kepada Anda cara menetapkan kriteria penyelesaian, memeriksa kemajuan dan menghentikan pekerjaan penyetelan Anda secara manual.

Tetapkan kriteria penyelesaian untuk pekerjaan tuning Anda

Selama optimasi hyperparameter, pekerjaan tuning akan meluncurkan beberapa pekerjaan pelatihan di dalam satu loop. Pekerjaan tuning akan melakukan hal berikut.

- Periksa pekerjaan pelatihan Anda untuk penyelesaian dan perbarui statistik yang sesuai
- Tentukan kombinasi hiperparameter apa yang akan dievaluasi selanjutnya.

AMT akan terus memeriksa pekerjaan pelatihan yang diluncurkan dari pekerjaan penyetelan Anda untuk memperbarui statistik. Statistik ini termasuk tuning job runtime dan pekerjaan pelatihan terbaik. Kemudian, AMT menentukan apakah itu harus menghentikan pekerjaan sesuai dengan kriteria penyelesaian Anda. Anda juga dapat memeriksa statistik ini dan menghentikan pekerjaan Anda secara manual. Untuk informasi selengkapnya tentang menghentikan pekerjaan secara manual, lihat Menghentikan pekerjaan penyetelan Anda secara manual bagian.

Misalnya, jika pekerjaan tuning Anda memenuhi tujuan Anda, Anda dapat berhenti menyetel lebih awal untuk menghemat sumber daya atau memastikan kualitas model. AMT memeriksa kinerja pekerjaan Anda terhadap kriteria penyelesaian Anda dan menghentikan pekerjaan tuning jika ada yang terpenuhi.

Anda dapat menentukan jenis kriteria penyelesaian berikut:

- `MaxNumberOfTrainingJobs`— Jumlah maksimum pekerjaan pelatihan yang harus dijalankan sebelum penyetelan dihentikan.
- `MaxNumberOfTrainingJobsNotImproving`— Jumlah maksimum pekerjaan pelatihan yang tidak meningkatkan kinerja terhadap metrik objektif dari pekerjaan pelatihan terbaik saat ini. Sebagai contoh, jika pekerjaan pelatihan terbaik mengembalikan metrik objektif yang memiliki akurasi`90%`, dan `MaxNumberOfTrainingJobsNotImproving` diatur ke`10`. Dalam contoh ini, penyetelan akan berhenti setelah pekerjaan `10` pelatihan gagal mengembalikan akurasi yang lebih tinggi dari`90%`.
- `MaxRuntimeInSeconds`— Batas atas waktu jam dinding dalam hitungan detik berapa lama pekerjaan penyetelan dapat berjalan.
- `TargetObjectiveMetricValue`— Nilai metrik objektif yang digunakan untuk mengevaluasi pekerjaan penyetelan. Setelah nilai ini terpenuhi, AMT menghentikan pekerjaan penyetelan.

- `CompleteOnConvergence`- Bendera untuk berhenti menyetel setelah algoritme internal menentukan bahwa pekerjaan penyetelan tidak mungkin meningkat lebih dari 1% dibandingkan metrik objektif dari pekerjaan pelatihan terbaik.

Memilih kriteria penyelesaian

Anda dapat memilih satu atau beberapa kriteria penyelesaian untuk menghentikan pekerjaan penyetelan hiperparameter Anda setelah suatu kondisi terpenuhi. Petunjuk berikut menunjukkan cara memilih kriteria penyelesaian dan cara memutuskan mana yang paling tepat untuk kasus penggunaan Anda.

- Gunakan `MaxNumberOfTrainingJobs` di ResourceLimitsAPI untuk menetapkan batas atas jumlah pekerjaan pelatihan yang dapat dijalankan sebelum pekerjaan penyetelan Anda dihentikan. Mulailah dengan jumlah besar dan sesuaikan berdasarkan kinerja model terhadap tujuan pekerjaan penyetelan Anda. Sebagian besar pengguna memasukkan nilai sekitar `50` atau lebih pekerjaan pelatihan untuk menemukan konfigurasi hyperparameter yang optimal. Pengguna yang mencari tingkat kinerja model yang lebih tinggi akan menggunakan `200` atau lebih banyak pekerjaan pelatihan.
- Gunakan `MaxNumberOfTrainingJobsNotImproving` di bidang BestObjectiveNotImprovingAPI untuk menghentikan pelatihan jika kinerja model gagal ditingkatkan setelah sejumlah pekerjaan tertentu. Kinerja model dievaluasi terhadap fungsi objektif. Setelah `MaxNumberOfTrainingJobsNotImproving` terpenuhi, AMT akan menghentikan pekerjaan penyetelan. Pekerjaan tuning cenderung membuat kemajuan paling besar di awal pekerjaan. Meningkatkan kinerja model terhadap fungsi objektif akan membutuhkan lebih banyak pekerjaan pelatihan menjelang akhir penyetelan. Pilih nilai `MaxNumberOfTrainingJobsNotImproving` dengan memeriksa kinerja pekerjaan pelatihan serupa terhadap metrik tujuan Anda.
- Gunakan `MaxRuntimeInSeconds` di ResourceLimitsAPI untuk menetapkan batas atas jumlah waktu jam dinding yang mungkin diperlukan oleh pekerjaan penyetelan. Gunakan bidang ini untuk memenuhi tenggat waktu dimana pekerjaan penyetelan harus diselesaikan atau untuk membatasi sumber daya komputasi.

  Untuk mendapatkan perkiraan total waktu komputasi dalam hitungan detik untuk pekerjaan penyetelan, gunakan rumus berikut:

  Perkiraan waktu komputasi maks dalam detik= * * `MaxRuntimeInSeconds` `MaxParallelTrainingJobs` `MaxInstancesPerTrainingJob`

> **Note**
>
> Durasi sebenarnya dari pekerjaan penyetelan mungkin sedikit menyimpang dari nilai yang ditentukan dalam bidang ini.

- Gunakan `TargetObjectiveMetricValue` di TuningJobCompletionCriteriaAPI untuk menghentikan pekerjaan penyetelan Anda. Anda menghentikan pekerjaan penyetelan setelah pekerjaan pelatihan apa pun yang diluncurkan oleh pekerjaan penyetelan mencapai nilai metrik objektif ini. Gunakan bidang ini jika kasus penggunaan Anda bergantung pada pencapaian tingkat kinerja tertentu, daripada menghabiskan sumber daya komputasi untuk menemukan model terbaik.
- Gunakan `CompleteOnConvergence` di TuningJobCompletionCriteriaAPI untuk menghentikan pekerjaan penyetelan setelah AMT mendeteksi bahwa pekerjaan penyetelan telah konvergen, dan sepertinya tidak akan membuat kemajuan signifikan lebih lanjut. Gunakan bidang ini jika tidak jelas nilai apa untuk kriteria penyelesaian lainnya yang harus digunakan. AMT menentukan konvergensi berdasarkan algoritma yang dikembangkan dan diuji pada berbagai tolok ukur yang beragam. Pekerjaan tuning didefinisikan telah menyatu ketika tidak ada pekerjaan pelatihan yang menghasilkan peningkatan yang signifikan (1% atau kurang). Peningkatan diukur terhadap metrik objektif yang dikembalikan oleh pekerjaan berkinerja tertinggi, sejauh ini.

Menggabungkan kriteria penyelesaian yang berbeda

Anda juga dapat menggabungkan salah satu kriteria penyelesaian yang berbeda dalam pekerjaan penyetelan yang sama. AMT akan menghentikan pekerjaan penyetelan ketika salah satu kriteria penyelesaian terpenuhi. Misalnya, jika Anda ingin menyetel model Anda hingga memenuhi metrik objektif, tetapi tidak ingin terus menyetel jika pekerjaan Anda telah konvergen, gunakan panduan berikut.

- Tentukan `TargetObjectiveMetricValue` di TuningJobCompletionCriteriaAPI untuk menetapkan nilai metrik sasaran sasaran yang akan dicapai.
- Setel CompleteOnConvergence`Enabled`untuk menghentikan pekerjaan penyetelan jika AMT telah menentukan bahwa kinerja model tidak mungkin meningkat.

Lacak kemajuan pekerjaan tuning

Anda dapat menggunakan `DescribeHyperParameterTuningJob` API untuk melacak kemajuan pekerjaan penyetelan Anda kapan saja saat sedang berjalan. Anda tidak perlu menentukan kriteria

penyelesaian untuk mendapatkan informasi pelacakan untuk pekerjaan penyetelan Anda. Gunakan bidang berikut untuk mendapatkan statistik tentang pekerjaan tuning Anda.

- BestTrainingJob— Objek yang menggambarkan pekerjaan pelatihan terbaik yang diperoleh sejauh ini, dievaluasi berdasarkan metrik tujuan Anda. Gunakan bidang ini untuk memeriksa kinerja model Anda saat ini dan nilai metrik objektif dari pekerjaan pelatihan terbaik ini.
- ObjectiveStatusCounters— Objek yang menentukan jumlah total pekerjaan pelatihan yang diselesaikan dalam pekerjaan tuning. Untuk memperkirakan durasi rata-rata pekerjaan penyetelan, gunakan `ObjectiveStatusCounters` dan total runtime pekerjaan penyetelan. Anda dapat menggunakan durasi rata-rata untuk memperkirakan berapa lama pekerjaan tuning Anda akan berjalan.
- `ConsumedResources`— Total sumber daya, seperti`RunTimeInSeconds`, dikonsumsi oleh pekerjaan tuning Anda. Bandingkan`ConsumedResources`, ditemukan di DescribeHyperParameterTuningJob API, terhadap `BestTrainingJob` di API yang sama. Anda juga dapat `ConsumedResources` membandingkan respons dari ListTrainingJobsForHyperParameterTuningJob API untuk menilai apakah pekerjaan penyetelan Anda membuat kemajuan yang memuaskan mengingat sumber daya yang dikonsumsi.
- TuningJobCompletionDetails— Menyetel informasi penyelesaian pekerjaan yang mencakup hal-hal berikut:
  - Stempel waktu kapan konvergensi terdeteksi jika pekerjaan telah konvergen.
  - Jumlah pekerjaan pelatihan yang belum meningkatkan kinerja model. Kinerja model dievaluasi terhadap metrik objektif dari pekerjaan pelatihan terbaik.

  Gunakan kriteria penyelesaian pekerjaan tuning untuk menilai seberapa besar kemungkinan pekerjaan tuning Anda untuk meningkatkan kinerja model Anda. Kinerja model dievaluasi terhadap metrik objektif terbaik jika dijalankan hingga selesai.

Menghentikan pekerjaan penyetelan Anda secara manual

Anda dapat menentukan apakah Anda harus membiarkan pekerjaan tuning berjalan sampai selesai atau apakah Anda harus menghentikan pekerjaan penyetelan secara manual. Untuk menentukan hal ini, gunakan informasi yang ditampilkan oleh parameter di `DescribeHyperParameterTuningJob` API, seperti yang ditunjukkan di bagian Tracking tuning job progress sebelumnya. Misalnya, jika kinerja model Anda tidak membaik setelah beberapa pekerjaan pelatihan selesai, Anda dapat memilih untuk menghentikan pekerjaan penyetelan. Kinerja model dievaluasi terhadap metrik objektif terbaik.

Untuk menghentikan pekerjaan penyetelan secara manual, gunakan StopHyperParameterTuningJobAPI dan berikan nama pekerjaan penyetelan yang akan dihentikan.

Tune Beberapa Algoritma dengan Optimasi Hyperparameter untuk Menemukan Model Terbaik

Untuk membuat pekerjaan optimasi hyperparameter (HPO) baru dengan Amazon SageMaker yang menyetel beberapa algoritme, Anda harus menyediakan pengaturan pekerjaan yang berlaku untuk semua algoritme yang akan diuji dan definisi pelatihan untuk masing-masing algoritme ini. Anda juga harus menentukan sumber daya yang ingin Anda gunakan untuk pekerjaan penyetelan.

- Pengaturan pekerjaan untuk mengonfigurasi termasuk start hangat, penghentian awal, dan strategi penyetelan. Awal yang hangat dan penghentian awal hanya tersedia saat menyetel satu algoritma.
- Definisi pekerjaan pelatihan untuk menentukan nama, sumber algoritme, metrik objektif, dan rentang nilai, bila diperlukan, untuk mengonfigurasi kumpulan nilai hiperparameter untuk setiap pekerjaan pelatihan. Ini mengkonfigurasi saluran untuk input data, lokasi output data, dan lokasi penyimpanan pos pemeriksaan untuk setiap pekerjaan pelatihan. Definisi ini juga mengonfigurasi sumber daya yang akan diterapkan untuk setiap pekerjaan pelatihan, termasuk jenis dan jumlah instans, pelatihan spot terkelola, dan kondisi penghentian.
- Sumber daya pekerjaan tuning: untuk diterapkan, termasuk jumlah maksimum pekerjaan pelatihan bersamaan yang dapat dijalankan oleh pekerjaan tuning hyperparameter secara bersamaan dan jumlah maksimum pekerjaan pelatihan yang dapat dijalankan oleh pekerjaan tuning hyperparameter.

Mulai

Anda dapat membuat pekerjaan penyetelan hyperparameter baru, mengkloning pekerjaan, menambahkan, atau mengedit tag ke pekerjaan dari konsol. Anda juga dapat menggunakan fitur pencarian untuk mencari pekerjaan berdasarkan nama, waktu pembuatan, atau statusnya. Atau, Anda juga dapat melakukan pekerjaan tuning hyperparameter dengan API. SageMaker

- Di konsol: Untuk membuat pekerjaan baru, buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/, pilih Pekerjaan tuning Hyperparameter dari menu Pelatihan, lalu pilih Create hyperparameter tuning job. Kemudian ikuti langkah-langkah konfigurasi untuk membuat pekerjaan pelatihan untuk setiap algoritme yang ingin Anda gunakan. Langkah-langkah ini didokumentasikan dalam Buat Job Tuning Optimasi Hyperparameter untuk Satu atau Lebih Algoritma (Konsol) topik.

> **Note**
>
> Saat Anda memulai langkah-langkah konfigurasi, perhatikan bahwa fitur mulai hangat dan penghentian awal tidak tersedia untuk digunakan dengan HPO multi-algoritma. Jika Anda ingin menggunakan fitur-fitur ini, Anda hanya dapat menyetel satu algoritma pada satu waktu.

- Dengan API: Untuk petunjuk penggunaan SageMaker API guna membuat tugas penyetelan hyperparameter, lihat Contoh: Hyperparameter Tuning Job. Saat Anda menelepon `CreateHyperParameterTuningJob`untuk menyetel beberapa algoritme, Anda harus memberikan daftar definisi pelatihan menggunakan `TrainingJobDefinitions`alih-alih menentukan satu. TrainingJobDefinition Anda harus memberikan pengaturan pekerjaan yang berlaku untuk semua algoritme yang akan diuji dan definisi pelatihan untuk masing-masing algoritme ini. Anda juga harus menentukan sumber daya yang ingin Anda gunakan untuk pekerjaan penyetelan. Pilih hanya satu dari jenis definisi ini tergantung pada jumlah algoritma yang sedang disetel.

Topik

- Buat Job Tuning Optimasi Hyperparameter untuk Satu atau Lebih Algoritma (Konsol)
- Kelola Pekerjaan Tuning dan Pelatihan Hyperparameter

Buat Job Tuning Optimasi Hyperparameter untuk Satu atau Lebih Algoritma (Konsol)

Panduan ini menunjukkan kepada Anda cara membuat pekerjaan penyetelan optimasi hyperparameter (HPO) baru untuk satu atau lebih algoritme. Untuk membuat pekerjaan HPO, tentukan pengaturan untuk pekerjaan penyetelan, dan buat definisi pekerjaan pelatihan untuk setiap algoritme yang disetel. Selanjutnya, konfigurasikan sumber daya untuk dan buat pekerjaan penyetelan. Bagian berikut memberikan rincian tentang cara menyelesaikan setiap langkah. Kami memberikan contoh cara menyetel beberapa algoritme menggunakan SageMaker SDK for Python client di akhir panduan ini.

Komponen pekerjaan tuning

Pekerjaan penyetelan HPO berisi tiga komponen berikut:

- Menyetel pengaturan pekerjaan

- Definisi pekerjaan pelatihan
- Konfigurasi pekerjaan penyetelan

Cara komponen-komponen ini disertakan dalam pekerjaan penyetelan HPO Anda tergantung pada apakah pekerjaan penyetelan Anda berisi satu atau beberapa algoritme pelatihan. Panduan berikut menjelaskan masing-masing komponen dan memberikan contoh dari kedua jenis pekerjaan tuning.

Menyetel pengaturan pekerjaan

Pengaturan pekerjaan tuning Anda diterapkan di semua algoritme dalam pekerjaan penyetelan HPO. Mulai hangat dan penghentian awal hanya tersedia saat Anda menyetel satu algoritma. Setelah Anda menentukan pengaturan pekerjaan, Anda dapat membuat definisi pelatihan individual untuk setiap algoritme atau variasi yang ingin Anda sesuaikan.

Awal yang hangat

Jika Anda mengkloning pekerjaan ini, Anda dapat menggunakan hasil dari pekerjaan penyetelan sebelumnya untuk meningkatkan kinerja pekerjaan penyetelan baru ini. Ini adalah fitur awal yang hangat, dan hanya tersedia saat menyetel satu algoritma. Dengan opsi start hangat, Anda dapat memilih hingga lima pekerjaan penyetelan hyperparameter sebelumnya untuk digunakan. Atau, Anda dapat menggunakan pembelajaran transfer untuk menambahkan data tambahan ke pekerjaan penyetelan induk. Saat Anda memilih opsi ini, Anda memilih satu pekerjaan penyetelan sebelumnya sebagai induknya.

> **Note**
>
> Warm start hanya kompatibel dengan pekerjaan tuning yang dibuat setelah 1 Oktober 2018. Untuk informasi selengkapnya, lihat Menjalankan pekerjaan awal yang hangat.

Berhenti lebih awal

Untuk mengurangi waktu komputasi dan menghindari overfitting model Anda, Anda dapat menghentikan pekerjaan pelatihan lebih awal. Berhenti lebih awal sangat membantu ketika pekerjaan pelatihan tidak mungkin meningkatkan metrik objektif terbaik saat ini dari pekerjaan penyetelan hiperparameter. Seperti warm start, fitur ini hanya tersedia saat menyetel satu algoritma. Ini adalah fitur otomatis tanpa opsi konfigurasi, dan dinonaktifkan secara default. Untuk informasi selengkapnya tentang cara kerja penghentian dini, algoritme yang mendukungnya, dan cara menggunakannya dengan algoritme Anda sendiri, lihat Hentikan Pekerjaan Pelatihan Lebih Dini.

Strategi penyetelan

Strategi penyetelan dapat berupa acak, Bayesian, atau. Hyperband Pilihan ini menentukan bagaimana algoritma penyetelan otomatis mencari rentang hyperparameter tertentu yang dipilih pada langkah selanjutnya. Pencarian acak memilih kombinasi nilai acak dari rentang yang ditentukan dan dapat dijalankan secara berurutan atau paralel. Optimasi Bayesian memilih nilai berdasarkan apa yang mungkin mendapatkan hasil terbaik sesuai dengan sejarah yang diketahui dari pilihan sebelumnya. Hyperbandmenggunakan strategi multi-fidelity yang secara dinamis mengalokasikan sumber daya untuk pekerjaan yang dimanfaatkan dengan baik dan secara otomatis menghentikan mereka yang berkinerja buruk. Konfigurasi baru yang dimulai setelah menghentikan konfigurasi lain dipilih secara acak.

Hyperbandhanya dapat digunakan dengan algoritma iteratif, atau algoritma yang menjalankan langkah-langkah dalam iterasi, seperti XGBoostatau Random Cut Forest. Hyperbandtidak dapat digunakan dengan algoritme non-iteratif, seperti pohon keputusan atau Tetangga K-terdekat. Untuk informasi selengkapnya tentang strategi penelusuran, lihat Cara Kerja Penyetelan Hyperparameter.

> **Note**
>
> Hyperbandmenggunakan mekanisme internal canggih untuk menerapkan penghentian dini. Oleh karena itu, saat Anda menggunakan fitur penghentian awal Hyperband internal, parameter `TrainingJobEarlyStoppingType` di `HyperParameterTuningJobConfig` API harus disetel ke`OFF`.

Tag

Untuk membantu mengelola pekerjaan penyetelan, Anda dapat memasukkan tag sebagai pasangan nilai kunci untuk menetapkan metadata ke pekerjaan penyetelan. Nilai dalam pasangan kunci-nilai tidak diperlukan. Anda dapat menggunakan kunci tanpa nilai. Untuk melihat kunci yang terkait dengan pekerjaan, pilih tab Tag pada halaman detail untuk menyetel pekerjaan. Untuk informasi selengkapnya tentang penggunaan tag untuk menyetel pekerjaan, lihatKelola Pekerjaan Tuning dan Pelatihan Hyperparameter.

Definisi pekerjaan pelatihan

Untuk membuat definisi pekerjaan pelatihan, Anda harus mengonfigurasi algoritme dan parameter, menentukan input dan output data, dan mengonfigurasi sumber daya. Berikan setidaknya satu

`TrainingJobDefinition`untuk setiap pekerjaan penyetelan HPO. Setiap definisi pelatihan menentukan konfigurasi untuk suatu algoritma.

Untuk membuat beberapa definisi untuk pekerjaan pelatihan Anda, Anda dapat mengkloning definisi pekerjaan. Mengkloning pekerjaan dapat menghemat waktu karena menyalin semua pengaturan pekerjaan, termasuk saluran data dan lokasi penyimpanan Amazon S3 untuk artefak keluaran. Anda dapat mengedit pekerjaan kloning untuk mengubah apa yang Anda butuhkan untuk kasus penggunaan Anda.

Topik

- Konfigurasikan algoritma dan parameter
- Tentukan input dan output data
- Konfigurasikan sumber daya pekerjaan pelatihan
- Menambahkan atau mengkloning pekerjaan pelatihan

Konfigurasikan algoritma dan parameter

Daftar berikut menjelaskan apa yang Anda butuhkan untuk mengonfigurasi kumpulan nilai hyperparameter untuk setiap pekerjaan pelatihan.

- Nama untuk pekerjaan tuning Anda
- Izin untuk mengakses layanan
- Parameter untuk opsi algoritma apa pun
- Metrik objektif
- Kisaran nilai hyperparameter, bila diperlukan

Nama

Berikan nama unik untuk definisi pelatihan.

Izin

Amazon SageMaker memerlukan izin untuk memanggil layanan lain atas nama Anda. Pilih peran AWS Identity and Access Management (IAM), atau biarkan AWS membuat peran dengan kebijakan `AmazonSageMakerFullAccess` IAM terlampir.

Pengaturan keamanan opsional

Pengaturan isolasi jaringan mencegah kontainer melakukan panggilan jaringan keluar. Ini diperlukan untuk penawaran pembelajaran AWS Marketplace mesin.

Anda juga dapat memilih untuk menggunakan virtual private cloud (VPC).

> **Note**
>
> Enkripsi antar kontainer hanya tersedia saat Anda membuat definisi pekerjaan dari API.

Opsi algoritma

Anda dapat memilih algoritma bawaan, algoritme Anda sendiri, wadah Anda sendiri dengan algoritme, atau Anda dapat berlangganan algoritme dariAWS Marketplace.

- Jika Anda memilih algoritme bawaan, ia memiliki informasi gambar Amazon Elastic Container Registry (Amazon ECR) yang telah diisi sebelumnya.
- Jika Anda memilih wadah Anda sendiri, Anda harus menentukan informasi gambar (Amazon ECR). Anda dapat memilih mode input untuk algoritma sebagai file atau pipa.
- Jika Anda berencana untuk memasok data menggunakan file CSV dari Amazon S3, Anda harus memilih file tersebut.

Metrik

Saat Anda memilih algoritme bawaan, metrik disediakan untuk Anda. Jika Anda memilih algoritma Anda sendiri, Anda harus menentukan metrik Anda. Anda dapat menentukan hingga 20 metrik untuk dipantau oleh pekerjaan penyetelan Anda. Anda harus memilih satu metrik sebagai metrik objektif. Untuk informasi selengkapnya tentang cara menentukan metrik untuk pekerjaan penyetelan, lihatTentukan metrik.

Metrik objektif

Untuk menemukan pekerjaan pelatihan terbaik, tetapkan metrik objektif dan apakah akan memaksimalkan atau meminimalkannya. Setelah pekerjaan pelatihan selesai, Anda dapat melihat halaman detail pekerjaan penyetelan. Halaman detail memberikan ringkasan pekerjaan pelatihan terbaik yang ditemukan menggunakan metrik objektif ini.

Konfigurasi hyperparameter

Saat Anda memilih algoritma bawaan, nilai default untuk hyperparameternya ditetapkan untuk Anda, menggunakan rentang yang dioptimalkan untuk algoritme yang sedang disetel. Anda dapat mengubah nilai-nilai ini sesuai keinginan Anda. Misalnya, alih-alih rentang, Anda dapat menetapkan nilai tetap untuk hyperparameter dengan menyetel tipe parameter ke statis. Setiap algoritma memiliki parameter wajib dan opsional yang berbeda. Untuk informasi selengkapnya, lihat Praktik Terbaik untuk Penyetelan Hyperparameter dan Menentukan Rentang Hyperparameter.

Tentukan input dan output data

Setiap definisi pekerjaan pelatihan untuk pekerjaan penyetelan harus mengonfigurasi saluran untuk input data, lokasi keluaran data, dan secara opsional, lokasi penyimpanan pos pemeriksaan apa pun untuk setiap pekerjaan pelatihan.

Konfigurasi data masukan

Data input ditentukan oleh saluran. Setiap saluran lokasi sumbernya sendiri (Amazon S3 atau Amazon Elastic File System), opsi kompresi, dan format. Anda dapat menentukan hingga 20 saluran sumber input. Jika algoritme yang Anda pilih mendukung beberapa saluran input, Anda juga dapat menentukannya. Misalnya, saat Anda menggunakan buku catatan prediksi XGBoost churn, Anda dapat menambahkan dua saluran: kereta api dan validasi.

Konfigurasi pos pemeriksaan

Pos pemeriksaan dibuat secara berkala selama pelatihan. Agar pos pemeriksaan disimpan, Anda harus memilih lokasi Amazon S3. Pos pemeriksaan digunakan dalam pelaporan metrik, dan juga digunakan untuk melanjutkan pekerjaan pelatihan spot yang dikelola. Untuk informasi selengkapnya, lihat Menggunakan Pos Pemeriksaan di Amazon SageMaker.

Konfigurasi data keluaran

Tentukan lokasi Amazon S3 untuk artefak pekerjaan pelatihan yang akan disimpan. Anda memiliki opsi untuk menambahkan enkripsi ke output menggunakan kunci AWS Key Management Service (AWS KMS).

Konfigurasikan sumber daya pekerjaan pelatihan

Setiap definisi pekerjaan pelatihan untuk pekerjaan penyetelan harus mengonfigurasi sumber daya yang akan diterapkan, termasuk jenis dan jumlah instans, pelatihan tempat terkelola, dan kondisi penghentian.

Konfigurasi sumber daya

Setiap definisi pelatihan dapat memiliki konfigurasi sumber daya yang berbeda. Anda memilih jenis instance dan jumlah node.

Pelatihan spot terkelola

Anda dapat menghemat biaya komputer untuk pekerjaan jika Anda memiliki fleksibilitas dalam waktu mulai dan akhir dengan memungkinkan SageMaker untuk menggunakan kapasitas cadangan untuk menjalankan pekerjaan. Untuk informasi selengkapnya, lihat Gunakan Pelatihan Spot Terkelola di Amazon SageMaker.

Kondisi berhenti

Kondisi berhenti menentukan durasi maksimum yang diizinkan untuk setiap pekerjaan pelatihan.

Menambahkan atau mengkloning pekerjaan pelatihan

Setelah Anda membuat definisi pekerjaan pelatihan untuk pekerjaan tuning, Anda akan kembali ke panel Training Job Definition (s). Panel ini adalah tempat Anda dapat membuat definisi pekerjaan pelatihan tambahan untuk melatih algoritme tambahan. Anda dapat memilih definisi Tambahkan pekerjaan pelatihan dan mengerjakan langkah-langkah untuk menentukan pekerjaan pelatihan lagi.

Atau, untuk mereplikasi definisi pekerjaan pelatihan yang ada dan mengeditnya untuk algoritma baru, pilih Clone dari menu Action. Opsi klon dapat menghemat waktu karena menyalin semua pengaturan pekerjaan, termasuk saluran data dan lokasi penyimpanan Amazon S3. Untuk informasi lebih lanjut tentang kloning, lihatKelola Pekerjaan Tuning dan Pelatihan Hyperparameter.

Konfigurasi pekerjaan penyetelan

Batasan sumber daya

Anda dapat menentukan jumlah maksimum pekerjaan pelatihan bersamaan yang dapat dijalankan oleh pekerjaan penyetelan hiperparameter secara bersamaan (paling banyak 10). Anda juga dapat menentukan jumlah maksimum pekerjaan pelatihan yang dapat dijalankan oleh pekerjaan tuning hyperparameter (paling banyak 500). Jumlah pekerjaan paralel tidak boleh melebihi jumlah node yang Anda minta di semua definisi pelatihan Anda. Jumlah total pekerjaan tidak dapat melebihi jumlah pekerjaan yang diharapkan dijalankan oleh definisi Anda.

Tinjau pengaturan pekerjaan, definisi pekerjaan pelatihan, dan batas sumber daya. Kemudian pilih Create hyperparameter tuning job.

Contoh pekerjaan penyetelan HPO

Untuk menjalankan pekerjaan pelatihan optimasi hyperparameter (HPO), pertama-tama buat definisi pekerjaan pelatihan untuk setiap algoritme yang sedang disetel. Selanjutnya, tentukan pengaturan pekerjaan penyetelan dan konfigurasikan sumber daya untuk pekerjaan penyetelan. Akhirnya, jalankan pekerjaan tuning.

Jika pekerjaan penyetelan HPO Anda berisi algoritme pelatihan tunggal, fungsi SageMaker penyetelan akan memanggil `HyperparameterTuner` API secara langsung dan meneruskan parameter Anda. Jika pekerjaan penyetelan HPO Anda berisi beberapa algoritma pelatihan, fungsi penyetelan Anda akan memanggil `create` fungsi API. `HyperparameterTuner` `create`Fungsi ini memberi tahu API untuk mengharapkan kamus yang berisi satu atau lebih estimator.

Pada bagian berikut, contoh kode menunjukkan cara menyetel pekerjaan yang berisi algoritme pelatihan tunggal atau beberapa algoritme menggunakan. SageMaker Python SDK

Buat definisi pekerjaan pelatihan

Saat Anda membuat pekerjaan penyetelan yang mencakup beberapa algoritme pelatihan, konfigurasi pekerjaan penyetelan Anda akan mencakup estimator dan metrik serta parameter lain untuk pekerjaan pelatihan Anda. Oleh karena itu, Anda perlu membuat definisi pekerjaan pelatihan terlebih dahulu, dan kemudian mengonfigurasi pekerjaan penyetelan Anda.

Contoh kode berikut menunjukkan bagaimana untuk mengambil dua SageMaker kontainer yang berisi built-in algoritma XGBoostdan. Linear Learner Jika pekerjaan penyetelan Anda hanya berisi satu algoritma pelatihan, hilangkan salah satu wadah dan salah satu penaksir.

```
import sagemaker
from sagemaker import image_uris

from sagemaker.estimator import Estimator

sess = sagemaker.Session()
region = sess.boto_region_name
role = sagemaker.get_execution_role()

bucket = sess.default_bucket()
prefix = "sagemaker/multi-algo-hpo"

# Define the training containers and intialize the estimators
xgb_container = image_uris.retrieve("xgboost", region, "latest")
ll_container = image_uris.retrieve("linear-learner", region, "latest")
```

```
xgb_estimator = Estimator(
    xgb_container,
    role=role,
    instance_count=1,
    instance_type="ml.m4.xlarge",
    output_path='s3://{}/{}/xgb_output".format(bucket, prefix)',
    sagemaker_session=sess,
)

ll_estimator = Estimator(
    ll_container,
    role,
    instance_count=1,
    instance_type="ml.c4.xlarge",
    output_path="s3://{}/{}/ll_output".format(bucket, prefix),
    sagemaker_session=sess,
)

# Set static hyperparameters
ll_estimator.set_hyperparameters(predictor_type="binary_classifier")
xgb_estimator.set_hyperparameters(
    eval_metric="auc",
    objective="binary:logistic",
    num_round=100,
    rate_drop=0.3,
    tweedie_variance_power=1.4,
)
```

Selanjutnya, tentukan data masukan Anda dengan menentukan kumpulan data pelatihan, validasi, dan pengujian, seperti yang ditunjukkan pada contoh kode berikut. Contoh ini menunjukkan cara menyetel beberapa algoritma pelatihan.

```
training_data = sagemaker.inputs.TrainingInput(
    s3_data="s3://{}/{}/train".format(bucket, prefix), content_type="csv"
)
validation_data = sagemaker.inputs.TrainingInput(
    s3_data="s3://{}/{}/validate".format(bucket, prefix), content_type="csv"
)
test_data = sagemaker.inputs.TrainingInput(
    s3_data="s3://{}/{}/test".format(bucket, prefix), content_type="csv"
)
```

```
train_inputs = {
    "estimator-1": {
        "train": training_data,
        "validation": validation_data,
        "test": test_data,
    },
    "estimator-2": {
        "train": training_data,
        "validation": validation_data,
        "test": test_data,
    },
}
```

Jika algoritma tuning Anda hanya berisi satu algoritma pelatihan, Anda `train_inputs` harus berisi hanya satu estimator.

Anda harus mengunggah input untuk kumpulan data pelatihan, validasi, dan pelatihan ke bucket Amazon S3 sebelum menggunakannya dalam pekerjaan penyetelan HPO.

Tentukan sumber daya dan pengaturan untuk pekerjaan penyetelan Anda

Bagian ini menunjukkan cara menginisialisasi tuner, menentukan sumber daya, dan menentukan pengaturan pekerjaan untuk pekerjaan penyetelan Anda. Jika pekerjaan penyetelan Anda berisi beberapa algoritme pelatihan, pengaturan ini diterapkan ke semua algoritme yang terdapat di dalam pekerjaan penyetelan Anda. Bagian ini memberikan dua contoh kode untuk mendefinisikan tuner. Contoh kode menunjukkan cara mengoptimalkan algoritme pelatihan tunggal diikuti dengan contoh cara menyetel beberapa algoritme pelatihan.

Tune algoritma pelatihan tunggal

Contoh kode berikut menunjukkan cara menginisialisasi tuner dan mengatur rentang hyperparameter untuk satu algoritma SageMaker bawaan,. XGBoost

```
from sagemaker.tuner import HyperparameterTuner
from sagemaker.parameter import ContinuousParameter, IntegerParameter

hyperparameter_ranges = {
    "max_depth": IntegerParameter(1, 10),
    "eta": ContinuousParameter(0.1, 0.3),
}

objective_metric_name = "validation:accuracy"
```

```
tuner = HyperparameterTuner(
    xgb_estimator,
    objective_metric_name,
    hyperparameter_ranges,
    objective_type="Maximize",
    max_jobs=5,
    max_parallel_jobs=2,
)
```

Tune beberapa algoritma pelatihan

Setiap pekerjaan pelatihan memerlukan konfigurasi yang berbeda, dan ini ditentukan menggunakan kamus. Contoh kode berikut menunjukkan cara menginisialisasi tuner dengan konfigurasi untuk dua algoritma SageMaker bawaan, dan. XGBoost Linear Learner Contoh kode juga menunjukkan cara mengatur strategi penyetelan dan pengaturan pekerjaan lainnya, seperti sumber daya komputasi untuk pekerjaan penyetelan. Contoh kode berikut menggunakan`metric_definitions_dict`, yang opsional.

```
from sagemaker.tuner import HyperparameterTuner
from sagemaker.parameter import ContinuousParameter, IntegerParameter

# Initialize your tuner
tuner = HyperparameterTuner.create(
    estimator_dict={
        "estimator-1": xgb_estimator,
        "estimator-2": ll_estimator,
    },
    objective_metric_name_dict={
        "estimator-1": "validation:auc",
        "estimator-2": "test:binary_classification_accuracy",
    },
    hyperparameter_ranges_dict={
        "estimator-1": {"eta": ContinuousParameter(0.1, 0.3)},
        "estimator-2": {"learning_rate": ContinuousParameter(0.1, 0.3)},
    },
    metric_definitions_dict={
        "estimator-1": [
            {"Name": "validation:auc", "Regex": "Overall test accuracy: (.*?);"}
        ],
        "estimator-2": [
            {
                "Name": "test:binary_classification_accuracy",
                "Regex": "Overall test accuracy: (.*?);",
```

```
            }
        ],
    },
    strategy="Bayesian",
    max_jobs=10,
    max_parallel_jobs=3,
)
```

Jalankan pekerjaan penyetelan HPO Anda

Sekarang Anda dapat menjalankan pekerjaan tuning Anda dengan meneruskan input pelatihan Anda ke `fit` fungsi kelas. `HyperparameterTuner` Contoh kode berikut menunjukkan cara meneruskan `train_inputs` parameter, yang didefinisikan dalam contoh kode sebelumnya, ke tuner Anda.

```
tuner.fit(inputs=train_inputs, include_cls_metadata ={}, estimator_kwargs ={})
```

Kelola Pekerjaan Tuning dan Pelatihan Hyperparameter

Pekerjaan tuning dapat berisi banyak pekerjaan pelatihan dan menciptakan dan mengelola pekerjaan ini dan definisi mereka dapat menjadi tugas yang kompleks dan berat. SageMaker menyediakan alat untuk membantu memfasilitasi pengelolaan pekerjaan ini. Pekerjaan tuning yang telah Anda jalankan dapat diakses dari SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/. Pilih pekerjaan tuning Hyperparameter dari menu Pelatihan untuk melihat daftar. Halaman ini juga merupakan tempat Anda memulai prosedur untuk membuat pekerjaan penyetelan baru dengan memilih Create hyperparameter tuning job.

Untuk melihat pekerjaan pelatihan menjalankan bagian dari pekerjaan penyetelan, pilih salah satu pekerjaan penyetelan hiperparameter dari daftar. Tab pada halaman pekerjaan penyetelan memungkinkan Anda memeriksa pekerjaan pelatihan, definisinya, tag dan konfigurasi yang digunakan untuk pekerjaan penyetelan, dan pekerjaan pelatihan terbaik yang ditemukan selama penyetelan. Anda dapat memilih pekerjaan pelatihan terbaik atau salah satu pekerjaan pelatihan lainnya yang termasuk dalam pekerjaan tuning untuk melihat semua pengaturan mereka. Dari sini Anda dapat membuat model yang menggunakan nilai hyperparameter yang ditemukan oleh pekerjaan pelatihan dengan memilih Buat Model atau Anda dapat mengkloning pekerjaan pelatihan dengan memilih Clone.

Kloning

Anda dapat menghemat waktu dengan mengkloning pekerjaan pelatihan yang termasuk dalam pekerjaan tuning hyperparameter. Kloning menyalin semua pengaturan pekerjaan, termasuk saluran

data, lokasi penyimpanan S3 untuk artefak keluaran. Anda dapat melakukan ini untuk pekerjaan pelatihan yang telah Anda jalankan dari halaman pekerjaan tuning, seperti yang baru saja dijelaskan, atau ketika Anda membuat definisi pekerjaan pelatihan tambahan saat membuat pekerjaan tuning hyperparameter, seperti yang dijelaskan dalam Menambahkan atau mengkloning pekerjaan pelatihan langkah prosedur itu.

Pemberian tag

Penyetelan Model Otomatis meluncurkan beberapa pekerjaan pelatihan dalam satu pekerjaan penyetelan induk tunggal untuk menemukan bobot ideal hiperparameter model. Tag dapat ditambahkan ke pekerjaan penyetelan induk seperti yang dijelaskan di Komponen pekerjaan tuning bagian dan tag ini kemudian disebarkan ke pekerjaan pelatihan individu di bawahnya. Pelanggan dapat menggunakan tag ini untuk tujuan, seperti alokasi biaya atau kontrol akses. Untuk menambahkan tag menggunakan SageMaker SDK, gunakan `AddTags`API. Untuk informasi selengkapnya tentang penggunaan penandaan untuk AWS sumber daya, lihat Menandai AWS sumber daya.

Contoh: Hyperparameter Tuning Job

Contoh ini menunjukkan cara membuat notebook baru untuk mengonfigurasi dan meluncurkan pekerjaan tuning hyperparameter. Pekerjaan tuning menggunakan Algoritma XGBoost untuk melatih model untuk memprediksi apakah pelanggan akan mendaftar untuk deposito berjangka di bank setelah dihubungi melalui telepon.

Anda menggunakan SDK tingkat rendah untuk Python (Boto3) untuk mengonfigurasi dan meluncurkan pekerjaan tuning hyperparameter, dan untuk memantau status pekerjaan tuning hyperparameter. AWS Management Console Anda juga dapat menggunakan Amazon SageMaker Python SDK SageMaker tingkat tinggi Amazon untuk mengonfigurasi, menjalankan, memantau, dan menganalisis pekerjaan penyetelan hyperparameter. Untuk informasi lebih lanjut, lihat https://github.com/aws/ sagemaker-python-sdk.

Prasyarat

Untuk menjalankan kode dalam contoh ini, Anda perlu

- AWSAkun dan pengguna administrator
- Bucket Amazon S3 untuk menyimpan kumpulan data pelatihan Anda dan artefak model yang dibuat selama pelatihan

- Sebuah instance SageMaker notebook yang sedang berjalan

Topik

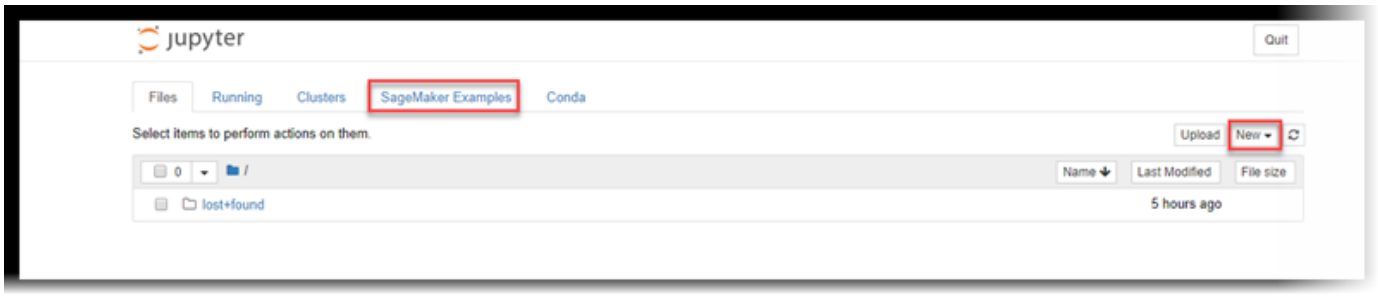
- Membuat Instance Notebook
- Dapatkan Klien Amazon SageMaker Boto 3
- Dapatkan Peran SageMaker Eksekusi
- Menggunakan bucket Amazon S3 untuk input dan output
- Unduh, Siapkan, dan Unggah Data Pelatihan
- Konfigurasikan dan Luncurkan Pekerjaan Tuning Hyperparameter
- Hapus

Membuat Instance Notebook

Buat notebook Jupyter yang berisi lingkungan pra-instal dengan instalasi Anaconda default dan Python3.

Untuk membuat notebook Jupyter

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Buka instance notebook yang sedang berjalan, dengan memilih Buka di samping namanya. Halaman server notebook Jupyter muncul:



3. Untuk membuat buku catatan, pilih File, Baru, dan conda_python3.
4. Beri nama buku catatan.

Langkah Selanjutnya

Dapatkan Klien Amazon SageMaker Boto 3

Dapatkan Klien Amazon SageMaker Boto 3

Impor Amazon SageMaker Python SDK,AWS SDK for Python (Boto3), dan pustaka Python lainnya. Di buku catatan Jupyter baru, tempel kode berikut ke sel pertama:

```
import sagemaker
import boto3

import numpy as np                                # For performing matrix operations
 and numerical processing
import pandas as pd                               # For manipulating tabular data
from time import gmtime, strftime
import os

region = boto3.Session().region_name
smclient = boto3.Session().client('sagemaker')
```

Sel kode sebelumnya mendefinisikan `region` dan `smclient` objek yang akan Anda gunakan untuk memanggil algoritma XGBoost bawaan dan mengatur pekerjaan tuning hyperparameter. SageMaker

Langkah Selanjutnya

[Dapatkan Peran SageMaker Eksekusi]

Dapatkan Peran SageMaker Eksekusi

Dapatkan peran eksekusi untuk instance notebook. Ini adalah peran IAM yang Anda buat untuk instance notebook Anda.

Untuk menemukan ARN dari peran eksekusi IAM yang dilampirkan ke instance notebook:

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi kiri, pilih Notebook lalu instance Notebook.
3. Dari daftar notebook, pilih buku catatan yang ingin Anda lihat.
4. ARN ada di bagian Izin dan enkripsi.

Atau, pengguna Amazon SageMaker Python SDK dapat mengambil ARN dari peran eksekusi yang dilampirkan ke profil pengguna mereka atau instance notebook dengan menjalankan kode berikut:

```
from sagemaker import get_execution_role
```

```
role = get_execution_role()
print(role)
```

Untuk informasi selengkapnya tentang penggunaan `get_execution_role` di Amazon SageMaker Python SDK, lihat Sesi. Untuk informasi lebih lanjut tentang peran, lihat SageMaker Peran.

Langkah Selanjutnya

Menggunakan bucket Amazon S3 untuk input dan output

Menggunakan bucket Amazon S3 untuk input dan output

Siapkan bucket S3 untuk mengunggah kumpulan data pelatihan dan menyimpan data keluaran pelatihan untuk pekerjaan penyetelan hyperparameter Anda.

Untuk menggunakan bucket S3 default

Gunakan kode berikut untuk menentukan bucket S3 default yang dialokasikan untuk sesi Anda SageMaker . `prefix`adalah jalur dalam ember tempat SageMaker menyimpan data untuk pekerjaan pelatihan saat ini.

```
sess = sagemaker.Session()
bucket = sess.default_bucket() # Set a default S3 bucket
prefix = 'DEMO-automatic-model-tuning-xgboost-dm'
```

Untuk menggunakan bucket S3 tertentu (Opsional)

Jika Anda ingin menggunakan bucket S3 tertentu, gunakan kode berikut dan ganti string dengan nama yang tepat dari bucket S3. Nama ember harus berisi**`sagemaker`**, dan unik secara global. Bucket harus berada di AWS Region yang sama dengan instance notebook yang Anda gunakan untuk contoh ini.

```
bucket = "sagemaker-your-preferred-s3-bucket"

sess = sagemaker.Session(
    default_bucket = bucket
)
```

> **Note**
>
> Nama bucket tidak perlu berisi **`sagemaker`** jika peran IAM yang Anda gunakan untuk menjalankan tugas penyetelan hyperparameter memiliki kebijakan yang memberikan izin. `S3FullAccess`

Langkah Selanjutnya

Unduh, Siapkan, dan Unggah Data Pelatihan

Unduh, Siapkan, dan Unggah Data Pelatihan

Untuk contoh ini, Anda menggunakan kumpulan data pelatihan informasi tentang nasabah bank yang mencakup pekerjaan pelanggan, status perkawinan, dan bagaimana mereka dihubungi selama kampanye pemasaran langsung bank. Untuk menggunakan kumpulan data untuk pekerjaan penyetelan hyperparameter, Anda mengunduhnya, mengubah data, dan kemudian mengunggahnya ke bucket Amazon S3.

Untuk informasi selengkapnya tentang kumpulan data dan transformasi data yang dilakukan contoh, lihat buku catatan HPO_XGBoost_Direct_Marketing_SageMaker_APIS di bagian Tuning Hyperparameter pada tab Contoh di instance notebook Anda. SageMaker

Unduh dan Jelajahi Dataset Pelatihan

Untuk mengunduh dan menjelajahi kumpulan data, jalankan kode berikut di buku catatan Anda:

```
!wget -N https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-
additional.zip
!unzip -o bank-additional.zip
data = pd.read_csv('./bank-additional/bank-additional-full.csv', sep=';')
pd.set_option('display.max_columns', 500)     # Make sure we can see all of the columns
pd.set_option('display.max_rows', 5)         # Keep the output on one page
data
```

Siapkan dan Unggah Data

Sebelum membuat pekerjaan tuning hyperparameter, siapkan data dan unggah ke bucket S3 di mana pekerjaan tuning hyperparameter dapat mengaksesnya.

Jalankan kode berikut di buku catatan Anda:

```
data['no_previous_contact'] = np.where(data['pdays'] == 999, 1, 0)
            # Indicator variable to capture when pdays takes a value of 999
data['not_working'] = np.where(np.in1d(data['job'], ['student', 'retired',
 'unemployed']), 1, 0)   # Indicator for individuals not actively employed
model_data = pd.get_dummies(data)
            # Convert categorical variables to sets of indicators
model_data
model_data = model_data.drop(['duration', 'emp.var.rate', 'cons.price.idx',
 'cons.conf.idx', 'euribor3m', 'nr.employed'], axis=1)

train_data, validation_data, test_data = np.split(model_data.sample(frac=1,
 random_state=1729), [int(0.7 * len(model_data)), int(0.9*len(model_data))])

pd.concat([train_data['y_yes'], train_data.drop(['y_no', 'y_yes'], axis=1)],
 axis=1).to_csv('train.csv', index=False, header=False)
pd.concat([validation_data['y_yes'], validation_data.drop(['y_no', 'y_yes'], axis=1)],
 axis=1).to_csv('validation.csv', index=False, header=False)
pd.concat([test_data['y_yes'], test_data.drop(['y_no', 'y_yes'], axis=1)],
 axis=1).to_csv('test.csv', index=False, header=False)

boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/
train.csv')).upload_file('train.csv')
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'validation/
validation.csv')).upload_file('validation.csv')
```

Langkah Selanjutnya

Konfigurasikan dan Luncurkan Pekerjaan Tuning Hyperparameter

Konfigurasikan dan Luncurkan Pekerjaan Tuning Hyperparameter

Hyperparameter adalah parameter tingkat tinggi yang mempengaruhi proses pembelajaran selama pelatihan model. Untuk mendapatkan prediksi model terbaik, Anda dapat mengoptimalkan konfigurasi hyperparameter atau menetapkan nilai hyperparameter. Proses menemukan konfigurasi optimal disebut tuning hyperparameter. Untuk mengonfigurasi dan meluncurkan pekerjaan penyetelan hyperparameter, selesaikan langkah-langkah dalam panduan ini.

Topik

- Pengaturan untuk pekerjaan tuning hyperparameter
- Konfigurasikan pekerjaan pelatihan
- Beri nama dan luncurkan pekerjaan penyetelan hyperparameter

- Memantau Kemajuan dari Hyperparameter Tuning Job
- Lihat Status Pekerjaan Pelatihan
- Lihat Training Job Terbaik

Pengaturan untuk pekerjaan tuning hyperparameter

Untuk menentukan pengaturan untuk pekerjaan tuning hyperparameter, tentukan objek JSON saat Anda membuat pekerjaan penyetelan. Berikan objek JSON ini sebagai nilai `HyperParameterTuningJobConfig` parameter ke `CreateHyperParameterTuningJob`API.

Dalam objek JSON ini, tentukan yang berikut ini:

Dalam objek JSON ini, Anda menentukan:

- `HyperParameterTuningJobObjective`— Metrik objektif yang digunakan untuk mengevaluasi kinerja pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning hyperparameter.
- `ParameterRanges`— Rentang nilai yang dapat digunakan oleh hyperparameter yang dapat disetel selama pengoptimalan. Lihat informasi yang lebih lengkap di Tentukan Rentang Hyperparameter
- `RandomSeed`— Nilai yang digunakan untuk menginisialisasi generator bilangan pseudo-acak. Menyetel seed acak akan memungkinkan strategi pencarian tuning hyperparameter menghasilkan konfigurasi yang lebih konsisten untuk pekerjaan penyetelan yang sama (opsional).
- `ResourceLimits`— Jumlah maksimum pekerjaan pelatihan dan pelatihan paralel yang dapat digunakan oleh pekerjaan tuning hyperparameter.

> **Note**
>
> Jika Anda menggunakan algoritme Anda sendiri untuk penyetelan hyperparameter, bukan algoritma SageMaker bawaan, Anda harus menentukan metrik untuk algoritme Anda. Untuk informasi selengkapnya, lihat Tentukan metrik.

Contoh kode berikut menunjukkan cara mengkonfigurasi pekerjaan tuning hyperparameter menggunakan algoritma XGBoost bawaan. Contoh kode menunjukkan cara menentukan rentang untuk`eta`,, `alpha`min_child_weight, dan `max_depth` hyperparameters. Untuk informasi lebih lanjut tentang ini dan hiperparameter lainnya, lihat Parameter XGBoost.

Dalam contoh kode ini, metrik objektif untuk pekerjaan tuning hyperparameter menemukan konfigurasi hyperparameter yang memaksimalkan. `validation:auc` SageMaker algoritma bawaan secara otomatis menulis metrik objektif ke CloudWatch Log. Contoh kode berikut juga menunjukkan cara mengatur`RandomSeed`.

```
tuning_job_config = {
    "ParameterRanges": {
      "CategoricalParameterRanges": [],
      "ContinuousParameterRanges": [
        {
          "MaxValue": "1",
          "MinValue": "0",
          "Name": "eta"
        },
        {
          "MaxValue": "2",
          "MinValue": "0",
          "Name": "alpha"
        },
        {
          "MaxValue": "10",
          "MinValue": "1",
          "Name": "min_child_weight"
        }
      ],
      "IntegerParameterRanges": [
        {
          "MaxValue": "10",
          "MinValue": "1",
          "Name": "max_depth"
        }
      ]
    },
    "ResourceLimits": {
      "MaxNumberOfTrainingJobs": 20,
      "MaxParallelTrainingJobs": 3
    },
    "Strategy": "Bayesian",
    "HyperParameterTuningJobObjective": {
      "MetricName": "validation:auc",
      "Type": "Maximize"
    },
    "RandomSeed" : 123
```

```
}
```

Konfigurasikan pekerjaan pelatihan

Pekerjaan tuning hyperparameter akan meluncurkan pekerjaan pelatihan untuk menemukan konfigurasi hiperparameter yang optimal. Pekerjaan pelatihan ini harus dikonfigurasi menggunakan SageMaker `CreateHyperParameterTuningJob`API.

Untuk mengonfigurasi pekerjaan pelatihan, tentukan objek JSON dan berikan sebagai nilai `TrainingJobDefinition` parameter di dalamnya`CreateHyperParameterTuningJob`.

Dalam objek JSON ini, Anda dapat menentukan yang berikut:

- `AlgorithmSpecification`— Jalur registri gambar Docker yang berisi algoritma pelatihan dan metadata terkait. Untuk menentukan algoritme, Anda dapat menggunakan algoritme buatan khusus Anda sendiri di dalam wadah Docker atau algoritme SageMaker bawaan (wajib).
- `InputDataConfig`— Konfigurasi input, termasuk`ChannelName`,`ContentType`, dan sumber data untuk data pelatihan dan pengujian Anda (wajib).
- `InputDataConfig`— Konfigurasi input, termasuk`ChannelName`,`ContentType`, dan sumber data untuk data pelatihan dan pengujian Anda (wajib).
- Lokasi penyimpanan untuk output algoritma. Tentukan bucket S3 tempat Anda ingin menyimpan output dari pekerjaan pelatihan.
- `RoleArn`— Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang SageMaker digunakan untuk melakukan tugas. Tugas termasuk membaca data input, mengunduh gambar Docker, menulis artefak model ke bucket S3, menulis CloudWatch log ke Amazon Logs, dan menulis metrik ke CloudWatch Amazon (wajib).
- `StoppingCondition`— Runtime maksimum dalam hitungan detik yang dapat dijalankan oleh pekerjaan pelatihan sebelum dihentikan. Nilai ini harus lebih besar dari waktu yang dibutuhkan untuk melatih model Anda (wajib).
- `MetricDefinitions`— Nama dan ekspresi reguler yang mendefinisikan metrik apa pun yang dipancarkan oleh pekerjaan pelatihan. Tentukan metrik hanya jika Anda menggunakan algoritme pelatihan khusus. Contoh dalam kode berikut menggunakan algoritma bawaan, yang sudah memiliki metrik yang ditentukan. Untuk informasi tentang mendefinisikan metrik (opsional), lihat. Tentukan metrik
- `TrainingImage`— Gambar kontainer Docker yang menentukan algoritma pelatihan (opsional).
- `StaticHyperParameters`— Nama dan nilai hyperparameters yang tidak disetel dalam pekerjaan tuning (opsional).

Contoh kode berikut menetapkan nilai statis untuk `eval_metric`num_round,`objective`,`rate_drop`,, dan `tweedie_variance_power` parameter algoritma Algoritma XGBoost bawaan.

SageMaker Python SDK v1

```
from sagemaker.amazon.amazon_estimator import get_image_uri
training_image = get_image_uri(region, 'xgboost', repo_version='1.0-1')

s3_input_train = 's3://{}/{}/train'.format(bucket, prefix)
s3_input_validation ='s3://{}/{}/validation/'.format(bucket, prefix)

training_job_definition = {
    "AlgorithmSpecification": {
      "TrainingImage": training_image,
      "TrainingInputMode": "File"
    },
    "InputDataConfig": [
      {
        "ChannelName": "train",
        "CompressionType": "None",
        "ContentType": "csv",
        "DataSource": {
          "S3DataSource": {
            "S3DataDistributionType": "FullyReplicated",
            "S3DataType": "S3Prefix",
            "S3Uri": s3_input_train
          }
        }
      },
      {
        "ChannelName": "validation",
        "CompressionType": "None",
        "ContentType": "csv",
        "DataSource": {
          "S3DataSource": {
            "S3DataDistributionType": "FullyReplicated",
            "S3DataType": "S3Prefix",
            "S3Uri": s3_input_validation
          }
        }
      }
    ],
```

```
    "OutputDataConfig": {
      "S3OutputPath": "s3://{}/{}/output".format(bucket,prefix)
    },
    "ResourceConfig": {
      "InstanceCount": 2,
      "InstanceType": "ml.c4.2xlarge",
      "VolumeSizeInGB": 10
    },
    "RoleArn": role,
    "StaticHyperParameters": {
      "eval_metric": "auc",
      "num_round": "100",
      "objective": "binary:logistic",
      "rate_drop": "0.3",
      "tweedie_variance_power": "1.4"
    },
    "StoppingCondition": {
      "MaxRuntimeInSeconds": 43200
    }
}
```

SageMaker Python SDK v2

```
training_image = sagemaker.image_uris.retrieve('xgboost', region, '1.0-1')

s3_input_train = 's3://{}/{}/train'.format(bucket, prefix)
s3_input_validation ='s3://{}/{}/validation/'.format(bucket, prefix)

training_job_definition = {
    "AlgorithmSpecification": {
      "TrainingImage": training_image,
      "TrainingInputMode": "File"
    },
    "InputDataConfig": [
      {
        "ChannelName": "train",
        "CompressionType": "None",
        "ContentType": "csv",
        "DataSource": {
          "S3DataSource": {
            "S3DataDistributionType": "FullyReplicated",
            "S3DataType": "S3Prefix",
            "S3Uri": s3_input_train
```

```
            }
          }
        },
        {
          "ChannelName": "validation",
          "CompressionType": "None",
          "ContentType": "csv",
          "DataSource": {
            "S3DataSource": {
              "S3DataDistributionType": "FullyReplicated",
              "S3DataType": "S3Prefix",
              "S3Uri": s3_input_validation
            }
          }
        }
    ],
    "OutputDataConfig": {
      "S3OutputPath": "s3://{}/{}/output".format(bucket,prefix)
    },
    "ResourceConfig": {
      "InstanceCount": 2,
      "InstanceType": "ml.c4.2xlarge",
      "VolumeSizeInGB": 10
    },
    "RoleArn": role,
    "StaticHyperParameters": {
      "eval_metric": "auc",
      "num_round": "100",
      "objective": "binary:logistic",
      "rate_drop": "0.3",
      "tweedie_variance_power": "1.4"
    },
    "StoppingCondition": {
      "MaxRuntimeInSeconds": 43200
    }
}
```

Beri nama dan luncurkan pekerjaan penyetelan hyperparameter

Setelah mengonfigurasi tugas tuning hyperparameter, Anda dapat meluncurkannya dengan memanggil API. `CreateHyperParameterTuningJob` Contoh kode berikut menggunakan

`tuning_job_config` dan`training_job_definition`. Ini didefinisikan dalam dua contoh kode sebelumnya untuk membuat pekerjaan tuning hyperparameter.

```
tuning_job_name = "MyTuningJob"
smclient.create_hyper_parameter_tuning_job(HyperParameterTuningJobName =
 tuning_job_name,
                                           HyperParameterTuningJobConfig =
 tuning_job_config,
                                           TrainingJobDefinition =
 training_job_definition)
```

Memantau Kemajuan dari Hyperparameter Tuning Job

Untuk memantau kemajuan pekerjaan penyetelan hyperparameter dan pekerjaan pelatihan yang diluncurkan, gunakan konsol Amazon. SageMaker

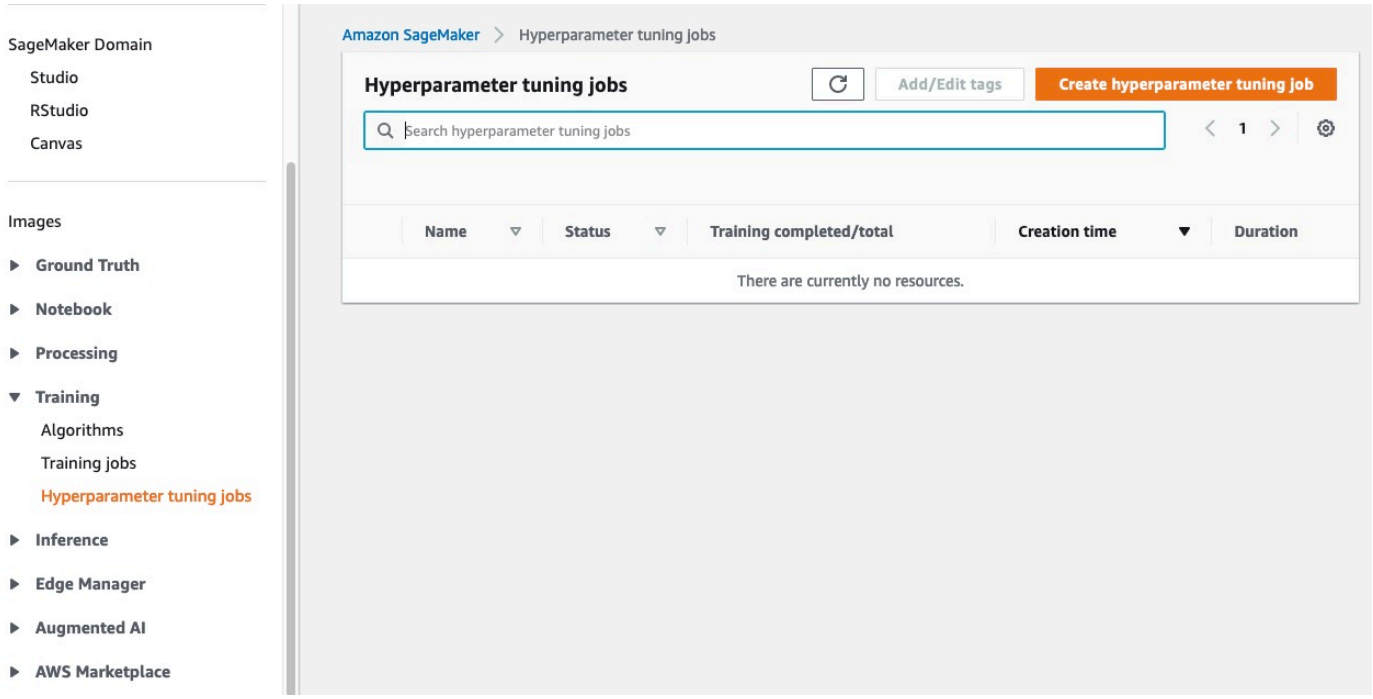
Topik

- Melihat Status dari Hyperparameter Tuning Job

Melihat Status dari Hyperparameter Tuning Job

Untuk melihat status pekerjaan tuning hyperparameter

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih pekerjaan tuning Hyperparameter.



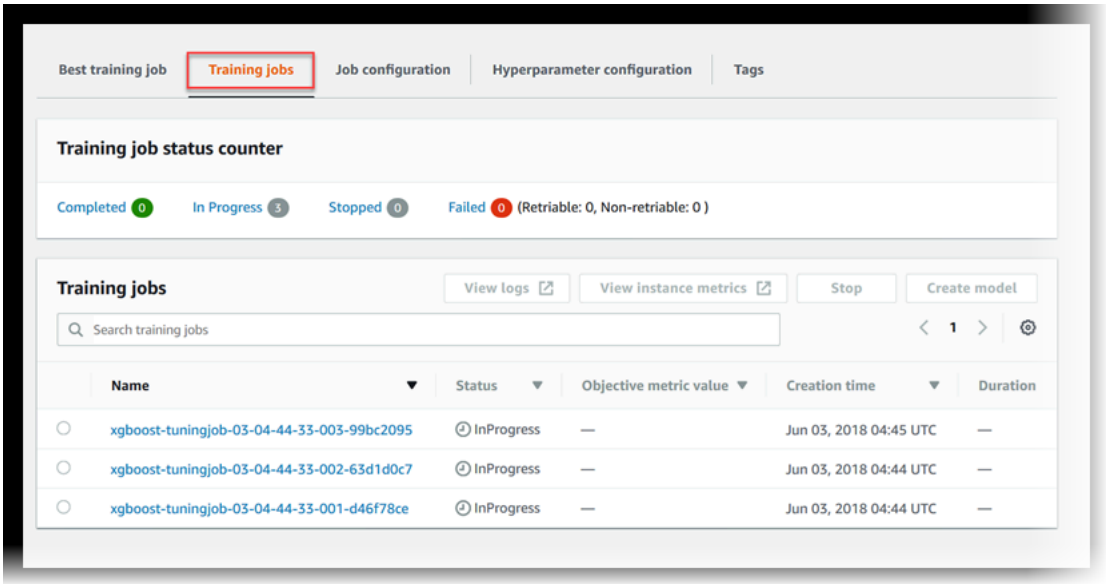
3. Dalam daftar pekerjaan tuning hyperparameter, periksa status pekerjaan tuning hyperparameter yang Anda luncurkan. Pekerjaan penyetelan dapat berupa:

   - `Completed`—Pekerjaan tuning hyperparameter berhasil diselesaikan.
   - `InProgress`—Pekerjaan tuning hyperparameter sedang berlangsung. Satu atau lebih pekerjaan pelatihan masih berjalan.
   - `Failed`—Pekerjaan tuning hyperparameter gagal.
   - `Stopped`—Pekerjaan tuning hyperparameter dihentikan secara manual sebelum selesai. Semua pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning hyperparameter dihentikan.
   - `Stopping`—Pekerjaan tuning hyperparameter sedang dalam proses berhenti.

Lihat Status Pekerjaan Pelatihan

Untuk melihat status pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning hyperparameter

1. Dalam daftar pekerjaan tuning hyperparameter, pilih pekerjaan yang Anda luncurkan.
2. Pilih pekerjaan Pelatihan.



3. Lihat status setiap pekerjaan pelatihan. Untuk melihat detail lebih lanjut tentang pekerjaan, pilih di daftar pekerjaan pelatihan. Untuk melihat ringkasan status semua pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning hyperparameter, lihat Penghitung status pekerjaan pelatihan.

   Pekerjaan pelatihan dapat berupa:

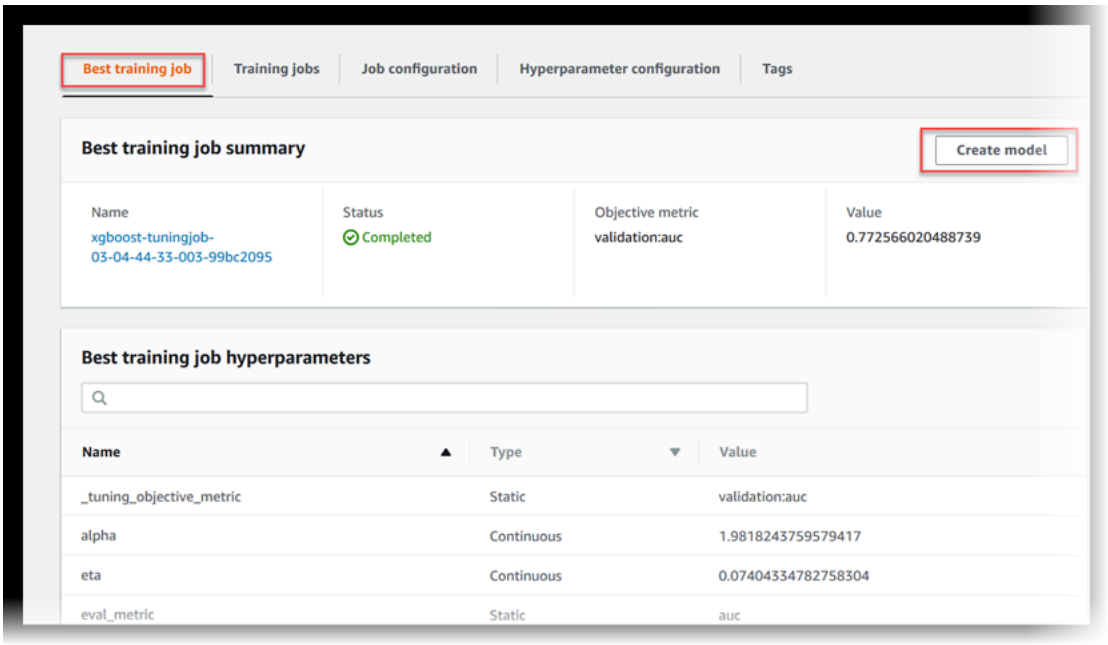
   - `Completed`—Pekerjaan pelatihan berhasil diselesaikan.
   - `InProgress`—Pekerjaan pelatihan sedang berlangsung.
   - `Stopped`—Pekerjaan pelatihan dihentikan secara manual sebelum selesai.
   - `Failed (Retryable)`—Pekerjaan pelatihan gagal, tetapi dapat dicoba lagi. Pekerjaan pelatihan yang gagal dapat dicoba kembali hanya jika gagal karena kesalahan layanan internal terjadi.
   - `Failed (Non-retryable)`—Pekerjaan pelatihan gagal dan tidak dapat dicoba lagi. Pekerjaan pelatihan yang gagal tidak dapat dicoba lagi ketika kesalahan klien terjadi.

> **Note**
>
> Pekerjaan tuning hyperparameter dapat dihentikan dan sumber daya yang mendasarinya dihapus, tetapi pekerjaan itu sendiri tidak dapat dihapus.

Lihat Training Job Terbaik

Pekerjaan tuning hyperparameter menggunakan metrik objektif yang dikembalikan oleh setiap pekerjaan pelatihan untuk mengevaluasi pekerjaan pelatihan. Sementara pekerjaan tuning hyperparameter sedang berlangsung, pekerjaan pelatihan terbaik adalah pekerjaan yang telah mengembalikan metrik objektif terbaik sejauh ini. Setelah pekerjaan tuning hyperparameter selesai, pekerjaan pelatihan terbaik adalah pekerjaan yang mengembalikan metrik objektif terbaik.

Untuk melihat pekerjaan pelatihan terbaik, pilih Pekerjaan pelatihan terbaik.



Untuk menerapkan pekerjaan pelatihan terbaik sebagai model yang dapat Anda host di SageMaker titik akhir, pilih Buat model.

Langkah Selanjutnya

[Hapus]

Hapus

Untuk menghindari biaya yang tidak perlu, ketika Anda selesai dengan contoh, gunakan AWS Management Console untuk menghapus sumber daya yang Anda buat untuk itu.

> **Note**
>
> Jika Anda berencana untuk menjelajahi contoh lain, Anda mungkin ingin menyimpan beberapa sumber daya ini, seperti instance notebook, bucket S3, dan peran IAM.

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/ dan hapus instance notebook. Hentikan instance sebelum menghapusnya.
2. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/ dan hapus bucket yang Anda buat untuk menyimpan artefak model dan kumpulan data pelatihan.
3. Buka konsol IAM di https://console.aws.amazon.com/iam/ dan hapus peran IAM. Jika Anda membuat kebijakan izin, Anda juga dapat menghapusnya.
4. Buka CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/ dan hapus semua grup log yang memiliki nama dimulai`/aws/sagemaker/`.

Hentikan Pekerjaan Pelatihan Lebih Awal

Hentikan pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning hyperparameter lebih awal ketika mereka tidak membaik secara signifikan yang diukur dengan metrik objektif. Menghentikan pekerjaan pelatihan lebih awal dapat membantu mengurangi waktu komputasi dan membantu Anda menghindari overfitting model Anda. Untuk mengonfigurasi pekerjaan tuning hyperparameter untuk menghentikan pekerjaan pelatihan lebih awal, lakukan salah satu hal berikut:

- Jika Anda menggunakan AWS SDK untuk Python (Boto3), `TrainingJobEarlyStoppingType` atur bidang objek yang Anda gunakan untuk mengonfigurasi pekerjaan `HyperParameterTuningJobConfig`penyetelan. `AUTO`
- Jika Anda menggunakan Amazon SageMaker Python SDK, atur `early_stopping_type` parameter objek ke. HyperParameterTuner`Auto`
- Di SageMaker konsol Amazon, dalam alur kerja kerja penyetelan hyperparameter Create hyperparameter, di bawah Pemberhentian awal, pilih Otomatis.

Untuk contoh buku catatan yang menunjukkan cara menggunakan penghentian awal, lihat https://github.com/awslabs/ amazon-sagemaker-examples /blob/master/hyperparameter_tuning/image_classification_early_stopping/hpo_image_classification_early_stopping.ipynb atau buka buku catatan di bagian Hyperparameter Tuning dari Contoh dalam instance notebook. **`hpo_image_classification_early_stopping.ipynb`** SageMaker Untuk informasi tentang menggunakan contoh buku catatan dalam contoh buku catatan, lihatContoh Notebook.

Bagaimana Berhenti Awal Bekerja

Saat Anda mengaktifkan penghentian dini untuk pekerjaan penyetelan hiperparameter, SageMaker evaluasi setiap pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning hyperparameter sebagai berikut:

- Setelah setiap zaman pelatihan, dapatkan nilai metrik objektif.
- Hitung rata-rata berjalan dari metrik objektif untuk semua pekerjaan pelatihan sebelumnya hingga zaman yang sama, dan kemudian hitung median dari semua rata-rata berjalan.
- Jika nilai metrik objektif untuk pekerjaan pelatihan saat ini lebih buruk (lebih tinggi saat meminimalkan atau lebih rendah saat memaksimalkan metrik objektif) daripada nilai rata-rata lari metrik objektif untuk pekerjaan pelatihan sebelumnya hingga zaman yang sama, SageMaker hentikan pekerjaan pelatihan saat ini.

Algoritma yang Mendukung Penghentian Awal

Untuk mendukung penghentian awal, algoritme harus memancarkan metrik objektif untuk setiap zaman. SageMaker Algoritme bawaan berikut mendukung penghentian awal:

- LightGBM
- CatBoost
- AutoGluon-Tabular
- TabTransformer
- Algoritma Linear—Didukung hanya jika Anda menggunakan `objective_loss` sebagai metrik tujuan.
- Algoritma XGBoost
- Klasifikasi Gambar - MXNet
- Deteksi
- Algoritma Urutan ke Urutan
- Wawasan IP

> **Note**
>
> Daftar algoritme bawaan yang mendukung penghentian awal ini adalah saat ini pada 13 Desember 2018. Algoritma bawaan lainnya mungkin mendukung penghentian awal di masa

depan. Jika suatu algoritma memancarkan metrik yang dapat digunakan sebagai metrik objektif untuk pekerjaan penyetelan hiperparameter (lebih disukai metrik validasi), maka itu mendukung penghentian awal.

Untuk menggunakan penghentian awal dengan algoritme Anda sendiri, Anda harus menulis algoritme Anda sedemikian rupa sehingga memancarkan nilai metrik objektif setelah setiap zaman. Daftar berikut menunjukkan bagaimana Anda dapat melakukannya dalam kerangka kerja yang berbeda:

TensorFlow

Gunakan `tf.keras.callbacks.ProgbarLogger` kelas. Untuk selengkapnya, lihat tf.keras.callbacks. ProgbarLogger API.

MXNet

Gunakan `mxnet.callback.LogValidationMetricsCallback`. Untuk selengkapnya, lihat API mxnet.callback.

Chainer

Perluas chainer dengan menggunakan `extensions.Evaluator` kelas. Untuk selengkapnya, lihat Chainer.Training.Extensions.Evaluator API.

PyTorch dan Spark

Tidak ada dukungan tingkat tinggi. Anda harus secara eksplisit menulis kode pelatihan Anda sehingga menghitung metrik objektif dan menulisnya ke log setelah setiap epoch.

Jalankan Pekerjaan Tuning Hyperparameter Mulai yang Hangat

Gunakan start hangat untuk memulai pekerjaan tuning hyperparameter menggunakan satu atau lebih pekerjaan tuning sebelumnya sebagai titik awal. Hasil pekerjaan penyetelan sebelumnya digunakan untuk menginformasikan kombinasi hiperparameter mana yang harus dicari dalam pekerjaan penyetelan baru. Penyetelan hyperparameter menggunakan pencarian Bayesian atau acak untuk memilih kombinasi nilai hyperparameter dari rentang yang Anda tentukan. Untuk informasi selengkapnya, lihat Bagaimana Hyperparameter Tuning Bekerja. Menggunakan informasi dari pekerjaan tuning hyperparameter sebelumnya dapat membantu meningkatkan kinerja pekerjaan tuning hyperparameter baru dengan membuat pencarian kombinasi hyperparameter terbaik lebih efisien.

> **Note**
>
> Pekerjaan penyetelan awal yang hangat biasanya membutuhkan waktu lebih lama untuk memulai daripada pekerjaan tuning hiperparameter standar, karena hasil dari pekerjaan induk harus dimuat sebelum pekerjaan dapat dimulai. Peningkatan waktu tergantung pada jumlah total pekerjaan pelatihan yang diluncurkan oleh pekerjaan orang tua.

Alasan untuk mempertimbangkan awal yang hangat meliputi yang berikut:

- Untuk secara bertahap meningkatkan jumlah pekerjaan pelatihan selama beberapa pekerjaan penyetelan berdasarkan hasil setelah setiap iterasi.
- Untuk menyetel model menggunakan data baru yang Anda terima.
- Untuk mengubah rentang hyperparameter yang Anda gunakan dalam pekerjaan penyetelan sebelumnya, ubah hyperparameters statis menjadi tunable, atau ubah hyperparameters yang dapat disetel ke nilai statis.
- Anda menghentikan pekerjaan hyperparameter sebelumnya lebih awal atau berhenti secara tak terduga.

Topik

Jenis Pekerjaan Tuning Mulai Hangat

Ada dua jenis pekerjaan penyetelan awal yang hangat:

`IDENTICAL_DATA_AND_ALGORITHM`

Pekerjaan penyetelan hiperparameter baru menggunakan data input dan gambar pelatihan yang sama dengan pekerjaan penyetelan induk. Anda dapat mengubah rentang hyperparameter untuk mencari dan jumlah maksimum pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning hyperparameter. Anda juga dapat mengubah hyperparameters dari tunable ke static, dan dari

static ke tunable, tetapi jumlah total static plus tunable hyperparameters harus tetap sama seperti di semua pekerjaan induk. Anda tidak dapat menggunakan versi baru dari algoritma pelatihan, kecuali perubahan dalam versi baru tidak memengaruhi algoritme itu sendiri. Misalnya, perubahan yang meningkatkan pencatatan atau menambahkan dukungan untuk format data yang berbeda diperbolehkan.

Gunakan data dan algoritme yang identik saat Anda menggunakan data pelatihan yang sama seperti yang Anda gunakan dalam pekerjaan penyetelan hiperparameter sebelumnya, tetapi Anda ingin meningkatkan jumlah total pekerjaan pelatihan atau mengubah rentang atau nilai hiperparameter.

Saat Anda menjalankan jenis pekerjaan penyetelan awal yang hangat`IDENTICAL_DATA_AND_ALGORITHM`, ada bidang tambahan dalam respons terhadap [`DescribeHyperParameterTuningJob`](#)nama`OverallBestTrainingJob`. Nilai bidang ini adalah [TrainingJobSummary](#)untuk pekerjaan pelatihan dengan nilai metrik objektif terbaik dari semua pekerjaan pelatihan yang diluncurkan oleh pekerjaan penyetelan ini dan semua pekerjaan induk yang ditentukan untuk pekerjaan penyetelan awal yang hangat.

`TRANSFER_LEARNING`

Pekerjaan penyetelan hiperparameter baru dapat mencakup data input, rentang hiperparameter, jumlah maksimum pekerjaan pelatihan bersamaan, dan jumlah maksimum pekerjaan pelatihan yang berbeda dari pekerjaan penyetelan hiperparameter induknya. Anda juga dapat mengubah hyperparameters dari tunable ke static, dan dari static ke tunable, tetapi jumlah total static plus tunable hyperparameters harus tetap sama seperti di semua pekerjaan induk. Gambar algoritma pelatihan juga bisa menjadi versi yang berbeda dari versi yang digunakan dalam pekerjaan tuning hyperparameter induk. Saat Anda menggunakan pembelajaran transfer, perubahan dalam kumpulan data atau algoritme yang secara signifikan memengaruhi nilai metrik objektif dapat mengurangi kegunaan penggunaan penyetelan awal yang hangat.

Pembatasan Penyetelan Mulai Hangat

Pembatasan berikut berlaku untuk semua pekerjaan penyetelan awal yang hangat:

- Pekerjaan tuning dapat memiliki maksimal 5 pekerjaan orang tua, dan semua pekerjaan orang tua harus dalam status terminal (`Completed`,`Stopped`, atau`Failed`) sebelum Anda memulai pekerjaan penyetelan baru.
- Metrik objektif yang digunakan dalam pekerjaan penyetelan baru harus sama dengan metrik objektif yang digunakan dalam pekerjaan induk.

- Jumlah total hiperparameter statis plus yang dapat disetel harus tetap sama antara pekerjaan induk dan pekerjaan penyetelan baru. Karena itu, jika Anda berpikir Anda mungkin ingin menggunakan hyperparameter sebagai tunable dalam pekerjaan tuning start hangat di masa depan, Anda harus menambahkannya sebagai hyperparameter statis saat Anda membuat pekerjaan penyetelan.
- Tipe setiap hiperparameter (kontinu, bilangan bulat, kategoris) tidak boleh berubah antara pekerjaan induk dan pekerjaan penyetelan baru.
- Jumlah total perubahan dari hiperparameter yang dapat disetel dalam pekerjaan induk ke hiperparameter statis dalam pekerjaan penyetelan baru, ditambah jumlah perubahan nilai hiperparameter statis tidak boleh lebih dari 10. Misalnya, jika pekerjaan induk memiliki hiperparameter kategoris yang dapat disetel dengan nilai yang mungkin `red` dan`blue`, Anda mengubah hiperparameter itu menjadi statis dalam pekerjaan penyetelan baru, yang dihitung sebagai 2 perubahan terhadap total 10 yang diizinkan. Jika hyperparameter yang sama memiliki nilai statis `red` dalam pekerjaan induk, dan Anda mengubah nilai statis ke `blue` dalam pekerjaan penyetelan baru, itu juga dihitung sebagai 2 perubahan.
- Penyetelan awal yang hangat tidak rekursif. Misalnya, jika Anda membuat `MyTuningJob3` pekerjaan penyetelan awal yang hangat dengan `MyTuningJob2` sebagai pekerjaan orang tua, dan itu sendiri `MyTuningJob2` merupakan pekerjaan penyetelan awal yang hangat dengan pekerjaan orang tua`MyTuningJob1`, informasi yang dipelajari saat menjalankan tidak `MyTuningJob1` digunakan. `MyTuningJob3` Jika Anda ingin menggunakan informasi dari`MyTuningJob1`, Anda harus secara eksplisit menambahkannya sebagai induk untuk. `MyTuningJob3`
- Pekerjaan pelatihan yang diluncurkan oleh setiap pekerjaan orang tua dalam pekerjaan penyetelan awal yang hangat dihitung terhadap 500 pekerjaan pelatihan maksimum untuk pekerjaan tuning.
- Pekerjaan tuning hyperparameter yang dibuat sebelum 1 Oktober 2018 tidak dapat digunakan sebagai pekerjaan induk untuk pekerjaan tuning awal yang hangat.

Notebook Contoh Penyetelan Mulai Hangat

Untuk contoh buku catatan yang menunjukkan cara menggunakan penyetelan mulai hangat, lihat https://github.com/awslabs/ amazon-sagemaker-examples /blob/master/hyperparameter_tuning/image_classification_warmstart/hpo_image_classification_warmstart.ipynb. Untuk petunjuk cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Contoh Notebook Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMaker Contoh untuk melihat daftar semua SageMaker sampel. Notebook contoh penyetelan awal hangat terletak di bagian tuning Hyperparameter, dan diberi nama.

`hpo_image_classification_warmstart.ipynb` Untuk membuka buku catatan, klik tab Use dan pilih Create copy.

Buat Job Tuning Mulai yang Hangat

Anda dapat menggunakan AWS SDK tingkat rendah untuk Python (Boto 3) atau SDK Python tingkat tinggi untuk membuat pekerjaan penyetelan SageMaker awal yang hangat.

Topik

- Buat Job Penyetelan Mulai yang Hangat (SageMaker API tingkat rendah untuk Python (Boto 3))
- Buat Job Penyetelan Mulai yang Hangat (SageMakerPython SDK)

Buat Job Penyetelan Mulai yang Hangat (SageMaker API tingkat rendah untuk Python (Boto 3))

Untuk menggunakan penyetelan mulai hangat, Anda menentukan nilai `HyperParameterTuningJobWarmStartConfig`objek, dan meneruskannya sebagai `WarmStartConfig` bidang dalam panggilan ke `CreateHyperParameterTuningJob`.

Kode berikut menunjukkan cara membuat `HyperParameterTuningJobWarmStartConfig`objek dan meneruskannya ke `CreateHyperParameterTuningJob`pekerjaan dengan menggunakan SageMaker API tingkat rendah untuk Python (Boto 3).

Buat `HyperParameterTuningJobWarmStartConfig` objek:

```
warm_start_config = {
          "ParentHyperParameterTuningJobs" : [
          {"HyperParameterTuningJobName" : 'MyParentTuningJob'}
          ],
          "WarmStartType" : "IdenticalDataAndAlgorithm"
}
```

Buat pekerjaan penyetelan awal yang hangat:

```
smclient = boto3.Session().client('sagemaker')
smclient.create_hyper_parameter_tuning_job(HyperParameterTuningJobName =
 'MyWarmStartTuningJob',
   HyperParameterTuningJobConfig = tuning_job_config, # See notebook for tuning
 configuration
   TrainingJobDefinition = training_job_definition, # See notebook for job definition
   WarmStartConfig = warm_start_config)
```

Buat Job Penyetelan Mulai yang Hangat (SageMakerPython SDK)

Untuk menggunakan Amazon SageMaker Python SDK untuk menjalankan pekerjaan penyetelan awal yang hangat, Anda:

- Tentukan pekerjaan induk dan tipe awal hangat dengan menggunakan `WarmStartConfig` objek.
- Lewati `WarmStartConfig` objek sebagai nilai `warm_start_config` argumen HyperparameterTunerobjek.
- Panggil `fit` metode `HyperparameterTuner` objek.

Untuk informasi selengkapnya tentang penggunaan Amazon SageMaker Python SDK untuk penyetelan hyperparameter, lihat https://github.com/aws/ #. sagemaker-python-sdk sagemaker-automatic-model-tuning

Contoh ini menggunakan estimator yang menggunakan Klasifikasi Gambar - MXNet algoritma untuk pelatihan. Kode berikut menetapkan rentang hyperparameter yang dicari pekerjaan warm start tuning untuk menemukan kombinasi nilai terbaik. Untuk informasi tentang pengaturan rentang hyperparameter, lihatTentukan Rentang Hyperparameter.

```
hyperparameter_ranges = {'learning_rate': ContinuousParameter(0.0, 0.1),
                         'momentum': ContinuousParameter(0.0, 0.99)}
```

Kode berikut mengonfigurasi pekerjaan penyetelan awal yang hangat dengan membuat `WarmStartConfig` objek.

```
from sagemaker.tuner import WarmStartConfig,WarmStartTypes

parent_tuning_job_name = "MyParentTuningJob"
warm_start_config =
 WarmStartConfig(warm_start_type=WarmStartTypes.IDENTICAL_DATA_AND_ALGORITHM,
 parents={parent_tuning_job_name})
```

Sekarang atur nilai untuk hyperparameters statis, yang merupakan hyperparameters yang menjaga nilai yang sama untuk setiap pekerjaan pelatihan yang diluncurkan oleh pekerjaan tuning start hangat. Dalam kode berikut, `imageclassification` adalah estimator yang dibuat sebelumnya.

```
imageclassification.set_hyperparameters(num_layers=18,
                                        image_shape='3,224,224',
                                        num_classes=257,
```

```
                                    num_training_samples=15420,
                                    mini_batch_size=128,
                                    epochs=30,
                                    optimizer='sgd',
                                    top_k='2',
                                    precision_dtype='float32',
                                    augmentation_type='crop')
```

Sekarang buat `HyperparameterTuner` objek dan lewati `WarmStartConfig` objek yang sebelumnya Anda buat sebagai `warm_start_config` argumen.

```
tuner_warm_start = HyperparameterTuner(imageclassification,
                            'validation:accuracy',
                            hyperparameter_ranges,
                            objective_type='Maximize',
                            max_jobs=10,
                            max_parallel_jobs=2,
                            base_tuning_job_name='warmstart',
                            warm_start_config=warm_start_config)
```

Akhirnya, panggil `fit` metode `HyperparameterTuner` objek untuk meluncurkan pekerjaan penyetelan awal yang hangat.

```
tuner_warm_start.fit(
        {'train': s3_input_train, 'validation': s3_input_validation},
        include_cls_metadata=False)
```

Batas Sumber Daya untuk Penyetelan Model Otomatis

SageMaker menetapkan batas default berikut untuk sumber daya yang digunakan oleh penyetelan model otomatis:

Sumber daya	Wilayah	Batas default	Dapat ditingkatkan hingga
Jumlah pekerjaan penyetelan hiperparameter paralel (bersamaan)	Semua	100	N/A

Sumber daya	Wilayah	Batas default	Dapat ditingkatkan hingga
Jumlah hiperpara meter yang dapat dicari *	Semua	30	N/A
Jumlah metrik yang ditentukan per pekerjaan penyetelan hyperparameter	Semua	20	N/A
Jumlah tugas pelatihan paralel per tugas penyetelan hyperparameter	Semua	10	100
[Optimasi Bayesian] Jumlah pekerjaan pelatihan per pekerjaan tuning hyperparameter	Semua	750	N/A
[Pencarian acak] Jumlah pekerjaan pelatihan per pekerjaan tuning hyperparameter	Semua	750	10000
[Hyperband] Jumlah pekerjaan pelatihan per pekerjaan tuning hyperparameter	Semua	750	N/A

Sumber daya	Wilayah	Batas default	Dapat ditingkatkan hingga
[Grid] Jumlah pekerjaan pelatihan per pekerjaan tuning hyperparameter, baik ditentukan secara eksplisit atau disimpulkan dari ruang pencarian	Semua	750	N/A
Waktu berjalan maksimum untuk pekerjaan tuning hyperparameter	Semua	30 hari	N/A

* Setiap hyperparameter kategoris dapat memiliki paling banyak 30 nilai yang berbeda.

Contoh batas sumber daya

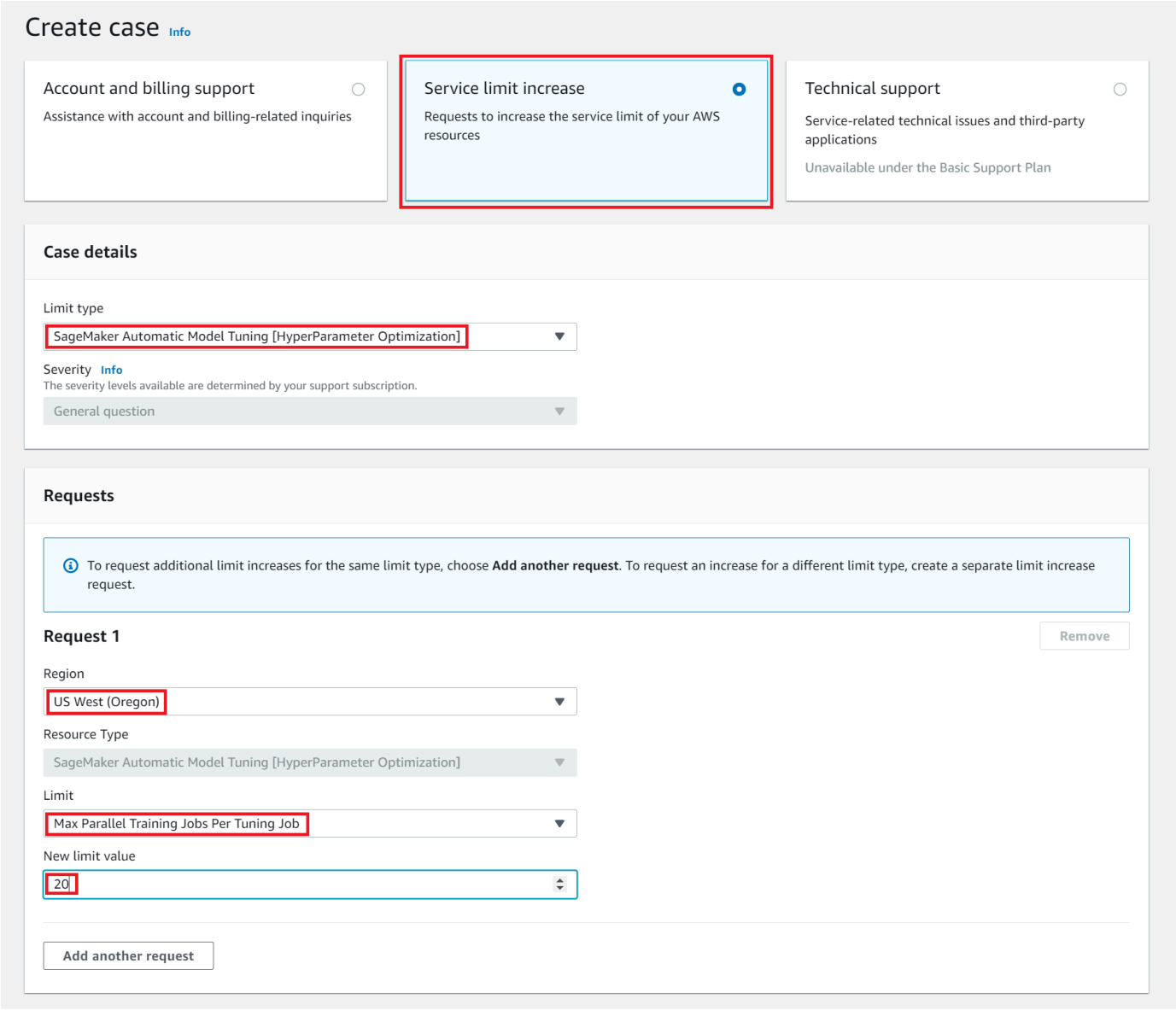
Saat Anda merencanakan pekerjaan penyetelan hyperparameter, Anda juga harus memperhitungkan batasan sumber daya pelatihan. Untuk informasi tentang batas sumber daya default untuk pekerjaan SageMaker pelatihan, lihat SageMakerBatas. Setiap instance pelatihan bersamaan di mana semua pekerjaan penyetelan hyperparameter Anda berjalan diperhitungkan terhadap jumlah total instans pelatihan yang diizinkan. Misalnya, jika Anda menjalankan 10 pekerjaan penyetelan hiperparameter bersamaan, masing-masing pekerjaan penyetelan hiperparameter tersebut menjalankan 100 total pekerjaan pelatihan dan 20 pekerjaan pelatihan bersamaan. Masing-masing pekerjaan pelatihan tersebut berjalan pada satu instance ml.m4.xlarge. Batasan berikut berlaku:

- Jumlah pekerjaan tuning hyperparameter bersamaan: Anda tidak perlu menambah batas, karena 10 pekerjaan tuning di bawah batas 100.
- Jumlah pekerjaan pelatihan per pekerjaan tuning hyperparameter: Anda tidak perlu menambah batas, karena 100 pekerjaan pelatihan di bawah batas 750.
- Jumlah pekerjaan pelatihan bersamaan per pekerjaan tuning hyperparameter: Anda perlu meminta peningkatan batas menjadi 20, karena batas default adalah 10.

- SageMaker pelatihan ml.m4.xlarge instance: Anda perlu meminta peningkatan batas menjadi 200, karena Anda memiliki 10 pekerjaan tuning hyperparameter, yang masing-masing menjalankan 20 pekerjaan pelatihan bersamaan. Batas defaultnya adalah 20 instance.
- SageMaker jumlah instans total pelatihan: Anda perlu meminta peningkatan batas menjadi 200, karena Anda memiliki 10 pekerjaan penyetelan hiperparameter, yang masing-masing menjalankan 20 pekerjaan pelatihan bersamaan. Batas defaultnya adalah 20 instance.

Untuk meminta peningkatan kuota:

1. Buka laman AWSPusat Dukungan, masuk jika perlu, dan pilih Buat kasus.
2. Di halaman Create case (Buat kasus), pilihService limit increase (Peningkatan batas layanan).
3. Pada panel Case details, pilih SageMaker Automatic Model Tuning [Hyperparameter Optimization] untuk tipe Limit
4. Pada panel Permintaan untuk Permintaan 1, pilih Wilayah, Batas sumber daya yang akan ditingkatkan dan nilai Batas Baru yang Anda minta. Pilih Tambahkan permintaan lain jika Anda memiliki permintaan tambahan untuk peningkatan kuota.



5. Di panel Deskripsi kasus, berikan deskripsi kasus penggunaan Anda.
6. Di panel opsi Kontak, pilih metode Kontak pilihan Anda (Web, Chat atau Telepon) dan kemudian pilih Kirim.

Praktik Terbaik untuk Tuning Hyperparameter

Optimasi Hyperparameter (HPO) bukanlah proses yang sepenuhnya otomatis. Untuk meningkatkan optimasi, ikuti praktik terbaik ini untuk penyetelan hyperparameter.

Topik

- Memilih strategi penyetelan

- Memilih jumlah hyperparameters
- Memilih rentang hyperparameter
- Menggunakan skala yang benar untuk hyperparameters
- Memilih jumlah pekerjaan pelatihan paralel terbaik
- Menjalankan pekerjaan pelatihan pada beberapa contoh
- Menggunakan benih acak untuk mereproduksi konfigurasi hyperparameter

Memilih strategi penyetelan

Untuk pekerjaan besar, menggunakan strategi tuning Hyperband dapat mengurangi waktu komputasi. Hyperband memiliki mekanisme penghentian dini untuk menghentikan pekerjaan yang berkinerja buruk. Hyperband juga dapat mengalokasikan kembali sumber daya ke konfigurasi hyperparameter yang dimanfaatkan dengan baik dan menjalankan pekerjaan paralel. Untuk pekerjaan pelatihan yang lebih kecil menggunakan lebih sedikit runtime, gunakan pencarian acak atau optimasi Bayesian.

Gunakan optimasi Bayesian untuk membuat keputusan yang semakin terinformasi tentang meningkatkan konfigurasi hyperparameter dalam proses berikutnya. Optimasi Bayesian menggunakan informasi yang dikumpulkan dari proses sebelumnya untuk meningkatkan proses selanjutnya. Karena sifatnya yang berurutan, optimasi Bayesian tidak dapat skala besar-besaran.

Gunakan pencarian acak untuk menjalankan sejumlah besar pekerjaan paralel. Dalam pencarian acak, pekerjaan selanjutnya tidak bergantung pada hasil dari pekerjaan sebelumnya dan dapat dijalankan secara independen. Dibandingkan dengan strategi lain, pencarian acak mampu menjalankan jumlah pekerjaan paralel terbesar.

Gunakan pencarian kisi untuk mereproduksi hasil pekerjaan penyetelan, atau jika kesederhanaan dan transparansi algoritma pengoptimalan penting. Anda juga dapat menggunakan pencarian kisi untuk menjelajahi seluruh ruang pencarian hyperparameter secara merata. Pencarian kisi secara metodis mencari melalui setiap kombinasi hyperparameter untuk menemukan nilai hyperparameter yang optimal. Tidak seperti pencarian grid, optimasi Bayesian, pencarian acak, dan Hyperband semuanya menggambar hiperparameter secara acak dari ruang pencarian. Karena pencarian grid menganalisis setiap kombinasi hyperparameters, nilai hyperparameter optimal akan identik antara pekerjaan tuning yang menggunakan hyperparameter yang sama.

Memilih jumlah hyperparameters

Selama pengoptimalan, kompleksitas komputasi dari pekerjaan penyetelan hiperparameter bergantung pada hal berikut:

- Jumlah hiperparameter
- Kisaran nilai yang SageMaker harus dicari Amazon

Meskipun Anda dapat secara bersamaan menentukan hingga 30 hyperparameters, membatasi pencarian Anda ke angka yang lebih kecil dapat mengurangi waktu komputasi. Mengurangi waktu komputasi memungkinkan SageMaker untuk konvergen lebih cepat ke konfigurasi hyperparameter yang optimal.

Memilih rentang hyperparameter

Rentang nilai yang Anda pilih untuk dicari dapat mempengaruhi optimasi hyperparameter. Misalnya, rentang yang mencakup setiap nilai hyperparameter yang mungkin dapat menyebabkan waktu komputasi yang besar dan model yang tidak menggeneralisasi dengan baik ke data yang tidak terlihat. Jika Anda tahu bahwa menggunakan subset dari rentang terbesar yang mungkin sesuai untuk kasus penggunaan Anda, pertimbangkan untuk membatasi rentang ke subset tersebut.

Menggunakan skala yang benar untuk hyperparameters

Selama penyetelan hyperparameter, SageMaker coba simpulkan apakah hyperparameter Anda berskala log atau berskala linier. Awalnya, SageMaker mengasumsikan penskalaan linier untuk hyperparameters. Jika hyperparameters diskalakan log, memilih skala yang benar akan membuat pencarian Anda lebih efisien. Anda juga dapat memilih `Auto` untuk `ScalingType` di CreateHyperParameterTuningJobAPI jika Anda SageMaker ingin mendeteksi skala untuk Anda.

Memilih jumlah pekerjaan pelatihan paralel terbaik

Anda dapat menggunakan hasil uji coba sebelumnya untuk meningkatkan kinerja uji coba berikutnya. Pilih jumlah pekerjaan paralel terbesar yang akan memberikan hasil tambahan yang berarti yang juga ada di wilayah Anda dan batasan komputasi akun. Gunakan `MaxParallelTrainingJobs`bidang untuk membatasi jumlah pekerjaan pelatihan yang dapat diluncurkan oleh pekerjaan tuning hyperparameter secara paralel. Untuk informasi selengkapnya, lihat Menjalankan beberapa pekerjaan HPO secara paralel di Amazon SageMaker.

Menjalankan pekerjaan pelatihan pada beberapa contoh

Ketika pekerjaan pelatihan berjalan pada beberapa mesin dalam mode terdistribusi, setiap mesin memancarkan metrik objektif. HPO hanya dapat menggunakan salah satu metrik objektif yang dipancarkan ini untuk mengevaluasi kinerja model, Dalam mode terdistribusi, HPO menggunakan metrik objektif yang dilaporkan oleh pekerjaan berjalan terakhir di semua instance.

Menggunakan benih acak untuk mereproduksi konfigurasi hyperparameter

Anda dapat menentukan bilangan bulat sebagai benih acak untuk penyetelan hyperparameter dan menggunakan seed tersebut selama pembuatan hyperparameter. Nanti, Anda dapat menggunakan seed yang sama untuk mereproduksi konfigurasi hyperparameter yang konsisten dengan hasil sebelumnya. Untuk pencarian acak dan strategi Hyperband, menggunakan seed acak yang sama dapat memberikan reproduktifitas hingga 100% dari konfigurasi hyperparameter sebelumnya untuk pekerjaan penyetelan yang sama. Untuk strategi Bayesian, menggunakan seed acak yang sama akan meningkatkan reproduktifitas untuk pekerjaan penyetelan yang sama.

Perbaiki data selama pelatihan dengan penyaringan SageMaker cerdas Amazon

> Penyaringan SageMaker pintar Amazon sedang dalam rilis pratinjau dan dapat berubah sewaktu-waktu.

SageMaker smart sifting adalah kemampuan SageMaker Pelatihan yang membantu meningkatkan efisiensi kumpulan data pelatihan Anda dan mengurangi total waktu dan biaya pelatihan.

Model pembelajaran mendalam modern seperti model bahasa besar (LLM) atau model transformator visi sering kali membutuhkan kumpulan data besar untuk mencapai akurasi yang dapat diterima. Misalnya, LLM sering membutuhkan triliunan token atau petabyte data untuk bertemu. Meningkatnya ukuran kumpulan data pelatihan, bersama dengan ukuran state-of-the-art model, dapat meningkatkan waktu komputasi dan biaya pelatihan model.

Selalu, sampel dalam kumpulan data tidak berkontribusi sama terhadap proses pembelajaran selama pelatihan model. Sebagian besar sumber daya komputasi yang disediakan selama pelatihan mungkin dihabiskan untuk memproses sampel mudah yang tidak berkontribusi secara substansif terhadap akurasi keseluruhan model. Idealnya, kumpulan data pelatihan hanya akan menyertakan sampel yang benar-benar meningkatkan konvergensi model. Memfilter data yang kurang bermanfaat dapat

mengurangi waktu pelatihan dan biaya komputasi. Namun, mengidentifikasi data yang kurang bermanfaat dapat menjadi tantangan dan berisiko. Praktis sulit untuk mengidentifikasi sampel mana yang kurang informatif sebelum pelatihan, dan akurasi model dapat terpengaruh jika sampel yang salah atau terlalu banyak sampel dikeluarkan.

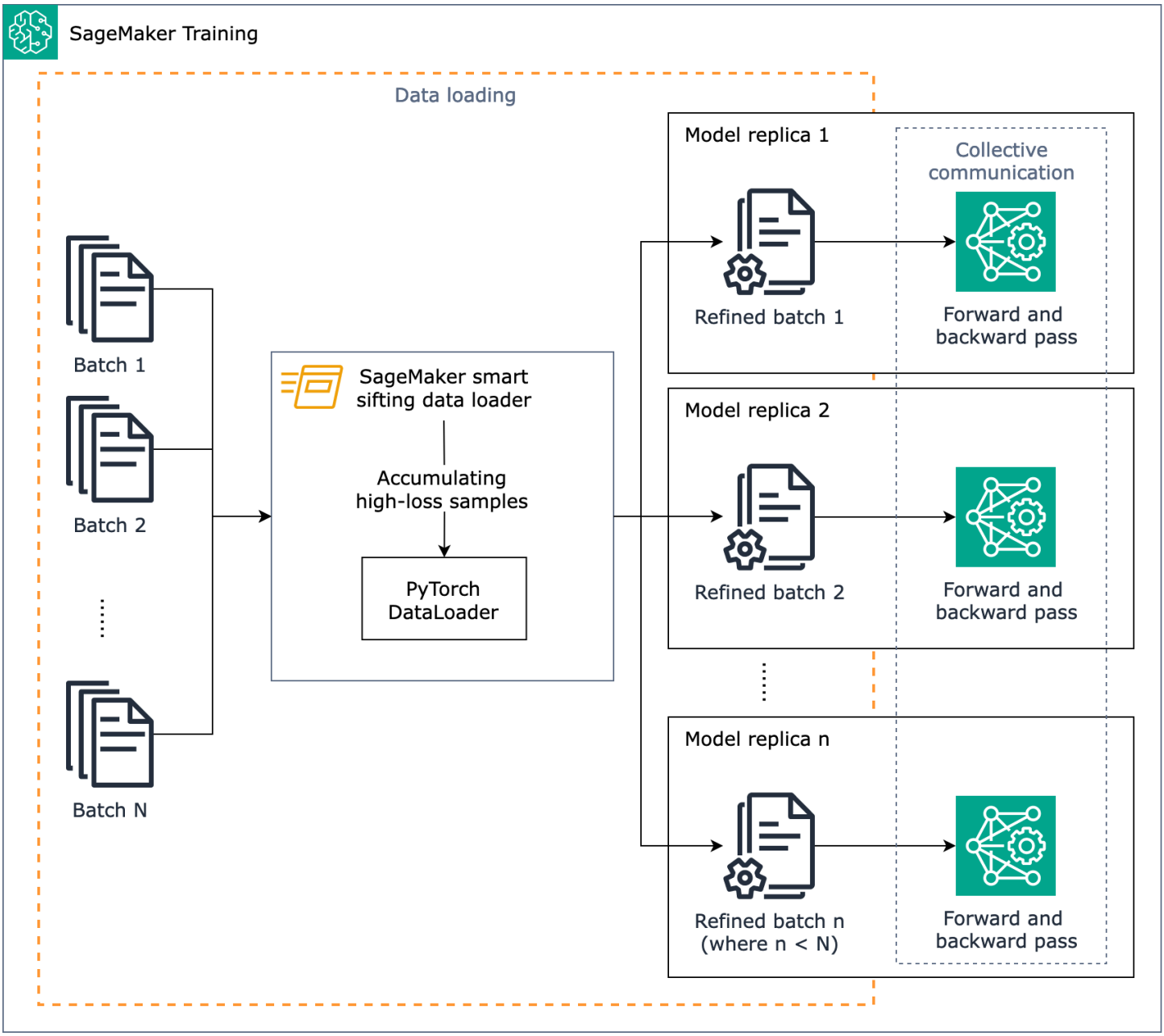
Pemilahan data secara cerdas dengan Amazon SageMaker dapat membantu mengurangi waktu dan biaya pelatihan dengan meningkatkan efisiensi data. Algoritma penyaringan SageMaker cerdas mengevaluasi nilai kehilangan setiap data selama tahap pemuatan data dari pekerjaan pelatihan dan mengecualikan sampel yang kurang informatif untuk model. Dengan menggunakan data yang disempurnakan untuk pelatihan, total waktu dan biaya pelatihan model Anda dikurangi dengan menghilangkan umpan maju dan mundur yang tidak perlu pada data yang tidak ditingkatkan. Oleh karena itu, ada dampak minimal atau tidak ada pada keakuratan model.

SageMaker smart sifting tersedia melalui SageMaker Training Deep Learning Containers (DLC) dan mendukung beban PyTorch kerja melalui. PyTorch `DataLoader` Hanya beberapa baris perubahan kode yang diperlukan untuk menerapkan penyaringan SageMaker cerdas dan Anda tidak perlu mengubah pelatihan atau alur kerja pemrosesan data yang ada.

Cara kerja penyaringan SageMaker cerdas

Tujuan dari penyaringan SageMaker cerdas adalah untuk menyaring data pelatihan Anda selama proses pelatihan dan hanya memberi makan sampel yang lebih informatif ke model. Selama pelatihan tipikal dengan PyTorch, data dikirim secara iteratif dalam batch ke loop pelatihan dan ke perangkat akselerator (seperti GPU atau chip Trainium) oleh. `PyTorchDataLoader` SageMaker penyaringan cerdas diimplementasikan pada tahap pemuatan data ini dan dengan demikian tidak tergantung pada pra-pemrosesan data hulu dalam jalur pelatihan Anda. SageMaker smart sifting menggunakan model Anda dan fungsi kerugian yang ditentukan pengguna untuk melakukan forward pass evaluatif dari setiap sampel data saat dimuat. Sampel yang mengembalikan nilai kerugian rendah memiliki dampak yang lebih kecil pada pembelajaran model dan dengan demikian dikeluarkan dari pelatihan, karena sudah mudah bagi model untuk membuat prediksi yang tepat tentang mereka dengan keyakinan tinggi. Sementara itu, sampel dengan kerugian yang relatif tinggi itulah yang masih perlu dipelajari oleh model, jadi ini disimpan untuk pelatihan. Input kunci yang dapat Anda atur untuk penyaringan SageMaker cerdas adalah proporsi data yang akan dikecualikan. Misalnya, dengan menetapkan proporsi ke 25%, sampel yang didistribusikan dalam kuartil terendah dari distribusi kerugian (diambil dari jumlah sampel sebelumnya yang ditentukan pengguna) dikeluarkan dari pelatihan. Sampel dengan kerugian tinggi diakumulasikan dalam kumpulan data yang disempurnakan. Kumpulan data yang disempurnakan dikirim ke loop pelatihan (pass maju dan mundur), dan model belajar dan melatih pada batch data yang disempurnakan.

Diagram berikut menunjukkan gambaran umum tentang bagaimana algoritma penyaringan SageMaker cerdas dirancang.



Singkatnya, penyaringan SageMaker cerdas beroperasi selama pelatihan saat data dimuat. Algoritma penyaringan SageMaker cerdas menjalankan perhitungan kerugian pada batch, dan menyaring data yang tidak ditingkatkan sebelum pass maju dan mundur dari setiap iterasi. Batch data yang disempurnakan kemudian digunakan untuk pass maju dan mundur.

SageMaker smart sifting berfungsi untuk pekerjaan pelatihan PyTorch berbasis dengan paralelisme data terdistribusi klasik, yang membuat replika model pada setiap pekerja dan kinerja GPU. `AllReduce` Ia bekerja dengan PyTorch DDP dan perpustakaan paralel data SageMaker terdistribusi.

Kerangka kerja dan AWS Wilayah yang didukung

Sebelum menggunakan pemuat data penyaringan SageMaker cerdas, periksa apakah kerangka kerja pilihan Anda didukung, apakah jenis instans tersedia di AWS akun Anda, dan apakah AWS akun Anda berada di salah satu AWS Wilayah yang didukung.

Kerangka Kerja yang Didukung

SageMaker smart sifting mendukung kerangka pembelajaran mendalam berikut dan tersedia melalui Deep Learning AWS Containers.

Topik

- PyTorch

PyTorch

Kerangka Kerja	Versi kerangka kerja	URI Wadah Pembelajaran Mendalam	
PyTorch	2.1.0	*763104351884* .dkr.ecr. *wilayah* .amazonaws.com/pytorch-training:2.1.0-gpu-py310-cu121-ubuntu20.04-sagemaker	

Untuk informasi selengkapnya tentang container yang sudah dibuat sebelumnya, lihat SageMaker Framework Container di GitHub repositori AWS Deep Learning Containers.

Wilayah AWS

Wadah yang dikemas dengan perpustakaan penyaringan SageMaker pintar tersedia di Wilayah AWS tempat AWSDeep Learning Containers berada dalam layanan.

Tipe instans

Anda dapat menggunakan penyaringan SageMaker cerdas untuk pekerjaan PyTorch pelatihan apa pun pada jenis instans apa pun. Sebaiknya gunakan instans P4d, P4de, atau P5.

Terapkan penyaringan SageMaker cerdas ke skrip pelatihan Anda

Pustaka penyaringan SageMaker cerdas dikemas dalam SageMaker kerangka DLC sebagai pustaka pelengkap. Ini memberikan logika penyaringan terhadap sampel pelatihan yang memiliki dampak yang relatif lebih rendah pada pelatihan model, dan model Anda dapat mencapai akurasi model yang diinginkan dengan sampel pelatihan yang lebih sedikit jika dibandingkan dengan pelatihan model dengan sampel data lengkap.

PyTorch

Instruksi ini menunjukkan cara mengaktifkan penyaringan SageMaker cerdas dengan skrip pelatihan Anda.

1. Konfigurasikan antarmuka penyaringan SageMaker cerdas.

   Pustaka penyaringan SageMaker cerdas mengimplementasikan teknik pengambilan sampel berbasis kerugian ambang batas relatif yang membantu menyaring sampel dengan dampak yang lebih rendah dalam mengurangi nilai kerugian. Algoritma penyaringan SageMaker cerdas menghitung nilai kerugian dari setiap sampel data input menggunakan pass maju, dan menghitung persentil relatifnya terhadap nilai kehilangan data sebelumnya.

   Dua parameter berikut adalah apa yang perlu Anda tentukan ke `RelativeProbabilisticSiftConfig` kelas untuk membuat objek konfigurasi penyaringan.

   - Tentukan proporsi data yang harus digunakan untuk pelatihan ke `beta_value` parameter.
   - Tentukan jumlah sampel yang digunakan dalam perbandingan dengan `loss_history_length` parameter.

   Contoh kode berikut menunjukkan pengaturan sebuah objek dari `RelativeProbabilisticSiftConfig` kelas.

```
from smart_sifting.sift_config.sift_configs import (
    RelativeProbabilisticSiftConfig
    LossConfig
    SiftingBaseConfig
)

sift_config=RelativeProbabilisticSiftConfig(
    beta_value=0.5,
    loss_history_length=500,
    loss_based_sift_config=LossConfig(
         sift_config=SiftingBaseConfig(sift_delay=0)
    )
)
```

Untuk informasi selengkapnya tentang `loss_based_sift_config` parameter dan class terkait, lihat the section called “SageMaker modul konfigurasi penyaringan cerdas” di bagian referensi SageMaker Smart Sifting Python SDK.

`sift_config`Objek dalam contoh kode sebelumnya digunakan pada langkah 4 untuk menyiapkan kelas. `SiftingDataloader`

2. (Opsional) Konfigurasikan kelas transformasi batch penyaringan SageMaker cerdas.

   Kasus penggunaan pelatihan yang berbeda memerlukan format data pelatihan yang berbeda. Mengingat berbagai format data, algoritma penyaringan SageMaker cerdas perlu mengidentifikasi cara melakukan penyaringan pada batch tertentu. Untuk mengatasi hal ini, SageMaker smart sifting menyediakan modul transformasi batch yang membantu mengonversi batch menjadi format standar yang dapat disaring secara efisien.

   a. SageMaker smart sifting menangani transformasi batch data pelatihan dalam format berikut: Daftar Python, kamus, tupel, dan tensor. Untuk format data ini, SageMaker smart sifting secara otomatis menangani konversi format data batch, dan Anda dapat melewati sisa langkah ini. Jika Anda melewati langkah ini, pada langkah 4 untuk mengkonfigurasi`SiftingDataloader`, biarkan `batch_transforms` parameter `SiftingDataloader` ke nilai defaultnya, yaitu`None`.

   b. Jika kumpulan data Anda tidak dalam format ini, Anda harus melanjutkan ke sisa langkah ini untuk membuat transformasi batch khusus menggunakan`SiftingBatchTransform`.

Dalam kasus di mana kumpulan data Anda tidak berada dalam salah satu format yang didukung oleh penyaringan SageMaker cerdas, Anda mungkin mengalami kesalahan. Kesalahan format data tersebut dapat diatasi dengan menambahkan `batch_transforms` parameter `batch_format_index` or ke `SiftingDataloader` kelas, yang Anda atur di langkah 4. Berikut ini menunjukkan contoh kesalahan karena format data yang tidak kompatibel dan resolusi untuk mereka.

Pesan Kesalahan	Resolusi
Batch tipe *`{type (batch)}`* tidak didukung secara default.	Kesalahan ini menunjukkan format batch tidak didukung secara default. Anda harus menerapkan kelas transformasi batch kustom, dan menggunakannya dengan menentukannya ke `batch_transforms` parameter `SiftingDataloader` kelas.
Tidak dapat mengindeks kumpulan tipe *`{type (batch)}`*	Kesalahan ini menunjukkan objek batch tidak dapat diindeks secara normal. Pengguna harus menerapkan transform asi batch khusus dan meneruskan ini menggunakan `batch_transforms` parameter.
Ukuran batch *`{batch_size}`* tidak cocok dengan ukuran dimensi 0 atau dimensi 1	Kesalahan ini terjadi ketika ukuran batch yang disediakan tidak sesuai dengan dimensi ke-0 atau ke-1 dari batch. Pengguna harus menerapkan transform asi batch khusus dan meneruskan ini menggunakan `batch_transforms` parameter.

Dimensi 0 dan dimensi 1 cocok dengan ukuran batch	Kesalahan ini menunjukkan bahwa karena beberapa dimensi cocok dengan ukuran batch yang disediakan, informasi lebih lanjut diperlukan untuk menyaring batch. Pengguna dapat memberikan `batch_format_index` parameter untuk menunjukkan apakah batch dapat diindeks berdasarkan sampel atau fitur. Pengguna juga dapat menerapkan transformasi batch khusus, tetapi ini lebih banyak pekerjaan daripada yang diperluka n.

Untuk mengatasi masalah yang disebutkan di atas, Anda perlu membuat kelas transformasi batch khusus menggunakan `SiftingBatchTransform` modul. Kelas transformasi batch harus terdiri dari sepasang fungsi transformasi dan reverse-transform. Pasangan fungsi mengonversi format data Anda ke format yang dapat diproses oleh algoritme penyaringan SageMaker cerdas. Setelah Anda membuat kelas transformasi batch, kelas mengembalikan `SiftingBatch` objek yang akan Anda berikan ke `SiftingDataloader` kelas di langkah 4.

Berikut ini adalah contoh kelas transformasi batch kustom `SiftingBatchTransform` modul.

- Contoh implementasi transformasi batch daftar kustom dengan penyaringan SageMaker cerdas untuk kasus di mana potongan dataloader memiliki input, mask, dan label.

```
from typing import Any

import torch

from smart_sifting.data_model.data_model_interface import
 SiftingBatchTransform
from smart_sifting.data_model.list_batch import ListBatch

class ListBatchTransform(SiftingBatchTransform):
    def transform(self, batch: Any):
        inputs = batch[0].tolist()
```

```
        labels = batch[-1].tolist()  # assume the last one is the list of
 labels
        return ListBatch(inputs, labels)

    def reverse_transform(self, list_batch: ListBatch):
        a_batch = [torch.tensor(list_batch.inputs),
 torch.tensor(list_batch.labels)]
        return a_batch
```

- Contoh implementasi transformasi batch daftar kustom dengan penyaringan SageMaker cerdas untuk kasus di mana tidak ada label yang diperlukan untuk transformasi terbalik.

```
class ListBatchTransformNoLabels(SiftingBatchTransform):
    def transform(self, batch: Any):
        return ListBatch(batch[0].tolist())

    def reverse_transform(self, list_batch: ListBatch):
        a_batch = [torch.tensor(list_batch.inputs)]
        return a_batch
```

- Contoh implementasi batch tensor khusus dengan penyaringan SageMaker cerdas untuk kasus di mana potongan pemuat data memiliki input, masker, dan label.

```
from typing import Any

from smart_sifting.data_model.data_model_interface import
 SiftingBatchTransform
from smart_sifting.data_model.tensor_batch import TensorBatch

class TensorBatchTransform(SiftingBatchTransform):
    def transform(self, batch: Any):
        a_tensor_batch = TensorBatch(
            batch[0], batch[-1]
        )  # assume the last one is the list of labels
        return a_tensor_batch

    def reverse_transform(self, tensor_batch: TensorBatch):
        a_batch = [tensor_batch.inputs, tensor_batch.labels]
        return a_batch
```

Setelah Anda membuat `SiftingBatchTransform` kelas transformasi batch yang diimplementasikan, Anda menggunakan kelas ini di langkah 4 untuk menyiapkan kelas. `SiftingDataloader` Sisa dari panduan ini mengasumsikan bahwa `ListBatchTransform` kelas dibuat. Pada langkah 4, kelas ini diteruskan kebatch_transforms.

3. Buat kelas untuk mengimplementasikan `Loss` antarmuka penyaringan SageMaker cerdas. Tutorial ini mengasumsikan bahwa kelas diberi nama`SiftingImplementedLoss`. Saat menyiapkan kelas ini, kami sarankan Anda menggunakan fungsi kerugian yang sama dalam loop pelatihan model. Ikuti sublangkah berikut untuk membuat kelas `Loss` implementasi penyaringan SageMaker cerdas.

   a. SageMaker smart sifting menghitung nilai kerugian untuk setiap sampel data pelatihan, sebagai lawan menghitung nilai kerugian tunggal untuk batch. Untuk memastikan bahwa penyaringan SageMaker cerdas menggunakan logika perhitungan kerugian yang sama, buat fungsi smart-sifting-implemented kerugian menggunakan `Loss` modul penyaringan SageMaker pintar yang menggunakan fungsi kerugian Anda dan hitung kerugian per sampel pelatihan.

   > **Tip**
   >
   > SageMaker algoritma penyaringan cerdas berjalan pada setiap sampel data, bukan pada seluruh batch, jadi Anda harus menambahkan fungsi inisialisasi untuk mengatur fungsi PyTorch kerugian tanpa strategi pengurangan apa pun.
   >
   > ```
   > class SiftingImplementedLoss(Loss):
   >     def __init__(self):
   >         self.loss = torch.nn.CrossEntropyLoss(reduction='none')
   > ```
   >
   > Ini juga ditunjukkan dalam contoh kode berikut.

   b. Tentukan fungsi kerugian yang menerima `original_batch` (atau `transformed_batch` jika Anda telah menyiapkan transformasi batch pada langkah 2) dan PyTorch model. Menggunakan fungsi kerugian yang ditentukan tanpa pengurangan, SageMaker smart sifting menjalankan forward pass untuk setiap sampel data untuk mengevaluasi nilai kerugiannya.

Kode berikut adalah contoh dari smart-sifting-implemented `Loss` antarmuka bernama`SiftingImplementedLoss`.

```
from typing import Any

import torch
import torch.nn as nn
from torch import Tensor

from smart_sifting.data_model.data_model_interface import SiftingBatch
from smart_sifting.loss.abstract_sift_loss_module import Loss

model=... # a PyTorch model based on torch.nn.Module

class SiftingImplementedLoss(Loss):
    # You should add the following initializaztion function
    # to calculate loss per sample, not per batch.
    def __init__(self):
        self.loss_no_reduction = torch.nn.CrossEntropyLoss(reduction='none')

    def loss(
        self,
        model: torch.nn.Module,
        transformed_batch: SiftingBatch,
        original_batch: Any = None,
    ) -> torch.Tensor:
        device = next(model.parameters()).device
        batch = [t.to(device) for t in original_batch] # use this if you use
 original batch and skipped step 2
        # batch = [t.to(device) for t in transformed_batch] # use this if you
 transformed batches in step 2

        # compute loss
        outputs = model(batch)
        return self.loss_no_reduction(outputs.logits, batch[2])
```

Sebelum loop pelatihan mencapai pass maju yang sebenarnya, perhitungan kerugian penyaringan ini dilakukan selama fase pemuatan data pengambilan batch di setiap iterasi. Nilai kerugian individu kemudian dibandingkan dengan nilai kerugian sebelumnya, dan persentil

relatifnya diperkirakan per objek yang telah `RelativeProbabilisticSiftConfig` Anda atur pada langkah 1.

4. Bungkus pemuat PyTroch data dengan SageMaker `SiftingDataloader` kelas.

   Terakhir, gunakan semua kelas implementasi penyaringan SageMaker cerdas yang Anda konfigurasikan pada langkah sebelumnya ke kelas SageMaker `SiftingDataloder` konfigurasi. Kelas ini adalah pembungkus untuk PyTorch `DataLoader`. Dengan membungkus PyTorch`DataLoader`, penyaringan SageMaker cerdas terdaftar untuk dijalankan sebagai bagian dari pemuatan data di setiap iterasi pekerjaan pelatihan. PyTorch Contoh kode berikut menunjukkan penerapan penyaringan SageMaker data ke a. PyTorch `DataLoader`

```
from smart_sifting.dataloader.sift_dataloader import SiftingDataloader
from torch.utils.data import DataLoader

train_dataloader = DataLoader(...) # PyTorch data loader

# Wrap the PyTorch data loader by SiftingDataloder
train_dataloader = SiftingDataloader(
    sift_config=sift_config, # config object of RelativeProbabilisticSiftConfig
    orig_dataloader=train_dataloader,
    batch_transforms=ListBatchTransform(), # Optional, this is the custom class
 from step 2
    loss_impl=SiftingImplementedLoss(), # PyTorch loss function wrapped by the
 Sifting Loss interface
    model=model,
    log_batch_data=False
)
```

Trafo Hugging Face

Ada dua cara untuk menerapkan SageMaker smart sifting ke dalam kelas TransformersTrainer.

> **Note**
>
> Jika Anda menggunakan salah satu DLC untuk PyTorch dengan paket penyaringan SageMaker pintar diinstal, perhatikan bahwa Anda perlu menginstal perpustakaan. `transformers` Anda dapat menginstal paket tambahan dengan memperluas DLC

atau meneruskan `requirements.txt` ke kelas training job launcher for PyTorch () [sagemaker.pytorch.PyTorch](#)di Python SDK. SageMaker

Pengaturan sederhana

Cara paling sederhana untuk menerapkan SageMaker smart sifting ke dalam `Trainer` kelas Transformers adalah dengan menggunakan fungsi tersebut. `enable_sifting` Fungsi ini menerima objek yang ada, dan membungkus `Trainer` objek yang ada `DataLoader` dengan. `SiftingDataloader` Anda dapat terus menggunakan objek pelatihan yang sama. Lihat contoh penggunaan berikut.

```
from smart_sifting.integrations.trainer import enable_sifting
from smart_sifting.loss.abstract_sift_loss_module import Loss
from smart_sifting.sift_config.sift_configs import (
    RelativeProbabilisticSiftConfig
    LossConfig
    SiftingBaseConfig
)

class SiftingImplementedLoss(Loss):
   def loss(self, model, transformed_batch, original_batch):
        loss_fct = MSELoss(reduction="none") # make sure to set reduction to "none"
        logits = model.bert(**original_batch)
        return loss_fct(logits, original_batch.get("labels"))

sift_config = RelativeProbabilisticSiftConfig(
    beta_value=0.5,
    loss_history_length=500,
    loss_based_sift_config=LossConfig(
         sift_config=SiftingBaseConfig(sift_delay=0)
    )
)

trainer = Trainer(...)
enable_sifting(trainer, sift_config, loss=SiftingImplementedLoss()) # updates the
 trainer with Sifting Loss and config
trainer.train()
```

`SiftingDataloader`Kelas adalah pemuat data iterable. Ukuran pasti dari kumpulan data yang dihasilkan tidak diketahui sebelumnya karena pengambilan sampel acak selama pengayakan.

Akibatnya, Hugging `Trainer` Face mengharapkan argumen pelatihan `max_steps`. Perhatikan bahwa argumen ini mengesampingkan parameter konfigurasi epoch. `num_train_epochs` Jika pemuat data asli Anda juga dapat diulang, atau pelatihan Anda menggunakan `max_steps` dan satu epoch, maka akan `SiftingDataloader` melakukan hal yang sama dengan dataloader yang ada. Jika dataloader asli tidak dapat diulang atau `max_steps` tidak disediakan, Pelatih Wajah Pemeluk mungkin akan menampilkan pesan kesalahan yang mirip dengan berikut ini.

```
args.max_steps must be set to a positive value if dataloader does not have a length,
was -1
```

Untuk mengatasi ini, `enable_sifting` fungsi menyediakan `set_epochs` parameter opsional. Hal ini memungkinkan pelatihan dengan epoch, menggunakan jumlah epoch yang disediakan oleh argumen num_train_epochs `Trainer` kelas, dan set `max_steps` ke integer sistem maksimum, memungkinkan pelatihan untuk maju sampai epoch yang ditentukan telah selesai.

Pengaturan khusus

Untuk integrasi kustom dari SageMaker smart sifting dataloader, Anda dapat menggunakan kelas Hugging Face khusus. `Trainer` Dalam setiap subclass dari`Trainer`, `get_train_dataloader()` fungsi dapat diganti untuk mengembalikan objek kelas sebagai gantinya. `SiftingDataloader` Untuk kasus dengan pelatih khusus yang ada, pendekatan ini mungkin kurang mengganggu tetapi memerlukan perubahan kode daripada opsi pengaturan sederhana. Berikut ini adalah contoh implementasi SageMaker smart sifting ke dalam kelas Hugging Face kustom. `Trainer`

```
from smart_sifting.sift_config.sift_configs import (
    RelativeProbabilisticSiftConfig
    LossConfig
    SiftingBaseConfig
)
from smart_sifting.dataloader.sift_dataloader import SiftingDataloader
from smart_sifting.loss.abstract_sift_loss_module import Loss
from smart_sifting.data_model.data_model_interface import SiftingBatch,
 SiftingBatchTransform
from smart_sifting.data_model.list_batch import ListBatch

class SiftingListBatchTransform(SiftingBatchTransform):
    def transform(self, batch: Any):
        inputs = batch[0].tolist()
        labels = batch[-1].tolist()  # assume the last one is the list of labels
        return ListBatch(inputs, labels)
```

```
    def reverse_transform(self, list_batch: ListBatch):
        a_batch = [torch.tensor(list_batch.inputs), torch.tensor(list_batch.labels)]
        return a_batch

class SiftingImplementedLoss():
    # You should add the following initializaztion function
    # to calculate loss per sample, not per batch.
    def __init__(self):
        self.celoss = torch.nn.CrossEntropyLoss(reduction='none')

    def loss(
        self,
        model: torch.nn.Module,
        transformed_batch: SiftingBatch,
        original_batch: Any = None,
    ) -> torch.Tensor:
        device = next(model.parameters()).device
        batch = [t.to(device) for t in original_batch]

        # compute loss
        outputs = model(batch)
        return self.celoss(outputs.logits, batch[2])

class SiftingImplementedTrainer(Trainer):
    def get_train_dataloader(self):
        dl = super().get_train_dataloader()

        sift_config = RelativeProbabilisticSiftConfig(
            beta_value=0.5,
            loss_history_length=500,
            loss_based_sift_config=LossConfig(
                sift_config=SiftingBaseConfig(sift_delay=0)
            )
        )

        return SiftingDataloader(
                sift_config=sift_config,
                orig_dataloader=dl,
                batch_transforms=SiftingListBatchTransform(),
                loss_impl=SiftingImplementedLoss(),
                model=self.model
        )
```

Menggunakan `Trainer` kelas yang dibungkus, buat objek itu sebagai berikut.

```
trainer = SiftingImplementedTrainer(
    model=model,
    args=training_args,
    train_dataset=small_train_dataset,
    eval_dataset=small_eval_dataset
)

trainer.train()
```

Praktik terbaik, pertimbangan, dan pemecahan masalah

Praktik terbaik

- Pemilahan data secara cerdas SageMaker menggunakan pass maju tambahan untuk menganalisis dan memfilter data pelatihan Anda. Pada gilirannya, ada lebih sedikit lintasan mundur karena data yang kurang berdampak dikeluarkan dari pekerjaan pelatihan Anda. Karena itu, model yang memiliki lintasan mundur yang panjang atau mahal melihat keuntungan efisiensi terbesar saat menggunakan penyaringan pintar. Sementara itu, jika forward pass model Anda membutuhkan waktu lebih lama dari backward pass, overhead dapat meningkatkan total waktu pelatihan. Untuk mengukur waktu yang dihabiskan oleh setiap pass, Anda dapat menjalankan pekerjaan pelatihan pilot dan mengumpulkan log yang mencatat waktu pada proses. Juga pertimbangkan untuk menggunakan SageMaker Profiler yang menyediakan alat profiling dan aplikasi UI. Untuk mempelajari selengkapnya, lihat Menggunakan Amazon SageMaker Profiler untuk membuat profil aktivitas pada sumber daya AWS komputasi.
- SageMaker smart sifting mendukung pelatihan PyTorch model dengan paralelisme data tradisional dan paralelisme data terdistribusi, yang membuat replika model di semua pekerja GPU dan menggunakan operasi. `AllReduce` Ini tidak bekerja dengan teknik paralelisme model, termasuk paralelisme data sharded.
- Karena penyaringan SageMaker cerdas berfungsi untuk pekerjaan paralelisme data, pastikan model yang Anda latih cocok di setiap memori GPU.
- SageMaker smart sifting berjalan pada data individual dalam batch selama pemuatan data, jadi pastikan Anda mengatur strategi pengurangan fungsi PyTorch kerugian menjadi non-reduksi. `"none"` Dengan `reduction` disetel ke `"mean"` or`"sum"`, fungsi kerugian mengembalikan nilai kerugian tunggal, yang menyebabkan penyaringan SageMaker cerdas tidak berfungsi dengan baik.

Pemecahan Masalah

Jika Anda mengalami kesalahan, Anda dapat menggunakan daftar berikut untuk mencoba memecahkan masalah. Jika Anda membutuhkan dukungan lebih lanjut, hubungi SageMaker tim di <sm-smart-sifting-feedback@amazon .com>.

Pengecualian dari perpustakaan penyaringan SageMaker pintar

Gunakan referensi pengecualian berikut yang diajukan oleh pustaka penyaringan SageMaker pintar untuk memecahkan masalah kesalahan dan mengidentifikasi penyebabnya.

Nama Pengecualian	Deskripsi
SiftConfigValidationException	SageMaker Dilempar dari pustaka penyaring an pintar jika ada kunci Config yang hilang atau tipe nilai yang tidak didukung untuk Kunci Sift
UnsupportedDataFormatException	SageMaker Dilempar dari perpustakaan penyaringan pintar jika ada yang tidak didukung DataFormat untuk logika Penyaringan
LossImplementationNotProvidedException	Dilempar jika hilang atau tidak menerapkan antarmuka Loss

Keamanan dalam penyaringan SageMaker cerdas

Karena pustaka penyaringan SageMaker cerdas menjalankan proses menghapus sampel pelatihan yang kurang berharga, diperlukan akses penuh ke kumpulan data pelatihan karena diproduksi oleh pemuat data. Akses ini tidak berbeda dengan akses yang sudah disediakan PyTorch dalam skenario pelatihan normal.

SageMaker penyaringan cerdas memiliki logging bawaan dengan implikasi keamanan. Secara default, SageMaker smart sifting logs hanyalah log tingkat aplikasi yang berisi metrik, latensi, dan kesalahan atau peringatan pengguna. Namun, pengguna dapat memilih untuk mengaktifkan log verbose, yang mencatat data batch penuh untuk menunjukkan sampel mana yang dihapus dari batch tertentu. Log ini dipancarkan menggunakan logger Python dan tidak diunggah atau disimpan di mana pun oleh perpustakaan. Dalam hal pengunggahan log otomatis ke CloudWatch atau layanan serupa, harap dicatat bahwa menggunakan log verbose dapat mengakibatkan data pelatihan sensitif diunggah dari instance pelatihan.

Di luar pencatatan yang disebutkan di atas, penyaringan SageMaker cerdas tidak memiliki fungsi jaringan apa pun juga tidak berinteraksi dengan sistem file lokal. Data pengguna disimpan sebagai objek dalam memori untuk keseluruhan waktu yang digunakan oleh perpustakaan.

SageMaker referensi SDK Python penyaringan cerdas

Halaman ini menyediakan referensi modul Python yang Anda butuhkan untuk menerapkan penyaringan SageMaker cerdas ke skrip pelatihan Anda.

SageMaker modul konfigurasi penyaringan cerdas

class smart_sifting.sift_config.sift_configs.RelativeProbabilisticSiftConfig()

Kelas konfigurasi penyaringan SageMaker cerdas.

Parameter

- `beta_value`(float) — Nilai beta (konstan) untuk menghitung probabilitas memilih sampel untuk pelatihan berdasarkan persentil kerugian dalam sejarah nilai kerugian. Menurunkan nilai beta menghasilkan persentase data yang diayak lebih rendah, dan meningkatkannya menghasilkan persentase data yang lebih tinggi yang diayak. Tidak ada nilai minimum atau maksimum untuk nilai beta, selain itu harus nilai positif. Lihat tabel referensi berikut untuk tingkat pengayakan sehubungan `beta_value` dengan.

`beta_value`	Proporsi data yang disimpan (%)	Proporsi data yang diayak (%)
0.1	90,91	9,01
0,25	80	20
0,5	66,67	33.33
1	50	50
2	33.33	66,67
3	25	75

10	9.09	90,92
100	0,99	99,01

- `loss_history_length`(int) — Jumlah kerugian pelatihan sebelumnya yang harus disimpan untuk pengambilan sampel berbasis kerugian ambang relatif.
- `loss_based_sift_config`(dict atau `LossConfig` objek) — Tentukan `LossConfig` objek yang mengembalikan konfigurasi antarmuka Loss penyaringan SageMaker cerdas.

class smart_sifting.sift_config.sift_configs.LossConfig()

Kelas konfigurasi untuk `loss_based_sift_config` parameter `RelativeProbabilisticSiftConfig` kelas.

Parameter

- `sift_config`(dict atau `SiftingBaseConfig` objek) - Tentukan `SiftingBaseConfig` objek yang mengembalikan kamus konfigurasi dasar penyaringan.

class smart_sifting.sift_config.sift_configs.SiftingBaseConfig()

Kelas konfigurasi untuk `sift_config` parameter`LossConfig`.

Parameter

- `sift_delay`(int) — Jumlah langkah pelatihan yang harus menunggu sebelum mulai menyaring. Kami menyarankan Anda mulai menyaring setelah semua lapisan dalam model memiliki pandangan yang cukup dari data pelatihan. Nilai default-nya adalah `1000`.
- `repeat_delay_per_epoch`(bool) — Tentukan apakah akan menunda penyaringan setiap zaman. Nilai default-nya adalah `False`.

SageMaker modul transformasi batch data penyaringan cerdas

class smart_sifting.data_model.data_model_interface.SiftingBatchTransform

Modul Python penyaringan SageMaker cerdas untuk menentukan cara melakukan transformasi batch. Dengan menggunakan ini, Anda dapat menyiapkan kelas transformasi batch yang mengonversi format data data pelatihan Anda ke `SiftingBatch` format, yang dapat disaring dan diakumulasikan oleh SageMaker smart sifting ke dalam batch yang diayak.

class smart_sifting.data_model.data_model_interface.SiftingBatch

Antarmuka untuk menentukan tipe data batch yang dapat diayak dan diakumulasikan.

class smart_sifting.data_model.list_batch.ListBatch

Modul untuk melacak batch daftar untuk penyaringan.

class smart_sifting.data_model.tensor_batch.TensorBatch

Modul untuk melacak batch tensor untuk diayak.

SageMaker modul implementasi kerugian penyaringan cerdas

class smart_sifting.loss.abstract_sift_loss_module.Loss

Modul pembungkus untuk mendaftarkan antarmuka penyaringan SageMaker pintar ke fungsi kehilangan model PyTorch berbasis.

SageMaker modul pembungkus pemuat data pengayak cerdas

class smart_sifting.dataloader.sift_dataloader.SiftingDataloader

Modul pembungkus untuk mendaftarkan antarmuka penyaringan SageMaker cerdas ke pemuat data model PyTorch berbasis.

Iterator Main Sifting Dataloader menyaring sampel pelatihan dari dataloader berdasarkan konfigurasi ayakan

Parameter

- `sift_config`(dict atau `RelativeProbabilisticSiftConfig` objek) — Sebuah `RelativeProbabilisticSiftConfig` objek.
- `orig_dataloader`(PyTorch DataLoader objek) - Tentukan objek PyTorch Dataloader yang akan dibungkus.
- `batch_transforms`(`SiftingBatchTransform`objek) — (Opsional) Jika format data Anda tidak didukung oleh transformasi default perpustakaan penyaringan SageMaker pintar, Anda perlu membuat kelas transformasi batch menggunakan `SiftingBatchTransform` modul. Parameter ini adalah untuk melewati kelas transformasi batch yang `SiftingDataloader` dapat dijalankan untuk mengonversi data agar algoritma penyaringan SageMaker cerdas dapat diterima.

- `model`(objek PyTorch model) — PyTorch Model asli
- `loss_impl`(fungsi kehilangan penyaringansmart_sifting.loss.abstract_sift_loss_module.Loss) - Fungsi kehilangan penyaringan yang dikonfigurasi dengan `Loss` modul dan membungkus fungsi kerugian. PyTorch
- `log_batch_data`(bool) - Tentukan apakah akan mencatat data batch. Jika disetel ke`True`, SageMaker smart sifting mencatat detail batch yang disimpan atau diayak. Kami menyarankan Anda menyalakannya hanya untuk pekerjaan pelatihan pilot. Saat logging aktif, sampel dimuat ke GPU dan ditransfer ke CPU, yang memperkenalkan overhead. Nilai default-nya adalah `False`.

SageMaker catatan rilis penyaringan cerdas

Lihat catatan rilis berikut untuk melacak pembaruan terbaru untuk kemampuan penyaringan SageMaker cerdas.

SageMaker catatan rilis smart sifting: 29 November 2023

Fitur Baru

- Meluncurkan perpustakaan penyaringan SageMaker pintar Amazon di AWS re:Invent 2023.

Migrasi ke AWS Deep Learning Containers

- Pustaka penyaringan SageMaker cerdas lulus pengujian integrasi dan tersedia dalam AWS Deep Learning Containers. Untuk menemukan daftar lengkap kontainer pra-bangun dengan pustaka penyaringan SageMaker pintar, lihat. the section called “Kerangka kerja dan AWS Wilayah yang didukung”

Debug dan tingkatkan kinerja model

Inti dari pelatihan model pembelajaran mesin, jaringan saraf pembelajaran mendalam, model transformator adalah dalam mencapai konvergensi model yang stabil, dan dengan demikian, state-of-the-art model memiliki jutaan, miliaran, atau triliunan parameter model. Jumlah operasi untuk memperbarui sejumlah besar parameter model selama setiap iterasi dapat dengan mudah menjadi astronomi. Untuk mengidentifikasi masalah konvergensi model, penting untuk dapat mengakses parameter model, aktivasi, dan gradien yang dihitung selama proses pengoptimalan.

Amazon SageMaker menyediakan dua alat debugging untuk membantu mengidentifikasi masalah konvergensi tersebut dan mendapatkan visibilitas ke dalam model Anda.

Amazon SageMaker dengan TensorBoard

Untuk menawarkan kompatibilitas yang lebih besar dengan alat komunitas sumber terbuka dalam platform SageMaker Pelatihan, SageMaker host TensorBoard sebagai aplikasi di Domain. SageMaker Anda dapat membawa pekerjaan pelatihan Anda SageMaker dan terus menggunakan penulis TensorBoard ringkasan untuk mengumpulkan tensor keluaran model. Karena TensorBoard diimplementasikan ke dalam SageMaker Domain, ini juga memberi Anda lebih banyak opsi untuk mengelola profil pengguna di bawah SageMaker Domain di AWS akun Anda, dan memberikan kontrol yang baik atas profil pengguna dengan memberikan akses ke tindakan dan sumber daya tertentu. Untuk mempelajari selengkapnya, lihat the section called “Gunakan TensorBoard”.

Amazon SageMaker Debugger

Amazon SageMaker Debugger adalah kemampuan SageMaker yang menyediakan alat untuk mendaftarkan kait ke panggilan balik untuk mengekstrak tensor keluaran model dan menyimpannya di Amazon Simple Storage Service. Ini menyediakan aturan bawaan untuk mendeteksi masalah konvergensi model, seperti overfitting, fungsi aktivasi jenuh, gradien menghilang, dan banyak lagi. Anda juga dapat mengatur aturan bawaan dengan Amazon CloudWatch Events dan AWS Lambda untuk mengambil tindakan otomatis terhadap masalah yang terdeteksi, dan menyiapkan Layanan Pemberitahuan Sederhana Amazon untuk menerima pemberitahuan email atau teks. Untuk mempelajari informasi lebih lanjut, lihat the section called “Gunakan SageMaker Debugger”.

Topik

- Gunakan TensorBoard untuk men-debug dan menganalisis pekerjaan pelatihan di Amazon SageMaker
- Gunakan Amazon SageMaker Debugger untuk men-debug dan meningkatkan kinerja model
- Akses wadah pelatihan AWS Systems Manager untuk debugging jarak jauh
- Catatan rilis untuk kemampuan debugging Amazon SageMaker

Gunakan TensorBoard untuk men-debug dan menganalisis pekerjaan pelatihan di Amazon SageMaker

Amazon SageMaker with TensorBoard adalah kemampuan Amazon SageMaker yang membawa alat visualisasi TensorBoardke SageMaker, terintegrasi dengan SageMaker Pelatihan dan Domain.

Ini menyediakan opsi untuk mengelola AWS akun Anda dan pengguna yang termasuk dalam akun melalui SageMaker Domain, untuk memberi pengguna Domain akses ke TensorBoard data dengan izin yang sesuai ke Amazon S3, dan membantu pengguna Domain melakukan tugas debugging model menggunakan plugin visualisasi. TensorBoard SageMaker dengan TensorBoard diperluas dengan plugin SageMaker Data Manager, dengan mana pengguna Domain dapat mengakses sejumlah pekerjaan pelatihan di satu tempat dalam TensorBoard aplikasi.

> **Note**
>
> Fitur ini untuk pelatihan dan debugging model pembelajaran mendalam menggunakan PyTorch atau TensorFlow kerangka kerja.

Untuk ilmuwan data

Pelatihan model besar dapat memiliki masalah ilmiah yang mengharuskan ilmuwan data untuk men-debug dan menyelesaikannya untuk meningkatkan konvergensi model dan menstabilkan proses penurunan gradien.

Saat Anda mengalami masalah pelatihan model, seperti kehilangan tidak konvergen, atau bobot dan gradien yang hilang atau meledak, Anda perlu mengakses data tensor untuk menyelam lebih dalam dan menganalisis parameter model, skalar, dan metrik khusus apa pun. SageMaker Dengan menggunakan with TensorBoard, Anda dapat memvisualisasikan tensor keluaran model yang diekstraksi dari pekerjaan pelatihan. Saat Anda bereksperimen dengan model yang berbeda, beberapa latihan, dan model hiperparameter, Anda dapat memilih beberapa pekerjaan pelatihan TensorBoard dan membandingkannya di satu tempat.

Untuk administrator

Melalui halaman TensorBoard arahan di SageMaker konsol atau SageMaker Domain, Anda dapat mengelola pengguna TensorBoard aplikasi jika Anda adalah administrator AWS akun atau SageMaker Domain. Setiap pengguna Domain dapat mengakses TensorBoard aplikasi mereka sendiri dengan izin yang diberikan. Sebagai administrator SageMaker Domain dan pengguna Domain, Anda dapat membuat dan menghapus TensorBoard aplikasi yang diberikan tingkat izin yang Anda miliki.

Kerangka kerja yang didukung dan Wilayah AWS

Fitur ini mendukung kerangka kerja pembelajaran mesin berikut danWilayah AWS.

Kerangka

- PyTorch
- TensorFlow
- Trafo Hugging Face

Wilayah AWS

- US East (N. Virginia) (`us-east-1`)
- US East (Ohio) (`us-east-2`)
- US West (Oregon) (`us-west-2`)
- Europe (Frankfurt) (`eu-central-1`)
- Europe (Ireland) (`eu-west-1`)

> **Note**
>
> Amazon SageMaker dengan TensorBoard menjalankan TensorBoard aplikasi pada `ml.r5.large` instans dan menimbulkan biaya setelah tingkat SageMaker gratis atau periode uji coba gratis fitur tersebut. Untuk informasi lebih lanjut, lihat Amazon SageMaker Harga.

Prasyarat

Daftar berikut menunjukkan prasyarat untuk mulai menggunakan dengan. SageMaker TensorBoard

- SageMaker Domain yang disiapkan dengan Amazon VPC di akun AndaAWS.

  Untuk petunjuk cara menyiapkan Domain, lihat Onboard to Amazon SageMaker Domain menggunakan penyiapan cepat. Anda juga perlu menambahkan profil pengguna Domain untuk pengguna individu untuk mengakses TensorBoard on SageMaker. Untuk informasi selengkapnya, lihat Menambahkan dan menghapus profil pengguna SageMaker Domain.
- Daftar berikut adalah set minimum izin untuk digunakan TensorBoard pada SageMaker.
  - `sagemaker:CreateApp`
  - `sagemaker:DeleteApp`

- `sagemaker:DescribeTrainingJob`
- `sagemaker:Search`
- `s3:GetObject`
- `s3:ListBucket`

Siapkan pekerjaan pelatihan dengan konfigurasi data TensorBoard keluaran

Pekerjaan pelatihan khas untuk pembelajaran mendalam SageMaker terdiri dari dua langkah utama: menyiapkan skrip pelatihan dan mengonfigurasi Peluncur pekerjaan SageMaker Pelatihan. Di bagian ini, Anda dapat memeriksa perubahan yang diperlukan untuk mengumpulkan data TensorBoard yang kompatibel dari SageMaker Pelatihan.

Langkah 1: Ubah skrip pelatihan Anda

Pastikan Anda menentukan tensor dan skalar keluaran mana yang akan dikumpulkan, dan ubah baris kode dalam skrip pelatihan Anda menggunakan salah satu alat berikut: TensorBoard X, TensorFlow Summary Writer, Summary Writer, atau PyTorch Debugger. SageMaker

Pastikan juga bahwa Anda menentukan jalur keluaran TensorBoard data sebagai direktori log (`log_dir`) untuk callback dalam wadah pelatihan.

Untuk informasi selengkapnya tentang callback per framework, lihat sumber daya berikut.

- Untuk PyTorch, gunakan torch.utils.tensorboard. SummaryWriter. Lihat juga bagian Using TensorBoard in PyTorch dan Log skalar di PyTorchtutorial. Atau, Anda dapat menggunakan TensorBoardX Summary Writer.

```
LOG_DIR="/opt/ml/output/tensorboard"
tensorboard_callback=torch.utils.tensorboard.writer.SummaryWriter(log_dir=LOG_DIR)
```

- Untuk TensorFlow, gunakan callback asli untuk, TensorBoard tf.keras.callbacks. TensorBoard.

```
LOG_DIR="/opt/ml/output/tensorboard"
tensorboard_callback=tf.keras.callbacks.TensorBoard(
    log_dir=LOG_DIR, histogram_freq=1)
```

- Untuk Transformers dengan PyTorch, Anda dapat menggunakan transformers.integrations. TensorBoardCallback.

Untuk Transformers dengan TensorFlow, gunakan`tf.keras.tensorboard.callback`, dan teruskan ke callback keras di transformer.

> **Tip**
>
> Anda juga dapat menggunakan jalur keluaran lokal kontainer yang berbeda. Namun, di[Langkah 2: Bangun peluncur SageMaker pelatihan dengan konfigurasi data TensorBoard](), Anda harus memetakan jalur dengan benar SageMaker agar berhasil mencari jalur lokal dan menyimpan TensorBoard data ke bucket keluaran S3.

- Untuk panduan tentang memodifikasi skrip pelatihan menggunakan pustaka Python SageMaker Debugger, lihat. [the section called “Langkah 1: Sesuaikan Skrip Pelatihan Anda untuk Mendaftarkan Hook”]()

Langkah 2: Bangun peluncur SageMaker pelatihan dengan konfigurasi data TensorBoard

Gunakan `sagemaker.debugger.TensorBoardOutputConfig` saat mengonfigurasi estimator SageMaker kerangka kerja. API konfigurasi ini memetakan bucket S3 yang Anda tentukan untuk menyimpan TensorBoard data dengan jalur lokal di container pelatihan (`/opt/ml/output/tensorboard`). Lewatkan objek modul ke `tensorboard_output_config` parameter kelas estimator. Cuplikan kode berikut menunjukkan contoh mempersiapkan TensorFlow estimator dengan parameter konfigurasi TensorBoard output.

> **Note**
>
> Contoh ini mengasumsikan bahwa Anda menggunakan SageMaker Python SDK. Jika Anda menggunakan SageMaker API tingkat rendah, Anda harus menyertakan yang berikut ini ke sintaks permintaan API. [CreateTrainingJob]()
>
> ```
> "TensorBoardOutputConfig": {
>   "LocalPath": "/opt/ml/output/tensorboard",
>   "S3OutputPath": "s3_output_bucket"
> }
> ```

```
from sagemaker.tensorflow import TensorFlow
from sagemaker.debugger import TensorBoardOutputConfig
```

```
# Set variables for training job information,
# such as s3_out_bucket and other unique tags.
...

LOG_DIR="/opt/ml/output/tensorboard"

output_path = os.path.join(
    "s3_output_bucket", "sagemaker-output", "date_str", "your-training_job_name"
)

tensorboard_output_config = TensorBoardOutputConfig(
    s3_output_path=os.path.join(output_path, 'tensorboard'),
    container_local_output_path=LOG_DIR
)

estimator = TensorFlow(
    entry_point="train.py",
    source_dir="src",
    role=role,
    image_uri=image_uri,
    instance_count=1,
    instance_type="ml.c5.xlarge",
    base_job_name="your-training_job_name",
    tensorboard_output_config=tensorboard_output_config,
    hyperparameters=hyperparameters
)
```

Cara mengakses TensorBoard di SageMaker

Anda dapat mengakses TensorBoard dengan dua metode: secara terprogram menggunakan `sagemaker.interactive_apps.tensorboard` modul yang menghasilkan URL yang tidak ditandatangani atau ditetapkan sebelumnya, atau menggunakan halaman TensorBoard arahan di konsol. SageMaker Setelah Anda membuka TensorBoard, SageMaker menjalankan TensorBoard plugin dan secara otomatis menemukan semua data output pekerjaan pelatihan dalam format file TensorBoard -kompatibel.

Topik

- Buka TensorBoard menggunakan sagemaker.interactive_apps.tensorboard modul
- Buka TensorBoard menggunakan get_app_url fungsi sebagai metode estimator kelas
- Buka TensorBoard melalui SageMaker konsol

Buka TensorBoard menggunakan **sagemaker.interactive_apps.tensorboard** modul

`sagemaker.interactive_apps.tensorboard`Modul ini menyediakan fungsi yang disebut `get_app_url` yang menghasilkan URL yang tidak ditandatangani atau ditetapkan sebelumnya untuk membuka TensorBoard aplikasi di lingkungan apa pun di atau Amazon SageMaker EC2. Ini untuk memberikan pengalaman terpadu bagi pengguna Studio Classic dan non-Studio Classic. Untuk lingkungan Studio, Anda dapat membuka TensorBoard dengan menjalankan `get_app_url()` fungsi apa adanya, atau Anda juga dapat menentukan nama pekerjaan untuk mulai melacak saat TensorBoard aplikasi terbuka. Untuk lingkungan non-Studio Classic, Anda dapat membuka TensorBoard dengan memberikan informasi Domain dan profil pengguna Anda ke fungsi utilitas. Dengan fungsi ini, terlepas dari di mana atau bagaimana Anda menjalankan kode pelatihan dan meluncurkan pekerjaan pelatihan, Anda dapat langsung mengakses TensorBoard dengan menjalankan `get_app_url` fungsi di notebook atau terminal Jupyter Anda.

> **Note**
>
> Fungsionalitas ini tersedia di SageMaker Python SDK v2.184.0 dan yang lebih baru. Untuk menggunakan fungsi ini, pastikan Anda memutakhirkan SDK dengan menjalankan`pip install sagemaker --upgrade`.

Topik

- Opsi 1: Untuk SageMaker Studio Classic
- Opsi 2: Untuk lingkungan Klasik non-Studio

Opsi 1: Untuk SageMaker Studio Classic

Jika Anda menggunakan SageMaker Studio Classic, Anda dapat langsung membuka TensorBoard aplikasi atau mengambil URL yang tidak ditandatangani dengan menjalankan `get_app_url` fungsi sebagai berikut. Karena Anda sudah berada dalam lingkungan Studio Classic dan masuk sebagai pengguna Domain, buat URL `get_app_url()` yang tidak ditandatangani karena tidak perlu mengautentikasi lagi.

Untuk membuka TensorBoard aplikasi

Kode berikut secara otomatis membuka TensorBoard aplikasi dari URL yang tidak ditandatangani yang dikembalikan `get_app_url()` fungsi di browser web default lingkungan Anda.

```
from sagemaker.interactive_apps import tensorboard

region = "us-west-2"
app = tensorboard.TensorBoardApp(region)

app.get_app_url(
    training_job_name="your-training_job_name" # Optional. Specify the job name to
 track a specific training job
)
```

Untuk mengambil URL yang tidak ditandatangani dan membuka aplikasi secara manual TensorBoard

Kode berikut mencetak URL yang tidak ditandatangani yang dapat Anda salin ke browser web dan membuka TensorBoard aplikasi.

```
from sagemaker.interactive_apps import tensorboard

region = "us-west-2"
app = tensorboard.TensorBoardApp(region)
print("Navigate to the following URL:")
print(
    app.get_app_url(
        training_job_name="your-training_job_name", # Optional. Specify the name of the
 job to track.
        open_in_default_web_browser=False           # Set to False to print the URL to
 terminal.
    )
)
```

Perhatikan bahwa jika Anda menjalankan dua contoh kode sebelumnya di luar lingkungan SageMaker Studio Classic, fungsi tersebut akan mengembalikan URL ke halaman TensorBoard landing di SageMaker konsol, karena ini tidak memiliki informasi login ke Domain dan profil pengguna Anda. Untuk membuat URL presigned, lihat Opsi 2 di bagian berikut.

Opsi 2: Untuk lingkungan Klasik non-Studio

Jika Anda menggunakan lingkungan non-Studio Classic, seperti instans SageMaker Notebook atau Amazon EC2, dan ingin TensorBoard membuka langsung dari lingkungan tempat Anda berada, Anda perlu membuat URL yang telah ditetapkan sebelumnya dengan informasi Domain dan profil pengguna Anda. URL presigned adalah URL yang masuk ke Amazon SageMaker Studio Classic saat URL dibuat dengan Domain dan profil pengguna Anda, dan karenanya diberikan akses ke semua

aplikasi dan file Domain yang terkait dengan Domain Anda. Untuk membuka TensorBoard melalui URL yang telah ditetapkan sebelumnya, gunakan `get_app_url` fungsi dengan nama Domain dan profil pengguna Anda sebagai berikut.

Perhatikan bahwa opsi ini mengharuskan pengguna Domain untuk memiliki `sagemaker:CreatePresignedDomainUrl` izin. Tanpa izin, pengguna Domain akan menerima kesalahan pengecualian.

> **Important**
>
> Jangan membagikan URL yang telah ditetapkan sebelumnya. `get_app_url`Fungsi ini membuat URL presigned, yang secara otomatis mengautentikasi dengan Domain dan profil pengguna Anda dan memberikan akses ke aplikasi dan file apa pun yang terkait dengan Domain Anda.

```
print(
    app.get_app_url(
        training_job_name="your-training_job_name", # Optional. Specify the name of the
 job to track.
        create_presigned_domain_url=True,           # Required to be set to True for
 creating a presigned URL.
        domain_id="your-domain-id",                 # Required if creating a presigned
 URL (create_presigned_domain_url=True).
        user_profile_name="your-user-profile-name", # Required if creating a presigned
 URL (create_presigned_domain_url=True).
        open_in_default_web_browser=False,          # Optional. Set to False to print
 the URL to terminal.
        optional_create_presigned_url_kwargs={}     # Optional. Add any additional args
 for Boto3 create_presigned_domain_url
    )
)
```

> **Tip**
>
> `get_app_url`Fungsi ini menjalankan [SageMaker.Client.create_presigned_domain_url](#)API AWS SDK for Python (Boto3) di backend. Karena Boto3 `create_presigned_domain_url` API membuat URL domain presigned yang kedaluwarsa dalam 300 detik secara default, URL TensorBoard aplikasi presigned juga kedaluwarsa dalam 300 detik. Jika Anda ingin

memperpanjang waktu kedaluwarsa, berikan `ExpiresInSeconds` argumen ke `optional_create_presigned_url_kwargs` argumen `get_app_url` fungsi sebagai berikut.

```
optional_create_presigned_url_kwargs={"ExpiresInSeconds": 1500}
```

> **Note**
>
> Jika salah satu masukan Anda yang diteruskan ke argumen tidak valid, fungsi akan mengeluarkan URL ke halaman TensorBoard arahan alih-alih membuka aplikasi. `get_app_url` TensorBoard Pesan output akan mirip dengan yang berikut ini.
>
> ```
> Navigate to the following URL:
> https://us-west-2.console.aws.amazon.com/sagemaker/home?region=us-west-2#/
> tensor-board-landing
> ```

Buka TensorBoard menggunakan **`get_app_url`** fungsi sebagai metode **`estimator`** kelas

Jika Anda sedang dalam proses menjalankan pekerjaan pelatihan menggunakan `estimator` kelas SDK SageMaker Python dan memiliki objek aktif `estimator` kelas, Anda juga dapat mengakses `get_app_url`fungsi sebagai metode kelas kelas. `estimator` Buka TensorBoard aplikasi atau ambil URL yang tidak ditandatangani dengan menjalankan `get_app_url` metode sebagai berikut. Metode `get_app_url` kelas menarik nama pekerjaan pelatihan dari estimator dan membuka TensorBoard aplikasi dengan pekerjaan yang ditentukan.

> **Note**
>
> Fungsionalitas ini tersedia di SageMaker Python SDK v2.184.0 dan yang lebih baru. Untuk menggunakan fungsi ini, pastikan Anda memutakhirkan SDK dengan menjalankan`pip install sagemaker --upgrade`.

Topik

- Opsi 1: Untuk SageMaker Studio Classic
- Opsi 2: Untuk lingkungan Klasik non-Studio

Opsi 1: Untuk SageMaker Studio Classic

Untuk membuka TensorBoard aplikasi

Kode berikut secara otomatis membuka TensorBoard aplikasi dari URL yang tidak ditandatangani yang dikembalikan `get_app_url()` metode di browser web default lingkungan Anda.

```
estimator.get_app_url(
    app_type=SupportedInteractiveAppTypes.TENSORBOARD # Required.
)
```

Untuk mengambil URL yang tidak ditandatangani dan membuka aplikasi secara manual TensorBoard

Kode berikut mencetak URL yang tidak ditandatangani yang dapat Anda salin ke browser web dan membuka TensorBoard aplikasi.

```
print(
    estimator.get_app_url(
        app_type=SupportedInteractiveAppTypes.TENSORBOARD, # Required.
        open_in_default_web_browser=False, # Optional. Set to False to print the URL to
 terminal.
    )
)
```

Perhatikan bahwa jika Anda menjalankan dua contoh kode sebelumnya di luar lingkungan SageMaker Studio Classic, fungsi tersebut akan mengembalikan URL ke halaman TensorBoard landing di SageMaker konsol, karena ini tidak memiliki informasi login ke Domain dan profil pengguna Anda. Untuk membuat URL presigned, lihat Opsi 2 di bagian berikut.

Opsi 2: Untuk lingkungan Klasik non-Studio

Jika Anda menggunakan lingkungan non-Studio Classic, seperti instans SageMaker Notebook dan Amazon EC2, dan ingin membuat URL yang telah ditetapkan sebelumnya untuk membuka TensorBoard aplikasi, gunakan `get_app_url` metode dengan informasi Domain dan profil pengguna Anda sebagai berikut.

Perhatikan bahwa opsi ini mengharuskan pengguna Domain untuk memiliki `sagemaker:CreatePresignedDomainUrl` izin. Tanpa izin, pengguna Domain akan menerima kesalahan pengecualian.

> **Important**
>
> Jangan membagikan URL yang telah ditetapkan sebelumnya. `get_app_url`Fungsi ini membuat URL presigned, yang secara otomatis mengautentikasi dengan Domain dan profil pengguna Anda dan memberikan akses ke aplikasi dan file apa pun yang terkait dengan Domain Anda.

```
print(
    estimator.get_app_url(
        app_type=SupportedInteractiveAppTypes.TENSORBOARD, # Required
        create_presigned_domain_url=True,           # Reguired to be set to True for
 creating a presigned URL.
        domain_id="your-domain-id",                 # Required if creating a presigned
 URL (create_presigned_domain_url=True).
        user_profile_name="your-user-profile-name", # Required if creating a presigned
 URL (create_presigned_domain_url=True).
        open_in_default_web_browser=False,            # Optional. Set to False to print
 the URL to terminal.
        optional_create_presigned_url_kwargs={}       # Optional. Add any additional
 args for Boto3 create_presigned_domain_url
    )
)
```

Buka TensorBoard melalui SageMaker konsol

Anda juga dapat menggunakan UI SageMaker konsol untuk membuka TensorBoard aplikasi. Ada dua opsi untuk membuka TensorBoard aplikasi melalui SageMaker konsol.

Topik

- Opsi 1: Luncurkan TensorBoard dari halaman detail Domain
- Opsi 2: Luncurkan TensorBoard dari halaman TensorBoard arahan

Opsi 1: Luncurkan TensorBoard dari halaman detail Domain

Arahkan ke halaman detail Domain

Prosedur berikut menunjukkan cara menavigasi ke halaman detail Domain.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.

2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Dari daftar Domain, pilih Domain tempat Anda ingin meluncurkan TensorBoard aplikasi.

Luncurkan aplikasi profil pengguna

Prosedur berikut menunjukkan cara meluncurkan aplikasi Studio Classic yang dicakup ke profil pengguna.

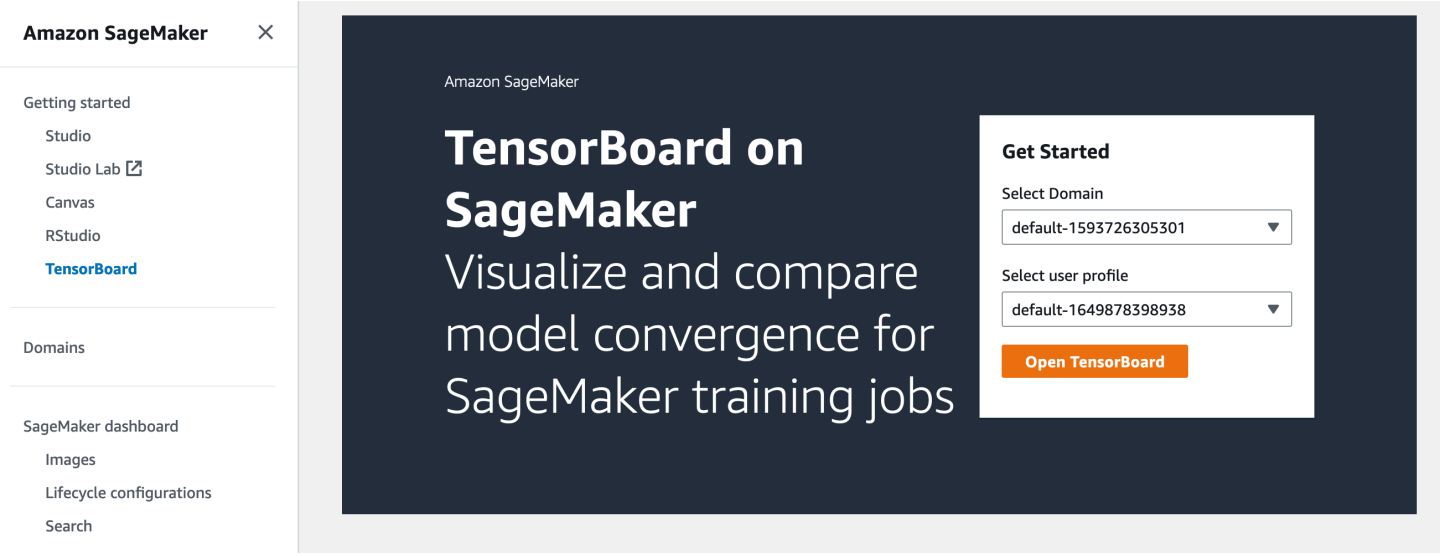
1. Pada halaman Detail domain, pilih tab Profil pengguna.
2. Identifikasi profil pengguna yang ingin Anda luncurkan aplikasi Studio Classic.
3. Pilih Luncurkan untuk profil pengguna yang dipilih, lalu pilih TensorBoard.

Opsi 2: Luncurkan TensorBoard dari halaman TensorBoard arahan

Prosedur berikut menjelaskan cara meluncurkan TensorBoard aplikasi dari halaman TensorBoard arahan.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih TensorBoard.
3. Di bawah Memulai, pilih Domain tempat Anda ingin meluncurkan aplikasi Studio Classic. Jika profil pengguna Anda hanya milik satu Domain, Anda tidak melihat opsi untuk memilih Domain.
4. Pilih profil pengguna yang ingin Anda luncurkan aplikasi Studio Classic. Jika tidak ada profil pengguna di Domain, pilih Buat profil pengguna. Untuk informasi selengkapnya, lihat Menambahkan dan Menghapus Profil Pengguna.
5. Pilih Buka TensorBoard.

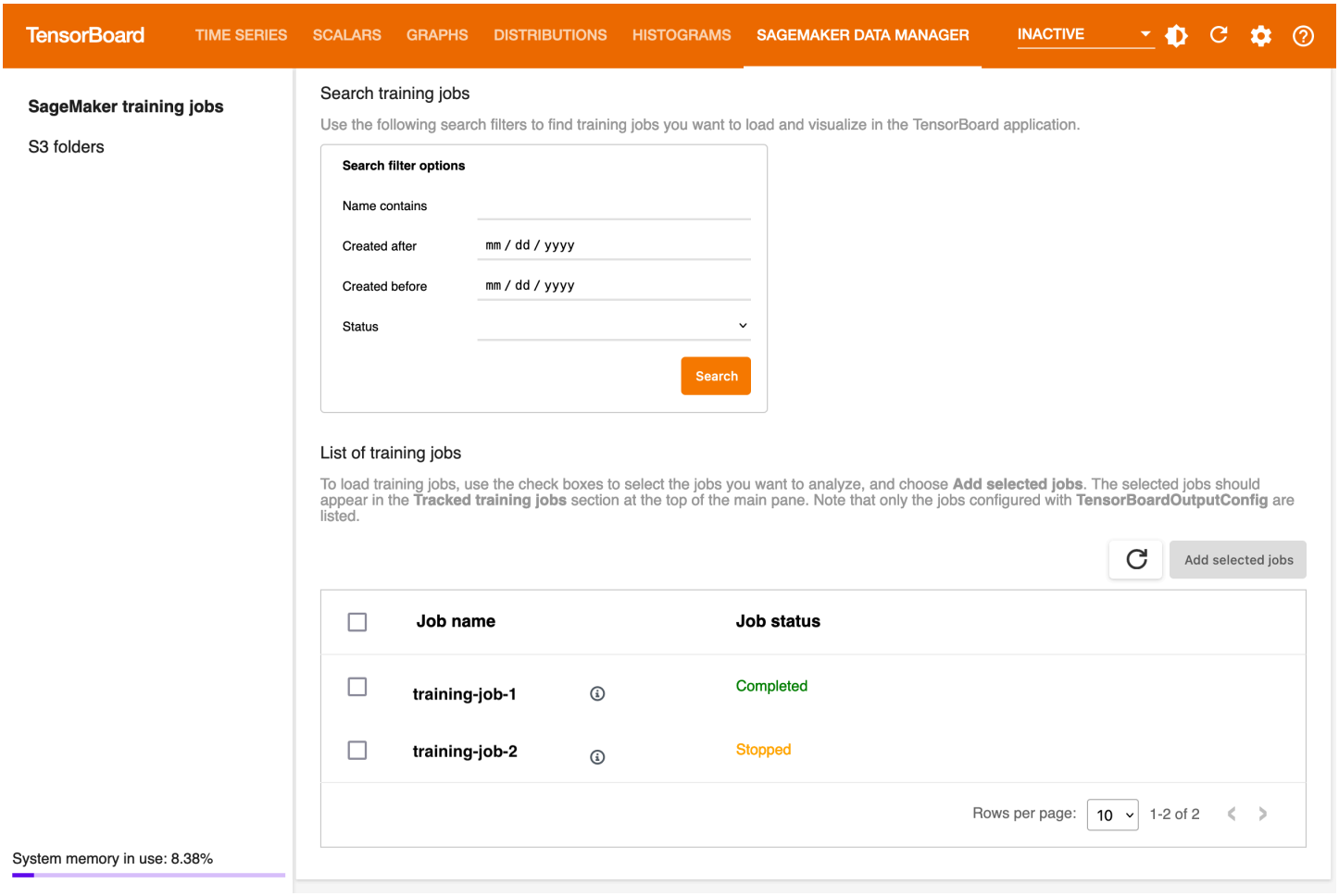
Tangkapan layar berikut menunjukkan lokasi TensorBoard di panel navigasi kiri SageMaker konsol dan SageMaker dengan halaman TensorBoard arahan di panel utama.



Mengakses dan memvisualisasikan data keluaran pelatihan di TensorBoard

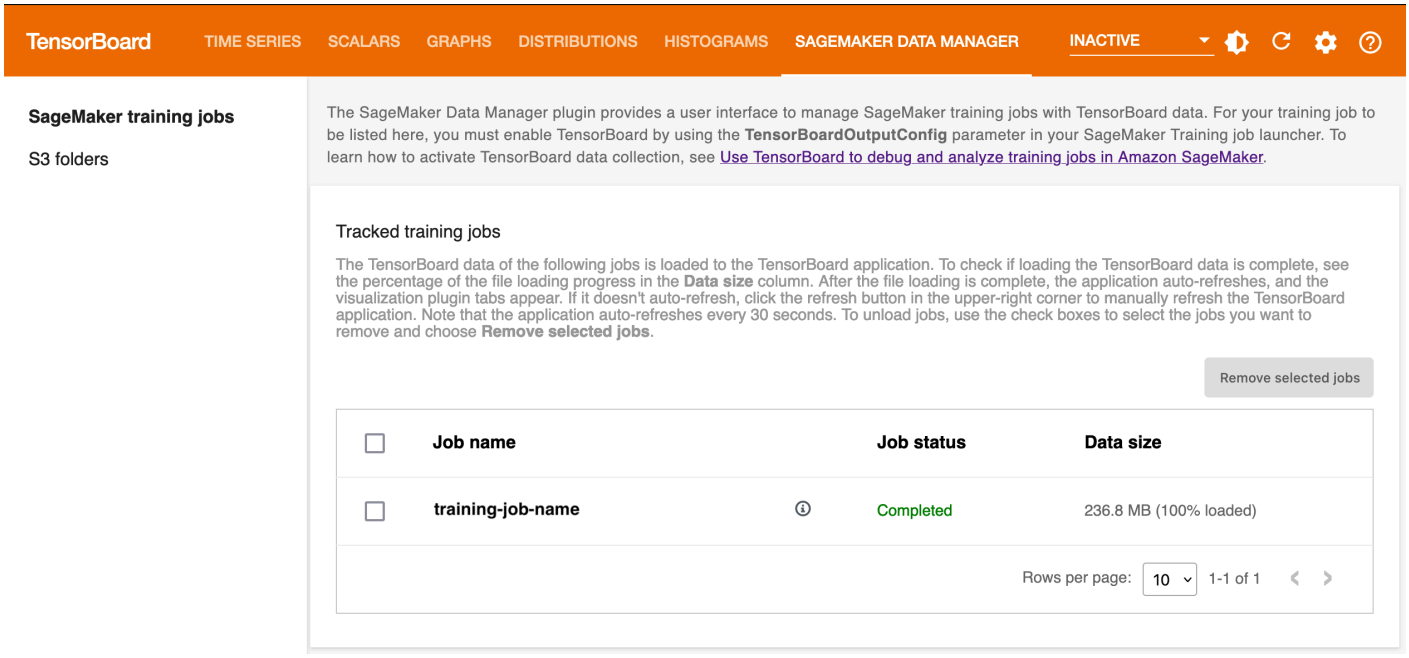
Anda dapat melakukan analisis online atau offline dengan memuat tensor keluaran yang dikumpulkan dari bucket S3 yang dipasangkan dengan pekerjaan pelatihan selama atau setelah pelatihan.

Saat Anda membuka TensorBoard aplikasi, TensorBoard buka dengan tab Manajer SageMaker Data. Tangkapan layar berikut menunjukkan tampilan penuh tab Pengelola SageMaker Data dalam TensorBoard aplikasi.



Di tab Pengelola SageMaker Data, Anda dapat memilih pekerjaan pelatihan dan data keluaran pelatihan TensorBoard yang kompatibel dengan beban dari Amazon S3.

1. Di bagian Cari pekerjaan pelatihan, gunakan filter untuk mempersempit daftar pekerjaan pelatihan yang ingin Anda temukan, muat, dan visualisasikan.
2. Di bagian Daftar pekerjaan pelatihan, gunakan kotak centang untuk memilih pekerjaan pelatihan dari mana Anda ingin menarik data dan memvisualisasikan untuk debugging.
3. Pilih Tambahkan pekerjaan yang dipilih. Pekerjaan yang dipilih akan muncul di bagian Pekerjaan pelatihan yang dilacak, seperti yang ditunjukkan pada gambar berikut.



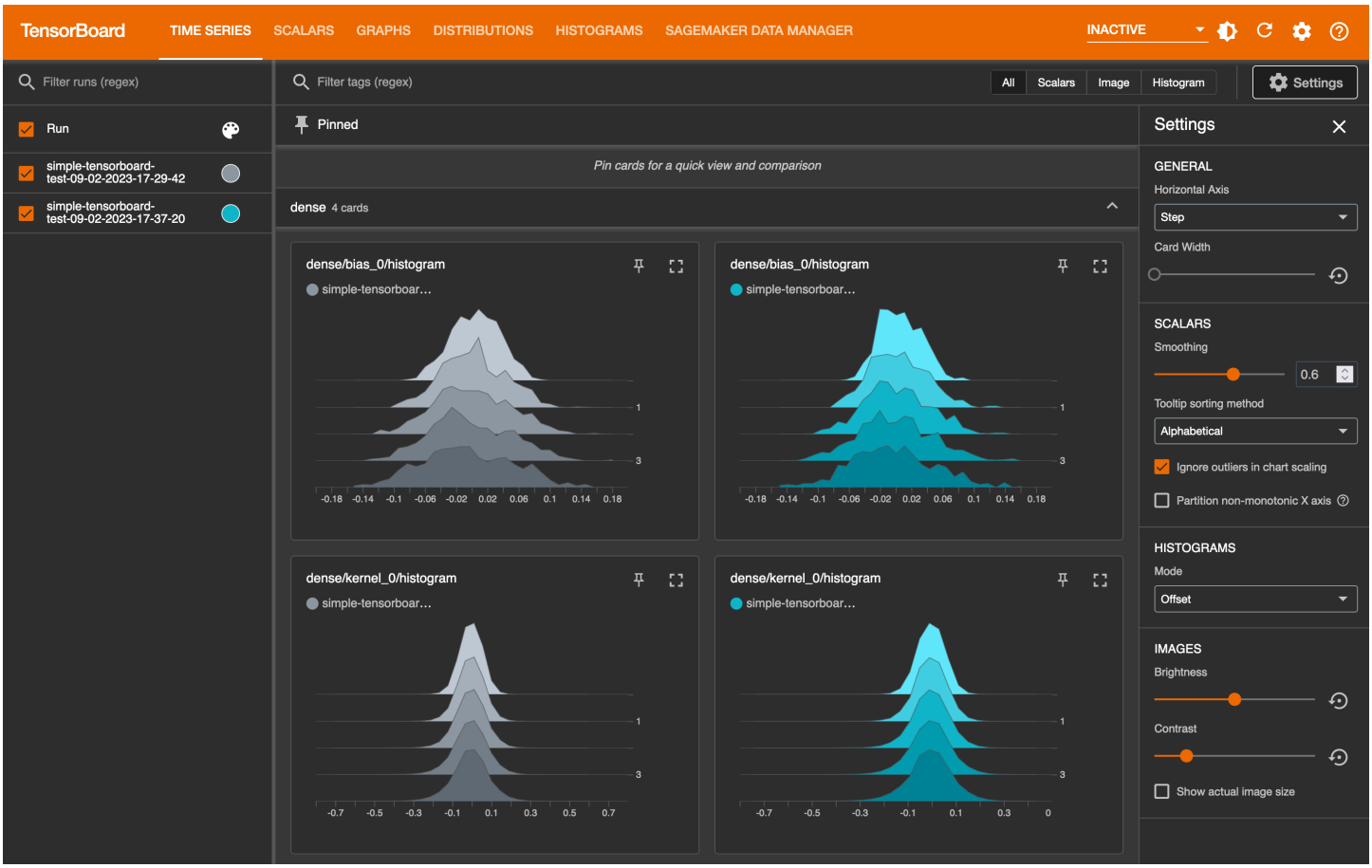
> **Note**
>
> Tab Pengelola SageMaker Data hanya menampilkan pekerjaan pelatihan yang dikonfigurasi dengan `TensorBoardOutputConfig` parameter. Pastikan Anda telah mengonfigurasi SageMaker estimator dengan parameter ini. Untuk informasi selengkapnya, lihat Langkah 2: Bangun peluncur SageMaker pelatihan dengan konfigurasi data TensorBoard .

> **Note**
>
> Tab visualisasi mungkin tidak muncul jika Anda menggunakan SageMaker dengan TensorBoard untuk pertama kalinya atau tidak ada data yang dimuat dari penggunaan sebelumnya. Setelah menambahkan pekerjaan pelatihan dan menunggu beberapa detik, segarkan pemirsa dengan memilih panah melingkar searah jarum jam di sudut kanan atas. Tab visualisasi akan muncul setelah data pekerjaan berhasil dimuat. Anda juga dapat mengatur untuk menyegarkan otomatis menggunakan tombol Pengaturan di sebelah tombol refresh di sudut kanan atas.

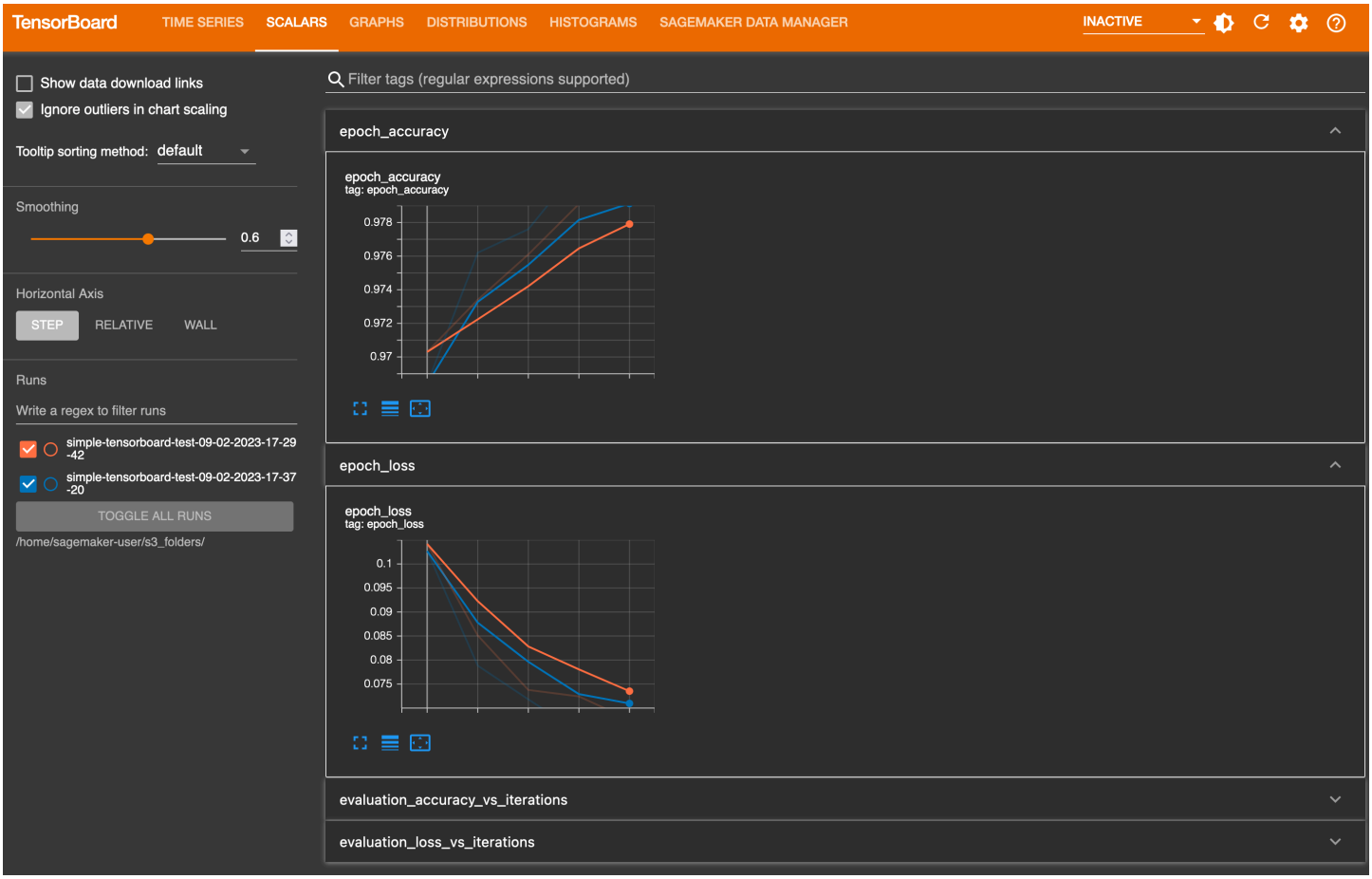
Jelajahi data keluaran pelatihan yang divisualisasikan di TensorBoard

Di tab grafis, Anda dapat melihat daftar pekerjaan pelatihan yang dimuat di panel kiri. Anda juga dapat menggunakan kotak centang pekerjaan pelatihan untuk menampilkan atau menyembunyikan visualisasi. Plugin TensorBoard dinamis diaktifkan secara dinamis tergantung pada bagaimana Anda telah mengatur skrip pelatihan Anda untuk menyertakan penulis ringkasan dan meneruskan panggilan balik untuk koleksi tensor dan skalar, dan oleh karena itu tab grafik juga muncul secara dinamis. Tangkapan layar berikut menunjukkan contoh tampilan setiap tab dengan visualisasi dua pekerjaan pelatihan yang mengumpulkan metrik untuk plugin deret waktu, skalar, grafik, distribusi, dan histogram.

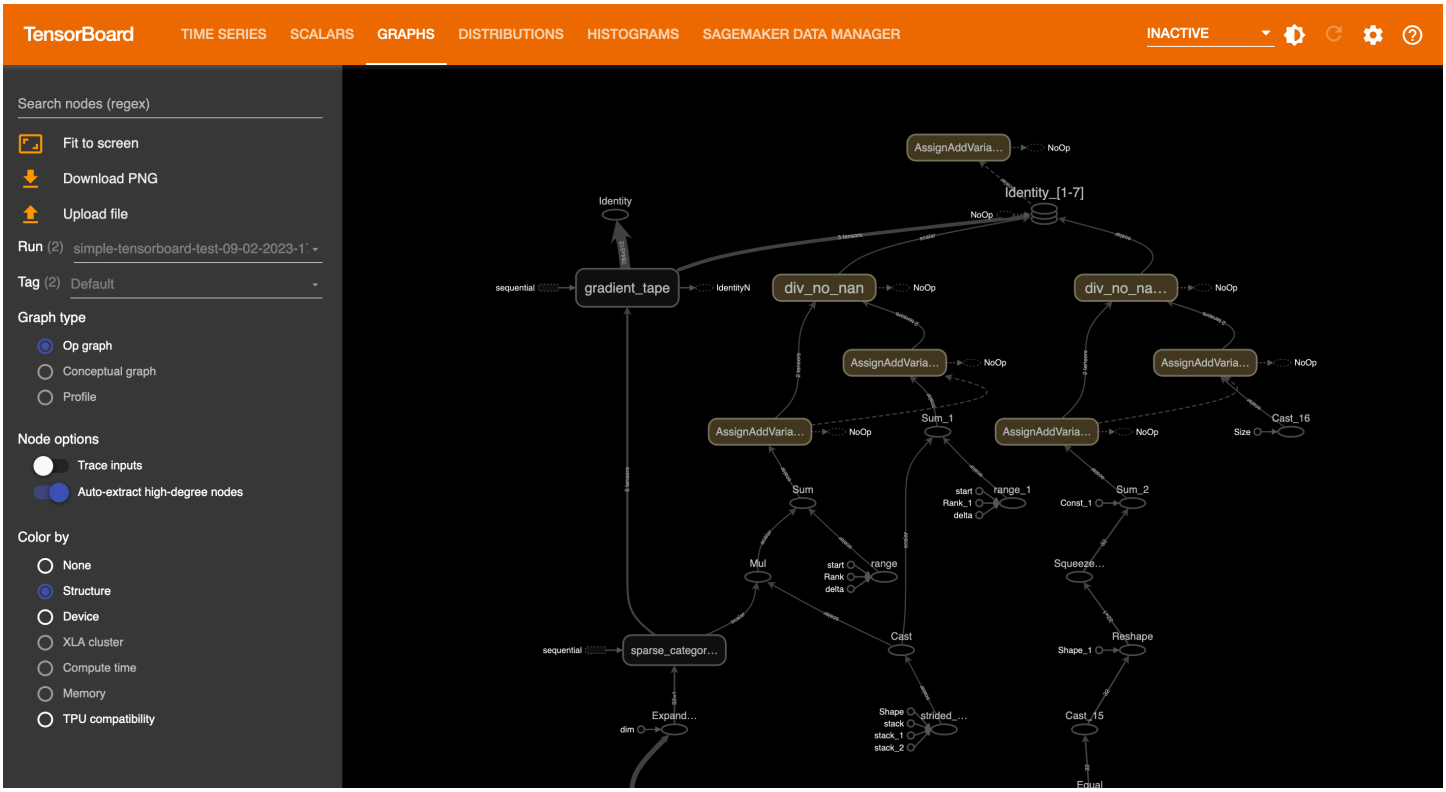
Tampilan tab TIME SERIES



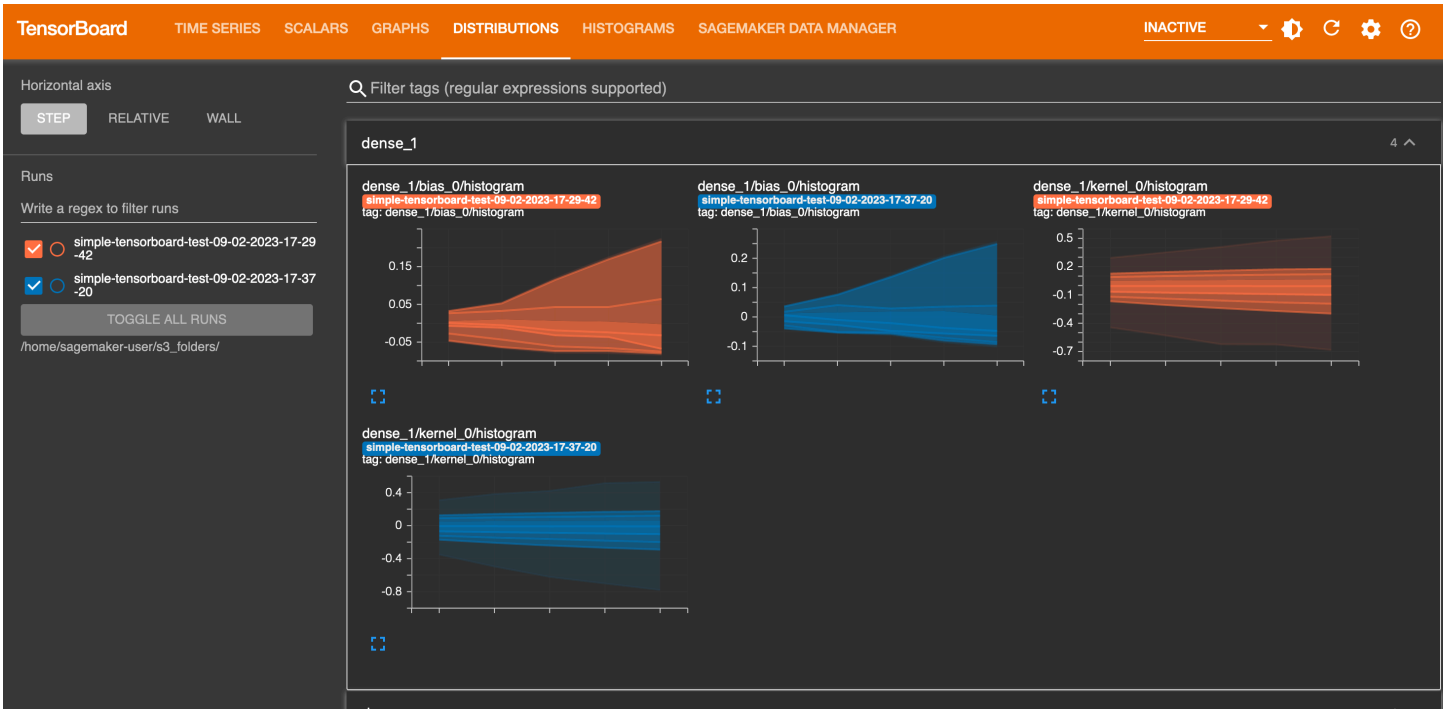
Tampilan tab SCALARS



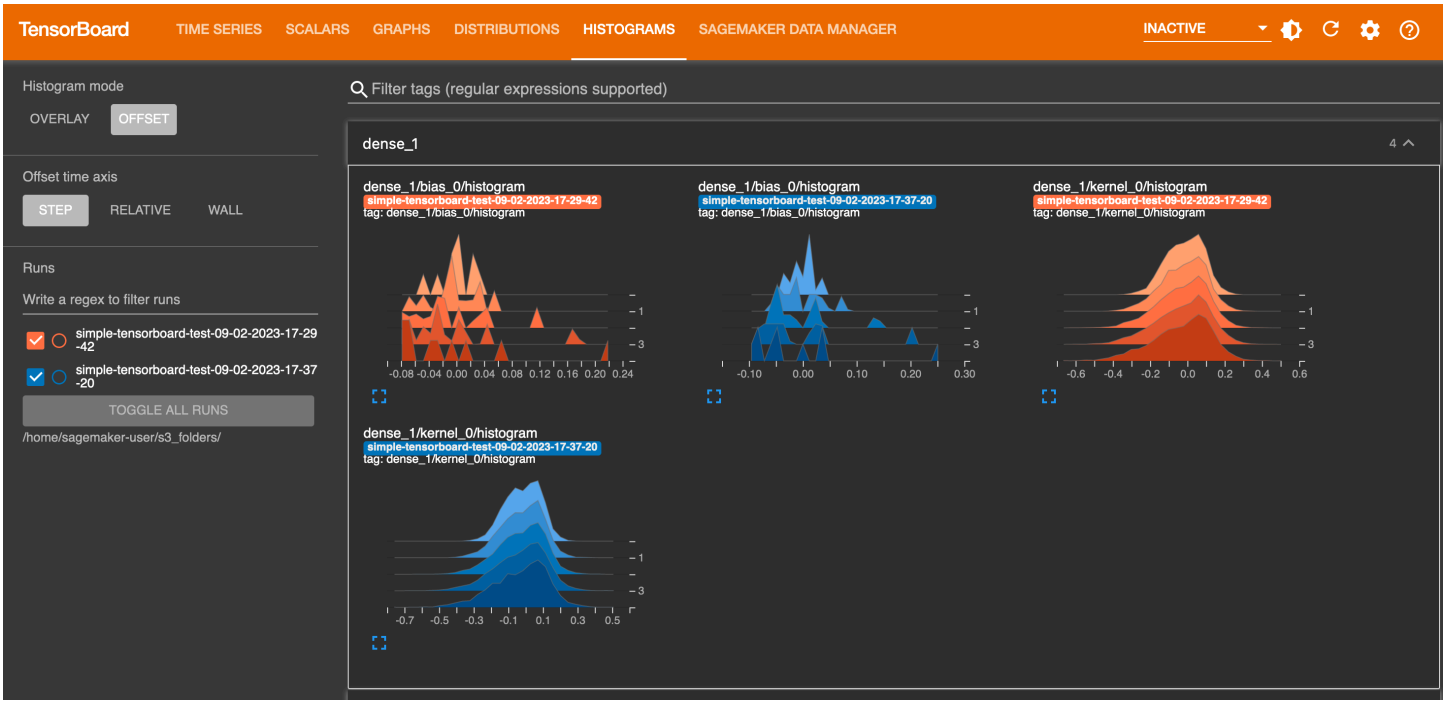
Tampilan tab GRAPHS



Tampilan tab DISTRIBUSI



Tampilan tab HISTOGRAMS



Hapus aplikasi yang tidak digunakan TensorBoard

Setelah Anda selesai memantau dan bereksperimen dengan pekerjaan di TensorBoard, matikan TensorBoard aplikasi.

1. Buka SageMaker konsol.
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih domain Anda.
5. Pilih profil pengguna Anda.
6. Di bawah Aplikasi, pilih Hapus Aplikasi untuk TensorBoard baris.
7. Pilih Ya, hapus aplikasi.
8. Ketik **`delete`** kotak teks, lalu pilih Hapus.
9. Pesan biru akan muncul di bagian atas layar: default sedang dihapus.

Pertimbangan

Pertimbangkan hal berikut saat menggunakan SageMaker dengan TensorBoard.

- Anda tidak dapat berbagi TensorBoard aplikasi untuk tujuan kolaborasi karena SageMaker Domain tidak mengizinkan berbagi aplikasi di antara pengguna. Pengguna dapat membagikan tensor keluaran yang disimpan dalam bucket S3, jika mereka memiliki akses ke bucket.
- Plugin visualisasi mungkin tidak muncul saat Anda pertama kali meluncurkan aplikasi. TensorBoard Setelah Anda memilih pekerjaan pelatihan di plugin SageMaker Data Manager, TensorBoard aplikasi memuat TensorBoard data dan mengisi plugin visualisasi.
- TensorBoard Aplikasi secara otomatis mati setelah 1 jam tidak aktif. Jika Anda ingin mematikan aplikasi ketika Anda selesai menggunakannya, pastikan untuk mematikan secara manual TensorBoard untuk menghindari membayar instans hosting itu. Untuk petunjuk tentang menghapus aplikasi, lihatHapus aplikasi yang tidak digunakan TensorBoard .

Gunakan Amazon SageMaker Debugger untuk men-debug dan meningkatkan kinerja model

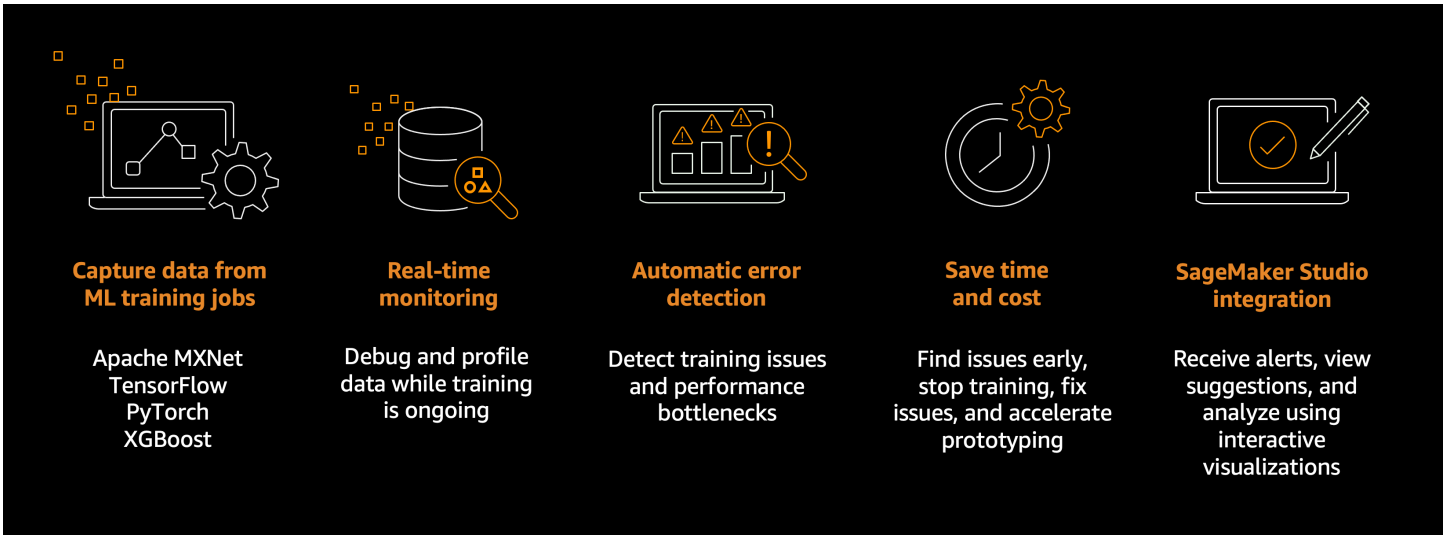
Mendebug tensor keluaran model dari pekerjaan pelatihan pembelajaran mesin secara real time dan mendeteksi masalah non-konvergen menggunakan Amazon SageMaker Debugger.

Amazon SageMaker Fitur Debugger

Pekerjaan pelatihan pembelajaran mesin (ML) dapat memiliki masalah seperti overfitting, fungsi aktivasi jenuh, dan gradien menghilang, yang dapat membahayakan kinerja model.

SageMaker Debugger menyediakan alat untuk men-debug pekerjaan pelatihan dan menyelesaikan masalah tersebut untuk meningkatkan kinerja model Anda. Debugger juga menawarkan alat untuk mengirim peringatan ketika anomali pelatihan ditemukan, mengambil tindakan terhadap masalah, dan mengidentifikasi akar penyebabnya dengan memvisualisasikan metrik dan tensor yang dikumpulkan.

SageMaker Debugger mendukung Apache MxNet, PyTorch, TensorFlow, dan kerangka kerja XGBoost. Untuk informasi selengkapnya tentang kerangka kerja dan versi yang didukung oleh SageMaker Debugger, lihatKerangka Kerja dan Algoritma yang Didukung.



Alur kerja Debugger tingkat tinggi adalah sebagai berikut:

1. Ubah skrip pelatihan Anda dengan`sagemaker-debugger`Python SDK jika diperlukan.
2. Konfigurasikan a SageMaker pekerjaan pelatihan dengan SageMaker Debugger.
   - Konfigurasikan menggunakan SageMaker API Estimator (untuk Python SDK).
   - Konfigurasikan menggunakan SageMaker [`CreateTrainingJob`permintaan (untuk Boto3 atau CLI)](#).
   - Konfigurasikan[kontainer pelatihan khusus](#)bersama SageMaker Debugger.
3. Mulai pekerjaan pelatihan dan pantau masalah pelatihan secara real time.
   - [Daftar Aturan Built-in Debugger](#).
4. Dapatkan peringatan dan ambil tindakan segera terhadap masalah pelatihan.
   - Terima teks dan email dan hentikan pekerjaan pelatihan saat masalah pelatihan ditemukan menggunakan[Tindakan bawaan Debugger untuk Aturan](#).
   - Siapkan tindakan Anda sendiri menggunakan[Amazon CloudWatch Event dan](#)[AWS Lambda](#).
5. Jelajahi analisis mendalam tentang masalah pelatihan.
   - Untuk men-debug tensor keluaran model, lihat[Visualisasikan Tensor Output Debugger di TensorBoard](#).
6. Perbaiki masalah, pertimbangkan saran yang diberikan oleh Debugger, dan ulangi langkah 1-5 hingga Anda mengoptimalkan model dan mencapai akurasi target.

The SageMaker Panduan pengembang debugger memandu Anda melalui topik-topik berikut.

Topik

Kerangka Kerja dan Algoritma yang Didukung

Tabel berikut menunjukkan SageMaker kerangka kerja dan algoritma pembelajaran mesin yang didukung oleh Debugger.

SageMaker-supported frameworks and algorithms	Debugging output tensors
TensorFlow	AWS TensorFlow wadah pembelajaran mendalam1.15.4 atau lebih baru
PyTorch	AWS PyTorch wadah pembelajaran mendalam1.5.0 atau lebih baru
MxNet	AWSWadah pembelajaran mendalam MxNet1.6.0 atau lebih baru
XGBoost	1.0-1, 1.2-1, 1.3-1
SageMaker estimator generik	Kontainer pelatihan khusus(tersedia untuk TensorFlow, PyTorch, MXNet, dan XGBoost dengan registrasi kait manual)

- Debugging tensor keluaran— Lacak dan debug parameter model, seperti bobot, gradien, bias, dan nilai skalar dari pekerjaan pelatihan Anda. Kerangka kerja pembelajaran mendalam yang tersedia adalah Apache MXNet, TensorFlow, PyTorch, dan XGBoost.

> **Important**
>
> Untuk TensorFlow kerangka kerja dengan Keras, SageMaker Debugger menghentikan dukungan perubahan kode nol untuk model debugging yang dibuat menggunakan`tf.keras`modul TensorFlow 2.6 dan kemudian. Ini karena perubahan melanggar yang diumumkan diTensorFlow Catatan rilis 2.6.0. Untuk petunjuk tentang cara memperbarui skrip pelatihan, lihatthe section called “TensorFlow”.

> **Important**
>
> Dari PyTorch v1.12.0 dan setelahnya, SageMaker Debugger menghentikan dukungan perubahan kode nol untuk model debugging.
> Ini karena melanggar perubahan yang menyebabkan SageMaker Debugger untuk mengganggu`torch.jit`fungsionalitas. Untuk petunjuk tentang cara memperbarui skrip pelatihan, lihatthe section called “PyTorch”.

Jika kerangka kerja atau algoritma yang ingin Anda latih dan debug tidak tercantum dalam tabel, bukaAWSForum Diskusidan meninggalkan umpan balik SageMaker Debugger.

Wilayah AWS

Amazon SageMaker Debugger tersedia di semua wilayah di mana Amazon SageMaker dalam pelayanan kecuali wilayah berikut.

- Asia Pasifik (Jakarta):`ap-southeast-3`

Untuk mengetahui apakah Amazon SageMaker Berada dalam pelayanan diWilayah AWS, lihatAWSLayanan Regional.

Gunakan Debugger dengan Wadah Pelatihan Kustom

Bawa wadah pelatihan Anda ke SageMaker dan dapatkan wawasan tentang pekerjaan pelatihan Anda menggunakan Debugger. Maksimalkan efisiensi kerja Anda dengan mengoptimalkan model Anda di instans Amazon EC2 menggunakan fitur pemantauan dan debugging.

Untuk informasi lebih lanjut tentang cara membangun kontainer pelatihan Anda dengan`sagemaker-debugger`pustaka klien, dorong ke Amazon Elastic Container Registry (Amazon ECR), dan pantau serta debug, lihatGunakan Debugger dengan Custom Training Containers.

Debugger Sumber Terbuka GitHub Repositori

API debugger disediakan melalui SageMaker Python SDK dan dirancang untuk membangun hook Debugger dan konfigurasi aturan untuk SageMaker CreateTrainingJobdan DescribeTrainingJobOperasi API. The`sagemaker-debugger`pustaka klien menyediakan alat untuk mendaftarkaitdan mengakses data pelatihan melaluipersidanganfitur, semua melalui operasi API yang fleksibel dan kuat. Ini mendukung kerangka pembelajaran mesin TensorFlow, PyTorch, MXNet, dan XGBoost pada Python 3.6 dan yang lebih baru.

Untuk sumber daya langsung tentang Debugger dan`sagemaker-debugger`Operasi API, lihat tautan berikut ini:

- Amazon SageMaker Dokumentasi Python SDK
- Amazon SageMaker Python SDK - API Debugger
- The`sagemaker-debugger`Dokumentasi Python SDKuntukAmazon SageMaker Pustaka klien sumber terbuka debugger
- The`sagemaker-debugger`PyPI

Jika Anda menggunakan SDK for Java untuk melakukan SageMaker melatih pekerjaan dan ingin mengonfigurasi API Debugger, lihat referensi berikut:

- Amazon SageMaker Operasi API debugger
- Mengonfigurasi Debugger Menggunakan Amazon SageMaker API

Amazon SageMaker Arsitektur Debugger

Topik ini memandu Anda melalui ikhtisar tingkat tinggi tentang Amazon. SageMaker Alur kerja debugger.

Debugger mendukung fungsionalitas profil untukoptimasi kinerjauntuk mengidentifikasi masalah komputasi, seperti kemacetan sistem dan kurangnya pemanfaatan, dan untuk membantu mengoptimalkan pemanfaatan sumber daya perangkat keras dalam skala besar.

Fungsi debugging debugger untukoptimasi modeladalah tentang menganalisis masalah pelatihan non-konvergen yang dapat muncul sambil meminimalkan fungsi kerugian menggunakan algoritma pengoptimalan, seperti penurunan gradien dan variasinya.

Diagram berikut menunjukkan arsitektur SageMaker Debugger. Blok dengan garis batas tebal adalah apa yang Debugger kelola untuk menganalisis pekerjaan pelatihan Anda.

Debugger menyimpan data berikut dari pekerjaan pelatihan Anda di bucket Amazon S3 Anda yang aman:

- Tensor keluaran— Koleksi skalar dan parameter model yang terus diperbarui selama lintasan maju dan mundur saat melatih model ML. Tensor keluaran mencakup nilai skalar (akurasi dan kerugian) dan matriks (bobot, gradien, lapisan input, dan lapisan keluaran).

> **Note**
>
> Secara default, Debugger memantau dan men-debug SageMaker pekerjaan pelatihan tanpa parameter khusus Debugger yang dikonfigurasi di SageMaker penaksir. Debugger mengumpulkan metrik sistem setiap 500 milidetik dan tensor keluaran dasar (output skalar seperti kehilangan dan akurasi) setiap 500 langkah. Ini juga menjalankan`ProfilerReport`aturan untuk menganalisis metrik sistem dan menggabungkan dasbor wawasan Studio Debugger dan laporan pembuatan profil. Debugger menyimpan data keluaran di bucket Amazon S3 Anda.

Aturan bawaan Debugger dijalankan pada container pemrosesan, yang dirancang untuk mengevaluasi model pembelajaran mesin dengan memproses data pelatihan yang dikumpulkan di bucket S3 Anda (lihatMemproses Data dan Evaluasi Model). Aturan bawaan sepenuhnya dikelola oleh Debugger. Anda juga dapat membuat aturan sendiri yang disesuaikan dengan model Anda untuk melihat masalah apa pun yang ingin Anda pantau.

Memulai Tutorial Debugger

Topik-topik berikut memandu Anda melalui tutorial dari dasar-dasar untuk kasus penggunaan lanjutan pemantauan, profil, dan debugging pekerjaan pelatihan SageMaker menggunakan Debugger. Jelajahi fitur Debugger dan pelajari bagaimana Anda dapat men-debug dan meningkatkan model pembelajaran mesin Anda secara efisien dengan menggunakan Debugger.

Topik

- Video Tutorial Debugger
- Notebook Debugger
- Debugger Demo Lanjutan dan Visualisasi

Video Tutorial Debugger

Video berikut menyediakan tur kemampuan Amazon SageMaker Debugger menggunakan instans SageMaker Studio dan SageMaker notebook.

Topik

- Debug Model dengan Amazon SageMaker Debugger di Studio
- Deep Dive di Amazon SageMaker Debugger dan Model Monitor SageMaker

Debug Model dengan Amazon SageMaker Debugger di Studio

Julien Simon, Penginjil AWS Teknis | Durasi: 14 menit 17 detik

Video tutorial ini menunjukkan cara menggunakan Amazon SageMaker Debugger untuk menangkap dan memeriksa informasi debugging dari model pelatihan. Contoh model pelatihan yang digunakan dalam video ini adalah jaringan saraf konvolusional sederhana (CNN) berdasarkan Keras dengan backend. TensorFlow SageMaker dalam TensorFlow kerangka kerja dan Debugger memungkinkan Anda membangun estimator secara langsung menggunakan skrip pelatihan dan men-debug pekerjaan pelatihan.

Debug Model dengan Amazon SageMaker Debugger (bagian 1)

Anda dapat menemukan contoh buku catatan dalam video di repositori Demo Studio ini yang disediakan oleh penulis. Anda perlu mengkloning file `debugger.ipynb` notebook dan skrip `mnist_keras_tf.py` pelatihan ke SageMaker Studio atau instance SageMaker notebook Anda. Setelah Anda mengkloning dua file, tentukan path `keras_script_path` ke `mnist_keras_tf.py` file di dalam `debugger.ipynb` notebook. Misalnya, jika Anda mengkloning dua file dalam direktori yang sama, atur sebagai`keras_script_path = "mnist_keras_tf.py"`.

Deep Dive di Amazon SageMaker Debugger dan Model Monitor SageMaker

Julien Simon, Penginjil AWS Teknis | Durasi: 44 menit 34 detik

Sesi video ini mengeksplorasi fitur-fitur canggih Debugger dan SageMaker Model Monitor yang membantu meningkatkan produktivitas dan kualitas model Anda. Pertama, video ini menunjukkan cara mendeteksi dan memperbaiki masalah pelatihan, memvisualisasikan tensor, dan meningkatkan model dengan Debugger. Selanjutnya, pada 22:41, video menunjukkan cara memantau model dalam produksi dan mengidentifikasi masalah prediksi seperti fitur yang hilang atau penyimpangan data menggunakan Model Monitor. SageMaker Akhirnya, ia menawarkan kiat pengoptimalan biaya untuk membantu Anda memaksimalkan anggaran pembelajaran mesin Anda.

Debug Model dengan Debugger (bagian 2)

Anda dapat menemukan contoh buku catatan dalam video di repositori AWS Dev Days 2020 yang ditawarkan oleh penulis.

Notebook Debugger

SageMaker Debugger contoh notebookdisediakan dalamaws/amazon-sagemaker-contohrepositori. Notebook contoh Debugger memandu Anda melalui kasus penggunaan dasar hingga lanjutan dari debugging dan profiling pekerjaan pelatihan.

Sebaiknya Anda menjalankan contoh notebook di SageMaker Studio atau instance SageMaker Notebook karena sebagian besar contoh dirancang untuk pekerjaan pelatihan di ekosistem SageMaker, termasuk Amazon EC2, Amazon S3, dan Amazon SageMaker Python SDK.

Untuk mengkloning contoh repositori ke SageMaker Studio, ikuti petunjuk diTur Studio Amazon SageMaker.

Untuk menemukan contoh dalam contoh Notebook SageMaker, ikuti instruksinya diNotebook Instans Notebook SageMaker.

Important

Untuk menggunakan fitur Debugger baru, Anda perlu meng-upgrade SDK SageMaker Python dan`SMDebug`perpustakaan klien. Dalam kernel iPython Anda, Jupyter Notebook, atau lingkungan JupyterLab, jalankan kode berikut untuk menginstal versi terbaru dari perpustakaan dan restart kernel.

```
import sys
import IPython
!{sys.executable} -m pip install -U sagemaker smdebug
IPython.Application.instance().kernel.do_shutdown(True)
```

Debugger Contoh Notebook untuk Profiling Training Jobs

Daftar berikut menunjukkan Debugger contoh notebook memperkenalkan adaptasi Debugger untuk memantau dan profil pelatihan pekerjaan untuk berbagai model pembelajaran mesin, dataset, dan kerangka kerja.

Judul Notebook	Kerangka Kerja	Model	Set data	Deskripsi
Analisis Data Pembuatan	TensorFlow	Keras Resnet50	Cifar-10	Notebook ini memberikan pengantar analisis interakti

Judul Notebook	Kerangka Kerja	Model	Set data	Deskripsi
Profil Debugger Amazon SageMaker				f data profil yang ditangkap oleh SageMaker Debugger. Jelajahi fungsionalitas lengkapSMDebugalat analisis interaktif.
Pelatihan pembelajaran mesin profil dengan Debugger Amazon SageMaker	TensorFlow	1-D Jaringan Saraf Konvolusional	Set data IMDB	Profil TensorFlow 1-D CNN untuk analisis sentimen data IMDB yang terdiri dari ulasan film yang diberi label memiliki sentimen positif atau negatif. Jelajahi wawasan Studio Debugger dan laporan profil Debugger.
Memprofilkan pelatihan model TensorFlow Resnet dengan berbagai pengaturan pelatihan terdistribusi	TensorFlow	Resnet50	Cifar-10	Jalankan pekerjaan pelatihan TensorFlow dengan berbagai pengaturan pelatihan terdistribusi, memantau pemanfaatan sumber daya sistem, dan kinerja model profil menggunakan Debugger.

Judul Notebook	Kerangka Kerja	Model	Set data	Deskripsi
Memprofilkan pelatihan model PyTorch Resnet dengan berbagai pengaturan pelatihan terdistribusi	PyTorch	Resnet50	Cifar-10	Jalankan pekerjaan pelatihan PyTorch dengan berbagai pengaturan pelatihan terdistribusi, memantau pemanfaatan sumber daya sistem, dan kinerja model profil menggunakan Debugger.

Debugger Contoh Notebook untuk Menganalisis Parameter Model

Daftar berikut menunjukkan contoh Debugger notebook memperkenalkan adaptasi Debugger untuk men-debug pekerjaan pelatihan untuk berbagai model pembelajaran mesin, dataset, dan kerangka kerja.

Judul Notebook	Kerangka Kerja	Model	Set data	Deskripsi
Debugger Amazon SageMaker - Gunakan aturan bawaan	TensorFlow	Jaringan Neural Convolutional	MNIST	Gunakan aturan bawaan Debugger Amazon SageMaker untuk men-debug model TensorFlow.
Debugger Amazon SageMaker - Tensorflow 2.1	TensorFlow	Resnet50	Cifar-10	Gunakan konfigurasi hook Debugger Amazon SageMaker dan aturan bawaan untuk men-debug model dengan framework Tensorflow 2.1.

Judul Notebook	Kerangka Kerja	Model	Set data	Deskripsi
Memvisualisasikan Tensor Debugging pelatihan MXNet	MXNet	Gluon Jaringan saraf konvolusional	Mode MNIST	Jalankan pekerjaan pelatihan dan konfigurasikan SageMaker Debugger untuk menyimpan semua tensor dari pekerjaan ini, lalu visualisasikan tensor itu di notebook.
Aktifkan Pelatihan Spot dengan Debugger Amazon SageMaker	MXNet	Gluon Jaringan saraf konvolusional	Mode MNIST	Pelajari cara Debugger mengumpulkan data tensor dari pekerjaan pelatihan pada instance spot, dan cara menggunakan aturan bawaan Debugger dengan pelatihan spot terkelola.
Jelaskan model XGBoost yang memprediksi pendapatan individu dengan Amazon SageMaker Debugger	XGBoost	Regresi XGboost	Set data Sensus Dewasa	Pelajari cara menggunakan hook Debugger dan aturan bawaan untuk mengumpulkan dan memvisualisasikan data tensor dari model regresi XGBoost, seperti nilai kerugian, fitur, dan nilai SHAP.

Untuk menemukan visualisasi lanjutan parameter model dan kasus penggunaan, lihat topik berikutnya diDebugger Demo Lanjutan dan Visualisasi.

Debugger Demo Lanjutan dan Visualisasi

Demo berikut memandu Anda melalui kasus penggunaan lanjutan dan skrip visualisasi menggunakan Debugger.

Topik

- Latih dan Tune Model Anda dengan Eksperimen dan Debugger Amazon SageMaker
- Menggunakan SageMaker Debugger untuk Memantau Pelatihan Model Autoencoder Convolutional
- Menggunakan SageMaker Debugger untuk Memantau Perhatian dalam Pelatihan Model BERT
- Menggunakan Debugger SageMaker untuk Memvisualisasikan Peta Aktivasi Kelas di Jaringan Neural Convolutional (CNNs)

Latih dan Tune Model Anda dengan Eksperimen dan Debugger Amazon SageMaker

Dr. Nathalie Rauschmayr,AWSApplied Scientist | Panjang: 49 menit 26 detik

Melatih dan Model Prune dengan SageMaker Eksperimen dan Debugger

Cari tahu bagaimana Amazon SageMaker Experiments and Debugger dapat menyederhanakan pengelolaan pekerjaan pelatihan Anda. Debugger Amazon SageMaker memberikan visibilitas transparan ke dalam pekerjaan pelatihan dan menghemat metrik pelatihan ke dalam bucket Amazon S3 Anda. SageMaker Experiments memungkinkan Anda untuk memanggil informasi pelatihan sebagaiuji cobamelalui SageMaker Studio dan mendukung visualisasi pekerjaan pelatihan. Ini membantu Anda menjaga kualitas model tetap tinggi sekaligus mengurangi parameter yang kurang penting berdasarkan peringkat penting.

Video ini menunjukkanpemangkasanteknik yang membuat model Resnet50 dan AlexNet pra-terlatih lebih ringan dan terjangkau sambil menjaga standar yang tinggi untuk akurasi model.

SageMaker Estimator melatih algoritma yang dipasok dari kebun binatang model PyTorch di sebuahAWSDeep Learning Containers dengan kerangka PyTorch, dan Debugger mengekstrak metrik pelatihan dari proses pelatihan.

Video ini juga menunjukkan cara mengatur aturan kustom Debugger untuk menonton keakuratan model yang dipangkas, untuk memicu peristiwa Amazon CloudWatch danAWS Lambdaberfungsi ketika akurasi menyentuh ambang batas, dan untuk secara otomatis menghentikan proses pemangkasan untuk menghindari iterasi berlebihan.

Tujuan pembelajaran adalah sebagai berikut:

- Pelajari cara menggunakan SageMaker untuk mempercepat pelatihan model ML dan meningkatkan kualitas model.
- Pahami cara mengelola iterasi pelatihan dengan SageMaker Experiments dengan secara otomatis menangkap parameter input, konfigurasi, dan hasil.

- Temukan bagaimana Debugger membuat proses pelatihan transparan dengan secara otomatis menangkap data tensor real-time dari metrik seperti bobot, gradien, dan output aktivasi jaringan saraf konvolusional.
- Gunakan CloudWatch untuk memicu Lambda saat Debugger menangkap masalah.
- Kuasai proses pelatihan SageMaker menggunakan SageMaker Experiments and Debugger.

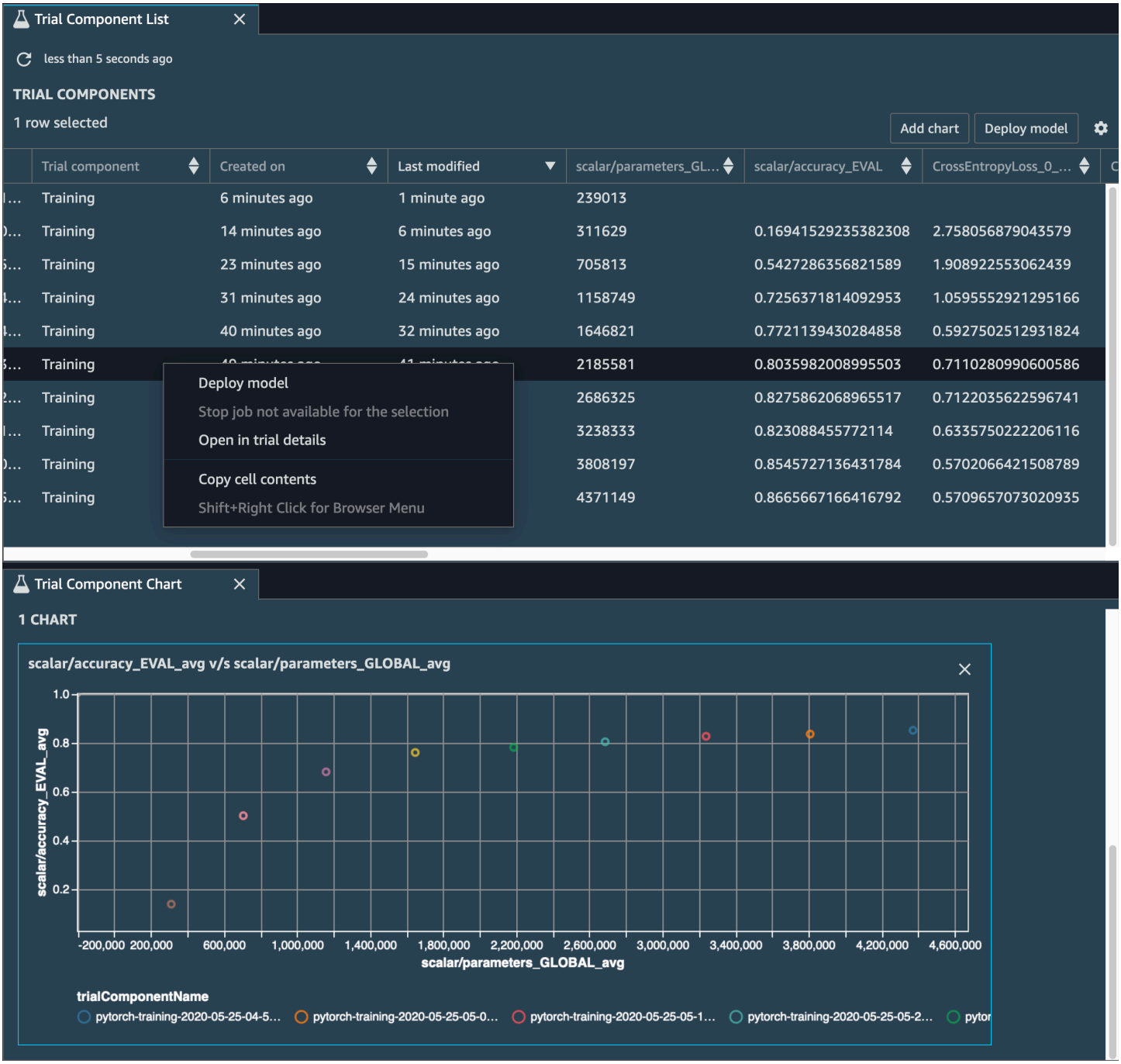
Anda dapat menemukan notebook dan skrip pelatihan yang digunakan dalam video ini dariSageMaker Debugger PyTorch Model Iteratif Pemangkasan.

Gambar berikut menunjukkan bagaimana proses pemangkasan model berulang mengurangi ukuran AlexNet dengan memotong 100 filter paling signifikan berdasarkan peringkat pentingnya yang dievaluasi oleh output aktivasi dan gradien.

Proses pemangkasan mengurangi 50 juta parameter awal menjadi 18 juta. Hal ini juga mengurangi perkiraan ukuran model dari 201 MB menjadi 73 MB.

```
Pruning iteration: 0
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 58, 55, 55]          21,112
              ReLU-2          [-1, 58, 55, 55]               0
         MaxPool2d-3          [-1, 58, 27, 27]               0
            Conv2d-4         [-1, 166, 27, 27]         240,866
              ReLU-5         [-1, 166, 27, 27]               0
         MaxPool2d-6         [-1, 166, 13, 13]               0
            Conv2d-7         [-1, 305, 13, 13]         455,975
              ReLU-8         [-1, 305, 13, 13]               0
            Conv2d-9         [-1, 206, 13, 13]         565,676
             ReLU-10         [-1, 206, 13, 13]               0
           Conv2d-11         [-1, 217, 13, 13]         402,535
             ReLU-12         [-1, 217, 13, 13]               0
        MaxPool2d-13           [-1, 217, 6, 6]               0
AdaptiveAvgPool2d-14           [-1, 217, 6, 6]               0
          Dropout-15                [-1, 7812]               0
           Linear-16                [-1, 4096]      32,002,048
             ReLU-17                [-1, 4096]               0
          Dropout-18                [-1, 4096]               0
           Linear-19                [-1, 4096]      16,781,312
             ReLU-20                [-1, 4096]               0
           Linear-21                 [-1, 101]         413,797
================================================================
Total params: 50,883,321
Trainable params: 50,883,321
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 7.27
Params size (MB): 194.10
Estimated Total Size (MB): 201.95
----------------------------------------------------------------
```

Anda juga perlu melacak akurasi model, dan gambar berikut menunjukkan bagaimana Anda dapat merencanakan proses pemangkasan model untuk memvisualisasikan perubahan akurasi model berdasarkan jumlah parameter di SageMaker Studio.



Di SageMaker Studio, pilihEksperimentab, pilih daftar tensor yang disimpan oleh Debugger dari proses pemangkasan, dan kemudian menulisDaftar Komponen Percobaanpanel. Pilih semua sepuluh iterasi dan kemudian pilihTambahkan baganuntuk membuatBagan Komponen Percobaan. Setelah Anda memutuskan model untuk disebarkan, pilih komponen percobaan dan pilih menu untuk melakukan tindakan atau memilihModel Deploy.

> **Note**
>
> Untuk menyebarkan model melalui SageMaker Studio menggunakan contoh notebook berikut, tambahkan baris di akhir`train`fungsi di`train.py`naskah

```
# In the train.py script, look for the train function in line 58.
def train(epochs, batch_size, learning_rate):
    ...
        print('acc:{:.4f}'.format(correct/total))
        hook.save_scalar("accuracy", correct/total, sm_metric=True)

    # Add the following code to line 128 of the train.py script to save the
 pruned models
    # under the current SageMaker Studio model directory
    torch.save(model.state_dict(), os.environ['SM_MODEL_DIR'] + '/model.pt')
```

Menggunakan SageMaker Debugger untuk Memantau Pelatihan Model Autoencoder Convolutional

Notebook ini menunjukkan bagaimana SageMaker Debugger memvisualisasikan tensor dari proses pembelajaran tanpa pengawasan (atau self-supervised) pada dataset gambar MNIST dari angka tulisan tangan.

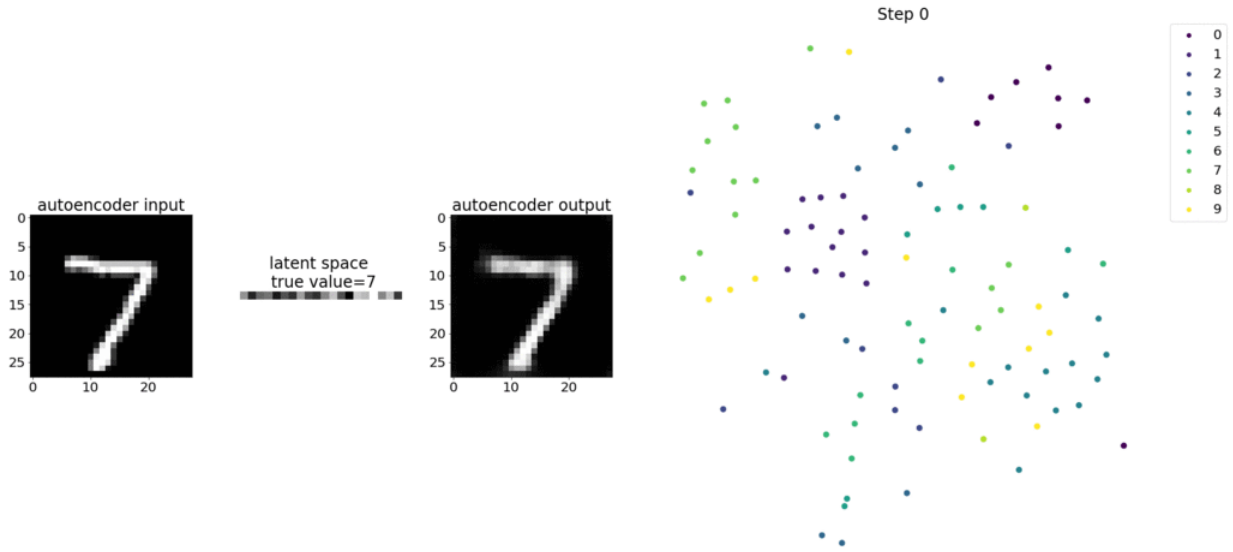
Model pelatihan dalam notebook ini adalah autoencoder convolutional dengan framework MXNet. Autoencoder convolutional memiliki jaringan saraf convolutional berbentuk kemacetan yang terdiri dari bagian encoder dan bagian decoder.

Encoder dalam contoh ini memiliki dua lapisan konvolusi untuk menghasilkan representasi terkompresi (variabel laten) dari gambar input. Dalam hal ini, encoder menghasilkan variabel laten ukuran (1, 20) dari gambar input asli ukuran (28, 28) dan secara signifikan mengurangi ukuran data untuk pelatihan sebanyak 40 kali.

Decoder memiliki duadekonvolusionallapisan dan memastikan bahwa variabel laten melestarikan informasi kunci dengan merekonstruksi gambar output.

Encoder convolutional memberi kekuatan algoritma pengelompokan dengan ukuran data input yang lebih kecil dan kinerja algoritma pengelompokan seperti k-means, K-nn, dan T-didistribusikan Stochastic Neighbor Embedding (T-SNE).

Contoh notebook ini menunjukkan bagaimana memvisualisasikan variabel laten menggunakan Debugger, seperti yang ditunjukkan dalam animasi berikut. Hal ini juga menunjukkan bagaimana algoritma T-SNE mengklasifikasikan variabel laten menjadi sepuluh cluster dan proyek mereka menjadi ruang dua dimensi. Skema warna plot pencar di sisi kanan gambar mencerminkan nilai sebenarnya untuk menunjukkan seberapa baik model BERT dan algoritma T-SNE mengatur variabel laten ke dalam cluster.



Menggunakan SageMaker Debugger untuk Memantau Perhatian dalam Pelatihan Model BERT

Dua arah Encode Representasi dari Transformers (BERT) adalah model representasi bahasa. Seperti nama model mencerminkan, model BERT dibangun di ataspembelajaran transferdanModel transformatoruntuk pemrosesan bahasa alami (NLP).

Model BERT adalah pra-dilatih pada tugas-tugas tanpa pengawasan seperti memprediksi kata-kata yang hilang dalam kalimat atau memprediksi kalimat berikutnya yang secara alami mengikuti kalimat sebelumnya. Data pelatihan berisi 3,3 miliar kata (token) teks bahasa Inggris, dari sumber seperti Wikipedia dan buku elektronik. Untuk contoh sederhana, model BERT dapat memberikanperhatianuntuk token kata kerja yang sesuai atau token kata ganti dari token subjek.

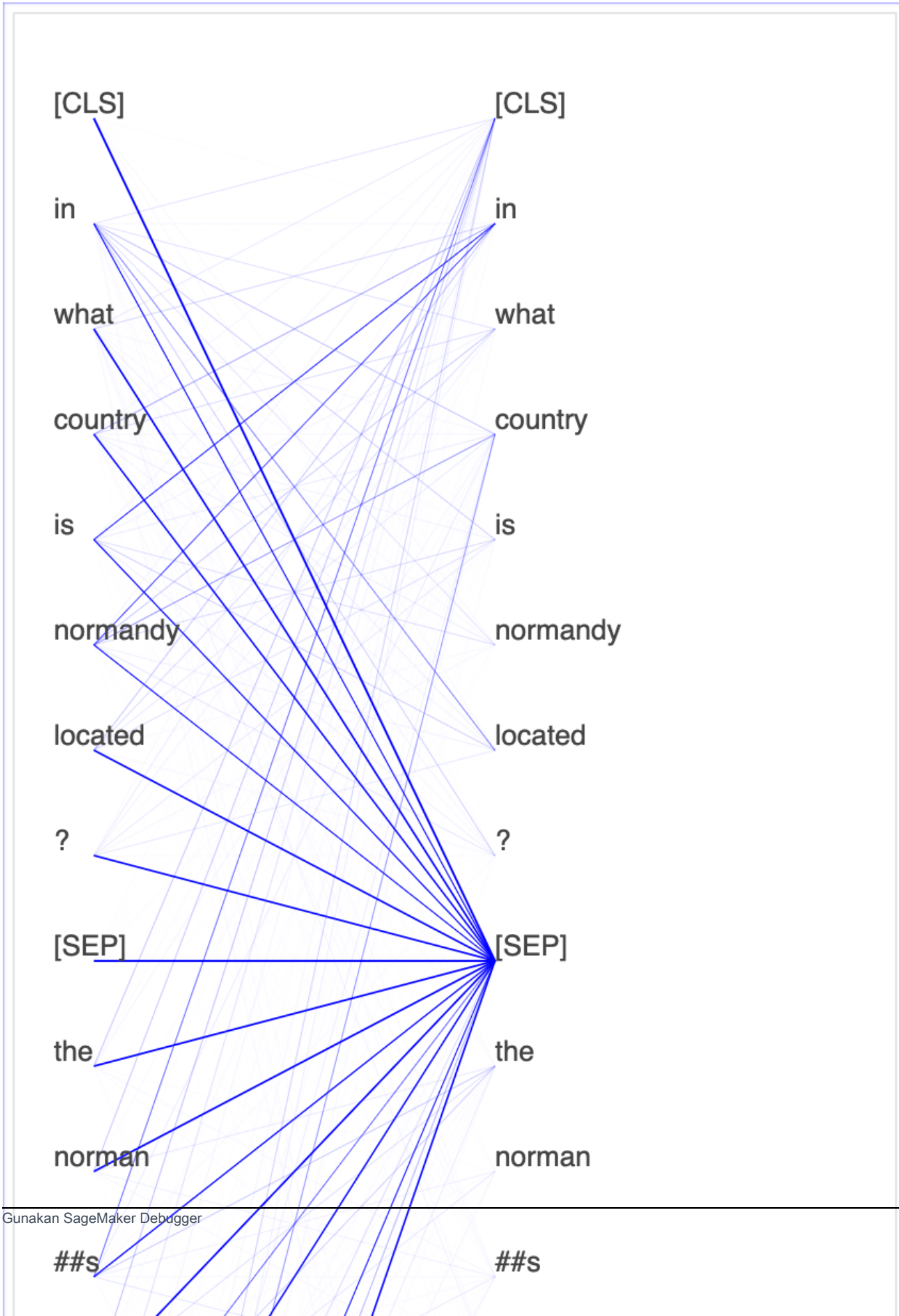
Model BERT pra-terlatih dapat disesuaikan dengan lapisan output tambahan untuk mencapai pelatihan model mutakhir dalam tugas NLP, seperti tanggapan otomatis terhadap pertanyaan, klasifikasi teks, dan banyak lainnya.

Debugger mengumpulkan tensor dari proses fine-tuning. Dalam konteks NLP, berat neuron disebutperhatian.

Notebook ini menunjukkan bagaimana menggunakanmodel BERT pra-terlatih dari kebun binatang model GluonNLPpada dataset Pertanyaan dan Menjawab Stanford dan cara mengatur SageMaker Debugger untuk memantau pekerjaan pelatihan.

Plottingskor perhatiandan neuron individu dalam query dan vektor kunci dapat membantu untuk mengidentifikasi penyebab prediksi model yang salah. Dengan SageMaker Debugger, Anda dapat mengambil tensor dan merencanakantampilan perhatian-kepalasecara real time sebagai pelatihan berlangsung dan memahami apa model adalah belajar.

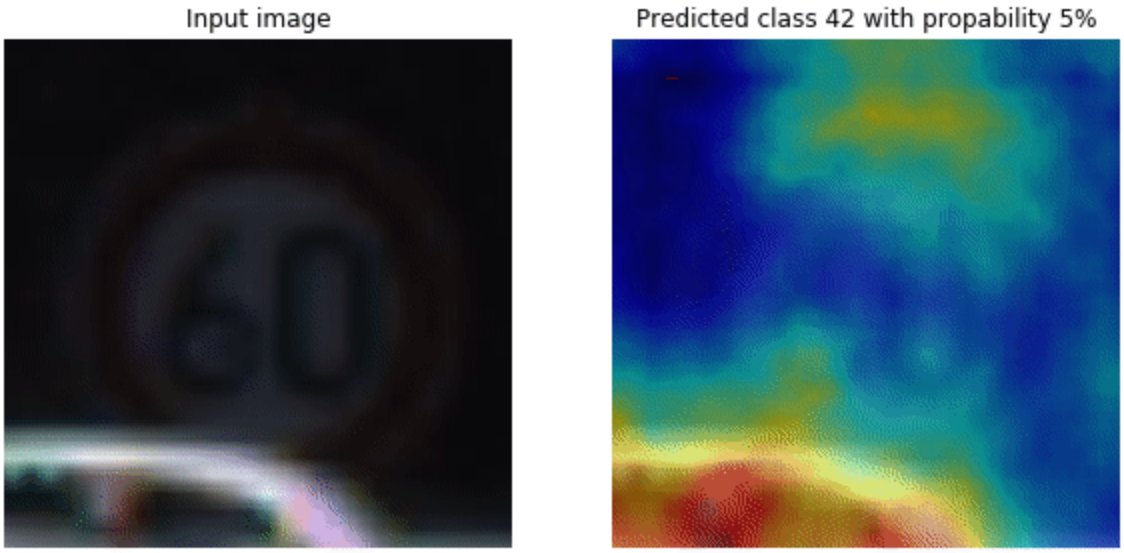
Animasi berikut menunjukkan skor perhatian dari 20 token input pertama untuk sepuluh iterasi dalam pekerjaan pelatihan yang disediakan dalam contoh notebook.


[CLS]
in
what
country
is
normandy
located
?
[SEP]
the
norman
##s
[CLS]
in
what
country
is
normandy
located
?
[SEP]
the
norman
##s

Menggunakan Debugger SageMaker untuk Memvisualisasikan Peta Aktivasi Kelas di Jaringan Neural Convolutional (CNNs)

Notebook ini menunjukkan bagaimana menggunakan SageMaker Debugger untuk merencanakan peta aktivasi kelas untuk deteksi gambar dan klasifikasi dalam jaringan saraf konvolusional (CNNs). Dalam pembelajaran mendalam, ajaringan saraf konvolusional (CNN atau ConvNet)adalah kelas jaringan saraf yang dalam, yang paling sering diterapkan untuk menganalisis citra visual. Salah satu aplikasi yang mengadopsi peta aktivasi kelas adalah mobil self-driving, yang memerlukan deteksi seketika dan klasifikasi gambar seperti rambu lalu lintas, jalan, dan hambatan.

Dalam buku catatan ini, model PyTorch Resnet dilatihDataset Tanda Lalu Lintas Jerman, yang berisi lebih dari 40 kelas objek terkait lalu lintas dan lebih dari 50.000 gambar secara total.



Selama proses pelatihan, SageMaker Debugger mengumpulkan tensor untuk merencanakan peta aktivasi kelas secara real time. Seperti yang ditunjukkan pada gambar animasi, peta aktivasi kelas (juga disebut sebagaipeta saliency) menyoroti daerah dengan aktivasi tinggi dalam warna merah.

Menggunakan tensor yang ditangkap oleh Debugger, Anda dapat memvisualisasikan bagaimana peta aktivasi berkembang selama pelatihan model. Model dimulai dengan mendeteksi tepi di sudut kiri bawah pada awal pekerjaan pelatihan. Seiring perkembangan pelatihan, fokus bergeser ke pusat dan mendeteksi tanda batas kecepatan, dan model berhasil memprediksi gambar input sebagai Kelas 3, yang merupakan kelas tanda batas kecepatan 60km/jam, dengan tingkat kepercayaan 97%.

Pekerjaan Pelatihan Debug Menggunakan Amazon SageMaker Debugger

Untuk mempersiapkan skrip latihan dan menjalankan pekerjaan pelatihan dengan SageMaker Debugger untuk men-debug kemajuan pelatihan model, Anda mengikuti proses dua langkah yang khas: memodifikasi skrip latihan Anda menggunakan `sagemaker-debugger` Python SDK, dan buat estimator menggunakan Python SDK. SageMaker SageMaker Lihat topik berikut untuk mempelajari cara menggunakan fungsi SageMaker debugging Debugger.

Topik

Langkah 1: Sesuaikan Skrip Pelatihan Anda untuk Mendaftarkan Hook

Amazon SageMaker Debugger dilengkapi dengan perpustakaan klien yang disebut`sagemaker-debugger`Python. Klasters`sagemaker-debugger`Python SDK menyediakan alat untuk mengadaptasi skrip latihan Anda sebelum pelatihan dan alat analisis setelah pelatihan. Di halaman ini, Anda akan belajar cara menyesuaikan skrip pelatihan Anda menggunakan pustaka klien.

Klasters`sagemaker-debugger`Python SDK menyediakan fungsi wrapper yang membantu mendaftarkan hook untuk mengekstrak tensor model, tanpa mengubah skrip latihan Anda. Untuk memulai dengan mengumpulkan tensor keluaran model dan men-debug mereka untuk menemukan masalah pelatihan, buat modifikasi berikut dalam skrip pelatihan Anda.

> **Tip**
>
> Saat Anda mengikuti halaman ini, gunakan`sagemaker-debugger`Dokumentasi SDK sumber terbukaReferensi API.

Topik

Adaptasikan PyTorch Skrip Pelatihan

Untuk mulai mengumpulkan tensor keluaran model dan masalah pelatihan debug, buat modifikasi berikut pada Anda PyTorch naskah pelatihan.

Untuk PyTorch 1.12.0

Jika Anda membawa PyTorch skrip pelatihan, Anda dapat menjalankan pekerjaan pelatihan dan mengekstrak tensor keluaran model dengan beberapa baris kode tambahan dalam skrip pelatihan Anda. Anda perlu menggunakanAPI kaitdi dalam`sagemaker-debugger`pustaka klien. Berjalan melalui petunjuk berikut yang memecah langkah-langkah dengan contoh kode.

1. Buat hook.

   (Direkomendasikan) Untuk pekerjaan pelatihan dalam SageMaker

   ```
   import smdebug.pytorch as smd
   hook=smd.get_hook(create_if_not_exists=True)
   ```

   Saat Anda memulai tugas pelatihanthe section called “Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK”dengan DebuggerHookConfig, TensorBoardConfig, atau Aturan dalam estimator Anda, SageMakermenambahkan file konfigurasi JSON ke instance latihan Anda yang diambil oleh`get_hook`fungsi. Perhatikan bahwa jika Anda tidak menyertakan salah satu API konfigurasi dalam estimator Anda, tidak akan ada file konfigurasi untuk hook untuk menemukan, dan fungsi kembali`None`.

   (Opsional) Untuk pekerjaan pelatihan di luar SageMaker

   Jika Anda menjalankan pekerjaan pelatihan dalam mode lokal, langsung SageMaker Instans notebook, instans Amazon EC2, atau perangkat lokal Anda sendiri, gunakan`smd.Hook`kelas untuk membuat hook. Namun, pendekatan ini hanya dapat menyimpan koleksi tensor dan dapat digunakan TensorBoard Visualisasi. SageMaker Aturan bawaan debugger tidak berfungsi dengan mode lokal karena Aturan memerlukan SageMaker Instans pelatihan dan S3 untuk menyimpan output dari instans jarak jauh secara real time. Klaster`smd.get_hook`Pengembalian API`None`dalam kasus ini.

   Jika Anda ingin membuat hook manual untuk menyimpan tensor dalam mode lokal, gunakan cuplikan kode berikut dengan logika untuk memeriksa apakah`smd.get_hook`Pengembalian API`None`dan membuat hook manual menggunakan`smd.Hook`Kelas Perhatikan bahwa Anda dapat menentukan direktori keluaran apa pun di komputer lokal Anda.

```
import smdebug.pytorch as smd
hook=smd.get_hook(create_if_not_exists=True)

if hook is None:
    hook=smd.Hook(
        out_dir='/path/to/your/local/output/',
        export_tensorboard=True
    )
```

2. Bungkus model Anda dengan metode kelas hook.

   Klasterhook.register_module()metode mengambil model Anda dan iterasi melalui setiap lapisan, mencari tensor apa pun yang cocok dengan ekspresi reguler yang akan Anda berikan melalui konfigurasithe section called “Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK”. Tensor yang dapat dikumpulkan melalui metode kait ini adalah bobot, bias, aktivasi, gradien, input, dan output.

   ```
   hook.register_module(model)
   ```

   > **Tip**
   >
   > Jika Anda mengumpulkan seluruh tensor keluaran dari model pembelajaran mendalam yang besar, ukuran total koleksi tersebut dapat tumbuh secara eksponensial dan dapat menyebabkan kemacetan. Jika Anda ingin menyimpan tensor tertentu, Anda juga dapat menggunakanhook.save_tensor()Metode. Metode ini membantu Anda memilih variabel untuk tensor tertentu dan menyimpan ke koleksi kustom bernama yang Anda inginkan. Untuk informasi selengkapnya, lihatLangkah 7dari instruksi ini.

3. Warp fungsi kerugian dengan metode kelas hook.

   Klasterhook.register_lossadalah untuk membungkus fungsi kerugian. Ini mengekstrak nilai kerugian setiapsave_intervalyang akan Anda tetapkan selama konfigurasithe section called “Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK”, dan menyelamatkan mereka ke"losses"koleksi.

   ```
   hook.register_loss(loss_function)
   ```

4. Tambahkan`hook.set_mode(ModeKeys.TRAIN)`di blok kereta. Ini menunjukkan bahwa pengumpulan tensor diekstraksi selama fase pelatihan.

```
def train():
    ...
    hook.set_mode(ModeKeys.TRAIN)
```

5. Tambahkan`hook.set_mode(ModeKeys.EVAL)`di blok validasi. Ini menunjukkan bahwa koleksi tensor diekstraksi selama fase validasi.

```
def validation():
    ...
    hook.set_mode(ModeKeys.EVAL)
```

6. Gunakan`hook.save_scalar()`untuk menyimpan skalar kustom. Anda dapat menyimpan nilai skalar yang tidak ada dalam model Anda. Misalnya, jika Anda ingin mencatat nilai akurasi yang dihitung selama evaluasi, tambahkan baris kode berikut di bawah garis tempat Anda menghitung akurasi.

```
hook.save_scalar("accuracy", accuracy)
```

   Perhatikan bahwa Anda perlu menyediakan string sebagai argumen pertama untuk memberi nama koleksi skalar khusus. Ini adalah nama yang akan digunakan untuk memvisualisasikan nilai skalar di TensorBoard, dan bisa berupa string yang Anda inginkan.

7. Gunakan`hook.save_tensor()`untuk menyimpan tensor khusus. Demikian pula`hook.save_scalar()`, Anda dapat menyimpan tensor tambahan, menentukan koleksi tensor Anda sendiri. Misalnya, Anda dapat mengekstrak data gambar input yang diteruskan ke model dan menyimpannya sebagai tensor khusus dengan menambahkan baris kode berikut, di mana`"images"`adalah nama contoh tensor kustom,`image_inputs`adalah variabel contoh untuk data gambar input.

```
hook.save_tensor("images", image_inputs)
```

   Perhatikan bahwa Anda harus memberikan string ke argumen pertama untuk memberi nama tensor khusus.`hook.save_tensor()`memiliki argumen ketiga`collections_to_write`untuk menentukan koleksi tensor untuk menyimpan tensor khusus. Defaultnya adalah `collections_to_write="default"`. Jika Anda tidak secara eksplisit menentukan argumen ketiga, tensor kustom disimpan ke`"default"`Pengumpulan tensor.

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan kethe section called “Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK”.

Adaptasikan TensorFlow Skrip Pelatihan

Untuk mulai mengumpulkan tensor keluaran model dan masalah pelatihan debug, buat modifikasi berikut pada Anda TensorFlow naskah pelatihan.

Buat hook untuk pekerjaan pelatihan di dalamnya SageMaker

```
import smdebug.tensorflow as smd

hook=smd.get_hook(hook_type="keras", create_if_not_exists=True)
```

Hal ini menciptakan hook ketika Anda memulai SageMaker Tugas pelatihan. Saat Anda memulai tugas pelatihanthe section called “Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK”dengan`DebuggerHookConfig`,`TensorBoardConfig`, atau`Rules`di estimator Anda, SageMaker menambahkan file konfigurasi JSON ke instance latihan Anda yang diambil oleh`smd.get_hook`Metode. Perhatikan bahwa jika Anda tidak menyertakan salah satu API konfigurasi dalam estimator Anda, tidak akan ada file konfigurasi untuk hook untuk menemukan, dan fungsi kembali`None`.

(Opsional) Buat hook untuk pekerjaan pelatihan di luar SageMaker

Jika Anda menjalankan pekerjaan pelatihan dalam mode lokal, langsung SageMaker Instans notebook, instans Amazon EC2, atau perangkat lokal Anda sendiri, gunakan`smd.Hook`kelas untuk membuat hook. Namun, pendekatan ini hanya dapat menyimpan koleksi tensor dan dapat digunakan TensorBoardVisualisasi. SageMaker Aturan bawaan debugger tidak berfungsi dengan mode lokal. Klasters`smd.get_hook`metode juga mengembalikan`None`dalam kasus ini.

Jika Anda ingin membuat hook manual, gunakan potongan kode berikut dengan logika untuk memeriksa apakah hook kembali`None`dan membuat hook manual menggunakan`smd.Hook`Kelas

```
import smdebug.tensorflow as smd

hook=smd.get_hook(hook_type="keras", create_if_not_exists=True)

if hook is None:
    hook=smd.KerasHook(
        out_dir='/path/to/your/local/output/',
```

```
        export_tensorboard=True
    )
```

Setelah menambahkan kode pembuatan kait, lanjutkan ke topik berikut untuk TensorFlowKeras.

> **Note**
>
> SageMaker Debugger saat ini mendukung TensorFlow Hanya Keras.

Daftarkan hook di TensorFlow Skrip pelatihan

Precedure berikut memandu Anda melalui cara menggunakan hook dan metodenya untuk mengumpulkan skalar dan tensor keluaran dari model dan pengoptimal Anda.

1. Bungkus model Keras dan pengoptimal Anda dengan metode kelas hook.

   Klasterhook.register_model()metode mengambil model Anda dan iterasi melalui setiap lapisan, mencari tensor apa pun yang cocok dengan ekspresi reguler yang akan Anda berikan melalui konfigurasithe section called “Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK”. Tensor yang dapat ditagih melalui metode kait ini adalah bobot, bias, dan aktivasi.

   ```
   model=tf.keras.Model(...)
   hook.register_model(model)
   ```

2. Bungkus pengoptimal denganhook.wrap_optimizer()Metode.

   ```
   optimizer=tf.keras.optimizers.Adam(...)
   optimizer=hook.wrap_optimizer(optimizer)
   ```

3. Kompilasi model dalam mode eager di TensorFlow.

   Untuk mengumpulkan tensor dari model, seperti tensor input dan output dari setiap lapisan, Anda harus menjalankan pelatihan dalam mode bersemangat. Jika tidak, SageMakerDebugger tidak akan dapat mengumpulkan tensor. Namun, tensor lain, seperti bobot model, bias, dan kerugian, dapat dikumpulkan tanpa berjalan secara eksplisit dalam mode bersemangat.

   ```
   model.compile(
       loss="categorical_crossentropy",
   ```

```
    optimizer=optimizer,
    metrics=["accuracy"],
    # Required for collecting tensors of each layer
    run_eagerly=True
)
```

4. Daftarkan hook ketf.keras.Model.fit()Metode.

   Untuk mengumpulkan tensor dari kait yang Anda daftarkan, tambahkancallbacks=[hook]ke Kerasmodel.fit()metode kelas. Ini akan melewatisagemaker-debuggerhook sebagai callback Keras.

```
model.fit(
    X_train, Y_train,
    batch_size=batch_size,
    epochs=epoch,
    validation_data=(X_valid, Y_valid),
    shuffle=True,
    callbacks=[hook]
)
```

5. TensorFlow 2.x hanya menyediakan variabel gradien simbolis yang tidak menyediakan akses ke nilai-nilai mereka. Untuk mengumpulkan gradien, bungkustf.GradientTapeoleh thehook.wrap_tape(), yang mengharuskan Anda untuk menulis langkah pelatihan Anda sendiri sebagai berikut.

```
def training_step(model, dataset):
    with hook.wrap_tape(tf.GradientTape()) as tape:
        pred=model(data)
        loss_value=loss_fn(labels, pred)
    grads=tape.gradient(loss_value, model.trainable_variables)
    optimizer.apply_gradients(zip(grads, model.trainable_variables))
```

   Dengan membungkus rekaman itu,sagemaker-debuggerhook dapat mengidentifikasi tensor keluaran seperti gradien, parameter, dan kerugian. Membungkus pita memastikan bahwahook.wrap_tape()metode di sekitar fungsi dari objek tape, sepertipush_tape(),pop_tape(),gradient(), akan mengatur penulis SageMaker Debugger dan simpan tensor yang disediakan sebagai masukangradient()(variabel dan kerugian yang dapat dilatih) dan outputgradient()(gradien).

> **Note**
>
> Untuk mengumpulkan loop pelatihan kustom, pastikan bahwa Anda menggunakan mode bersemangat. Jika tidak, SageMaker Debugger tidak dapat mengumpulkan tensor apa pun.

Untuk daftar lengkap tindakan yang`sagemaker-debugger`hook API menawarkan untuk membuat kait dan menyimpan tensor, lihatMetode Hookdi dalam*`sagemaker-debugger`*Dokumentasi Python.

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan kethe section called “Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK”.

Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK

Untuk mengonfigurasi SageMaker estimator dengan SageMaker Debugger, gunakan Amazon SageMaker Python SDK dan tentukan parameter khusus Debugger. Untuk sepenuhnya memanfaatkan fungsi debugging, ada tiga parameter yang perlu Anda konfigurasi:`debugger_hook_config`,`tensorboard_output_config`, dan`rules`.

> **Important**
>
> Sebelum membuat dan menjalankan metode estimator fit untuk meluncurkan pekerjaan pelatihan, pastikan Anda menyesuaikan skrip pelatihan Anda mengikuti instruksi dithe section called “Langkah 1: Sesuaikan Skrip Pelatihan Anda untuk Mendaftarkan Hook”.

Buat SageMaker Estimator dengan parameter khusus Debugger

Contoh kode dalam bagian ini menunjukkan cara membangun SageMaker estimator dengan parameter khusus Debugger.

> **Note**
>
> Contoh kode berikut adalah template untuk membangun estimator SageMaker kerangka kerja dan tidak langsung dieksekusi. Anda perlu melanjutkan ke bagian berikutnya dan mengkonfigurasi parameter khusus Debugger.

PyTorch

```
# An example of constructing a SageMaker PyTorch estimator
import boto3
import sagemaker
from sagemaker.pytorch import PyTorch
from sagemaker.debugger import CollectionConfig, DebuggerHookConfig, Rule,
 rule_configs

session=boto3.session.Session()
region=session.region_name

debugger_hook_config=DebuggerHookConfig(...)
rules=[
    Rule.sagemaker(rule_configs.built_in_rule())
]

estimator=PyTorch(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-demo",
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="1.12.0",
    py_version="py37",

    # Debugger-specific parameters
    debugger_hook_config=debugger_hook_config,
    rules=rules
)

estimator.fit(wait=False)
```

TensorFlow

```
# An example of constructing a SageMaker TensorFlow estimator
import boto3
import sagemaker
from sagemaker.tensorflow import TensorFlow
from sagemaker.debugger import CollectionConfig, DebuggerHookConfig, Rule,
 rule_configs

session=boto3.session.Session()
```

```
region=session.region_name

debugger_hook_config=DebuggerHookConfig(...)
rules=[
    Rule.sagemaker(rule_configs.built_in_rule()),
    ProfilerRule.sagemaker(rule_configs.BuiltInRule())
]

estimator=TensorFlow(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-demo",
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="2.9.0",
    py_version="py39",

    # Debugger-specific parameters
    debugger_hook_config=debugger_hook_config,
    rules=rules
)

estimator.fit(wait=False)
```

MXNet

```
# An example of constructing a SageMaker MXNet estimator
import sagemaker
from sagemaker.mxnet import MXNet
from sagemaker.debugger import CollectionConfig, DebuggerHookConfig, Rule,
 rule_configs

debugger_hook_config=DebuggerHookConfig(...)
rules=[
    Rule.sagemaker(rule_configs.built_in_rule())
]

estimator=MXNet(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-demo",
    instance_count=1,
    instance_type="ml.p3.2xlarge",
```

```
    framework_version="1.7.0",
    py_version="py37",

    # Debugger-specific parameters
    debugger_hook_config=debugger_hook_config,
    rules=rules
)

estimator.fit(wait=False)
```

XGBoost

```
# An example of constructing a SageMaker XGBoost estimator
import sagemaker
from sagemaker.xgboost.estimator import XGBoost
from sagemaker.debugger import CollectionConfig, DebuggerHookConfig, Rule,
 rule_configs

debugger_hook_config=DebuggerHookConfig(...)
rules=[
    Rule.sagemaker(rule_configs.built_in_rule())
]

estimator=XGBoost(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-demo",
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="1.5-1",

    # Debugger-specific parameters
    debugger_hook_config=debugger_hook_config,
    rules=rules
)

estimator.fit(wait=False)
```

Generic estimator

```
# An example of constructing a SageMaker generic estimator using the XGBoost
 algorithm base image
import boto3
```

```
import sagemaker
from sagemaker.estimator import Estimator
from sagemaker import image_uris
from sagemaker.debugger import CollectionConfig, DebuggerHookConfig, Rule,
 rule_configs

debugger_hook_config=DebuggerHookConfig(...)
rules=[
    Rule.sagemaker(rule_configs.built_in_rule())
]

region=boto3.Session().region_name
xgboost_container=sagemaker.image_uris.retrieve("xgboost", region, "1.5-1")

estimator=Estimator(
    role=sagemaker.get_execution_role()
    image_uri=xgboost_container,
    base_job_name="debugger-demo",
    instance_count=1,
    instance_type="ml.m5.2xlarge",

    # Debugger-specific parameters
    debugger_hook_config=debugger_hook_config,
    rules=rules
)

estimator.fit(wait=False)
```

Konfigurasikan parameter berikut untuk mengaktifkan SageMaker Debugger:

- `debugger_hook_config`(objek `DebuggerHookConfig`) - Diperlukan untuk mengaktifkan hook dalam skrip latihan yang disesuaikan selamathe section called “Langkah 1: Sesuaikan Skrip Pelatihan Anda untuk Mendaftarkan Hook”, konfigurasikan peluncur SageMaker pelatihan (estimator) untuk mengumpulkan tensor keluaran dari pekerjaan pelatihan Anda, dan menyimpan tensor ke bucket S3 aman atau mesin lokal Anda. Untuk mempelajari cara mengkonfigurasi`debugger_hook_config` parameter, lihatKonfigurasikan SageMaker Debugger untuk Menyimpan Tensor.
- `rules`(daftar `Rule`objek) - Konfigurasikan parameter ini untuk mengaktifkan aturan bawaan SageMaker Debugger yang ingin Anda jalankan secara real time. Aturan bawaan adalah logika yang secara otomatis men-debug kemajuan pelatihan model Anda dan menemukan

masalah latihan dengan menganalisis tensor keluaran yang disimpan di bucket S3 aman Anda. Untuk mempelajari cara mengkonfigurasi`rules` parameter, lihatKonfigurasikan Aturan Bawaan Debugger. Untuk menemukan daftar lengkap aturan bawaan untuk men-debug tensor keluaran, lihatthe section called “Aturan Debugger”. Jika Anda ingin membuat logika sendiri untuk mendeteksi masalah pelatihan apa pun, lihatthe section called “Buat Aturan Khusus”.

> **Note**
>
> Aturan bawaan hanya akan tersedia melalui instans SageMaker pelatihan. Anda tidak dapat menggunakannya dalam mode lokal.

- `tensorboard_output_config`(objek `TensorBoardOutputConfig`) - Konfigurasikan SageMaker Debugger untuk mengumpulkan tensor keluaran dalam format TensorBoard yang kompatibel dan simpan ke jalur keluaran S3 Anda yang ditentukan dalam`TensorBoardOutputConfig` objek. Untuk mempelajari selengkapnya, lihat the section called “Visualisasikan Tensor Output Debugger di TensorBoard”.

  > **Note**
  >
  > `tensorboard_output_config`Harus dikonfigurasi dengan`debugger_hook_config` parameter, yang juga mengharuskan Anda untuk menyesuaikan skrip pelatihan Anda dengan menambahkan`sagemaker-debugger` hook.

> **Note**
>
> SageMaker Debugger menyimpan tensor keluaran dengan aman di subfolder bucket S3 Anda. Misalnya, format URI bucket S3 default di akun Anda adalah`s3://sagemaker-<region>-<12digit_account_id>/<base-job-name>/<debugger-subfolders>/`. Ada dua subfolder yang dibuat oleh SageMaker Debugger:`debug-output`, dan`rule-output`. Jika Anda menambahkan`tensorboard_output_config` parameter, Anda juga akan menemukan`tensorboard-output` folder.

Lihat topik berikut untuk menemukan lebih banyak contoh cara mengkonfigurasi parameter khusus Debugger secara rinci.

Topik

- Konfigurasikan SageMaker Debugger untuk Menyimpan Tensor
- Konfigurasikan Aturan Bawaan Debugger
- Matikan Debugger
- Berguna SageMaker Estimator Classmethods untuk Debugger

Konfigurasikan SageMaker Debugger untuk Menyimpan Tensor

Tensor adalah pengumpulan data parameter yang diperbarui dari umpan mundur dan maju dari setiap iterasi pelatihan. SageMaker Debugger mengumpulkan tensor keluaran untuk menganalisis keadaan pekerjaan pelatihan. SageMaker Operasi debugger `CollectionConfig`dan `DebuggerHookConfig`API menyediakan metode untuk mengelompokkan tensor ke dalam koleksi dan menyimpannya ke bucket S3 target.

> **Note**
>
> Setelah dikonfigurasi dan diaktifkan dengan benar, SageMaker Debugger menyimpan tensor keluaran dalam bucket S3 default, kecuali ditentukan lain. Format URI bucket S3 default adalah`s3://sagemaker-<region>-<12digit_account_id>/<training-job-name>/debug-output/`.

Saat membangun SageMaker estimator, aktifkan SageMaker Debugger dengan menentukan`debugger_hook_config` parameter. Langkah-langkah berikut mencakup contoh cara mengatur operasi`debugger_hook_config` penggunaan`CollectionConfig` dan`DebuggerHookConfig` API untuk menarik tensor dari pekerjaan pelatihan Anda dan menyimpannya.

Mengonfigurasi Koleksi Tensor Menggunakan**`CollectionConfig`** API

Gunakan operasi`CollectionConfig` API untuk mengonfigurasi koleksi tensor. Debugger menyediakan koleksi tensor bawaan yang mencakup berbagai parameter ekspresi reguler (regex) jika menggunakan kerangka kerja deep learning yang didukung Debugger dan algoritma machine learning. Seperti yang ditunjukkan pada kode contoh berikut, tambahkan koleksi tensor bawaan yang ingin Anda debug.

```
from sagemaker.debugger import CollectionConfig
```

```
collection_configs=[
    CollectionConfig(name="weights"),
    CollectionConfig(name="gradients")
]
```

Koleksi sebelumnya menyiapkan hook Debugger untuk menyimpan tensor setiap 500 langkah berdasarkan`"save_interval"` nilai default.

Untuk daftar lengkap koleksi bawaan Debugger yang tersedia, lihat Koleksi Bawaan Debugger.

Jika Anda ingin menyesuaikan koleksi bawaan, seperti mengubah interval penyimpanan dan tensor regex, gunakan`CollectionConfig` template berikut untuk menyesuaikan parameter.

```
from sagemaker.debugger import CollectionConfig

collection_configs=[
    CollectionConfig(
        name="tensor_collection",
        parameters={
            "key_1": "value_1",
            "key_2": "value_2",
            ...
            "key_n": "value_n"
        }
    )
]
```

Untuk informasi selengkapnya tentang kunci parameter yang tersedia, lihat CollectionConfigdi Amazon SageMaker Python SDK. Misalnya, contoh kode berikut menunjukkan bagaimana Anda dapat menyesuaikan interval penyimpanan pengumpulan tensor “kerugian” pada fase pelatihan yang berbeda: simpan kerugian setiap 100 langkah dalam fase pelatihan dan kehilangan validasi setiap 10 langkah dalam fase validasi.

```
from sagemaker.debugger import CollectionConfig

collection_configs=[
    CollectionConfig(
        name="losses",
        parameters={
            "train.save_interval": "100",
            "eval.save_interval": "10"
```

```
        }
    )
]
```

> **Tip**
>
> Objek konfigurasi koleksi tensor ini dapat digunakan DebuggerHookConfiguntuk operasi Rule API.

Konfigurasikan**`DebuggerHookConfig`** API untuk Menyimpan Tensor

Gunakan DebuggerHookConfigAPI untuk membuat`debugger_hook_config` objek menggunakan`collection_configs` objek yang Anda buat pada langkah sebelumnya.

```
from sagemaker.debugger import DebuggerHookConfig

debugger_hook_config=DebuggerHookConfig(
    collection_configs=collection_configs
)
```

Debugger menyimpan tensor keluaran pelatihan model ke bucket S3 default. Format URI bucket S3 default adalah`s3://sagemaker-<region>-<12digit_account_id>/<training-job-name>/debug-output/`.

Jika Anda ingin menentukan URI bucket S3 yang tepat, gunakan contoh kode berikut:

```
from sagemaker.debugger import DebuggerHookConfig

debugger_hook_config=DebuggerHookConfig(
    s3_output_path="specify-your-s3-bucket-uri"
    collection_configs=collection_configs
)
```

Untuk informasi selengkapnya, lihat DebuggerHookConfigdi Amazon SageMaker Python SDK.

Contoh Notebook dan Sampel Kode untuk Mengkonfigurasi Debugger Hook

Bagian berikut menyediakan notebook dan contoh kode tentang cara menggunakan kait Debugger untuk menyimpan, mengakses, dan memvisualisasikan tensor keluaran.

Topik

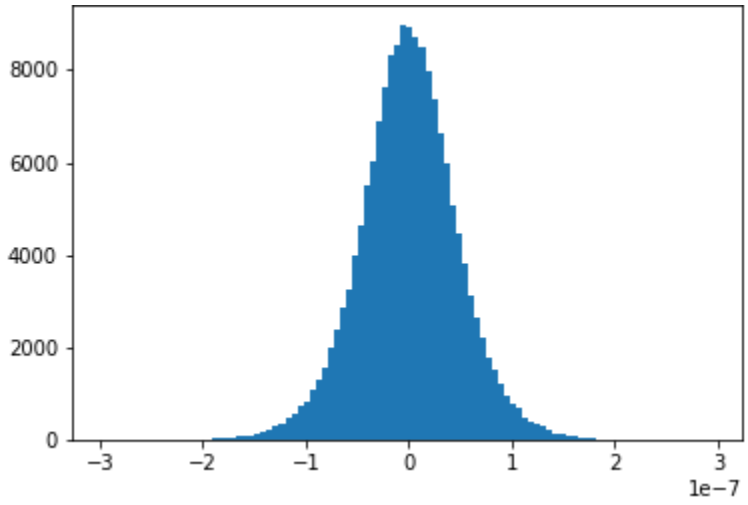
- Notebook Contoh Visualisasi Tensor
- Simpan Tensor Menggunakan Koleksi Bawaan Debugger
- Simpan Tensor Menggunakan Koleksi Built-in yang Dimodifikasi Debugger
- Simpan Tensor Menggunakan Koleksi Kustom Debugger

Notebook Contoh Visualisasi Tensor

Dua contoh notebook berikut menunjukkan penggunaan lanjutan Amazon SageMaker Debugger untuk memvisualisasikan tensor. Debugger memberikan pandangan transparan ke dalam pelatihan model pembelajaran mendalam.

- Analisis Tensor Interaktif di Notebook SageMaker Studio dengan MXNet

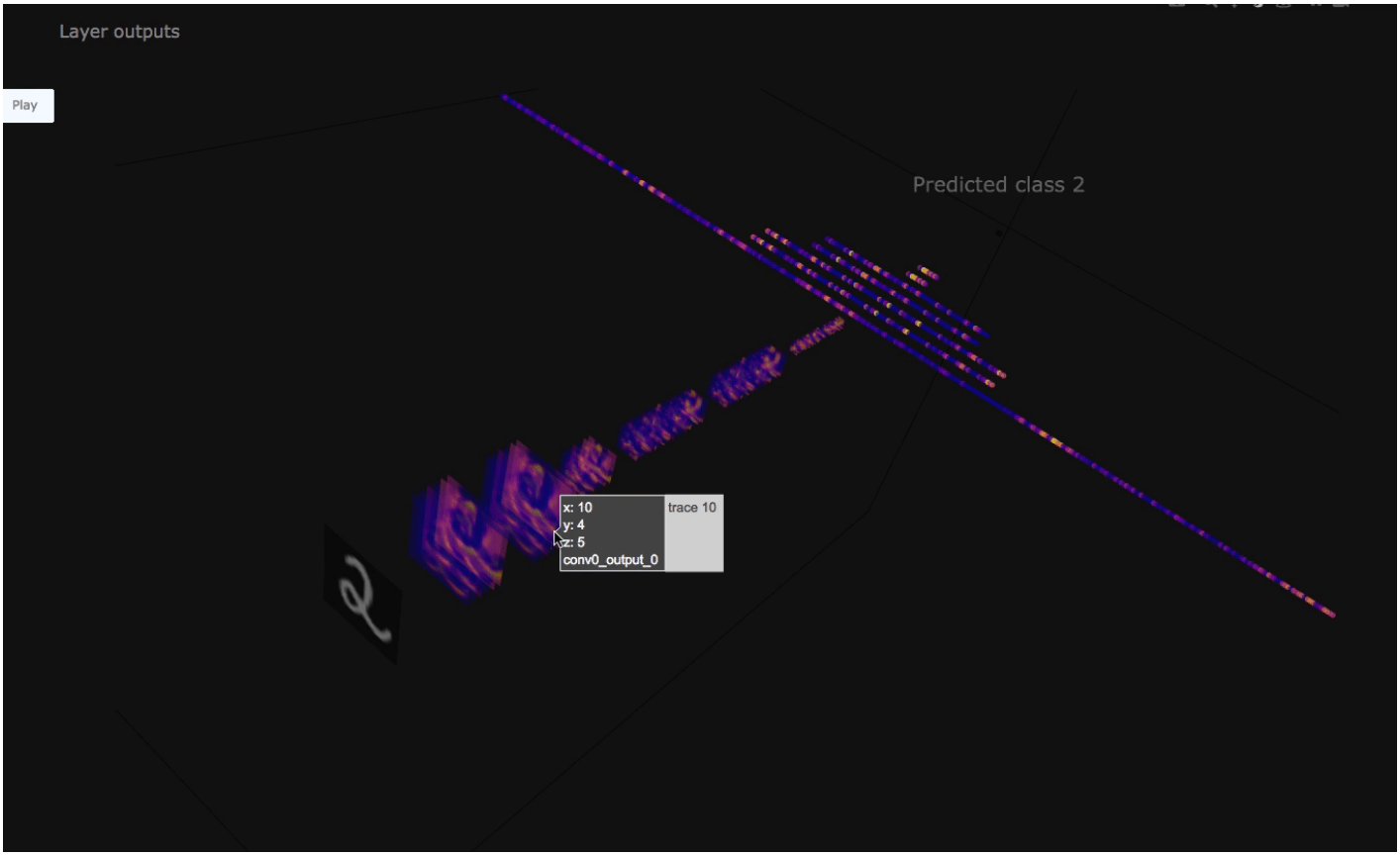
  Contoh notebook ini menunjukkan cara memvisualisasikan tensor yang disimpan menggunakan Amazon SageMaker Debugger. Dengan memvisualisasikan tensor, Anda dapat melihat bagaimana nilai tensor berubah saat melatih algoritme pembelajaran mendalam. Notebook ini mencakup pekerjaan pelatihan dengan jaringan saraf yang dikonfigurasi dengan buruk dan menggunakan Amazon SageMaker Debugger untuk mengumpulkan dan menganalisis tensor, termasuk gradien, output aktivasi, dan bobot. Misalnya, plot berikut menunjukkan distribusi gradien lapisan konvolusional yang menderita masalah gradien yang hilang.



  Notebook ini juga menggambarkan bagaimana pengaturan hyperparameter awal yang baik meningkatkan proses pelatihan dengan menghasilkan plot distribusi tensor yang sama.

- Visualisasi dan Debugging Tensor dari Pelatihan Model MXNet

Contoh notebook ini menunjukkan cara menyimpan dan memvisualisasikan tensor dari pekerjaan pelatihan model MXNet Gluon menggunakan Amazon SageMaker Debugger. Ini menggambarkan bahwa Debugger diatur untuk menyimpan semua tensor ke bucket Amazon S3 dan mengambil output ReLu aktivasi untuk visualisasi. Gambar berikut menunjukkan visualisasi tiga dimensi dari output ReLu aktivasi. Skema warna diatur ke biru untuk menunjukkan nilai mendekati 0 dan kuning untuk menunjukkan nilai yang mendekati 1.



Dalam notebook ini,`TensorPlot` kelas yang diimpor dari`tensor_plot.py` dirancang untuk merencanakan jaringan saraf konvolusional (CNN) yang mengambil gambar dua dimensi untuk input. `tensor_plot.py`Skrip yang disediakan dengan notebook mengambil tensor menggunakan Debugger dan memvisualisasikan CNN. Anda dapat menjalankan notebook ini di SageMaker Studio untuk mereproduksi visualisasi tensor dan mengimplementasikan model jaringan saraf konvolusional Anda sendiri.

- Analisis Tensor Waktu Nyata di SageMaker Notebook dengan MXNet

  Contoh ini memandu Anda menginstal komponen yang diperlukan untuk memancarkan tensor dalam pekerjaan SageMaker pelatihan Amazon dan menggunakan operasi Debugger API untuk mengakses tensor tersebut saat pelatihan berjalan. Model CNN gluon dilatih pada set

data Fashion MNIST. Saat pekerjaan berjalan, Anda akan melihat bagaimana Debugger mengambil output aktivasi dari lapisan konvolusional pertama dari masing-masing 100 batch dan memvisualisasikannya. Juga, ini akan menunjukkan cara memvisualisasikan bobot setelah pekerjaan selesai.

Simpan Tensor Menggunakan Koleksi Bawaan Debugger

Anda dapat menggunakan koleksi tensor bawaan menggunakan`CollectionConfig` API dan menyimpannya menggunakan`DebuggerHookConfig` API. Contoh berikut menunjukkan cara untuk menggunakan pengaturan default konfigurasi kait Debugger untuk membangun SageMaker TensorFlow estimator. Anda juga dapat memanfaatkan ini untuk penaksir MXNet PyTorch,, dan XGBoost.

> **Note**
>
> Dalam contoh kode berikut,`s3_output_path` parameter untuk`DebuggerHookConfig` adalah opsional. Jika Anda tidak menentukannya, Debugger menyimpan tensor di`s3://<output_path>/debug-output/`, di`<output_path>` mana jalur keluaran default dari pekerjaan SageMaker pelatihan. Misalnya:
>
> ```
> "s3://sagemaker-us-east-1-111122223333/sagemaker-debugger-training-YYYY-MM-DD-HH-MM-SS-123/debug-output"
> ```

```
import sagemaker
from sagemaker.tensorflow import TensorFlow
from sagemaker.debugger import DebuggerHookConfig, CollectionConfig

# use Debugger CollectionConfig to call built-in collections
collection_configs=[
        CollectionConfig(name="weights"),
        CollectionConfig(name="gradients"),
        CollectionConfig(name="losses"),
        CollectionConfig(name="biases")
    ]

# configure Debugger hook
# set a target S3 bucket as you want
sagemaker_session=sagemaker.Session()
```

```
BUCKET_NAME=sagemaker_session.default_bucket()
LOCATION_IN_BUCKET='debugger-built-in-collections-hook'

hook_config=DebuggerHookConfig(
    s3_output_path='s3://{BUCKET_NAME}/{LOCATION_IN_BUCKET}'.
                   format(BUCKET_NAME=BUCKET_NAME,
                          LOCATION_IN_BUCKET=LOCATION_IN_BUCKET),
    collection_configs=collection_configs
)

# construct a SageMaker TensorFlow estimator
sagemaker_estimator=TensorFlow(
    entry_point='directory/to/your_training_script.py',
    role=sm.get_execution_role(),
    base_job_name='debugger-demo-job',
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="2.9.0",
    py_version="py39",

    # debugger-specific hook argument below
    debugger_hook_config=hook_config
)

sagemaker_estimator.fit()
```

Untuk melihat daftar koleksi bawaan Debugger, lihat Koleksi Bawaan Debugger.

Simpan Tensor Menggunakan Koleksi Built-in yang Dimodifikasi Debugger

Anda dapat memodifikasi koleksi bawaan Debugger menggunakan operasi`CollectionConfig` API. Contoh berikut menunjukkan bagaimana untuk men-tweak`losses` koleksi built-in dan membangun SageMaker TensorFlow estimator. Anda juga dapat menggunakan ini untuk penaksir MXNet PyTorch,, dan XGBoost.

```
import sagemaker
from sagemaker.tensorflow import TensorFlow
from sagemaker.debugger import DebuggerHookConfig, CollectionConfig

# use Debugger CollectionConfig to call and modify built-in collections
collection_configs=[
    CollectionConfig(
                name="losses",
```

```
                parameters={"save_interval": "50"})]

# configure Debugger hook
# set a target S3 bucket as you want
sagemaker_session=sagemaker.Session()
BUCKET_NAME=sagemaker_session.default_bucket()
LOCATION_IN_BUCKET='debugger-modified-collections-hook'

hook_config=DebuggerHookConfig(
    s3_output_path='s3://{BUCKET_NAME}/{LOCATION_IN_BUCKET}'.
                   format(BUCKET_NAME=BUCKET_NAME,
                          LOCATION_IN_BUCKET=LOCATION_IN_BUCKET),
    collection_configs=collection_configs
)

# construct a SageMaker TensorFlow estimator
sagemaker_estimator=TensorFlow(
    entry_point='directory/to/your_training_script.py',
    role=sm.get_execution_role(),
    base_job_name='debugger-demo-job',
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="2.9.0",
    py_version="py39",

    # debugger-specific hook argument below
    debugger_hook_config=hook_config
)

sagemaker_estimator.fit()
```

Untuk daftar lengkap`CollectionConfig` parameter, lihat Debugger CollectionConfig API.

Simpan Tensor Menggunakan Koleksi Kustom Debugger

Anda juga dapat menyimpan jumlah tensor yang dikurangi alih-alih set lengkap tensor (misalnya, jika Anda ingin mengurangi jumlah data yang disimpan di bucket Amazon S3 Anda). Contoh berikut menunjukkan cara menyesuaikan konfigurasi kait Debugger untuk menentukan tensor target yang ingin Anda simpan. Anda dapat menggunakan ini untuk TensorFlow, MXNet, PyTorch, dan XGBoost estimator.

```
import sagemaker
from sagemaker.tensorflow import TensorFlow
```

```
from sagemaker.debugger import DebuggerHookConfig, CollectionConfig

# use Debugger CollectionConfig to create a custom collection
collection_configs=[
        CollectionConfig(
            name="custom_activations_collection",
            parameters={
                "include_regex": "relu|tanh", # Required
                "reductions": "mean,variance,max,abs_mean,abs_variance,abs_max"
            })
    ]

# configure Debugger hook
# set a target S3 bucket as you want
sagemaker_session=sagemaker.Session()
BUCKET_NAME=sagemaker_session.default_bucket()
LOCATION_IN_BUCKET='debugger-custom-collections-hook'

hook_config=DebuggerHookConfig(
    s3_output_path='s3://{BUCKET_NAME}/{LOCATION_IN_BUCKET}'.
                    format(BUCKET_NAME=BUCKET_NAME,
                           LOCATION_IN_BUCKET=LOCATION_IN_BUCKET),
    collection_configs=collection_configs
)

# construct a SageMaker TensorFlow estimator
sagemaker_estimator=TensorFlow(
    entry_point='directory/to/your_training_script.py',
    role=sm.get_execution_role(),
    base_job_name='debugger-demo-job',
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="2.9.0",
    py_version="py39",

    # debugger-specific hook argument below
    debugger_hook_config=hook_config
)

sagemaker_estimator.fit()
```

Untuk daftar lengkap`CollectionConfig` parameter, lihat Debugger CollectionConfig.

Konfigurasikan Aturan Bawaan Debugger

Aturan bawaan Amazon SageMaker Debugger menganalisis tensor yang dipancarkan selama pelatihan model. SageMakerDebugger menawarkan operasi `Rule` API yang memantau kemajuan pekerjaan pelatihan dan kesalahan untuk keberhasilan melatih model Anda. Misalnya, aturan dapat mendeteksi apakah gradien terlalu besar atau terlalu kecil, apakah model overfitting atau overtraining, dan apakah pekerjaan pelatihan tidak mengurangi fungsi kerugian dan meningkatkan. Untuk melihat daftar lengkap aturan bawaan yang tersedia, lihatDaftar Aturan Built-in Debugger.

Dalam topik berikut, Anda akan belajar cara menggunakan aturan bawaan SageMaker Debugger.

Topik

- Gunakan Aturan Bawaan Debugger dengan Pengaturan Parameter Default
- Gunakan Aturan Bawaan Debugger dengan Nilai Parameter Kustom
- Contoh Notebook dan Sampel Kode untuk Mengkonfigurasi Aturan Debugger

Gunakan Aturan Bawaan Debugger dengan Pengaturan Parameter Default

Untuk menentukan Debugger built-in aturan dalam estimator, Anda perlu mengkonfigurasi daftar objek. Contoh kode berikut menunjukkan struktur dasar daftar Debugger built-in aturan:

```
from sagemaker.debugger import Rule, rule_configs

rules=[
    Rule.sagemaker(rule_configs.built_in_rule_name_1()),
    Rule.sagemaker(rule_configs.built_in_rule_name_2()),
    ...
    Rule.sagemaker(rule_configs.built_in_rule_name_n()),
    ... # You can also append more profiler rules in the
 ProfilerRule.sagemaker(rule_configs.*()) format.
]
```

Untuk informasi selengkapnya tentang nilai parameter default dan deskripsi aturan bawaan, lihatDaftar Aturan Built-in Debugger.

Untuk menemukan referensi SageMaker Debugger API, lihat `sagemaker.debugger.rule_configs`dan `sagemaker.debugger.Rule`.

Misalnya, untuk memeriksa keseluruhan kinerja pelatihan dan kemajuan model Anda, buat SageMaker estimator dengan konfigurasi aturan bawaan berikut.

```
from sagemaker.debugger import Rule, rule_configs

rules=[
    Rule.sagemaker(rule_configs.loss_not_decreasing()),
    Rule.sagemaker(rule_configs.overfit()),
    Rule.sagemaker(rule_configs.overtraining()),
    Rule.sagemaker(rule_configs.stalled_training_rule())
]
```

Ketika Anda memulai pekerjaan pelatihan, Debugger mengumpulkan data pemanfaatan sumber daya sistem setiap 500 milidetik dan nilai kerugian dan akurasi setiap 500 langkah secara default. Debugger menganalisis pemanfaatan sumber daya untuk mengidentifikasi apakah model Anda mengalami masalah bottleneck. The`loss_not_decreasing`,`overfit`,`overtraining`, dan `stalled_training_rule` monitor jika model Anda mengoptimalkan fungsi kerugian tanpa masalah pelatihan tersebut. Jika aturan mendeteksi anomali pelatihan, status evaluasi aturan berubah menjadi. `IssueFound` Anda dapat mengatur tindakan otomatis, seperti memberi tahu masalah pelatihan dan menghentikan pekerjaan pelatihan menggunakan Amazon CloudWatch Events danAWS Lambda. Untuk informasi selengkapnya, lihat Aksi di Amazon SageMaker Aturan Debugger.

Gunakan Aturan Bawaan Debugger dengan Nilai Parameter Kustom

Jika Anda ingin menyesuaikan nilai parameter aturan bawaan dan menyesuaikan regex koleksi tensor, konfigurasikan `base_config` dan `rule_parameters` parameter untuk dan metode kelas. `ProfilerRule.sagemaker` `Rule.sagemaker` Jika metode `Rule.sagemaker` kelas, Anda juga dapat menyesuaikan koleksi tensor melalui `collections_to_save` parameter. Instruksi bagaimana menggunakan `CollectionConfig` kelas disediakan diMengonfigurasi Koleksi Tensor Menggunakan`CollectionConfig` API.

Gunakan template konfigurasi berikut untuk aturan bawaan untuk menyesuaikan nilai parameter. Dengan mengubah parameter aturan yang Anda inginkan, Anda dapat menyesuaikan sensitivitas aturan yang akan dipicu.

- `base_config`Argumen adalah tempat Anda memanggil metode aturan bawaan.
- `rule_parameters`Argumennya adalah untuk menyesuaikan nilai kunci default dari aturan bawaan yang tercantum dalamDaftar Aturan Built-in Debugger.
- `collections_to_save`Argumen mengambil konfigurasi tensor melalui `CollectionConfig` API, yang membutuhkan `name` dan `parameters` argumen.

- Untuk menemukan koleksi tensor yang tersedianame, lihat Koleksi Tensor Bawaan Debugger.
- Untuk daftar lengkap yang dapat disesuaikanparameters, lihat Debugger CollectionConfig API.

Untuk informasi selengkapnya tentang kelas, metode, dan parameter aturan Debugger, lihat kelas Aturan SageMaker Debugger di Amazon SageMaker Python SDK.

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs, CollectionConfig

rules=[
    Rule.sagemaker(
        base_config=rule_configs.built_in_rule_name(),
        rule_parameters={
                "key": "value"
        },
        collections_to_save=[
            CollectionConfig(
                name="tensor_collection_name",
                parameters={
                    "key": "value"
                }
            )
        ]
    )
]
```

Deskripsi parameter dan contoh penyesuaian nilai disediakan untuk setiap aturan diDaftar Aturan Built-in Debugger.

Contoh Notebook dan Sampel Kode untuk Mengkonfigurasi Aturan Debugger

Pada bagian berikut, notebook dan contoh kode tentang cara menggunakan aturan Debugger untuk memantau pekerjaan SageMaker pelatihan disediakan.

Topik

- Debugger Built-in Aturan Contoh Notebook
- Debugger Built-in Aturan Contoh Kode
- Gunakan Aturan Bawaan Debugger dengan Modifikasi Parameter

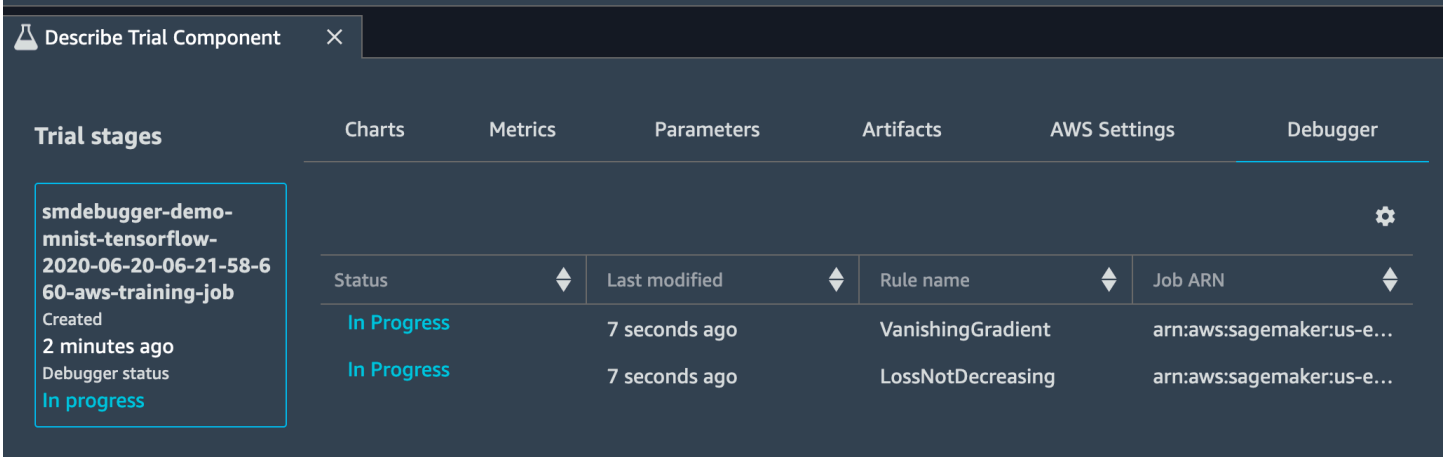
Debugger Built-in Aturan Contoh Notebook

Contoh notebook berikut menunjukkan cara menggunakan aturan bawaan Debugger saat menjalankan pekerjaan pelatihan dengan Amazon: SageMaker

- Menggunakan aturan bawaan SageMaker Debugger dengan TensorFlow
- Menggunakan aturan bawaan SageMaker Debugger dengan Managed Spot Training dan MXNet
- Menggunakan aturan bawaan SageMaker Debugger dengan modifikasi parameter untuk analisis pekerjaan pelatihan waktu nyata dengan XGBoost

Saat menjalankan contoh notebook di SageMaker Studio, Anda dapat menemukan uji coba pekerjaan pelatihan yang dibuat di tab Daftar Eksperimen Studio. Misalnya, seperti yang ditunjukkan pada gambar berikut, Anda dapat menemukan dan membuka jendela Jelaskan Komponen Uji Coba dari pekerjaan pelatihan Anda saat ini. Pada tab Debugger, Anda dapat memeriksa apakah aturan Debugger, `vanishing_gradient()` dan`loss_not_decreasing()`, memantau sesi pelatihan secara paralel. Untuk instruksi lengkap tentang cara menemukan komponen uji coba pekerjaan pelatihan Anda di UI Studio, lihat SageMakerStudio - View Experiments, Trials, dan Trial Components.

```
[29]: rules = [
          Rule.sagemaker(rule_configs.vanishing_gradient()),
          Rule.sagemaker(
              base_config=rule_configs.loss_not_decreasing(),
              collections_to_save=[
                  CollectionConfig(
                      name="losses",
                      parameters={
                            #"save_interval": "50",
                            "train.save_interval": "50",
                            "eval.save_interval": "10"}


                  )
              ]
          )
      ]


      estimator = TensorFlow(
          role=sagemaker.get_execution_role(),
          base_job_name='smdebugger-demo-mnist-tensorflow',
          train_instance_count=1,
          train_instance_type='ml.m4.xlarge',
          train_volume_size=400,
          entry_point=entrypoint_script,
          framework_version='1.15',
          py_version='py3',
          train_max_run=3600,
          script_mode=True,
          hyperparameters=hyperparameters,
          ## New parameter
          rules = rules
      )
```



Ada dua cara menggunakan aturan bawaan Debugger di SageMaker lingkungan: gunakan aturan bawaan seperti yang disiapkan atau sesuaikan parameternya sesuai keinginan. Topik berikut menunjukkan cara menggunakan aturan bawaan dengan kode contoh.

Debugger Built-in Aturan Contoh Kode

Contoh kode berikut menunjukkan bagaimana untuk mengatur Debugger built-in aturan menggunakan `Rule.sagemaker` metode. Untuk menentukan aturan bawaan yang ingin Anda jalankan, gunakan operasi `rules_configs` API untuk memanggil aturan bawaan. Untuk menemukan daftar lengkap aturan bawaan Debugger dan nilai parameter default, lihatDaftar Aturan Built-in Debugger.

```
import sagemaker
from sagemaker.tensorflow import TensorFlow
from sagemaker.debugger import Rule, CollectionConfig, rule_configs

# call built-in rules that you want to use.
built_in_rules=[
            Rule.sagemaker(rule_configs.vanishing_gradient())
            Rule.sagemaker(rule_configs.loss_not_decreasing())
]

# construct a SageMaker estimator with the Debugger built-in rules
sagemaker_estimator=TensorFlow(
    entry_point='directory/to/your_training_script.py',
    role=sm.get_execution_role(),
    base_job_name='debugger-built-in-rules-demo',
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="2.9.0",
    py_version="py39",

    # debugger-specific arguments below
    rules=built_in_rules
)
sagemaker_estimator.fit()
```

> **Note**
>
> Aturan bawaan Debugger berjalan secara paralel dengan pekerjaan pelatihan Anda. Jumlah maksimum wadah aturan bawaan untuk pekerjaan pelatihan adalah 20.

Untuk informasi selengkapnya tentang kelas, metode, dan parameter aturan Debugger, lihat kelas Aturan SageMaker Debugger di Amazon SageMaker Python SDK.

Untuk menemukan contoh cara menyesuaikan parameter aturan Debugger, lihat Gunakan Aturan Bawaan Debugger dengan Modifikasi Parameter bagian berikut.

Gunakan Aturan Bawaan Debugger dengan Modifikasi Parameter

Contoh kode berikut menunjukkan struktur built-in aturan untuk menyesuaikan parameter. Dalam contoh ini, `stalled_training_rule` mengumpulkan pengumpulan `losses` tensor dari pekerjaan pelatihan di setiap 50 langkah dan tahap evaluasi di setiap 10 langkah. Jika proses pelatihan mulai mengulur-ulur dan tidak mengumpulkan output tensor selama 120 detik, maka akan `stalled_training_rule` menghentikan pekerjaan pelatihan.

```
import sagemaker
from sagemaker.tensorflow import TensorFlow
from sagemaker.debugger import Rule, CollectionConfig, rule_configs

# call the built-in rules and modify the CollectionConfig parameters

base_job_name_prefix= 'smdebug-stalled-demo-' + str(int(time.time()))

built_in_rules_modified=[
    Rule.sagemaker(
        base_config=rule_configs.stalled_training_rule(),
        rule_parameters={
                'threshold': '120',
                'training_job_name_prefix': base_job_name_prefix,
                'stop_training_on_fire' : 'True'
        }
        collections_to_save=[
            CollectionConfig(
                name="losses",
                parameters={
                      "train.save_interval": "50"
                      "eval.save_interval": "10"
                }
            )
        ]
    )
]

# construct a SageMaker estimator with the modified Debugger built-in rule
sagemaker_estimator=TensorFlow(
    entry_point='directory/to/your_training_script.py',
    role=sm.get_execution_role(),
```

```
    base_job_name=base_job_name_prefix,
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="2.9.0",
    py_version="py39",

    # debugger-specific arguments below
    rules=built_in_rules_modified
)
sagemaker_estimator.fit()
```

Untuk konfigurasi lanjutan aturan bawaan Debugger menggunakan `CreateTrainingJob` API, lihatMengonfigurasi Debugger Menggunakan Amazon SageMaker API.

Matikan Debugger

Jika Anda ingin benar-benar mematikan Debugger, lakukan salah satu hal berikut:

- Sebelum memulai pekerjaan pelatihan, lakukan hal berikut:

  Untuk menghentikan pemantauan dan pembuatan profil, sertakan`disable_profiler` parameter ke estimator Anda dan atur ke`True`.

  > **Warning**
  >
  > Jika Anda menonaktifkannya, Anda tidak akan dapat melihat dasbor wawasan Studio Debugger yang komprehensif dan laporan profil yang dibuat secara otomatis.

  Untuk menghentikan debugging, atur`debugger_hook_config` parameternya`False`.

  > **Warning**
  >
  > Jika Anda menonaktifkannya, Anda tidak akan dapat mengumpulkan tensor keluaran dan tidak akan dapat men-debug parameter model.

  ```
  estimator=Estimator(
      ...
      disable_profiler=True
      debugger_hook_config=False
  ```

```
)
```

Untuk informasi selengkapnya tentang parameter khusus Debugger, lihat SageMaker Estimator di Amazon SageMaker Python SDK.

- Saat pekerjaan pelatihan sedang berjalan, lakukan hal berikut:

  Untuk menonaktifkan pemantauan dan pembuatan profil saat pekerjaan pelatihan Anda berjalan, gunakan metode kelas estimator berikut:

  ```
  estimator.disable_profiling()
  ```

  Untuk menonaktifkan profil kerangka kerja saja dan menjaga pemantauan sistem, gunakan`update_profiler` metode ini:

  ```
  estimator.update_profiler(disable_framework_metrics=true)
  ```

  Untuk informasi selengkapnya tentang metode ekstensi estimator, lihat estimator.disable_profiling dan estimator.update_profiler classmethods dalam dokumentasi Amazon SageMaker Python SDK.

Berguna SageMaker Estimator Classmethods untuk Debugger

Metode kelas estimator berikut berguna untuk mengakses informasi pekerjaan SageMaker pelatihan Anda dan mengambil jalur keluaran data pelatihan yang dikumpulkan oleh Debugger. Metode berikut dapat dieksekusi setelah Anda memulai pekerjaan pelatihan dengan`estimator.fit()` metode ini.

- Untuk memeriksa URI bucket S3 dasar dari pekerjaan SageMaker pelatihan:

  ```
  estimator.output_path
  ```

- Untuk memeriksa nama pekerjaan dasar pekerjaan SageMaker pelatihan:

  ```
  estimator.latest_training_job.job_name
  ```

- Untuk melihat konfigurasi operasi`CreateTrainingJob` API lengkap dari pekerjaan SageMaker pelatihan:

  ```
  estimator.latest_training_job.describe()
  ```

- Untuk memeriksa daftar lengkap aturan Debugger saat pekerjaan SageMaker pelatihan sedang berjalan:

```
estimator.latest_training_job.rule_job_summary()
```

- Untuk memeriksa URI bucket S3 tempat data parameter model (tensor keluaran) disimpan:

```
estimator.latest_job_debugger_artifacts_path()
```

- Untuk memeriksa URI bucket S3 di mana data kinerja model (metrik sistem dan kerangka kerja) disimpan:

```
estimator.latest_job_profiler_artifacts_path()
```

- Untuk memeriksa konfigurasi aturan Debugger untuk men-debug tensor keluaran:

```
estimator.debugger_rule_configs
```

- Untuk memeriksa daftar aturan Debugger untuk debugging saat pekerjaan SageMaker pelatihan sedang berjalan:

```
estimator.debugger_rules
```

- Untuk memeriksa konfigurasi aturan Debugger untuk pemantauan dan pembuatan profil sistem dan metrik kerangka kerja:

```
estimator.profiler_rule_configs
```

- Untuk memeriksa daftar aturan Debugger untuk pemantauan dan pembuatan profil saat pekerjaan SageMaker pelatihan sedang berjalan:

```
estimator.profiler_rules
```

Untuk informasi selengkapnya tentang kelas SageMaker estimator dan metodenya, lihat Estimator API di Amazon SageMaker Python SDK.

SageMaker Laporan Interaktif Debugger untuk XGBoost

Menerima laporan pelatihan yang dibuat secara otomatis oleh Debugger. Laporan Debugger memberikan wawasan tentang pekerjaan pelatihan Anda dan menyarankan rekomendasi untuk meningkatkan kinerja model Anda.

> **Note**
>
> Anda dapat mengunduh laporan Debugger saat pekerjaan pelatihan Anda berjalan atau setelah pekerjaan selesai. Selama pelatihan, Debugger secara bersamaan memperbarui laporan yang mencerminkan status evaluasi aturan saat ini. Anda dapat mengunduh laporan Debugger lengkap hanya setelah pekerjaan pelatihan selesai.

> **Important**
>
> Dalam laporan tersebut, plot dan dan rekomendasi disediakan untuk tujuan informasi dan tidak pasti. Anda bertanggung jawab untuk membuat penilaian independen Anda sendiri terhadap informasi tersebut.

SageMaker Laporan Pelatihan Debugger XGBoost

Untuk pekerjaan pelatihan SageMaker XGBoost, gunakanCreateXgboostReport aturan Debugger untuk menerima laporan pelatihan komprehensif tentang kemajuan dan hasil pelatihan. Mengikuti panduan ini, tentukanCreateXgboostReport aturan saat membuat estimator XGBoost, unduh laporan menggunakan Amazon SageMaker Python SDK atau konsol Amazon S3, dan dapatkan wawasan tentang hasil pelatihan.

> **Important**
>
> Dalam laporan tersebut, plot dan dan rekomendasi disediakan untuk tujuan informasi dan tidak pasti. Anda bertanggung jawab untuk membuat penilaian independen Anda sendiri terhadap informasi tersebut.

Topik

- Buat Estimator SageMaker XGBoost dengan Aturan Laporan XGBoost Debugger

- Unduh Laporan Pelatihan XGBoost Debugger
- Panduan Laporan Pelatihan XGBoost Debugger

Buat Estimator SageMaker XGBoost dengan Aturan Laporan XGBoost Debugger

CreateXgboostReportAturan ini mengumpulkan tensor keluaran berikut dari pekerjaan pelatihan Anda:

- `hyperparameters`- Menyimpan pada langkah pertama.
- `metrics`- Menghemat kerugian dan akurasi setiap 5 langkah.
- `feature_importance`- Menyimpan setiap 5 langkah.
- `predictions`- Menyimpan setiap 5 langkah.
- `labels`- Menyimpan setiap 5 langkah.

Tensor keluaran disimpan di bucket S3 default. Sebagai contoh, `s3://sagemaker-<region>-<12digit_account_id>/<base-job-name>/debug-output/`.

Saat Anda membuat SageMaker estimator untuk pekerjaan pelatihan XGBoost, tentukan aturan seperti yang ditunjukkan dalam kode contoh berikut.

Using the SageMaker generic estimator

```
import boto3
import sagemaker
from sagemaker.estimator import Estimator
from sagemaker import image_uris
from sagemaker.debugger import Rule, rule_configs

rules=[
    Rule.sagemaker(rule_configs.create_xgboost_report())
]

region = boto3.Session().region_name
xgboost_container=sagemaker.image_uris.retrieve("xgboost", region, "1.2-1")

estimator=Estimator(
    role=sagemaker.get_execution_role()
    image_uri=xgboost_container,
    base_job_name="debugger-xgboost-report-demo",
```

```
    instance_count=1,
    instance_type="ml.m5.2xlarge",

    # Add the Debugger XGBoost report rule
    rules=rules
)

estimator.fit(wait=False)
```

Unduh Laporan Pelatihan XGBoost Debugger

Unduh laporan pelatihan Debugger XGBoost saat pekerjaan pelatihan Anda berjalan atau setelah pekerjaan selesai menggunakan Amazon SageMaker Python SDK danAWS Command Line Interface (CLI).

Download using the SageMaker Python SDK and AWS CLI

1. Periksa URI basis keluaran S3 default pekerjaan saat ini.

   ```
   estimator.output_path
   ```

2. Periksa nama pekerjaan saat ini.

   ```
   estimator.latest_training_job.job_name
   ```

3. Laporan Debugger XGBoost disimpan di bawah`<default-s3-output-base-uri>/<training-job-name>/rule-output`. Konfigurasikan jalur keluaran aturan sebagai berikut:

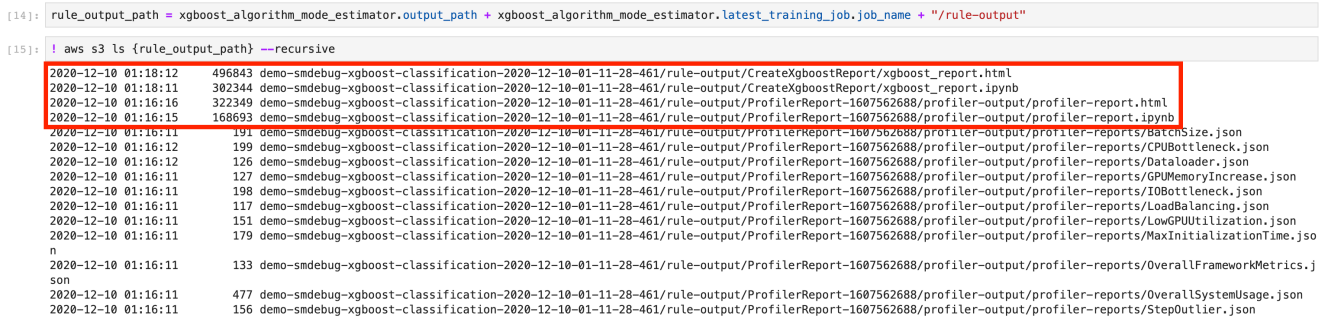
   ```
   rule_output_path = estimator.output_path + "/" +
    estimator.latest_training_job.job_name + "/rule-output"
   ```

4. Untuk memeriksa apakah laporan dihasilkan, daftar direktori dan file secara rekursif di bawah`rule_output_path` menggunakan`aws s3 ls` dengan`--recursive` opsi.

   ```
   ! aws s3 ls {rule_output_path} --recursive
   ```

   Ini harus mengembalikan daftar lengkap file di bawah folder autogenerated yang diberi nama`CreateXgboostReport` dan`ProfilerReport-1234567890`. Laporan pelatihan XGBoost disimpan di`CreateXgboostReport`, dan laporan pembuatan profil disimpan

dalam`ProfilerReport-1234567890` folder. Untuk mempelajari lebih lanjut tentang laporan pembuatan profil yang dihasilkan secara default dengan pekerjaan pelatihan XGBoost, lihatSageMaker Laporan profil debugger.



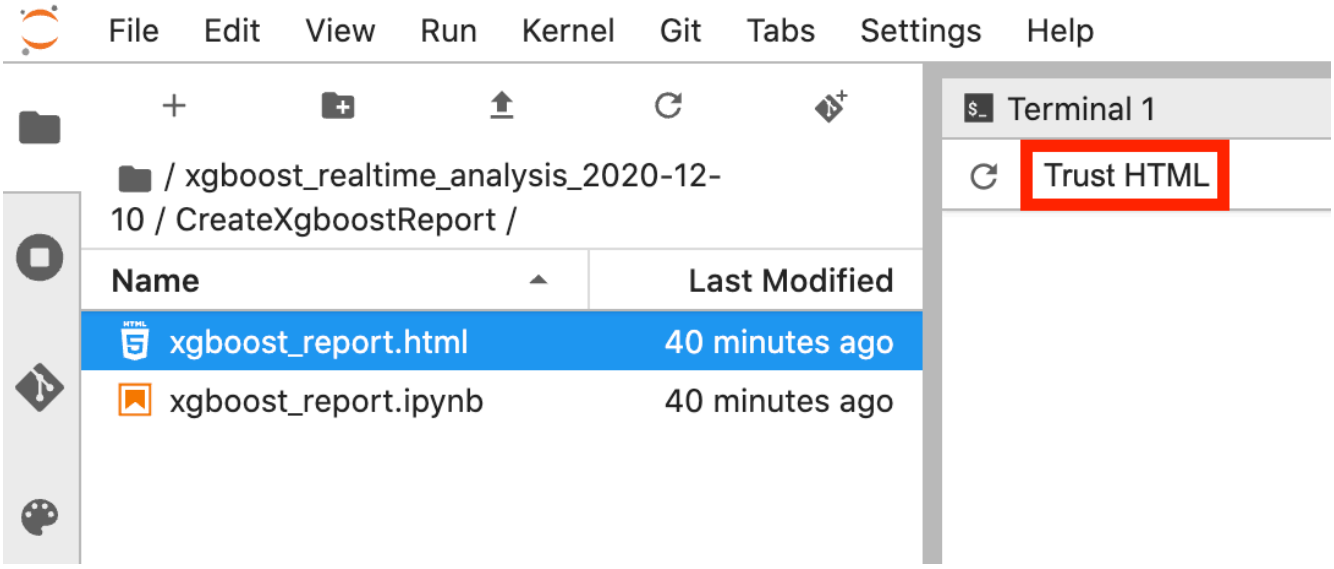
`xgboost_report.html`Ini adalah laporan pelatihan XGBoost autogenerated oleh Debugger. `xgboost_report.ipynb`Ini adalah buku catatan Jupyter yang digunakan untuk mengumpulkan hasil pelatihan ke dalam laporan. Anda dapat mengunduh semua file, menelusuri file laporan HTML, dan memodifikasi laporan menggunakan notebook.

5. Unduh file secara rekursif menggunakan`aws s3 cp`. Perintah berikut menyimpan semua file keluaran aturan ke`ProfilerReport-1234567890` folder di bawah direktori kerja saat ini.

```
! aws s3 cp {rule_output_path} ./ --recursive
```

> **Tip**
>
> Jika Anda menggunakan server notebook Jupyter, jalankan`!pwd` untuk memverifikasi direktori kerja saat ini.

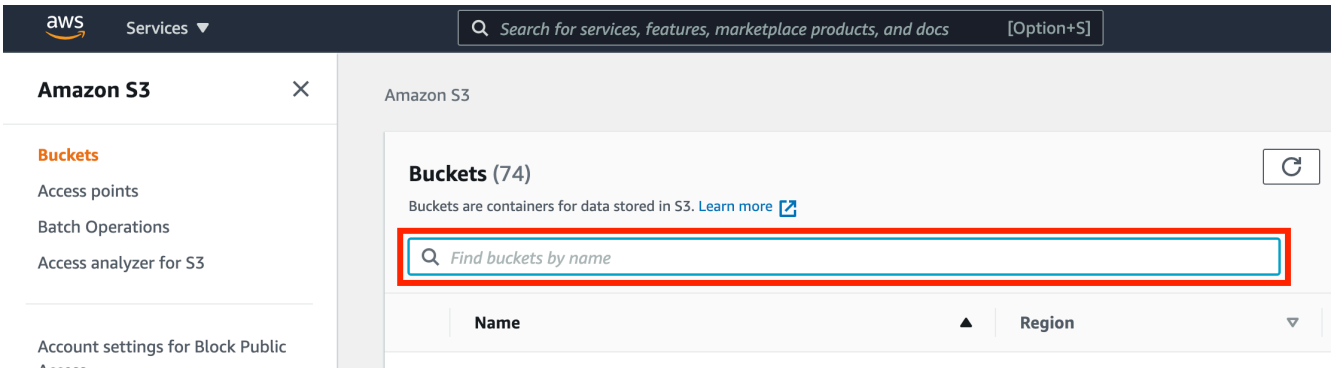
6. Di bawah`/CreateXgboostReport` direktori, buka`xgboost_report.html`. Jika Anda menggunakan JupyterLab, pilih Trust HTML untuk melihat laporan pelatihan Debugger yang dibuat secara otomatis.



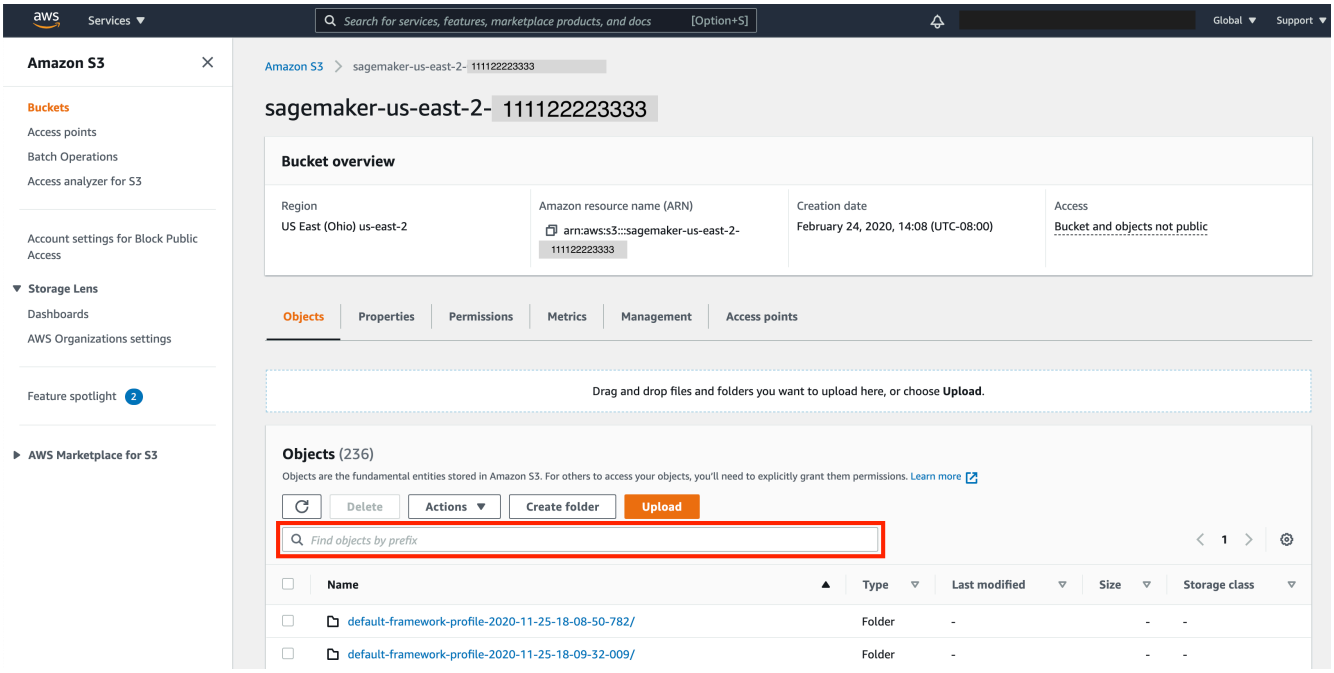
7. Buka `xgboost_report.ipynb` file untuk mengeksplorasi bagaimana laporan dihasilkan. Anda dapat menyesuaikan dan memperluas laporan pelatihan menggunakan file notebook Jupyter.

Download using the Amazon S3 console

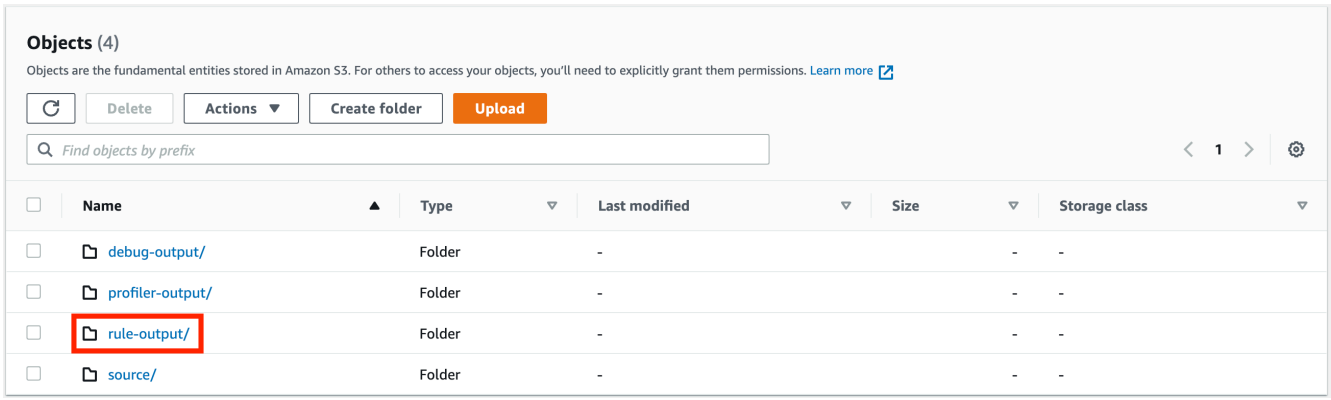
1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Cari bucket S3 dasar. Misalnya, jika Anda belum menentukan nama pekerjaan dasar apa pun, nama bucket S3 dasar harus dalam format berikut: `sagemaker-<region>-111122223333`. Cari bucket S3 dasar melalui kolom Find bucket by name.



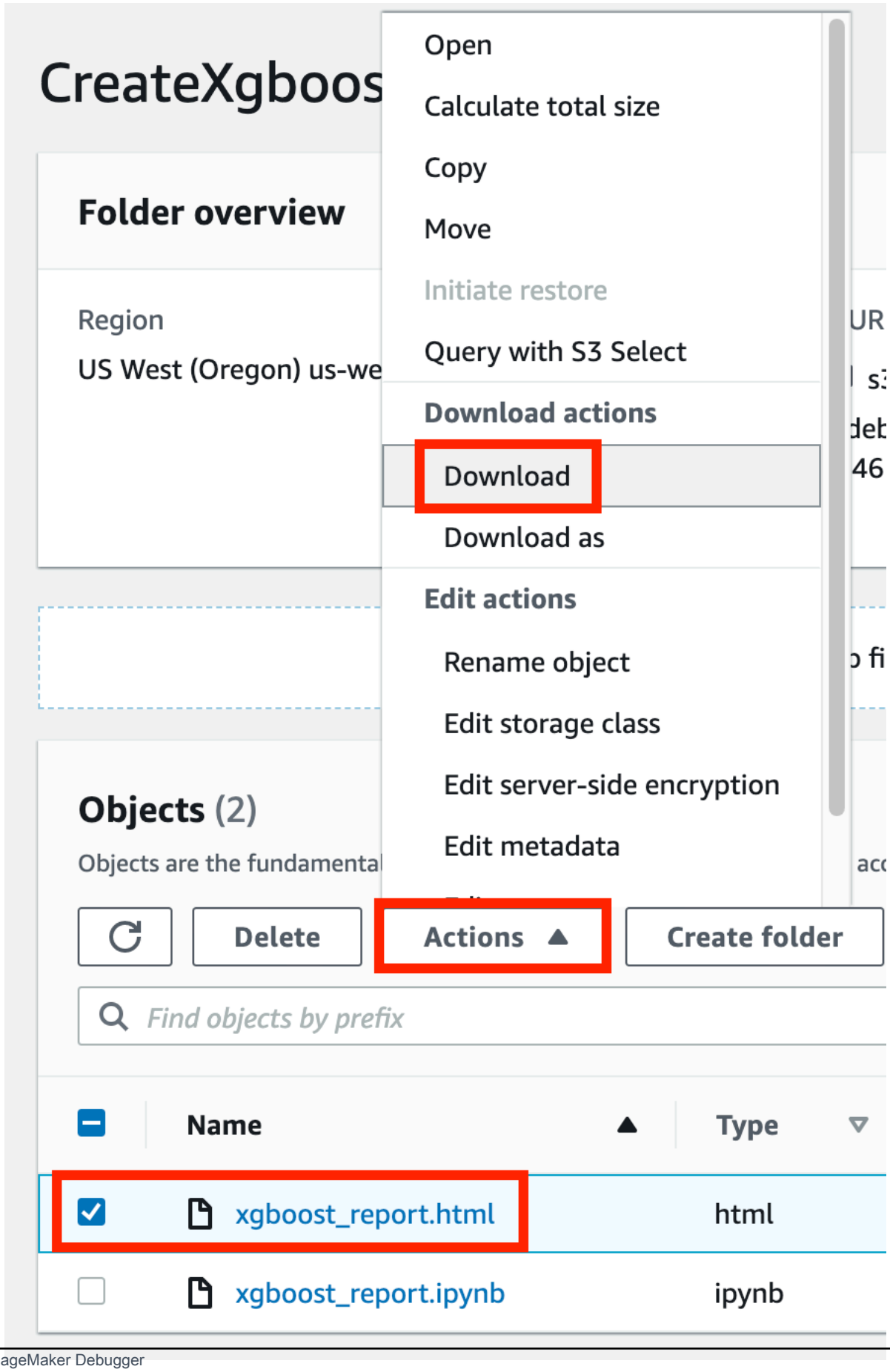
3. Di bucket S3 dasar, cari nama pekerjaan pelatihan dengan memasukkan awalan nama pekerjaan Anda di Cari objek dengan awalan dan kemudian pilih nama pekerjaan pelatihan.



4. Di bucket S3 pekerjaan pelatihan, pilih rule-output/ subfolder. Harus ada tiga subfolder untuk data pelatihan yang dikumpulkan oleh Debugger: debug-output/, profiler-output/, dan rule-output/.



5. Di folder rule-output/, pilih folder CreateXgboostReport/. Folder berisi xbgoost_report.html (laporan autogenerated dalam html) dan xbgoost_report.ipynb (notebook Jupyter dengan skrip yang digunakan untuk menghasilkan laporan).
6. Pilih file xbgoost_report.html, pilih Unduh tindakan, lalu pilih Unduh.


CreateXgboos
Folder overview
Region
US West (Oregon) us-we
Open
Calculate total size
Copy
Move
Initiate restore
Query with S3 Select
Download actions
Download
Download as
Edit actions
Rename object
Edit storage class
Edit server-side encryption
Edit metadata
Objects (2)
Objects are the fundamental
Delete
Actions
Create folder
Find objects by prefix
Name
Type
xgboost_report.html
html
xgboost_report.ipynb
ipynb

7. Buka file xbgoost_report.html yang diunduh di browser web.

Panduan Laporan Pelatihan XGBoost Debugger

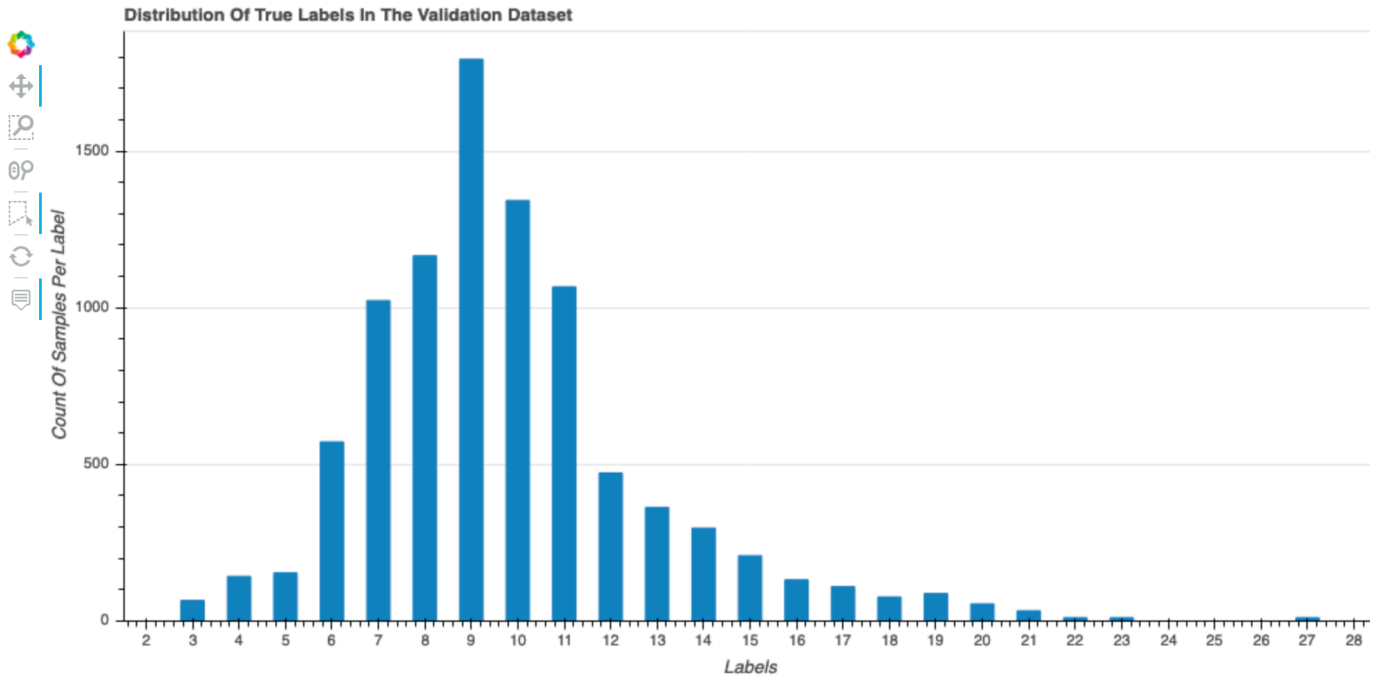
Bagian ini memandu Anda melalui laporan pelatihan Debugger XGBoost. Laporan ini dikumpulkan secara otomatis tergantung pada regex tensor keluaran, mengenali jenis pekerjaan pelatihan Anda di antara klasifikasi biner, klasifikasi multiclass, dan regresi.

> **Important**
>
> Dalam laporan tersebut, plot dan dan rekomendasi disediakan untuk tujuan informasi dan tidak pasti. Anda bertanggung jawab untuk membuat penilaian independen Anda sendiri terhadap informasi tersebut.

Topik

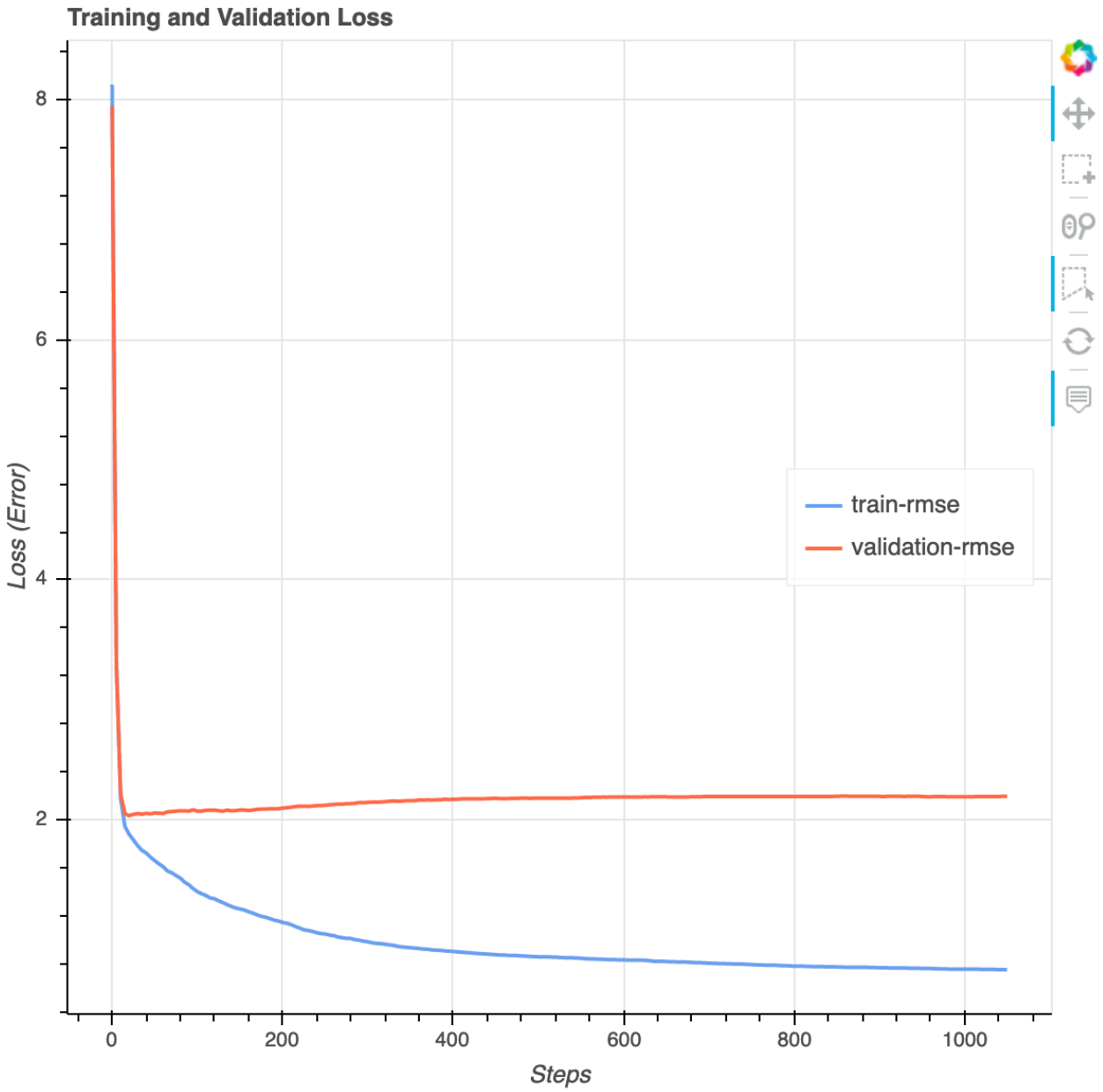
Distribusi Label Sejati dari Dataset

Histogram ini menunjukkan distribusi kelas berlabel (untuk klasifikasi) atau nilai (untuk regresi) dalam kumpulan data asli Anda. Kecenderungan dalam kumpulan data Anda dapat berkontribusi pada ketidakakuratan. Visualisasi ini tersedia untuk jenis model berikut: klasifikasi biner, multiclassification, dan regresi.



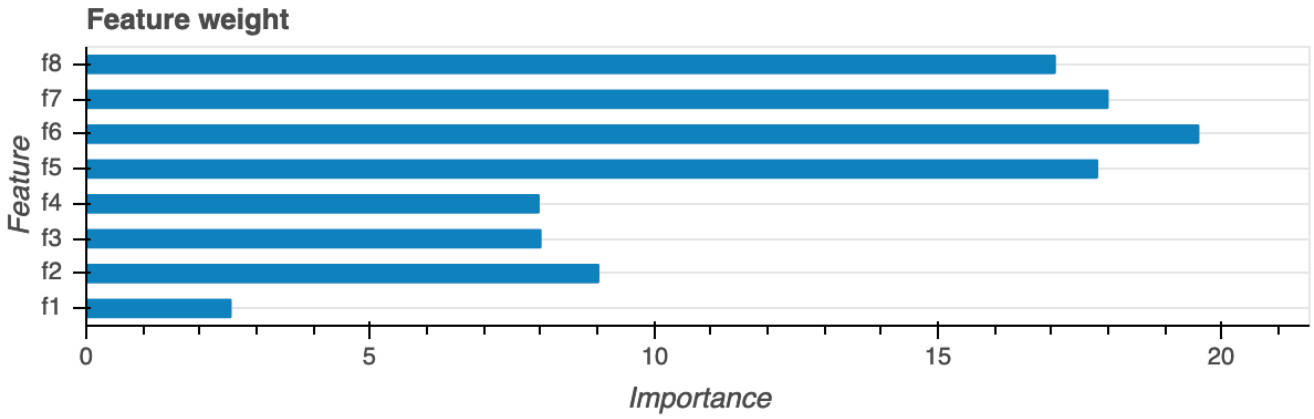
Kerugian versus Grafik Langkah

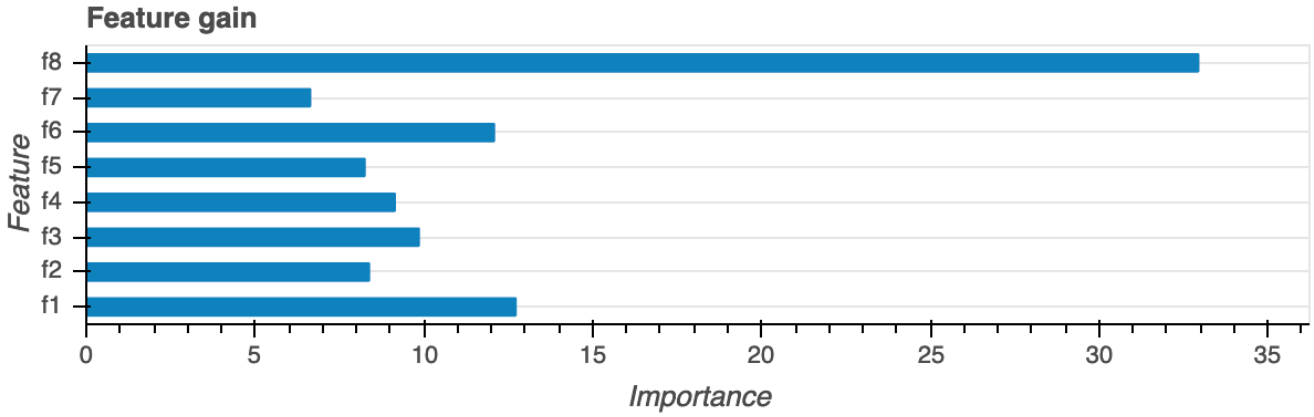
Ini adalah diagram garis yang menunjukkan perkembangan kehilangan data pelatihan dan data validasi selama langkah-langkah pelatihan. Kerugian adalah apa yang Anda definisikan dalam fungsi tujuan Anda, seperti mean squared error. Anda dapat mengukur apakah modelnya overfit atau underfit dari plot ini. Bagian ini menyediakan wawasan yang dapat digunakan untuk menentukan bagaimana menyelesaikan masalah yang terlalu cocok dan kurang cocok. Visualisasi ini tersedia untuk jenis model berikut: klasifikasi biner, multiclassification, dan regresi.

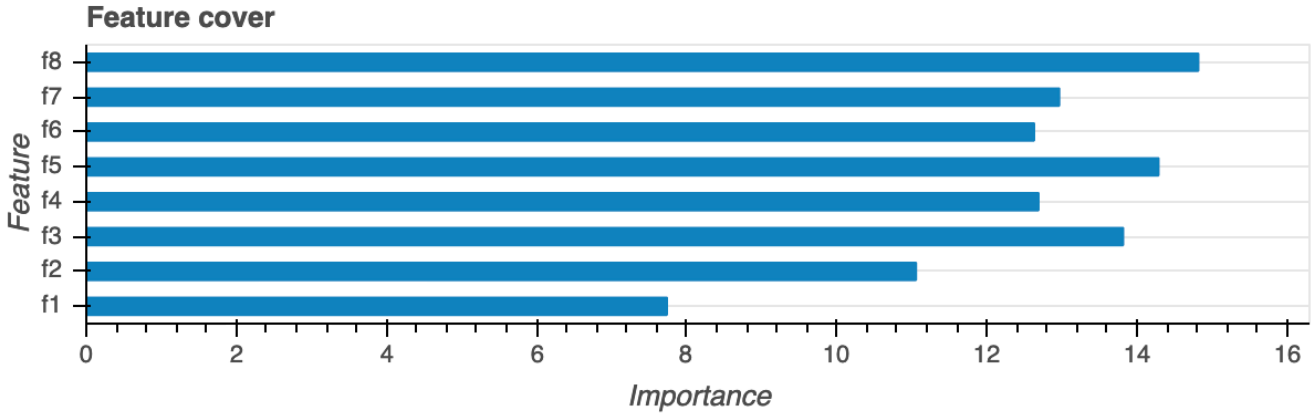


Pentingnya Fitur

Ada tiga jenis visualisasi penting fitur yang disediakan: Berat, Keuntungan, dan Cakupan. Kami memberikan definisi rinci untuk masing-masing dari tiga dalam laporan. Visualisasi pentingnya fitur membantu Anda mempelajari fitur apa saja dalam kumpulan data latihan yang berkontribusi pada prediksi. Visualisasi pentingnya fitur tersedia untuk jenis model berikut: klasifikasi biner, multiclassification, dan regresi.



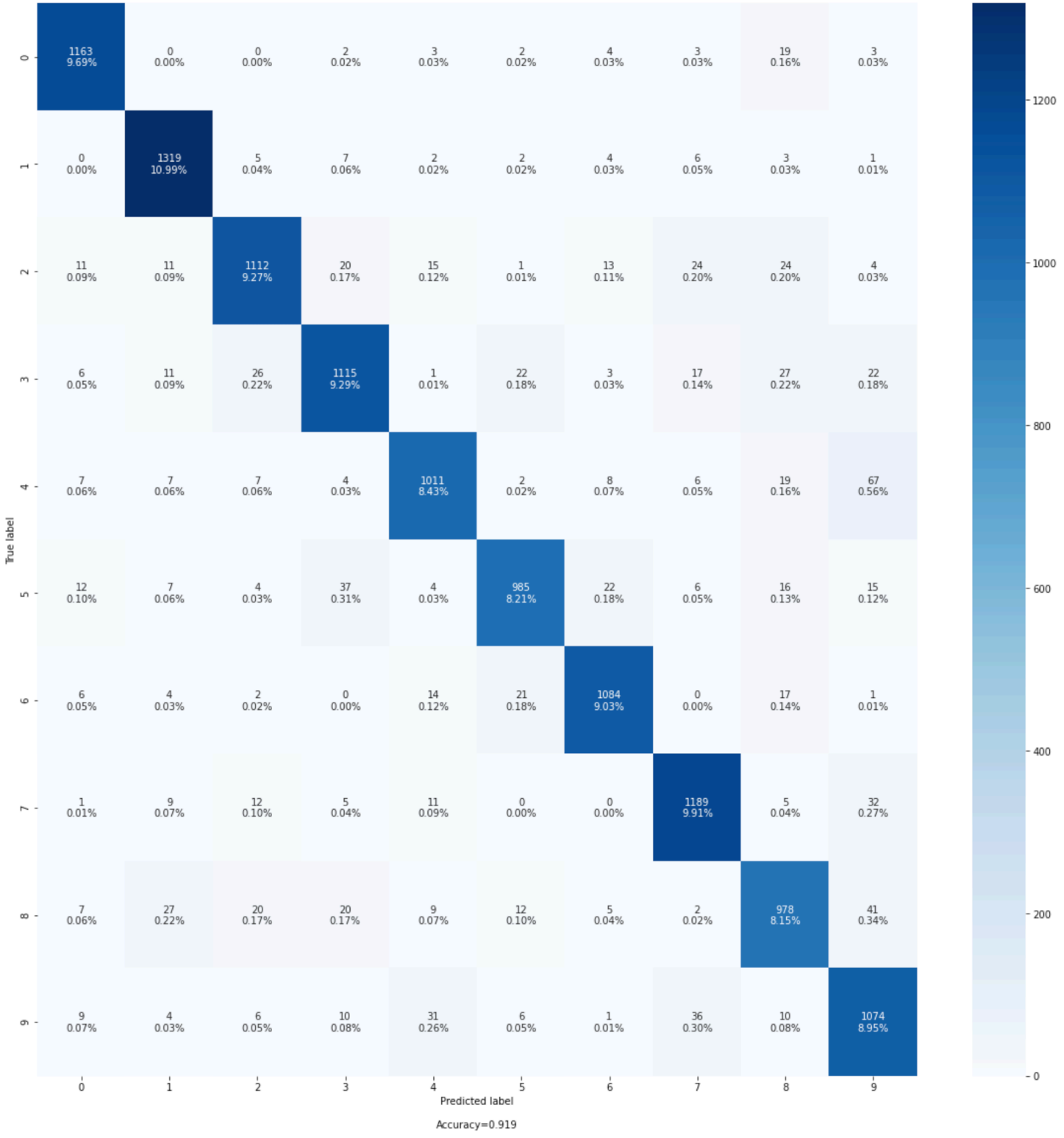




Matriks Kebingungan

Visualisasi ini hanya berlaku untuk model klasifikasi biner dan multiclass. Akurasi saja mungkin tidak cukup untuk mengevaluasi kinerja model. Untuk beberapa kasus penggunaan, seperti perawatan kesehatan dan deteksi penipuan, penting juga untuk mengetahui tingkat positif palsu dan tingkat

negatif palsu. Matriks kebingungan memberi Anda dimensi tambahan untuk mengevaluasi kinerja model Anda.



Evaluasi Matriks Kebingungan

Bagian ini memberi Anda lebih banyak wawasan tentang metrik mikro, makro, dan tertimbang tentang presisi, penarikan, dan skor F1 untuk model Anda.

Overall Accuracy

```
    Overall Accuracy: 0.919
```

Micro Performance Metrics

```
    Performance metrics calculated globally by counting the total true positives, false negatives, and false positive
s.

    Micro Precision: 0.919
    Micro Recall: 0.919
    Micro F1-score: 0.919
```

Macro Performance Metrics

```
    Performance metrics calculated for each label, and find their unweighted mean.
    This does not take the class imbalance problem into account.

    Macro Precision: 0.919
    Macro Recall: 0.918
    Macro F1-score: 0.918
```

Weighted Performance Metrics

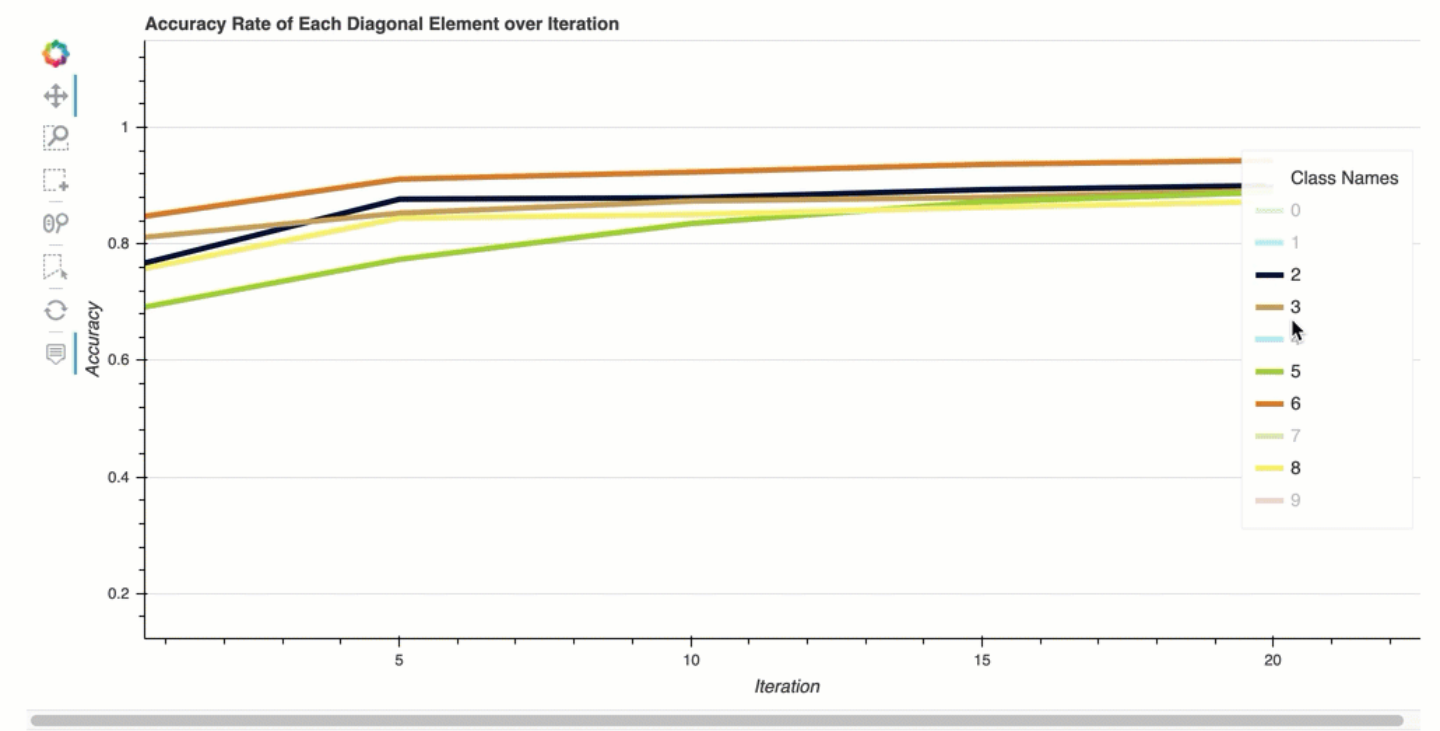
```
    Performance metrics calculated for each label and their average weighted by support
    (the number of true instances for each label).
    This extends the macro option to take the class imbalance into account.
    It might result in an F-score that is not between precision and recall.

    Weighted Precision: 0.92
    Weighted Recall: 0.919
    Weighted F1-score: 0.919
```

Classification Report

```
        The summary of the precision, recall, and F1-score for each class.

              precision    recall  f1-score   support

         0.0       0.95      0.97      0.96      1199
         1.0       0.94      0.98      0.96      1349
         2.0       0.93      0.90      0.92      1235
         3.0       0.91      0.89      0.90      1250
         4.0       0.92      0.89      0.90      1138
         5.0       0.94      0.89      0.91      1108
         6.0       0.95      0.94      0.95      1149
         7.0       0.92      0.94      0.93      1264
         8.0       0.87      0.87      0.87      1121
         9.0       0.85      0.90      0.88      1187

    accuracy                           0.92     12000
   macro avg       0.92      0.92      0.92     12000
weighted avg       0.92      0.92      0.92     12000
```

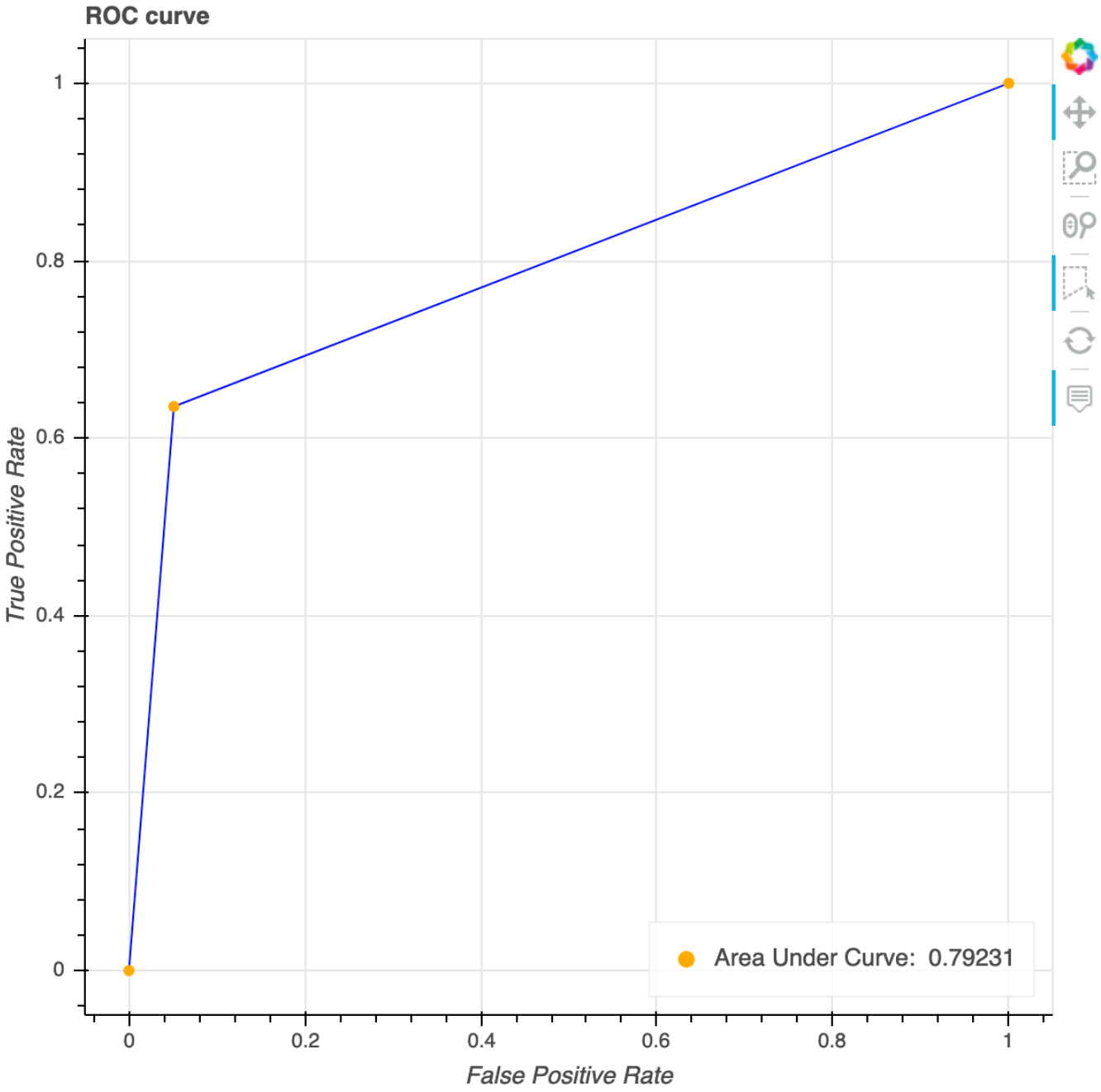
Tingkat Akurasi Setiap Elemen Diagonal Selama Iterasi

Visualisasi ini hanya berlaku untuk klasifikasi biner dan model klasifikasi multiclass. Ini adalah bagan garis yang memplot nilai diagonal dalam matriks kebingungan di seluruh langkah pelatihan untuk setiap kelas. Plot ini menunjukkan kepada Anda bagaimana keakuratan setiap kelas berlangsung selama langkah-langkah pelatihan. Anda dapat mengidentifikasi kelas yang kurang berkinerja dari plot ini.



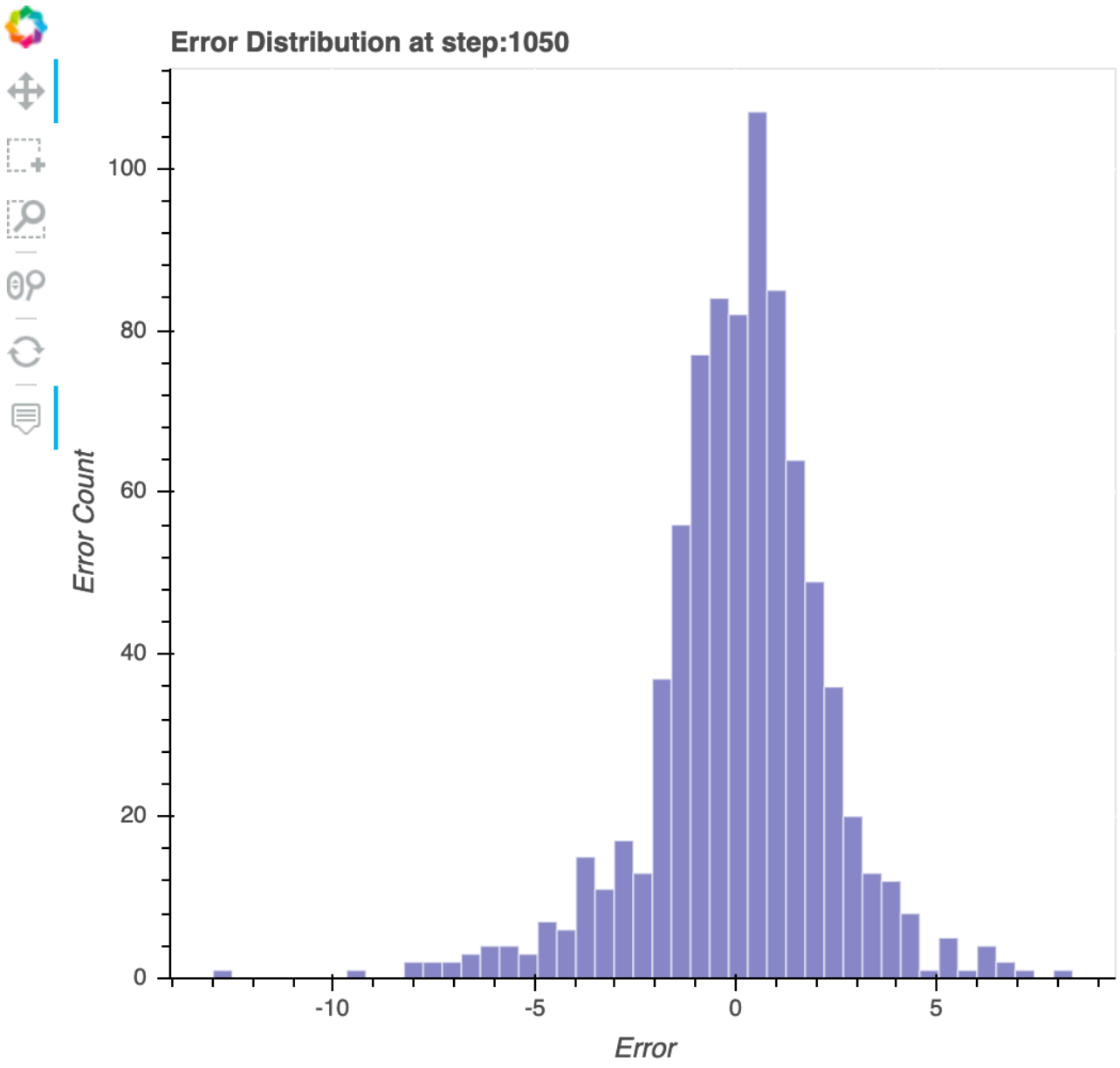
Kurva Karakteristik Operasi Penerima

Visualisasi ini hanya berlaku untuk model klasifikasi biner. Kurva Karakteristik Operasi Penerima biasanya digunakan untuk mengevaluasi kinerja model klasifikasi biner. Sumbu y dari kurva adalah True Positive Rate (TPF) dan sumbu x adalah tingkat positif palsu (FPR). Plot juga menampilkan nilai untuk area di bawah kurva (AUC). Semakin tinggi nilai AUC, semakin prediktif classifier Anda. Anda juga dapat menggunakan kurva ROC untuk memahami trade-off antara TPR dan FPR dan mengidentifikasi ambang klasifikasi optimal untuk kasus penggunaan Anda. Ambang klasifikasi dapat disesuaikan untuk menyetel perilaku model untuk mengurangi lebih dari satu atau jenis kesalahan lainnya (FP/FN).



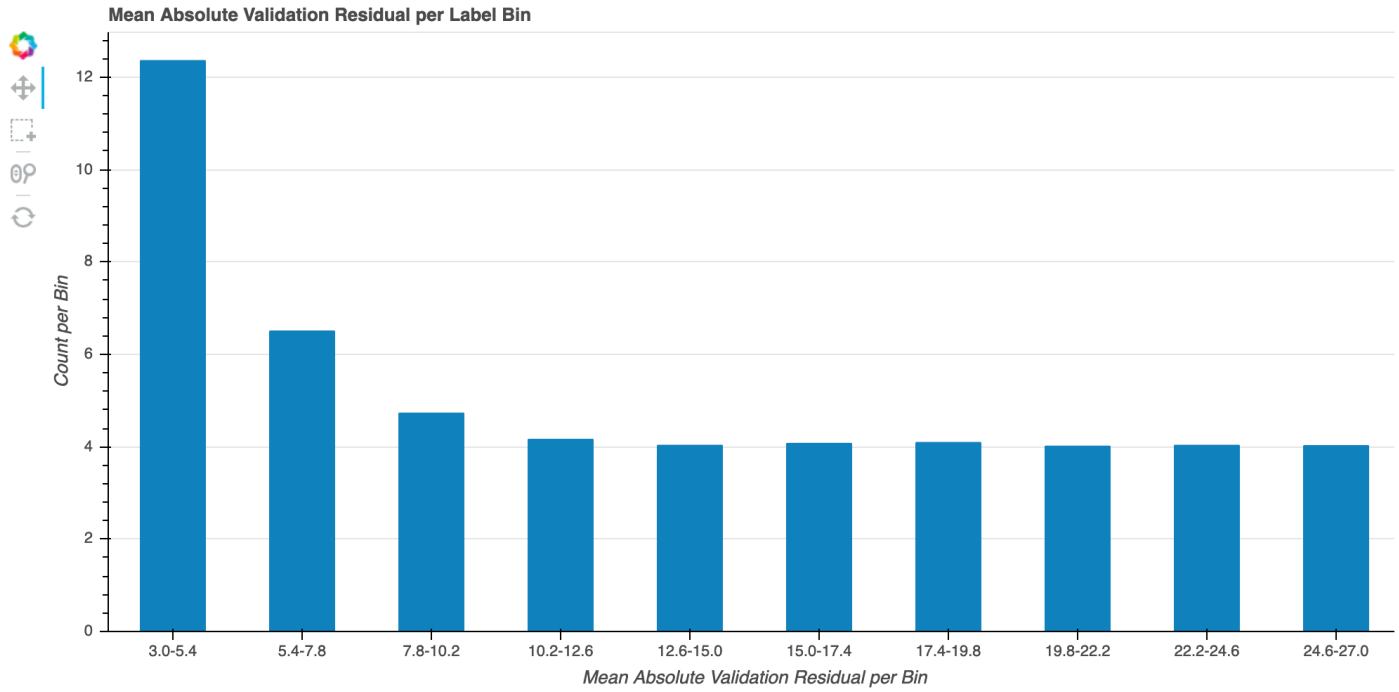
Distribusi Residu pada Langkah Tersimpan Terakhir

Visualisasi ini adalah bagan kolom yang menunjukkan distribusi sisa pada langkah terakhir Debugger menangkap. Dalam visualisasi ini, Anda dapat memeriksa apakah distribusi residu mendekati distribusi normal yang berpusat pada nol. Jika residu miring, fitur Anda mungkin tidak cukup untuk memprediksi label.



Absolute Validasi Kesalahan per Label Bin Selama Iterasi

Visualisasi ini hanya berlaku untuk model regresi. Nilai target sebenarnya dibagi menjadi 10 interval. Visualisasi ini menunjukkan bagaimana kemajuan kesalahan validasi untuk setiap interval di seluruh langkah pelatihan dalam plot baris. Kesalahan validasi absolut adalah nilai absolut perbedaan antara prediksi dan aktual selama validasi. Anda dapat mengidentifikasi interval berkinerja buruk dari visualisasi ini.



Aksi di Amazon SageMaker Aturan Debugger

Berdasarkan status Evaluasi aturan Debugger, Anda dapat mengatur tindakan otomatis seperti menghentikan tugas pelatihan dan mengirimkan notifikasi menggunakan Amazon Simple Notification Service (Amazon SNS). Anda juga dapat membuat tindakan Anda sendiri menggunakan Amazon CloudWatch Acara danAWS Lambda. Untuk mempelajari cara mengatur tindakan otomatis berdasarkan status evaluasi aturan Debugger, lihat topik berikut.

Topik

- Tindakan bawaan Debugger untuk Aturan
- Buat Tindakan pada Aturan Menggunakan Amazon CloudWatch danAWS Lambda

Tindakan bawaan Debugger untuk Aturan

Gunakan tindakan bawaan Debugger untuk menanggapi masalah yang ditemukan olehAturan Debugger. Debugger`rule_configs`class menyediakan alat untuk mengonfigurasi daftar tindakan, termasuk menghentikan tugas pelatihan secara otomatis dan mengirim notifikasi menggunakan Amazon Simple Notification Service (Amazon SNS) saat aturan Debugger menemukan masalah pelatihan.

Langkah 1: Siapkan Amazon SNS, Buat SMDebugRules Topik, dan Berlangganan Topik

Bagian ini memandu Anda melalui cara mengatur Amazon SNS**`SMDebugRules`**topik, berlangganan, dan konfirmasikan langganan untuk menerima pemberitahuan dari aturan Debugger.

> **Note**
>
> Untuk informasi selengkapnya tentang penagihan untuk Amazon SNS, lihatHarga Amazon SNSdanAmazon SNS FAQ.

Untuk membuat SMDebugRules tema

1. Masuk ke AWS Management Console dan buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Di panel navigasi kiri, pilih Topics (Topik).
3. Di halaman Topics (Topik), pilih Create topic (Buat topik).
4. Di halaman Create topic (Buat topik), di bagian Details (Detail), lakukan hal-hal berikut:
   a. UntukTipe, pilihStandaruntuk jenis topik.
   b. Di Nama, masukkan **`SMDebugRules`**.
5. Lewati semua pengaturan opsional lainnya dan pilihBuat topik. Jika Anda ingin mempelajari lebih lanjut tentang pengaturan opsional, lihatMembuat topik Amazon SNS.

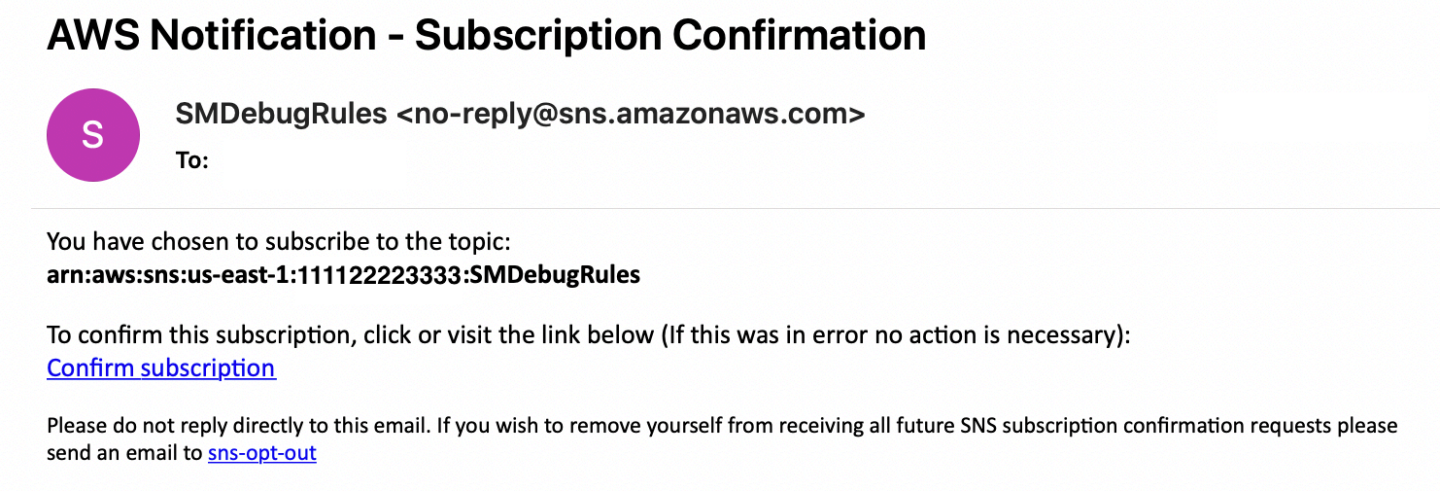
Untuk berlangganan SMDebugRules tema

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Di panel navigasi kiri, pilih Subscriptions (Langganan).
3. Di halaman Subscriptions (Langganan), pilih Create subscription (Buat langganan).
4. Di halaman Create subscription (Buat langganan), di bagian Details (Detail), lakukan:
   a. UntukTopik ARN, pilihSMDebugRulestopik ARN. ARN harus dalam format`arn:aws:sns:<region-id>:111122223333:SMDebugRules`.
   b. UntukProtokol, pilihEmailatauSMS.
   c. UntukTitik akhir, masukkan nilai titik akhir, seperti alamat email atau nomor telepon yang ingin Anda terima notifikasi.

> **Note**
>
> Pastikan Anda mengetik alamat email dan nomor telepon yang benar. Nomor telepon harus menyertakan+, kode negara, dan nomor telepon, tanpa karakter atau spasi khusus. Misalnya, nomor telepon +1 (222) 333-4444 diformat sebagai**+12223334444**.

5. Lewati semua pengaturan opsional lainnya dan pilihBuat langganan. Jika Anda ingin mempelajari lebih lanjut tentang pengaturan opsional, lihatBerlangganan topik Amazon SNS.

Setelah Anda berlangganSMDebugRulestopik, Anda menerima pesan konfirmasi berikut di email atau melalui telepon:



Untuk informasi selengkapnya tentang Amazon SNS, lihatOlahpesan teks seluler (SMS)danPemberitahuan emaildiPanduan Pengembang Amazon SNS.

Langkah 2: Siapkan Peran IAM Anda untuk Melampirkan Kebijakan yang Diperlukan

Di langkah ini, Anda dapat menambahkan kebijakan yang diperlukan ke peran IAM Anda.

Untuk menambahkan kebijakan yang diperlukan ke peran IAM Anda

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di bagian Navigasi di bagian kiri, pilihKebijakan, dan pilihBuat kebijakan.
3. PadaBuat kebijakanhalaman, lakukan hal berikut untuk membuat kebijakan sns-access baru:

a. Pilih tab JSON.

b. Tempel string JSON yang diformat dengan huruf tebal di kode berikut ke`"Statement"`, menggantikan 12 digitAWSID akun dengan AndaAWSID akun.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "sns:Publish",
                "sns:CreateTopic",
                "sns:Subscribe"
            ],
            "Resource": "arn:aws:sns:*:111122223333:SMDebugRules"
        }
    ]
}
```

c. Di bagian bawah halaman, pilihKebijakan peninjauan.

d. Pada halaman Tinjau kebijakan, untuk Nama, masukkan **`sns-access`**.

e. Di bagian bawah halaman, pilihBuat kebijakan.

4. Kembali ke konsol IAM, dan pilihPerandi bagian Navigasi di bagian kiri.

5. Cari peran IAM yang Anda gunakan SageMaker pelatihan model dan pilih peran IAM itu.

6. PadaIzintab dariRingkasanhalaman, pilihLampirkan kebijakan.

7. Cari yangsns-aksesPolis, pilih kotak di samping Polis, lalu pilihLampirkan kebijakan.

Untuk lebih banyak contoh pengaturan kebijakan IAM untuk Amazon SNS, lihatContoh kasus untuk pengendalian akses Amazon SNS.

Langkah 3: Konfigurasikan Aturan Debugger dengan Tindakan Bawaan

Setelah berhasil menyelesaikan pengaturan yang diperlukan pada langkah-langkah sebelumnya, Anda dapat mengonfigurasi tindakan bawaan Debugger untuk aturan debugging seperti yang ditunjukkan pada contoh skrip berikut. Anda dapat memilih tindakan bawaan mana yang akan digunakan saat membangun`actions`daftar objek. The`rule_configs`adalah modul pembantu

yang menyediakan alat tingkat tinggi untuk mengonfigurasi aturan dan tindakan bawaan Debugger. Tindakan bawaan berikut tersedia untuk Debugger:

- `rule_configs.StopTraining()`— Menghentikan pekerjaan pelatihan ketika aturan Debugger menemukan masalah.
- `rule_configs.Email("abc@abc.com")`— Mengirim pemberitahuan melalui email ketika aturan Debugger menemukan masalah. Gunakan alamat email yang Anda gunakan saat Anda mengatur langganan SNS.
- `rule_configs.SMS("+1234567890")`— Mengirim pemberitahuan melalui pesan teks ketika aturan Debugger menemukan masalah. Gunakan nomor telepon yang Anda gunakan saat mengatur langganan topik SNS Anda.

> **Note**
>
> Pastikan Anda mengetik alamat email dan nomor telepon yang benar. Nomor telepon harus menyertakan+, kode negara, dan nomor telepon, tanpa karakter atau spasi khusus. Misalnya, nomor telepon +1 (222) 333-4444 diformat sebagai**+12223334444**.

Anda dapat menggunakan semua tindakan bawaan atau subset tindakan dengan membungkusnya menggunakan`rule_configs.ActionList()`metode, yang mengambil tindakan bawaan dan mengkonfigurasi daftar tindakan.

Untuk menambahkan ketiga tindakan bawaan ke satu aturan

Jika Anda ingin menetapkan ketiga tindakan bawaan ke satu aturan, konfigurasikan daftar tindakan bawaan Debugger saat membuat estimator. Gunakan template berikut untuk membuat estimator, dan Debugger akan menghentikan pekerjaan pelatihan dan mengirim pemberitahuan melalui email dan teks untuk aturan apa pun yang Anda gunakan untuk memantau kemajuan pekerjaan pelatihan Anda.

```
from sagemaker.debugger import Rule, rule_configs

# Configure an action list object for Debugger rules
actions = rule_configs.ActionList(
    rule_configs.StopTraining(),
    rule_configs.Email("abc@abc.com"),
    rule_configs.SMS("+1234567890")
)
```

```
# Configure rules for debugging with the actions parameter
rules = [
    Rule.sagemaker(
        base_config=rule_configs.built_in_rule(),         # Required
        rule_parameters={"paramter_key": value },        # Optional
        actions=actions
    )
]

estimator = Estimator(
    ...
    rules = rules
)

estimator.fit(wait=False)
```

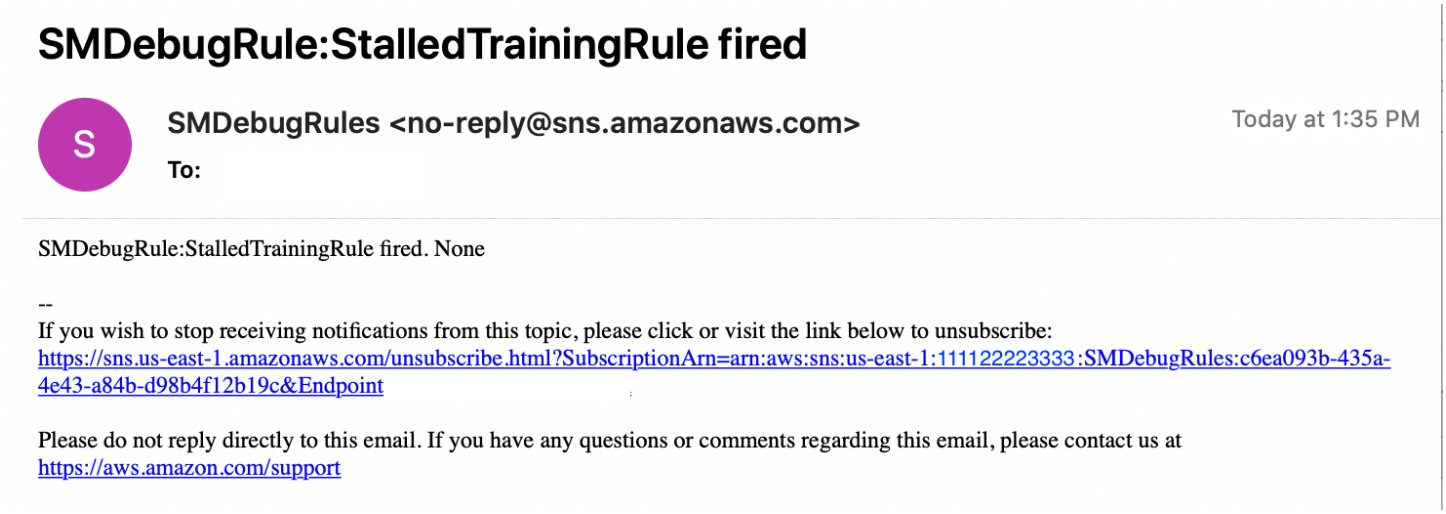
Untuk membuat beberapa objek aksi bawaan untuk menetapkan tindakan yang berbeda ke satu aturan

Jika Anda ingin menetapkan tindakan bawaan yang akan dipicu pada nilai ambang batas yang berbeda dari satu aturan, Anda dapat membuat beberapa objek tindakan bawaan seperti yang ditunjukkan pada skrip berikut. Untuk menghindari kesalahan konflik dengan menjalankan aturan yang sama, Anda harus mengirimkan nama pekerjaan aturan yang berbeda (tentukan string yang berbeda untuk aturan`name`atribut) seperti yang ditunjukkan di contoh berikut script template. Contoh ini menunjukkan cara mengaturStalledTrainingRuleuntuk mengambil dua tindakan berbeda: kirim email ke`abc@abc.com`ketika pekerjaan pelatihan berhenti selama 60 detik, dan hentikan pekerjaan pelatihan jika berhenti selama 120 detik.

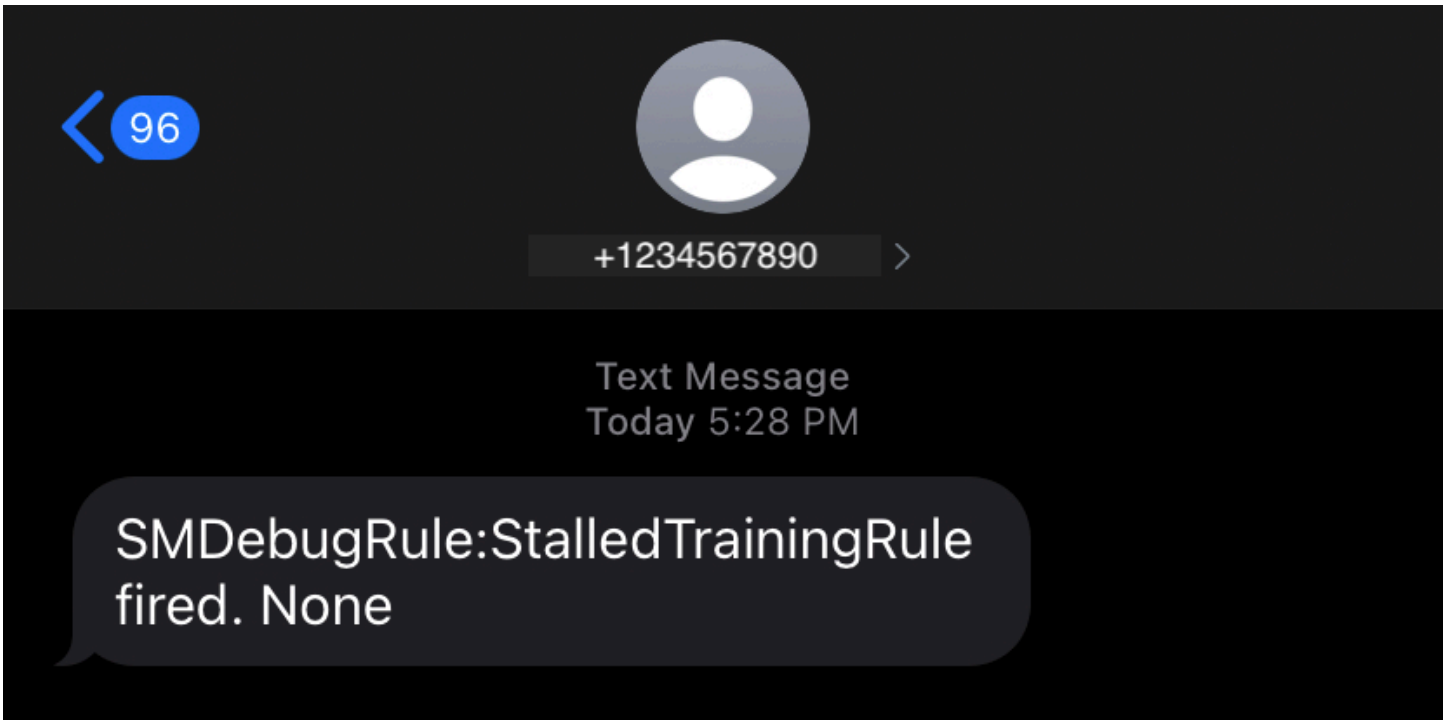
```
from sagemaker.debugger import Rule, rule_configs
import time

base_job_name_prefix= 'smdebug-stalled-demo-' + str(int(time.time()))

# Configure an action object for StopTraining
action_stop_training = rule_configs.ActionList(
    rule_configs.StopTraining()
)

# Configure an action object for Email
action_email = rule_configs.ActionList(
    rule_configs.Email("abc@abc.com")
)
```

```
# Configure a rule with the Email built-in action to trigger if a training job stalls
 for 60 seconds
stalled_training_job_rule_email = Rule.sagemaker(
        base_config=rule_configs.stalled_training_rule(),
        rule_parameters={
                "threshold": "60",
                "training_job_name_prefix": base_job_name_prefix
        },
        actions=action_email
)
stalled_training_job_rule_text.name="StalledTrainingJobRuleEmail"

# Configure a rule with the StopTraining built-in action to trigger if a training job
 stalls for 120 seconds
stalled_training_job_rule = Rule.sagemaker(
        base_config=rule_configs.stalled_training_rule(),
        rule_parameters={
                "threshold": "120",
                "training_job_name_prefix": base_job_name_prefix
        },
        actions=action_stop_training
)
stalled_training_job_rule.name="StalledTrainingJobRuleStopTraining"

estimator = Estimator(
    ...
    rules = [stalled_training_job_rule_email, stalled_training_job_rule]
)

estimator.fit(wait=False)
```

Saat pekerjaan pelatihan sedang berjalan, tindakan bawaan Debugger mengirimkan email notifikasi dan pesan teks setiap kali aturan menemukan masalah dengan pekerjaan pelatihan Anda. Tangkapan layar berikut menunjukkan contoh pemberitahuan email untuk pekerjaan pelatihan yang memiliki masalah pekerjaan pelatihan yang macet.



Screenshot berikut menunjukkan contoh notifikasi teks yang Debugger mengirimkan saat aturan menemukan StalledTraining masalah



Pertimbangan untuk menggunakan Debugger Built-in Actions

- Untuk menggunakan tindakan bawaan Debugger, koneksi internet diperlukan. Fitur ini tidak didukung dalam mode isolasi jaringan yang disediakan oleh Amazon SageMaker atau Amazon VPC.
- Tindakan bawaan tidak dapat digunakan untukAturan profiler.

- Tindakan bawaan tidak dapat digunakan pada pekerjaan pelatihan dengan gangguan pelatihan spot.
- Dalam pemberitahuan email atau teks,`None`muncul di akhir pesan. Ini tidak memiliki arti apa pun, sehingga Anda dapat mengabaikan teks`None`.

Buat Tindakan pada Aturan Menggunakan Amazon CloudWatch danAWS Lambda

Amazon CloudWatch mengumpulkan Amazon SageMaker log pekerjaan pelatihan model dan Amazon SageMaker Aturan debugger memproses log pekerjaan. Konfigurasikan Debugger dengan Amazon CloudWatch Acara danAWS Lambdauntuk mengambil tindakan berdasarkan status evaluasi aturan Debugger.

CloudWatch Log untuk Aturan Debugger dan Pekerjaan Pelatihan

Untuk menemukan log pekerjaan pelatihan dan log pekerjaan aturan Debugger

1. Buka CloudWatch konsol dihttps://console.aws.amazon.com/cloudwatch/.
2. Di bagian Navigasi di bagian bawahLogsimpul, pilihGrup Log.
3. Di daftar grup log, lakukan hal berikut:
   - Pilih/aws/pembuat sagem/TrainingJobsuntuk pelatihan log pekerjaan.
   - Pilih/aws/pembuat sagem/ProcessingJobsuntuk log pekerjaan aturan Debugger.

Anda dapat menggunakan status pekerjaan aturan pelatihan dan Debugger di CloudWatch log untuk mengambil tindakan lebih lanjut ketika ada masalah pelatihan.

Untuk informasi lebih lanjut tentang memantau pekerjaan pelatihan menggunakan CloudWatch, lihatPantau Amazon SageMaker.

Mengatur Debugger untuk Pengakhiran Pekerjaan Pelatihan Otomatis Menggunakan CloudWatch dan Lambda

Aturan Debugger memantau status pekerjaan pelatihan, dan CloudWatch Aturan acara mengamati status evaluasi pekerjaan pelatihan aturan Debugger.

Langkah 1: Buat Fungsi Lambda

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol tersebut di https://console.aws.amazon.com/lambda/.
2. Di bagian Navigasi di bagian kiri, pilihFungsidan kemudian memilihBuat fungsi.
3. PadaBuat fungsihalaman, pilihPenulis dari awalpilihan
4. DiInformasi dasarbagian, masukkanNama fungsi(Sebagai contohdebugger-rule-stop-training-pekerjaan).
5. Untuk Waktu pengoperasian, pilih Python 3.7.
6. UntukIzin, perluas opsi drop-down, dan pilihUbah peran eksekusi default.
7. UntukPeran eksekusi, pilihGunakan peran yang adadan pilih peran IAM yang Anda gunakan untuk pekerjaan pelatihan di SageMaker.

   > **Note**
   >
   > Pastikan Anda menggunakan peran eksekusi dengan`AmazonSageMakerFullAccess`dan`AWSLambdaBasicExecutionRole`terlampir. Jika tidak, fungsi Lambda tidak akan bereaksi dengan benar terhadap perubahan status aturan Debugger dari pekerjaan pelatihan. Jika Anda tidak yakin peran eksekusi mana yang digunakan, jalankan kode berikut di sel notebook Jupyter untuk mengambil output peran eksekusi:
   >
   > ```
   > import sagemaker
   > sagemaker.get_execution_role()
   > ```

8. Di bagian bawah halaman, pilih Buat Fungsi.

Gambar berikut menunjukkan contoh dariBuat fungsihalaman dengan bidang input dan pilihan selesai.

Create function Info

Choose one of the following options to create your function.

Author from scratch (selected)
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

debugger-rule-stop-training-job

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.7

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the **IAM console**.

- ○ Create a new role with basic Lambda permissions
- ◉ Use an existing role
- ○ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

service-role/AmazonSageMaker-ExecutionRole-20200611T110452

View the AmazonSageMaker-ExecutionRole-20200611T110452 role on the IAM console.

▶ **Advanced settings**

Cancel **Create function**

Langkah 2: Konfigurasikan fungsi Lambda

Untuk mengonfigurasi fungsi Lambda

1. DiKode fungsibagian dari halaman konfigurasi, paste script Python berikut di panel editor kode Lambda. The`lambda_handler`fungsi memantau status evaluasi aturan Debugger yang dikumpulkan oleh CloudWatch dan memicu`StopTrainingJob`Operasi API. TheAWS SDK for Python (Boto3) `client`untuk SageMaker menyediakan metode tingkat tinggi,`stop_training_job`, yang memicu`StopTrainingJob`Operasi API.

```
import json
import boto3
import logging

logger = logging.getLogger()
logger.setLevel(logging.INFO)

def lambda_handler(event, context):
    training_job_name = event.get("detail").get("TrainingJobName")
    logging.info(f'Evaluating Debugger rules for training job:
 {training_job_name}')
    eval_statuses = event.get("detail").get("DebugRuleEvaluationStatuses", None)

    if eval_statuses is None or len(eval_statuses) == 0:
        logging.info("Couldn't find any debug rule statuses, skipping...")
        return {
            'statusCode': 200,
            'body': json.dumps('Nothing to do')
        }

    # should only attempt stopping jobs with InProgress status
    training_job_status = event.get("detail").get("TrainingJobStatus", None)
    if training_job_status != 'InProgress':
        logging.debug(f"Current Training job status({training_job_status}) is not
 'InProgress'. Exiting")
        return {
            'statusCode': 200,
            'body': json.dumps('Nothing to do')
        }

    client = boto3.client('sagemaker')

    for status in eval_statuses:
```

```
        logging.info(status.get("RuleEvaluationStatus") + ', RuleEvaluationStatus='
+ str(status))
        if status.get("RuleEvaluationStatus") == "IssuesFound":
            secondary_status = event.get("detail").get("SecondaryStatus", None)
            logging.info(
                f'About to stop training job, since evaluation of rule
configuration {status.get("RuleConfigurationName")} resulted in "IssuesFound". ' +
                f'\ntraining job "{training_job_name}" status is
"{training_job_status}", secondary status is "{secondary_status}"' +
                f'\nAttempting to stop training job "{training_job_name}"'
            )
            try:
                client.stop_training_job(
                    TrainingJobName=training_job_name
                )
            except Exception as e:
                logging.error(
                    "Encountered error while trying to "
                    "stop training job {}: {}".format(
                        training_job_name, str(e)
                    )
                )
                raise e
    return None
```

Untuk informasi selengkapnya tentang antarmuka editor kode Lambda, lihatMembuat fungsi menggunakanAWSEditor konsol Lambda.

2. Lewati semua pengaturan lainnya dan pilihMenyimpandi bagian atas halaman konfigurasi.

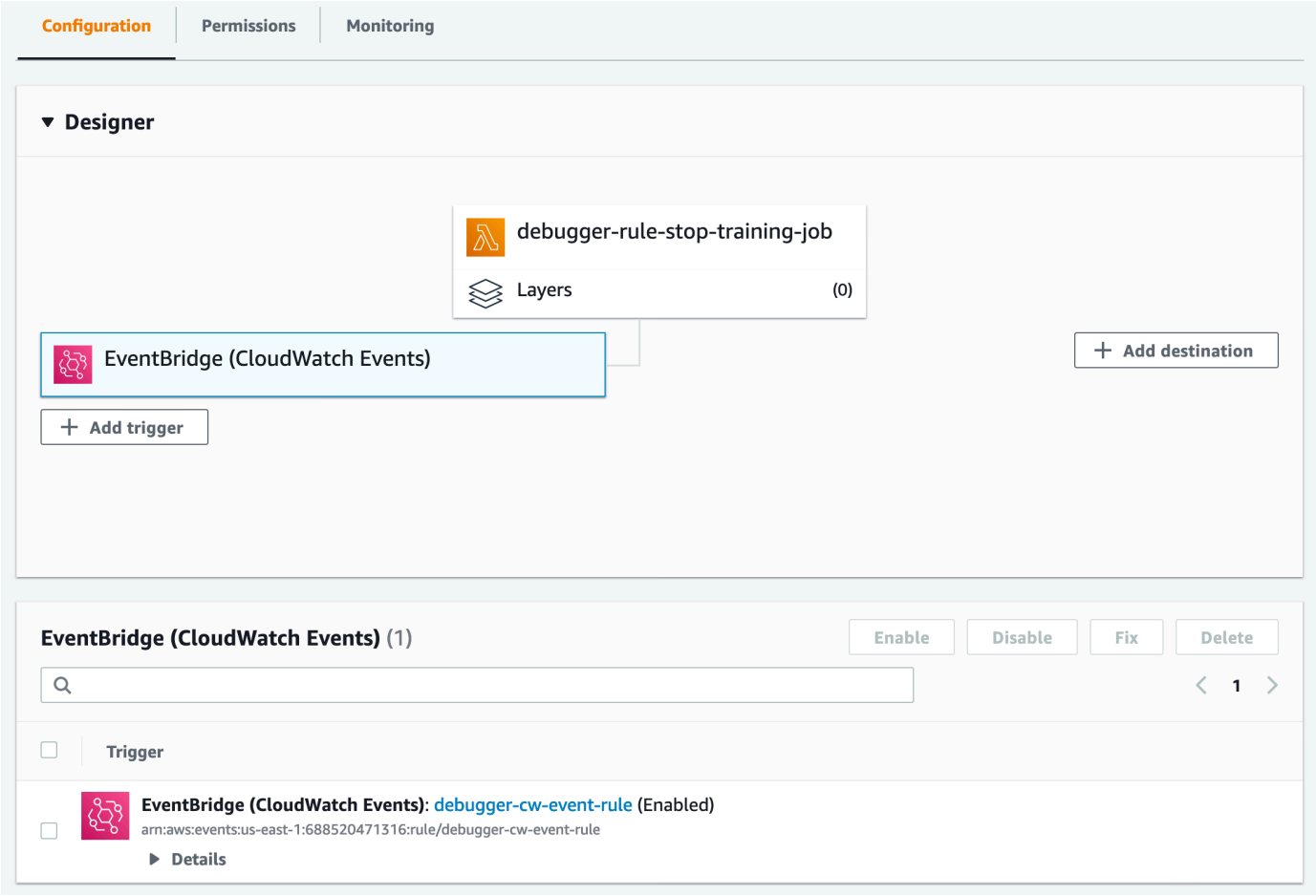
Langkah 3: Buat langganan CloudWatch Aturan Acara dan Tautan ke Fungsi Lambda untuk Debugger

Untuk membuat CloudWatch Aturan acara dan tautan ke fungsi Lambda untuk Debugger

1. Buka CloudWatch konsol dihttps://console.aws.amazon.com/cloudwatch/.
2. Di bagian Navigasi di bagian kiri, pilihAturandi bawahAcarasimpul.
3. Pilih Buat aturan.
4. DiSumber Acarabagian dariLangkah 1: Buat aturanhalaman, pilihSageMakeruntukNama Layanan, dan pilihSageMakerTraining Job State ChangeuntukTipe Acara. Event Pattern Preview akan terlihat seperti contoh string JSON berikut:

```
{
    "source": [
        "aws.sagemaker"
    ],
    "detail-type": [
        "SageMaker Training Job State Change"
    ]
}
```

5. DiTargetbagian, pilihTambahkan target*, dan pilihdebugger-rule-stop-training-pekerjaanFungsi Lambda yang Anda buat. Langkah ini menghubungkan CloudWatch Aturan acara dengan fungsi Lambda.
6. PilihKonfigurasikan detaildan pergi keLangkah 2: Konfigurasikan detail aturanhalaman.
7. Buat langganan CloudWatch nama definisi aturan. Sebagai contoh, debugger-cw-event-rule.
8. PilihBuat aturanuntuk menyelesaikan.
9. Kembali ke halaman konfigurasi fungsi Lambda dan segarkan halaman. Konfirmasikan bahwa itu dikonfigurasi dengan benar diDesainerpanel The CloudWatch Aturan peristiwa harus didaftarkan sebagai pemicu fungsi Lambda. Desain konfigurasi akan terlihat seperti contoh berikut:



Jalankan Contoh Notebook untuk Menguji Pemutusan Pekerjaan Pelatihan Otomatis

Anda dapat menjalankan contoh notebook berikut, yang disiapkan untuk bereksperimen dengan menghentikan pekerjaan pelatihan menggunakan aturan bawaan Debugger.

- Amazon SageMaker Debugger - Bereaksi terhadap CloudWatch Peristiwa dari Aturan

  Notebook contoh ini menjalankan pekerjaan pelatihan yang memiliki masalah gradien menghilang. DebuggerVanishingGradientaturan bawaan digunakan saat membangun SageMaker TensorFlow penaksir. Ketika aturan Debugger mendeteksi masalah, pekerjaan pelatihan dihentikan.

- Mendeteksi Pelatihan yang Terhenti dan Memanggil Tindakan Menggunakan SageMaker Aturan Debugger

  Notebook contoh ini menjalankan skrip pelatihan dengan baris kode yang memaksanya untuk tidur selama 10 menit. DebuggerStalledTrainingRuleaturan bawaan menimbulkan masalah dan menghentikan pekerjaan pelatihan.

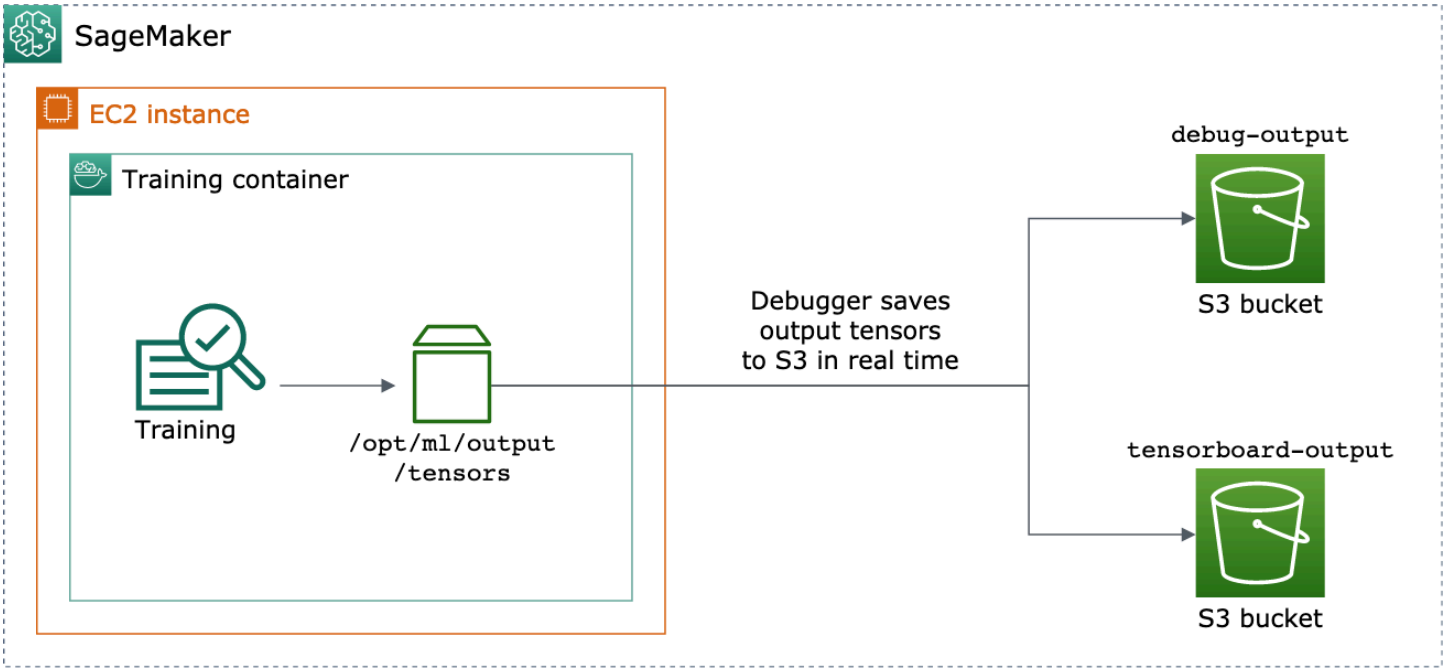
Nonaktifkan CloudWatch Aturan Acara untuk Berhenti Menggunakan Pemutusan Pekerjaan Pelatihan Otomatis

Jika Anda ingin menonaktifkan pemutusan pekerjaan pelatihan otomatis, Anda harus menonaktifkan CloudWatch Aturan acara. Di LambdaDesainerpanel, pilihEventBridge (CloudWatch Acara)blok yang ditautkan ke fungsi Lambda. Ini menunjukkan sebuahEventBridgepanel di bawahDesainerpanel (misalnya, lihat tangkapan layar sebelumnya). Pilih kotak di sampingEventBridge (CloudWatch Acara): debugger-cw-event-rule, dan kemudian pilihNonaktifkan. Jika Anda ingin menggunakan fungsionalitas penghentian otomatis nanti, Anda dapat mengaktifkan CloudWatch Aturan acara lagi.

Visualisasikan Tensor Output SageMaker Debugger Amazon di TensorBoard

> **Important**
>
> Halaman ini tidak digunakan lagi untuk mendukung Amazon SageMaker denganTensoBoard, yang memberikan TensorBoard pengalaman komprehensif yang terintegrasi dengan SageMaker Pelatihan dan fungsionalitas kontrol akses Domain. SageMaker Untuk mempelajari selengkapnya, lihat Gunakan TensorBoard untuk men-debug dan menganalisis pekerjaan pelatihan di Amazon SageMaker.

Gunakan SageMaker Debugger untuk membuat file tensor keluaran yang kompatibel dengannya. TensorBoard Muat file untuk memvisualisasikan TensorBoard dan menganalisis pekerjaan SageMaker pelatihan Anda. Debugger secara otomatis menghasilkan file tensor keluaran yang kompatibel dengannya. TensorBoard Untuk konfigurasi kait apa pun yang Anda sesuaikan untuk menyimpan tensor keluaran, Debugger memiliki fleksibilitas untuk membuat ringkasan skalar, distribusi, dan histogram yang dapat Anda impor. TensorBoard



Anda dapat mengaktifkan ini dengan melewati `DebuggerHookConfig` dan `TensorBoardOutputConfig` objek `keestimator`.

Prosedur berikut menjelaskan cara menyimpan skalar, bobot, dan bias sebagai tensor penuh, histogram, dan distribusi yang dapat divisualisasikan. TensorBoard Debugger menyimpannya ke jalur lokal wadah pelatihan (jalur defaultnya `/opt/ml/output/tensors`) dan disinkronkan ke lokasi Amazon S3 yang dilewatkan melalui objek konfigurasi output Debugger.

Untuk menyimpan file tensor keluaran yang TensorBoard kompatibel menggunakan Debugger

1. Mengatur objek `tensorboard_output_config` konfigurasi untuk menyimpan TensorBoard output menggunakan `TensorBoardOutputConfig` kelas Debugger. Untuk `s3_output_path` parameter, tentukan bucket S3 default dari SageMaker sesi saat ini atau bucket S3 yang disukai. Contoh ini tidak menambahkan `container_local_output_path` parameter; sebaliknya, itu diatur ke jalur lokal default `/opt/ml/output/tensors`.

```
import sagemaker
from sagemaker.debugger import TensorBoardOutputConfig

bucket = sagemaker.Session().default_bucket()
tensorboard_output_config = TensorBoardOutputConfig(
    s3_output_path='s3://{}'.format(bucket)
)
```

Untuk informasi tambahan, lihat Debugger `TensorBoardOutputConfig` API di Amazon SageMaker Python SDK.

2. Konfigurasikan hook Debugger dan sesuaikan nilai parameter kait. Misalnya, kode berikut mengonfigurasi hook Debugger untuk menyimpan semua output skalar setiap 100 langkah dalam fase pelatihan dan 10 langkah dalam fase validasi, `weights` parameter setiap 500 langkah (`save_interval`nilai default untuk menyimpan koleksi tensor adalah 500), dan `bias` parameter setiap 10 langkah global hingga langkah global mencapai 500.

```
from sagemaker.debugger import CollectionConfig, DebuggerHookConfig

hook_config = DebuggerHookConfig(
    hook_parameters={
        "train.save_interval": "100",
        "eval.save_interval": "10"
    },
    collection_configs=[
        CollectionConfig("weights"),
        CollectionConfig(
            name="biases",
            parameters={
                "save_interval": "10",
                "end_step": "500",
                "save_histogram": "True"
            }
        ),
    ]
)
```

Untuk informasi selengkapnya tentang API konfigurasi Debugger, lihat Debugger `CollectionConfig` dan `DebuggerHookConfig` API di Amazon SageMaker Python SDK.

3. Membangun SageMaker estimator dengan parameter Debugger melewati objek konfigurasi. Contoh berikut menunjukkan cara membuat SageMaker estimator generik. Anda dapat mengganti `estimator` dan `Estimator` dengan kelas induk estimator SageMaker kerangka kerja lain dan kelas estimator. Estimator SageMaker kerangka kerja yang tersedia untuk fungsi ini adalah`TensorFlow`,`PyTorch`, dan`MXNet`.

```
from sagemaker.estimator import Estimator

estimator = Estimator(
```

```
    ...
    # Debugger parameters
    debugger_hook_config=hook_config,
    tensorboard_output_config=tensorboard_output_config
)
estimator.fit()
```

`estimator.fit()`Metode ini memulai pekerjaan pelatihan, dan Debugger menulis file tensor keluaran secara real time ke jalur output Debugger S3 dan ke jalur keluaran S3. TensorBoard Untuk mengambil jalur keluaran, gunakan metode estimator berikut:

- Untuk jalur output Debugger S3, gunakan. `estimator.latest_job_debugger_artifacts_path()`
- Untuk jalur output TensorBoard S3, gunakan`estimator.latest_job_tensorboard_artifacts_path()`.

4. Setelah pelatihan selesai, periksa nama tensor keluaran yang disimpan:

```
from smdebug.trials import create_trial
trial = create_trial(estimator.latest_job_debugger_artifacts_path())
trial.tensor_names()
```

5. Periksa data TensorBoard keluaran di Amazon S3:

```
tensorboard_output_path=estimator.latest_job_tensorboard_artifacts_path()
print(tensorboard_output_path)
!aws s3 ls {tensorboard_output_path}/
```

6. Unduh data TensorBoard keluaran ke instance notebook Anda. Misalnya, AWS CLI perintah berikut mengunduh TensorBoard file ke `/logs/fit` bawah direktori kerja saat ini dari instance notebook Anda.

```
!aws s3 cp --recursive {tensorboard_output_path} ./logs/fit
```

7. Kompres direktori file ke file TAR untuk diunduh ke mesin lokal Anda.

```
!tar -cf logs.tar logs
```

8. Unduh dan ekstrak file Tensorboard TAR ke direktori di perangkat Anda, luncurkan server notebook Jupyter, buka notebook baru, dan jalankan aplikasi. TensorBoard

```
!tar -xf logs.tar
%load_ext tensorboard
%tensorboard --logdir logs/fit
```

Daftar Aturan Built-in Debugger

Gunakan aturan bawaan Debugger yang disediakan oleh Amazon SageMaker Debugger dan analisis metrik dan tensor yang dikumpulkan saat melatih model Anda. Aturan bawaan Debugger memantau berbagai kondisi umum yang sangat penting untuk keberhasilan pekerjaan pelatihan. Anda dapat memanggil aturan bawaan menggunakanAmazon SageMaker Pythonatau yang rendah SageMaker Operasi API. Tidak ada biaya tambahan untuk menggunakan aturan bawaan. Untuk informasi selengkapnya tentang penagihan, lihatAmazon SageMaker HargaHalaman

> **Note**
>
> Jumlah maksimum aturan bawaan yang dapat Anda lampirkan ke pekerjaan pelatihan adalah 20. SageMaker Debugger sepenuhnya mengelola aturan bawaan dan menganalisis pekerjaan pelatihan Anda secara serempak.

> **Important**
>
> Untuk menggunakan fitur Debugger baru, Anda perlu memutakhirkan SageMaker Python SDK dan pustaka klien SMDebug. Di kernel IPython Anda, notebook Jupyter, atau JupyterLab lingkungan, jalankan kode berikut untuk menginstal versi terbaru dari perpustakaan dan restart kernel.
>
> ```
> import sys
> import IPython
> !{sys.executable} -m pip install -U sagemaker smdebug
> IPython.Application.instance().kernel.do_shutdown(True)
> ```

Aturan Debugger

Aturan berikut adalah aturan bawaan Debugger yang dapat dipanggil menggunakan`Rule.sagemaker`metode kelas.

Aturan bawaan debugger untuk menghasilkan laporan pelatihan

Cakupan yang benar	Aturan yang ada
Laporan Pelatihan untuk SageMaker Pekerjaan pelatihan XGBoost	• create_xgboost_report

Aturan bawaan debugger untuk men-debug data pelatihan model (tensor keluaran)

Cakupan yang benar	Aturan yang ada
Kerangka pembelajaran mendalam (TensorFlow, MXNet, dan PyTorch)	• dead_relu • exploding_tensor • poor_weight_initialization • saturated_activation • vanishing_gradient • weight_update_ratio
Kerangka pembelajaran mendalam (TensorFlow, MXNet, dan PyTorch) dan algoritma XGBoost	• all_zero • class_imbalance • loss_not_decreasing • overfit • overtraining • similar_across_runs • stalled_training_rule • tensor_variance • unchanged_tensor
Aplikasi pembelajaran mendalam	• check_input_images • nlp_sequence_ratio
Algoritma XGBoost	• confusion • feature_importance_overweight • tree_depth

Untuk menggunakan aturan bawaan dengan nilai parameter default— gunakan format konfigurasi berikut:

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs

rules = [
    Rule.sagemaker(rule_configs.built_in_rule_name_1()),
    Rule.sagemaker(rule_configs.built_in_rule_name_2()),
    ...
    Rule.sagemaker(rule_configs.built_in_rule_name_n())
]
```

Untuk menggunakan aturan bawaan dengan menyesuaikan nilai parameter— gunakan format konfigurasi berikut:

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs

rules = [
    Rule.sagemaker(
        base_config=rule_configs.built_in_rule_name(),
        rule_parameters={
                "key": "value"
        }
        collections_to_save=[
            CollectionConfig(
                name="tensor_collection_name",
                parameters={
                    "key": "value"
                }
            )
        ]
    )
]
```

Untuk menemukan kunci yang tersedia untuk`rule_parameters`parameter, lihat tabel deskripsi parameter.

Contoh kode konfigurasi aturan disediakan untuk setiap aturan bawaan di bawah tabel deskripsi parameter.

- Untuk instruksi lengkap dan contoh penggunaan aturan bawaan Debugger, lihatDebugger Built-in Aturan Contoh Kode.

- Untuk instruksi lengkap tentang penggunaan aturan bawaan dengan tingkat rendah SageMaker Operasi API, lihatMengonfigurasi Debugger Menggunakan Amazon SageMaker API.

CreateXgboostReport

The CreateXgboostReport aturan mengumpulkan tensor keluaran dari pekerjaan pelatihan XGBoost dan membuat laporan pelatihan yang komprehensif secara otomatis. Anda dapat mengunduh laporan pembuatan profil yang komprehensif saat pekerjaan pelatihan sedang berjalan atau setelah pekerjaan pelatihan selesai, dan memeriksa kemajuan pelatihan atau hasil akhir dari pekerjaan pelatihan. The CreateXgboostReport aturan mengumpulkan tensor keluaran berikut secara default:

- `hyperparameters`— Menyimpan pada langkah pertama
- `metrics`— Menghemat kerugian dan akurasi setiap 5 langkah
- `feature_importance`— Menyimpan setiap 5 langkah
- `predictions`— Menyimpan setiap 5 langkah
- `labels`— Menyimpan setiap 5 langkah

Deskripsi Parameter untuk CreateXgboostReport Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String

```
rules=[
    Rule.sagemaker(
        rule_configs.create_xgboost_report()
    )
]
```

DeadRelu

Aturan ini mendeteksi ketika persentase fungsi aktivasi unit linier yang diperbaiki (ReLU) dalam percobaan dianggap mati karena aktivitas aktivasi mereka telah turun di bawah ambang batas. Jika persentase RelU tidak aktif dalam lapisan lebih besar dari`threshold_layer`nilai Relus yang tidak aktif, aturan mengembalikan`True`.

Deskripsi Parameter untuk DeadRelu Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandin gkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai default: `".*relu_output"`
`threshold_inactivity`	Mendefinisikan tingkat aktivitas di bawah mana ReLU dianggap mati. Sebuah ReLU mungkin aktif di awal percobaan dan kemudian

Nama Parameter	Deskripsi
	perlahan-lahan mati selama proses pelatihan . Jika ReLU aktif kurang dari`threshold _inactivity` , itu dianggap mati. Opsional Nilai yang benar: Float Nilai yang benar:`1.0`(dalam persentase)
`threshold_layer`	Pengembalian`True`jika persentase RelU tidak aktif dalam lapisan lebih besar dari`threshold _layer` . Pengembalian`False`jika persentase RelU tidak aktif dalam lapisan kurang dari`threshold _layer` . Opsional Nilai yang benar: Float Nilai yang benar:`50.0`(dalam persentase)

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.dead_relu(),
        rule_parameters={
                "tensor_regex": ".*relu_output|.*ReLU_output",
                "threshold_inactivity": "1.0",
                "threshold_layer": "50.0"
        },
        collections_to_save=[
            CollectionConfig(
                name="custom_relu_collection",
                parameters={
                    "include_regex: ".*relu_output|.*ReLU_output",
                    "save_interval": "500"
                }
```

```
            )
        ]
    )
]
```

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Aturan ini tidak tersedia untuk algoritma XGBoost.

ExplodingTensor

Aturan ini mendeteksi apakah tensor yang dipancarkan selama pelatihan memiliki nilai yang tidak terbatas, baik tak terbatas atau NaN (bukan angka). Jika nilai tidak terbatas terdeteksi, aturan kembali`True`.

Deskripsi Parameter untuk ExplodingTensor Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`collection_names`	Daftar nama koleksi yang tensornya diperiksa oleh aturan. Opsional Nilai valid: String Nilai default: None

Nama Parameter	Deskripsi
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandin gkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai valid: String Nilai default: `None`
`only_nan`	`True`untuk memonitor`base_trial` Tensor hanya untuk`NaN`nilai dan bukan untuk ketidakte rbatasan. `False`Untuk mengobati keduanya`NaN`dan tak terhingga sebagai nilai yang meledak dan untuk memantau keduanya. Opsional Nilai default: `False`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.exploding_tensor(),
        rule_parameters={
                "tensor_regex": ".*gradient",
                "only_nan": "False"
        },
        collections_to_save=[
            CollectionConfig(
```

```
                name="gradients",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Aturan ini tidak tersedia untuk algoritma XGBoost.

PoorWeightInitialization

Aturan ini mendeteksi jika parameter model Anda telah diinisialisasi dengan buruk.

Inisialisasi yang baik memecah simetri bobot dan gradien dalam jaringan saraf dan mempertahankan varians aktivasi yang sepadan di seluruh lapisan. Jika tidak, jaringan saraf tidak belajar secara efektif. Inisialisasi seperti Xavier bertujuan untuk menjaga varians konstan di seluruh aktivasi, yang sangat relevan untuk melatih jaring saraf yang sangat dalam. Inisialisasi yang terlalu kecil dapat menyebabkan gradien menghilang. Inisialisasi yang terlalu besar dapat menyebabkan gradien meledak. Aturan ini memeriksa varians input aktivasi lintas lapisan, distribusi gradien, dan konvergensi kerugian untuk langkah awal untuk menentukan apakah jaringan saraf telah diinisialisasi dengan buruk.

Deskripsi Parameter untuk PoorWeightInitialization Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan

Nama Parameter	Deskripsi
	Nilai valid: String
`activation_inputs_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandingkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai valid: String Nilai default: `".*relu_input"`
`threshold`	Jika rasio antara varians minimum dan maksimum bobot per lapisan melebihi`threshold` pada satu langkah, aturan kembali`True`. Opsional Nilai yang benar: Float Nilai default: `10.0`

Nama Parameter	Deskripsi
`distribution_range`	Jika perbedaan minimum antara persentil ke-5 dan ke-95 dari distribusi gradien kurang dari`distribution_range` , aturannya kembali`True`. Opsional Nilai yang benar: Float Nilai default: `0.001`
`patience`	Jumlah langkah untuk menunggu sampai kerugian dianggap tidak lagi berkurang. Opsional Nilai yang valid: Integer Nilai default: 5
`steps`	Jumlah langkah yang dianalisis aturan ini. Anda biasanya hanya perlu memeriksa beberapa iterasi pertama. Opsional Nilai yang valid: Float Nilai default: `10`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.poor_weight_initialization(),
        rule_parameters={
                "activation_inputs_regex": ".*relu_input|.*ReLU_input",
                "threshold": "10.0",
                "distribution_range": "0.001",
                "patience": "5",
```

```
                "steps": "10"
        },
        collections_to_save=[
            CollectionConfig(
                name="custom_relu_collection",
                parameters={
                    "include_regex": ".*relu_input|.*ReLU_input",
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Aturan ini tidak tersedia untuk algoritma XGBoost.

SaturatedActivation

Aturan ini mendeteksi jika lapisan aktivasi tanh dan sigmoid menjadi jenuh. Lapisan aktivasi jenuh ketika input lapisan mendekati maksimum atau minimum fungsi aktivasi. Minimum dan maksimum fungsi aktivasi tanh dan sigmoid ditentukan oleh masing-masing`min_threshold`dan`max_thresholds`nilai. Jika aktivitas node turun di bawah`threshold_inactivity`persentase, itu dianggap jenuh. Jika lebih dari`threshold_layer`persen dari node jenuh, aturan kembali`True`.

Deskripsi Parameter untuk SaturatedActivation Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan

Nama Parameter	Deskripsi
	Nilai valid: String
`collection_names`	Daftar nama koleksi yang tensornya diperiksa oleh aturan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai default: Tidak ada
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandingkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai valid: String Nilai default: `".*tanh_input|.*sigmoid_input".`

Nama Parameter	Deskripsi
`threshold_tanh_min`	Ambang batas minimum dan maksimum yang menentukan ekstrem input untuk fungsi aktivasi tanh, didefinisikan sebagai:`(min_threshold, max_threshold)` . Nilai default ditentukan berdasarkan ambang gradien menghilang 0,0000001. Opsional Nilai yang valid: Float Nilai yang benar:`-9.4999`
`threshold_tanh_max`	Ambang batas minimum dan maksimum yang menentukan ekstrem input untuk fungsi aktivasi tanh, didefinisikan sebagai:`(min_threshold, max_threshold)` . Nilai default ditentukan berdasarkan ambang gradien menghilang 0,0000001. Opsional Nilai yang benar: Float Nilai yang benar:`9.4999`
`threshold_sigmoid_min`	Ambang batas minimum dan maksimum yang menentukan ekstrem input untuk fungsi aktivasi sigmoid, didefinisikan sebagai:`(min_threshold, max_threshold)` . Nilai default ditentukan berdasarkan ambang gradien menghilang 0,0000001. Opsional Nilai yang benar: Float Nilai yang benar:`-23`

Nama Parameter	Deskripsi
`threshold_sigmoid_max`	Ambang batas minimum dan maksimum yang menentukan ekstrem input untuk fungsi aktivasi sigmoid, didefinisikan sebagai:`(min_threshold, max_threshold)` . Nilai default ditentukan berdasarkan ambang gradien menghilang 0,0000001. Opsional Nilai yang benar: Float Nilai yang benar:`16.99999`
`threshold_inactivity`	Persentase ketidakaktifan di bawah mana lapisan aktivasi dianggap jenuh. Aktivasi mungkin aktif di awal percobaan dan kemudian perlahan-lahan menjadi kurang aktif selama proses pelatihan. Opsional Nilai yang benar: Float Nilai yang benar:`1.0`
`threshold_layer`	Pengembalian`True`jika jumlah aktivasi jenuh dalam lapisan lebih besar dari`threshold_layer` Persentase Pengembalian`False`jika jumlah aktivasi jenuh dalam lapisan kurang dari`threshold_layer` Persentase Opsional Nilai yang benar: Float Nilai yang benar:`50.0`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.saturated_activation(),
        rule_parameters={
                "tensor_regex": ".*tanh_input|.*sigmoid_input",
                "threshold_tanh_min": "-9.4999",
                "threshold_tanh_max": "9.4999",
                "threshold_sigmoid_min": "-23",
                "threshold_sigmoid_max": "16.99999",
                "threshold_inactivity": "1.0",
                "threshold_layer": "50.0"
        },
        collections_to_save=[
            CollectionConfig(
                name="custom_activations_collection",
                parameters={
                    "include_regex": ".*tanh_input|.*sigmoid_input"
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Aturan ini tidak tersedia untuk algoritma XGBoost.

VanishingGradient

Aturan ini mendeteksi jika gradien dalam percobaan menjadi sangat kecil atau turun ke magnitudo nol. Jika rata-rata nilai absolut gradien turun di bawah yang ditentukan`threshold`, aturannya kembali`True`.

Parameter Deskripsi untuk VanishingGradient Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold`	Nilai di mana gradien ditentukan untuk menghilang. Opsional Nilai yang benar: Float Nilai default: `0.0000001` .

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.vanishing_gradient(),
        rule_parameters={
                "threshold": "0.0000001"
        },
        collections_to_save=[
            CollectionConfig(
                name="gradients",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Aturan ini tidak tersedia untuk algoritma XGBoost.

WeightUpdateRatio

Aturan ini melacak rasio pembaruan terhadap bobot selama pelatihan dan mendeteksi apakah rasio itu terlalu besar atau terlalu kecil. Jika rasio pembaruan terhadap bobot lebih besar dari`large_threshold value`atau jika rasio ini lebih kecil dari`small_threshold`, aturannya kembali`True`.

Kondisi untuk pelatihan adalah yang terbaik ketika pembaruan sepadan dengan gradien. Pembaruan yang terlalu besar dapat mendorong bobot menjauh dari nilai optimal, dan pembaruan yang sangat kecil menghasilkan konvergensi yang sangat lambat. Aturan ini mengharuskan bobot tersedia untuk dua langkah pelatihan, dan`train.save_interval`perlu diatur sama dengan`num_steps`.

Deskripsi Parameter untuk WeightUpdateRatio Aturan

Nama Parameter,	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`num_steps`	Jumlah langkah di mana aturan memeriksa untuk menentukan apakah tensor telah berubah. Jumlah langkah di mana Anda ingin membandingkan rasio berat. Jika Anda tidak memberikan nilai, aturan berjalan secara default terhadap langkah saat ini dan langkah tersimpan sebelumnya. Jika Anda mengganti default dengan meneruskan nilai untuk

Nama Parameter,	Deskripsi
	parameter ini, perbandingan dilakukan antara bobot pada langkahsdan pada satu langkah>= `s - num_steps` . Opsional Nilai yang valid: Integer Nilai default: `None`
`large_threshold`	Nilai maksimum yang dapat diambil oleh rasio pembaruan terhadap bobot sebelum aturan kembali`True`. Opsional Nilai yang benar: Float Nilai default: `10.0`
`small_threshold`	Nilai minimum yang dapat diambil oleh rasio pembaruan terhadap bobot, di bawahnya aturan dikembalikan`True`. Opsional Nilai yang benar: Float Nilai default: `0.00000001`

Nama Parameter,	Deskripsi
`epsilon`	Konstanta kecil digunakan untuk memastikan bahwa Debugger tidak membagi dengan nol saat menghitung pembaruan rasio untuk ditimbang. Opsional Nilai yang benar: Float Nilai default: `0.000000001`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.weight_update_ratio(),
        rule_parameters={
                "num_steps": "100",
                "large_threshold": "10.0",
                "small_threshold": "0.00000001",
                "epsilon": "0.000000001"
        },
        collections_to_save=[
            CollectionConfig(
                name="weights",
                parameters={
                    "train.save_interval": "100"
                }
            )
        ]
    )
]
```

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihat Konfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Aturan ini tidak tersedia untuk algoritma XGBoost.

AllZero

Aturan ini mendeteksi jika semua atau persentase tertentu dari nilai tensor adalah nol.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost. Anda harus menentukan`collection_names`atau`tensor_regex`Parameter Jika kedua parameter ditentukan, aturan memeriksa penyatuan tensor dari kedua set.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Parameter Deskripsi untuk AllZero Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`collection_names`	Daftar nama koleksi yang tensornya diperiksa oleh aturan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai default: `None`
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandin

Nama Parameter	Deskripsi
	gkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai default: None
`threshold`	Menentukan persentase nilai dalam tensor yang harus nol agar aturan ini dipanggil. Opsional Nilai yang benar: Float Nilai default: 100 (dalam persentase)

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.all_zero(),
        rule_parameters={
                "tensor_regex": ".*",
                "threshold": "100"
        },
        collections_to_save=[
            CollectionConfig(
                name="all",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

ClassImbalance

Aturan ini mengukur ketidakseimbangan pengambilan sampel antara kelas dan melempar kesalahan jika ketidakseimbangan melebihi ambang batas atau jika terlalu banyak kesalahan prediksi untuk kelas yang kurang terwakili terjadi sebagai akibat dari ketidakseimbangan.

Model klasifikasi membutuhkan kelas yang seimbang dalam kumpulan data pelatihan atau bobot/pengambilan sampel kelas yang tepat selama pelatihan. Aturan melakukan pemeriksaan berikut:

- Ini menghitung kejadian per kelas. Jika rasio jumlah sampel antara kelas terkecil dan terbesar lebih besar dari`threshold_imbalance`, kesalahan dilemparkan.
- Ini memeriksa akurasi prediksi per kelas. Jika resampling atau pembobotan belum diterapkan dengan benar, maka model dapat mencapai akurasi tinggi untuk kelas dengan banyak sampel pelatihan, tetapi akurasi rendah untuk kelas dengan sedikit sampel pelatihan. Jika sebagian kecil dari kesalahan prediksi untuk kelas tertentu di atas`threshold_misprediction`, kesalahan dilemparkan.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk ClassImbalance Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold_imbalance`	Ketidakseimbangan yang dapat diterima antara jumlah sampel di kelas terkecil dan di kelas terbesar. Melebihi nilai ambang batas ini menimbulkan kesalahan.

Nama Parameter	Deskripsi
	Opsional Nilai yang benar: Float Nilai default: `10`
`threshold_misprediction`	Batas fraksi misprediksi diperbolehkan untuk setiap kelas. Melebihi ambang batas ini menimbulkan kesalahan. Kelas yang kurang terwakili paling berisiko melewati ambang batas ini. Opsional Nilai yang benar: Float Nilai default: `0.7`
`samples`	Jumlah label yang harus diproses sebelum ketidakseimbangan dievaluasi. Aturan mungkin tidak dipicu sampai telah melihat sampel yang cukup di beberapa langkah. Semakin banyak kelas yang berisi kumpulan data Anda, semakin besar ini`sample`nomor harus. Opsional Nilai yang valid: Integer Nilai yang benar:`500`(dengan asumsi kumpulan data seperti MNIST dengan 10 kelas)

Nama Parameter	Deskripsi
`argmax`	Jika`True`,np.argmaxditerapkan pada tensor prediksi. Diperlukan ketika Anda memiliki vektor probabilitas untuk setiap kelas. Ini digunakan untuk menentukan kelas mana yang memiliki probabilitas tertinggi. Bersyarat Nilai yang benar: Boolean Nilai default: `False`
`labels_regex`	Nama tensor yang berisi label. Opsional Nilai valid: String Nilai default: `".*labels"`
`predictions_regex`	Nama tensor yang berisi prediksi. Opsional Nilai valid: String Nilai default: `".*predictions"`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.class_imbalance(),
        rule_parameters={
                "threshold_imbalance": "10",
                "threshold_misprediction": "0.7",
                "samples": "500",
                "argmax": "False",
                "labels_regex": ".*labels",
                "predictions_regex": ".*predictions"
        },
```

```
        collections_to_save=[
            CollectionConfig(
                name="custom_output_collection",
                parameters={
                    "include_regex": ".*labels|.*predictions",
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

LossNotDecreasing

Aturan ini mendeteksi ketika kerugian tidak menurun nilainya pada tingkat yang memadai. Kerugian ini harus skalar.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost. Anda harus menentukan`collection_names`atau`tensor_regex`Parameter Jika kedua parameter ditentukan, aturan memeriksa penyatuan tensor dari kedua set.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk LossNotDecreasing Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`collection_names`	Daftar nama koleksi yang tensornya diperiksa oleh aturan. Opsional

Nama Parameter	Deskripsi
	Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai default: `None`
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandin gkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai default: `None`
`use_losses_collection`	Jika disetel ke`True`, mencari kerugian dalam koleksi bernama “kerugian” saat koleksi hadir. Opsional Nilai yang benar: Boolean Nilai default: `True`

Nama Parameter	Deskripsi
`num_steps`	Jumlah minimum langkah setelah aturan memeriksa apakah kerugian telah menurun. Evaluasi aturan terjadi setiap`num_steps` . Aturan membandingkan kerugian untuk langkah ini dengan kerugian pada langkah yang setidaknya`num_steps` di belakang langkah saat ini. Misalnya, misalkan kerugian diselamatkan setiap tiga langkah, tetapi`num_steps` diatur ke 10. Pada langkah 21, kerugian untuk langkah 21 dibandingkan dengan kerugian untuk langkah 9. Langkah selanjutnya di mana kerugian diperiksa adalah langkah 33, karena sepuluh langkah setelah langkah 21 adalah langkah 31, dan pada langkah 31 dan langkah 32 kerugian tidak disimpan. Opsional Nilai yang valid: Integer Nilai default: `10`
`diff_percent`	Perbedaan persentase minimum dimana kerugian harus berkurang antara`num_steps` . Opsional Nilai yang benar:`0.0`< mengapung <`100` Nilai yang benar:`0.1`(dalam persentase)

Nama Parameter	Deskripsi
`increase_threshold_percent`	Persentase ambang batas maksimum bahwa kerugian dibiarkan meningkat dalam kasus kerugian telah meningkat Opsional Nilai yang benar:`0`< mengapung <`100` Nilai yang benar:`5`(dalam persentase)
`mode`	Nama mode Debugger untuk menanyakan nilai tensor untuk pemeriksaan aturan. Jika ini tidak diteruskan, aturan memeriksa secara default untuk`mode.EVAL` , maka`mode.TRAIN` , dan kemudian`mode.GLOBAL` . Opsional Nilai yang valid: String (`EVAL`,`TRAIN`, atau`GLOBAL`) Nilai default: `GLOBAL`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.loss_not_decreasing(),
        rule_parameters={
                "tensor_regex": ".*",
                "use_losses_collection": "True",
                "num_steps": "10",
                "diff_percent": "0.1",
                "increase_threshold_percent": "5",
                "mode": "GLOBAL"
        },
        collections_to_save=[
            CollectionConfig(
                name="losses",
                parameters={
```

```
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

Overfit

Aturan ini mendeteksi jika model Anda terlalu cocok dengan data pelatihan dengan membandingkan validasi dan kerugian pelatihan.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Cara standar untuk mencegah overfitting adalah dengan mengatur model Anda.

Deskripsi Parameter untuk Aturan Overfit

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak

Nama Parameter	Deskripsi
	ada pola yang diteruskan, aturan membandingkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang valid: Tidak ada
`start_step`	Langkah untuk mulai membandingkan validasi dan kehilangan pelatihan. Opsional Nilai yang valid: Integer Nilai default: `0`
`patience`	Jumlah langkah yang`ratio_threshold` diizinkan untuk melebihi nilai yang ditetapkan sebelum model dianggap overfit. Opsional Nilai yang valid: Integer Nilai default: `1`

Nama Parameter	Deskripsi
`ratio_threshold`	Rasio maksimum perbedaan antara kehilangan validasi rata-rata dan kehilangan pelatihan rata-rata terhadap kerugian pelatihan rata-rata. Jika ambang batas ini terlampaui untuk`patience`jumlah langkah, model sedang overfit dan aturan kembali`True`. Opsional Nilai yang benar: Float Nilai default: `0.1`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.overfit(),
        rule_parameters={
                "tensor_regex": ".*",
                "start_step": "0",
                "patience": "1",
                "ratio_threshold": "0.1"
        },
        collections_to_save=[
            CollectionConfig(
                name="losses",
                parameters={
                    "train.save_interval": "100",
                    "eval.save_interval": "10"
                }
            )
        ]
    )
]
```

Latihan berlebihan

Aturan ini mendeteksi jika model sedang dilatih berlebihan. Setelah sejumlah iterasi pelatihan pada model yang berperilaku baik (baik pelatihan dan kehilangan validasi menurun), model mendekati

minimum fungsi kerugian dan tidak membaik lagi. Jika model melanjutkan pelatihan, dapat terjadi bahwa kehilangan validasi mulai meningkat, karena model mulai overfitting. Aturan ini menetapkan ambang batas dan kondisi untuk menentukan apakah model tidak membaik, dan mencegah masalah overfitting karena overtraining.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

> **Note**
>
> Overtraining dapat dihindari dengan berhenti lebih awal. Untuk informasi tentang pemberhentian lebih awal, lihatHentikan Pekerjaan Pelatihan Lebih Awal. Untuk contoh yang menunjukkan cara menggunakan pelatihan spot dengan Debugger, lihatAktifkan Pelatihan Spot dengan Amazon SageMaker Debugger.

Deskripsi Parameter untuk Aturan Overtraining

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`patience_train`	Jumlah langkah untuk menunggu sebelum kehilangan pelatihan dianggap tidak membaik lagi. Opsional Nilai yang valid: Integer

Nama Parameter	Deskripsi
	Nilai default: 5
`patience_validation`	Jumlah langkah menunggu sebelum kehilangan validasi dianggap tidak membaik lagi. Opsional Nilai yang valid: Integer Nilai default: `10`
`delta`	Ambang minimum dengan seberapa banyak kesalahan harus ditingkatkan sebelum dianggap sebagai optimal baru. Opsional Nilai yang benar: Float Nilai default: `0.01`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.overtraining(),
        rule_parameters={
                "patience_train": "5",
                "patience_validation": "10",
                "delta": "0.01"
        },
        collections_to_save=[
            CollectionConfig(
                name="losses",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

SimilarAcrossRuns

Aturan ini membandingkan tensor yang dikumpulkan dari uji coba dasar dengan tensor dari uji coba lain.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk SimilarAcrossRuns Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`other_trials`	Nama pekerjaan pelatihan lengkap yang tensornya ingin Anda bandingkan dengan tensor yang dikumpulkan dari saat ini`base_trial` . Diperlukan Nilai valid: String
`collection_names`	Daftar nama koleksi yang tensornya diperiksa oleh aturan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang valid: Tidak ada

Nama Parameter	Deskripsi
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandingkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang valid: Tidak ada

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.similar_across_runs(),
        rule_parameters={
                "other_trials": "<specify-another-job-name>",
                "collection_names": "losses",
                "tensor_regex": ".*"
        },
        collections_to_save=[
            CollectionConfig(
                name="losses",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

StalledTrainingRule

StalledTrainingRule mendeteksi jika tidak ada kemajuan yang dibuat pada pekerjaan pelatihan, dan menghentikan pekerjaan pelatihan jika aturan tersebut berlaku. Aturan ini mengharuskan tensor disimpan secara berkala dalam interval waktu yang ditentukan oleh tensor`threshold`Parameter Aturan ini terus memantau tensor baru, dan jika tidak ada tensor baru yang dipancarkan untuk aturan interval ambang akan diaktifkan.

Deskripsi Parameter untuk StalledTrainingRule Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold`	Ambang batas yang menentukan berapa banyak waktu dalam hitungan detik aturan menunggu output tensor sampai memicu masalah pelatihan yang macet. Nilai default adalah 1800 detik. Opsional Nilai yang valid: Integer Nilai default: `1800`
`stop_training_on_fire`	Jika disetel ke`True`, perhatikan apakah pekerjaan pelatihan dasar menghasilkan tensor di”`threshold` “Detik Opsional Nilai yang benar: Boolean

Nama Parameter	Deskripsi
	Nilai default: `False`
`training_job_name_prefix`	Awalan nama pekerjaan pelatihan dasar. Jika`stop_training_on_fire` benar, aturan mencari SageMaker pekerjaan pelatihan dengan awalan ini di akun yang sama. Jika ada ketidakaktifan yang ditemukan, aturannya membutuhkan`StopTrainingJob` tindakan. Perhatikan jika ada beberapa pekerjaan yang ditemukan dengan awalan yang sama, aturan akan melewatkan penghentian. Penting bahwa awalan ditetapkan unik per setiap pekerjaan pelatihan. Opsional Nilai valid: String

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.stalled_training_rule(),
        rule_parameters={
                "threshold": "1800",
                "stop_training_on_fire": "True",
                "training_job_name_prefix": "<specify-training-base-job-name>"
        },
        collections_to_save=[
            CollectionConfig(
                name="losses",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

TensorVariance

Aturan ini mendeteksi jika Anda memiliki tensor dengan varians yang sangat tinggi atau rendah. Varians yang sangat tinggi atau rendah dalam tensor dapat menyebabkan saturasi neuron, yang mengurangi kemampuan belajar jaringan saraf. Varians yang sangat tinggi dalam tensor juga pada akhirnya dapat menyebabkan tensor meledak. Gunakan aturan ini untuk mendeteksi masalah tersebut sejak dini.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost. Anda harus menentukan`collection_names`atau`tensor_regex`Parameter Jika kedua parameter ditentukan, aturan memeriksa penyatuan tensor dari kedua set.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk TensorVariance Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`collection_names`	Daftar nama koleksi yang tensornya diperiksa oleh aturan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang valid: Tidak ada
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor

Nama Parameter	Deskripsi
	bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandingkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang valid: Tidak ada
`max_threshold`	Ambang batas atas varians tensor. Opsional Nilai yang benar: Float Nilai yang valid: Tidak ada
`min_threshold`	Ambang batas bawah varians tensor. Opsional Nilai yang benar: Float Nilai yang valid: Tidak ada

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.tensor_variance(),
        rule_parameters={
                "collection_names": "weights",
                "max_threshold": "10",
                "min_threshold": "0.00001",
        },
```

```
        collections_to_save=[
            CollectionConfig(
                name="weights",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

UnchangedTensor

Aturan ini mendeteksi apakah tensor tidak lagi berubah di seluruh langkah.

Aturan ini menjalankannumpy.allclosemetode untuk memeriksa apakah tensor tidak berubah.

Aturan ini dapat diterapkan baik ke salah satu kerangka kerja pembelajaran mendalam yang didukung (TensorFlow, MXNet, dan PyTorch) atau ke algoritma XGBoost. Anda harus menentukan`collection_names`atau`tensor_regex`Parameter Jika kedua parameter ditentukan, aturan memeriksa penyatuan tensor dari kedua set.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk UnchangedTensor Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`collection_names`	Daftar nama koleksi yang tensornya diperiksa oleh aturan. Opsional

Nama Parameter	Deskripsi
	Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang valid: Tidak ada
`tensor_regex`	Daftar pola regex digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandingkan semua tensor yang dikumpulkan dalam uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang valid: Tidak ada

Nama Parameter	Deskripsi
`num_steps`	Jumlah langkah di mana aturan memeriksa untuk menentukan apakah tensor telah berubah. Ini memeriksa yang terakhir`num_steps` yang tersedia. Mereka tidak perlu berturut-turut. Jika`num_steps` adalah 2, pada langkah s tidak selalu memeriksa s-1 dan s. Jika s-1 tidak tersedia, ia memeriksa langkah terakhir yang tersedia bersama dengan s. Dalam hal ini, ia memeriksa langkah terakhir yang tersedia dengan langkah saat ini. Opsional Nilai yang valid: Integer Nilai default: `3`
`rtol`	Parameter toleransi relatif yang akan diteruska n ke`numpy.allclose` Metode Opsional Nilai yang benar: Float Nilai default: `1e-05`
`atol`	Parameter toleransi absolut yang akan diteruskan ke`numpy.allclose` Metode Opsional Nilai yang benar: Float Nilai default: `1e-08`

Nama Parameter	Deskripsi
`equal_nan`	Apakah akan membandingkan NaNs sebagai sama. Jika`True`, NaNs dalam array input a dianggap sama dengan NaNs dalam array input b dalam array output. Parameter ini diteruskan ke`numpy.allclose` Metode Opsional Nilai yang benar: Boolean Nilai default: `False`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.unchanged_tensor(),
        rule_parameters={
                "collection_names": "losses",
                "tensor_regex": "",
                "num_steps": "3",
                "rtol": "1e-05",
                "atol": "1e-08",
                "equal_nan": "False"
        },
        collections_to_save=[
            CollectionConfig(
                name="losses",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

CheckInputImages

Aturan ini memeriksa apakah gambar input telah dinormalisasi dengan benar. Secara khusus, ini mendeteksi jika rata-rata data sampel berbeda lebih dari nilai ambang dari nol. Banyak model visi komputer mengharuskan data input memiliki mean nol dan varians unit.

Aturan ini berlaku untuk aplikasi pembelajaran mendalam.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk CheckInputImages Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold_mean`	Ambang batas yang menentukan seberapa besar rata-rata data input dapat berbeda dari 0. Opsional Nilai yang benar: Float Nilai default: `0.2`
`threshold_samples`	Jumlah gambar yang harus diambil sampelnya sebelum kesalahan dapat dilemparkan. Jika nilainya terlalu rendah, estimasi rata-rata dataset tidak akan akurat. Opsional Nilai yang valid: Integer

Nama Parameter	Deskripsi
	Nilai default: `500`
`regex`	Nama tensor data input. Opsional Nilai valid: String Nilai yang benar:`".*hybridsequential0_input_0"` (nama tensor input untuk model Apache MXNet menggunakan HybridSequential)
`channel`	Posisi saluran warna dalam larik bentuk tensor input. Opsional Nilai yang valid: Integer Nilai yang benar:1(misalnya, MxNet mengharapkan data input dalam bentuk (batch_size, channel, height, width))

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.check_input_images(),
        rule_parameters={
                "threshold_mean": "0.2",
                "threshold_samples": "500",
                "regex": ".*hybridsequential0_input_0",
                "channel": "1"
        },
        collections_to_save=[
            CollectionConfig(
                name="custom_inputs_collection",
                parameters={
                    "include_regex": ".*hybridsequential0_input_0",
                    "save_interval": "500"
```

```
                }
            )
        ]
    )
]
```

NLPSequenceRatio

Aturan ini menghitung rasio token tertentu mengingat sisa urutan input yang berguna untuk mengoptimalkan kinerja. Misalnya, Anda dapat menghitung persentase padding end-of-sentence (EOS) token dalam urutan input Anda. Jika jumlah token EOS terlalu tinggi, strategi bucketing alternatif harus dilakukan. Anda juga dapat menghitung persentase token yang tidak diketahui dalam urutan input Anda. Jika jumlah kata yang tidak diketahui terlalu tinggi, kosakata alternatif dapat digunakan.

Aturan ini berlaku untuk aplikasi pembelajaran mendalam.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk NLPSequenceRatio Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`tensor_regex`	Daftar pola regex yang digunakan untuk membatasi perbandingan ini dengan tensor bernilai skalar tertentu. Aturan hanya memeriksa tensor yang cocok dengan pola regex yang ditentukan dalam daftar. Jika tidak ada pola yang diteruskan, aturan membandin gkan semua tensor yang dikumpulkan dalam

Nama Parameter	Deskripsi
	uji coba secara default. Hanya tensor bernilai skalar yang dapat dicocokkan. Opsional Nilai yang valid: Daftar string atau string yang dipisahkan koma Nilai yang benar:`".*embedding0_input_0"` (dengan asumsi penyematan sebagai lapisan awal jaringan)
`token_values`	Sebuah string dari daftar nilai numerik token. Misalnya, “3, 0". Opsional Nilai yang valid: String nilai numerik yang dipisahkan koma Nilai default: `0`
`token_thresholds_percent`	Sebuah string dari daftar ambang batas (dalam persentase) yang sesuai dengan masing-masing`token_values` . Misalnya, “50,0, 50,0”. Opsional Nilai yang valid: String yang dipisahkan koma Nilai default: `"50"`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.nlp_sequence_ratio(),
        rule_parameters={
                "tensor_regex": ".*embedding0_input_0",
                "token_values": "0",
```

```
                "token_thresholds_percent": "50"
        },
        collections_to_save=[
            CollectionConfig(
                name="custom_inputs_collection",
                parameters={
                    "include_regex": ".*embedding0_input_0"
                }
            )
        ]
    )
]
```

Kebingungan

Aturan ini mengevaluasi kebaikan matriks kebingungan untuk masalah klasifikasi.

Ini menciptakan matriks ukuran`category_no*category_no`dan mengisinya dengan data yang berasal dari (`labels`,`predictions`) pasangan. Untuk masing-masing (`labels`,`predictions`) pasangan, hitungan di`confusion[labels][predictions]`bertambah dengan 1. Ketika matriks terisi penuh, rasio data nilai on-diagonal dan nilai off-diagonal dievaluasi sebagai berikut:

- Untuk elemen pada diagonal:`confusion[i][i]/sum_j(confusion[j][j])>=min_diag`
- Untuk elemen di luar diagonal:`confusion[j][i])/sum_j(confusion[j][i])<=max_off_diag`

Aturan ini dapat diterapkan pada algoritma XGBoost.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk Aturan Kebingungan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan

Nama Parameter	Deskripsi
	Nilai valid: String
`category_no`	Jumlah kategori. Opsional Nilai yang valid: Integer ≥2 Nilai default: `"None"`
`labels`	The`labels`koleksi tensor atau vektor 1-d dari label sejati. Opsional Nilai valid: String Nilai default: `"labels"`
`predictions`	The`predictions` koleksi tensor atau vektor 1-d dari label yang diperkirakan. Opsional Nilai valid: String Nilai default: `"predictions"`
`labels_collection`	Aturan memeriksa tensor dalam koleksi ini untuk`labels`. Opsional Nilai valid: String Nilai default: `"labels"`

Nama Parameter	Deskripsi
`predictions_collection`	Aturan memeriksa tensor dalam koleksi ini untuk`predictions` . Opsional Nilai valid: String Nilai default: `"predictions"`
`min_diag`	Ambang minimum untuk rasio data pada diagonal. Opsional Nilai yang benar:`0`≤mengapung ≤`1` Nilai default: `0.9`
`max_off_diag`	Ambang batas maksimum untuk rasio data dari diagonal. Opsional Nilai yang benar: Tidak ada`0`≤mengapung ≤`1` Nilai default: `0.1`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.confusion(),
        rule_parameters={
                "category_no": "10",
                "labels": "labels",
                "predictions": "predictions",
                "labels_collection": "labels",
                "predictions_collection": "predictions",
                "min_diag": "0.9",
                "max_off_diag": "0.1"
        },
```

```
        collections_to_save=[
            CollectionConfig(
                name="labels",
                parameters={
                    "save_interval": "500"
                }
            ),
            CollectionConfig(
                name="predictions",
                parameters={
                    "include_regex": "500"
                }
            )
        ]
    )
]
```

> **Note**
>
> Aturan ini menyimpulkan nilai default untuk parameter opsional jika nilainya tidak ditentukan.

FeatureImportanceOverweight

Aturan ini mengakumulasi bobot n nilai kepentingan fitur terbesar per langkah dan memastikan bahwa mereka tidak melebihi ambang batas. Misalnya, Anda dapat mengatur ambang batas untuk 3 fitur teratas agar tidak menampung lebih dari 80 persen dari total bobot model.

Aturan ini hanya berlaku untuk algoritma XGBoost.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihatKonfigurasikan Aturan Bawaan Debugger.

Deskripsi Parameter untuk FeatureImportanceOverweight Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger.

Nama Parameter	Deskripsi
	Diperlukan Nilai valid: String
`threshold`	Mendefinisikan ambang batas untuk proporsi jumlah kumulatif darinfitur terbesar. AngkanDitentukan olehn`features` Parameter Opsional Nilai yang benar: Float Nilai default: `0.8`
`nfeatures`	Jumlah fitur terbesar. Opsional Nilai yang valid: Integer Nilai default: `3`
`tensor_regex`	Ekspresi reguler (regex) tensor menamai aturan untuk dianalisis. Opsional Nilai valid: String Nilai default: `".*feature_importance/weight"`

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.feature_importance_overweight(),
        rule_parameters={
                "threshold": "0.8",
                "nfeatures": "3",
                "tensor_regex": ".*feature_importance/weight"
```

```
        },
        collections_to_save=[
            CollectionConfig(
                name="feature_importance",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

TreeDepth

Aturan ini mengukur kedalaman pohon dalam model XGBoost. XGBoost menolak perpecahan jika mereka tidak meningkatkan kerugian. Ini mengatur pelatihan. Akibatnya, pohon itu mungkin tidak tumbuh sedalam yang didefinisikan oleh`depth`Parameter

Aturan ini hanya berlaku untuk algoritma XGBoost.

Untuk contoh cara mengonfigurasi dan menerapkan aturan bawaan, lihat[Konfigurasikan Aturan Bawaan Debugger](#).

Deskripsi Parameter untuk TreeDepth Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`depth`	Kedalaman pohon. Kedalaman pohon diperoleh dengan menghitung logaritma basis 2 dari ID node terbesar. Opsional

Nama Parameter	Deskripsi
	Nilai yang benar: Float Nilai default: 4

```
built_in_rules = [
    Rule.sagemaker(
        base_config=rule_configs.tree_depth(),
        rule_parameters={
                "depth": "4"
        },
        collections_to_save=[
            CollectionConfig(
                name="tree",
                parameters={
                    "save_interval": "500"
                }
            )
        ]
    )
]
```

Buat Aturan Kustom Debugger untuk Analisis Job Pelatihan

Anda dapat membuat aturan khusus untuk memantau pekerjaan pelatihan Anda menggunakan API Aturan Debugger dan open sourceperpustakaan Python smdebugyang menyediakan alat untuk membangun wadah aturan Anda sendiri.

Topik

- Prasyarat untuk Membuat Aturan Kustom Debugger
- Menggunakan Debugger Client Librarysmdebuguntuk Membuat Script Python Aturan Kustom
- Gunakan API Debugger untuk Menjalankan Aturan Kustom Anda Sendiri

Prasyarat untuk Membuat Aturan Kustom Debugger

Untuk membuat aturan khusus Debugger, Anda memerlukan beberapa prasyarat berikut.

- Aturan Debugger SageMaker. API Khusus

- Open source smdebug Python perpustakaan
- Script python aturan kustom Anda sendiri
- URL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Kustom

Menggunakan Debugger Client Library**smdebug**untuk Membuat Script Python Aturan Kustom

ParametersmdebugAturan API menyediakan antarmuka untuk mengatur aturan kustom Anda sendiri. Script python berikut adalah contoh bagaimana membangun aturan kustom,`CustomGradientRule`. Aturan kustom tutorial ini menonton jika gradien menjadi terlalu besar dan mengatur ambang default sebagai 10. Aturan kustom mengambil uji coba dasar yang dibuat oleh estimator SageMaker ketika memulai pekerjaan pelatihan.

```
from smdebug.rules.rule import Rule

class CustomGradientRule(Rule):
    def __init__(self, base_trial, threshold=10.0):
        super().__init__(base_trial)
        self.threshold = float(threshold)

    def invoke_at_step(self, step):
        for tname in self.base_trial.tensor_names(collection="gradients"):
            t = self.base_trial.tensor(tname)
            abs_mean = t.reduction_value(step, "mean", abs=True)
            if abs_mean > self.threshold:
                return True
        return False
```

Anda dapat menambahkan beberapa kelas aturan kustom sebanyak yang Anda inginkan dalam skrip python yang sama dan menyebarkannya ke uji coba pekerjaan pelatihan dengan membangun objek aturan khusus di bagian berikut.

Gunakan API Debugger untuk Menjalankan Aturan Kustom Anda Sendiri

Contoh kode berikut menunjukkan cara mengonfigurasi aturan khusus denganAmazon SageMaker Python SDK. Contoh ini mengasumsikan bahwa skrip aturan khusus yang Anda buat pada langkah sebelumnya terletak pada 'path/to/my_custom_rule.py'.

```
from sagemaker.debugger import Rule, CollectionConfig

custom_rule = Rule.custom(
```

```
    name='MyCustomRule',
    image_uri='759209512951.dkr.ecr.us-west-2.amazonaws.com/sagemaker-debugger-rule-
evaluator:latest',
    instance_type='ml.t3.medium',
    source='path/to/my_custom_rule.py',
    rule_to_invoke='CustomGradientRule',
    collections_to_save=[CollectionConfig("gradients")],
    rule_parameters={"threshold": "20.0"}
)
```

Daftar berikut menjelaskan Debugger`Rule.custom`Argumen API.

- `name`(str): Tentukan nama aturan khusus yang Anda inginkan.
- `image_uri`(str): Ini adalah gambar dari wadah yang memiliki logika memahami aturan kustom Anda. Ini sumber dan mengevaluasi koleksi tensor yang ditentukan yang Anda simpan dalam pekerjaan pelatihan. Anda dapat menemukan daftar gambar evaluator aturan SageMaker open source dariURL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Kustom.
- `instance_type`(str): Anda perlu menentukan sebuah instance untuk membangun sebuah wadah aturan docker. Ini memunculkan instance secara paralel dengan wadah pelatihan.
- `source`(str): Ini adalah jalur lokal atau URI Amazon S3 ke skrip aturan khusus Anda.
- `rule_to_invoke`(str): Ini menentukan implementasi kelas Aturan tertentu dalam skrip aturan kustom Anda. SageMaker hanya mendukung satu aturan yang akan dievaluasi pada suatu waktu dalam pekerjaan aturan.
- `collections_to_save`(str): Ini menentukan koleksi tensor mana yang akan Anda simpan agar aturan dijalankan.
- `rule_parameters`(kamus): Ini menerima input parameter dalam format kamus. Anda dapat menyesuaikan parameter yang Anda konfigurasi dalam skrip aturan kustom.

Setelah Anda mengatur`custom_rule`objek, Anda dapat menggunakannya untuk membangun estimator SageMaker untuk setiap pekerjaan pelatihan. Tentukan`entry_point`ke naskah pelatihan Anda. Anda tidak perlu melakukan perubahan skrip pelatihan Anda.

```
from sagemaker.tensorflow import TensorFlow

estimator = TensorFlow(
                role=sagemaker.get_execution_role(),
                base_job_name='smdebug-custom-rule-demo-tf-keras',
                entry_point='path/to/your_training_script.py'
```

```
                train_instance_type='ml.p2.xlarge'
                ...

                # debugger-specific arguments below
                rules = [custom_rule]
)

estimator.fit()
```

Untuk variasi lainnya dan contoh lanjutan menggunakan aturan kustom Debugger, lihat contoh notebook berikut.

- Pantau pekerjaan pelatihan Anda dengan aturan kustom Amazon SageMaker Debugger
- PyTorch model berulang pemangkasan Resnet dan AlexNet
- Memicu Amazon CloudWatch Events menggunakan Aturan Debugger untuk Mengambil Tindakan Berdasarkan Status Pelatihan dengan TensorFlow

Gunakan Debugger dengan Custom Training Containers

Amazon SageMaker Debugger tersedia untuk semua model deep learning yang Anda bawa ke AmazonSageMaker. SageMaker`Estimator`APIAWS CLI, API, dan Debugger memungkinkan Anda menggunakan gambar dasar Docker apa pun untuk membuat dan menyesuaikan kontainer untuk melatih model Anda. Untuk menggunakan Debugger dengan kontainer yang disesuaikan, Anda perlu membuat perubahan minimal pada skrip latihan Anda untuk mengimplementasikan callback hook Debugger dan mengambil tensor dari pekerjaan pelatihan.

Anda memerlukan sumber daya berikut untuk membangun wadah yang disesuaikan dengan Debugger.

- SDK Amazon SageMaker Python
- Pustaka klien sumber terbuka SMDebug
- Gambar dasar Docker pilihan Anda
- Skrip pelatihan Anda dengan hook Debugger terdaftar — Untuk informasi selengkapnya tentang mendaftarkan hook Debugger ke skrip latihan Anda, lihat. Daftarkan Debugger Hook ke Script Pelatihan Anda

Untuk contoh end-to-end menggunakan Debugger dengan wadah pelatihan kustom, lihat contoh notebook berikut.

- Bangun Wadah Pelatihan Kustom dan Pekerjaan Pelatihan Debug dengan Debugger

> **Tip**
>
> Container kustom dengan panduan Debugger ini merupakan perpanjangan dari Mengadaptasi wadah pelatihan Anda sendiri panduan yang memandu Anda secara menyeluruh bagaimana membangun dan mendorong wadah pelatihan khusus Anda ke Amazon ECR.

Bersiaplah untuk Membangun Wadah Pelatihan Khusus

Untuk membangun wadah buruh pelabuhan, struktur dasar file akan terlihat seperti berikut ini:

```
### debugger_custom_container_test_notebook.ipynb      # a notebook to run python
 snippet codes
### debugger_custom_container_test_folder              # this is a docker folder
    ###  your-training-script.py                       # your training script with
 Debugger hook
    ###  Dockerfile                                    # a Dockerfile to build your own
 container
```

Daftarkan Debugger Hook ke Script Pelatihan Anda

Untuk men-debug pelatihan model Anda, Anda perlu menambahkan hook Debugger ke skrip pelatihan Anda.

> **Note**
>
> Langkah ini diperlukan untuk mengumpulkan parameter model (tensor keluaran) untuk men-debug pelatihan model Anda. Jika Anda hanya ingin memantau dan profil, Anda dapat melewati langkah pendaftaran kait ini dan mengecualikan `debugger_hook_config` parameter saat membuat estimater.

Contoh kode berikut menunjukkan struktur skrip pelatihan menggunakan model Keras ResNet 50 dan cara meneruskan hook Debugger sebagai callback Keras untuk debugging. Untuk menemukan skrip pelatihan lengkap, lihat skrip TensorFlow pelatihan dengan kait SageMaker Debugger.

```
# An example of training script (your-training-script.py)
import tensorflow.compat.v2 as tf
from tensorflow.keras.applications.resnet50 import ResNet50
import smdebug.tensorflow as smd

def train(batch_size, epoch, model, hook):

    ...
    model.fit(X_train, Y_train,
              batch_size=batch_size,
              epochs=epoch,
              validation_data=(X_valid, Y_valid),
              shuffle=True,

              # smdebug modification: Pass the Debugger hook in the main() as a Keras
 callback
              callbacks=[hook])

def main():
    parser=argparse.ArgumentParser(description="Train resnet50 cifar10")

    # hyperparameter settings
    parser.add_argument(...)

    args = parser.parse_args()

    model=ResNet50(weights=None, input_shape=(32,32,3), classes=10)

    # Add the following line to register the Debugger hook for Keras.
    hook=smd.KerasHook.create_from_json_file()

    # Start the training.
    train(args.batch_size, args.epoch, model, hook)

if __name__ == "__main__":
    main()
```

Untuk informasi selengkapnya tentang mendaftarkan hook Debugger untuk kerangka kerja dan algoritme yang didukung, lihat tautan berikut di pustaka klien SMDebug:

- Kait SMDebug TensorFlow

- Kait SMDebug PyTorch
- SMDebug MXNet kait
- SMDebug XGBoost kait

Dalam contoh skrip pelatihan notebook berikut, Anda dapat menemukan lebih banyak contoh tentang cara menambahkan kait Debugger ke skrip pelatihan dan mengumpulkan tensor keluaran secara rinci:

- Debugger dalam mode skrip dengan kerangka TensorFlow 2.1

  Untuk melihat perbedaan antara menggunakan Debugger dalam Deep Learning Container dan dalam mode skrip, buka notebook ini dan letakkan dan Debugger sebelumnya dalam contoh notebook Deep Learning Container TensorFlow v2.1 berdampingan.

  Dalam mode skrip, bagian konfigurasi kait dihapus dari skrip tempat Anda mengatur estimator. Sebagai gantinya, fitur kait Debugger digabungkan ke dalam skrip pelatihan, skrip ResNetpelatihan TensorFlow Keras dalam mode skrip. Skrip pelatihan mengimpor `smdebug` perpustakaan di lingkungan TensorFlow Keras yang diperlukan untuk berkomunikasi dengan algoritma TensorFlow ResNet 50. Ini juga secara manual mengimplementasikan fungsionalitas `smdebug` hook dengan menambahkan `callbacks=[hook]` argumen di dalam `train` fungsi (di baris 49), dan dengan menambahkan konfigurasi hook manual (dalam baris 89) yang disediakan melalui SageMaker Python SDK.

  Contoh mode skrip ini menjalankan pekerjaan pelatihan dalam kerangka kerja TF 2.1 untuk perbandingan langsung dengan perubahan skrip nol dalam contoh TF 2.1. Manfaat menyiapkan Debugger dalam mode skrip adalah fleksibilitas untuk memilih versi kerangka kerja yang tidak tercakup oleh AWS Deep Learning Containers.
- Menggunakan Amazon SageMaker Debugger dalam PyTorch Container dalam Mode Skrip

  Notebook ini memungkinkan Debugger dalam mode skrip dalam kerangka PyTorch v1.3.1. PyTorchv1.3.1 didukung oleh SageMaker kontainer, dan contoh ini menunjukkan detail tentang cara memodifikasi skrip pelatihan.

  SageMakerPyTorchEstimator sudah dalam mode skrip secara default. Di notebook, garis untuk mengaktifkan `script_mode` tidak termasuk dalam konfigurasi estimator.

  Notebook ini menunjukkan langkah-langkah rinci untuk mengubah skrip PyTorch pelatihan asli ke versi modifikasi untuk mengaktifkan Debugger. Selain itu, contoh ini menunjukkan bagaimana

Anda dapat menggunakan aturan bawaan Debugger untuk mendeteksi masalah pelatihan seperti masalah gradien yang hilang, dan fitur uji coba Debugger untuk memanggil dan menganalisis tensor yang disimpan.

Membuat dan Mengkonfigurasi Dockerfile

Buka SageMaker JupyterLab dan buat folder baru, `debugger_custom_container_test_folder` dalam contoh ini, untuk menyimpan skrip latihan Anda dan`Dockerfile`. Contoh kode berikut adalah `Dockerfile` yang mencakup pujian penting buruh pelabuhan membangun. Tempel kode berikut ke dalam file `Dockerfile` teks dan simpan. Unggah skrip latihan Anda ke folder yang sama.

```
# Specify a docker base image
FROM tensorflow/tensorflow:2.2.0rc2-gpu-py3
RUN /usr/bin/python3 -m pip install --upgrade pip
RUN pip install --upgrade protobuf

# Install required packages to enable the SageMaker Python SDK and the smdebug library
RUN pip install sagemaker-training
RUN pip install smdebug
CMD ["bin/bash"]
```

Jika Anda ingin menggunakan gambar Kontainer Pembelajaran AWS Mendalam yang sudah dibuat sebelumnya, lihat Gambar Kontainer Pembelajaran AWS Mendalam yang Tersedia.

Bangun dan Dorong Kontainer Pelatihan Khusus ke Amazon ECR

Buat notebook uji,`debugger_custom_container_test_notebook.ipynb`, dan jalankan kode berikut di sel notebook. Ini akan mengakses `debugger_byoc_test_docker` direktori, membangun buruh pelabuhan dengan yang ditentukan`algorithm_name`, dan mendorong wadah buruh pelabuhan ke Amazon ECR Anda.

```
import boto3

account_id = boto3.client('sts').get_caller_identity().get('Account')
ecr_repository = 'sagemaker-debugger-mnist-byoc-tf2'
tag = ':latest'

region = boto3.session.Session().region_name

uri_suffix = 'amazonaws.com'
```

```
if region in ['cn-north-1', 'cn-northwest-1']:
    uri_suffix = 'amazonaws.com.cn'
byoc_image_uri = '{}.dkr.ecr.{}.{}/{}'.format(account_id, region, uri_suffix,
 ecr_repository + tag)

!docker build -t $ecr_repository docker
!$(aws ecr get-login --region $region --registry-ids $account_id --no-include-email)
!aws ecr create-repository --repository-name $ecr_repository
!docker tag {ecr_repository + tag} $byoc_image_uri
!docker push $byoc_image_uri
```

> **Tip**
>
> Jika Anda menggunakan salah satu gambar dasar AWS Deep Learning Container, jalankan kode berikut untuk masuk ke Amazon ECR dan akses ke repositori gambar Deep Learning Container.
>
> ```
> ! aws ecr get-login-password --region {region} | docker login --username AWS --
> password-stdin 763104351884.dkr.ecr.us-east-1.amazonaws.com
> ```

Jalankan dan Debug Training Jobs Menggunakan Custom Training Container

Setelah Anda membuat dan mendorong container buruh pelabuhan ke Amazon ECR, konfigurasikan SageMaker estimator dengan skrip latihan Anda dan parameter khusus Debugger. Setelah Anda mengeksekusi`estimator.fit()`, Debugger akan mengumpulkan tensor keluaran, memantaunya, dan mendeteksi masalah pelatihan. Dengan menggunakan tensor yang disimpan, Anda dapat menganalisis pekerjaan pelatihan lebih lanjut dengan menggunakan fitur dan alat `smdebug` inti. Mengonfigurasi alur kerja proses pemantauan aturan Debugger dengan Amazon CloudWatch Events danAWS Lambda, Anda dapat mengotomatiskan proses pekerjaan latihan berhenti setiap kali aturan Debugger menemukan masalah pelatihan.

```
import sagemaker
from sagemaker.estimator import Estimator
from sagemaker.debugger import Rule, DebuggerHookConfig, CollectionConfig, rule_configs

profiler_config=ProfilerConfig(...)
debugger_hook_config=DebuggerHookConfig(...)
rules=[
    Rule.sagemaker(rule_configs.built_in_rule()),
```

```
    ProfilerRule.sagemaker(rule_configs.BuiltInRule())
]

estimator=Estimator(
    image_uri=byoc_image_uri,
    entry_point="./debugger_custom_container_test_folder/your-training-script.py"
    role=sagemaker.get_execution_role(),
    base_job_name='debugger-custom-container-test',
    instance_count=1,
    instance_type='ml.p3.2xlarge',

    # Debugger-specific parameters
    profiler_config=profiler_config,
    debugger_hook_config=debugger_hook_config,
    rules=rules
)

# start training
estimator.fit()
```

Mengonfigurasi Debugger Menggunakan Amazon SageMaker API

Topik sebelumnya berfokus pada penggunaan Debugger melalui Amazon SageMaker Python SDK, yang merupakan pembungkus sekitarAWS SDK for Python (Boto3)dan SageMaker Operasi API. Ini menawarkan pengalaman tingkat tinggi dalam mengakses Amazon SageMaker Operasi API. Jika Anda perlu mengkonfigurasi SageMaker Operasi APIAWSBoto3 atauAWS Command Line Interface(CLI) untuk SDK lain, seperti Java, Go, dan C ++, bagian ini membahas cara mengonfigurasi operasi API tingkat rendah berikut.

Topik

- JSON (AWS CLI)
- AWSBoto3

JSON (AWS CLI)

Amazon SageMaker Debugger built-in aturan dapat dikonfigurasi untuk pekerjaan pelatihan menggunakanDebugHookConfig,DebugRuleConfiguration,ProfilerConfig, danProfilerRuleConfigurationobjek melalui SageMaker CreateTrainingJobOperasi API. Anda perlu menentukan URI gambar yang tepat di`RuleEvaluatorImage`parameter, dan contoh berikut memandu Anda dalam mengatur string JSON untuk mintaCreateTrainingJob.

Kode berikut menunjukkan template JSON lengkap untuk menjalankan pekerjaan pelatihan dengan pengaturan yang diperlukan dan konfigurasi Debugger. Simpan template sebagai file JSON di direktori kerja Anda dan jalankan pekerjaan pelatihan menggunakanAWSCLI. Misalnya, simpan kode berikut sebagai`debugger-training-job-cli.json`.

> **Note**
>
> Pastikan Anda menggunakan image container Docker yang benar. Untuk menemukanAWSGambar Deep Learning Container, lihatDeep Learning Containers. Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihatGunakan Debugger Docker Images untuk Built-in atau Custom Rules.

```
{
   "TrainingJobName": "debugger-aws-cli-test",
   "RoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonSageMaker-
ExecutionRole-YYYYMMDDT123456",
   "AlgorithmSpecification": {
      // Specify a training Docker container image URI (Deep Learning Container or your
 own training container) to TrainingImage.
      "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/tensorflow-
training:2.4.1-gpu-py37-cu110-ubuntu18.04",
      "TrainingInputMode": "File",
      "EnableSageMakerMetricsTimeSeries": false
   },
   "HyperParameters": {
      "sagemaker_program": "entry_point/tf-hvd-train.py",
      "sagemaker_submit_directory": "s3://sagemaker-us-west-2-111122223333/debugger-
boto3-profiling-test/source.tar.gz"
   },
   "OutputDataConfig": {
      "S3OutputPath": "s3://sagemaker-us-west-2-111122223333/debugger-aws-cli-test/
output"
   },
   "DebugHookConfig": {
      "S3OutputPath": "s3://sagemaker-us-west-2-111122223333/debugger-aws-cli-test/
debug-output",
      "CollectionConfigurations": [
         {
            "CollectionName": "losses",
            "CollectionParameters" : {
```

```
                "train.save_interval": "50"
            }
         }
      ]
   },
   "DebugRuleConfigurations": [
      {
         "RuleConfigurationName": "LossNotDecreasing",
         "RuleEvaluatorImage": "895741380848.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
debugger-rules:latest",
         "RuleParameters": {"rule_to_invoke": "LossNotDecreasing"}
      }
   ],
   "ProfilerConfig": {
      "S3OutputPath": "s3://sagemaker-us-west-2-111122223333/debugger-aws-cli-test/
profiler-output",
      "ProfilingIntervalInMilliseconds": 500,
      "ProfilingParameters": {
          "DataloaderProfilingConfig": "{\"StartStep\": 5, \"NumSteps\": 3,
 \"MetricsRegex\": \".*\", }",
          "DetailedProfilingConfig": "{\"StartStep\": 5, \"NumSteps\": 3, }",
          "PythonProfilingConfig": "{\"StartStep\": 5, \"NumSteps\": 3, \"ProfilerName
\": \"cprofile\", \"cProfileTimer\": \"total_time\"}",
          "LocalPath": "/opt/ml/output/profiler/"
      }
   },
   "ProfilerRuleConfigurations": [
      {
         "RuleConfigurationName": "ProfilerReport",
         "RuleEvaluatorImage": "895741380848.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
debugger-rules:latest",
         "RuleParameters": {"rule_to_invoke": "ProfilerReport"}
      }
   ],
   "ResourceConfig": {
      "InstanceType": "ml.p3.8xlarge",
      "InstanceCount": 1,
      "VolumeSizeInGB": 30
   },

   "StoppingCondition": {
      "MaxRuntimeInSeconds": 86400
   }
```

```
}
```

Setelah menyimpan file JSON, jalankan perintah berikut di terminal Anda. (Gunakan ! di awal baris jika Anda menggunakan notebook Jupyter.)

```
aws sagemaker create-training-job --cli-input-json file://debugger-training-job-
cli.json
```

Untuk mengkonfigurasi aturan Debugger untuk parameter model debugging

Contoh kode berikut menunjukkan cara mengkonfigurasi built-in`VanishingGradient`aturan menggunakan ini SageMaker API.

Untuk mengaktifkan Debugger mengumpulkan tensor keluaran

Tentukan konfigurasi kait Debugger sebagai berikut:

```
"DebugHookConfig": {
    "S3OutputPath": "s3://<default-bucket>/<training-job-name>/debug-output",
    "CollectionConfigurations": [
        {
            "CollectionName": "gradients",
            "CollectionParameters" : {
                "save_interval": "500"
            }
        }
    ]
}
```

Ini akan membuat pekerjaan pelatihan menyelamatkan koleksi tensor,`gradients`, setiap`save_interval`dari 500 langkah. Untuk menemukan`CollectionName`nilai, lihatKumpulan Bawaandi dalamDokumentasi pustaka klien SMDebug. Untuk menemukan`CollectionParameters`kunci parameter dan nilai-nilai, lihat`sagemaker.debugger.CollectionConfig`class diSageMaker Dokumentasi Python.

Untuk mengaktifkan aturan Debugger untuk men-debug tensor keluaran

Berikut`DebugRuleConfigurations`Contoh API menunjukkan cara menjalankan`VanishingGradient`aturan pada disimpan`gradients`pengumpulan.

```
"DebugRuleConfigurations": [
```

```
    {
        "RuleConfigurationName": "VanishingGradient",
        "RuleEvaluatorImage": "503895931360.dkr.ecr.us-east-1.amazonaws.com/sagemaker-
debugger-rules:latest",
        "RuleParameters": {
            "rule_to_invoke": "VanishingGradient",
            "threshold": "20.0"
        }
    }
]
```

Dengan konfigurasi seperti yang ada dalam sampel ini, Debugger memulai pekerjaan evaluasi aturan untuk pekerjaan pelatihan Anda menggunakan`VanishingGradient`aturan pada koleksi`gradients`tensor. Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihatGunakan Debugger Docker Images untuk Built-in atau Custom Rules. Untuk menemukan pasangan nilai kunci`RuleParameters`LihatDaftar Aturan Built-in Debugger.

Untuk mengonfigurasi aturan bawaan Debugger untuk pembuatan profil sistem dan metrik kerangka kerja

Contoh kode berikut menunjukkan cara menentukan ProfilerConfig Operasi API untuk memungkinkan pengumpulan metrik sistem dan kerangka kerja.

Untuk mengaktifkan profil Debugger untuk mengumpulkan metrik sistem dan kerangka kerja

Target Step

```
"ProfilerConfig": {
    // Optional. Path to an S3 bucket to save profiling outputs
    "S3OutputPath": "s3://<default-bucket>/<training-job-name>/profiler-output",
    // Available values for ProfilingIntervalInMilliseconds: 100, 200, 500, 1000 (1
 second), 5000 (5 seconds), and 60000 (1 minute) milliseconds.
    "ProfilingIntervalInMilliseconds": 500,
    "ProfilingParameters": {
        "DataloaderProfilingConfig": "{ \"StartStep\": 5, \"NumSteps\": 3,
 \"MetricsRegex\": \".*\" }",
        "DetailedProfilingConfig": "{ \"StartStep\": 5, \"NumSteps\": 3 }",
        // For PythonProfilingConfig,
        // available ProfilerName options: cProfile, Pyinstrument
        // available cProfileTimer options only when using cProfile: cpu, off_cpu,
 total_time
```

```
        "PythonProfilingConfig": "{ \"StartStep\": 5, \"NumSteps\": 3,
 \"ProfilerName\": \"cProfile\", \"cProfileTimer\": \"total_time\" }",
        // Optional. Local path for profiling outputs
        "LocalPath": "/opt/ml/output/profiler/"
    }
}
```

Target Time Duration

```
"ProfilerConfig": {
    // Optional. Path to an S3 bucket to save profiling outputs
    "S3OutputPath": "s3://<default-bucket>/<training-job-name>/profiler-output",
    // Available values for ProfilingIntervalInMilliseconds: 100, 200, 500, 1000 (1
 second), 5000 (5 seconds), and 60000 (1 minute) milliseconds.
    "ProfilingIntervalInMilliseconds": 500,
    "ProfilingParameters": {
        "DataloaderProfilingConfig": "{ \"StartTimeInSecSinceEpoch\": 12345567789,
 \"DurationInSeconds\": 10, \"MetricsRegex\": \".*\" }",
        "DetailedProfilingConfig": "{ \"StartTimeInSecSinceEpoch\": 12345567789,
 \"DurationInSeconds\": 10 }",
        // For PythonProfilingConfig,
        // available ProfilerName options: cProfile, Pyinstrument
        // available cProfileTimer options only when using cProfile: cpu, off_cpu,
 total_time
        "PythonProfilingConfig": "{ \"StartTimeInSecSinceEpoch\": 12345567789,
 \"DurationInSeconds\": 10, \"ProfilerName\": \"cProfile\", \"cProfileTimer\":
 \"total_time\" }",
        // Optional. Local path for profiling outputs
        "LocalPath": "/opt/ml/output/profiler/"
    }
}
```

Untuk mengaktifkan aturan Debugger untuk membuat profil metrik

Contoh kode berikut menunjukkan cara mengonfigurasi`ProfilerReport`aturan.

```
"ProfilerRuleConfigurations": [
    {
        "RuleConfigurationName": "ProfilerReport",
        "RuleEvaluatorImage": "895741380848.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
debugger-rules:latest",
        "RuleParameters": {
```

```
            "rule_to_invoke": "ProfilerReport",
            "CPUBottleneck_cpu_threshold": "90",
            "IOBottleneck_threshold": "90"
        }
    }
]
```

Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihatGunakan Debugger Docker Images untuk Built-in atau Custom Rules. Untuk menemukan pasangan nilai kunci`RuleParameters`LihatDaftar Aturan Built-in Debugger.

Update Debugger Profiling Konfigurasi Menggunakan**`UpdateTrainingJob`**Operasi API

Konfigurasi pembuatan profil debugger dapat diperbarui saat pekerjaan pelatihan Anda berjalan dengan menggunakanUpdateTrainingJobOperasi API. Konfigurasikan baruProfilerConfigdanProfilerRuleConfigurationobjek, dan tentukan nama pekerjaan pelatihan ke`TrainingJobName`Parameter.

```
{
    "ProfilerConfig": {
        "DisableProfiler": boolean,
        "ProfilingIntervalInMilliseconds": number,
        "ProfilingParameters": {
            "string" : "string"
        }
    },
    "ProfilerRuleConfigurations": [
        {
            "RuleConfigurationName": "string",
            "RuleEvaluatorImage": "string",
            "RuleParameters": {
                "string" : "string"
            }
        }
    ],
    "TrainingJobName": "your-training-job-name-YYYY-MM-DD-HH-MM-SS-SSS"
}
```

Tambahkan Konfigurasi Aturan Kustom Debugger ke CreateTrainingJob Operasi API

Aturan khusus dapat dikonfigurasi untuk pekerjaan pelatihan menggunakan DebugHookConfigdan DebugRuleConfigurationbenda-benda di CreateTrainingJobOperasi API. Contoh kode berikut

menunjukkan cara mengonfigurasi kebiasaan`ImproperActivation`ditulis dengan`smdebug`library SageMaker Operasi API. Contoh ini mengasumsikan bahwa Anda telah menulis aturan khususcustom_rules.pyfile dan unggah ke keranjang Amazon S3. Contoh ini menyediakan image Docker yang sudah dibuat sebelumnya yang dapat Anda gunakan untuk menjalankan aturan kustom Anda. Ini terdaftar di[URL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Kustom](). Anda menentukan alamat registri URL untuk image Docker yang sudah dibuat sebelumnya di`RuleEvaluatorImage`Parameter.

```
"DebugHookConfig": {
    "S3OutputPath": "s3://<default-bucket>/<training-job-name>/debug-output",
    "CollectionConfigurations": [
        {
            "CollectionName": "relu_activations",
            "CollectionParameters": {
                "include_regex": "relu",
                "save_interval": "500",
                "end_step": "5000"
            }
        }
    ]
},
"DebugRulesConfigurations": [
    {
        "RuleConfigurationName": "improper_activation_job",
        "RuleEvaluatorImage": "552407032007.dkr.ecr.ap-south-1.amazonaws.com/sagemaker-
debugger-rule-evaluator:latest",
        "InstanceType": "ml.c4.xlarge",
        "VolumeSizeInGB": 400,
        "RuleParameters": {
           "source_s3_uri": "s3://bucket/custom_rules.py",
           "rule_to_invoke": "ImproperActivation",
           "collection_names": "relu_activations"
        }
    }
]
```

Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihat[Gunakan Debugger Docker Images untuk Built-in atau Custom Rules](). Untuk menemukan pasangan nilai kunci`RuleParameters`Lihat[Daftar Aturan Built-in Debugger]().

AWSBoto3

Amazon SageMaker Debugger built-in aturan dapat dikonfigurasi untuk pekerjaan pelatihan menggunakan`create_training_job()`FungsiAWSBoto3 SageMakerklien. Anda perlu menentukan URI gambar yang tepat di`RuleEvaluatorImage`parameter, dan contoh berikut memandu Anda dalam mengatur badan permintaan`create_training_job()`Fungsi.

Kode berikut menunjukkan contoh lengkap tentang cara mengkonfigurasi Debugger untuk`create_training_job()`meminta tubuh dan memulai pekerjaan pelatihan di`us-west-2`, dengan asumsi bahwa naskah pelatihan`entry_point/train.py`disiapkan menggunakan TensorFlow. Untuk menemukan end-to-end contoh notebook, lihatProfiling TensorFlow Job Pelatihan Multi Node Multi GPU dengan Amazon SageMakerDebugger (Debugger).

> **Note**
>
> Pastikan Anda menggunakan image container Docker yang benar. Untuk menemukanAWSGambar Deep Learning Container, lihatDeep Learning Containers. Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihatGunakan Debugger Docker Images untuk Built-in atau Custom Rules.

```
import sagemaker, boto3
import datetime, tarfile

# Start setting up a SageMaker session and a Boto3 SageMaker client
session = sagemaker.Session()
region = session.boto_region_name
bucket = session.default_bucket()

# Upload a training script to a default Amazon S3 bucket of the current SageMaker
 session
source = 'source.tar.gz'
project = 'debugger-boto3-test'

tar = tarfile.open(source, 'w:gz')
tar.add ('entry_point/train.py') # Specify the directory and name of your training
 script
tar.close()

s3 = boto3.client('s3')
s3.upload_file(source, bucket, project+'/'+source)
```

```
# Set up a Boto3 session client for SageMaker
sm = boto3.Session(region_name=region).client("sagemaker")

# Start a training job
sm.create_training_job(
    TrainingJobName='debugger-boto3-'+datetime.datetime.now().strftime('%Y-%m-%d-%H-%M-
%S'),
    HyperParameters={
        'sagemaker_submit_directory': 's3://'+bucket+'/'+project+'/'+source,
        'sagemaker_program': '/entry_point/train.py' # training scrip file location and
 name under the sagemaker_submit_directory
    },
    AlgorithmSpecification={
        # Specify a training Docker container image URI (Deep Learning Container or
 your own training container) to TrainingImage.
        'TrainingImage': '763104351884.dkr.ecr.us-west-2.amazonaws.com/tensorflow-
training:2.4.1-gpu-py37-cu110-ubuntu18.04',
        'TrainingInputMode': 'File',
        'EnableSageMakerMetricsTimeSeries': False
    },
    RoleArn='arn:aws:iam::111122223333:role/service-role/AmazonSageMaker-
ExecutionRole-20201014T161125',
    OutputDataConfig={'S3OutputPath': 's3://'+bucket+'/'+project+'/output'},
    ResourceConfig={
        'InstanceType': 'ml.p3.8xlarge',
        'InstanceCount': 1,
        'VolumeSizeInGB': 30
    },
    StoppingCondition={
        'MaxRuntimeInSeconds': 86400
    },
    DebugHookConfig={
        'S3OutputPath': 's3://'+bucket+'/'+project+'/debug-output',
        'CollectionConfigurations': [
            {
                'CollectionName': 'losses',
                'CollectionParameters' : {
                    'train.save_interval': '500',
                    'eval.save_interval': '50'
                }
            }
        ]
    },
```

```
    DebugRuleConfigurations=[
        {
            'RuleConfigurationName': 'LossNotDecreasing',
            'RuleEvaluatorImage': '895741380848.dkr.ecr.us-west-2.amazonaws.com/
sagemaker-debugger-rules:latest',
            'RuleParameters': {'rule_to_invoke': 'LossNotDecreasing'}
        }
    ],
    ProfilerConfig={
        'S3OutputPath': 's3://'+bucket+'/'+project+'/profiler-output',
        'ProfilingIntervalInMilliseconds': 500,
        'ProfilingParameters': {
            'DataloaderProfilingConfig': '{"StartStep": 5, "NumSteps": 3,
 "MetricsRegex": ".*", }',
            'DetailedProfilingConfig': '{"StartStep": 5, "NumSteps": 3, }',
            'PythonProfilingConfig': '{"StartStep": 5, "NumSteps": 3, "ProfilerName":
 "cprofile", "cProfileTimer": "total_time"}',
            'LocalPath': '/opt/ml/output/profiler/' # Optional. Local path for
 profiling outputs
        }
    },
    ProfilerRuleConfigurations=[
        {
            'RuleConfigurationName': 'ProfilerReport',
            'RuleEvaluatorImage': '895741380848.dkr.ecr.us-west-2.amazonaws.com/
sagemaker-debugger-rules:latest',
            'RuleParameters': {'rule_to_invoke': 'ProfilerReport'}
        }
    ]
)
```

Untuk mengkonfigurasi aturan Debugger untuk parameter model debugging

Contoh kode berikut menunjukkan cara mengkonfigurasi built-in`VanishingGradient`aturan menggunakan ini SageMaker API.

Untuk mengaktifkan Debugger mengumpulkan tensor keluaran

Tentukan konfigurasi kait Debugger sebagai berikut:

```
DebugHookConfig={
    'S3OutputPath': 's3://<default-bucket>/<training-job-name>/debug-output',
    'CollectionConfigurations': [
```

```
        {
            'CollectionName': 'gradients',
            'CollectionParameters' : {
                'train.save_interval': '500',
                'eval.save_interval': '50'
            }
        }
    ]
}
```

Ini akan membuat pekerjaan pelatihan menyimpan koleksi tensor,`gradients`, setiap`save_interval`dari 500 langkah. Untuk menemukan`CollectionName`nilai, lihatKumpulan Bawaandi dalamDokumentasi pustaka klien SMDebug. Untuk menemukan`CollectionParameters`kunci parameter dan nilai-nilai, lihat`sagemaker.debugger.CollectionConfig`class diSageMaker Dokumentasi Python.

Untuk mengaktifkan aturan Debugger untuk men-debug tensor keluaran

Berikut`DebugRuleConfigurations`Contoh API menunjukkan cara menjalankan`VanishingGradient`aturan pada disimpan`gradients`pengumpulan.

```
DebugRuleConfigurations=[
    {
        'RuleConfigurationName': 'VanishingGradient',
        'RuleEvaluatorImage': '895741380848.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
debugger-rules:latest',
        'RuleParameters': {
            'rule_to_invoke': 'VanishingGradient',
            'threshold': '20.0'
        }
    }
]
```

Dengan konfigurasi seperti yang ada dalam sampel ini, Debugger memulai pekerjaan evaluasi aturan untuk pekerjaan pelatihan Anda menggunakan`VanishingGradient`aturan pada koleksi`gradients`tensor. Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihatGunakan Debugger Docker Images untuk Built-in atau Custom Rules. Untuk menemukan pasangan nilai kunci`RuleParameters`LihatDaftar Aturan Built-in Debugger.

Untuk mengonfigurasi aturan bawaan Debugger untuk pembuatan profil sistem dan metrik kerangka kerja

Contoh kode berikut menunjukkan cara menentukan ProfilerConfig Operasi API untuk memungkinkan pengumpulan metrik sistem dan kerangka kerja.

Untuk mengaktifkan profil Debugger untuk mengumpulkan metrik sistem dan kerangka kerja

Target Step

```
ProfilerConfig={
    'S3OutputPath': 's3://<default-bucket>/<training-job-name>/profiler-output', #
 Optional. Path to an S3 bucket to save profiling outputs
    # Available values for ProfilingIntervalInMilliseconds: 100, 200, 500, 1000 (1
 second), 5000 (5 seconds), and 60000 (1 minute) milliseconds.
    'ProfilingIntervalInMilliseconds': 500,
    'ProfilingParameters': {
        'DataloaderProfilingConfig': '{
            "StartStep": 5,
            "NumSteps": 3,
            "MetricsRegex": ".*"
        }',
        'DetailedProfilingConfig': '{
            "StartStep": 5,
            "NumSteps": 3
        }',
        'PythonProfilingConfig': '{
            "StartStep": 5,
            "NumSteps": 3,
            "ProfilerName": "cprofile",  # Available options: cprofile, pyinstrument
            "cProfileTimer": "total_time"  # Include only when using cprofile.
 Available options: cpu, off_cpu, total_time
        }',
        'LocalPath': '/opt/ml/output/profiler/' # Optional. Local path for profiling
 outputs
    }
}
```

Target Time Duration

```
ProfilerConfig={
    'S3OutputPath': 's3://<default-bucket>/<training-job-name>/profiler-output', #
 Optional. Path to an S3 bucket to save profiling outputs
```

```
    # Available values for ProfilingIntervalInMilliseconds: 100, 200, 500, 1000 (1
 second), 5000 (5 seconds), and 60000 (1 minute) milliseconds.
    'ProfilingIntervalInMilliseconds': 500,
    'ProfilingParameters': {
        'DataloaderProfilingConfig': '{
            "StartTimeInSecSinceEpoch": 12345567789,
            "DurationInSeconds": 10,
            "MetricsRegex": ".*"
        }',
        'DetailedProfilingConfig': '{
            "StartTimeInSecSinceEpoch": 12345567789,
            "DurationInSeconds": 10
        }',
        'PythonProfilingConfig': '{
            "StartTimeInSecSinceEpoch": 12345567789,
            "DurationInSeconds": 10,
            "ProfilerName": "cprofile",  # Available options: cprofile, pyinstrument
            "cProfileTimer": "total_time"  # Include only when using cprofile.
 Available options: cpu, off_cpu, total_time
        }',
        'LocalPath': '/opt/ml/output/profiler/' # Optional. Local path for profiling
 outputs
    }
}
```

Untuk mengaktifkan aturan Debugger untuk membuat profil metrik

Contoh kode berikut menunjukkan cara mengonfigurasi`ProfilerReport`aturan.

```
ProfilerRuleConfigurations=[
    {
        'RuleConfigurationName': 'ProfilerReport',
        'RuleEvaluatorImage': '895741380848.dkr.ecr.us-west-2.amazonaws.com/sagemaker-
debugger-rules:latest',
        'RuleParameters': {
            'rule_to_invoke': 'ProfilerReport',
            'CPUBottleneck_cpu_threshold': '90',
            'IOBottleneck_threshold': '90'
        }
    }
]
```

Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihatGunakan Debugger Docker Images untuk Built-in atau Custom Rules. Untuk menemukan pasangan nilai kunci`RuleParameters`LihatDaftar Aturan Built-in Debugger.

Update Debugger Profiling Konfigurasi Menggunakan**`UpdateTrainingJob`**Operasi API

Konfigurasi pembuatan profil debugger dapat diperbarui saat pekerjaan pelatihan Anda berjalan dengan menggunakan`update_training_job()`FungsiAWSBoto3 SageMakerklien. Konfigurasikan baruProfilerConfigdanProfilerRuleConfigurationobjek, dan tentukan nama pekerjaan pelatihan ke`TrainingJobName`Parameter.

```
ProfilerConfig={
    'DisableProfiler': boolean,
    'ProfilingIntervalInMilliseconds': number,
    'ProfilingParameters': {
        'string' : 'string'
    }
},
ProfilerRuleConfigurations=[
    {
        'RuleConfigurationName': 'string',
        'RuleEvaluatorImage': 'string',
        'RuleParameters': {
            'string' : 'string'
        }
    }
],
TrainingJobName='your-training-job-name-YYYY-MM-DD-HH-MM-SS-SSS'
```

Tambahkan Konfigurasi Aturan Kustom Debugger ke CreateTrainingJob Operasi API

Aturan khusus dapat dikonfigurasi untuk pekerjaan pelatihan menggunakan DebugHookConfigdan DebugRuleConfigurationobjek menggunakanAWSBoto3 SageMaker klien`create_training_job()`Fungsi. Contoh kode berikut menunjukkan cara mengonfigurasi kebiasaan`ImproperActivation`ditulis dengan`smdebug`library SageMaker Operasi API. Contoh ini mengasumsikan bahwa Anda telah menulis aturan khususcustom_rules.pyfile dan unggah ke keranjang Amazon S3. Contoh ini menyediakan image Docker yang sudah dibuat sebelumnya yang dapat Anda gunakan untuk menjalankan aturan kustom Anda. Ini terdaftar diURL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Kustom. Anda menentukan alamat registri URL untuk image Docker yang sudah dibuat sebelumnya di`RuleEvaluatorImage`Parameter.

```
DebugHookConfig={
    'S3OutputPath': 's3://<default-bucket>/<training-job-name>/debug-output',
    'CollectionConfigurations': [
        {
            'CollectionName': 'relu_activations',
            'CollectionParameters': {
                'include_regex': 'relu',
                'save_interval': '500',
                'end_step': '5000'
            }
        }
    ]
},
DebugRulesConfigurations=[
    {
        'RuleConfigurationName': 'improper_activation_job',
        'RuleEvaluatorImage': '552407032007.dkr.ecr.ap-south-1.amazonaws.com/sagemaker-
debugger-rule-evaluator:latest',
        'InstanceType': 'ml.c4.xlarge',
        'VolumeSizeInGB': 400,
        'RuleParameters': {
           'source_s3_uri': 's3://bucket/custom_rules.py',
           'rule_to_invoke': 'ImproperActivation',
           'collection_names': 'relu_activations'
        }
    }
]
```

Untuk menemukan daftar lengkap image Docker yang tersedia untuk menggunakan aturan Debugger, lihatGunakan Debugger Docker Images untuk Built-in atau Custom Rules. Untuk menemukan pasangan nilai kunci`RuleParameters`LihatDaftar Aturan Built-in Debugger.

Praktik Terbaik untuk Amazon SageMaker Debugger

Gunakan panduan berikut saat Anda menjalankan pekerjaan pelatihan dengan Debugger.

Topik

- Pilih Framework Machine Learning
- Gunakan Dasbor Wawasan Debugger Studio
- Unduh Laporan Debugger dan Dapatkan Lebih Banyak Wawasan
- Tangkap Data dari Training Job Anda dan Simpan Data ke Amazon S3

- Analisis Data dengan Armada Aturan Built-in Debugger
- Mengambil Tindakan Berdasarkan Status Aturan Bawaan
- Selami jauh ke dalam Data Menggunakan Pustaka Klien SMDebug
- Memantau dan Menganalisis Metrik Job Pelatihan
- Pemanfaatan Sistem Pemantauan dan Deteksi Kemacetan
- Operasi Kerangka Profil
- Tensor Keluaran Model Debugging

Pilih Framework Machine Learning

Anda dapat memilih kerangka pembelajaran mesin dan menggunakan wadah pelatihan SageMaker pra-bangun atau wadah Anda sendiri. Gunakan Debugger untuk mendeteksi masalah pelatihan dan kinerja, dan menganalisis kemajuan pelatihan pekerjaan pelatihan Anda. SageMaker SageMaker memberi Anda opsi untuk menggunakan kontainer pra-bangun yang disiapkan untuk sejumlah lingkungan kerangka pembelajaran mesin untuk melatih model Anda di Amazon EC2. Pekerjaan pelatihan apa pun dapat disesuaikan untuk dijalankan di AWS Deep Learning Containers, wadah SageMaker pelatihan, dan wadah khusus.

Gunakan Dasbor Wawasan Debugger Studio

Dengan dasbor wawasan Studio Debugger, Anda mengendalikan pekerjaan pelatihan Anda. Gunakan dasbor Studio Debugger untuk menjaga performa model Anda di instans Amazon EC2 tetap terkendali dan dioptimalkan. Untuk setiap pekerjaan SageMaker pelatihan yang berjalan di instans Amazon EC2, Debugger memantau pemanfaatan sumber daya dan data keluaran model dasar (nilai kehilangan dan akurasi). Melalui dasbor Studio Debugger, dapatkan wawasan tentang pekerjaan pelatihan Anda dan tingkatkan kinerja pelatihan model Anda. Untuk mempelajari selengkapnya, lihat Amazon SageMaker Debugger UI di Eksperimen Klasik Amazon SageMaker Studio.

Unduh Laporan Debugger dan Dapatkan Lebih Banyak Wawasan

Anda dapat melihat hasil agregat dan mendapatkan wawasan dalam laporan Debugger. Debugger menggabungkan hasil pelatihan dan pembuatan profil yang dikumpulkan dari analisis aturan bawaan ke dalam laporan per pekerjaan pelatihan. Anda dapat menemukan informasi lebih rinci tentang hasil pelatihan Anda melalui laporan Debugger. Untuk mempelajari selengkapnya, lihat SageMaker Laporan interaktif debugger.

Tangkap Data dari Training Job Anda dan Simpan Data ke Amazon S3

Anda dapat menggunakan kait Debugger untuk menyimpan tensor keluaran. Setelah Anda memilih wadah dan kerangka kerja yang sesuai dengan skrip pelatihan Anda, gunakan kait Debugger untuk mengonfigurasi tensor mana yang akan disimpan dan direktori mana yang akan menyimpannya, seperti bucket Amazon S3. Hook Debugger membantu Anda membangun konfigurasi dan menyimpannya di akun Anda untuk digunakan dalam analisis berikutnya, di mana ia diamankan untuk digunakan dengan aplikasi yang paling sensitif terhadap privasi. Untuk mempelajari selengkapnya, lihat Konfigurasikan SageMaker Debugger untuk Menyimpan Tensor.

Analisis Data dengan Armada Aturan Built-in Debugger

Anda dapat menggunakan aturan bawaan Debugger untuk memeriksa tensor secara paralel dengan pekerjaan pelatihan. Untuk menganalisis data kinerja pelatihan, Debugger menyediakan aturan bawaan yang memperhatikan perilaku proses pelatihan yang tidak normal. Misalnya, aturan Debugger mendeteksi masalah saat proses pelatihan mengalami masalah kemacetan sistem atau masalah pelatihan, seperti gradien menghilang, tensor yang meledak, overfitting, atau overtraining. Jika perlu, Anda juga dapat membuat aturan yang disesuaikan dengan membuat definisi aturan dengan kriteria Anda sendiri untuk menentukan masalah pelatihan. Untuk mempelajari lebih lanjut tentang aturan Debugger, lihat Konfigurasikan Aturan Bawaan Debugger petunjuk mendetail tentang penggunaan Amazon Python SageMaker SDK. Untuk daftar lengkap aturan bawaan Debugger, lihat. Daftar Aturan Built-in Debugger Jika Anda ingin membuat aturan khusus, lihatBuat Aturan Kustom Debugger untuk Analisis Job Pelatihan.

Mengambil Tindakan Berdasarkan Status Aturan Bawaan

Anda dapat menggunakan Debugger dengan Amazon CloudWatch Events dan. AWS Lambda Anda dapat mengotomatiskan tindakan berdasarkan status aturan, seperti menghentikan pekerjaan pelatihan lebih awal dan menyiapkan notifikasi melalui email atau teks. Ketika aturan Debugger mendeteksi masalah dan memicu status `"IssuesFound"` evaluasi, CloudWatch Peristiwa mendeteksi perubahan status aturan dan memanggil fungsi Lambda untuk mengambil tindakan. Untuk mengonfigurasi tindakan otomatis ke masalah pelatihan Anda, lihatBuat Tindakan pada Aturan Menggunakan Amazon CloudWatch danAWS Lambda.

Selami jauh ke dalam Data Menggunakan Pustaka Klien SMDebug

Anda dapat menggunakan alat SMDebug untuk mengakses dan menganalisis data pelatihan yang dikumpulkan oleh Debugger. `create_trial`Kelas `TrainingJob` and memuat metrik dan tensor yang disimpan oleh Debugger. Kelas-kelas ini menyediakan metode kelas yang diperluas

untuk menganalisis data secara real time atau setelah pelatihan selesai. Pustaka SMDebug juga menyediakan alat visualisasi: menggabungkan garis waktu metrik kerangka kerja untuk menggabungkan profil yang berbeda, bagan garis, dan peta panas untuk melacak pemanfaatan sistem, dan histogram untuk menemukan outlier durasi langkah. Untuk mempelajari lebih lanjut tentang alat pustaka SMDebug, lihat. Menganalisis data menggunakan pustaka klien Python Debugger

Memantau dan Menganalisis Metrik Job Pelatihan

Amazon CloudWatch mendukung metrik kustom resolusi tinggi, dan resolusi terbaiknya adalah 1 detik. Namun, semakin halus resolusinya, semakin pendek umur metrik. CloudWatch Untuk resolusi frekuensi 1 detik, CloudWatch metrik tersedia selama 3 jam. Untuk informasi selengkapnya tentang resolusi dan umur CloudWatch metrik, lihat GetMetricStatisticsdi Referensi Amazon CloudWatch API.

Jika Anda ingin membuat profil pekerjaan pelatihan Anda dengan resolusi yang lebih baik hingga perincian 100 milidetik (0,1 detik) dan menyimpan metrik pelatihan tanpa batas waktu di Amazon S3 untuk analisis khusus kapan saja, pertimbangkan untuk menggunakan Amazon Debugger. SageMaker SageMaker Debugger menyediakan aturan bawaan untuk secara otomatis mendeteksi masalah pelatihan umum; ia mendeteksi masalah pemanfaatan sumber daya perangkat keras (seperti kemacetan CPU, GPU, dan I/O) dan masalah model non-konvergen (seperti overfit, gradien menghilang, dan tensor yang meledak).

SageMaker Debugger juga menyediakan visualisasi melalui Studio Classic dan laporan profilingnya. Tidak seperti CloudWatch metrik, yang mengakumulasi tingkat pemanfaatan sumber daya inti CPU dan GPU dan merata-ratanya di beberapa instance, Debugger melacak tingkat pemanfaatan setiap inti. Ini memungkinkan Anda mengidentifikasi penggunaan sumber daya perangkat keras yang tidak seimbang saat Anda meningkatkan skala ke cluster komputasi yang lebih besar. Untuk menjelajahi visualisasi Debugger, lihat Panduan Dasbor Wawasan SageMaker Debugger, Panduan Laporan Profil Debugger, dan Menganalisis Data Menggunakan Pustaka Klien SMDebug.

Pemanfaatan Sistem Pemantauan dan Deteksi Kemacetan

Dengan pemantauan Amazon SageMaker Debugger, Anda dapat mengukur pemanfaatan sumber daya sistem perangkat keras untuk instans Amazon EC2. Pemantauan tersedia untuk setiap pekerjaan SageMaker pelatihan yang dibangun dengan estimator SageMaker kerangka kerja (TensorFlow, PyTorch, dan MXNet) dan estimator SageMaker generik SageMaker (algoritme bawaan dan wadah khusus Anda sendiri). Aturan bawaan debugger untuk pemantauan mendeteksi masalah kemacetan sistem dan memberi tahu Anda saat mereka mendeteksi masalah kemacetan.

Untuk mempelajari cara mengaktifkan pemantauan sistem Debugger, lihat Konfigurasikan estimator dengan parameter untuk pembuatan profil dasar menggunakan Amazon SageMaker Modul Python Debugger dan kemudian. Konfigurasikan pengaturan untuk pembuatan profil dasar pemanfaatan sumber daya sistem

Untuk daftar lengkap aturan bawaan yang tersedia untuk pemantauan, lihatAturan bawaan debugger untuk membuat profil pemanfaatan sumber daya sistem perangkat keras (metrik sistem).

Operasi Kerangka Profil

Dengan pembuatan profil Amazon SageMaker Debugger, Anda dapat membuat profil operasi kerangka pembelajaran mendalam. Anda dapat membuat profil pelatihan model Anda dengan wadah SageMaker TensorFlow pelatihan, wadah SageMaker PyTorch kerangka kerja, dan wadah pelatihan Anda sendiri. Dengan menggunakan fitur profiling Debugger, Anda dapat menelusuri operator dan fungsi Python yang dijalankan untuk melakukan pekerjaan pelatihan. Debugger mendukung pembuatan profil terperinci, pembuatan profil Python, profil pemuat data, dan pembuatan profil pelatihan terdistribusi Horovod. Anda dapat menggabungkan garis waktu yang diprofilkan untuk berkorelasi dengan kemacetan sistem. Aturan bawaan debugger untuk membuat profil masalah terkait operasi kerangka kerja jam tangan, termasuk waktu inisialisasi pelatihan yang berlebihan karena pengunduhan data sebelum pelatihan dimulai dan pencilan durasi langkah dalam loop pelatihan.

Untuk mempelajari cara mengkonfigurasi Debugger untuk pembuatan profil kerangka kerja, lihat Konfigurasikan estimator dengan parameter untuk pembuatan profil dasar menggunakan Amazon SageMaker Modul Python Debugger dan kemudian. Konfigurasikan untuk pembuatan profil kerangka kerja

Untuk daftar lengkap aturan bawaan yang tersedia untuk pembuatan profil, lihatAturan bawaan debugger untuk membuat profil metrik kerangka kerja.

Tensor Keluaran Model Debugging

Debugging tersedia untuk kerangka pembelajaran mendalam menggunakan AWS Deep Learning Containers dan wadah SageMaker pelatihan. Untuk versi kerangka kerja yang didukung sepenuhnya (lihat versi diKerangka Kerja dan Algoritma yang Didukung), Debugger secara otomatis mendaftarkan kait untuk mengumpulkan tensor keluaran, dan Anda dapat langsung menjalankan skrip pelatihan Anda. Untuk versi dengan satu tanda bintang, Anda perlu mendaftarkan kait secara manual untuk mengumpulkan tensor. Debugger menyediakan koleksi tensor yang telah dikonfigurasi sebelumnya dengan nama umum yang dapat Anda gunakan di berbagai kerangka kerja. Jika Anda ingin menyesuaikan konfigurasi tensor keluaran, Anda juga dapat menggunakan operasi CollectionConfig

dan DebuggerHookConfig API dan Amazon SageMaker Python SDK untuk mengonfigurasi koleksi tensor Anda sendiri. Aturan bawaan debugger untuk debugging menganalisis tensor keluaran dan mengidentifikasi masalah pengoptimalan model yang menghalangi model Anda meminimalkan fungsi kerugian. Misalnya, aturan mengidentifikasi overfitting, overtraining, loss not decreasing, exploding tensor, dan menghilang gradien.

Untuk mempelajari cara mengonfigurasi Debugger untuk men-debug tensor keluaran, lihat dan kemudian. Langkah 2: Peluncuran dan Debug Training Jobs Menggunakan SageMaker Python SDK Konfigurasikan SageMaker Debugger untuk Menyimpan Tensor

Untuk daftar lengkap aturan bawaan yang tersedia untuk debugging, lihatAturan bawaan debugger untuk men-debug data pelatihan model (tensor keluaran).

Amazon SageMaker Debugger Topik Lanjutan dan Dokumentasi Referensi

Bagian berikut berisi topik lanjutan, dokumentasi referensi untuk operasi API, pengecualian, dan batasan yang diketahui untuk Debugger.

Topik

- Amazon SageMaker Operasi API debugger
- Gunakan Debugger Docker Images untuk Built-in atau Custom Rules
- Amazon SageMaker Pengecualian debugger
- Pertimbangan untuk Amazon SageMaker Debugger
- Statistik Penggunaan Debugger Amazon SageMaker

Amazon SageMaker Operasi API debugger

Amazon SageMaker Debugger memiliki operasi API di beberapa lokasi yang digunakan untuk mengimplementasikan pemantauan dan analisis pelatihan model.

Amazon SageMaker Debugger juga menyediakan open source`sagemaker-debugger`Python SDKyang digunakan untuk mengonfigurasi aturan bawaan, menentukan aturan khusus, dan mendaftarkan kait untuk mengumpulkan data tensor keluaran dari pekerjaan pelatihan.

KlasterAmazon SageMaker Python SDKadalah SDK tingkat tinggi yang berfokus pada eksperimen pembelajaran mesin. SDK dapat digunakan untuk menerapkan aturan bawaan atau kustom yang ditentukan dengan`SMDebug`Pustaka Python untuk memantau dan menganalisis tensor ini menggunakan SageMaker penaksir.

Debugger telah menambahkan operasi dan jenis ke Amazon SageMaker API yang memungkinkan platform untuk menggunakan Debugger saat melatih model dan mengelola konfigurasi input dan output.

- `CreateTrainingJob`dan`UpdateTrainingJob`gunakan API Debugger berikut untuk mengonfigurasi koleksi tensor, aturan, gambar aturan, dan opsi pembuatan profil:
  - `CollectionConfiguration`
  - `DebugHookConfig`
  - `DebugRuleConfiguration`
  - `TensorBoardOutputConfig`
  - `ProfilerConfig`
  - `ProfilerRuleConfiguration`
- `DescribeTrainingJob`memberikan deskripsi lengkap tentang pekerjaan pelatihan, termasuk konfigurasi Debugger berikut dan status evaluasi aturan:
  - `DebugHookConfig`
  - `DebugRuleConfiguration`
  - `DebugRuleEvaluationStatus`
  - `ProfilerConfig`
  - `ProfilerRuleConfiguration`
  - `ProfilerRuleEvaluationStatus`

Operasi API konfigurasi aturan menggunakan SageMaker Memproses fungsionalitas saat menganalisis pelatihan model. Untuk informasi lebih lanjut tentang SageMaker Memproses, lihatMemproses data.

Gunakan Debugger Docker Images untuk Built-in atau Custom Rules

Amazon SageMaker menyediakan dua set gambar Docker untuk aturan: satu set untuk mengevaluasi aturan yang disediakan oleh SageMaker (aturan bawaan) dan satu set untuk mengevaluasi aturan kustom yang disediakan dalam file sumber Python.

Jika Anda menggunakanSDK Python Amazon, Anda cukup menggunakan operasi Debugger API tingkat tinggi SageMaker dengan operasi SageMaker Estimator API, tanpa harus secara manual mengambil gambar Debugger Docker dan mengkonfigurasi`ConfigureTrainingJob`API.

Jika Anda tidak menggunakan SDK SageMaker Python, Anda harus mengambil gambar dasar kontainer pra-dibangun yang relevan untuk aturan Debugger. Debugger Amazon SageMaker menyediakan gambar Docker pra-dibangun untuk aturan bawaan dan kustom, dan gambar disimpan di Amazon Elastic Container Registry (Amazon ECR). Untuk menarik gambar dari repositori Amazon ECR (atau untuk mendorong gambar ke satu), gunakan URL registri nama lengkap gambar menggunakan`CreateTrainingJob`API. SageMaker menggunakan pola URL berikut untuk alamat registri gambar wadah aturan Debugger.

```
<account_id>.dkr.ecr.<Region>.amazonaws.com/<ECR repository name>:<tag>
```

Untuk ID akun di masing-masingAWSWilayah, nama repositori Amazon ECR, dan nilai tag, lihat topik berikut.

Topik

- URL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Built-in
- URL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Kustom

URL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Built-in

Gunakan nilai berikut untuk komponen URL registri untuk gambar yang menyediakan aturan bawaan untuk Debugger Amazon SageMaker. Untuk ID akun, lihat tabel berikut.

Nama repositori ECR: sagemaker-debugger-aturan

Tag: Terbaru

Contoh URL registri lengkap:

`904829902805.dkr.ecr.ap-south-1.amazonaws.com/sagemaker-debugger-rules:latest`

ID Akun untuk Built-in Aturan Container Images olehAWSWilayah

Wilayah	account_id
af-south-1	314341159256
ap-east-1	199566480951
ap-northeast-1	430734990657

Wilayah	account_id
ap-northeast-2	578805364391
ap-south-1	904829902805
ap-southeast-1	972752614525
ap-southeast-2	184798709955
ca-central-1	519511493484
cn-north-1	618459771430
cn-northwest-1	658757709296
eu-central-1	482524230118
eu-north-1	314864569078
eu-south-1	563282790590
eu-west-1	929884845733
eu-west-2	250201462417
eu-west-3	447278800020
me-south-1	986000313247
sa-east-1	818342061345
us-east-1	503895931360
us-east-2	915447279597
us-west-1	685455198987
us-west-2	895741380848
us-gov-west-1	515509971035

URL Registry Debugger Amazon SageMaker untuk Evaluator Aturan Kustom

Gunakan nilai berikut untuk komponen URL registri untuk gambar yang menyediakan evaluator aturan khusus untuk Debugger Amazon SageMaker. Untuk ID akun, lihat tabel berikut.

Nama repositori ECR: sagemaker-debugger-aturan-evaluator

Tag: Terbaru

Contoh URL registri lengkap:

`552407032007.dkr.ecr.ap-south-1.amazonaws.com/sagemaker-debugger-rule-evaluator:latest`

ID Akun untuk Gambar Kontainer Aturan Kustom olehAWSWilayah

Wilayah	account_id
af-south-1	515950693465
ap-east-1	645844755771
ap-northeast-1	670969264625
ap-northeast-2	326368420253
ap-south-1	552407032007
ap-southeast-1	631532610101
ap-southeast-2	445670767460
ca-central-1	105842248657
cn-north-1	617202126805
cn-northwest-1	658559488188
eu-central-1	691764027602
eu-north-1	091235270104
eu-south-1	335033873580

Wilayah	account_id
eu-west-1	606966180310
eu-west-2	074613877050
eu-west-3	224335253976
me-south-1	050406412588
sa-east-1	466516958431
us-east-1	864354269164
us-east-2	840043622174
us-west-1	952348334681
us-west-2	759209512951
us-gov-west-1	515361955729

Amazon SageMaker Pengecualian debugger

Amazon SageMaker Debugger dirancang untuk menyadari bahwa tensor yang diperlukan untuk menjalankan aturan mungkin tidak tersedia di setiap langkah. Akibatnya, ini menimbulkan beberapa pengecualian, yang memungkinkan Anda mengontrol apa yang terjadi ketika tensor hilang. Pengecualian ini tersedia dimodul smdebug.exceptions. Anda dapat mengimpornya sebagai berikut:

```
from smdebug.exceptions import *
```

Pengecualian berikut tersedia:

- `TensorUnavailableForStep`- Tensor yang diminta tidak tersedia untuk langkah tersebut. Ini mungkin berarti bahwa langkah ini mungkin tidak disimpan sama sekali oleh hook, atau bahwa langkah ini mungkin telah menyimpan beberapa tensor tetapi tensor yang diminta bukan bagian darinya. Perhatikan bahwa ketika Anda melihat pengecualian ini, itu berarti tensor ini tidak akan pernah tersedia untuk langkah ini di masa future. Jika tensor memiliki pengurangan yang disimpan untuk langkah tersebut, tensor akan memberi tahu Anda bahwa tensor dapat ditanyakan.

- `TensorUnavailable`— Tensor ini tidak diselamatkan atau belum diselamatkan oleh`smdebug`API. Ini berarti bahwa tensor ini tidak pernah terlihat untuk langkah apa pun`smdebug`.
- `StepUnavailable`- Langkah itu tidak disimpan dan Debugger tidak memiliki data dari langkah.
- `StepNotYetAvailable`— Langkah belum dilihat oleh`smdebug`. Mungkin tersedia di future jika pelatihan masih berlangsung. Debugger secara otomatis memuat data baru saat tersedia.
- `NoMoreData`- Dibesarkan saat pelatihan berakhir. Setelah Anda melihat ini, Anda tahu bahwa tidak ada lagi langkah dan tidak ada lagi tensor yang harus diselamatkan.
- `IndexReaderException`- Pembaca indeks tidak valid.
- `InvalidWorker`- Seorang pekerja dipanggil yang tidak valid.
- `RuleEvaluationConditionMet`— Evaluasi aturan pada langkah tersebut mengakibatkan kondisi terpenuhi.
- `InsufficientInformationForRuleInvocation`- Informasi yang tidak memadai diberikan untuk memohon aturan.

Pertimbangan untuk Amazon SageMaker Debugger

Pertimbangkan hal berikut saat menggunakan Amazon SageMaker Debugger.

Pertimbangan untuk Pelatihan Terdistribusi

Daftar berikut menunjukkan ruang lingkup validitas dan pertimbangan untuk menggunakan Debugger pada pekerjaan pelatihan dengan kerangka pembelajaran mendalam dan berbagai opsi pelatihan terdistribusi.

- Horovod

  Lingkup validitas menggunakan Debugger untuk pekerjaan pelatihan dengan Horovod

Deep learning Framework	Apache MXNet	TensorFlow 1.x	TensorFlow 2.x	TensorFlow 2.x dengan Keras	PyTorch
Memantau kemacetan sistem	Ya	Ya	Ya	Ya	Ya
Membuat Profil	Tidak	Tidak	Tidak	Ya	Ya

Deep learning Framework	Apache MXNet	TensorFlow 1.x	TensorFlow 2.x	TensorFlow 2.x dengan Keras	PyTorch
Operasi kerangka kerja					
Debugging tensor keluaran model	Ya	Ya	Ya	Ya	Ya

- SageMaker data yang didistribusikan secara parallel

  Lingkup validitas menggunakan Debugger untuk pekerjaan pelatihan dengan SageMakerdata yang didistribusikan secara parallel

Deep learning Framework	TensorFlow 2.x	TensorFlow 2.x dengan Keras	PyTorch
Memantau kemacetan sistem	Ya	Ya	Ya
Membuat Profil Operasi kerangka kerja	Tidak*	Tidak**	Ya
Debugging tensor keluaran model	Ya	Ya	Ya

  * Debugger tidak mendukung profil kerangka kerja untuk TensorFlow 2.x.

  ** SageMaker parallel data yang didistribusikan tidak men-support TensorFlow 2.x dengan implementasi Keras.

- SageMaker Model didistribusikan parallel— Debugger tidak men-support SageMaker model didistribusikan pelatihan parallel.

- Pelatihan yang didistribusikan dengan SageMaker pos pemeriksaan- Debugger tidak tersedia untuk pekerjaan pelatihan ketika opsi pelatihan terdistribusi dan SageMaker pos pemeriksaan diaktifkan. Anda mungkin akan melihat kesalahan yang akan Anda lihat seperti berikut ini:

  ```
  SMDebug Does Not Currently Support Distributed Training Jobs With Checkpointing
   Enabled
  ```

  Untuk menggunakan Debugger untuk pekerjaan pelatihan dengan opsi pelatihan terdistribusi, Anda perlu menonaktifkan SageMaker checkpointing dan tambahkan fungsi checkpointing manual ke skrip pelatihan Anda. Untuk informasi lebih lanjut tentang cara menggunakan Debugger dengan opsi pelatihan dan pos pemeriksaan yang didistribusikan, lihatMenggunakan data SageMaker terdistribusi paralel dengan Amazon SageMaker Debugger dan pos pemeriksaandanMenyimpan Pos Pemeriksaan.
- Server parameter- Debugger tidak mendukung pelatihan terdistribusi berbasis server parameter.
- Profiling operasi kerangka pelatihan terdistribusi, sepertiAllReducedoperasi SageMaker data terdistribusi parallel danOperasi horovod, tidak tersedia.

Pertimbangan untuk Sistem Monitoring Bottleneck dan Profiling Framework Operations

- UntukAWS TensorFlow, metrik pemuat data tidak dapat dikumpulkan menggunakan defaultlocal_pathpengaturanFrameworkProfilekelas. Jalur harus dikonfigurasi secara manual dan diakhiri"/". Misalnya:

  ```
  FrameworkProfile(local_path="/opt/ml/output/profiler/")
  ```

- UntukAWS TensorFlow, konfigurasi profil pemuat data tidak dapat diperbarui saat pekerjaan pelatihan sedang berjalan.
- UntukAWS TensorFlow, sebuahNoneTypekesalahan mungkin terjadi ketika Anda menggunakan alat analisis dan contoh notebook dengan TensorFlow 2.3 pekerjaan pelatihan dan opsi pembuatan profil terperinci.
- Pembuatan profil Python dan pembuatan profil terperinci hanya didukung untuk Keras API.
- Untuk mengakses fitur deep profiling untuk TensorFlow dan PyTorch, saat ini Anda harus menentukan yang terbaruAWSgambar wadah pembelajaran mendalam dengan CUDA 11. Misalnya, Anda harus menentukan URI gambar tertentu di TensorFlowdan PyTorch estimator sebagai berikut:
  - Untuk TensorFlow

```
image_uri = f"763104351884.dkr.ecr.{region}.amazonaws.com/tensorflow-
training:2.3.1-gpu-py37-cu110-ubuntu18.04"
```

- Untuk PyTorch

```
image_uri = f"763104351884.dkr.ecr.{region}.amazonaws.com/pytorch-training:1.6.0-
gpu-py36-cu110-ubuntu18.04"
```

Pertimbangan untuk Debugging Model Output Tensor

- Hindari menggunakan operasi API fungsional. Debugger tidak dapat mengumpulkan tensor keluaran model PyTorch dan skrip pelatihan MXNet terdiri dari operasi API fungsional.
  - Debugger tidak dapat mengumpulkan tensor keluaran model dari`torch.nn.functional`Operasi API. Saat Anda menulis PyTorch script pelatihan, dianjurkan untuk menggunakan`torch.nn`modul sebagai gantinya.
  - Debugger tidak dapat mengumpulkan tensor keluaran model dari objek fungsional MXNet di blok hibrida. Misalnya, ReLu Aktivasi (`F.relu`) output tidak dapat dikumpulkan dari contoh berikut`mxnet.gluon.HybridBlock`bersama`F`di dalam`hybrid_forward`fungsi.

```
import mxnet as mx
from mxnet.gluon import HybridBlock, nn

class Model(HybridBlock):
    def __init__(self, **kwargs):
        super(Model, self).__init__(**kwargs)
        # use name_scope to give child Blocks appropriate names.
        with self.name_scope():
            self.dense0 = nn.Dense(20)
            self.dense1 = nn.Dense(20)

    def hybrid_forward(self, F, x):
        x = F.relu(self.dense0(x))
        return F.relu(self.dense1(x))

model = Model()
model.initialize(ctx=mx.cpu(0))
model.hybridize()
model(mx.nd.zeros((10, 10), ctx=mx.cpu(0)))
```

Statistik Penggunaan Debugger Amazon SageMaker

Pertimbangkan hal berikut saat menggunakan laporan yang dibuat otomatis oleh Amazon SageMaker Debugger.

Debugger Profiling Laporan Penggunaan

Untuk semua pekerjaan pelatihan SageMaker, Amazon SageMaker Debugger menjalankanProfilerReportaturan dan autogeneratesSageMaker Laporan profil debugger. ParameterProfilerReportaturan menyediakan file notebook Jupyter (`profiler-report.ipynb`) yang menghasilkan file HTML yang sesuai (`profiler-report.html`).

Debugger mengumpulkan statistik penggunaan laporan profil dengan memasukkan kode di notebook Jupyter yang mengumpulkan unik`ProfilerReport`pekerjaan pengolahan aturan ARN jika pengguna membuka final`profiler-report.html`berkas.

Debugger hanya mengumpulkan informasi tentang apakah pengguna membuka laporan HTML akhir. IniTIDAKmengumpulkan informasi apa pun dari pekerjaan pelatihan, data pelatihan, skrip pelatihan, pekerjaan pemrosesan, log, atau isi dari laporan profil itu sendiri.

Anda dapat memilih untuk tidak mengumpulkan statistik penggunaan menggunakan salah satu opsi berikut ini.

(Direkomendasikan) Opsi 1: Memilih Keluar Sebelum Menjalankan Job Pelatihan

Untuk memilih keluar, Anda perlu menambahkan Debugger berikut`ProfilerReport`konfigurasi aturan untuk permintaan pekerjaan pelatihan Anda.

SageMaker Python SDK

```
estimator=sagemaker.estimator.Estimator(
    ...

    rules=ProfilerRule.sagemaker(
        base_config=rule_configs.ProfilerReport()
        rule_parameters={"opt_out_telemetry": "True"}
    )
)
```

AWS CLI

```
"ProfilerRuleConfigurations": [
```

```
    {
        "RuleConfigurationName": "ProfilerReport-1234567890",
        "RuleEvaluatorImage": "895741380848.dkr.ecr.us-west-2.amazonaws.com/
sagemaker-debugger-rules:latest",
        "RuleParameters": {
            "rule_to_invoke": "ProfilerReport",
            "opt_out_telemetry": "True"
        }
    }
]
```

AWS SDK for Python (Boto3)

```
ProfilerRuleConfigurations=[
    {
        'RuleConfigurationName': 'ProfilerReport-1234567890',
        'RuleEvaluatorImage': '895741380848.dkr.ecr.us-west-2.amazonaws.com/
sagemaker-debugger-rules:latest',
        'RuleParameters': {
            'rule_to_invoke': 'ProfilerReport',
            'opt_out_telemetry': 'True'
        }
    }
]
```

Opsi 2: Memilih Keluar Setelah Job Pelatihan Telah Selesai

Untuk memilih keluar setelah pelatihan selesai, Anda perlu memodifikasi`profiler-report.ipynb`berkas.

> **Note**
>
> HTML melaporkan autogenerated tanpaOpsi 1sudah ditambahkan ke permintaan pekerjaan pelatihan Anda masih melaporkan statistik penggunaan bahkan setelah Anda memilih untuk tidak menggunakanOpsi 2.

1. Ikuti petunjuk tentang mengunduh berkas laporan profil Debugger diUnduh SageMaker Laporan profil debuggerhalaman.

2. Di`/ProfilerReport-1234567890/profiler-output`direktori, terbuka`profiler-report.ipynb`.
3. Tambahkan**`opt_out=True`**ke`setup_profiler_report()`dalam sel kode kelima seperti yang ditunjukkan dalam kode contoh berikut:

```
setup_profiler_report(processing_job_arn, opt_out=True)
```

4. Jalankan sel kode untuk menyelesaikan memilih keluar.

Akses wadah pelatihan AWS Systems Manager untuk debugging jarak jauh

Anda dapat terhubung dengan aman ke wadah SageMaker pelatihan melalui AWS Systems Manager (SSM). Ini memberi Anda akses tingkat shell ke pekerjaan pelatihan debug yang berjalan di dalam wadah. Anda juga dapat mencatat perintah dan tanggapan yang dialirkan ke Amazon CloudWatch. Jika Anda menggunakan Amazon Virtual Private Cloud (VPC) Anda sendiri untuk melatih model, Anda dapat menggunakannya untuk menyiapkan titik akhir VPC AWS PrivateLink untuk SSM dan terhubung ke kontainer secara pribadi melalui SSM.

Anda dapat terhubung ke SageMaker Framework Container atau terhubung ke wadah pelatihan Anda sendiri yang diatur dengan lingkungan SageMaker Pelatihan.

Siapkan izin IAM

Untuk mengaktifkan SSM dalam wadah SageMaker pelatihan Anda, Anda perlu menyiapkan peran IAM untuk kontainer. Agar Anda atau pengguna di AWS akun Anda dapat mengakses wadah pelatihan melalui SSM, Anda perlu menyiapkan pengguna IAM dengan izin untuk menggunakan SSM.

Peran IAM

Untuk wadah SageMaker pelatihan untuk memulai dengan agen SSM, berikan peran IAM dengan izin SSM.

Untuk mengaktifkan debugging jarak jauh untuk pekerjaan pelatihan Anda, SageMaker perlu memulai agen SSM di wadah pelatihan saat pekerjaan pelatihan dimulai. Untuk memungkinkan agen SSM berkomunikasi dengan layanan SSM, tambahkan kebijakan berikut ke peran IAM yang Anda gunakan untuk menjalankan pekerjaan pelatihan Anda.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        }
    ]
 }
```

Pengguna IAM

Tambahkan kebijakan berikut untuk memberikan izin sesi SSM kepada pengguna IAM untuk terhubung ke target SSM. Dalam hal ini, target SSM adalah wadah SageMaker pelatihan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:TerminateSession"
            ],
            "Resource": "*"
        }
    ]
}
```

Anda dapat membatasi pengguna IAM untuk terhubung hanya ke kontainer untuk pekerjaan pelatihan tertentu dengan menambahkan `Condition` kunci, seperti yang ditunjukkan dalam contoh kebijakan berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
                "ssm:StartSession",
                "ssm:TerminateSession"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:target-id": [
                        "sagemaker-training-job:*"
                    ]
                }
            }
        }
    ]
}
```

Anda juga dapat secara eksplisit menggunakan tombol `sagemaker:EnableRemoteDebug` kondisi untuk membatasi debugging jarak jauh. Berikut ini adalah contoh kebijakan bagi pengguna IAM untuk membatasi debugging jarak jauh.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyRemoteDebugInTrainingJob",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:UpdateTrainingJob"
            ],
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "sagemaker:EnableRemoteDebug": false
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Kunci kondisi untuk Amazon SageMaker di Referensi Otorisasi AWS Layanan.

Cara mengaktifkan debugging jarak jauh untuk pekerjaan SageMaker pelatihan

Di bagian ini, pelajari cara mengaktifkan debugging jarak jauh saat memulai atau memperbarui pekerjaan pelatihan di Amazon SageMaker.

SageMaker Python SDK

Menggunakan kelas estimator di SageMaker Python SDK, Anda dapat mengaktifkan atau menonaktifkan debugging jarak jauh menggunakan `enable_remote_debug` parameter atau metode and. `enable_remote_debug()` `disable_remote_debug()`

Untuk mengaktifkan debugging jarak jauh saat Anda membuat pekerjaan pelatihan

Untuk mengaktifkan debugging jarak jauh saat Anda membuat pekerjaan pelatihan baru, setel `enable_remote_debug` parameternya ke`True`. Nilai defaultnya adalah`False`, jadi jika Anda tidak menyetel parameter ini sama sekali, atau Anda secara eksplisit mengaturnya`False`, fungsionalitas debugging jarak jauh dinonaktifkan.

```
import sagemaker

session = sagemaker.Session()

estimator = sagemaker.estimator.Estimator(
    ...,
    sagemaker_session=session,
    image_uri="<your_image_uri>", #must be owned by your organization or Amazon
 DLCs
    role=role,
    instance_type="ml.m5.xlarge",
    instance_count=1,
    output_path=output_path,
    max_run=1800,
    enable_remote_debug=True
)
```

Untuk mengaktifkan debugging jarak jauh dengan memperbarui pekerjaan pelatihan

Dengan menggunakan metode kelas estimator berikut, Anda dapat mengaktifkan atau menonaktifkan debugging jarak jauh saat pekerjaan pelatihan berjalan saat pekerjaan sedang `Downloading` atau. `SecondaryStatus` `Training`

```
# Enable RemoteDebug
estimator.enable_remote_debug()

# Disable RemoteDebug
estimator.disable_remote_debug()
```

AWS SDK for Python (Boto3)

Untuk mengaktifkan debugging jarak jauh saat Anda membuat pekerjaan pelatihan

Untuk mengaktifkan debugging jarak jauh saat Anda membuat pekerjaan pelatihan baru, tetapkan nilai `EnableRemoteDebug` kunci ke `True` dalam `RemoteDebugConfig` parameter.

```
import boto3

sm = boto3.Session(region_name=region).client("sagemaker")

# Start a training job
sm.create_training_job(
    ...,
    TrainingJobName=job_name,
    AlgorithmSpecification={
        // Specify a training Docker container image URI
        // (Deep Learning Container or your own training container) to
 TrainingImage.
        "TrainingImage": "<your_image_uri>",
        "TrainingInputMode": "File"
    },
    RoleArn=iam_role_arn,
    OutputDataConfig=output_path,
    ResourceConfig={
        "InstanceType": "ml.m5.xlarge",
        "InstanceCount": 1,
        "VolumeSizeInGB": 30
    },
    StoppingCondition={
        "MaxRuntimeInSeconds": 86400
    },
    RemoteDebugConfig={
```

```
        "EnableRemoteDebug": True
    }
)
```

Untuk mengaktifkan debugging jarak jauh dengan memperbarui pekerjaan pelatihan

Dengan menggunakan `update_traing_job` API, Anda dapat mengaktifkan atau menonaktifkan debugging jarak jauh saat pekerjaan pelatihan berjalan saat pekerjaan sedang `Downloading` atau`Training`. `SecondaryStatus`

```
# Update a training job
sm.update_training_job(
    TrainingJobName=job_name,
    RemoteDebugConfig={
        "EnableRemoteDebug": True     # True | False
    }
)
```

AWS Command Line Interface (CLI)

Untuk mengaktifkan debugging jarak jauh saat Anda membuat pekerjaan pelatihan

Siapkan file `CreateTrainingJob` permintaan dalam format JSON, sebagai berikut.

```
// train-with-remote-debug.json
{
    "TrainingJobName": job_name,
    "RoleArn": iam_role_arn,
    "AlgorithmSpecification": {
        // Specify a training Docker container image URI (Deep Learning Container or
 your own training container) to TrainingImage.
        "TrainingImage": "<your_image_uri>",
        "TrainingInputMode": "File"
    },
    "OutputDataConfig": {
        "S3OutputPath": output_path
    },
    "ResourceConfig": {
        "InstanceType": "ml.m5.xlarge",
        "InstanceCount": 1,
        "VolumeSizeInGB": 30
    },
    "StoppingCondition": {
```

```
        "MaxRuntimeInSeconds": 86400
    },
    "RemoteDebugConfig": {
        "EnableRemoteDebug": True
    }
}
```

Setelah menyimpan file JSON, jalankan perintah berikut di terminal tempat Anda mengirimkan pekerjaan pelatihan. Contoh perintah berikut mengasumsikan bahwa file JSON diberi nama. `train-with-remote-debug.json` Jika Anda menjalankannya dari notebook Jupyter, tambahkan tanda seru (`!`) ke awal baris.

```
aws sagemaker create-training-job \
    --cli-input-json file://train-with-remote-debug.json
```

Untuk mengaktifkan debugging jarak jauh dengan memperbarui pekerjaan pelatihan

Siapkan file `UpdateTrainingJob` permintaan dalam format JSON, sebagai berikut.

```
// update-training-job-with-remote-debug-config.json
{
    "TrainingJobName": job_name,
    "RemoteDebugConfig": {
        "EnableRemoteDebug": True
    }
}
```

Setelah menyimpan file JSON, jalankan perintah berikut di terminal tempat Anda mengirimkan pekerjaan pelatihan. Contoh perintah berikut mengasumsikan bahwa file JSON diberi nama. `train-with-remote-debug.json` Jika Anda menjalankannya dari notebook Jupyter, tambahkan tanda seru (`!`) ke awal baris.

```
aws sagemaker update-training-job \
    --cli-input-json file://update-training-job-with-remote-debug-config.json
```

Akses wadah pelatihan Anda

Anda dapat mengakses wadah pelatihan saat pekerjaan pelatihan yang sesuai adalah`Training`. `SecondaryStatus` Contoh kode berikut menunjukkan cara memeriksa status pekerjaan pelatihan

Anda menggunakan `DescribeTrainingJob` API, cara memeriksa log masuk pekerjaan pelatihan CloudWatch, dan cara masuk ke wadah pelatihan.

Untuk memeriksa status pekerjaan pelatihan

SageMaker Python SDK

Untuk memeriksa `SecondaryStatus` pekerjaan pelatihan, jalankan kode SDK SageMaker Python berikut.

```
import sagemaker

session = sagemaker.Session()

# Describe the job status
training_job_info = session.describe_training_job(job_name)
print(training_job_info)
```

AWS SDK for Python (Boto3)

Untuk memeriksa pekerjaan pelatihan, jalankan kode SDK for Python (Boto3) berikut. `SecondaryStatus`

```
import boto3

session = boto3.session.Session()
region = session.region_name
sm = boto3.Session(region_name=region).client("sagemaker")

# Describe the job status
sm.describe_training_job(TrainingJobName=job_name)
```

AWS Command Line Interface (CLI)

Untuk memeriksa `SecondaryStatus` pekerjaan pelatihan, jalankan AWS CLI perintah berikut untuk SageMaker.

```
aws sagemaker describe-training-job \
    --training-job-name job_name
```

Untuk menemukan nama host dari wadah pelatihan

Untuk terhubung ke wadah pelatihan melalui SSM, gunakan format ini untuk ID target:`sagemaker-training-job:<training-job-name>_algo-<n>`, di `algo-<n>` mana nama host kontainer. Jika pekerjaan Anda berjalan pada satu contoh, host selalu`algo-1`. Jika Anda menjalankan tugas pelatihan terdistribusi pada beberapa instans, SageMaker buat jumlah host dan aliran log yang sama. Misalnya, jika Anda menggunakan 4 instance, SageMaker membuat,`algo-1`, `algo-2``algo-3`, dan`algo-4`. Anda harus menentukan aliran log mana yang ingin Anda debug, dan nomor host-nya. Untuk mengakses aliran log yang terkait dengan pekerjaan pelatihan, lakukan hal berikut.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Pelatihan, lalu pilih Pekerjaan pelatihan.
3. Dari daftar pekerjaan Pelatihan, pilih pekerjaan pelatihan yang ingin Anda debug. Halaman detail pekerjaan pelatihan terbuka.
4. Di bagian Monitor, pilih Lihat log. Daftar aliran log pekerjaan pelatihan terkait terbuka di CloudWatch konsol.
5. Nama aliran log muncul dalam `<training-job-name>/algo-<n>-<time-stamp>` format, dengan `algo-<n>` mewakili nama host.

Untuk mempelajari selengkapnya tentang cara SageMaker mengelola informasi konfigurasi untuk pelatihan terdistribusi multi-instance, lihat Konfigurasi Pelatihan Terdistribusi.

Untuk mengakses wadah pelatihan

Gunakan perintah berikut di terminal untuk memulai sesi SSM (`aws ssm start-session`) dan sambungkan ke wadah pelatihan.

```
aws ssm start-session --target sagemaker-training-job:<training-job-name>_algo-<n>
```

Misalnya, jika nama pekerjaan pelatihan `training-job-test-remote-debug` dan nama host adalah`algo-1`, ID target menjadi`sagemaker-training-job:training-job-test-remote-debug_algo-1`. Jika output dari perintah ini mirip dengan`Starting session with SessionId:xxxxx`, koneksi berhasil.

Akses SSM dengan AWS PrivateLink

Jika wadah pelatihan Anda berjalan dalam Amazon Virtual Private Cloud yang tidak terhubung ke internet publik, Anda dapat menggunakannya AWS PrivateLink untuk mengaktifkan SSM. AWS PrivateLinkmembatasi semua lalu lintas jaringan antara instans titik akhir Anda, SSM, dan

Amazon EC2 ke jaringan Amazon. Untuk informasi selengkapnya tentang cara mengatur akses SSM denganAWS PrivateLink, lihat Menyiapkan titik akhir VPC Amazon untuk Pengelola Sesi.

Log perintah dan hasil sesi SSM

Setelah mengikuti petunjuk di Buat dokumen preferensi Manajer Sesi (baris perintah), Anda dapat membuat dokumen SSM yang menentukan preferensi Anda untuk sesi SSM. Anda dapat menggunakan dokumen SSM untuk mengonfigurasi opsi sesi, termasuk enkripsi data, durasi sesi, dan pencatatan. Misalnya, Anda dapat menentukan apakah akan menyimpan data log sesi di bucket Amazon Simple Storage Service (Amazon S3) atau di grup Amazon CloudWatch Logs. Anda dapat membuat dokumen yang menentukan preferensi umum untuk semua sesi untuk AWS akun danWilayah AWS, atau dokumen yang menentukan preferensi untuk sesi individual.

Memecahkan masalah dengan memeriksa log kesalahan dari SSM

Amazon SageMaker mengunggah kesalahan dari agen SSM ke CloudWatch Log Anda di grup `/aws/sagemaker/TrainingJobs` log. Aliran log agen SSM diberi nama dalam format ini:. `<job-name>/algo-<n>-<timestamp>/ssm` Misalnya, jika Anda membuat pekerjaan pelatihan dua simpul bernama`training-job-test-remote-debug`, log pekerjaan pelatihan `training-job-test-remote-debug/algo-<n>-<timestamp>` dan beberapa log kesalahan agen SSM akan diunggah ke Log `training-job-test-remote-debug/algo-<n>-<timestamp>/ssm` Anda. CloudWatch Dalam contoh ini, Anda dapat meninjau aliran `*/ssm` log untuk memecahkan masalah SSM.

```
training-job-test-remote-debug/algo-1-1680535238
training-job-test-remote-debug/algo-2-1680535238
training-job-test-remote-debug/algo-1-1680535238/ssm
training-job-test-remote-debug/algo-2-1680535238/ssm
```

Pertimbangan

Pertimbangkan hal berikut saat menggunakan debugging SageMaker jarak jauh.

- Debugging jarak jauh tidak didukung untuk kontainer SageMaker algoritme atau kontainer dari SageMaker onAWS Marketplace.
- Anda tidak dapat memulai sesi SSM untuk kontainer yang memiliki isolasi jaringan diaktifkan karena isolasi mencegah panggilan jaringan keluar.

Catatan rilis untuk kemampuan debugging Amazon SageMaker

Lihat catatan rilis berikut untuk melacak pembaruan terbaru untuk kemampuan debugging Amazon SageMaker.

21 Desember 2023

Fitur baru

Merilis fungsionalitas debugging jarak jauh, kemampuan debugging baru SageMaker yang memberi Anda akses tingkat shell ke wadah pelatihan. Dengan rilis ini, Anda dapat men-debug pekerjaan pelatihan dengan masuk ke wadah pekerjaan yang berjalan pada instance SageMaker HTML. Untuk mempelajari selengkapnya, lihat the section called “Akses wadah pelatihan melalui SSM untuk debugging jarak jauh”.

7 September 2023

Fitur baru

Ditambahkan modul utilitas baru `sagemaker.interactive_apps.tensorboard.TensorBoardApp` yang menyediakan fungsi yang disebut`get_app_url()`. `get_app_url()`Fungsi ini menghasilkan URL yang tidak ditandatangani atau ditetapkan sebelumnya untuk membuka TensorBoard aplikasi di lingkungan apa pun di atau Amazon SageMaker EC2. Ini untuk memberikan pengalaman terpadu bagi pengguna Studio Classic dan non-Studio Classic. Untuk lingkungan Studio Classic, Anda dapat membuka TensorBoard dengan menjalankan `get_app_url()` fungsi apa adanya, atau Anda juga dapat menentukan nama pekerjaan untuk mulai melacak saat TensorBoard aplikasi terbuka. Untuk lingkungan non-Studio Classic, Anda dapat membuka TensorBoard dengan memberikan informasi Domain Anda ke fungsi utilitas. Dengan fungsi ini, terlepas dari di mana atau bagaimana Anda menjalankan kode pelatihan dan meluncurkan pekerjaan pelatihan, Anda dapat langsung mengakses TensorBoard dengan menjalankan `get_app_url` fungsi di notebook atau terminal Jupyter Anda. Fungsionalitas ini tersedia di SageMaker Python SDK v2.184.0 dan yang lebih baru. Untuk informasi selengkapnya, lihat the section called “Cara mengakses TensorBoard di SageMaker”.

4 April 2023

Fitur baru

Dirilis SageMaker dengan TensorBoard, kemampuan yang menjadi tuan rumah TensorBoard SageMaker. TensorBoard tersedia sebagai aplikasi melalui SageMaker Domain, dan platform

SageMaker Pelatihan mendukung pengumpulan data TensorBoard keluaran ke S3 dan memuatnya secara otomatis ke host TensorBoard di SageMaker. Dengan kemampuan ini, Anda dapat menjalankan pekerjaan pelatihan yang disiapkan dengan penulis TensorBoard ringkasan SageMaker, menyimpan file TensorBoard output di Amazon S3, membuka TensorBoard aplikasi langsung dari SageMaker konsol, dan memuat file output menggunakan plugin Pengelola SageMaker Data yang diimplementasikan ke antarmuka yang dihosting TensorBoard . Anda tidak perlu menginstal TensorBoard secara manual dan meng-host secara lokal di SageMaker IDE atau mesin lokal. Untuk mempelajari selengkapnya, lihat the section called “Gunakan TensorBoard”.

16 Maret 2023

Catatan penghentian

SageMaker Debugger menghentikan fitur pembuatan profil kerangka kerja mulai dari 2.11 dan 2.0. TensorFlow PyTorch Anda masih dapat menggunakan fitur ini di versi kerangka kerja dan SDK sebelumnya sebagai berikut.

- SageMaker Python SDK <= v2.130.0
- PyTorch >= v1.6.0, < v2.0
- TensorFlow >= v2.3.1, < v2.11

Dengan penghentian, SageMaker Debugger juga menghentikan dukungan untuk tiga berikut untuk pembuatan profil kerangka kerja. `ProfilerRules`

- MaxInitializationTime
- OverallFrameworkMetrics
- StepOutlier

21 Februari 2023

Perubahan lainnya

- Tab laporan XGBoost telah dihapus dari dasbor profiler SageMaker Debugger. Anda masih dapat mengakses laporan XGBoost dengan mengunduhnya sebagai notebook Jupyter atau file HTML. Untuk informasi selengkapnya, lihat Laporan Pelatihan SageMaker Debugger XGBoost.

- Mulai dari rilis ini, aturan profiler bawaan tidak diaktifkan secara default. Untuk menggunakan aturan SageMaker profiler Debugger untuk mendeteksi masalah komputasi tertentu, Anda perlu menambahkan aturan saat mengonfigurasi peluncur pekerjaan pelatihan. SageMaker

1 Desember 2020

Amazon SageMaker Debugger meluncurkan fitur profil mendalam di re:Invent 2020.

3 Desember 2019

Amazon SageMaker Debugger awalnya diluncurkan di re:Invent 2019.

Profil dan optimalkan kinerja komputasi

Saat melatih model pembelajaran state-of-the-art mendalam yang berkembang pesat dalam ukuran, menskalakan pekerjaan pelatihan model tersebut ke cluster GPU besar dan mengidentifikasi masalah kinerja komputasi dari miliaran dan triliunan operasi dan komunikasi dalam setiap iterasi proses penurunan gradien menjadi tantangan.

SageMaker menyediakan alat profil untuk memvisualisasikan dan mendiagnosis masalah komputasi kompleks yang timbul dari menjalankan pekerjaan pelatihan pada sumber daya komputasi awan. AWS Ada dua opsi pembuatan profil yang SageMaker menawarkan: Amazon SageMaker Profiler dan monitor pemanfaatan sumber daya di Amazon SageMaker Studio Classic. Lihat perkenalan berikut dari dua fungsi untuk mendapatkan wawasan cepat dan pelajari mana yang akan digunakan tergantung pada kebutuhan Anda.

Amazon SageMaker Profiler

Amazon SageMaker Profiler adalah kemampuan pembuatan profil yang dapat digunakan untuk menyelami sumber daya komputasi yang disediakan saat melatih model pembelajaran mendalam, dan mendapatkan visibilitas ke detail tingkat operasi. SageMaker SageMaker Profiler menyediakan modul Python untuk menambahkan anotasi PyTorch di seluruh TensorFlow atau melatih skrip dan mengaktifkan Profiler. SageMaker Anda dapat mengakses modul melalui SageMaker Python SDK dan AWS Deep Learning Containers.

Dengan SageMaker Profiler, Anda dapat melacak semua aktivitas pada CPU dan GPU, seperti pemanfaatan CPU dan GPU, kernel berjalan pada GPU, peluncuran kernel pada CPU, operasi sinkronisasi, operasi memori di seluruh CPU dan GPU, latensi antara peluncuran kernel dan proses yang sesuai, dan transfer data antara CPU dan GPU.

SageMaker Profiler juga menawarkan antarmuka pengguna (UI) yang memvisualisasikan profil, ringkasan statistik peristiwa yang diprofilkan, dan garis waktu pekerjaan pelatihan untuk melacak dan memahami hubungan waktu peristiwa antara GPU dan CPU.

Untuk mempelajari lebih lanjut tentang SageMaker Profiler, lihatthe section called “Gunakan SageMaker Profiler”.

Memantau sumber daya AWS komputasi di Amazon SageMaker Studio Classic

SageMaker juga menyediakan antarmuka pengguna di Studio Classic untuk memantau pemanfaatan sumber daya pada tingkat tinggi, tetapi dengan perincian yang lebih besar dibandingkan dengan metrik pemanfaatan default yang dikumpulkan dari hingga. SageMaker CloudWatch

Untuk pekerjaan pelatihan apa pun yang Anda jalankan dalam SageMaker menggunakan SageMaker Python SDK, SageMaker mulailah membuat profil metrik pemanfaatan sumber daya dasar, seperti pemanfaatan CPU, pemanfaatan GPU, pemanfaatan memori GPU, jaringan, dan waktu tunggu I/O. Ini mengumpulkan metrik pemanfaatan sumber daya ini setiap 500 milidetik.

Dibandingkan dengan CloudWatch metrik Amazon, yang mengumpulkan metrik pada interval 1 detik, fungsionalitas pemantauan SageMaker memberikan perincian yang lebih halus ke dalam metrik pemanfaatan sumber daya hingga interval 100 milidetik (0,1 detik), sehingga Anda dapat menyelam jauh ke dalam metrik pada tingkat operasi atau langkah.

Untuk mengakses dasbor untuk memantau metrik pemanfaatan sumber daya dari pekerjaan pelatihan, lihat UI SageMaker Debugger di Eksperimen Studio. SageMaker

Topik

- Menggunakan Amazon SageMaker Profiler untuk membuat profil aktivitas pada sumber daya AWS komputasi
- Pantau pemanfaatan sumber daya AWS komputasi di Amazon Studio Classic SageMaker
- Catatan rilis untuk kemampuan pembuatan profil Amazon SageMaker

Menggunakan Amazon SageMaker Profiler untuk membuat profil aktivitas pada sumber daya AWS komputasi

Amazon SageMaker Profiler saat ini dalam rilis pratinjau dan tersedia tanpa biaya di didukungW ilayah AWS. Versi Amazon SageMaker Profiler yang tersedia secara umum (jika ada) dapat mencakup fitur dan harga yang berbeda dari yang ditawarkan dalam pratinjau.

Amazon SageMaker Profiler adalah kemampuan Amazon SageMaker yang memberikan tampilan terperinci ke sumber daya AWS komputasi yang disediakan selama pelatihan model pembelajaran mendalam. SageMaker Ini berfokus pada pembuatan profil penggunaan CPU dan GPU, kernel berjalan pada GPU, peluncuran kernel pada CPU, operasi sinkronisasi, operasi memori di seluruh CPU dan GPU, latensi antara peluncuran kernel dan proses yang sesuai, dan transfer data antara CPU dan GPU. SageMaker Profiler juga menawarkan antarmuka pengguna (UI) yang memvisualisasikan profil, ringkasan statistik peristiwa yang diprofilkan, dan garis waktu pekerjaan pelatihan untuk melacak dan memahami hubungan waktu peristiwa antara GPU dan CPU.

> **Note**
>
> SageMaker Profiler mendukung PyTorch dan TensorFlow dan tersedia dalam AWSDeep Learning Containers untuk SageMaker. Untuk mempelajari selengkapnya, lihat the section called “Gambar kerangka kerja yang didukungWilayah AWS,, dan jenis instance”.

Untuk ilmuwan data

Melatih model pembelajaran mendalam pada cluster komputasi besar sering kali memiliki masalah optimasi komputasi, seperti kemacetan, latensi peluncuran kernel, batas memori, dan pemanfaatan sumber daya yang rendah.

Untuk mengidentifikasi masalah kinerja komputasi tersebut, Anda perlu membuat profil lebih dalam ke sumber daya komputasi untuk memahami kernel mana yang memperkenalkan latensi dan operasi mana yang menyebabkan kemacetan. Ilmuwan data dapat mengambil manfaat dari menggunakan UI SageMaker Profiler untuk memvisualisasikan profil rinci pekerjaan pelatihan. UI menyediakan dasbor yang dilengkapi dengan bagan ringkasan dan antarmuka garis waktu untuk melacak setiap peristiwa pada sumber daya komputasi. Ilmuwan data juga dapat menambahkan anotasi khusus untuk melacak bagian-bagian tertentu dari pekerjaan pelatihan menggunakan modul SageMaker Profiler Python.

Untuk administrator

Melalui halaman landing Profiler di SageMaker konsol atau SageMaker Domain, Anda dapat mengelola pengguna aplikasi Profiler jika Anda adalah administrator AWS akun atau SageMaker Domain. Setiap pengguna Domain dapat mengakses aplikasi Profiler mereka sendiri dengan izin yang diberikan. Sebagai administrator SageMaker Domain dan pengguna Domain, Anda dapat membuat dan menghapus aplikasi Profiler yang diberikan tingkat izin yang Anda miliki.

Gambar kerangka kerja yang didukungWilayah AWS,, dan jenis instance

Fitur ini mendukung kerangka kerja pembelajaran mesin berikut danWilayah AWS.

> **Note**
>
> Untuk menggunakan fitur ini, pastikan untuk menginstal setidaknya SDK Python SageMaker versi 2.180.0.

SageMaker gambar kerangka pra-instal dengan Profiler SageMaker

SageMaker Profiler sudah diinstal sebelumnya di AWSDeep Learning Containers berikut untuk. SageMaker

PyTorchgambar

PyTorch versi	AWSURI gambar DLC
2.0.0	*763104351884* .dkr.ecr. <region>. amazonaws.com/pytorch-training:2.0.0-gpu-py310-cu118-ubuntu20.04-sagemaker
1.13.1	*763104351884* .dkr.ecr. <region>. amazonaws.com/pytorch-training:1.13.1-gpu-py39-cu117-ubuntu20.04-sagemaker

TensorFlow gambar

TensorFlow versi	AWSURI gambar DLC
2.13.0	*763104351884* .dkr.ecr. <region>. amazonaws.com/tensorflow-training:2.13.0-gpu-py310-cu118-ubuntu20.04-sagemaker
2.12.0	*763104351884* .dkr.ecr. <region>. amazonaws.com/tensorflow-training:2.12.0-gpu-py310-cu118-ubuntu20.04-sagemaker
2.11.0	*763104351884* .dkr.ecr. <region>. amazonaws.com/tensorflow-training:2.11.0-gpu-py39-cu112-ubuntu20.04-sagemaker

Jika Anda ingin menggunakan SageMaker Profiler untuk gambar kerangka kerja lain atau gambar Docker Anda sendiri, instal SageMaker Profiler menggunakan file biner paket Profiler SageMaker Python yang disediakan di bagian berikut.

SageMaker Profiler Python paket file biner

Jika Anda ingin mengonfigurasi wadah Docker Anda sendiri, gunakan SageMaker Profiler di wadah pra-bangun lainnya untuk PyTorch dan TensorFlow, atau instal paket Profiler SageMaker Python secara lokal, gunakan salah satu file biner berikut. Bergantung pada versi Python dan CUDA di lingkungan Anda, pilih salah satu dari yang berikut ini.

PyTorch

- Python3.8, CUDA 11.3: https://smppy.s3.amazonaws.com/pytorch/cu113/smprof-0.3.334-cp38-cp38-linux_x86_64.whl
- Python3.9, CUDA 11.7: https://smppy.s3.amazonaws.com/pytorch/cu117/smprof-0.3.334-cp39-cp39-linux_x86_64.whl
- Python3.10, CUDA 11.8: https://smppy.s3.amazonaws.com/pytorch/cu118/smprof-0.3.334-cp310-cp310-linux_x86_64.whl
- Python3.10, CUDA 12.1: https://smppy.s3.amazonaws.com/pytorch/cu121/smprof-0.3.334-cp310-cp310-linux_x86_64.whl

TensorFlow

- Python3.9, CUDA 11.2: https://smppy.s3.amazonaws.com/tensorflow/cu112/smprof-0.3.334-cp39-cp39-linux_x86_64.whl
- Python3.10, CUDA 11.8: https://smppy.s3.amazonaws.com/tensorflow/cu118/smprof-0.3.334-cp310-cp310-linux_x86_64.whl

Untuk informasi selengkapnya tentang cara menginstal SageMaker Profiler menggunakan file biner, lihatthe section called “(Opsional) Instal paket SageMaker Profiler Python”.

Wilayah AWS yang Didukung

SageMaker Profiler tersedia di berikut Wilayah AWS ini.

- US East (N. Virginia) (`us-east-1`)
- US East (Ohio) (`us-east-2`)
- US West (Oregon) (`us-west-2`)
- Europe (Frankfurt) (`eu-central-1`)
- Europe (Ireland) (`eu-west-1`)

Tipe instans yang didukung

SageMaker Profiler mendukung pembuatan profil pekerjaan pelatihan pada jenis contoh berikut.

Profil CPU dan GPU

- `ml.g4dn.12xlarge`
- `ml.g5.24xlarge`
- `ml.g5.48xlarge`
- `ml.p3dn.24xlarge`
- `ml.p4de.24xlarge`
- `ml.p4d.24xlarge`
- `ml.p5.48xlarge`

Hanya profil GPU

- `ml.g5.2xlarge`
- `ml.g5.4xlarge`
- `ml.g5.8xlarge`
- `ml.g5.16.xlarge`

Prasyarat

Daftar berikut menunjukkan prasyarat untuk mulai menggunakan Profiler. SageMaker

- SageMaker Domain yang disiapkan dengan Amazon VPC di akun AndaAWS.

  Untuk petunjuk cara menyiapkan Domain, lihat Onboard to Amazon SageMaker Domain menggunakan penyiapan cepat. Anda juga perlu menambahkan profil pengguna Domain untuk pengguna individu untuk mengakses aplikasi UI Profiler. Untuk informasi selengkapnya, lihat Menambahkan dan menghapus profil pengguna SageMaker Domain.
- Daftar berikut adalah set minimum izin untuk menggunakan aplikasi UI Profiler.
  - `sagemaker:CreateApp`
  - `sagemaker:DeleteApp`
  - `sagemaker:DescribeTrainingJob`
  - `sagemaker:Search`
  - `s3:GetObject`
  - `s3:ListBucket`

Mempersiapkan dan menjalankan pekerjaan pelatihan dengan SageMaker Profiler

Menyiapkan untuk menjalankan pekerjaan pelatihan dengan SageMaker Profiler terdiri dari dua langkah: mengadaptasi skrip pelatihan dan mengonfigurasi peluncur pekerjaan SageMaker pelatihan.

Topik

- Langkah 1: Sesuaikan skrip pelatihan Anda menggunakan modul SageMaker Profiler Python
- Langkah 2: Buat estimator SageMaker kerangka kerja dan aktifkan SageMaker Profiler
- (Opsional) Instal paket SageMaker Profiler Python

Langkah 1: Sesuaikan skrip pelatihan Anda menggunakan modul SageMaker Profiler Python

Untuk mulai menangkap kernel berjalan pada GPU saat tugas pelatihan berjalan, modifikasi skrip pelatihan Anda menggunakan modul SageMaker Profiler Python. Impor pustaka `start_profiling()` dan tambahkan `stop_profiling()` metode dan untuk menentukan awal dan akhir pembuatan profil. Anda juga dapat menggunakan anotasi kustom opsional untuk menambahkan penanda dalam skrip pelatihan untuk memvisualisasikan aktivitas perangkat keras selama operasi tertentu di setiap langkah.

Perhatikan bahwa annotator mengekstrak operasi dari GPU. Untuk operasi pembuatan profil di CPU, Anda tidak perlu menambahkan anotasi tambahan apa pun. Profiling CPU juga diaktifkan saat Anda menentukan konfigurasi profil, yang akan Anda praktikkan. the section called “Langkah 2: Buat estimator SageMaker kerangka kerja dan aktifkan SageMaker Profiler”

> **Note**
>
> Membuat profil seluruh pekerjaan pelatihan bukanlah penggunaan sumber daya yang paling efisien. Kami merekomendasikan pembuatan profil paling banyak 300 langkah pekerjaan pelatihan.

> **Important**
>
> Rilis pada 14 Desember 2023 melibatkan perubahan besar. Nama paket SageMaker Profiler Python diubah `smppy` dari menjadi. `smprof` Ini efektif dalam SageMaker Framework Containers untuk TensorFlow v2.12 dan yang lebih baru.
> Jika Anda menggunakan salah satu versi sebelumnya dari SageMaker Framework Containers seperti TensorFlow v2.11.0, paket Profiler SageMaker Python masih tersedia sebagai. `smppy` Jika Anda tidak yakin tentang versi atau nama paket mana yang harus Anda gunakan, ganti pernyataan impor paket SageMaker Profiler dengan cuplikan kode berikut.
>
> ```
> try:
>     import smprof
> except ImportError:
>     # backward-compatability for TF 2.11 and PT 1.13.1 images
>     import smppy as smprof
> ```

Pendekatan 1. Gunakan pengelola konteks `smprof.annotate` untuk membubuhi keterangan fungsi lengkap

Anda dapat membungkus fungsi penuh dengan manajer `smprof.annotate()` konteks. Pembungkus ini direkomendasikan jika Anda ingin membuat profil berdasarkan fungsi alih-alih baris kode. Contoh skrip berikut menunjukkan bagaimana menerapkan manajer konteks untuk membungkus loop pelatihan dan fungsi penuh di setiap iterasi.

```
import smprof

SMProf = smprof.SMProfiler.instance()
config = smprof.Config()
config.profiler = {
    "EnableCuda": "1",
}
SMProf.configure(config)
SMProf.start_profiling()

for epoch in range(args.epochs):
    if world_size > 1:
        sampler.set_epoch(epoch)
    tstart = time.perf_counter()
    for i, data in enumerate(trainloader, 0):
        with smprof.annotate("step_"+str(i)):
            inputs, labels = data
            inputs = inputs.to("cuda", non_blocking=True)
            labels = labels.to("cuda", non_blocking=True)

            optimizer.zero_grad()

            with smprof.annotate("Forward"):
                outputs = net(inputs)
            with smprof.annotate("Loss"):
                loss = criterion(outputs, labels)
            with smprof.annotate("Backward"):
                loss.backward()
            with smprof.annotate("Optimizer"):
                optimizer.step()

SMProf.stop_profiling()
```

Pendekatan 2. Gunakan `smprof.annotation_begin()` dan `smprof.annotation_end()` untuk membubuhi keterangan baris kode tertentu dalam fungsi

Anda juga dapat menentukan anotasi untuk membuat profil baris kode tertentu. Anda dapat mengatur titik awal dan titik akhir pembuatan profil yang tepat pada tingkat baris kode individual, bukan oleh fungsinya. Misalnya, dalam skrip berikut, `step_annotator` didefinisikan pada awal setiap iterasi dan berakhir pada akhir iterasi. Sementara itu, anotator rinci lainnya untuk setiap operasi didefinisikan dan membungkus operasi target di setiap iterasi.

```
import smprof

SMProf = smprof.SMProfiler.instance()
config = smprof.Config()
config.profiler = {
    "EnableCuda": "1",
}
SMProf.configure(config)
SMProf.start_profiling()

for epoch in range(args.epochs):
    if world_size > 1:
        sampler.set_epoch(epoch)
    tstart = time.perf_counter()
    for i, data in enumerate(trainloader, 0):
        step_annotator = smprof.annotation_begin("step_" + str(i))

        inputs, labels = data
        inputs = inputs.to("cuda", non_blocking=True)
        labels = labels.to("cuda", non_blocking=True)
        optimizer.zero_grad()

        forward_annotator = smprof.annotation_begin("Forward")
        outputs = net(inputs)
        smprof.annotation_end(forward_annotator)

        loss_annotator = smprof.annotation_begin("Loss")
        loss = criterion(outputs, labels)
        smprof.annotation_end(loss_annotator)

        backward_annotator = smprof.annotation_begin("Backward")
        loss.backward()
        smprof.annotation_end(backward_annotator)
```

```
        optimizer_annotator = smprof.annotation_begin("Optimizer")
        optimizer.step()
        smprof.annotation_end(optimizer_annotator)

        smprof.annotation_end(step_annotator)

SMProf.stop_profiling()
```

Setelah membuat anotasi dan menyiapkan modul inisiasi profiler, simpan skrip untuk dikirimkan menggunakan peluncur pekerjaan SageMaker pelatihan di Langkah 2 berikut. Peluncur sampel mengasumsikan bahwa skrip pelatihan diberi nama. `train_with_profiler_demo.py`

Langkah 2: Buat estimator SageMaker kerangka kerja dan aktifkan SageMaker Profiler

Prosedur berikut menunjukkan cara menyiapkan estimator SageMaker kerangka kerja untuk pelatihan menggunakan SageMaker Python SDK.

1. Siapkan `profiler_config` objek menggunakan `ProfilerConfig` dan `Profiler` modul sebagai berikut.

   ```
   from sagemaker import ProfilerConfig, Profiler
   profiler_config = ProfilerConfig(
       profile_params = Profiler(cpu_profiling_duration=3600)
   )
   ```

   Berikut ini adalah deskripsi `Profiler` modul dan argumennya.

   - `Profiler`: Modul untuk mengaktifkan SageMaker Profiler dengan pekerjaan pelatihan.
     - `cpu_profiling_duration`(int): Tentukan durasi waktu dalam hitungan detik untuk pembuatan profil pada CPU. Defaultnya adalah 3600 detik.
2. Buat estimator SageMaker kerangka kerja dengan `profiler_config` objek yang dibuat pada langkah sebelumnya. Kode berikut menunjukkan contoh membuat PyTorch estimator. Jika Anda ingin membuat TensorFlow estimator, impor `sagemaker.tensorflow.TensorFlow` sebagai gantinya, dan tentukan salah satu TensorFlowversi yang didukung oleh SageMaker Profiler. Untuk informasi selengkapnya tentang kerangka kerja dan jenis instance yang didukung, lihatthe section called “SageMaker gambar kerangka pra-instal dengan Profiler SageMaker ”.

   ```
   import sagemaker
   from sagemaker.pytorch import PyTorch
   ```

```
estimator = PyTorch(
    framework_version="2.0.0",
    role=sagemaker.get_execution_role(),
    entry_point="train_with_profiler_demo.py", # your training job entry point
    source_dir=source_dir, # source directory for your training script
    output_path=output_path,
    base_job_name="sagemaker-profiler-demo",
    hyperparameters=hyperparameters, # if any
    instance_count=1, # Recommended to test with < 8
    instance_type=ml.p4d.24xlarge,
    profiler_config=profiler_config
)
```

3. Mulai pekerjaan pelatihan dengan menjalankan `fit` metode. Dengan`wait=False`, Anda dapat membungkam log pekerjaan pelatihan dan membiarkannya berjalan di latar belakang.

```
estimator.fit(wait=False)
```

Saat menjalankan pekerjaan pelatihan atau setelah pekerjaan selesai, Anda dapat pergi ke topik berikutnya di the section called “Buka aplikasi UI SageMaker Profiler” dan mulai menjelajahi dan memvisualisasikan profil yang disimpan.

Jika Anda ingin langsung mengakses data profil yang disimpan di bucket Amazon S3, gunakan skrip berikut untuk mengambil URI S3.

```
import os
# This is an ad-hoc function to get the S3 URI
# to where the profile output data is saved
def get_detailed_profiler_output_uri(estimator):
    config_name = None
    for processing in estimator.profiler_rule_configs:
        params = processing.get("RuleParameters", dict())
        rule = config_name = params.get("rule_to_invoke", "")
        if rule == "DetailedProfilerProcessing":
            config_name = processing.get("RuleConfigurationName")
            break
    return os.path.join(
        estimator.output_path,
        estimator.latest_training_job.name,
        "rule-output",
        config_name,
    )
```

```
print(
    f"Profiler output S3 bucket: ",
    get_detailed_profiler_output_uri(estimator)
)
```

(Opsional) Instal paket SageMaker Profiler Python

Untuk menggunakan SageMaker Profiler pada PyTorch atau gambar TensorFlow kerangka kerja yang tidak tercantum dalamthe section called “SageMaker gambar kerangka pra-instal dengan Profiler SageMaker ”, atau pada wadah Docker kustom Anda sendiri untuk pelatihan, Anda dapat menginstal SageMaker Profiler dengan menggunakan salah satu. the section called “SageMaker Profiler Python paket file biner”

Opsi 1: Instal paket SageMaker Profiler saat meluncurkan pekerjaan pelatihan

Jika Anda ingin menggunakan SageMaker Profiler untuk pekerjaan pelatihan menggunakan PyTorch atau TensorFlow gambar yang tidak tercantumthe section called “SageMaker gambar kerangka pra-instal dengan Profiler SageMaker ”, buat `requirements.txt` file dan temukan di bawah jalur yang Anda tentukan ke `source_dir` parameter estimator SageMaker kerangka kerja di Langkah 2. Untuk informasi selengkapnya tentang menyiapkan `requirements.txt` file secara umum, lihat Menggunakan pustaka pihak ketiga dalam dokumentasi SageMaker Python SDK. Dalam `requirements.txt` file, tambahkan salah satu jalur bucket S3 untuk file. the section called “SageMaker Profiler Python paket file biner”

```
# requirements.txt
https://smppy.s3.amazonaws.com/tensorflow/cu112/smprof-0.3.332-cp39-cp39-
linux_x86_64.whl
```

Opsi 2: Instal paket SageMaker Profiler di wadah Docker khusus Anda

Jika Anda menggunakan wadah Docker khusus untuk pelatihan, tambahkan salah satunya the section called “SageMaker Profiler Python paket file biner” ke Dockerfile Anda.

```
# Install the smprof package version compatible with your CUDA version
RUN pip install https://smppy.s3.amazonaws.com/tensorflow/cu112/smprof-0.3.332-cp39-
cp39-linux_x86_64.whl
```

Untuk panduan menjalankan kontainer Docker khusus untuk pelatihan SageMaker secara umum, lihat Mengadaptasi wadah pelatihan Anda sendiri.

Buka aplikasi UI SageMaker Profiler

Anda dapat mengakses aplikasi SageMaker Profiler UI melalui opsi berikut.

Topik

Opsi 1: Luncurkan UI SageMaker Profiler dari halaman detail Domain

Jika Anda memiliki akses ke SageMaker konsol, Anda dapat mengambil opsi ini.

Arahkan ke halaman detail Domain

Prosedur berikut menunjukkan cara menavigasi ke halaman detail Domain.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Domain.
3. Dari daftar Domain, pilih Domain tempat Anda ingin meluncurkan aplikasi SageMaker Profiler.

Luncurkan aplikasi UI SageMaker Profiler

Prosedur berikut menunjukkan cara meluncurkan aplikasi SageMaker Profiler yang dicakup ke profil pengguna.

1. Pada halaman Detail domain, pilih tab Profil pengguna.
2. Identifikasi profil pengguna yang ingin Anda luncurkan aplikasi UI SageMaker Profiler.
3. Pilih Luncurkan untuk profil pengguna yang dipilih, dan pilih Profiler.

Opsi 2: Luncurkan aplikasi UI SageMaker Profiler dari halaman arahan SageMaker Profiler di konsol SageMaker

Prosedur berikut menjelaskan cara meluncurkan aplikasi UI SageMaker Profiler dari halaman arahan SageMaker Profiler di SageMaker konsol. Jika Anda memiliki akses ke SageMaker konsol, Anda dapat mengambil opsi ini.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Profiler.
3. Di bawah Memulai, pilih Domain tempat Anda ingin meluncurkan aplikasi Studio Classic. Jika profil pengguna Anda hanya milik satu Domain, Anda tidak melihat opsi untuk memilih Domain.
4. Pilih profil pengguna yang ingin Anda luncurkan aplikasi UI SageMaker Profiler. Jika tidak ada profil pengguna di Domain, pilih Buat profil pengguna. Untuk informasi selengkapnya tentang membuat profil pengguna baru, lihat Menambahkan dan Menghapus Profil Pengguna.
5. Pilih Open Profiler.

Opsi 3: Gunakan fungsi peluncur aplikasi di SageMaker Python SDK

Jika Anda adalah pengguna SageMaker Domain dan hanya memiliki akses ke SageMaker Studio, Anda dapat mengakses aplikasi SageMaker Profiler UI melalui SageMaker Studio Classic dengan menjalankan `sagemaker.interactive_apps.detail_profiler_app.DetailProfilerApp`fungsinya.

Perhatikan bahwa SageMaker Studio Classic adalah pengalaman UI Studio sebelumnya sebelum re:Invent 2023, dan dimigrasikan sebagai aplikasi ke UI Studio yang baru dirancang di re:Invent 2023. Aplikasi UI SageMaker Profiler tersedia di tingkat SageMaker Domain, dan karenanya memerlukan ID Domain dan nama profil pengguna Anda. Saat ini, `DetailedProfilerApp` fungsi hanya berfungsi dalam aplikasi SageMaker Studio Classic; fungsi tersebut dengan benar mengambil informasi Domain dan profil pengguna dari SageMaker Studio Classic.

Untuk Domain, pengguna Domain, dan Studio yang dibuat sebelum re:invent 2023, Studio Classic akan menjadi pengalaman default kecuali Anda telah memperbaruinya mengikuti petunjuk di Migrating from Amazon Studio Classic. SageMaker Jika ini kasus Anda, tidak ada tindakan lebih lanjut yang diperlukan, dan Anda dapat langsung meluncurkan aplikasi UI SageMaker Profiler dengan menjalankan `DetailProfilerApp` funciton.

Jika Anda membuat Domain dan Studio baru setelah re:Invent 2023, luncurkan aplikasi Studio Classic dalam UI Studio dan kemudian jalankan `DetailProfilerApp` fungsi untuk meluncurkan aplikasi UI Profiler. SageMaker

Perhatikan bahwa `DetailedProfilerApp` fungsi tersebut tidak berfungsi di IDE pembelajaran SageMaker mesin lainnya, seperti JupyterLab aplikasi SageMaker Studio, aplikasi Editor Kode SageMaker Studio, dan instance SageMaker Notebook. Jika Anda menjalankan `DetailedProfilerApp` fungsi di IDE tersebut, ia mengembalikan URL ke halaman arahan Profiler di SageMaker konsol, bukan tautan langsung untuk membuka aplikasi UI Profiler.

Jelajahi data keluaran profil yang divisualisasikan di UI SageMaker Profiler

Bagian ini berjalan melalui UI SageMaker Profiler dan memberikan tips tentang cara menggunakan dan mendapatkan wawasan darinya.

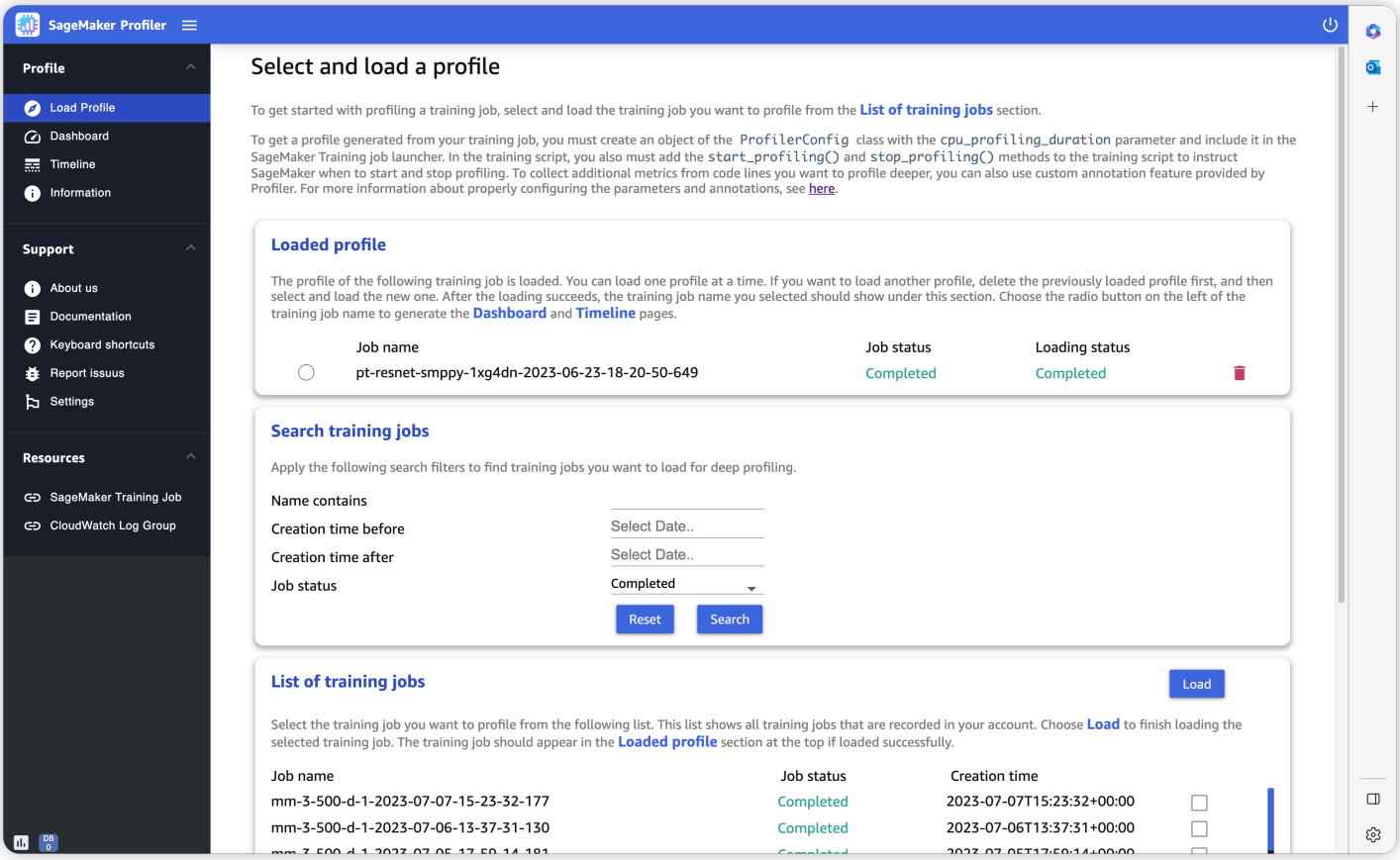
Memuat profil

Saat Anda membuka UI SageMaker Profiler, halaman Load profile akan terbuka. Untuk memuat dan menghasilkan Dashboard dan Timeline, lakukan prosedur berikut.

Untuk memuat profil pekerjaan pelatihan

1. Dari bagian Daftar pekerjaan pelatihan, gunakan kotak centang untuk memilih pekerjaan pelatihan yang ingin Anda muat profilnya.
2. Pilih Muat. Nama pekerjaan akan muncul di bagian Profil yang dimuat di bagian atas.
3. Pilih tombol radio di sebelah kiri nama Job untuk menghasilkan Dashboard dan Timeline. Perhatikan bahwa ketika Anda memilih tombol radio, UI secara otomatis membuka Dasbor. Perhatikan juga bahwa jika Anda menghasilkan visualisasi saat status pekerjaan dan status pemuatan masih tampak sedang berlangsung, UI SageMaker Profiler akan menghasilkan plot Dasbor dan Garis Waktu hingga data profil terbaru yang dikumpulkan dari pekerjaan pelatihan yang sedang berlangsung atau data profil yang dimuat sebagian.

> **Tip**
>
> Anda dapat memuat dan memvisualisasikan satu profil pada satu waktu. Untuk memuat profil lain, Anda harus terlebih dahulu membongkar profil yang dimuat sebelumnya. Untuk membongkar profil, gunakan ikon tempat sampah di ujung kanan profil di bagian Profil yang dimuat.



Dasbor

Setelah Anda selesai memuat dan memilih tugas pelatihan, UI membuka halaman Dasbor yang dilengkapi dengan panel berikut secara default.

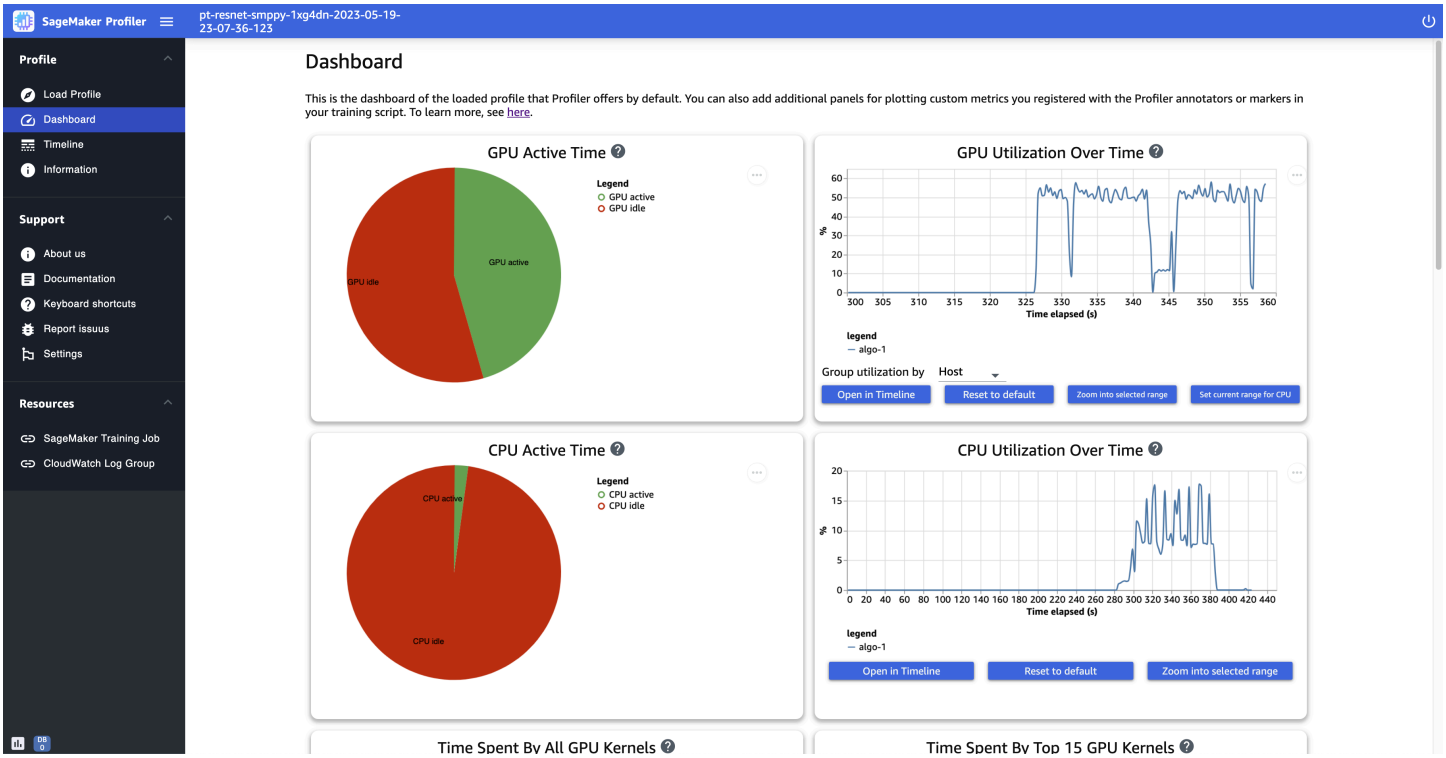
- Waktu aktif GPU - Diagram lingkaran ini menunjukkan persentase waktu aktif GPU versus waktu idle GPU. Anda dapat memeriksa apakah GPU Anda lebih aktif daripada menganggur di seluruh pekerjaan pelatihan. Waktu aktif GPU didasarkan pada titik data profil dengan tingkat pemanfaatan lebih besar dari 0%, sedangkan waktu idle GPU adalah titik data yang diprofilkan dengan pemanfaatan 0%.
- Pemanfaatan GPU dari waktu ke waktu - Grafik garis waktu ini menunjukkan tingkat pemanfaatan GPU rata-rata dari waktu ke waktu per node, menggabungkan semua node dalam satu bagan. Anda dapat memeriksa apakah GPU memiliki beban kerja yang tidak seimbang, masalah pemanfaatan yang kurang, kemacetan, atau masalah idle selama interval waktu tertentu. Untuk melacak tingkat pemanfaatan pada tingkat GPU individu dan menjalankan kernel terkait, gunakan file. the section called “Antarmuka garis waktu” Perhatikan bahwa pengumpulan aktivitas GPU

dimulai dari tempat Anda menambahkan fungsi starter profiler `SMProf.start_profiling()` dalam skrip latihan, dan berhenti di. `SMProf.stop_profiling()`

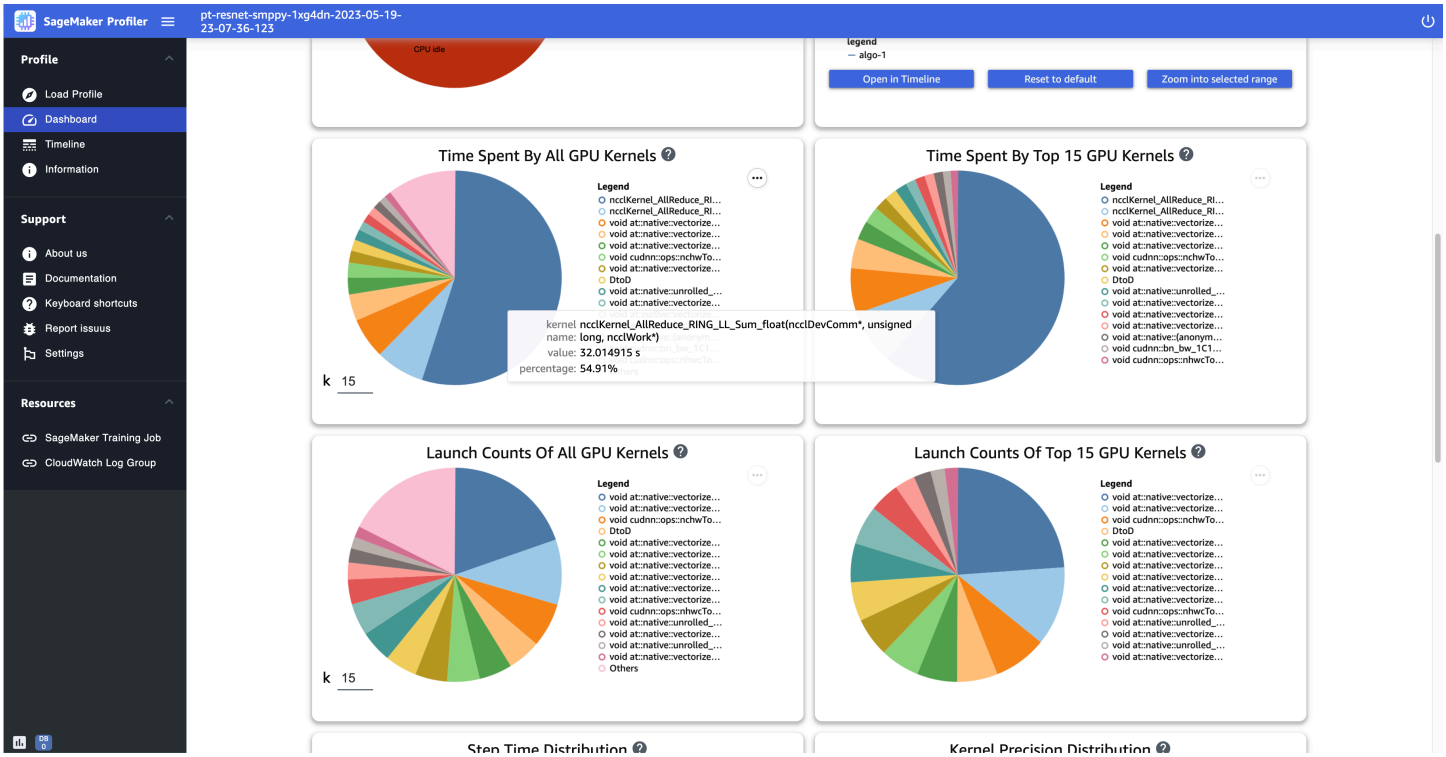
- Waktu aktif CPU - Bagan lingkaran ini menunjukkan persentase waktu aktif CPU versus waktu idle CPU. Anda dapat memeriksa apakah CPU Anda lebih aktif daripada menganggur di seluruh pekerjaan pelatihan. Waktu aktif CPU didasarkan pada titik data yang diprofilkan dengan tingkat pemanfaatan lebih besar dari 0%, sedangkan waktu idle CPU adalah titik data yang diprofilkan dengan pemanfaatan 0%.
- Pemanfaatan CPU dari waktu ke waktu — Grafik garis waktu ini menunjukkan tingkat pemanfaatan CPU rata-rata dari waktu ke waktu per node, menggabungkan semua node dalam satu bagan. Anda dapat memeriksa apakah CPU mengalami kemacetan atau kurang dimanfaatkan selama interval waktu tertentu. Untuk melacak tingkat pemanfaatan CPU yang selaras dengan pemanfaatan GPU individu dan menjalankan kernel, gunakan file. the section called “Antarmuka garis waktu” Perhatikan bahwa metrik pemanfaatan dimulai dari awal dari inisialisasi pekerjaan.
- Waktu yang dihabiskan oleh semua kernel GPU — Bagan lingkaran ini menunjukkan semua kernel GPU yang dioperasikan selama pekerjaan pelatihan. Ini menunjukkan 15 kernel GPU teratas secara default sebagai sektor individu dan semua kernel lainnya dalam satu sektor. Arahkan kursor ke sektor untuk melihat informasi lebih rinci. Nilai menunjukkan total waktu kernel GPU yang dioperasikan dalam hitungan detik, dan persentasenya didasarkan pada seluruh waktu profil.
- Waktu yang dihabiskan oleh 15 kernel GPU teratas — Bagan lingkaran ini menunjukkan semua kernel GPU yang dioperasikan selama pekerjaan pelatihan. Ini menunjukkan 15 kernel GPU teratas sebagai sektor individual. Arahkan kursor ke sektor untuk melihat informasi lebih rinci. Nilai menunjukkan total waktu kernel GPU yang dioperasikan dalam hitungan detik, dan persentasenya didasarkan pada seluruh waktu profil.
- Jumlah peluncuran semua kernel GPU - Bagan lingkaran ini menunjukkan jumlah hitungan untuk setiap kernel GPU yang diluncurkan selama pekerjaan pelatihan. Ini menunjukkan 15 kernel GPU teratas sebagai sektor individu dan semua kernel lainnya dalam satu sektor. Arahkan kursor ke sektor untuk melihat informasi lebih rinci. Nilai menunjukkan jumlah total kernel GPU yang diluncurkan, dan persentasenya didasarkan pada seluruh hitungan semua kernel.
- Jumlah peluncuran dari 15 kernel GPU teratas - Bagan lingkaran ini menunjukkan jumlah hitungan setiap kernel GPU yang diluncurkan selama pekerjaan pelatihan. Ini menunjukkan 15 kernel GPU teratas. Arahkan kursor ke sektor untuk melihat informasi lebih rinci. Nilai menunjukkan jumlah total kernel GPU yang diluncurkan, dan persentasenya didasarkan pada seluruh hitungan semua kernel.
- Distribusi waktu langkah - Histogram ini menunjukkan distribusi durasi langkah pada GPU. Plot ini dibuat hanya setelah Anda menambahkan anotator langkah dalam skrip pelatihan Anda.

- Distribusi presisi kernel - Diagram lingkaran ini menunjukkan persentase waktu yang dihabiskan untuk menjalankan kernel dalam tipe data yang berbeda seperti FP32, FP16, INT32, dan INT8.
- Distribusi aktivitas GPU - Diagram lingkaran ini menunjukkan persentase waktu yang dihabiskan untuk aktivitas GPU, seperti menjalankan kernel, memori (`memcpy`dan`memset`), dan sinkronisasi (). `sync`
- Distribusi operasi memori GPU - Diagram lingkaran ini menunjukkan persentase waktu yang dihabiskan untuk operasi memori GPU. Ini memvisualisasikan `memcopy` kegiatan dan membantu mengidentifikasi apakah pekerjaan pelatihan Anda menghabiskan waktu berlebihan untuk operasi memori tertentu.
- Buat histogram baru — Buat diagram baru dari metrik khusus yang Anda anotasi secara manual selama. the section called “Langkah 1: Sesuaikan skrip pelatihan Anda menggunakan modul SageMaker Profiler Python” Saat menambahkan anotasi khusus ke histogram baru, pilih atau ketik nama anotasi yang Anda tambahkan dalam skrip pelatihan. Misalnya, dalam skrip pelatihan demo di Langkah 1,,`step`,`Forward`, `Backward0ptimize`, dan `Loss` merupakan anotasi khusus. Saat membuat histogram baru, nama anotasi ini akan muncul di menu tarik-turun untuk pemilihan metrik. Jika Anda memilih**`Backward`**, UI menambahkan histogram waktu yang dihabiskan untuk lintasan mundur sepanjang waktu yang diprofilkan ke Dasbor. Jenis histogram ini berguna untuk memeriksa apakah ada outlier yang memakan waktu lebih lama secara tidak normal dan menyebabkan masalah bottleneck.

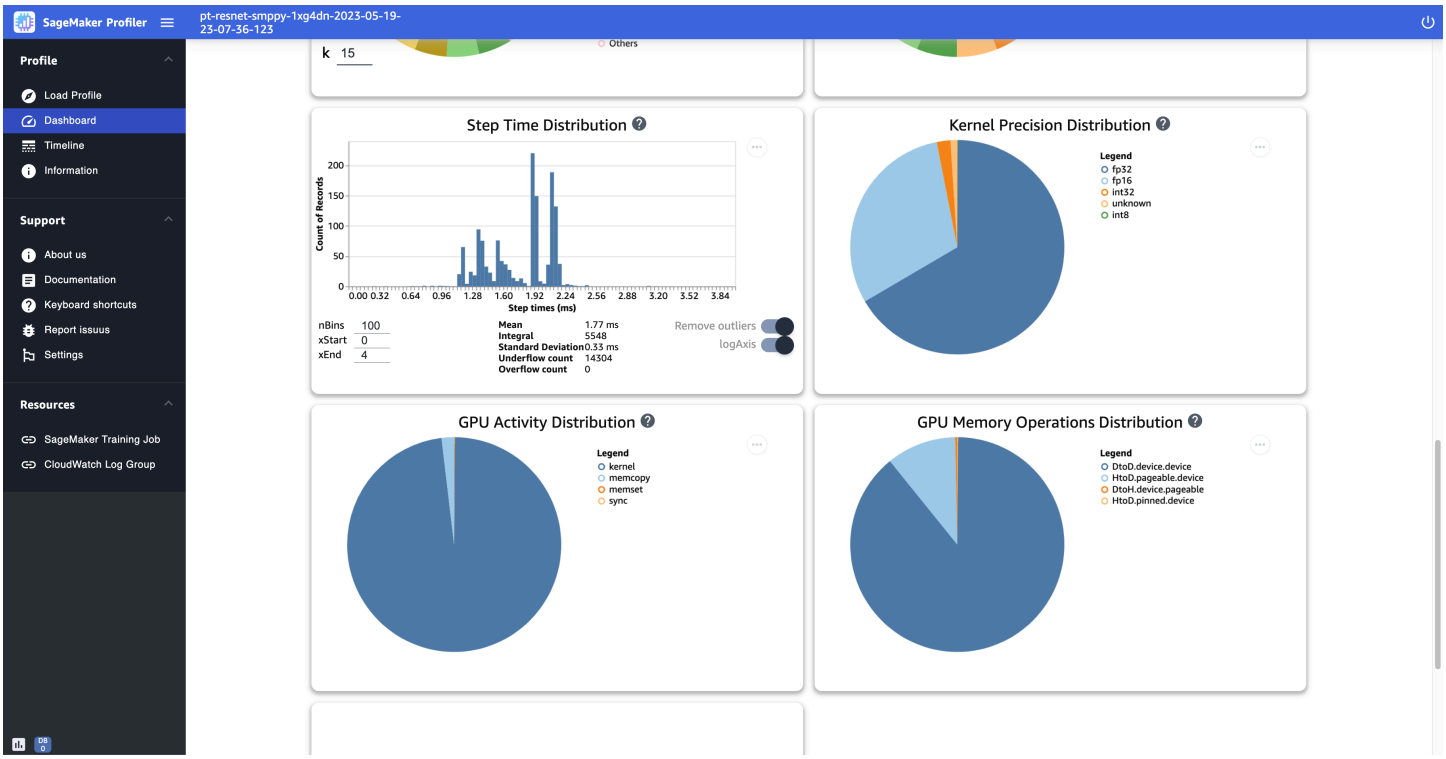
Tangkapan layar berikut menunjukkan rasio waktu aktif GPU dan CPU serta tingkat pemanfaatan GPU dan CPU rata-rata sehubungan dengan waktu per node komputasi.



Tangkapan layar berikut menunjukkan contoh diagram lingkaran untuk membandingkan berapa kali kernel GPU diluncurkan dan mengukur waktu yang dihabiskan untuk menjalankannya. Dalam Waktu yang dihabiskan oleh semua kernel GPU dan jumlah Peluncuran semua panel kernel GPU, Anda juga dapat menentukan bilangan bulat ke bidang input untuk k untuk menyesuaikan jumlah legenda yang akan ditampilkan di plot. Misalnya, jika Anda menentukan 10, plot menunjukkan 10 kernel teratas yang paling banyak dijalankan dan diluncurkan masing-masing.



Tangkapan layar berikut menunjukkan contoh histogram durasi waktu langkah, dan diagram lingkaran untuk distribusi presisi kernel, distribusi aktivitas GPU, dan distribusi operasi memori GPU.



Antarmuka garis waktu

Untuk mendapatkan tampilan rinci ke sumber daya komputasi pada tingkat operasi dan kernel yang dijadwalkan pada CPU dan berjalan di GPU, gunakan antarmuka Timeline.

Anda dapat memperbesar dan memperkecil dan menggeser ke kiri atau kanan di antarmuka garis waktu menggunakan mouse, `[w, a, s, d]` tombol, atau empat tombol panah pada keyboard.

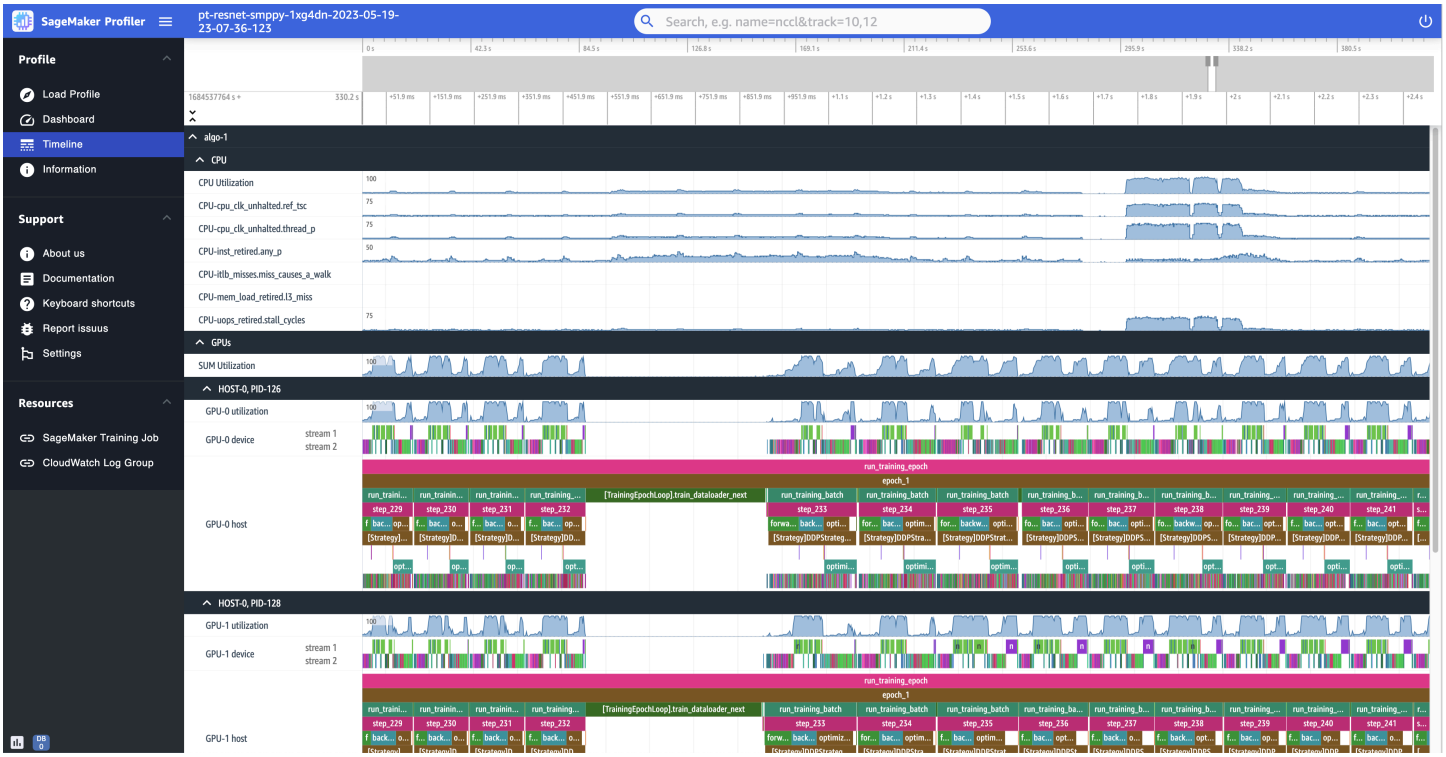
> **Tip**
>
> Untuk tips lebih lanjut tentang pintasan keyboard untuk berinteraksi dengan antarmuka Timeline, pilih Pintasan keyboard di panel kiri.

Trek garis waktu diatur dalam struktur pohon, memberi Anda informasi dari tingkat host ke tingkat perangkat. Misalnya, jika Anda menjalankan `N` instance dengan delapan GPU di masing-masing, struktur garis waktu setiap instance adalah sebagai berikut.

- algo-i $_{node}$ — Inilah SageMaker tag untuk menetapkan pekerjaan ke instance yang disediakan. Digit i $_{node}$ ditetapkan secara acak. Misalnya, jika Anda menggunakan 4 instance, bagian ini diperluas dari algo-1 ke algo-4.
  - CPU - Di bagian ini, Anda dapat memeriksa tingkat pemanfaatan CPU rata-rata dan penghitung kinerja.
  - GPU — Di bagian ini, Anda dapat memeriksa tingkat pemanfaatan GPU rata-rata, tingkat pemanfaatan GPU individu, dan kernel.
    - Pemanfaatan SUM — Tingkat pemanfaatan GPU rata-rata per instans.
    - HOST-0 PID-123 — Nama unik yang ditetapkan untuk setiap trek proses. Akronim PID adalah ID proses, dan nomor yang ditambahkan padanya adalah nomor ID proses yang direkam selama pengambilan data dari proses. Bagian ini menunjukkan informasi berikut dari proses.
      - $_{num_gpu}$Pemanfaatan GPU-i — Tingkat pemanfaatan GPU ke-i dari waktu ke waktu$_{num_gpu}$.
      - $_{num_gpu}$Perangkat GPU-i — Kernel berjalan pada perangkat GPU $_{num\_gpu}$ ke-i.
        - stream i $_{cuda\_stream}$ — CUDA stream yang menunjukkan kernel berjalan pada perangkat GPU. Untuk mempelajari lebih lanjut tentang aliran CUDA, lihat slide dalam PDF di CUDA C/C++ Streams and Concurrency yang disediakan oleh NVIDIA.
      - Host GPU-i - Kernel diluncurkan pada $_{num\_gpu}$ host GPU $_{num\_gpu}$ ke-i.

Beberapa tangkapan layar berikut menunjukkan Timeline profil pekerjaan pelatihan yang dijalankan pada `ml.p4d.24xlarge` instance, yang masing-masing dilengkapi dengan 8 GPU Tensor Core NVIDIA A100.

Berikut ini adalah tampilan profil yang diperbesar, mencetak selusin langkah termasuk pemuat data intermiten antara `step_232` dan `step_233` untuk mengambil kumpulan data berikutnya.



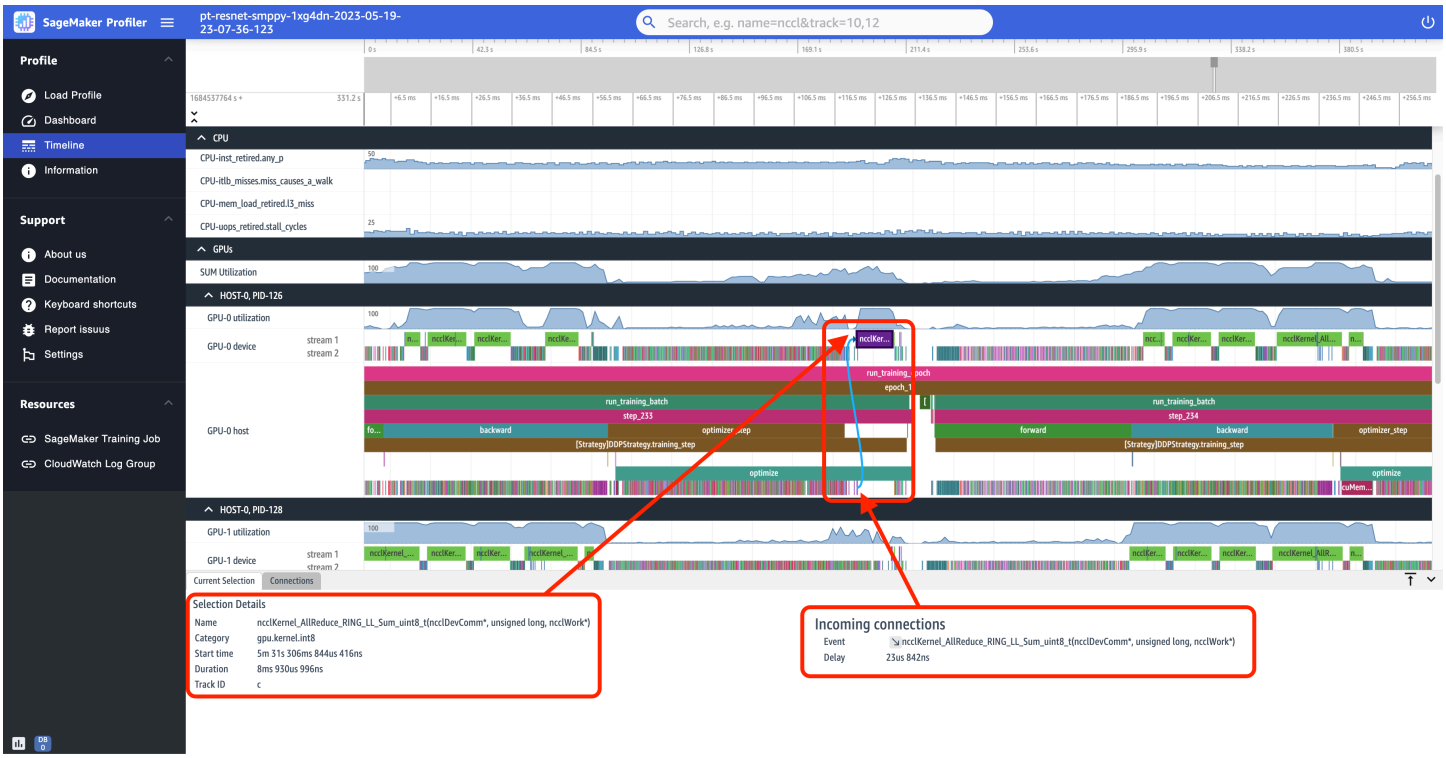
Untuk setiap CPU, Anda dapat melacak pemanfaatan CPU dan penghitung kinerja, seperti `"clk_unhalted_ref.tsc"` dan`"itlb_misses.miss_causes_a_walk"`, yang merupakan indikasi instruksi yang dijalankan pada CPU.

Untuk setiap GPU, Anda dapat melihat timeline host dan timeline perangkat. Peluncuran kernel ada di timeline host dan kernel berjalan ada di timeline perangkat. Anda juga dapat melihat anotasi (seperti maju, mundur, dan mengoptimalkan) jika Anda telah menambahkan skrip pelatihan di timeline host GPU.

Dalam tampilan timeline, Anda juga dapat melacak launch-and-run pasangan kernel. Ini membantu Anda memahami bagaimana peluncuran kernel yang dijadwalkan pada host (CPU) dijalankan pada perangkat GPU yang sesuai.

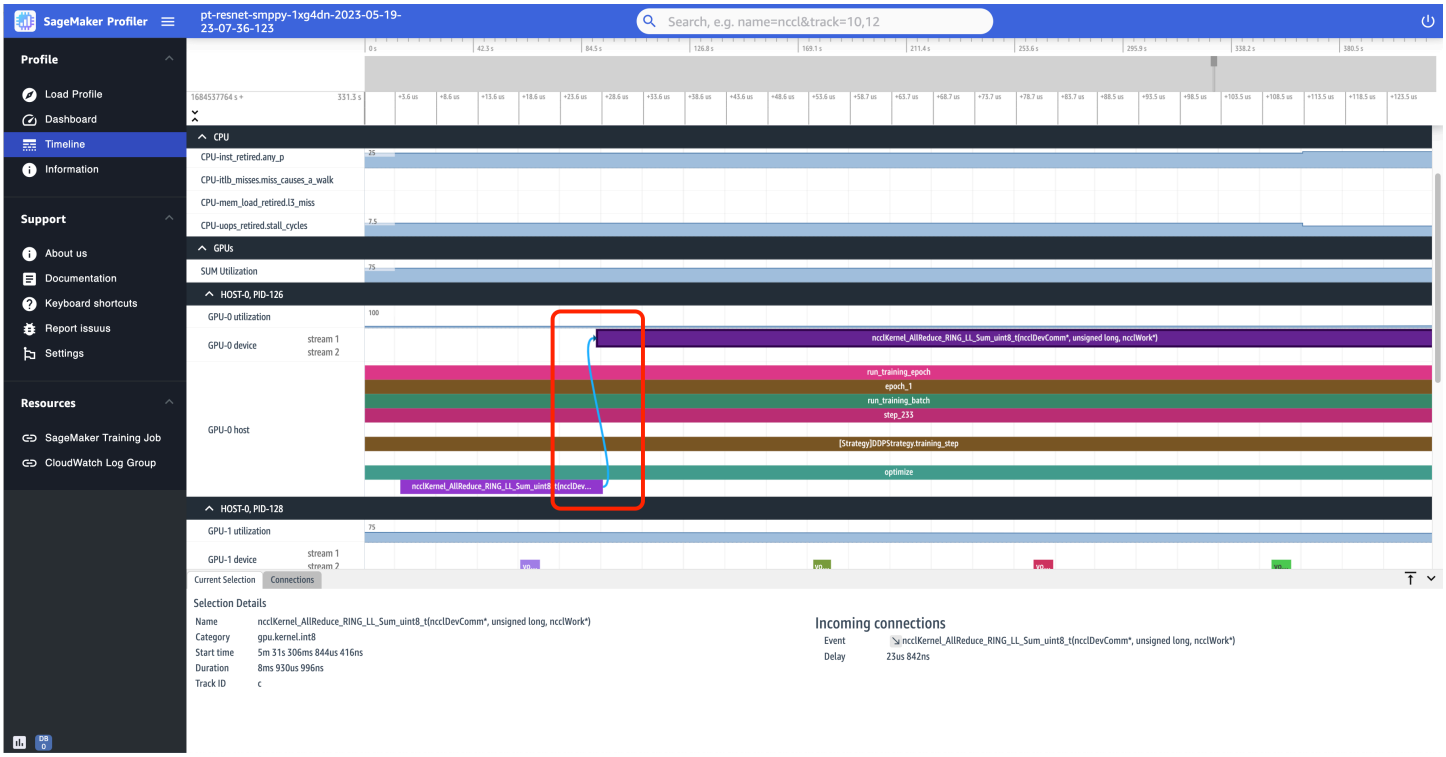
> **Tip**
>
> Tekan `f` tombol untuk memperbesar kernel yang dipilih.

Tangkapan layar berikut adalah tampilan yang diperbesar ke dalam `step_233` dan `step_234` dari tangkapan layar sebelumnya. Interval timeline yang dipilih dalam tangkapan layar berikut adalah `AllReduce` operasi, langkah komunikasi dan sinkronisasi penting dalam pelatihan terdistribusi, dijalankan pada perangkat GPU-0. Dalam tangkapan layar, perhatikan bahwa peluncuran kernel di host GPU-0 terhubung ke kernel yang dijalankan di aliran perangkat GPU-0 1, ditunjukkan dengan panah dalam warna cyan.



Juga dua tab informasi muncul di panel bawah UI saat Anda memilih interval garis waktu, seperti yang ditunjukkan pada tangkapan layar sebelumnya. Tab Seleksi Saat Ini menunjukkan rincian kernel yang dipilih dan peluncuran kernel yang terhubung dari host. Arah koneksi selalu dari host (CPU) ke perangkat (GPU) karena setiap kernel GPU selalu dipanggil dari CPU. Tab Connections menunjukkan pasangan peluncuran dan jalankan kernel yang dipilih. Anda dapat memilih salah satu dari mereka untuk memindahkannya ke tengah tampilan Timeline.

Tangkapan layar berikut memperbesar lebih jauh ke peluncuran `AllReduce` operasi dan menjalankan pasangan.



Informasi

Di Informasi, Anda dapat mengakses informasi tentang pekerjaan pelatihan yang dimuat, seperti jenis instans, Nama Sumber Daya Amazon (ARN) sumber daya komputasi yang disediakan untuk pekerjaan, nama node, dan hyperparameter.

Pengaturan

Instance aplikasi SageMaker Profiler UI dikonfigurasi untuk dimatikan setelah 2 jam waktu idle secara default. Di Pengaturan, gunakan pengaturan berikut untuk menyesuaikan timer shutdown otomatis.

- Aktifkan shutdown otomatis aplikasi - Pilih dan atur ke Diaktifkan agar aplikasi mati secara otomatis setelah jumlah jam waktu idle yang ditentukan. Untuk mematikan fungsi auto-shutdown, pilih Dinonaktifkan.
- Ambang batas shutdown otomatis dalam hitungan jam - Jika Anda memilih Diaktifkan untuk Aktifkan shutdown otomatis aplikasi, Anda dapat mengatur batas waktu dalam beberapa jam agar aplikasi dimatikan secara otomatis. Ini diatur ke 2 secara default.

Pertanyaan yang sering diajukan tentang penggunaan SageMaker Profiler

Gunakan pertanyaan umum berikut untuk menemukan jawaban tentang penggunaan SageMaker Profiler.

Q. Saya mendapatkan pesan kesalahan, **`ModuleNotFoundError: No module named 'smppy'`**

Sejak Desember 2023, nama paket SageMaker Profiler Python telah berubah dari `smppy` menjadi `smprof` menyelesaikan masalah nama paket duplikat; sudah `smppy` digunakan oleh paket open source.

Oleh karena itu, jika Anda telah menggunakan `smppy` sejak sebelum Desember 2023 dan mengalami `ModuleNotFoundError` masalah ini, mungkin karena nama paket yang sudah ketinggalan zaman dalam skrip pelatihan Anda saat `smprof` paket yang terakhir diinstal atau menggunakan salah satu yang terbaru. the section called “SageMaker gambar kerangka pra-instal dengan Profiler SageMaker ” Dalam hal ini, pastikan Anda mengganti semua sebutan `smppy` dengan `smprof` seluruh skrip pelatihan Anda.

Saat memperbarui nama paket SageMaker Profiler Python dalam skrip pelatihan Anda, untuk menghindari kebingungan seputar versi nama paket mana yang harus Anda gunakan, pertimbangkan untuk menggunakan pernyataan impor bersyarat seperti yang ditunjukkan dalam cuplikan kode berikut.

```
try:
    import smprof
except ImportError:
    # backward-compatability for TF 2.11 and PT 1.13.1 images
    import smppy as smprof
```

Perhatikan juga bahwa jika Anda telah menggunakan `smppy` saat meningkatkan ke versi terbaru PyTorch atau TensorFlow versi, pastikan Anda menginstal `smprof` paket terbaru dengan mengikuti petunjuk dithe section called “(Opsional) Instal paket SageMaker Profiler Python”.

Q. Saya mendapatkan pesan kesalahan, **`ModuleNotFoundError: No module named 'smprof'`**

Pertama, pastikan Anda menggunakan salah satu SageMaker Framework Container yang didukung secara resmi. Jika Anda tidak menggunakan salah satunya, Anda dapat menginstal `smprof` paket dengan mengikuti instruksi dithe section called “(Opsional) Instal paket SageMaker Profiler Python”.

Q. Saya tidak dapat mengimpor **`ProfilerConfig`**

Jika Anda tidak dapat mengimpor `ProfilerConfig` skrip peluncur pekerjaan menggunakan SageMaker Python SDK, lingkungan lokal Anda atau kernel Jupyter mungkin memiliki versi Python SDK yang sudah ketinggalan zaman secara signifikan. SageMaker Pastikan Anda memutakhirkan SDK ke versi terbaru.

```
$ pip install --upgrade sagemaker
```

Q. Saya mendapatkan pesan kesalahan, **`aborted: core dumped when importing smprof into my training script`**

Dalam versi sebelumnya`smprof`, masalah ini terjadi dengan PyTorch 2.0+ dan PyTorch Lightning. Untuk mengatasi masalah ini, instal juga `smprof` paket terbaru dengan mengikuti petunjuk dithe section called “(Opsional) Instal paket SageMaker Profiler Python”.

T. Saya tidak dapat menemukan UI SageMaker Profiler dari SageMaker Studio. Bagaimana saya bisa menemukannya?

Jika Anda memiliki akses ke SageMaker konsol, pilih salah satu opsi berikut.

- the section called “Opsi 1: Luncurkan UI SageMaker Profiler dari halaman detail Domain”
- the section called “Opsi 2: Luncurkan aplikasi UI SageMaker Profiler dari halaman arahan SageMaker Profiler di konsol SageMaker”

Jika Anda adalah pengguna Domain dan tidak memiliki akses ke SageMaker konsol, Anda dapat mengakses aplikasi melalui SageMaker Studio Classic. Jika ini kasus Anda, pilih opsi berikut.

- the section called “Opsi 3: Gunakan fungsi peluncur aplikasi di SageMaker Python SDK”

Pertimbangan

Pertimbangkan hal berikut saat menggunakan SageMaker Profiler.

- SageMaker Profiler tidak kompatibel dengan kolam hangat yang SageMaker dikelola.

Pantau pemanfaatan sumber daya AWS komputasi di Amazon Studio Classic SageMaker

Untuk melacak pemanfaatan sumber daya komputasi dari pekerjaan pelatihan Anda, gunakan alat pemantauan yang ditawarkan oleh Amazon SageMaker Debugger.

Untuk pekerjaan pelatihan apa pun yang Anda jalankan dalam SageMaker menggunakan SageMaker Python SDK, Debugger mengumpulkan metrik pemanfaatan sumber daya dasar, seperti pemanfaatan CPU, pemanfaatan GPU, pemanfaatan memori GPU, jaringan, dan waktu tunggu I/O setiap 500 milidetik. Untuk melihat dasbor metrik pemanfaatan sumber daya dari pekerjaan pelatihan Anda, cukup gunakan UI SageMaker Debugger di Studio Experiments. SageMaker

Operasi dan langkah-langkah pembelajaran mendalam dapat beroperasi dalam interval milidetik. Dibandingkan dengan CloudWatch metrik Amazon, yang mengumpulkan metrik pada interval 1 detik, Debugger memberikan perincian yang lebih halus ke dalam metrik pemanfaatan sumber daya hingga interval 100 milidetik (0,1 detik) sehingga Anda dapat menyelam jauh ke dalam metrik pada tingkat operasi atau langkah.

Jika Anda ingin mengubah interval waktu pengumpulan metrik, Anda dapat menambahkan parameter untuk konfigurasi profil ke peluncur pekerjaan pelatihan Anda. Misalnya, jika Anda menggunakan SageMaker Python SDK, Anda harus meneruskan `profiler_config` parameter saat membuat objek estimator. Untuk mempelajari cara menyesuaikan interval pengumpulan metrik pemanfaatan sumber daya, lihat the section called “Template kode untuk mengkonfigurasi SageMaker objek estimator dengan SageMaker Modul Python debugger di SageMakerSDK Python” dan kemudianthe section called “Konfigurasikan pengaturan untuk pembuatan profil dasar pemanfaatan sumber daya sistem”.

Selain itu, Anda dapat menambahkan alat pendeteksi masalah yang disebut aturan profil bawaan yang disediakan oleh SageMaker Debugger. Aturan pembuatan profil bawaan menjalankan analisis terhadap metrik pemanfaatan sumber daya dan mendeteksi masalah kinerja komputasi. Untuk informasi selengkapnya, lihat the section called “Konfigurasikan aturan profiler bawaan”. Anda dapat menerima hasil analisis aturan melalui UI SageMaker Debugger di SageMaker Studio Experiments atau SageMaker Debugger Profiling Report. Anda juga dapat membuat aturan pembuatan profil khusus menggunakan SageMaker Python SDK.

Untuk mempelajari selengkapnya tentang fungsi pemantauan yang disediakan oleh SageMaker Debugger, lihat topik berikut.

Topik

- Konfigurasikan estimator dengan parameter untuk pembuatan profil dasar menggunakan Amazon SageMaker Modul Python Debugger
- Konfigurasikan aturan profiler bawaan yang dikelola oleh Amazon SageMaker Debugger
- Daftar aturan profiler bawaan Debugger
- Amazon SageMaker Debugger UI di Eksperimen Klasik Amazon SageMaker Studio
- SageMaker Laporan interaktif debugger
- Menganalisis data menggunakan pustaka klien Python Debugger

Konfigurasikan estimator dengan parameter untuk pembuatan profil dasar menggunakan Amazon SageMaker Modul Python Debugger

Secara default, SageMaker Profil dasar debugger aktif secara default dan memantau metrik pemanfaatan sumber daya, seperti pemanfaatan CPU, pemanfaatan GPU, pemanfaatan memori GPU, Jaringan, dan waktu tunggu I/O, semuanya SageMaker pekerjaan pelatihan yang diajukan menggunakanAmazon SageMaker SDK Python. SageMaker Debugger mengumpulkan metrik pemanfaatan sumber daya ini setiap 500 milidetik. Anda tidak perlu membuat perubahan tambahan dalam kode, skrip pelatihan, atau peluncur pekerjaan untuk melacak pemanfaatan sumber daya dasar. Jika Anda ingin mengakses dasbor metrik pemanfaatan sumber daya dari pekerjaan pelatihan Anda di SageMaker Studio, Anda dapat melompat keAmazon SageMaker Debugger UI di Eksperimen Klasik Amazon SageMaker Studio.

Jika Anda ingin mengubah interval pengumpulan metrik untuk pembuatan profil dasar, Anda dapat menentukan parameter khusus Debugger saat membuat SageMaker pelatihan peluncur pekerjaan menggunakan SageMaker SDK Python,AWS SDK for Python (Boto3), atauAWS Command Line Interface(CLI). Dalam panduan ini, kami fokus pada cara mengubah opsi profil menggunakanAmazon SageMaker SDK Python.

Jika Anda ingin mengaktifkan masalah pemanfaatan sumber daya sistem secara otomatis, Anda dapat menambahkan`rules`parameter dalam objek estimator untuk mengaktifkan aturan.

> **Important**
>
> Untuk menggunakan yang terbaru SageMaker Fitur debugger, Anda perlu memutakhirkan SageMaker Python SDK dan`SMDebug`perpustakaan klien. Di kernel IPython Anda, Jupyter Notebook, atau JupyterLab lingkungan, jalankan kode berikut untuk menginstal versi terbaru dari perpustakaan dan restart kernel.

```
import sys
import IPython
!{sys.executable} -m pip install -U sagemaker smdebug
IPython.Application.instance().kernel.do_shutdown(True)
```

Template kode untuk mengkonfigurasi SageMaker objek estimator dengan SageMaker Modul Python debugger di SageMakerSDK Python

Untuk menyesuaikan konfigurasi profil dasar (`profiler_config`) atau tambahkan aturan profiler (`rules`), pilih salah satu tab untuk menyiapkan templat SageMaker penaksir. Di halaman berikutnya, Anda dapat menemukan informasi lebih lanjut tentang cara mengkonfigurasi dua parameter.

> **Note**
>
> Contoh kode berikut tidak dapat dieksekusi secara langsung. Lanjutkan ke bagian berikutnya untuk mempelajari cara mengkonfigurasi setiap parameter.

PyTorch

```
# An example of constructing a SageMaker PyTorch estimator
import boto3
import sagemaker
from sagemaker.pytorch import PyTorch
from sagemaker.debugger import ProfilerConfig, ProfilerRule, rule_configs

session=boto3.session.Session()
region=session.region_name

profiler_config=ProfilerConfig(...)
rules=[
    ProfilerRule.sagemaker(rule_configs.BuiltInRule())
]

estimator=PyTorch(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-profiling-demo",
    instance_count=1,
```

```
    instance_type="ml.p3.2xlarge",
    framework_version="1.12.0",
    py_version="py37",

    # SageMaker Debugger parameters
    profiler_config=profiler_config,
    rules=rules
)

estimator.fit(wait=False)
```

TensorFlow

```
# An example of constructing a SageMaker TensorFlow estimator
import boto3
import sagemaker
from sagemaker.tensorflow import TensorFlow
from sagemaker.debugger import ProfilerConfig, ProfilerRule, rule_configs

session=boto3.session.Session()
region=session.region_name

profiler_config=ProfilerConfig(...)
rules=[
    ProfilerRule.sagemaker(rule_configs.BuiltInRule())
]

estimator=TensorFlow(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-profiling-demo",
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="2.8.0",
    py_version="py37",

    # SageMaker Debugger parameters
    profiler_config=profiler_config,
    rules=rules
)

estimator.fit(wait=False)
```

MXNet

```
# An example of constructing a SageMaker MXNet estimator
import sagemaker
from sagemaker.mxnet import MXNet
from sagemaker.debugger import ProfilerConfig, ProfilerRule, rule_configs

profiler_config=ProfilerConfig(...)
rules=[
    ProfilerRule.sagemaker(rule_configs.BuiltInRule())
]

estimator=MXNet(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-profiling-demo",
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="1.7.0",
    py_version="py37",

    # SageMaker Debugger parameters
    profiler_config=profiler_config,
    rules=rules
)

estimator.fit(wait=False)
```

> **Note**
>
> Untuk MXNet, saat mengonfigurasi`profiler_config`parameter, Anda hanya dapat mengkonfigurasi untuk pemantauan sistem. Metrik kerangka kerja profil tidak didukung untuk MXNet.

XGBoost

```
# An example of constructing a SageMaker XGBoost estimator
import sagemaker
from sagemaker.xgboost.estimator import XGBoost
from sagemaker.debugger import ProfilerConfig, ProfilerRule, rule_configs
```

```
profiler_config=ProfilerConfig(...)
rules=[
    ProfilerRule.sagemaker(rule_configs.BuiltInRule())
]

estimator=XGBoost(
    entry_point="directory/to/your_training_script.py",
    role=sagemaker.get_execution_role(),
    base_job_name="debugger-profiling-demo",
    instance_count=1,
    instance_type="ml.p3.2xlarge",
    framework_version="1.5-1",

    # Debugger-specific parameters
    profiler_config=profiler_config,
    rules=rules
)

estimator.fit(wait=False)
```

> **Note**
>
> Untuk XGBoost, saat mengonfigurasi`profiler_config`parameter, Anda hanya dapat mengkonfigurasi untuk pemantauan sistem. Metrik kerangka kerja profil tidak didukung untuk XGBoost.

Generic estimator

```
# An example of constructing a SageMaker generic estimator using the XGBoost
 algorithm base image
import boto3
import sagemaker
from sagemaker.estimator import Estimator
from sagemaker import image_uris
from sagemaker.debugger import ProfilerConfig, DebuggerHookConfig, Rule,
 ProfilerRule, rule_configs

profiler_config=ProfilerConfig(...)
rules=[
    ProfilerRule.sagemaker(rule_configs.BuiltInRule())
]
```

```
region=boto3.Session().region_name
xgboost_container=sagemaker.image_uris.retrieve("xgboost", region, "1.5-1")

estimator=Estimator(
    role=sagemaker.get_execution_role()
    image_uri=xgboost_container,
    base_job_name="debugger-demo",
    instance_count=1,
    instance_type="ml.m5.2xlarge",

    # Debugger-specific parameters
    profiler_config=profiler_config,
    rules=rules
)

estimator.fit(wait=False)
```

Berikut ini memberikan deskripsi singkat tentang parameter.

- `profiler_config`— Konfigurasikan Debugger untuk mengumpulkan metrik sistem dan metrik kerangka kerja dari pekerjaan pelatihan Anda dan simpan ke URI bucket S3 atau mesin lokal Anda yang aman. Anda dapat mengatur seberapa sering atau longgar mengumpulkan metrik sistem. Untuk mempelajari cara mengkonfigurasi`profiler_config`parameter, lihatKonfigurasikan pengaturan untuk pembuatan profil dasar pemanfaatan sumber daya sistemdanKonfigurasikan untuk pembuatan profil kerangka kerja.
- `rules`— Konfigurasikan parameter ini untuk mengaktifkan SageMaker Aturan bawaan debugger yang ingin Anda jalankan secara paralel. Pastikan bahwa pekerjaan pelatihan Anda memiliki akses ke bucket S3 ini. Aturan berjalan pada wadah pemrosesan dan secara otomatis menganalisis pekerjaan pelatihan Anda untuk menemukan masalah kinerja komputasi dan operasional. TheProfilerReportrule adalah aturan paling terintegrasi yang menjalankan semua aturan pembuatan profil bawaan dan menyimpan hasil pembuatan profil sebagai laporan ke dalam bucket S3 aman Anda. Untuk mempelajari cara mengkonfigurasi`rules`parameter, lihatKonfigurasikan aturan profiler bawaan yang dikelola oleh Amazon SageMaker Debugger.

> **Note**
>
> Debugger menyimpan data keluaran dengan aman di subfolder bucket S3 default Anda. Misalnya, format URI bucket S3 default adalah`s3://sagemaker-<region>-<12digit_account_id>/<base-job-name>/<debugger-subfolders>/`. Ada tiga subfolder yang dibuat oleh Debugger:`debug-output`,`profiler-output`, dan`rule-output`. Anda juga dapat mengambil URI bucket S3 default menggunakanSageMaker metode kelas estimator.

Lihat topik berikut untuk mengetahui cara mengonfigurasi parameter khusus Debugger secara detail.

Topik

- Konfigurasikan pengaturan untuk pembuatan profil dasar pemanfaatan sumber daya sistem
- Konfigurasikan untuk pembuatan profil kerangka kerja
- Memperbarui pemantauan sistem Debugger dan konfigurasi profil kerangka kerja saat pekerjaan pelatihan sedang berjalan
- Matikan Debugger

Konfigurasikan pengaturan untuk pembuatan profil dasar pemanfaatan sumber daya sistem

Untuk menyesuaikan interval waktu untuk mengumpulkan metrik pemanfaatan, gunakan`ProfilerConfig`Operasi API untuk membuat objek parameter saat membuat SageMaker kerangka kerja atau estimator generik tergantung pada preferensi Anda.

> **Note**
>
> Secara default, untuk semua SageMaker pekerjaan pelatihan, Debugger mengumpulkan metrik pemanfaatan sumber daya dari instans Amazon EC2 setiap 500 milidetik untuk pemantauan sistem, tanpa parameter khusus Debugger yang ditentukan dalam SageMaker penaksir.
>
> Debugger menyimpan metrik sistem di bucket S3 default. Format URI bucket S3 default adalah`s3://sagemaker-<region>-<12digit_account_id>/<training-job-name>/profiler-output/`.

Contoh kode berikut menunjukkan cara mengaturprofiler_configparameter dengan interval waktu pemantauan sistem 1000 milidetik.

```
from sagemaker.debugger import ProfilerConfig

profiler_config=ProfilerConfig(
    system_monitor_interval_millis=1000
)
```

- system_monitor_interval_millis(int) — Tentukan interval pemantauan dalam milidetik untuk merekam metrik sistem. Nilai yang tersedia adalah 100, 200, 500, 1000 (1 detik), 5000 (5 detik), dan 60000 (1 menit) milidetik. Nilai default adalah 500 milidetik.

Untuk melihat kemajuan pemantauan sistem, lihatBuka dasbor Amazon SageMaker Debugger Insights.

Konfigurasikan untuk pembuatan profil kerangka kerja

> **Warning**
>
> MendukungAmazon SageMaker Profiler, SageMakerDebugger menghentikan fitur pembuatan profil kerangka kerja mulai dari TensorFlow 2.11 dan PyTorch 2.0. Anda masih dapat menggunakan fitur ini di versi kerangka kerja dan SDK sebelumnya sebagai berikut.
>
> - SageMaker Python SDK <= v2.130.0
> - PyTorch >= v1.6.0, < v2.0
> - TensorFlow >= v2.3.1, < v2.11
>
> Lihat juga 16 Maret 2023.

Untuk mengaktifkan pembuatan profil kerangka kerja Debugger, konfigurasikanframework_profile_paramsparameter ketika Anda membangun estimator. Pemprofilan kerangka kerja debugger mengumpulkan metrik kerangka kerja, seperti data dari tahap inisialisasi, proses pemuat data, operator Python dari kerangka kerja pembelajaran mendalam dan skrip pelatihan, pembuatan profil terperinci di dalam dan di antara langkah-langkah, dengan opsi CProfile atau Pyinstrument. MenggunakanFrameworkProfilekelas, Anda dapat mengonfigurasi opsi pembuatan profil kerangka kustom.

> **Note**
>
> Sebelum memulai dengan pembuatan profil kerangka kerja Debugger, verifikasi bahwa kerangka kerja yang digunakan untuk membangun model Anda didukung oleh Debugger untuk pembuatan profil kerangka kerja. Untuk informasi selengkapnya, lihat Kerangka Kerja dan Algoritma yang Didukung.
>
> Debugger menyimpan metrik kerangka kerja dalam bucket S3 default. Format URI bucket S3 default adalah`s3://sagemaker-<region>-<12digit_account_id>/<training-job-name>/profiler-output/`.

Mulai pekerjaan pelatihan dengan pembuatan profil kerangka kerja default

Contoh kode berikut adalah yang paling sederhana`profiler_config`pengaturan parameter untuk memulai pemantauan sistem default dan profil kerangka kerja default. The`FrameworkProfile`kelas dalam kode contoh berikut memulai profil kerangka kerja default ketika pekerjaan pelatihan dimulai. Pemprofilan kerangka kerja debugger mencakup opsi berikut: pembuatan profil terperinci, profil pemuat data, dan profil Python.

```
from sagemaker.debugger import ProfilerConfig, FrameworkProfile

profiler_config=ProfilerConfig(
    framework_profile_params=FrameworkProfile()
)
```

Dengan ini`profiler_config`konfigurasi parameter, Debugger memanggil pengaturan default pemantauan dan pembuatan profil. Debugger memantau metrik sistem setiap 500 milidetik; profil langkah kelima dengan opsi profil terperinci; langkah ketujuh dengan opsi profil pemuat data; dan langkah kesembilan, kesepuluh, dan kesebelas dengan opsi profil Python.

Untuk menemukan opsi konfigurasi profil yang tersedia, pengaturan parameter default, dan contoh cara mengonfigurasinya, lihatMulai pekerjaan pelatihan dengan pemantauan sistem default dan pembuatan profil kerangka kerja yang disesuaikan dengan opsi pembuatan profil yang berbedadanSageMaker API Debugger - FrameworkProfilediAmazon SageMaker SDK Python.

Jika Anda ingin mengubah interval pemantauan sistem dan mengaktifkan profil kerangka kerja default, Anda dapat menentukan`system_monitor_interval_millis`parameter secara eksplisit dengan`framework_profile_params`parameter. Misalnya, untuk memantau setiap 1000 milidetik dan mengaktifkan profil kerangka kerja default, gunakan kode contoh berikut.

```
from sagemaker.debugger import ProfilerConfig, FrameworkProfile

profiler_config=ProfilerConfig(
    system_monitor_interval_millis=1000,
    framework_profile_params=FrameworkProfile()
)
```

Untuk informasi lebih lanjut tentang`FrameworkProfile`kelas, lihat[SageMaker API Debugger - FrameworkProfile](#)di[Amazon SageMaker SDK Python](#).

Mulai pekerjaan pelatihan dengan pemantauan sistem default dan pembuatan profil kerangka kerja yang disesuaikan untuk langkah target atau rentang waktu target

Jika Anda ingin menentukan langkah target atau interval waktu target untuk membuat profil pekerjaan pelatihan Anda, Anda perlu menentukan parameter untuk`FrameworkProfile`kelas. Contoh kode berikut menunjukkan cara menentukan rentang target untuk pembuatan profil bersama dengan pemantauan sistem.

- Untuk rentang langkah target

  Dengan konfigurasi contoh berikut, Debugger memantau seluruh pekerjaan pelatihan setiap 500 milidetik (pemantauan default) dan profil rentang langkah target dari langkah 5 hingga langkah 15 (untuk 10 langkah).

  ```
  from sagemaker.debugger import ProfilerConfig, FrameworkProfile

  profiler_config=ProfilerConfig(
      framework_profile_params=FrameworkProfile(start_step=5, num_steps=10)
  )
  ```

  Dengan konfigurasi contoh berikut, Debugger memantau seluruh pekerjaan pelatihan setiap 1000 milidetik dan profil rentang langkah target dari langkah 5 hingga langkah 15 (untuk 10 langkah).

  ```
  from sagemaker.debugger import ProfilerConfig, FrameworkProfile

  profiler_config=ProfilerConfig(
      system_monitor_interval_millis=1000,
      framework_profile_params=FrameworkProfile(start_step=5, num_steps=10)
  )
  ```

- Untuk rentang waktu target

Dengan konfigurasi contoh berikut, Debugger memantau seluruh pekerjaan pelatihan setiap 500 milidetik (pemantauan default) dan profil rentang waktu target dari waktu Unix saat ini selama 600 detik.

```
import time
from sagemaker.debugger import ProfilerConfig, FrameworkProfile

profiler_config=ProfilerConfig(
    framework_profile_params=FrameworkProfile(start_unix_time=int(time.time()),
 duration=600)
)
```

Dengan konfigurasi contoh berikut, Debugger memantau seluruh pekerjaan pelatihan setiap 1000 milidetik dan profil rentang waktu target dari waktu Unix saat ini selama 600 detik.

```
import time
from sagemaker.debugger import ProfilerConfig, FrameworkProfile

profiler_config=ProfilerConfig(
    system_monitor_interval_millis=1000,
    framework_profile_params=FrameworkProfile(start_unix_time=int(time.time()),
 duration=600)
)
```

Pembuatan profil kerangka kerja dilakukan untuk semua opsi pembuatan profil pada langkah target atau rentang waktu.

Untuk menemukan informasi selengkapnya tentang opsi pembuatan profil yang tersedia, lihatSageMaker API Debugger - FrameworkProfilediAmazon SageMaker SDK Python.

Bagian selanjutnya menunjukkan Anda cara membuat skrip opsi profil yang tersedia.

Mulai pekerjaan pelatihan dengan pemantauan sistem default dan pembuatan profil kerangka kerja yang disesuaikan dengan opsi pembuatan profil yang berbeda

Anda dapat menggunakan kelas konfigurasi profiling berikut untuk mengelola opsi pembuatan profil kerangka kerja:

- DetailedProfilingConfig— Tentukan langkah target atau rentang waktu untuk operasi kerangka profil menggunakan profiler kerangka kerja asli (TensorFlow profiler dan PyTorch profiler). Misalnya, jika menggunakan TensorFlow, kait Debugger mengaktifkan TensorFlow profiler untuk mengumpulkan TensorFlow-metrik kerangka kerja khusus. Pembuatan profil terperinci memungkinkan Anda untuk membuat profil semua operator kerangka kerja pada tahap awal (sebelum langkah pertama), dalam beberapa langkah, dan di antara langkah-langkah pekerjaan pelatihan.

  > **Note**
  >
  > Profil terperinci dapat secara signifikan meningkatkan konsumsi memori GPU. Kami tidak menyarankan untuk mengaktifkan profil terperinci selama lebih dari beberapa langkah.

- DataloaderProfilingConfig— Tentukan langkah target atau rentang waktu untuk membuat profil proses pemuat data kerangka pembelajaran mendalam. Debugger mengumpulkan setiap peristiwa pemuat data dari kerangka kerja.

  > **Note**
  >
  > Pembuatan profil pemuat data dapat menurunkan kinerja pelatihan sambil mengumpulkan informasi dari pemuat data. Kami tidak menyarankan untuk mengaktifkan profil pemuat data selama lebih dari beberapa langkah.
  > Debugger telah dikonfigurasi sebelumnya untuk membubuhi keterangan proses pemuat data hanya untukAWSwadah pembelajaran mendalam. Debugger tidak dapat memprofilkan proses pemuat data dari wadah pelatihan khusus atau eksternal lainnya.

- PythonProfilingConfig— Tentukan langkah target atau rentang waktu untuk memprofilkan fungsi Python. Anda juga dapat memilih antara dua profiler Python: CProfile dan Pyinstrument.
  - CProfil— Profiler Python standar. cProfile mengumpulkan informasi untuk setiap operator Python yang dipanggil selama pelatihan. Dengan CProfile, Debugger menghemat waktu kumulatif dan anotasi untuk setiap panggilan fungsi, memberikan detail lengkap tentang fungsi Python. Dalam pembelajaran mendalam, misalnya, fungsi yang paling sering disebut mungkin adalah filter convolutional dan operator backward pass, dan profil CProfile masing-masing. Untuk opsi CProfile, Anda dapat memilih opsi timer lebih lanjut: total waktu, waktu CPU, dan waktu off-CPU. Meskipun Anda dapat membuat profil setiap panggilan fungsi yang dijalankan pada prosesor (baik CPU dan GPU) dalam waktu CPU, Anda juga dapat mengidentifikasi I/O atau kemacetan jaringan dengan opsi waktu off-CPU. Defaultnya adalah total waktu, dan Debugger

memprofilkan waktu CPU dan off-CPU. Dengan CProfile, Anda dapat menelusuri setiap fungsi saat menganalisis data profil.

- Pyinstrument— Pyinstrument adalah profiler Python overhead rendah yang bekerja berdasarkan pengambilan sampel. Dengan opsi Pyinstrument, Debugger mengambil sampel pembuatan profil peristiwa setiap milidetik. Karena Pyinstrument mengukur waktu jam dinding yang telah berlalu alih-alih waktu CPU, opsi Pyinstrument dapat menjadi pilihan yang lebih baik daripada opsi CProfile untuk mengurangi kebisingan profil (menyaring panggilan fungsi yang tidak relevan yang secara kumulatif cepat) dan menangkap operator yang sebenarnya komputasi intensif (secara kumulatif lambat) untuk melatih model Anda. Dengan Pyinstrument, Anda dapat melihat pohon panggilan fungsi dan lebih memahami struktur dan akar penyebab kelambatan.

> **Note**
>
> Mengaktifkan pembuatan profil Python dapat memperlambat waktu pelatihan secara keseluruhan. cProfile memprofilkan operator Python yang paling sering disebut di setiap panggilan, sehingga waktu pemrosesan pada pembuatan profil meningkat sehubungan dengan jumlah panggilan. Untuk Pyinstrument, waktu pembuatan profil kumulatif meningkat sehubungan dengan waktu karena mekanisme pengambilan sampelnya.

Contoh konfigurasi berikut menunjukkan struktur lengkap saat Anda menggunakan opsi profil yang berbeda dengan nilai yang ditentukan.

```
import time
from sagemaker.debugger import (ProfilerConfig,
                                FrameworkProfile,
                                DetailedProfilingConfig,
                                DataloaderProfilingConfig,
                                PythonProfilingConfig,
                                PythonProfiler, cProfileTimer)

profiler_config=ProfilerConfig(
    system_monitor_interval_millis=500,
    framework_profile_params=FrameworkProfile(
        detailed_profiling_config=DetailedProfilingConfig(
            start_step=5,
            num_steps=1
        ),
        dataloader_profiling_config=DataloaderProfilingConfig(
            start_step=7,
```

```
            num_steps=1
        ),
        python_profiling_config=PythonProfilingConfig(
            start_step=9,
            num_steps=1,
            python_profiler=PythonProfiler.CPROFILE,
            cprofile_timer=cProfileTimer.TOTAL_TIME
        )
    )
)
```

Untuk informasi selengkapnya tentang opsi pembuatan profil yang tersedia, lihatDetailedProfilingConfig,DataloaderProfilingConfig, danPythonProfilingConfigdiAmazon SageMaker SDK Python.

Memperbarui pemantauan sistem Debugger dan konfigurasi profil kerangka kerja saat pekerjaan pelatihan sedang berjalan

Jika Anda ingin mengaktifkan atau memperbarui konfigurasi pemantauan Debugger untuk pekerjaan pelatihan yang sedang berjalan, gunakan yang berikut SageMaker metode ekstensi estimator:

- Untuk mengaktifkan pemantauan sistem Debugger untuk pekerjaan pelatihan yang sedang berjalan dan menerima laporan pembuatan profil Debugger, gunakan yang berikut ini:

  ```
  estimator.enable_default_profiling()
  ```

  Saat Anda menggunakan`enable_default_profiling`metode, Debugger memulai pemantauan sistem default dan`ProfileReport`aturan bawaan, yang menghasilkan laporan profil komprehensif di akhir pekerjaan pelatihan. Metode ini hanya dapat dipanggil jika pekerjaan pelatihan saat ini berjalan tanpa pemantauan dan pembuatan profil Debugger.

  Untuk informasi selengkapnya, lihatestimator.enable_default_profilingdiAmazon SageMaker SDK Python.
- Untuk memperbarui konfigurasi pemantauan sistem, gunakan yang berikut ini:

  ```
  estimator.update_profiler(
      system_monitor_interval_millis=500
  )
  ```

  Untuk informasi selengkapnya, lihatestimator.update_profilerdiAmazon SageMaker SDK Python.

Matikan Debugger

Jika Anda ingin mematikan Debugger, lakukan salah satu dari yang berikut:

- Sebelum memulai pekerjaan, lakukan hal berikut:

  Untuk mematikan profiling, sertakan`disable_profiler`parameter ke estimator Anda dan atur ke`True`.

  > **Warning**
  >
  > Jika menonaktifkannya, Anda tidak akan dapat melihat dasbor wawasan Studio Debugger yang komprehensif dan laporan pembuatan profil yang dibuat secara otomatis.

  Untuk mematikan debugging, atur`debugger_hook_config`parameter untuk`False`.

  > **Warning**
  >
  > Jika Anda menonaktifkannya, Anda tidak akan dapat mengumpulkan tensor dan tidak dapat men-debug parameter model Anda.

  ```
  estimator=Estimator(
      ...
      disable_profiler=True
      debugger_hook_config=False
  )
  ```

  Untuk informasi selengkapnya tentang parameter khusus Debugger, lihatSageMaker Estimatordi[Amazon SageMaker SDK Python](https://sagemaker.readthedocs.io/en/stable/).

- Saat pekerjaan pelatihan sedang berjalan, lakukan hal berikut:

  Untuk menonaktifkan pemantauan dan pembuatan profil saat tugas pelatihan Anda berjalan, gunakan metode kelas estimator berikut:

  ```
  estimator.disable_profiling()
  ```

Untuk menonaktifkan profil kerangka kerja saja dan menjaga pemantauan sistem, gunakan`update_profiler`Metode:

```
estimator.update_profiler(disable_framework_metrics=true)
```

Untuk informasi selengkapnya tentang metode ekstensi, lihatestimator.disable_profilingdanestimator.update_profilerclassmethods diAmazon SageMaker SDK Pythondokumentasi.

Konfigurasikan aturan profiler bawaan yang dikelola oleh Amazon SageMaker Debugger

Amazon SageMaker Aturan profiler bawaan debugger menganalisis metrik sistem dan operasi kerangka kerja yang dikumpulkan selama pelatihan model. Debugger menawarkan`ProfilerRule`Operasi API yang membantu mengonfigurasi aturan untuk memantau sumber daya dan operasi komputasi pelatihan dan untuk mendeteksi anomali. Misalnya, aturan pembuatan profil dapat membantu Anda mendeteksi apakah ada masalah komputasi seperti kemacetan CPU, waktu tunggu I/O yang berlebihan, beban kerja yang tidak seimbang di seluruh pekerja GPU, dan sumber daya komputasi yang kurang dimanfaatkan. Untuk melihat aturan yang tersedia, lihatDaftar aturan profiler bawaan Debugger.

> **Note**
>
> Aturan bawaan disediakan melalui Amazon SageMaker memproses wadah dan dikelola sepenuhnya oleh SageMaker Debugger tanpa biaya tambahan. Untuk informasi tentang penagihan, lihatAmazon SageMaker Hargahalaman.

Dalam topik berikut, pelajari cara menggunakan aturan bawaan Debugger.

Topik

- Gunakan SageMaker Aturan profiler bawaan debugger dengan pengaturan parameter defaultnya
- Gunakan aturan profiler bawaan Debugger dengan nilai parameter khusus

Gunakan SageMaker Aturan profiler bawaan debugger dengan pengaturan parameter defaultnya

Untuk menambahkan SageMaker Aturan bawaan debugger di estimator Anda, Anda perlu mengonfigurasi `rules`daftar objek. Kode contoh menunjukkan struktur dasar dari daftar SageMaker Aturan bawaan debugger.

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs

rules=[
    ProfilerRule.sagemaker(rule_configs.BuiltInProfilerRuleName_1()),
    ProfilerRule.sagemaker(rule_configs.BuiltInProfilerRuleName_2()),
    ...
    ProfilerRule.sagemaker(rule_configs.BuiltInProfilerRuleName_n()),
    ... # You can also append more debugging rules in the
 Rule.sagemaker(rule_configs.*()) format.
]

estimator=Estimator(
    ...
    rules=rules
)
```

Untuk informasi selengkapnya tentang aturan yang tersedia, lihatDaftar aturan profiler bawaan Debugger.

Untuk menggunakan aturan pembuatan profil dan memeriksa kinerja komputasi dan kemajuan pekerjaan pelatihan Anda, tambahkan`ProfilerReport`aturan SageMaker Debugger. Aturan ini mengaktifkan semua aturan bawaan di bawahDebugger ProfilerRule `ProfilerRule`keluarga. Selanjutnya, aturan ini menghasilkan laporan profil agregat. Untuk informasi selengkapnya, lihatLaporan Profil yang Dihasilkan Menggunakan SageMaker Debugger. Anda dapat menggunakan kode berikut untuk menambahkan aturan laporan pembuatan profil ke estimator pelatihan Anda.

```
from sagemaker.debugger import Rule, rule_configs

rules=[
    ProfilerRule.sagemaker(rule_configs.ProfilerReport())
]
```

Ketika Anda memulai pekerjaan pelatihan dengan`ProfilerReport`aturan, Debugger mengumpulkan data pemanfaatan sumber daya setiap 500 milidetik. Debugger menganalisis pemanfaatan sumber daya untuk mengidentifikasi apakah model Anda mengalami masalah

kemacetan. Jika aturan mendeteksi anomali pelatihan, status evaluasi aturan berubah menjadi`IssueFound`. Anda dapat mengatur tindakan otomatis, seperti memberi tahu masalah pelatihan dan menghentikan pekerjaan pelatihan menggunakan Amazon CloudWatch Acara danAWS Lambda. Untuk informasi selengkapnya, lihat Aksi di Amazon SageMaker Aturan Debugger.

Gunakan aturan profiler bawaan Debugger dengan nilai parameter khusus

Jika Anda ingin menyesuaikan nilai parameter aturan bawaan dan menyesuaikan regex koleksi tensor, konfigurasikan`base_config`dan`rule_parameters`parameter untuk`ProfilerRule.sagemaker`dan`Rule.sagemaker`metode kelas. Dalam kasus`Rule.sagemaker`metode kelas, Anda juga dapat menyesuaikan koleksi tensor melalui`collections_to_save`parameter. Untuk instruksi tentang cara menggunakan`CollectionConfig`kelas, lihatMengonfigurasi Koleksi Tensor MenggunakanCollectionConfig API.

Gunakan templat konfigurasi berikut untuk aturan bawaan untuk menyesuaikan nilai parameter. Dengan mengubah parameter aturan sesuai keinginan, Anda dapat menyesuaikan sensitivitas aturan yang akan dimulai.

- The`base_config`argumen adalah tempat Anda memanggil metode aturan bawaan.
- The`rule_parameters`argumennya adalah untuk menyesuaikan nilai kunci default dari aturan bawaan yang tercantum dalamDaftar aturan profiler bawaan Debugger.

Untuk informasi lebih lanjut tentang kelas, metode, dan parameter aturan Debugger, lihatSageMakerkelas Aturan DebuggerdiAmazon SageMaker SDK Python.

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs, CollectionConfig

rules=[
    ProfilerRule.sagemaker(
        base_config=rule_configs.BuiltInProfilerRuleName(),
        rule_parameters={
                "key": "value"
        }
    )
]
```

Deskripsi parameter dan contoh kustomisasi nilai disediakan untuk setiap aturan diDaftar aturan profiler bawaan Debugger.

Untuk konfigurasi JSON tingkat rendah dari aturan bawaan Debugger menggunakan`CreateTrainingJob`API, lihat[Mengonfigurasi Debugger Menggunakan Amazon SageMaker API](#).

Daftar aturan profiler bawaan Debugger

Menggunakan aturan profiler bawaan Debugger yang disediakan oleh Amazon SageMaker Debugger dan analisis metrik yang dikumpulkan saat melatih model Anda. Aturan bawaan Debugger memantau berbagai kondisi umum yang sangat penting untuk keberhasilan menjalankan pekerjaan pelatihan yang berkinerja. Anda dapat memanggil aturan profiler bawaan menggunakan[Amazon SageMaker Python SDK](#)atau tingkat rendah SageMaker Operasi API. Tidak ada biaya tambahan untuk menggunakan aturan bawaan. Untuk informasi selengkapnya tentang penagihan, lihat[Amazon SageMaker Harga](#)halaman.

> **Note**
>
> Jumlah maksimum aturan profiler bawaan yang dapat Anda lampirkan ke pekerjaan pelatihan adalah 20. SageMaker Debugger sepenuhnya mengelola aturan bawaan dan menganalisis pekerjaan pelatihan Anda secara serempak.

> **Important**
>
> Untuk menggunakan fitur Debugger baru, Anda perlu memutakhirkan SageMaker Python SDK dan pustaka klien SMDebug. Di kernel IPython Anda, notebook Jupyter, atau JupyterLab lingkungan, jalankan kode berikut untuk menginstal versi terbaru dari perpustakaan dan restart kernel.
>
> ```
> import sys
> import IPython
> !{sys.executable} -m pip install -U sagemaker smdebug
> IPython.Application.instance().kernel.do_shutdown(True)
> ```

Aturan profiler

Aturan berikut adalah aturan bawaan Debugger yang dapat dipanggil menggunakan`ProfilerRule.sagemaker`metode kelas.

Aturan bawaan debugger untuk membuat laporan pembuatan profil

Cakupan Validitas	Bawaan Aturan Bawaan
Laporan Profiling untuk setiap SageMaker pekerjaan pelatihan	• ProfilerReport

Aturan bawaan debugger untuk membuat profil pemanfaatan sumber daya sistem perangkat keras (metrik sistem)

Cakupan Validitas	Bawaan Aturan Bawaan
Aturan pemantauan sistem generik untuk apa pun SageMaker pekerjaan pelatihan	• BatchSize • CPUBottleneck • GPUMemoryIncrease • IOBottleneck • LoadBalancing • LowGPUUtilization • OverallSystemUsage

Aturan bawaan debugger untuk membuat profil metrik kerangka kerja

Cakupan Validitas	Bawaan Aturan Bawaan
Aturan pembuatan profil untuk kerangka pembelajaran mendalam (TensorFlow dan PyTorch)	• MaxInitializationTime • OverallFrameworkMetrics • StepOutlier

Warning

MendukungAmazon SageMaker Profiler, SageMakerDebugger menghentikan fitur pembuatan profil kerangka kerja mulai dari TensorFlow 2.11 dan PyTorch 2.0. Anda masih dapat menggunakan fitur ini di versi kerangka kerja dan SDK sebelumnya sebagai berikut.

- SageMaker Python SDK <= v2.130.0

- PyTorch >= v1.6.0, < v2.0
- TensorFlow >= v2.3.1, < v2.11

Lihat juga 16 Maret 2023.

Untuk menggunakan aturan bawaan dengan nilai parameter default— gunakan format konfigurasi berikut:

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs

rules = [
    ProfilerRule.sagemaker(rule_configs.BuiltInRuleName_1()),
    ProfilerRule.sagemaker(rule_configs.BuiltInRuleName_2()),
    ...
    ProfilerRule.sagemaker(rule_configs.BuiltInRuleName_n())
]
```

Untuk menggunakan aturan bawaan dengan menyesuaikan nilai parameter— gunakan format konfigurasi berikut:

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs

rules = [
    ProfilerRule.sagemaker(
        base_config=rule_configs.BuiltInRuleName(),
        rule_parameters={
                "key": "value"
        }
    )
]
```

Untuk menemukan kunci yang tersedia untuk`rule_parameters`parameter, lihat tabel deskripsi parameter.

Contoh kode konfigurasi aturan disediakan untuk setiap aturan bawaan di bawah tabel deskripsi parameter.

- Untuk instruksi lengkap dan contoh penggunaan aturan bawaan Debugger, lihatDebugger Built-in Aturan Contoh Kode.

- Untuk instruksi lengkap tentang penggunaan aturan bawaan dengan tingkat rendah SageMaker Operasi API, lihatMengonfigurasi Debugger Menggunakan Amazon SageMaker API.

ProfilerReport

The ProfilerReport aturan memanggil semua aturan bawaan untuk pemantauan dan pembuatan profil. Ini membuat laporan profil dan pembaruan saat aturan individual dipicu. Anda dapat mengunduh laporan pembuatan profil yang komprehensif saat pekerjaan pelatihan sedang berjalan atau setelah pekerjaan pelatihan selesai. Anda dapat menyesuaikan nilai parameter aturan untuk menyesuaikan sensitivitas aturan pemantauan dan pembuatan profil bawaan. Kode contoh berikut menunjukkan format dasar untuk menyesuaikan parameter aturan bawaan melalui ProfilerReport aturan.

```
rules=[
    ProfilerRule.sagemaker(
        rule_configs.ProfilerReport(
            <BuiltInRuleName>_<parameter_name> = value
        )
    )
]
```

Jika Anda memicu ini ProfilerReport aturan tanpa parameter yang disesuaikan seperti yang ditunjukkan dalam kode contoh berikut, maka ProfilerReport aturan memicu semua aturan bawaan untuk pemantauan dan pembuatan profil dengan nilai parameter defaultnya.

```
rules=[ProfilerRule.sagemaker(rule_configs.ProfilerReport())]
```

Kode contoh berikut menunjukkan cara menentukan dan menyesuaikan aturan CPUBottleNeck`cpu_threshold`parameter dan aturan ioBottleNeck`threshold`Parameter

```
rules=[
    ProfilerRule.sagemaker(
        rule_configs.ProfilerReport(
            CPUBottleneck_cpu_threshold = 90,
            IOBottleneck_threshold = 90
        )
    )
]
```

Untuk menjelajahi apa yang ada dalam laporan profiler, lihatSageMaker Laporan Profil Debugger. Selain itu, karena aturan ini mengaktifkan semua aturan pembuatan profil, Anda juga dapat memeriksa status analisis aturan menggunakanSageMaker UI debugger SageMaker Eksperimen Studio.

Deskripsi Parameter untuk OverallSystemUsage Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`<BuiltInRuleName>_<parameter_name>`	Parameter yang dapat disesuaikan untuk menyesuaikan ambang batas aturan pemantauan dan pembuatan profil bawaan lainnya. Opsional Nilai default: `None`

BatchSize

The BatchSize aturan membantu mendeteksi jika GPU kurang dimanfaatkan karena ukuran batch yang kecil. Untuk mendeteksi masalah ini, aturan ini memantau pemanfaatan CPU rata-rata, pemanfaatan GPU, dan pemanfaatan memori GPU. Jika pemanfaatan pada CPU, GPU, dan memori GPU rata-rata rendah, ini mungkin menunjukkan bahwa pekerjaan pelatihan dapat berjalan pada jenis instance yang lebih kecil atau dapat berjalan dengan ukuran batch yang lebih besar. Analisis ini tidak berfungsi untuk kerangka kerja yang mengalokasikan memori secara berlebihan. Namun, meningkatkan ukuran batch dapat menyebabkan kemacetan pemrosesan atau pemuatan data karena lebih banyak waktu pra-pemrosesan data diperlukan di setiap iterasi.

Deskripsi Parameter untuk BatchSize Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`cpu_threshold_p95`	Mendefinisikan ambang batas untuk kuantil ke-95 pemanfaatan CPU dalam persentase. Opsional Nilai yang valid: Integer Nilai default:`70`(dalam persentase)
`gpu_threshold_p95`	Mendefinisikan ambang batas untuk kuantil 95 pemanfaatan GPU dalam persentase. Opsional Nilai yang valid: Integer Nilai default:`70`(dalam persentase)
`gpu_memory_threshold_p95`	Mendefinisikan ambang batas untuk kuantil ke-95 pemanfaatan memori GPU dalam persentase. Opsional Nilai yang valid: Integer Nilai default:`70`(dalam persentase)

Nama Parameter	Deskripsi
`patience`	Mendefinisikan jumlah titik data yang dapat Anda lampirkan ke aturan mulai evaluasi. Beberapa langkah pertama pekerjaan pelatihan biasanya menunjukkan volume proses data yang tinggi, jadi pertahankan aturan tetap sabar dan cegah agar tidak dipanggil terlalu cepat dengan sejumlah data profil tertentu yang Anda tentukan dengan parameter ini. Opsional Nilai yang valid: Integer Nilai default:`100`
`window`	Ukuran jendela untuk menghitung kuantil. Opsional Nilai yang valid: Integer Nilai default:`500`
`scan_interval_us`	Interval waktu file timeline dipindai. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

CPUBottleneck

Aturan CPUBottleNeck membantu mendeteksi jika GPU kurang dimanfaatkan karena kemacetan CPU. Rule mengembalikan True jika jumlah bottleneck CPU melebihi ambang batas yang telah ditentukan.

Deskripsi Parameter untuk Aturan CPUBottleNeck

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold`	Mendefinisikan ambang batas untuk proporsi waktu yang terhambat dengan total waktu pelatihan. Jika proporsi melebihi persentase yang ditentukan ke parameter ambang batas, aturan akan mengalihkan status aturan ke Benar. Opsional Nilai yang valid: Integer Nilai default:`50`(dalam persentase)
`gpu_threshold`	Ambang batas yang mendefinisikan pemanfaat an GPU rendah. Opsional Nilai yang valid: Integer Nilai default:`10`(dalam persentase)
`cpu_threshold`	Ambang batas yang mendefinisikan pemanfaat an CPU yang tinggi. Opsional Nilai yang valid: Integer

Nama Parameter	Deskripsi
	Nilai default:`90`(dalam persentase)
`patience`	Mendefinisikan jumlah titik data yang dapat Anda lampirkan ke aturan mulai evaluasi. Beberapa langkah pertama pekerjaan pelatihan biasanya menunjukkan volume proses data yang tinggi, jadi pertahankan aturan tetap sabar dan cegah agar tidak dipanggil terlalu cepat dengan sejumlah data profil tertentu yang Anda tentukan dengan parameter ini. Opsional Nilai yang valid: Integer Nilai default:`100`
`scan_interval_us`	Interval waktu dengan mana file timeline dipindai. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

GPUMemoryIncrease

GPUMemoryIncrease aturan membantu mendeteksi peningkatan besar dalam penggunaan memori pada GPU.

Deskripsi Parameter untuk GPUMemoryIncrease Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke

Nama Parameter	Deskripsi
	pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`increase`	Mendefinisikan ambang batas untuk peningkat an memori absolut. Opsional Nilai yang valid: Integer Nilai default:`10`(dalam persentase)
`patience`	Mendefinisikan jumlah titik data yang dapat Anda lampirkan ke aturan mulai evaluasi. Beberapa langkah pertama pekerjaan pelatihan biasanya menunjukkan volume proses data yang tinggi, jadi pertahankan aturan tetap sabar dan cegah agar tidak dipanggil terlalu cepat dengan sejumlah data profil tertentu yang Anda tentukan dengan parameter ini. Opsional Nilai yang valid: Integer Nilai default:`100`
`window`	Ukuran jendela untuk menghitung kuantil. Opsional Nilai yang valid: Integer Nilai default:`500`

Nama Parameter	Deskripsi
`scan_interval_us`	Interval waktu file timeline dipindai. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

IoBottleneck

Aturan ini membantu mendeteksi apakah GPU kurang dimanfaatkan karena kemacetan data IO. Rule mengembalikan True jika jumlah bottleneck IO melebihi ambang batas yang telah ditentukan.

Deskripsi Parameter untuk Aturan IoBottleNeck

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold`	Mendefinisikan ambang batas ketika Aturan untuk mengembalikan Benar. Opsional Nilai yang valid: Integer Nilai default:`50`(dalam persentase)
`gpu_threshold`	Ambang batas yang menentukan kapan GPU dianggap kurang dimanfaatkan. Opsional

Nama Parameter	Deskripsi
	Nilai yang valid: Integer Nilai default:`70`(dalam persentase)
`io_threshold`	Ambang batas yang menentukan waktu tunggu IO yang tinggi. Opsional Nilai yang valid: Integer Nilai default:`50`(dalam persentase)
`patience`	Mendefinisikan jumlah titik data yang dapat Anda lampirkan ke aturan mulai evaluasi. Beberapa langkah pertama pekerjaan pelatihan biasanya menunjukkan volume proses data yang tinggi, jadi pertahankan aturan tetap sabar dan cegah agar tidak dipanggil terlalu cepat dengan sejumlah data profil tertentu yang Anda tentukan dengan parameter ini. Opsional Nilai yang valid: Integer Nilai default:`1000`
`scan_interval_us`	Interval waktu file timeline dipindai. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

LoadBalancing

The LoadBalancing aturan membantu mendeteksi masalah dalam penyeimbangan beban kerja di antara beberapa GPU.

Deskripsi Parameter untuk LoadBalancing Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold`	Mendefinisikan persentase beban kerja. Opsional Nilai yang valid: Integer Nilai default:`0.5`(proporsi tanpa unit)
`patience`	Mendefinisikan jumlah titik data yang dapat Anda lampirkan ke aturan mulai evaluasi. Beberapa langkah pertama pekerjaan pelatihan biasanya menunjukkan volume proses data yang tinggi, jadi pertahankan aturan tetap sabar dan cegah agar tidak dipanggil terlalu cepat dengan sejumlah data profil tertentu yang Anda tentukan dengan parameter ini. Opsional Nilai yang valid: Integer Nilai default:`10`
`scan_interval_us`	Interval waktu file timeline dipindai.

Nama Parameter	Deskripsi
	Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

Pemanfaatan LowGPU

Aturan LowGPUUutilization membantu mendeteksi apakah pemanfaatan GPU rendah atau mengalami fluktuasi. Ini diperiksa untuk setiap GPU pada setiap pekerja. Aturan mengembalikan True jika kuantil ke-95 di bawah threshold_p95 yang menunjukkan kurang dimanfaatkan. Aturan mengembalikan true jika kuantil ke-95 di atas threshold_p95 dan kuantil ke-5 di bawah threshold_p5 yang menunjukkan fluktuasi.

Deskripsi Parameter untuk Aturan LowGPUUtilization

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`threshold_p95`	Ambang batas untuk kuantil ke-95 di bawah mana GPU dianggap kurang dimanfaatkan. Opsional Nilai yang valid: Integer Nilai default:`70`(dalam persentase)
`threshold_p5`	Ambang batas untuk kuantil ke-5. Default adalah 10 persen.

Nama Parameter	Deskripsi
	Opsional Nilai yang valid: Integer Nilai default:`10`(dalam persentase)
`patience`	Mendefinisikan jumlah titik data yang dapat Anda lampirkan ke aturan mulai evaluasi. Beberapa langkah pertama pekerjaan pelatihan biasanya menunjukkan volume proses data yang tinggi, jadi pertahankan aturan tetap sabar dan cegah agar tidak dipanggil terlalu cepat dengan sejumlah data profil tertentu yang Anda tentukan dengan parameter ini. Opsional Nilai yang valid: Integer Nilai default:`1000`
`window`	Ukuran jendela untuk menghitung kuantil. Opsional Nilai yang valid: Integer Nilai default:`500`
`scan_interval_us`	Interval waktu file timeline dipindai. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

OverallSystemUsage

The OverallSystemUsage aturan mengukur penggunaan sistem secara keseluruhan per node pekerja. Aturan saat ini hanya mengumpulkan nilai per node dan menghitung persentilnya.

Deskripsi Parameter untuk OverallSystemUsage Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`scan_interval_us`	Interval waktu untuk memindai file timeline. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

MaxInitializationTime

The MaxInitializationTime aturan membantu mendeteksi jika inisialisasi pelatihan memakan terlalu banyak waktu. Aturan menunggu sampai langkah pertama tersedia.

Deskripsi Parameter untuk MaxInitializationTime Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan

Nama Parameter	Deskripsi
	Nilai valid: String
`threshold`	Mendefinisikan ambang batas dalam hitungan menit untuk menunggu langkah pertama tersedia. Opsional Nilai yang valid: Integer Nilai default:`20`(dalam hitungan menit)
`scan_interval_us`	Interval waktu dengan mana file timeline dipindai. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

OverallFrameworkMetrics

The OverallFrameworkMetrics aturan merangkum waktu yang dihabiskan untuk metrik kerangka kerja, seperti pass maju dan mundur, dan pemuatan data.

Deskripsi Parameter untuk OverallFrameworkMetrics Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String

Nama Parameter	Deskripsi
`scan_interval_us`	Interval waktu untuk memindai file timeline. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

StepOutlier

The StepOutlier aturan membantu mendeteksi outlier dalam durasi langkah. Aturan ini kembali`True`jika ada outlier dengan durasi langkah lebih besar dari`stddev`sigma dari seluruh durasi langkah dalam rentang waktu.

Deskripsi Parameter untuk StepOutlier Aturan

Nama Parameter	Deskripsi
`base_trial`	Nama pekerjaan pelatihan uji coba dasar. Parameter ini secara otomatis diatur ke pekerjaan pelatihan saat ini oleh Amazon SageMaker Debugger. Diperlukan Nilai valid: String
`stddev`	Mendefinisikan faktor yang digunakan untuk mengalikan standar deviasi. Misalnya, aturan dipanggil secara default ketika durasi langkah lebih besar atau lebih kecil dari 5 kali standar deviasi. Opsional Nilai yang valid: Integer Nilai default:`5`(dalam hitungan menit)

Nama Parameter	Deskripsi
`mode`	Mode di mana langkah-langkah telah disimpan dan Aturan mana yang harus dijalankan. Per aturan default akan berjalan pada langkah-l angkah dari fase EVAL dan TRAIN Opsional Nilai yang valid: Integer Nilai default:`5`(dalam hitungan menit)
`n_outliers`	Berapa banyak outlier yang harus diabaikan sebelum aturan mengembalikan True Opsional Nilai yang valid: Integer Nilai default: `10`
`scan_interval_us`	Interval waktu dengan mana file timeline dipindai. Opsional Nilai yang valid: Integer Nilai default:`60000000`(dalam mikrodetik)

Amazon SageMaker Debugger UI di Eksperimen Klasik Amazon SageMaker Studio

Gunakan dasbor Amazon SageMaker Debugger Insights di Amazon SageMaker Studio Classic Experiments untuk menganalisis performa model dan hambatan sistem saat menjalankan tugas pelatihan di instans Amazon Elastic Compute Cloud (Amazon EC2). Dapatkan wawasan tentang pekerjaan pelatihan Anda dan tingkatkan kinerja dan akurasi pelatihan model Anda dengan dasbor Debugger. Secara default, Debugger memantau metrik sistem (CPU, GPU, memori GPU, jaringan, dan data I/O) setiap 500 milidetik dan tensor keluaran dasar (kehilangan dan akurasi) setiap 500 iterasi untuk pekerjaan pelatihan. Anda juga dapat menyesuaikan nilai parameter konfigurasi

Debugger lebih lanjut dan menyesuaikan interval penyimpanan melalui UI Studio Classic atau menggunakan Amazon Python SageMaker SDK.

> **Important**
>
> Jika Anda menggunakan aplikasi Studio Classic yang sudah ada, hapus aplikasi dan mulai ulang untuk menggunakan fitur Studio Classic terbaru. Untuk petunjuk tentang cara memulai ulang dan memperbarui lingkungan Studio Classic, lihat Memperbarui Amazon SageMaker Studio Classic.

Topik

Buka dasbor Amazon SageMaker Debugger Insights

Di dasbor SageMaker Debugger Insights di Studio Classic, Anda dapat melihat informasi pemanfaatan sumber daya komputasi, pemanfaatan sumber daya, dan kemacetan sistem dari pekerjaan pelatihan Anda yang berjalan di instans Amazon EC2 secara real time dan setelah pelatihan

> **Note**
>
> Dasbor SageMaker Debugger Insights menjalankan aplikasi Studio Classic pada `ml.m5.4xlarge` instance untuk memproses dan merender visualisasi. Setiap tab SageMaker Debugger Insights menjalankan satu sesi kernel Studio Classic. Beberapa sesi kernel untuk beberapa tab SageMaker Debugger Insights berjalan pada satu instance. Saat Anda menutup tab SageMaker Debugger Insights, sesi kernel yang sesuai juga ditutup. Aplikasi Studio Classic tetap aktif dan dikenakan biaya untuk penggunaan `ml.m5.4xlarge` instans. Untuk informasi tentang harga, lihat halaman SageMaker Harga Amazon.

> **Important**
>
> Setelah selesai menggunakan dasbor SageMaker Debugger Insights, Anda harus mematikan `ml.m5.4xlarge` instance untuk menghindari biaya yang bertambah. Untuk petunjuk tentang cara mematikan instance, lihatMatikan instans Amazon SageMaker Debugger Insights.

Untuk membuka dasbor SageMaker Debugger Insights

1. Di halaman Beranda Studio Classic, pilih Eksperimen di panel navigasi kiri.
2. Cari pekerjaan pelatihan Anda di halaman Eksperimen. Jika tugas latihan Anda disiapkan dengan Eksperimen yang dijalankan, pekerjaan tersebut akan muncul di tab Eksperimen; jika Anda tidak menyiapkan Eksperimen yang dijalankan, pekerjaan tersebut akan muncul di tab Jalankan yang tidak ditetapkan.
3. Pilih (klik) tautan nama pekerjaan pelatihan untuk melihat detail pekerjaan.
4. Di bawah menu OVERVIEW, pilih Debuggger. Ini harus menunjukkan dua bagian berikut.
   - Di bagian Aturan debugger, Anda dapat menelusuri status aturan bawaan Debugger yang terkait dengan pekerjaan pelatihan.
   - Di bagian wawasan Debugger, Anda dapat menemukan tautan untuk membuka Wawasan SageMaker Debugger di dasbor.
5. Di bagian SageMaker Debugger Insights, pilih tautan nama pekerjaan pelatihan untuk membuka dasbor SageMaker Debugger Insights. Ini membuka jendela Debug [your-training-job-name]. Di jendela ini, Debugger memberikan ikhtisar kinerja komputasi pekerjaan pelatihan Anda di instans Amazon EC2 dan membantu Anda mengidentifikasi masalah dalam pemanfaatan sumber daya komputasi.

Anda juga dapat mengunduh laporan profil agregat dengan menambahkan ProfilerReportaturan bawaan Debugger. SageMaker Untuk selengkapnya, lihat Mengkonfigurasi Aturan Profiler Bawaan dan Laporan Profil yang Dihasilkan Menggunakan SageMaker Debugger.

Pengontrol SageMaker dasbor Amazon Debugger Insights

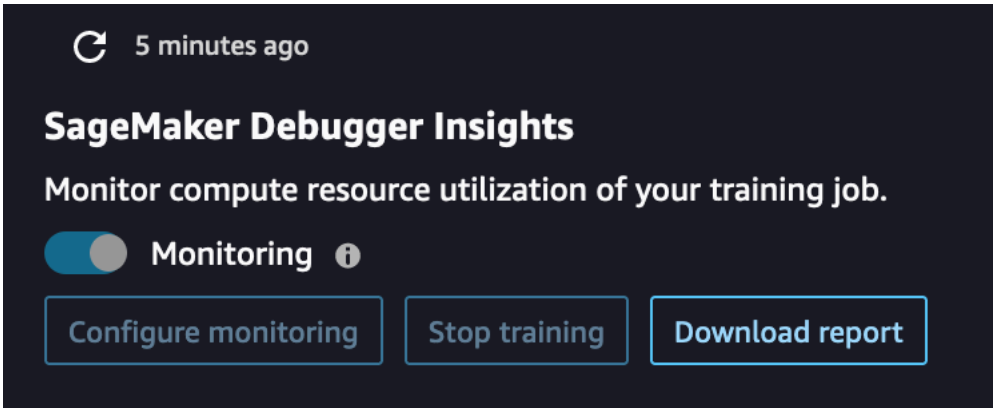
Ada berbagai komponen pengontrol Debugger untuk pemantauan dan pembuatan profil. Dalam panduan ini, Anda belajar tentang komponen pengontrol Debugger.

> **Note**
>
> Dasbor SageMaker Debugger Insights menjalankan aplikasi Studio Classic pada `ml.m5.4xlarge` instance untuk memproses dan merender visualisasi. Setiap tab SageMaker Debugger Insights menjalankan satu sesi kernel Studio Classic. Beberapa sesi kernel untuk beberapa tab SageMaker Debugger Insights berjalan pada satu instance. Saat Anda menutup tab SageMaker Debugger Insights, sesi kernel yang sesuai juga ditutup. Aplikasi Studio Classic tetap aktif dan dikenakan biaya untuk penggunaan `ml.m5.4xlarge` instans. Untuk informasi tentang harga, lihat halaman SageMaker Harga Amazon.

> **Important**
>
> Setelah selesai menggunakan dasbor SageMaker Debugger Insights, matikan `ml.m5.4xlarge` instance untuk menghindari biaya yang bertambah. Untuk petunjuk tentang cara mematikan instance, lihatMatikan instans Amazon SageMaker Debugger Insights.

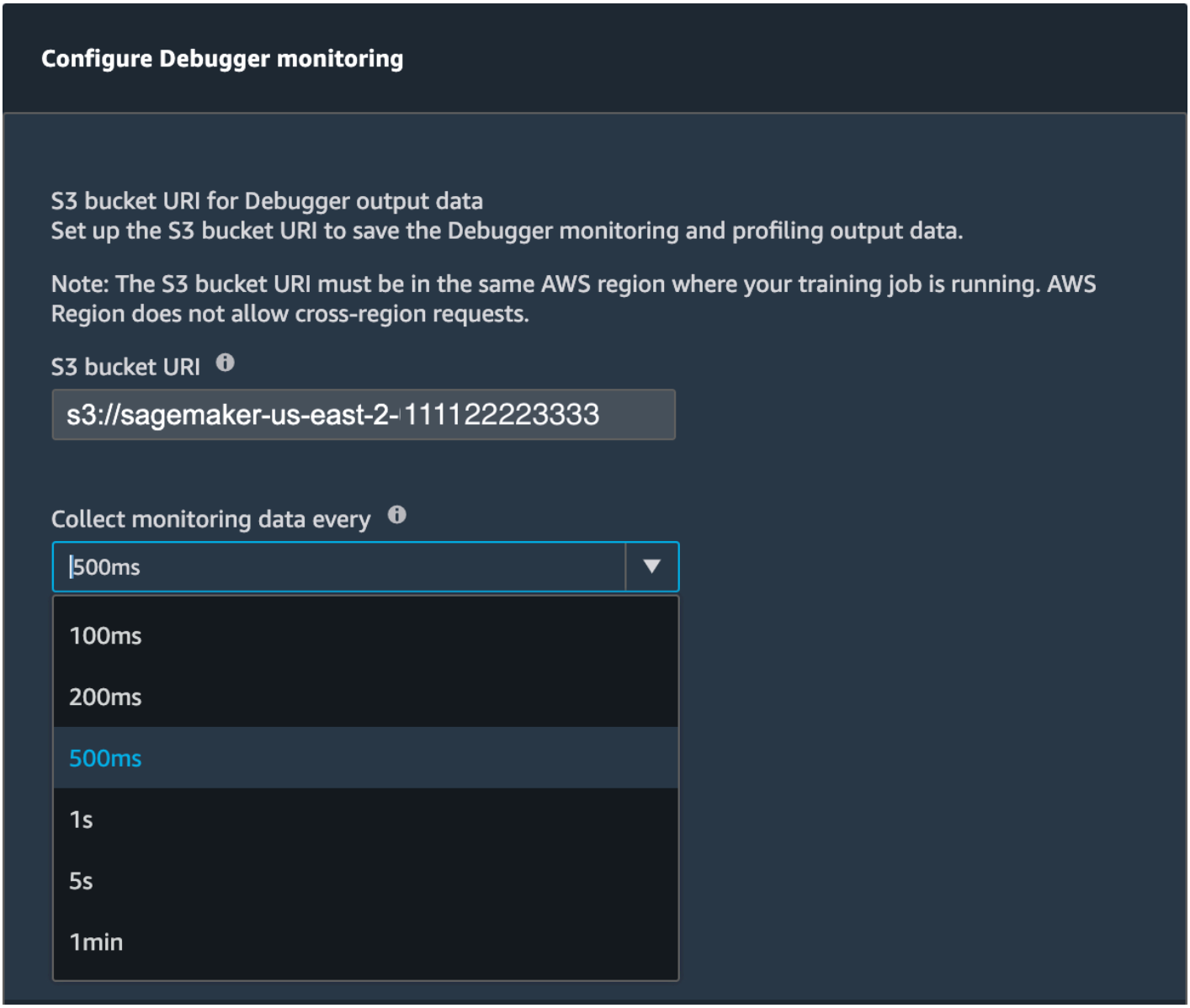
SageMaker UI pengontrol Wawasan Debugger

Menggunakan pengontrol Debugger yang terletak di sudut kiri atas dasbor Insights, Anda dapat menyegarkan dasbor, mengonfigurasi, atau memperbarui pengaturan Debugger untuk memantau metrik sistem, menghentikan pekerjaan pelatihan, dan mengunduh laporan pembuatan profil Debugger.



- Jika Anda ingin menyegarkan dasbor secara manual, pilih tombol refresh (panah bundar di sudut kiri atas) seperti yang ditunjukkan pada tangkapan layar sebelumnya.

- Tombol sakelar Monitoring aktif secara default untuk setiap pekerjaan SageMaker pelatihan yang dimulai menggunakan Python SageMaker SDK. Jika tidak diaktifkan, Anda dapat menggunakan tombol sakelar untuk memulai pemantauan. Selama pemantauan, Debugger hanya mengumpulkan metrik pemanfaatan sumber daya untuk mendeteksi masalah komputasi seperti kemacetan CPU dan kurangnya pemanfaatan GPU. Untuk daftar lengkap masalah pemanfaatan sumber daya yang dipantau Debugger, lihat. Aturan bawaan debugger untuk membuat profil pemanfaatan sumber daya sistem perangkat keras (metrik sistem)
- Tombol Configure monitoring membuka jendela pop-up yang dapat Anda gunakan untuk mengatur atau memperbarui frekuensi pengumpulan data dan jalur S3 untuk menyimpan data.



Anda dapat menentukan nilai untuk bidang berikut.

- URI bucket S3: Tentukan URI bucket S3 dasar.
- Kumpulkan data pemantauan setiap: Pilih interval waktu untuk mengumpulkan metrik sistem. Anda dapat memilih salah satu interval pemantauan dari daftar dropdown. Interval yang tersedia adalah 100 milidetik, 200 milidetik, 500 milidetik (default), 1 detik, 5 detik, dan 1 menit.

> **Note**
>
> Jika Anda memilih salah satu interval waktu yang lebih rendah, Anda meningkatkan perincian metrik pemanfaatan sumber daya, sehingga Anda dapat menangkap lonjakan dan anomali dengan resolusi waktu yang lebih tinggi. Namun, semakin tinggi resolusinya, semakin besar ukuran metrik sistem untuk diproses. Ini mungkin memperkenalkan overhead tambahan dan berdampak pada keseluruhan waktu pelatihan dan pemrosesan.

- Dengan menggunakan tombol Stop training, Anda dapat menghentikan pekerjaan pelatihan ketika Anda menemukan anomali dalam pemanfaatan sumber daya.
- Dengan menggunakan tombol Unduh laporan, Anda dapat mengunduh laporan profil agregat dengan menggunakan ProfilerReportaturan bawaan Debugger. SageMaker Tombol diaktifkan saat Anda menambahkan ProfilerReportaturan bawaan ke estimator. Untuk selengkapnya, lihat Mengkonfigurasi Aturan Profiler Bawaan dan Laporan Profil yang Dihasilkan Menggunakan SageMaker Debugger.

Jelajahi dasbor Amazon SageMaker Debugger Insights

Saat memulai pekerjaan SageMaker pelatihan, SageMaker Debugger mulai memantau pemanfaatan sumber daya instans Amazon EC2 secara default. Anda dapat melacak tingkat pemanfaatan sistem, ikhtisar statistik, dan analisis aturan bawaan melalui dasbor Wawasan. Panduan ini memandu Anda melalui konten dasbor SageMaker Debugger Insights di bawah tab berikut: Metrik dan Aturan Sistem.

> **Note**
>
> Dasbor SageMaker Debugger Insights menjalankan aplikasi Studio Classic pada `ml.m5.4xlarge` instance untuk memproses dan merender visualisasi. Setiap tab SageMaker Debugger Insights menjalankan satu sesi kernel Studio Classic. Beberapa sesi kernel untuk beberapa tab SageMaker Debugger Insights berjalan pada satu instance. Saat Anda menutup tab SageMaker Debugger Insights, sesi kernel yang sesuai juga ditutup.

Aplikasi Studio Classic tetap aktif dan dikenakan biaya untuk penggunaan `ml.m5.4xlarge` instans. Untuk informasi tentang harga, lihat halaman SageMaker Harga Amazon.

Important

Setelah selesai menggunakan dasbor SageMaker Debugger Insights, matikan `ml.m5.4xlarge` instance untuk menghindari biaya yang bertambah. Untuk petunjuk tentang cara mematikan instance, lihatMatikan instans Amazon SageMaker Debugger Insights.

Important

Dalam laporan, plot dan rekomendasi disediakan untuk tujuan informasi dan tidak definitif. Anda bertanggung jawab untuk membuat penilaian independen Anda sendiri atas informasi tersebut.

Topik

- Metrik sistem
- Aturan

Metrik sistem

Di tab Metrik Sistem, Anda dapat menggunakan tabel ringkasan dan plot waktu untuk memahami pemanfaatan sumber daya.

Ringkasan pemanfaatan sumber daya

Tabel ringkasan ini menunjukkan statistik metrik pemanfaatan sumber daya komputasi dari semua node (dilambangkan sebagai algo- n). Metrik pemanfaatan sumber daya meliputi pemanfaatan CPU total, pemanfaatan GPU total, pemanfaatan memori CPU total, pemanfaatan memori GPU total, total waktu tunggu I/O, dan total jaringan dalam byte. Tabel menunjukkan nilai minimum dan maksimum, dan persentil p99, p90, dan p50.

System Metrics Rules

Resource utilization summary

System usage statistics

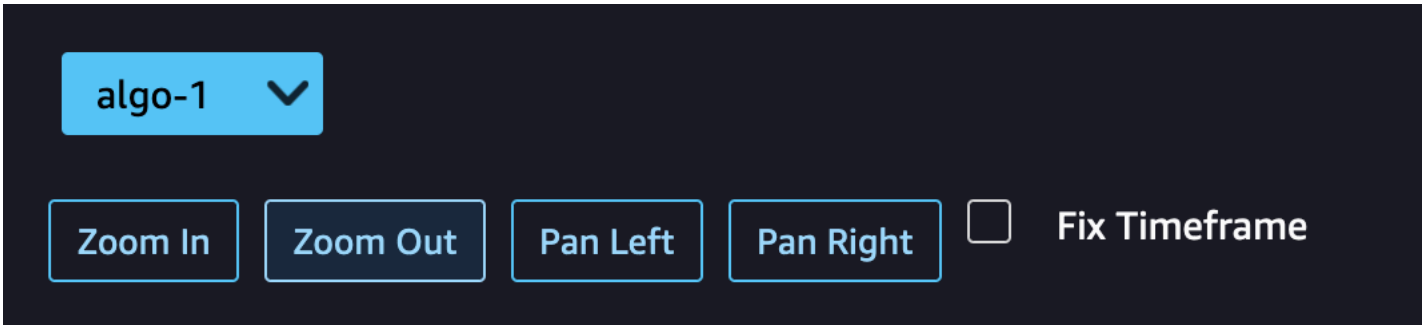
Node	Metric	Unit	Max	p99	p95	p50	Min
algo-1	Network	MB/s	37.82	33.68	32.83	12.39	0
algo-2	Network	MB/s	37.51	33.51	32.69	9.54	0
algo-1	GPU	%	69	20.61	18.27	6.81	0
algo-2	GPU	%	70	20.89	18.68	6.53	0
algo-1	CPU	%	100	94.58	78.95	51.71	0
algo-2	CPU	%	100	94.76	78.48	49.72	0
algo-1	CPU memory	%	5	4.98	4.92	4.16	1
algo-2	CPU memory	%	5	4.98	4.91	4.15	1
algo-1	GPU memory	%	32	9.6	7.71	2.27	0
algo-2	GPU memory	%	33	9.59	7.76	2.21	0
algo-1	I/O	%	100	20.41	0	0	0
algo-2	I/O	%	92	19.45	0	0	0

Plot deret waktu pemanfaatan sumber daya

Gunakan grafik deret waktu untuk melihat rincian lebih lanjut tentang pemanfaatan sumber daya dan mengidentifikasi pada interval waktu berapa setiap instance menunjukkan tingkat pemanfaatan yang tidak diinginkan, seperti pemanfaatan GPU yang rendah dan kemacetan CPU yang dapat menyebabkan pemborosan instance mahal.

UI pengontrol grafik deret waktu

Tangkapan layar berikut menunjukkan pengontrol UI untuk menyesuaikan grafik deret waktu.

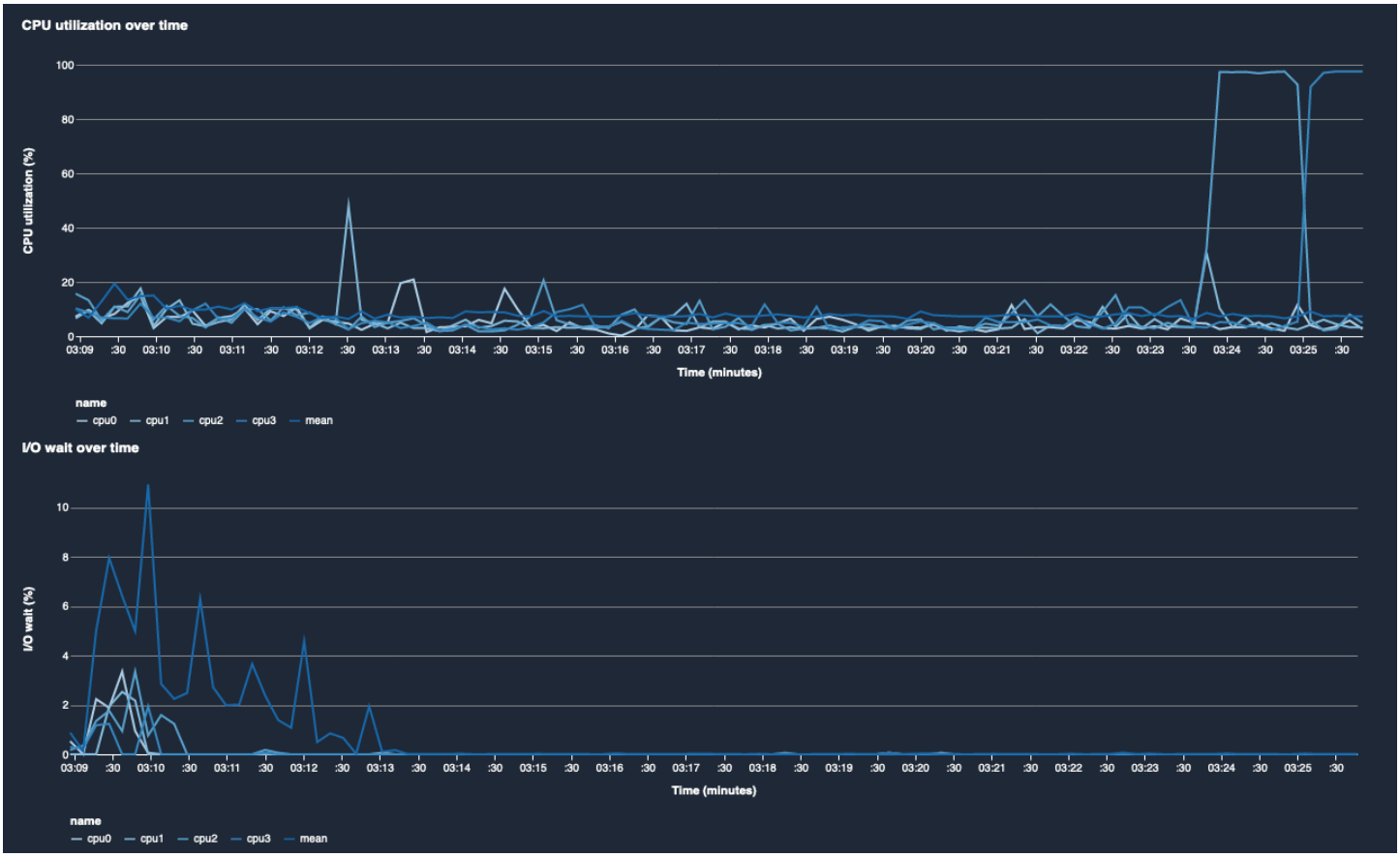


- algo-1: Gunakan menu dropdown ini untuk memilih node yang ingin Anda lihat.
- Zoom In: Gunakan tombol ini untuk memperbesar grafik deret waktu dan melihat interval waktu yang lebih pendek.

- Perkecil: Gunakan tombol ini untuk memperkecil grafik deret waktu dan melihat interval waktu yang lebih luas.
- Pan Kiri: Pindahkan grafik deret waktu ke interval waktu sebelumnya.
- Pan Kanan: Pindahkan grafik deret waktu ke interval waktu berikutnya.
- Perbaiki Jangka Waktu: Gunakan kotak centang ini untuk memperbaiki atau mengembalikan grafik deret waktu untuk menampilkan seluruh tampilan dari titik data pertama ke titik data terakhir.

Pemanfaatan CPU dan waktu tunggu I/O

Dua grafik pertama menunjukkan pemanfaatan CPU dan waktu tunggu I/O dari waktu ke waktu. Secara default, grafik menunjukkan rata-rata tingkat pemanfaatan CPU dan waktu tunggu I/O yang dihabiskan pada inti CPU. Anda dapat memilih satu atau lebih inti CPU dengan memilih label untuk membuat grafik pada bagan tunggal dan membandingkan pemanfaatan di seluruh inti. Anda dapat menyeret dan memperbesar dan memperkecil untuk melihat lebih dekat pada interval waktu tertentu.

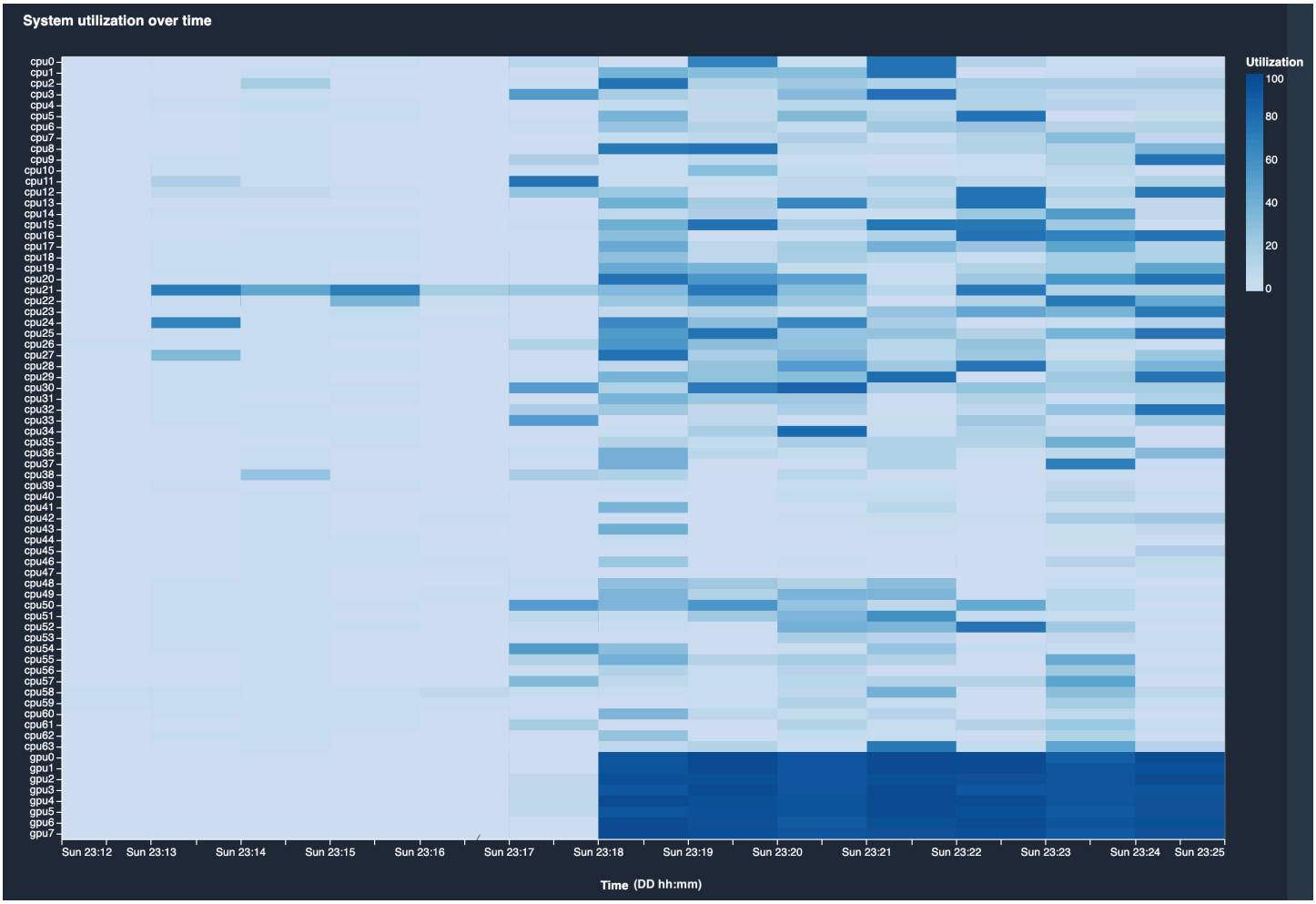


Pemanfaatan GPU dan pemanfaatan memori GPU

Grafik berikut menunjukkan pemanfaatan GPU dan pemanfaatan memori GPU dari waktu ke waktu. Secara default, grafik menunjukkan tingkat pemanfaatan rata-rata dari waktu ke waktu. Anda dapat memilih label inti GPU untuk melihat tingkat pemanfaatan setiap inti. Mengambil rata-rata tingkat pemanfaatan atas jumlah total inti GPU menunjukkan pemanfaatan rata-rata dari seluruh sumber daya sistem perangkat keras. Dengan melihat tingkat pemanfaatan rata-rata, Anda dapat memeriksa keseluruhan penggunaan sumber daya sistem dari instans Amazon EC2. Gambar berikut menunjukkan contoh pekerjaan pelatihan pada `ml.p3.16xlarge` instance dengan 8 core GPU. Anda dapat memantau apakah pekerjaan pelatihan didistribusikan dengan baik, sepenuhnya memanfaatkan semua GPU.

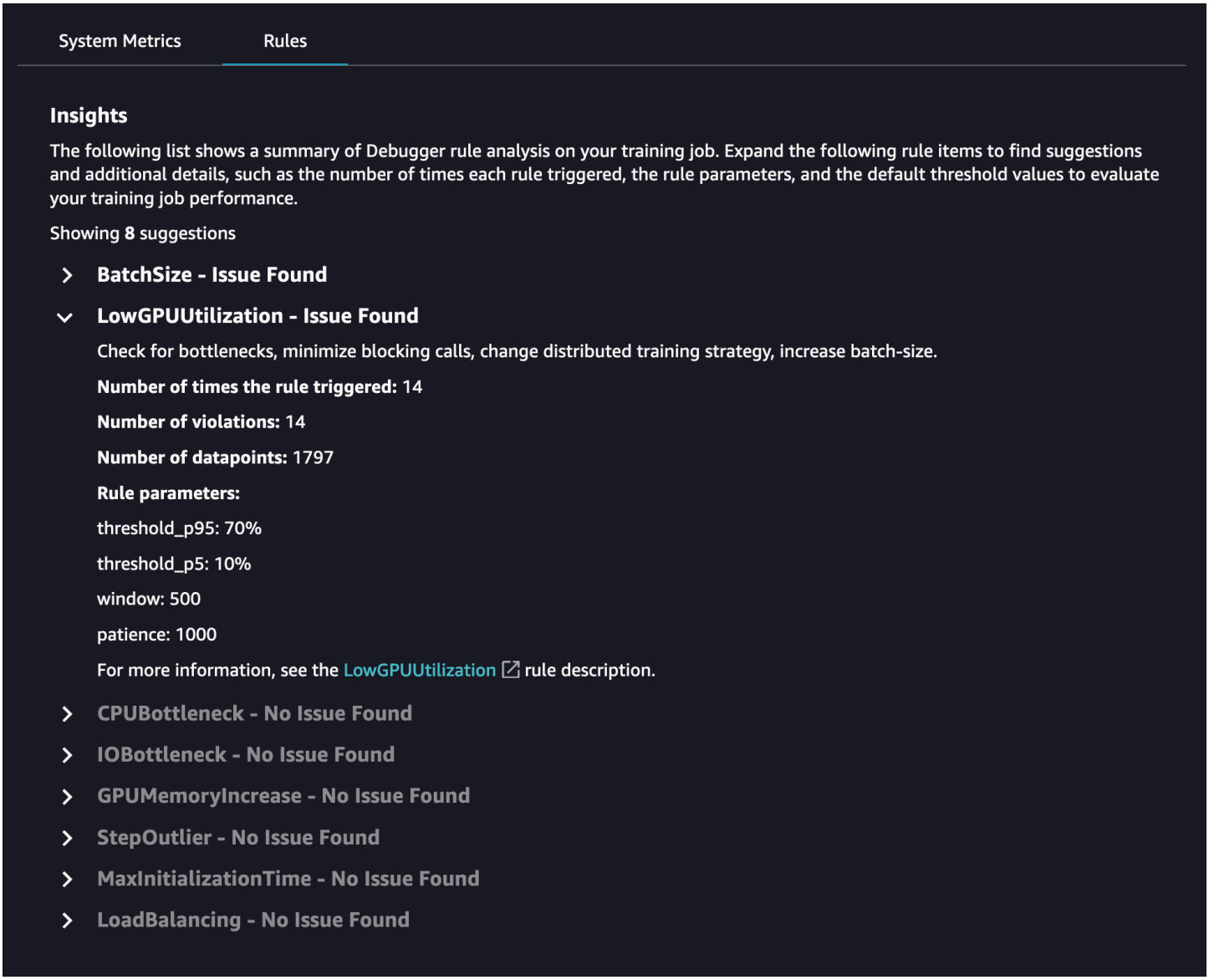
Pemanfaatan sistem secara keseluruhan dari waktu ke waktu

Peta panas berikut menunjukkan contoh seluruh pemanfaatan sistem `ml.p3.16xlarge` instance dari waktu ke waktu, diproyeksikan ke plot dua dimensi. Setiap inti CPU dan GPU tercantum dalam sumbu vertikal, dan pemanfaatannya dicatat dari waktu ke waktu dengan skema warna, di mana warna-warna cerah mewakili pemanfaatan rendah dan warna yang lebih gelap mewakili pemanfaatan tinggi. Lihat bilah warna berlabel di sisi kanan plot untuk mengetahui tingkat warna mana yang sesuai dengan tingkat pemanfaatan mana.



Aturan

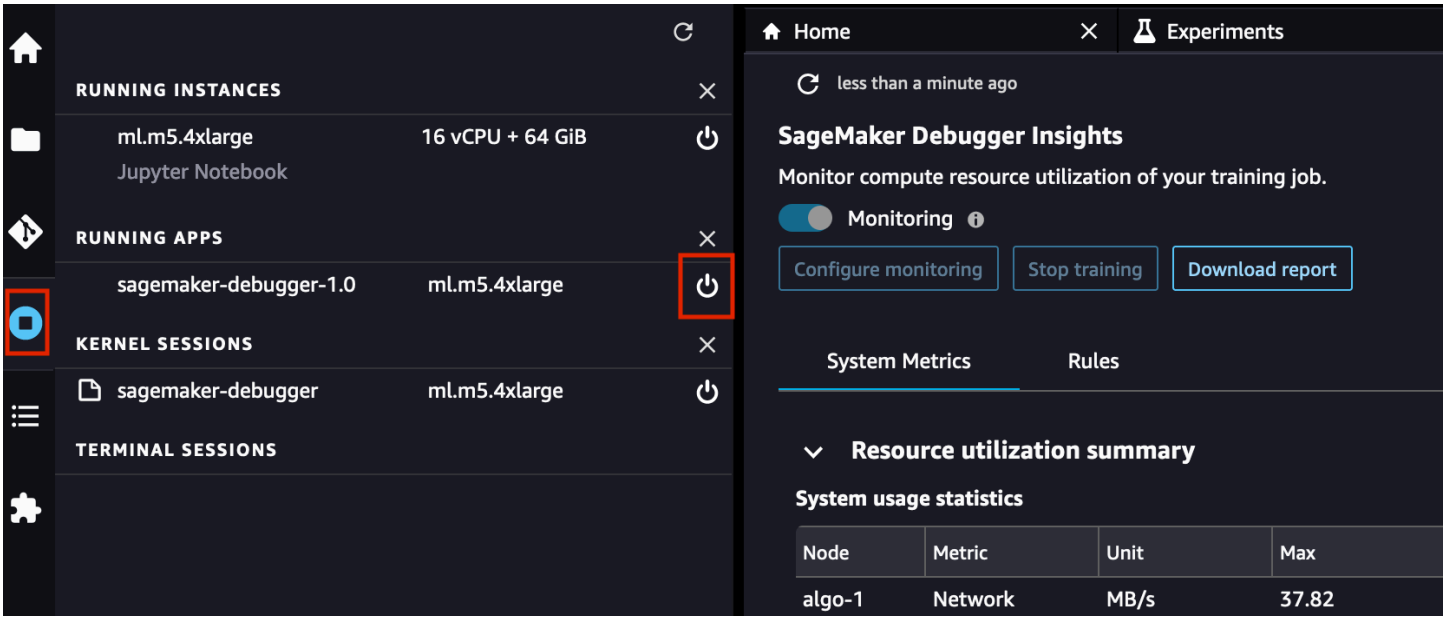
Gunakan tab Aturan untuk menemukan ringkasan analisis aturan pembuatan profil pada pekerjaan pelatihan Anda. Jika aturan pembuatan profil diaktifkan dengan pekerjaan pelatihan, teks akan muncul disorot dengan teks putih solid. Aturan tidak aktif diredupkan dalam teks abu-abu. Untuk mengaktifkan aturan ini, ikuti instruksi dithe section called “Konfigurasikan aturan profiler bawaan”.



Matikan instans Amazon SageMaker Debugger Insights

Saat Anda tidak menggunakan dasbor SageMaker Debugger Insights, Anda harus mematikan instance aplikasi untuk menghindari biaya tambahan.

Untuk mematikan instance aplikasi SageMaker Debugger Insights di Studio Classic



1. Di Studio Classic, pilih ikon Running Instances dan Kernels
   ()
2. Di bawah daftar RUNNING APPS, cari aplikasi sagemaker-debugger-1.0. Pilih ikon shutdown
   ( )
   di sebelah aplikasi. Dasbor SageMaker Debugger Insights berjalan pada sebuah instance. `ml.m5.4xlarge` Instance ini juga menghilang dari RUNNING INSTANCES saat Anda mematikan aplikasi sagemaker-debugger-1.0.

SageMaker Laporan interaktif debugger

Terima laporan pembuatan profil yang dibuat secara otomatis oleh Debugger. Laporan Debugger memberikan wawasan tentang pekerjaan pelatihan Anda dan menyarankan rekomendasi untuk meningkatkan kinerja model Anda. Screenshot berikut menunjukkan kolase dari laporan profiling Debugger. Untuk mempelajari selengkapnya, lihat SageMaker Laporan profil debugger.

> **Note**
>
> Anda dapat mengunduh laporan Debugger saat pekerjaan pelatihan Anda berjalan atau setelah pekerjaan selesai. Selama pelatihan, Debugger secara bersamaan memperbarui laporan yang mencerminkan status evaluasi aturan saat ini. Anda dapat mengunduh laporan Debugger lengkap hanya setelah pekerjaan pelatihan selesai.

> **Important**
>
> Dalam laporan, plot dan dan rekomendasi disediakan untuk tujuan informasi dan tidak definitif. Anda bertanggung jawab untuk membuat penilaian independen Anda sendiri atas informasi tersebut.

SageMaker Laporan profil debugger

Untuk apapun SageMaker pekerjaan pelatihan, SageMaker DebuggerProfilerReportaturan memanggil semuaaturan pemantauan dan pembuatan profildan menggabungkan analisis aturan ke dalam laporan yang komprehensif. Mengikuti panduan ini, unduh laporan menggunakanAmazon SageMaker SDK Pythonatau konsol S3, dan pelajari apa yang dapat Anda interpretasikan dari hasil pembuatan profil.

> **Important**
>
> Dalam laporan tersebut, plot dan dan rekomendasi disediakan untuk tujuan informasi dan tidak definitif. Anda bertanggung jawab untuk membuat penilaian independen Anda sendiri atas informasi tersebut.

Unduh SageMaker Laporan profil debugger

Unduh SageMaker Laporan pembuatan profil debugger saat pekerjaan pelatihan Anda berjalan atau setelah pekerjaan selesai menggunakanAmazon SageMaker SDK PythondanAWS Command Line Interface(CLI).

> **Note**
>
> Untuk mendapatkan laporan pembuatan profil yang dihasilkan oleh SageMaker Debugger, Anda harus menggunakan built-inProfilerReportaturan yang ditawarkan oleh SageMaker Debugger. Untuk mengaktifkan aturan dengan pekerjaan pelatihan Anda, lihatKonfigurasikan Aturan Profiler Bawaan.

> **Tip**
>
> Anda juga dapat mengunduh laporan dengan satu klik di SageMaker Dasbor wawasan Studio Debugger. Ini tidak memerlukan skrip tambahan untuk mengunduh laporan. Untuk mengetahui cara mengunduh laporan dari Studio, lihatBuka dasbor Amazon SageMaker Debugger Insights.

Download using SageMaker Python SDK and AWS CLI

1. Periksa URI basis output S3 default pekerjaan saat ini.

   ```
   estimator.output_path
   ```

2. Periksa nama pekerjaan saat ini.

   ```
   estimator.latest_training_job.job_name
   ```

3. Laporan pembuatan profil Debugger disimpan di bawah<default-s3-output-base-uri>/<training-job-name>/rule-output. Konfigurasikan jalur keluaran aturan sebagai berikut:

```
rule_output_path = estimator.output_path +
 estimator.latest_training_job.job_name + "/rule-output"
```

4. Untuk memeriksa apakah laporan dibuat, daftar direktori dan file secara rekursif di bawahrule_output_pathmemakaiaws s3 lsdengan--recursivepilihan.

```
! aws s3 ls {rule_output_path} --recursive
```

Ini akan mengembalikan daftar lengkap file di bawah folder yang dibuat otomatis yang dinamai sebagaiProfilerReport-1234567890. Nama folder adalah kombinasi dari string:ProfilerReportdan tag 10 digit unik berdasarkan stempel waktu Unix saat ProfilerReportaturan dimulai.

```
s3://sagemaker-us-east-2-111122223333 /sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output
2020-11-28 07:26:08     452088 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-report.html
2020-11-28 07:26:07     324474 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-report.ipynb
2020-11-28 07:26:03       1122 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/BatchSize.json
2020-11-28 07:26:03      10349 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/CPUBottleneck.json
2020-11-28 07:26:03        126 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/Dataloader.json
2020-11-28 07:26:03        130 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/GPUMemoryIncrease.json
2020-11-28 07:26:03       1997 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/IOBottleneck.json
2020-11-28 07:26:03        785 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/LoadBalancing.json
2020-11-28 07:26:03        728 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/LowGPUUtilization.json
2020-11-28 07:26:03        233 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/MaxInitializationTime.json
2020-11-28 07:26:03       1585 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/OverallFrameworkMetrics.json
2020-11-28 07:26:03        575 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/OverallSystemUsage.json
2020-11-28 07:26:03       2208 sagemaker-debugger-mnist-byoc-tf2-2020-11-28-06-32-33-097/rule-output/ProfilerReport-1606545153/profiler-output/profiler-reports/StepOutlier.json
```

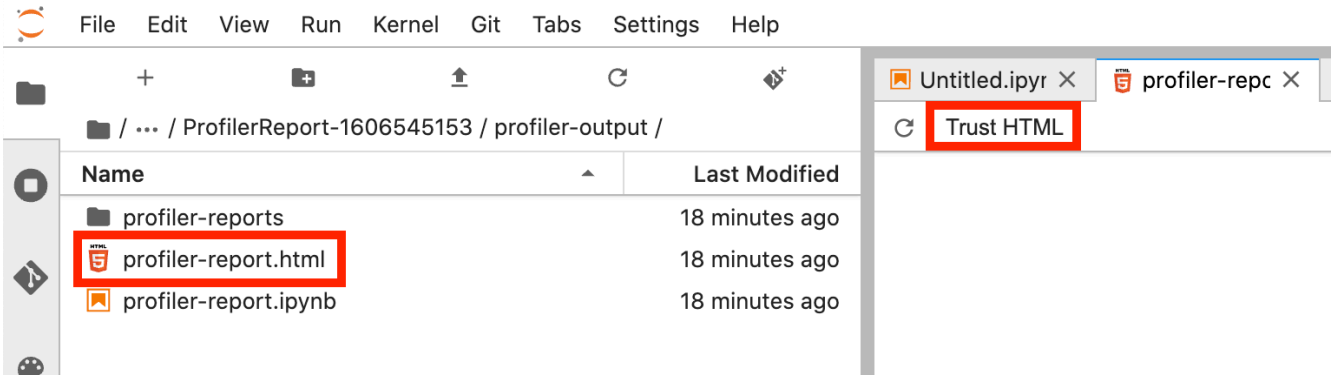
Theprofiler-report.htmladalah laporan profil yang dibuat secara otomatis oleh Debugger. File yang tersisa adalah komponen analisis aturan bawaan yang disimpan di JSON dan notebook Jupyter yang digunakan untuk menggabungkannya ke dalam laporan.

5. Unduh file secara rekursif menggunakanaws s3 cp. Perintah berikut menyimpan semua file output aturan keProfilerReport-1234567890folder di bawah direktori kerja saat ini.

```
! aws s3 cp {rule_output_path} ./ --recursive
```

> **Tip**
>
> Jika menggunakan server notebook Jupyter, jalankan!pwduntuk memeriksa ulang direktori kerja saat ini.

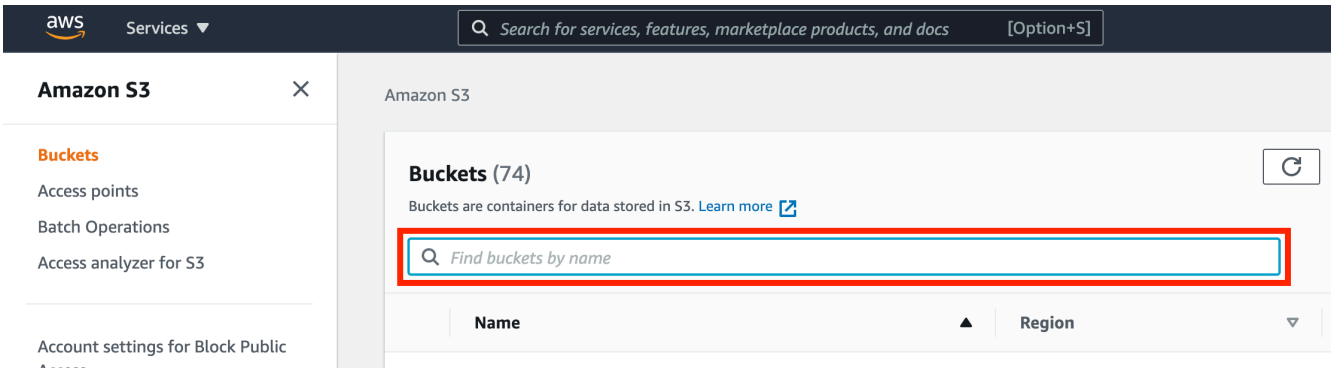
6. Di bawah`/ProfilerReport-1234567890/profiler-output`direktori, buka`profiler-report.html`. Jika menggunakan JupyterLab, pilihPercaya HTMLuntuk melihat laporan pembuatan profil Debugger yang dibuat secara otomatis.



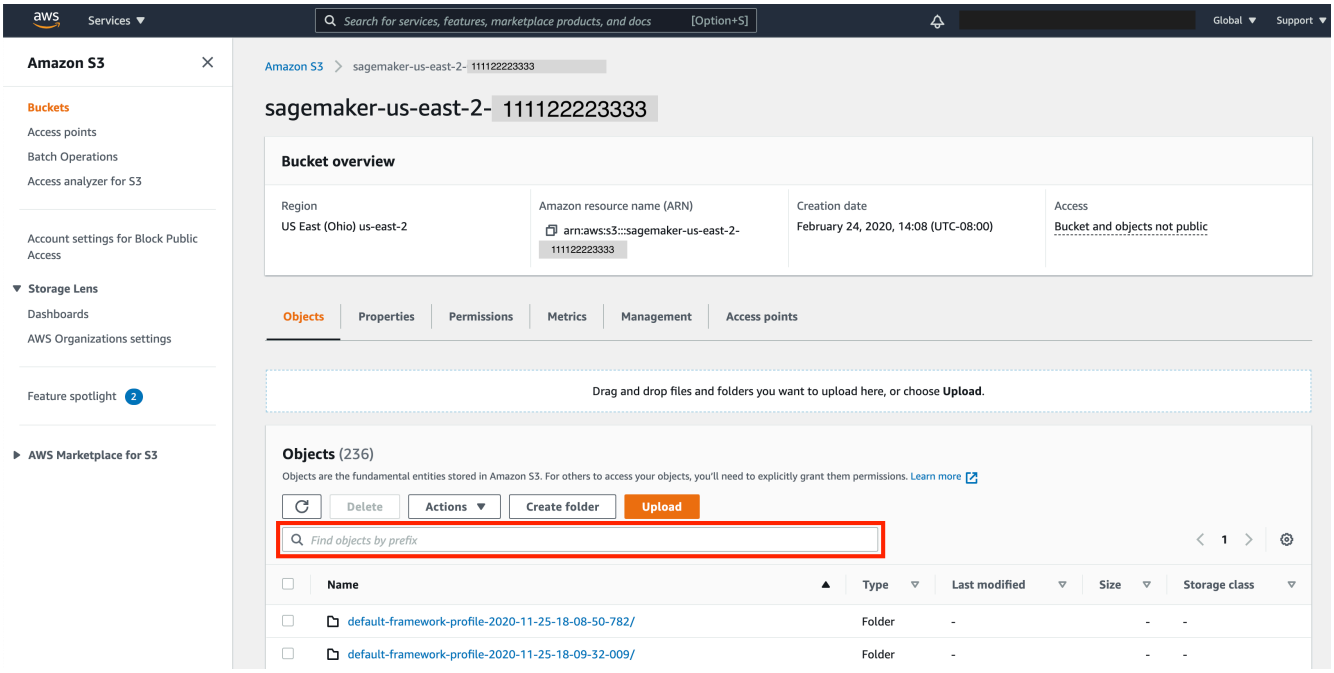
7. Buka`profiler-report.ipynb`file untuk mengeksplorasi bagaimana laporan dihasilkan. Anda juga dapat menyesuaikan dan memperpanjang laporan profil menggunakan file notebook Jupyter.

Download using Amazon S3 Console

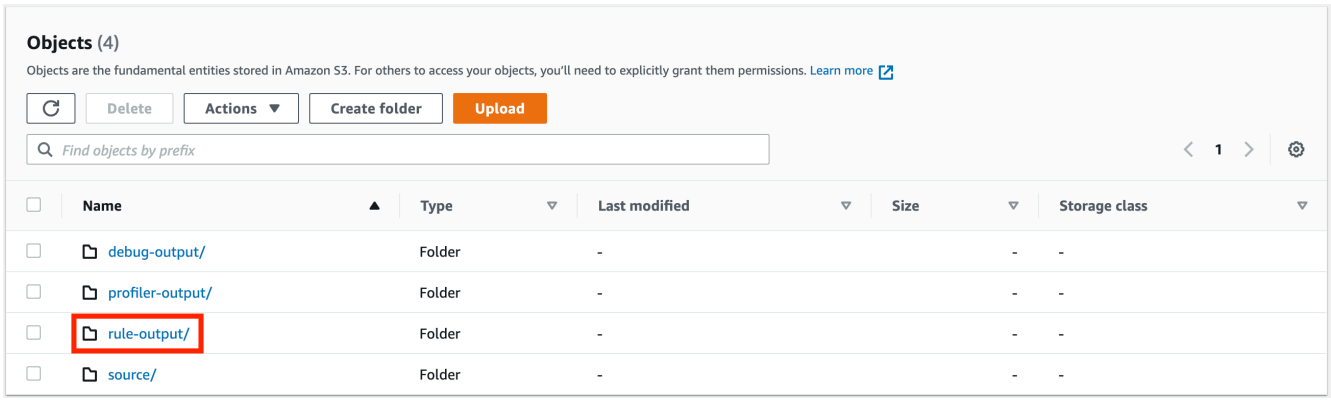
1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Cari ember dasar S3. Misalnya, jika Anda belum menentukan nama pekerjaan dasar apa pun, nama bucket S3 dasar harus dalam format berikut:`sagemaker-<region>-111122223333`. Cari ember S3 dasar melaluiTemukan ember dengan namalapangan.



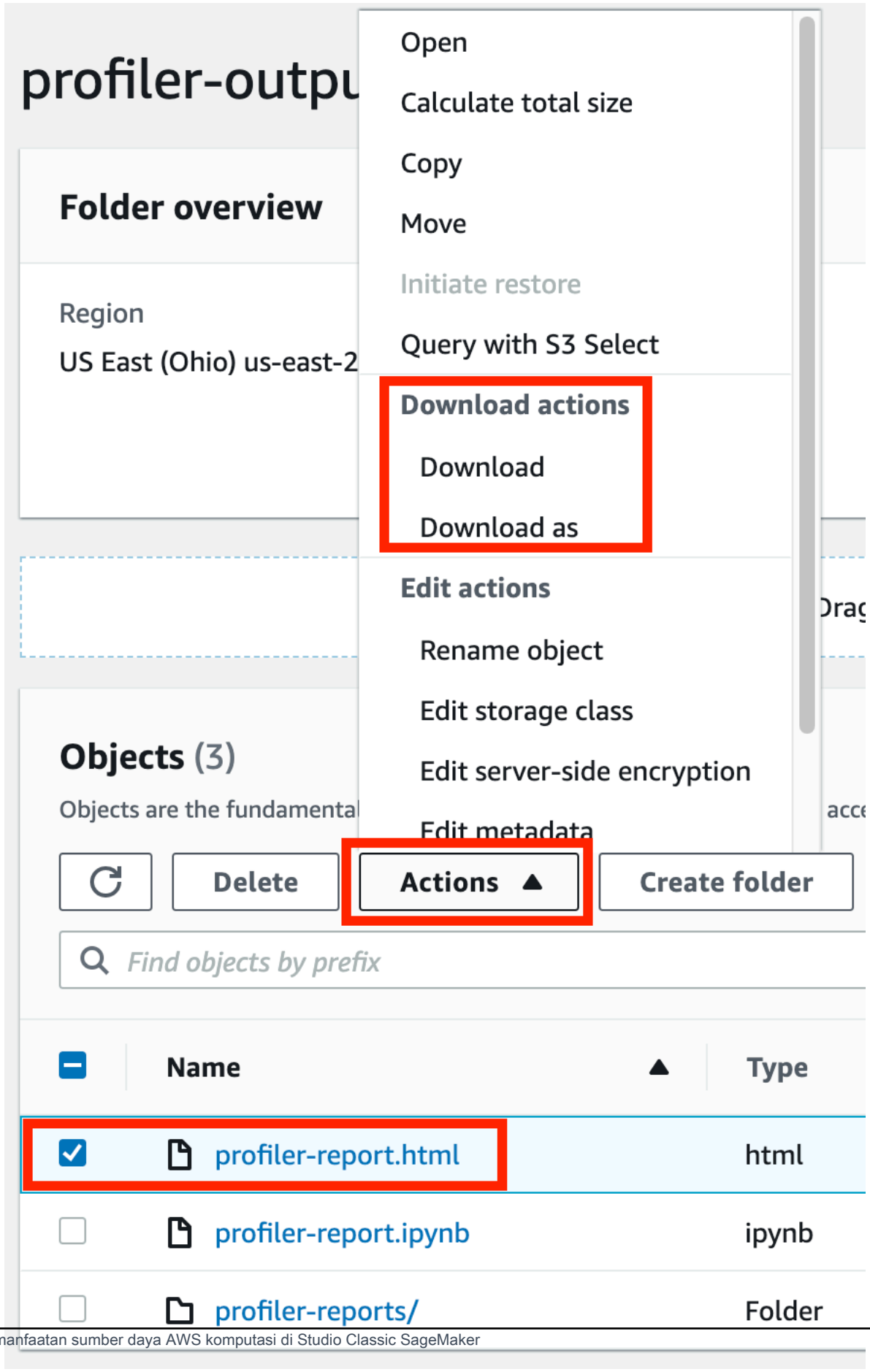
3. Di bucket S3 dasar, cari nama pekerjaan pelatihan dengan menentukan awalan nama pekerjaan Anda ke dalamTemukan objek dengan awalanbidang masukan. Pilih nama pekerjaan pelatihan.



4. Dalam bucket S3 pekerjaan pelatihan, harus ada tiga subfolder untuk data pelatihan yang dikumpulkan oleh Debugger:debug-output/,profil-output/, dankeluaran-aturan/. Pilihkeluaran-aturan/.



5. Dikeluaran-aturan/folder, pilihProfilerReport-1234567890, dan pilihprofil-output/folder. Theprofil-output/folder berisiprofiler-report.html(laporan pembuatan profil yang dibuat secara otomatis dalam html),profiler-report.ipynb(buku catatan Jupyter dengan skrip yang digunakan untuk menghasilkan laporan), dan aprofiler-laporan/folder (berisi analisis aturan file JSON yang digunakan sebagai komponen laporan).
6. Pilihprofiler-report.htmlberkas, pilihTindakan, danUnduh.


profiler-outpu
Folder overview
Region
US East (Ohio) us-east-2
Open
Calculate total size
Copy
Move
Initiate restore
Query with S3 Select
Download actions
Download
Download as
Edit actions
Rename object
Edit storage class
Edit server-side encryption
Edit metadata
Objects (3)
Objects are the fundamental
Delete
Actions
Create folder
Find objects by prefix
Name
Type
profiler-report.html
html
profiler-report.ipynb
ipynb
profiler-reports/
Folder

7. Buka yang diunduhprofiler-report.htmlfile di browser web.

> **Note**
>
> Jika Anda memulai tugas latihan tanpa mengonfigurasi parameter khusus Debugger, Debugger akan menghasilkan laporan hanya berdasarkan aturan pemantauan sistem karena parameter Debugger tidak dikonfigurasi untuk menyimpan metrik kerangka kerja. Untuk mengaktifkan profil metrik kerangka kerja dan menerima laporan profil Debugger yang diperluas, konfigurasikan`profiler_config`parameter saat membangun atau memperbarui SageMaker penaksir.
> Untuk mempelajari cara mengkonfigurasi`profiler_config`parameter sebelum memulai pekerjaan pelatihan, lihatKonfigurasikan untuk pembuatan profil kerangka kerja.
> Untuk memperbarui pekerjaan pelatihan saat ini dan mengaktifkan profil metrik kerangka kerja, lihatPerbarui Konfigurasi Profil Kerangka Debugger.

Panduan laporan pembuatan profil debugger

Bagian ini memandu Anda melalui bagian laporan profiling debugger. Laporan pembuatan profil dibuat berdasarkan aturan bawaan untuk pemantauan dan pembuatan profil. Laporan menunjukkan plot hasil hanya untuk aturan yang menemukan masalah.

> **Important**
>
> Dalam laporan tersebut, plot dan dan rekomendasi disediakan untuk tujuan informasi dan tidak definitif. Anda bertanggung jawab untuk membuat penilaian independen Anda sendiri atas informasi tersebut.

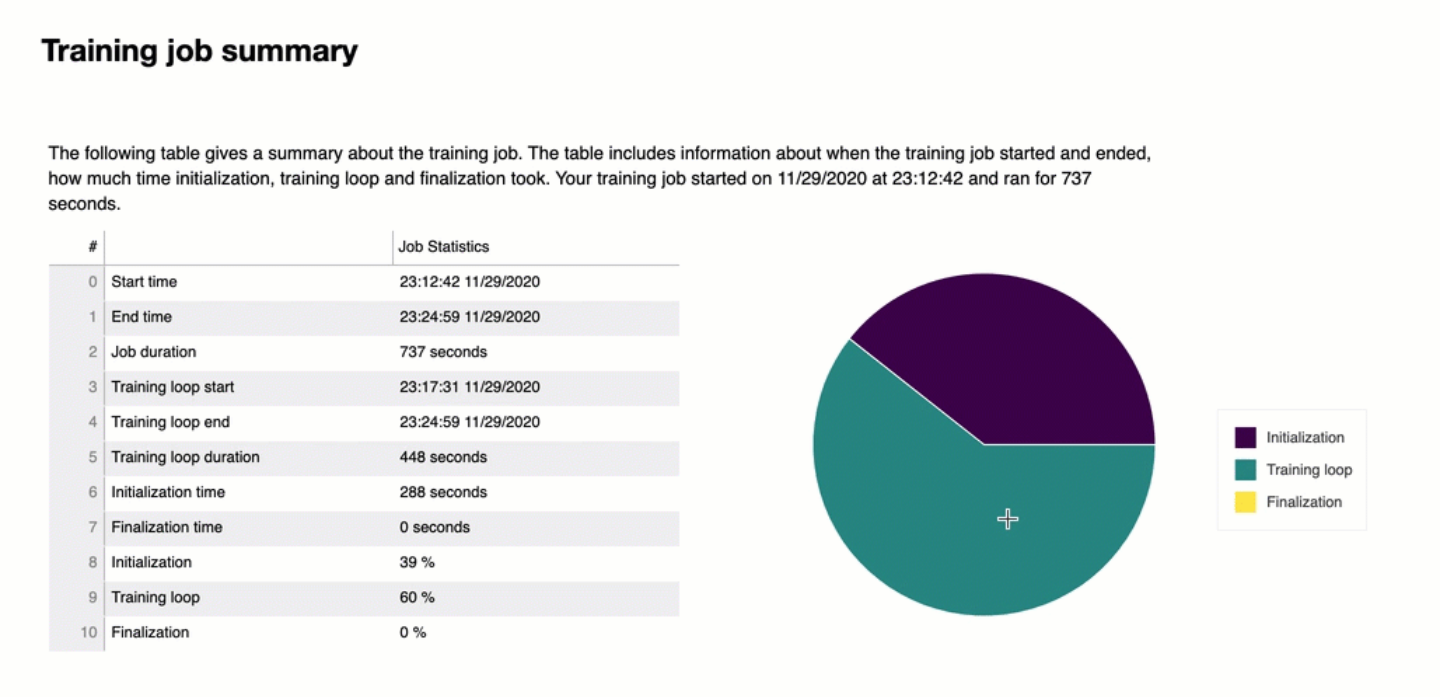
Topik

- Ringkasan pekerjaan pelatihan
- Statistik penggunaan sistem
- Ringkasan metrik kerangka kerja
- Ringkasan aturan
- Menganalisis loop pelatihan — durasi langkah
- Analisis pemanfaatan GPU

- Ukuran batch
- Kemacetan CPU
- Kemacetan I/O
- Load balancing dalam pelatihan multi-GPU
- Analisis memori GPU

Ringkasan pekerjaan pelatihan

Di awal laporan, Debugger memberikan ringkasan pekerjaan pelatihan Anda. Di bagian ini, Anda dapat meninjau durasi waktu dan stempel waktu pada fase pelatihan yang berbeda.



Training job summary

The following table gives a summary about the training job. The table includes information about when the training job started and ended, how much time initialization, training loop and finalization took. Your training job started on 11/29/2020 at 23:12:42 and ran for 737 seconds.

#		Job Statistics
0	Start time	23:12:42 11/29/2020
1	End time	23:24:59 11/29/2020
2	Job duration	737 seconds
3	Training loop start	23:17:31 11/29/2020
4	Training loop end	23:24:59 11/29/2020
5	Training loop duration	448 seconds
6	Initialization time	288 seconds
7	Finalization time	0 seconds
8	Initialization	39 %
9	Training loop	60 %
10	Finalization	0 %

Tabel ringkasan berisi informasi berikut:

- start_time— Waktu yang tepat ketika pekerjaan pelatihan dimulai.
- waktu_akhir— Waktu yang tepat ketika pekerjaan pelatihan selesai.
- job_duration_id_seconds— Total waktu pelatihan daristart_timekewaktu_akhir.
- pelatihan_loop_start— Waktu yang tepat ketika langkah pertama dari zaman pertama telah dimulai.
- training_loop_end— Waktu yang tepat ketika langkah terakhir dari zaman terakhir telah selesai.
- training_loop_duration_in_seconds— Total waktu antara waktu mulai loop pelatihan dan waktu akhir loop pelatihan.

- inisialisasi_in_seconds— Waktu yang dihabiskan untuk menginisialisasi pekerjaan pelatihan. Fase inisialisasi mencakup periode daristart_timekepelatihan_loop_startwaktu. Waktu inisialisasi dihabiskan untuk menyusun skrip pelatihan, memulai skrip pelatihan, membuat dan menginisialisasi model, memulai instans EC2, dan mengunduh data pelatihan.
- finalisasi_in_seconds— Waktu yang dihabiskan untuk menyelesaikan pekerjaan pelatihan, seperti menyelesaikan pelatihan model, memperbarui artefak model, dan menutup instans EC2. Fase finalisasi mencakup periode daritraining_loop_endwaktu untukwaktu_akhir.
- inisialisasi (%)Persentase waktu yang dihabiskan untukinisialisasiatas totaljob_duration_id_seconds.
- lingkaran pelatihan (%)Persentase waktu yang dihabiskan untuklingkaran pelatihanatas totaljob_duration_id_seconds.
- finalisasi (%)Persentase waktu yang dihabiskan untukfinalisasiatas totaljob_duration_id_seconds.

Statistik penggunaan sistem

Di bagian ini, Anda dapat melihat ikhtisar statistik pemanfaatan sistem.

System usage statistics

The 95th quantile of the total GPU utilization on node algo-2 is 74%. GPUs on node algo-2 are well utilized

The following table shows usage statistics per worker node such as total CPU and GPU utilization, total CPU and memory footprint. The table also include total IO wait time and total sent/received bytes. The table shows min and max values as well as p99, p90 and p50 percentiles.

#	*node*	metric	unit	max	p99	p95	p50	min
0	algo-1	Network	bytes	218817581.57	168.02	0	0	0
10	algo-1	I/O	percentage	13.2653125	5.5928312500000	0.1955937499999	0	0
8	algo-1	GPU memory	percentage	32.25	26.25	21	0	0
2	algo-1	GPU	percentage	75	74.5	74.25	0	0
6	algo-1	CPU memory	percentage	5.05	5.01	4.98	2.17	0.55
4	algo-1	CPU	percentage	32.955625	22.629131250000	17.034	3.7024999999999	0
1	algo-2	Network	bytes	4135.24	0	0	0	0
11	algo-2	I/O	percentage	20.1875	8.1552500000000	1.7478124999999	0	0
9	algo-2	GPU memory	percentage	38	31.75	21.75	0	0
3	algo-2	GPU	percentage	75	74.5	74.25	0	0
7	algo-2	CPU memory	percentage	5.05	5.02	4.99	2.17	0.55
5	algo-2	CPU	percentage	35.004374999999	25.699968750000	18.334296875	3.77828125	0

Laporan profiling Debugger mencakup informasi berikut:

- simpul— Daftar nama node. Jika menggunakan pelatihan terdistribusi pada multi node (beberapa instance EC2), nama node dalam format`algo-n`.
- metris— Metrik sistem yang dikumpulkan oleh Debugger: CPU, GPU, memori CPU, memori GPU, I/O, dan metrik Jaringan.
- unit— Unit metrik sistem.
- maks— Nilai maksimum setiap metrik sistem.
- p99Persentil ke-99 dari setiap pemanfaatan sistem.
- p95- Persentil ke-95 dari setiap pemanfaatan sistem.
- p50— Persentil ke-50 (median) dari setiap pemanfaatan sistem.
- min— Nilai minimum setiap metrik sistem.

Ringkasan metrik kerangka kerja

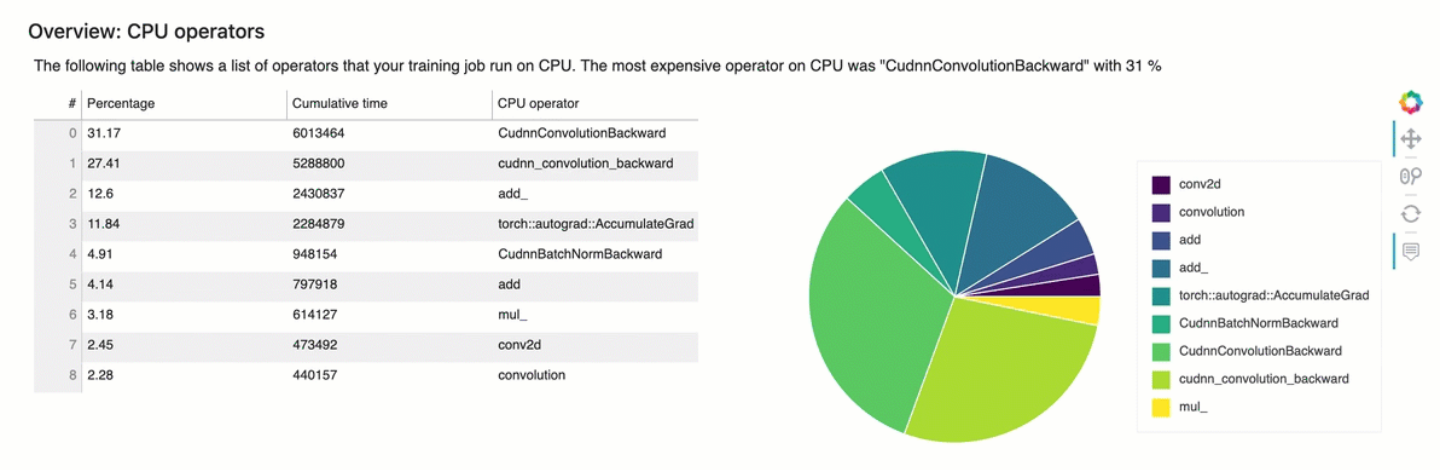
Pada bagian ini, diagram lingkaran berikut menunjukkan rincian operasi kerangka kerja pada CPU dan GPU.

Setiap diagram lingkaran menganalisis metrik kerangka kerja yang dikumpulkan dalam berbagai aspek sebagai berikut:

- Rasio antara fase TRAIN/EVAL dan lainnya— Menunjukkan rasio antara durasi waktu yang dihabiskan untuk fase pelatihan yang berbeda.
- Rasio antara pass maju dan mundur— Menunjukkan rasio antara durasi waktu yang dihabiskan untuk pass maju dan mundur dalam loop pelatihan.
- Rasio antara operator CPU/GPU— Menunjukkan rasio antara waktu yang dihabiskan pada operator yang berjalan pada CPU atau GPU, seperti operator convolutional.
- Metrik umum dicatat dalam kerangka— Menunjukkan rasio antara waktu yang dihabiskan untuk metrik kerangka kerja utama, seperti pemuatan data, pass maju dan mundur.

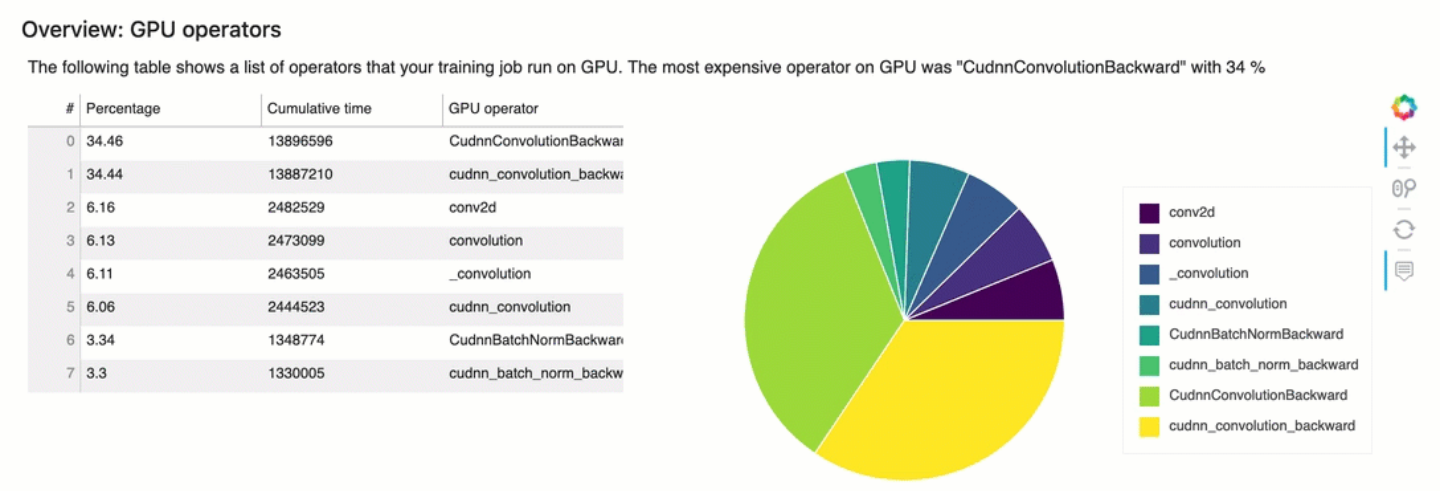
Ikhtisar: Operator CPU

Bagian ini memberikan informasi tentang operator CPU secara rinci. Tabel menunjukkan persentase waktu dan waktu kumulatif absolut yang dihabiskan untuk operator CPU yang paling sering disebut.



Ikhtisar: operator GPU

Bagian ini memberikan informasi tentang operator GPU secara rinci. Tabel menunjukkan persentase waktu dan waktu kumulatif absolut yang dihabiskan untuk operator GPU yang paling sering disebut.



Ringkasan aturan

Pada bagian ini, Debugger menggabungkan semua hasil evaluasi aturan, analisis, deskripsi aturan, dan saran.

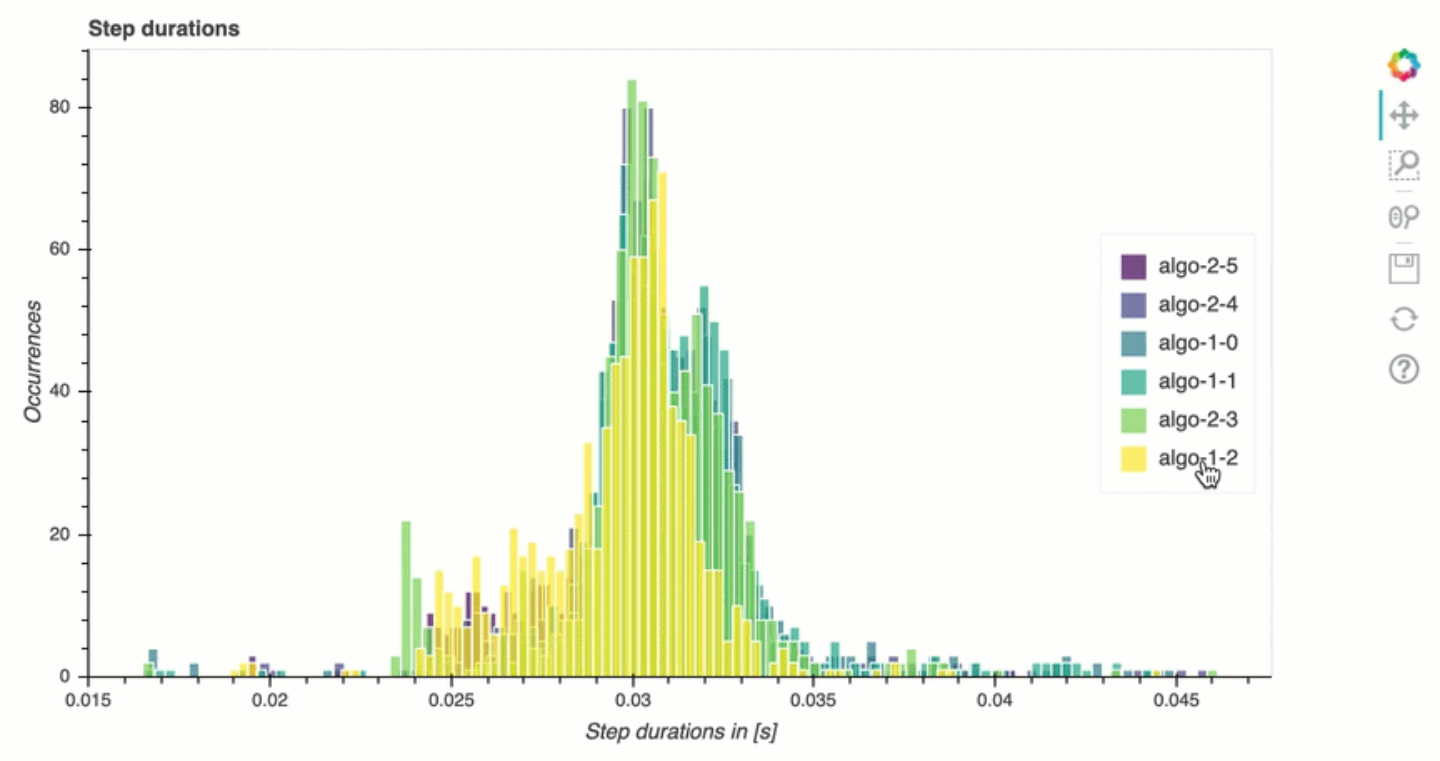
Rules summary

The following table shows a summary of the executed profiler rules. The table is sorted by the rules that triggered most frequently. In your training job this was the case for rule LoadBalancing. It has processed 5467 datapoints and triggered 263 times.

	Description	Recommendation	Number of times rule triggered	Number of datapoints	Rule parameters
LoadBalancing	Detect issues in workload balancing between multiple GPUs. Workload imbalance can for instance occur in data parallel training when gradients are accumulated on primary GPU so this GPU will be overused with regards to other GPUs limiting the effect of parallelization.	Choose different distributed training strategy or different distributed training framework	263	5467	threshold:0.2 patience:1000
LowGPUUtilization	Checks if GPU utilization is low or suffers from fluctuations. This can happen if there are bottlenecks, many blocking calls due to synchronizations or batch size too small.	Check for bottlenecks, minimize blocking calls, change distributed training strategy, increase batch-size.	244	5467	threshold_p95:70 threshold_p5:10 window:500 patience:1000
BatchSize	Checks if GPU is under-utilized because of the batch size being too small. To detect this the rule analyzes the average GPU memory footprint, CPU and GPU utilization.	Run on a smaller instance type or increase batch size	211	5466	cpu_threshold_p95:70 gpu_threshold_p95:70 gpu_memory_threshold_p95:70 patience:1000 window:500
GPUMemoryIncrease	If model and/or batch size is too large then training will run out of memory and crash.	Choose a larger instance type with more memory (if it is not a memory leak) or apply model parallelism (Rubik)	25	5467	increase:5 patience:1000 window:10
CPUBottleneck	Checks if CPU usage is high but GPU usage is low at the same time, it may indicate a CPU bottleneck where GPU is waiting for data to arrive from CPU. The rule triggers if number of CPU bottlenecks exceeds a predefined threshold.	CPU bottlenecks can happen when data preprocessing is very compute intensive. You should consider increasing the number of data-loader processes or apply pre-fetching.	18	10938	threshold:50 cpu_threshold:90 gpu_threshold:10 patience:1000
IOBottleneck	If IO wait time is high but at the same time GPU usage is low, it may indicate an IO bottleneck where GPU is waiting for data to arrive from disk. The rule triggers if number of IO bottlenecks exceeds a predefined threshold.	Pre-fetch data or choose different file formats such as binary formats which improves read performance.	0	10938	threshold:50 io_threshold:50 gpu_threshold:10 patience:1000
StepOutlier	Detect outliers in step duration. Time for forward and backward pass should be roughly the same throughout the training. If there are significant outliers it would indicate an issue due to a system stall or a bottleneck.	Check for bottlenecks	0	4803	threshold:3 mode:None n_outliers:10 stddev:3
MaxInitializationTime	Checks if the training initialization is taking too much time. The rule waits until first step is available. This can happen if you are running in File mode and a lot of data needs to be downloaded from Amazon S3.	Switch from File to Pipe mode	0	4803	threshold:20

Menganalisis loop pelatihan — durasi langkah

Di bagian ini, Anda dapat menemukan statistik rinci durasi langkah pada setiap inti GPU dari setiap node. Debugger mengevaluasi nilai rata-rata, maksimum, p99, p95, p50, dan minimum durasi langkah, dan mengevaluasi pencilan langkah. Histogram berikut menunjukkan durasi langkah yang ditangkap pada node pekerja dan GPU yang berbeda. Anda dapat mengaktifkan atau menonaktifkan histogram setiap pekerja dengan memilih legenda di sisi kanan. Anda dapat memeriksa apakah ada GPU tertentu yang menyebabkan pencilan durasi langkah.



Analisis pemanfaatan GPU

Bagian ini menunjukkan statistik rinci tentang pemanfaatan inti GPU berdasarkan aturan LowGPUUtilization. Ini juga merangkum statistik pemanfaatan GPU, mean, p95, dan p5 untuk menentukan apakah pekerjaan pelatihan kurang memanfaatkan GPU.

Ukuran batch

Bagian ini menunjukkan statistik terperinci dari total pemanfaatan CPU, pemanfaatan GPU individu, dan jejak memori GPU. The BatchSize aturan menentukan apakah Anda perlu mengubah ukuran batch untuk memanfaatkan GPU dengan lebih baik. Anda dapat memeriksa apakah ukuran batch terlalu kecil sehingga kurang dimanfaatkan atau terlalu besar menyebabkan pemanfaatan berlebihan dan masalah memori. Dalam plot, kotak menunjukkan rentang persentil p25 dan p75 (masing-masing diisi dengan ungu tua dan kuning cerah) dari median (p50), dan bilah kesalahan menunjukkan persentil ke-5 untuk batas bawah dan persentil ke-95 untuk batas atas.

Kemacetan CPU

Di bagian ini, Anda dapat menelusuri kemacetan CPU yang terdeteksi oleh aturan CPUBottleNeck dari pekerjaan pelatihan Anda. Aturan memeriksa apakah pemanfaatan CPU di atas`cpu_threshold`(90% secara default) dan juga jika pemanfaatan GPU di bawah`gpu_threshold`(10% secara default).

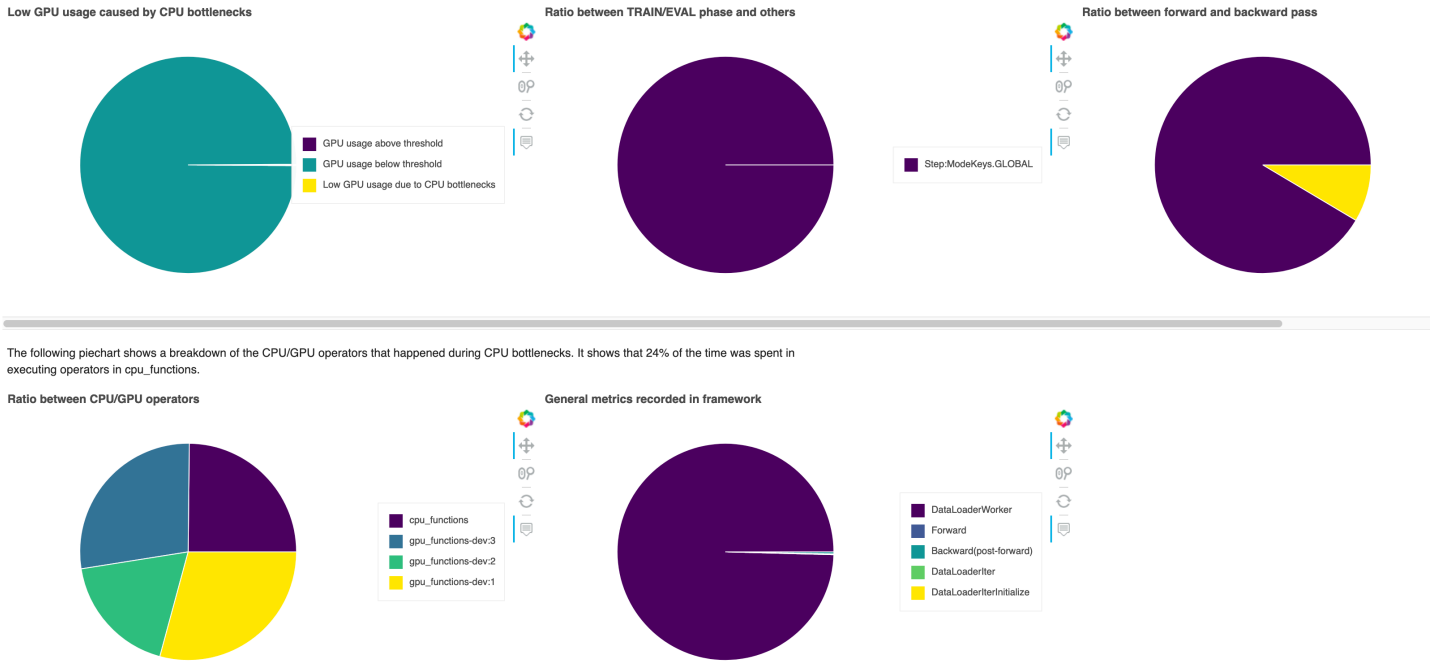


Diagram lingkaran menunjukkan informasi berikut:

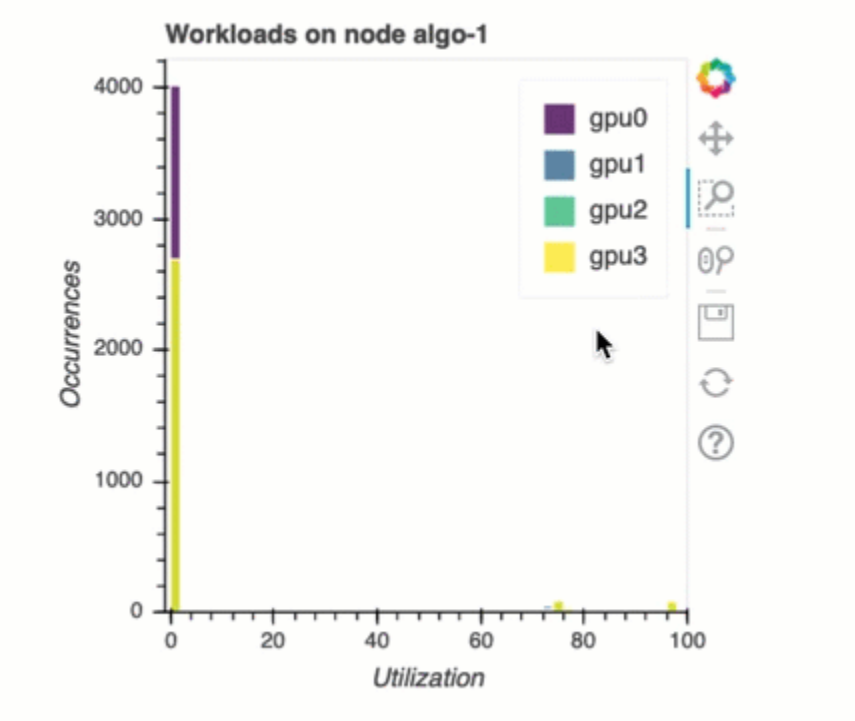
- Penggunaan GPU rendah yang disebabkan oleh kemacetan CPU— Menunjukkan rasio titik data antara yang dengan pemanfaatan GPU di atas dan di bawah ambang batas dan yang sesuai dengan kriteria bottleneck CPU.
- Rasio antara fase TRAIN/EVAL dan lainnya— Menunjukkan rasio antara durasi waktu yang dihabiskan untuk fase pelatihan yang berbeda.
- Rasio antara pass maju dan mundur— Menunjukkan rasio antara durasi waktu yang dihabiskan untuk pass maju dan mundur dalam loop pelatihan.
- Rasio antara operator CPU/GPU— Menunjukkan rasio antara durasi waktu yang dihabiskan untuk GPU dan CPU oleh operator Python, seperti proses pemuat data dan operator pass maju dan mundur.
- Metrik umum dicatat dalam kerangka— Menunjukkan metrik kerangka kerja utama dan rasio antara durasi waktu yang dihabiskan untuk metrik.

Kemacetan I/O

Di bagian ini, Anda dapat menemukan ringkasan kemacetan I/O. Aturan mengevaluasi waktu tunggu I/O dan tingkat pemanfaatan GPU dan monitor jika waktu yang dihabiskan untuk permintaan I/O melebihi persentase ambang batas dari total waktu pelatihan. Ini mungkin menunjukkan kemacetan I/O di mana GPU menunggu data tiba dari penyimpanan.

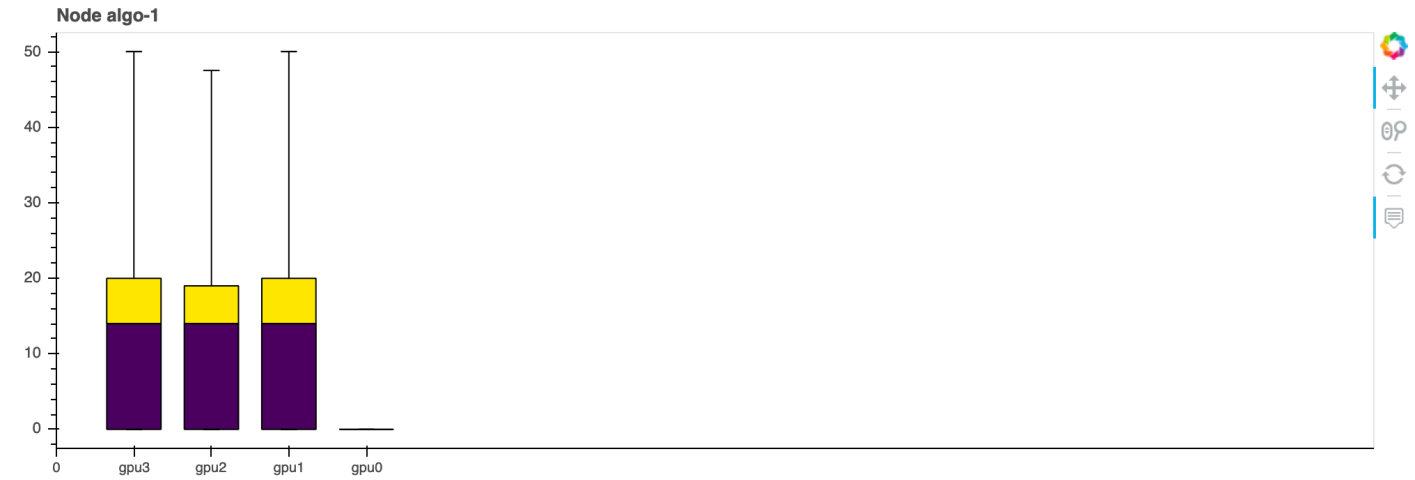
Load balancing dalam pelatihan multi-GPU

SAYADi bagian ini, Anda dapat mengidentifikasi masalah penyeimbangan beban kerja di seluruh GPU.



Analisis memori GPU

Di bagian ini, Anda dapat menganalisis pemanfaatan memori GPU yang dikumpulkan oleh GPUMemoryIncrease aturan. Dalam plot, kotak menunjukkan rentang persentil p25 dan p75 (masing-masing diisi dengan ungu tua dan kuning cerah) dari median (p50), dan bilah kesalahan menunjukkan persentil ke-5 untuk batas bawah dan persentil ke-95 untuk batas atas.



Menganalisis data menggunakan pustaka klien Python Debugger

Saat tugas pelatihan Anda berjalan atau setelah selesai, Anda dapat mengakses data pelatihan yang dikumpulkan oleh Debugger menggunakan Amazon SageMaker Python SDK dan pustaka klien SMDebug. Pustaka klien Python Debugger menyediakan alat analisis dan visualisasi yang memungkinkan Anda menelusuri data pekerjaan pelatihan Anda.

Untuk menginstal perpustakaan dan menggunakan alat analisisnya (dalam JupyterLab notebook atau kernel IPython)

```
! pip install -U smdebug
```

Topik berikut memandu Anda melalui cara menggunakan alat Python Debugger untuk memvisualisasikan dan menganalisis data pelatihan yang dikumpulkan oleh Debugger.

Menganalisis metrik sistem dan kerangka kerja

- Akses data profil
- Plot metrik sistem dan data metrik kerangka kerja
- Akses data profil menggunakan alat parsing data panda
- Mengakses data statistik profil Python
- Gabungkan garis waktu beberapa file jejak profil
- Membuat profil pemuat data

Akses data profil

`TrainingJob`Kelas SMDebug membaca data dari bucket S3 tempat metrik sistem dan kerangka kerja disimpan.

Untuk mengatur **`TrainingJob`** objek dan mengambil file acara profiling dari pekerjaan pelatihan

```
from smdebug.profiler.analysis.notebook_utils.training_job import TrainingJob
tj = TrainingJob(training_job_name, region)
```

> **Tip**
>
> Anda perlu menentukan `training_job_name` dan `region` parameter untuk masuk ke pekerjaan pelatihan. Ada dua cara untuk menentukan informasi pekerjaan pelatihan:

- Gunakan SageMaker Python SDK saat estimator masih melekat pada pekerjaan pelatihan.

```
import sagemaker
training_job_name=estimator.latest_training_job.job_name
region=sagemaker.Session().boto_region_name
```

- Lulus string secara langsung.

```
training_job_name="your-training-job-name-YYYY-MM-DD-HH-MM-SS-SSS"
region="us-west-2"
```

> **Note**
>
> Secara default, SageMaker Debugger mengumpulkan metrik sistem untuk memantau pemanfaatan sumber daya perangkat keras dan kemacetan sistem. Menjalankan fungsi berikut, Anda mungkin menerima pesan kesalahan terkait tidak tersedianya metrik kerangka kerja. Untuk mengambil data profil kerangka kerja dan mendapatkan wawasan tentang operasi kerangka kerja, Anda harus mengaktifkan pembuatan profil kerangka kerja.
>
> - Jika Anda menggunakan SageMaker Python SDK untuk memanipulasi permintaan pekerjaan pelatihan Anda, teruskan `framework_profile_params` ke `profiler_config` argumen estimator Anda. Untuk mempelajari lebih lanjut, lihat Mengkonfigurasi Profil Kerangka SageMaker Debugger.
> - Jika Anda menggunakan Studio Classic, aktifkan pembuatan profil menggunakan tombol toggle Profiling di dasbor wawasan Debugger. Untuk mempelajari lebih lanjut, lihat Pengontrol Dasbor Wawasan SageMaker Debugger.

Untuk mengambil deskripsi deskripsi pekerjaan pelatihan dan URI bucket S3 tempat data metrik disimpan

```
tj.describe_training_job()
tj.get_config_and_profiler_s3_output_path()
```

Untuk memeriksa apakah metrik sistem dan kerangka kerja tersedia dari URI S3

```
tj.wait_for_sys_profiling_data_to_be_available()
```

```
tj.wait_for_framework_profiling_data_to_be_available()
```

Untuk membuat objek pembaca sistem dan kerangka kerja setelah data metrik tersedia

```
system_metrics_reader = tj.get_systems_metrics_reader()
framework_metrics_reader = tj.get_framework_metrics_reader()
```

Untuk menyegarkan dan mengambil file acara pelatihan terbaru

Objek pembaca memiliki metode yang diperluas`refresh_event_file_list()`,, untuk mengambil file acara pelatihan terbaru.

```
system_metrics_reader.refresh_event_file_list()
framework_metrics_reader.refresh_event_file_list()
```

Plot metrik sistem dan data metrik kerangka kerja

Anda dapat menggunakan objek metrik sistem dan algoritma untuk kelas visualisasi berikut untuk memplot grafik garis waktu dan histogram.

> **Note**
>
> Untuk memvisualisasikan data dengan metrik yang dipersempit dalam metode plot objek visualisasi berikut, tentukan dan parameternya. `select_dimensions` `select_events` Misalnya, jika Anda menentukan`select_dimensions=["GPU"]`, metode plot memfilter metrik yang menyertakan kata kunci “GPU”. Jika Anda menentukan`select_events=["total"]`, metode plot memfilter metrik yang menyertakan tag peristiwa “total” di akhir nama metrik. Jika Anda mengaktifkan parameter ini dan memberikan string kata kunci, kelas visualisasi mengembalikan bagan dengan metrik yang difilter.

- `MetricsHistogram`Kelas

```
from smdebug.profiler.analysis.notebook_utils.metrics_histogram import
 MetricsHistogram

metrics_histogram = MetricsHistogram(system_metrics_reader)
metrics_histogram.plot(
    starttime=0,
```

```
    endtime=system_metrics_reader.get_timestamp_of_latest_available_file(),
    select_dimensions=["CPU", "GPU", "I/O"], # optional
    select_events=["total"]                  # optional
)
```

- `StepTimelineChart`Kelas

```
from smdebug.profiler.analysis.notebook_utils.step_timeline_chart import
 StepTimelineChart

view_step_timeline_chart = StepTimelineChart(framework_metrics_reader)
```

- `StepHistogram`Kelas

```
from smdebug.profiler.analysis.notebook_utils.step_histogram import StepHistogram

step_histogram = StepHistogram(framework_metrics_reader)
step_histogram.plot(
    starttime=step_histogram.last_timestamp - 5 * 1000 * 1000,
    endtime=step_histogram.last_timestamp,
    show_workers=True
)
```

- `TimelineCharts`Kelas

```
from smdebug.profiler.analysis.notebook_utils.timeline_charts import TimelineCharts

view_timeline_charts = TimelineCharts(
    system_metrics_reader,
    framework_metrics_reader,
    select_dimensions=["CPU", "GPU", "I/O"], # optional
    select_events=["total"]                  # optional
)

view_timeline_charts.plot_detailed_profiler_data([700,710])
```

- `Heatmap`Kelas

```
from smdebug.profiler.analysis.notebook_utils.heatmap import Heatmap

view_heatmap = Heatmap(
    system_metrics_reader,
    framework_metrics_reader,
```

```
    select_dimensions=["CPU", "GPU", "I/O"], # optional
    select_events=["total"],                 # optional
    plot_height=450
)
```

Akses data profil menggunakan alat parsing data panda

`PandasFrame`Kelas berikut menyediakan alat untuk mengonversi data profil yang dikumpulkan ke bingkai data Pandas.

```
from smdebug.profiler.analysis.utils.profiler_data_to_pandas import PandasFrame
```

`PandasFrame`Kelas mengambil jalur keluaran bucket S3 `tj` objek, dan metodenya `get_all_system_metrics()` `get_all_framework_metrics()` mengembalikan metrik sistem dan metrik kerangka kerja dalam format data Pandas.

```
pf = PandasFrame(tj.profiler_s3_output_path)
system_metrics_df = pf.get_all_system_metrics()
framework_metrics_df = pf.get_all_framework_metrics(
    selected_framework_metrics=[
        'Step:ModeKeys.TRAIN',
        'Step:ModeKeys.GLOBAL'
    ]
)
```

Mengakses data statistik profil Python

Profil Python menyediakan metrik kerangka kerja yang terkait dengan fungsi dan operator Python dalam skrip pelatihan Anda dan kerangka kerja pembelajaran mendalam. SageMaker

Mode Pelatihan dan Fase untuk Profil Python

Untuk membuat profil interval tertentu selama pelatihan ke statistik partisi untuk masing-masing interval ini, Debugger menyediakan alat untuk mengatur mode dan fase.

Untuk mode pelatihan, gunakan `PythonProfileModes` kelas berikut:

```
from smdebug.profiler.python_profile_utils import PythonProfileModes
```

Kelas ini menyediakan opsi berikut:

- `PythonProfileModes.TRAIN`— Gunakan jika Anda ingin membuat profil langkah-langkah target dalam fase pelatihan. Opsi mode ini hanya tersedia untuk TensorFlow.
- `PythonProfileModes.EVAL`— Gunakan jika Anda ingin membuat profil langkah-langkah target dalam fase evaluasi. Opsi mode ini hanya tersedia untuk TensorFlow.
- `PythonProfileModes.PREDICT`— Gunakan jika Anda ingin membuat profil langkah target dalam fase prediksi. Opsi mode ini hanya tersedia untuk TensorFlow.
- `PythonProfileModes.GLOBAL`— Gunakan jika Anda ingin membuat profil langkah target dalam fase global, yang mencakup tiga fase sebelumnya. Opsi mode ini hanya tersedia untuk PyTorch.
- `PythonProfileModes.PRE_STEP_ZERO`— Gunakan jika Anda ingin membuat profil langkah-langkah target dalam tahap inisialisasi sebelum langkah pelatihan pertama dari epoch pertama dimulai. Fase ini mencakup pengajuan pekerjaan awal, mengunggah skrip pelatihan ke instans EC2, menyiapkan instans EC2, dan mengunduh data input. Opsi mode ini tersedia untuk keduanya TensorFlow dan PyTorch.
- `PythonProfileModes.POST_HOOK_CLOSE`— Gunakan jika Anda ingin membuat profil langkah target di tahap finalisasi setelah pekerjaan pelatihan selesai dan kait Debugger ditutup. Fase ini mencakup data profil sementara pekerjaan pelatihan diselesaikan dan diselesaikan. Opsi mode ini tersedia untuk keduanya TensorFlow dan PyTorch.

Untuk fase pelatihan, gunakan `StepPhase` kelas berikut:

```
from smdebug.profiler.analysis.utils.python_profile_analysis_utils import StepPhase
```

Kelas ini menyediakan opsi berikut:

- `StepPhase.START`— Gunakan untuk menentukan titik awal fase inisialisasi.
- `StepPhase.STEP_START`— Gunakan untuk menentukan langkah awal fase pelatihan.
- `StepPhase.FORWARD_PASS_END`— Gunakan untuk menentukan langkah-langkah di mana pass maju berakhir. Opsi ini hanya tersedia untuk PyTorch.
- `StepPhase.STEP_END`— Gunakan untuk menentukan langkah akhir dalam fase pelatihan. Opsi ini hanya tersedia untuk TensorFlow.
- `StepPhase.END`— Gunakan untuk menentukan titik akhir dari fase finalisasi (post-hook-close). Jika hook callback tidak ditutup, profil fase finalisasi tidak terjadi.

Alat Analisis Profil Python

Debugger mendukung profiling Python dengan dua alat profiling:

- CProfile — Profiler python standar. cProfile mengumpulkan metrik kerangka kerja pada waktu CPU untuk setiap fungsi yang dipanggil saat pembuatan profil diaktifkan.
- Pyinstrument - Ini adalah peristiwa profil pengambilan sampel profiler Python overhead rendah setiap milidetik.

Untuk mempelajari lebih lanjut tentang opsi pembuatan profil Python dan apa yang dikumpulkan, lihat. Mulai pekerjaan pelatihan dengan pemantauan sistem default dan pembuatan profil kerangka kerja yang disesuaikan dengan opsi pembuatan profil yang berbeda

Metode berikut dari`PythonProfileAnalysis`,`cProfileAnalysis`, `PyinstrumentAnalysis` kelas disediakan untuk mengambil dan menganalisis data profil Python. Setiap fungsi memuat data terbaru dari URI S3 default.

```
from smdebug.profiler.analysis.python_profile_analysis import PythonProfileAnalysis,
 cProfileAnalysis, PyinstrumentAnalysis
```

Untuk mengatur objek profiling Python untuk analisis, gunakan cProfileAnalysis atau PyinstrumentAnalysis kelas seperti yang ditunjukkan dalam kode contoh berikut. Ini menunjukkan bagaimana mengatur `cProfileAnalysis` objek, dan jika Anda ingin menggunakan`PyinstrumentAnalysis`, ganti nama kelas.

```
python_analysis = cProfileAnalysis(
    local_profile_dir=tf_python_stats_dir,
    s3_path=tj.profiler_s3_output_path
)
```

Metode berikut tersedia untuk `PyinstrumentAnalysis` kelas `cProfileAnalysis` dan untuk mengambil data statistik profil Python:

- `python_analysis.fetch_python_profile_stats_by_time(start_time_since_epoch_in_se end_time_since_epoch_in_secs)`— Mengambil waktu mulai dan waktu akhir, dan mengembalikan statistik fungsi statistik langkah yang waktu mulai atau berakhir tumpang tindih dengan interval yang disediakan.
- `python_analysis.fetch_python_profile_stats_by_step(start_step, end_step, mode, start_phase, end_phase)`— Mengambil langkah awal dan langkah akhir dan

mengembalikan statistik fungsi dari semua statistik langkah yang diprofilkan `step` memenuhi. `start_step <= step < end_step`

- `start_step`dan `end_step` (str) - Tentukan langkah awal dan langkah akhir untuk mengambil data statistik profil Python.
- `mode`(str) — Tentukan mode pekerjaan pelatihan menggunakan kelas `PythonProfileModes` enumerator. Default-nya adalah `PythonProfileModes.TRAIN`. Opsi yang tersedia disediakan di bagian Mode Pelatihan dan Fase untuk Profiling Python.
- `start_phase`(str) - Tentukan fase awal pada langkah target menggunakan kelas `StepPhase` enumerator. Parameter ini memungkinkan pembuatan profil antara fase pelatihan yang berbeda. Default-nya adalah `StepPhase.STEP_START`. Opsi yang tersedia disediakan di bagian Mode Pelatihan dan Fase untuk Profiling Python.
- `end_phase`(str) - Tentukan fase akhir pada langkah target menggunakan kelas `StepPhase` enumerator. Parameter ini mengatur fase akhir pelatihan. Opsi yang tersedia sama dengan yang untuk `start_phase` parameter. Default-nya adalah `StepPhase.STEP_END`. Opsi yang tersedia disediakan di bagian Mode Pelatihan dan Fase untuk Profiling Python.

- `python_analysis.fetch_profile_stats_between_modes(start_mode, end_mode)`— Mengambil statistik dari profil Python antara mode awal dan akhir.
- `python_analysis.fetch_pre_step_zero_profile_stats()`— Mengambil statistik dari profil Python hingga langkah 0.
- `python_analysis.fetch_post_hook_close_profile_stats()`— Mengambil statistik dari profil Python setelah hook ditutup.
- `python_analysis.list_profile_stats()`— DataFrame Mengembalikan statistik profil Python. Setiap baris menyimpan metadata untuk setiap instance profil dan file statistik yang sesuai (satu per langkah).
- `python_analysis.list_available_node_ids()`— Mengembalikan daftar ID node yang tersedia untuk statistik profil Python.

Metode khusus `cProfileAnalysis` kelas:

- `fetch_profile_stats_by_training_phase()`— Mengambil dan menggabungkan statistik profil Python untuk setiap kemungkinan kombinasi mode awal dan akhir. Misalnya, jika fase pelatihan dan validasi dilakukan saat pembuatan profil terperinci diaktifkan, kombinasinya adalah`(PRE_STEP_ZERO, TRAIN)`,, `(TRAIN, TRAIN)` `(TRAIN, EVAL)`(`EVAL, EVAL`), dan. `(EVAL, POST_HOOK_CLOSE)` Semua file statistik dalam masing-masing kombinasi ini digabungkan.

- `fetch_profile_stats_by_job_phase()`— Mengambil dan menggabungkan statistik profil Python berdasarkan fase pekerjaan. Fase pekerjaan adalah `initialization` (pembuatan profil hingga langkah 0), `training_loop` (pelatihan dan validasi), dan `finalization` (pembuatan profil setelah kait ditutup).

Gabungkan garis waktu beberapa file jejak profil

Pustaka klien SMDebug menyediakan analisis profil dan alat visualisasi untuk menggabungkan garis waktu metrik sistem, metrik kerangka kerja, dan data profil Python yang dikumpulkan oleh Debugger.

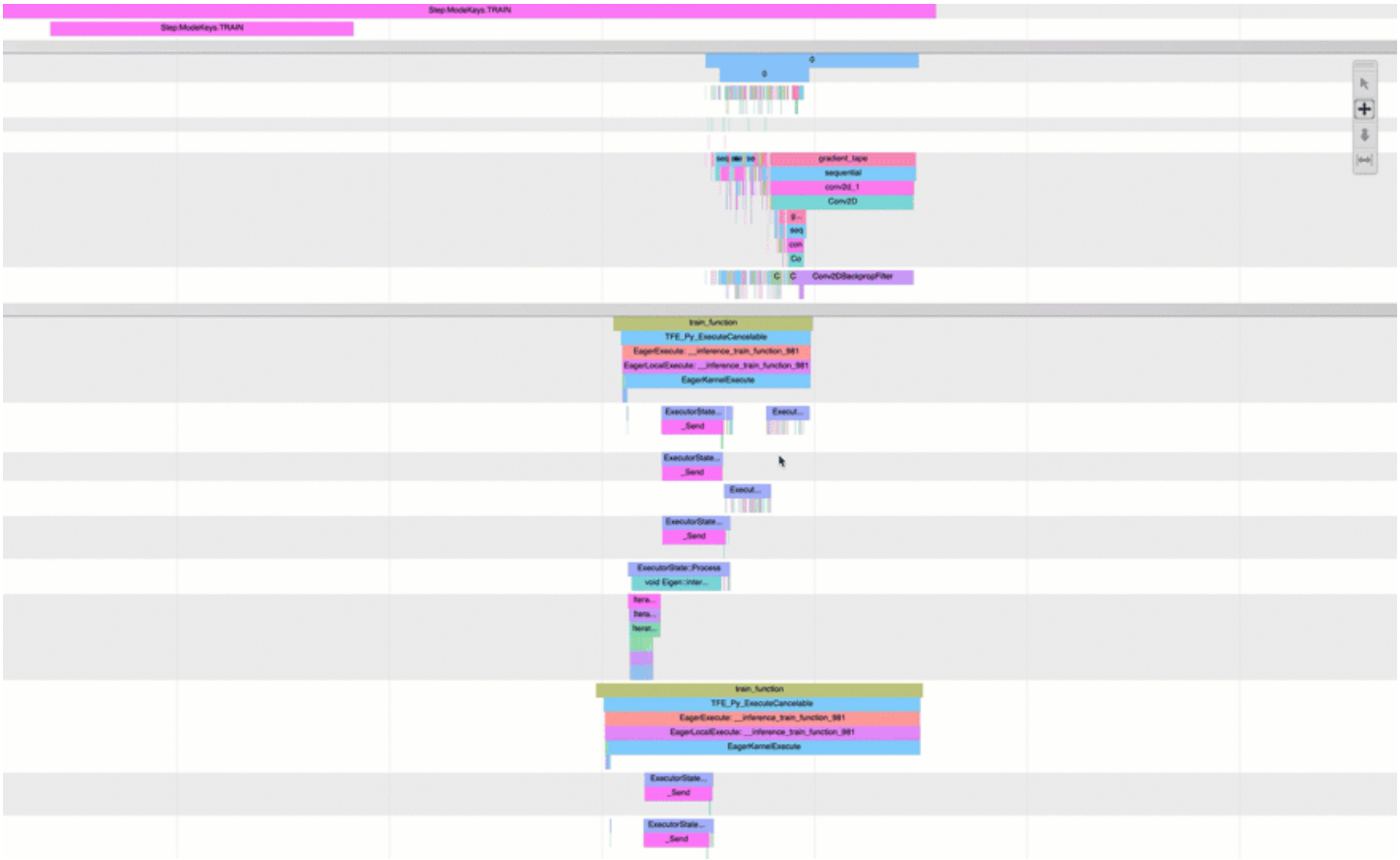
> **Tip**
>
> Sebelum melanjutkan, Anda perlu mengatur TrainingJob objek yang akan digunakan di seluruh contoh di halaman ini. Untuk informasi selengkapnya tentang menyiapkan TrainingJob objek, lihatAkses data profil.

`MergedTimeline`Kelas menyediakan alat untuk mengintegrasikan dan menghubungkan informasi profil yang berbeda dalam satu garis waktu. Setelah Debugger menangkap data profil dan anotasi dari berbagai fase pekerjaan pelatihan, file JSON dari peristiwa pelacakan disimpan dalam direktori default. `tracefolder`

- Untuk anotasi di lapisan Python, file jejak disimpan di. `*pythontimeline.json`
- Untuk anotasi di lapisan TensorFlow C ++, file jejak disimpan di. `*model_timeline.json`
- Profiler Tensorflow menyimpan peristiwa dalam file. `*trace.json.gz`

> **Tip**
>
> Jika Anda ingin membuat daftar semua file jejak JSON, gunakan AWS CLI perintah berikut:
>
> ```
> ! aws s3 ls {tj.profiler_s3_output_path} --recursive | grep '\.json$'
> ```

Seperti yang ditunjukkan pada tangkapan layar animasi berikut, menempatkan dan menyelaraskan peristiwa jejak yang diambil dari sumber profil yang berbeda dalam satu plot dapat memberikan gambaran umum tentang seluruh peristiwa yang terjadi dalam fase yang berbeda dari pekerjaan pelatihan.



> **Tip**
>
> Untuk berinteraksi dengan timeline gabungan pada aplikasi traicing menggunakan keyboard, gunakan `W` tombol untuk memperbesar, tombol untuk menggeser ke kiri, `A` tombol untuk memperkecil, dan `S` tombol untuk beralih ke kanan. `D`

Beberapa file JSON jejak peristiwa dapat digabungkan menjadi satu file JSON peristiwa jejak menggunakan operasi `MergedTimeline` API berikut dan metode kelas dari modul. `smdebug.profiler.analysis.utils.merge_timelines`

```
from smdebug.profiler.analysis.utils.merge_timelines import MergedTimeline

combined_timeline = MergedTimeline(path, file_suffix_filter, output_directory)
combined_timeline.merge_timeline(start, end, unit)
```

Operasi `MergedTimeline` API melewati parameter berikut:

- `path`(str) - Tentukan folder root (`/profiler-output`) yang berisi file jejak profil sistem dan kerangka kerja. Anda dapat menemukan `profiler-output` menggunakan metode kelas SageMaker estimator atau objek. TrainingJob Misalnya, `estimator.latest_job_profiler_artifacts_path()` atau `tj.profiler_s3_output_path`.
- `file_suffix_filter`(daftar) - Tentukan daftar filter akhiran file untuk menggabungkan garis waktu. Filter akhiran yang tersedia adalah `["model_timeline.json", "pythontimeline.json", "trace.json.gz"]`. Jika parameter ini tidak ditentukan secara manual, semua file jejak digabungkan secara default.
- `output_directory`(str) - Tentukan jalur untuk menyimpan file JSON garis waktu gabungan. Defaultnya adalah ke direktori yang ditentukan untuk `path` parameter.

`merge_timeline()`Classmethod melewati parameter berikut untuk menjalankan proses penggabungan:

- `start`(int) - Tentukan waktu mulai (dalam mikrodetik dan dalam format waktu Unix) atau mulai langkah untuk menggabungkan garis waktu.
- end(int) - Tentukan waktu akhir (dalam mikrodetik dan dalam format waktu Unix) atau langkah akhir untuk menggabungkan garis waktu.
- `unit`(str) — Pilih antara `"time"` dan`"step"`. Default-nya adalah `"time"`.

Dengan menggunakan kode contoh berikut, jalankan `merge_timeline()` metode dan unduh file JSON yang digabungkan.

- Gabungkan timeline dengan opsi `"time"` unit. Contoh kode berikut menggabungkan semua file jejak yang tersedia antara waktu mulai Unix (waktu Unix nol mutlak) dan waktu Unix saat ini, yang berarti Anda dapat menggabungkan garis waktu untuk seluruh durasi pelatihan.

```
import time
from smdebug.profiler.analysis.utils.merge_timelines import MergedTimeline
from smdebug.profiler.profiler_constants import CONVERT_TO_MICROSECS

combined_timeline = MergedTimeline(tj.profiler_s3_output_path, output_directory="./")
combined_timeline.merge_timeline(0, int(time.time() * CONVERT_TO_MICROSECS))
```

- Gabungkan timeline dengan opsi `"step"` unit. Kode contoh berikut menggabungkan semua jadwal yang tersedia antara langkah 3 dan langkah 9.

```
from smdebug.profiler.analysis.utils.merge_timelines import MergedTimeline

combined_timeline = MergedTimeline(tj.profiler_s3_output_path, output_directory="./")
combined_timeline.merge_timeline(3, 9, unit="step")
```

Buka aplikasi penelusuran Chrome `chrome://tracing` di browser Chrome, dan buka file JSON. Anda dapat menjelajahi output untuk memplot timeline gabungan.

Membuat profil pemuat data

Pada tahun PyTorch, iterator pemuat data, seperti `SingleProcessingDataLoaderIter` dan`MultiProcessingDataLoaderIter`, dimulai pada awal setiap iterasi melalui kumpulan data. Selama fase inisialisasi, PyTorch mengaktifkan proses pekerja tergantung pada jumlah pekerja yang dikonfigurasi, menetapkan antrian data untuk mengambil data dan utas. `pin_memory`

Untuk menggunakan alat analisis profil pemuat PyTorch data, impor `PT_dataloader_analysis` kelas berikut:

```
from smdebug.profiler.analysis.utils.pytorch_dataloader_analysis import
 PT_dataloader_analysis
```

Lulus data profil yang diambil sebagai objek data bingkai Pandas di bagian: Akses data profil menggunakan alat parsing data panda

```
pt_analysis = PT_dataloader_analysis(pf)
```

Fungsi-fungsi berikut tersedia untuk `pt_analysis` objek:

`S3SystemMetricsReader`Kelas SMDebug membaca metrik sistem dari bucket S3 yang ditentukan ke parameter. `s3_trial_path`

- `pt_analysis.analyze_dataloaderIter_initialization()`

  Analisis menghasilkan median dan durasi maksimum untuk inisialisasi ini. Jika ada outlier, (yaitu durasi lebih besar dari 2 * median), fungsi mencetak waktu mulai dan akhir untuk durasi tersebut. Ini dapat digunakan untuk memeriksa metrik sistem selama interval waktu tersebut.

  Daftar berikut menunjukkan analisis apa yang tersedia dari metode kelas ini:

  - Jenis iterator pemuat data apa yang diinisialisasi.
  - Jumlah pekerja per iterator.
  - Periksa apakah iterator diinisialisasi dengan atau tanpa pin_memory.
  - Berapa kali iterator diinisialisasi selama pelatihan.
- `pt_analysis.analyze_dataloaderWorkers()`

  Daftar berikut menunjukkan analisis apa yang tersedia dari metode kelas ini:

  - Jumlah proses pekerja yang diputar selama seluruh pelatihan.
  - Durasi rata-rata dan maksimum untuk proses pekerja.
  - Waktu mulai dan berakhir untuk proses pekerja yang merupakan outlier.
- `pt_analysis.analyze_dataloader_getnext()`

  Daftar berikut menunjukkan analisis apa yang tersedia dari metode kelas ini:

  - Jumlah GetNext panggilan yang dilakukan selama pelatihan.
  - Median dan durasi maksimum dalam mikrodetik untuk GetNext panggilan.
  - Waktu mulai, Waktu berakhir, durasi, dan id pekerja untuk durasi GetNext panggilan outlier.
- `pt_analysis.analyze_batchtime(start_timestamp, end_timestamp, select_events=[".*"], select_dimensions=[".*"])`

  Debugger mengumpulkan waktu mulai dan berakhir dari semua panggilan. GetNext Anda dapat menemukan jumlah waktu yang dihabiskan oleh skrip pelatihan pada satu kumpulan data. Dalam jendela waktu yang ditentukan, Anda dapat mengidentifikasi panggilan yang tidak secara langsung berkontribusi pada pelatihan. Panggilan ini dapat berasal dari operasi berikut: menghitung akurasi, menambahkan kerugian untuk tujuan debugging atau logging, dan mencetak informasi debugging. Operasi seperti ini dapat dihitung secara intensif atau memakan waktu. Kami dapat mengidentifikasi operasi tersebut dengan menghubungkan profiler Python, metrik sistem, dan metrik kerangka kerja.

  Daftar berikut menunjukkan analisis apa yang tersedia dari metode kelas ini:

  - Waktu profil yang dihabiskan untuk setiap kumpulan data`BatchTime_in_seconds`,, dengan menemukan perbedaan antara waktu mulai GetNext panggilan saat ini dan berikutnya.
  - Temukan outlier di dalam `BatchTime_in_seconds` dan waktu mulai dan berakhir untuk outlier tersebut.
  - Dapatkan metrik sistem dan kerangka kerja selama stempel waktu tersebut`BatchTime_in_seconds`. Ini menunjukkan di mana waktu dihabiskan.

- `pt_analysis.plot_the_window()`

  Plot bagan garis waktu antara stempel waktu awal dan stempel waktu akhir.

Catatan rilis untuk kemampuan pembuatan profil Amazon SageMaker

Lihat catatan rilis berikut untuk melacak pembaruan terbaru untuk kemampuan pembuatan profil Amazon SageMaker.

14 Desember 2023

Pembaruan mata uang

SageMaker Profiler telah menambahkan dukungan untuk TensorFlow v2.13.0.

Melanggar perubahan

Rilis ini melibatkan perubahan besar. Nama paket SageMaker Profiler Python diubah `smppy` dari menjadi. `smprof` Jika Anda telah menggunakan versi paket sebelumnya saat Anda sudah mulai menggunakan SageMaker Framework Container terbaru untuk TensorFlow tercantum di bagian berikut, pastikan bahwa Anda memperbarui nama paket dari `smppy` ke `smprof` dalam pernyataan impor dalam skrip pelatihan Anda.

Migrasi ke AWS Deep Learning Containers

Rilis SageMaker Profiler ini lulus pengujian benchmark dan dimigrasikan ke AWSDeep Learning Containers berikut.

- TensorFlow v2.13.0
- TensorFlow v2.12.0

Jika Anda menggunakan versi sebelumnya dari wadah kerangka kerja yang didukung seperti TensorFlow v2.11.0, paket Profiler SageMaker Python masih tersedia sebagai. `smppy` Jika Anda tidak yakin versi atau nama paket mana yang harus Anda gunakan, ganti pernyataan impor paket SageMaker Profiler dengan cuplikan kode berikut.

```
try:
    import smprof
except ImportError:
```

```
    # backward-compatability for TF 2.11 and PT 1.13.1 images
    import smppy as smprof
```

24 Agustus 2023

Fitur baru

Amazon SageMaker Profiler yang dirilis, kemampuan pembuatan profil dan visualisasi SageMaker untuk menyelami sumber daya komputasi yang disediakan sambil melatih model pembelajaran mendalam dan mendapatkan visibilitas ke detail tingkat operasi. SageMaker Profiler menyediakan modul Python `smppy` () untuk menambahkan anotasi PyTorch di seluruh TensorFlow atau melatih skrip dan mengaktifkan Profiler. SageMaker Anda dapat mengakses modul melalui SageMaker Python SDK dan AWS Deep Learning Containers. Untuk pekerjaan apa pun yang dijalankan dengan modul SageMaker Profiler Python, Anda dapat memuat data profil di SageMaker aplikasi UI Profiler yang menyediakan dasbor ringkasan dan garis waktu terperinci. Untuk mempelajari selengkapnya, lihat Menggunakan Amazon SageMaker Profiler untuk membuat profil aktivitas pada sumber daya AWS komputasi.

Rilis paket SageMaker Profiler Python ini diintegrasikan ke dalam Framework Containers SageMaker PyTorch berikut untuk dan. TensorFlow

- PyTorch v2.0.0
- PyTorch v1.13.1
- TensorFlow v2.12.0
- TensorFlow v2.11.0

Pelatihan terdistribusi di Amazon SageMaker

SageMaker menyediakan perpustakaan pelatihan terdistribusi dan mendukung berbagai opsi pelatihan terdistribusi untuk tugas pembelajaran mendalam seperti visi komputer (CV) dan pemrosesan bahasa alami (NLP). Dengan perpustakaan pelatihan SageMaker terdistribusi, Anda dapat menjalankan data kustom paralel yang sangat terukur dan hemat biaya serta memodelkan pekerjaan pelatihan pembelajaran mendalam paralel. Anda juga dapat menggunakan kerangka kerja dan paket pelatihan terdistribusi lainnya seperti PyTorch DistributedDataParallel (DDP)`torchrun`, MPI (`mpirun`), dan server parameter. Sepanjang dokumentasi, instruksi dan contoh berfokus pada cara mengatur opsi pelatihan terdistribusi untuk tugas pembelajaran mendalam menggunakan SageMaker Python SDK.

> **Tip**
>
> Untuk mempelajari praktik terbaik untuk komputasi terdistribusi pelatihan pembelajaran mesin (ML) dan pekerjaan pemrosesan secara umum, lihatKomputasi terdistribusi dengan praktik SageMaker terbaik.

Sebelum Anda memulai

SageMaker Pelatihan mendukung pelatihan terdistribusi pada satu instans serta beberapa instance, sehingga Anda dapat menjalankan berbagai ukuran pelatihan dalam skala besar. Kami menyarankan Anda untuk menggunakan kelas estimator kerangka kerja seperti PyTorchdan TensorFlowdi SageMaker Python SDK, yang merupakan peluncur pekerjaan pelatihan dengan berbagai opsi pelatihan terdistribusi. Saat Anda membuat objek estimator, objek akan menyiapkan infrastruktur pelatihan terdistribusi, menjalankan `CreateTrainingJob` API di backend, menemukan Wilayah tempat sesi Anda berjalan, dan menarik salah satu wadah pembelajaran AWS mendalam yang telah dibuat sebelumnya yang dikemas dengan sejumlah pustaka termasuk kerangka kerja pembelajaran mendalam, kerangka kerja pelatihan terdistribusi, dan driver EFA. Jika Anda ingin memasang sistem file FSx ke instance pelatihan, Anda harus meneruskan subnet VPC dan ID grup keamanan Anda ke estimator. Sebelum menjalankan pekerjaan pelatihan terdistribusi Anda di SageMaker, baca panduan umum berikut tentang pengaturan infrastruktur dasar.

Zona ketersediaan dan backplane jaringan

Saat menggunakan beberapa instance (juga disebut node), penting untuk memahami jaringan yang menghubungkan instance, bagaimana mereka membaca data pelatihan, dan bagaimana mereka berbagi informasi di antara mereka sendiri. Misalnya, saat Anda menjalankan tugas pelatihan paralel data terdistribusi, sejumlah faktor, seperti komunikasi antara node cluster komputasi untuk menjalankan `AllReduce` operasi dan transfer data antara node dan penyimpanan data di Amazon Simple Storage Service atau Amazon FSx for Lustre, memainkan peran penting untuk mencapai penggunaan sumber daya komputasi yang optimal dan kecepatan pelatihan yang lebih cepat. Untuk mengurangi overhead komunikasi, pastikan Anda mengonfigurasi instance, subnet VPC, dan penyimpanan data di Availability Zone yang sama. Wilayah AWS

Instans GPU dengan jaringan yang lebih cepat dan penyimpanan throughput tinggi

Anda secara teknis dapat menggunakan instance apa pun untuk pelatihan terdistribusi. Untuk kasus di mana Anda perlu menjalankan pekerjaan pelatihan terdistribusi multi-node untuk melatih model

besar, seperti model bahasa besar (LLM) dan model difusi, yang memerlukan pergantian antar simpul yang lebih cepat, kami merekomendasikan instance GPU berkemampuan EFA yang didukung oleh. SageMaker Khususnya, untuk mencapai pekerjaan pelatihan terdistribusi yang paling berkinerja di SageMaker, kami merekomendasikan instans P4d dan P4de yang dilengkapi dengan GPU NVIDIA A100. Ini juga dilengkapi dengan penyimpanan instans lokal latensi rendah throughput tinggi dan jaringan intra-node yang lebih cepat. Untuk penyimpanan data, kami merekomendasikan Amazon FSx for Lustre yang menyediakan throughput tinggi untuk menyimpan kumpulan data pelatihan dan pos pemeriksaan model.

Memulai pelatihan terdistribusi di Amazon SageMaker

Jika Anda sudah terbiasa dengan pelatihan terdistribusi, pilih salah satu opsi berikut yang sesuai dengan strategi atau kerangka kerja pilihan Anda untuk memulai. Jika Anda ingin belajar tentang pelatihan terdistribusi secara umum, lihatthe section called “Konsep pelatihan terdistribusi dasar”.

Perpustakaan pelatihan SageMaker terdistribusi dioptimalkan untuk lingkungan SageMaker pelatihan, membantu menyesuaikan pekerjaan pelatihan terdistribusi Anda SageMaker, dan meningkatkan kecepatan dan throughput pelatihan. Perpustakaan menawarkan strategi pelatihan paralel data dan model paralel. Mereka menggabungkan teknologi perangkat lunak dan perangkat keras untuk meningkatkan komunikasi antar-GPU dan antar-simpul, dan memperluas SageMaker kemampuan pelatihan dengan opsi bawaan yang memerlukan perubahan kode minimal pada skrip pelatihan Anda.

Gunakan perpustakaan paralelisme data SageMaker terdistribusi (SMDDP)

Pustaka SMDDP meningkatkan komunikasi antar node dengan implementasi `AllReduce` dan operasi komunikasi `AllGather` kolektif yang dioptimalkan untuk infrastruktur AWS jaringan dan topologi instans Amazon SageMaker ML. Anda dapat menggunakan pustaka SMDDP sebagai backend paket pelatihan terdistribusi PyTorch berbasis: distributed PyTorch data parallel (DDP), PyTorch full sharded data parallelism (FSDP), dan Megatron-. DeepSpeedDeepSpeed Contoh kode berikut menunjukkan cara mengatur `PyTorch` estimator untuk meluncurkan pekerjaan pelatihan terdistribusi pada dua `ml.p4d.24xlarge` instance.

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(
    ...,
    instance_count=2,
    instance_type="ml.p4d.24xlarge",
```

```
    # Activate distributed training with SMDDP
    distribution={ "pytorchddp": { "enabled": True } }  # mpirun, activates SMDDP
 AllReduce OR AllGather
    # distribution={ "torch_distributed": { "enabled": True } }  # torchrun, activates
 SMDDP AllGather
    # distribution={ "smdistributed": { "dataparallel": { "enabled": True } } }  #
 mpirun, activates SMDDP AllReduce OR AllGather
)
```

Untuk mempelajari cara menyiapkan skrip pelatihan Anda dan meluncurkan pekerjaan pelatihan paralel data terdistribusi SageMaker, lihat. the section called “SageMaker perpustakaan paralelisme data terdistribusi”

Gunakan perpustakaan paralelisme SageMaker model (SMP)

SageMaker menyediakan perpustakaan SMP dan mendukung berbagai teknik pelatihan terdistribusi, seperti paralelisme data sharded, pipelining, paralelisme tensor, sharding status pengoptimal, dan banyak lagi. Untuk mempelajari lebih lanjut tentang apa yang ditawarkan perpustakaan SMP, lihatthe section called “Fitur Inti”.

Untuk menggunakan SageMaker pustaka paralelisme model, konfigurasikan `distribution` parameter estimator SageMaker kerangka kerja. Estimator kerangka kerja yang didukung adalah PyTorchdan TensorFlow. Contoh kode berikut menunjukkan bagaimana membangun estimator kerangka kerja untuk pelatihan terdistribusi dengan pustaka paralelisme model pada dua instance. `ml.p4d.24xlarge`

```
from sagemaker.framework import Framework

distribution={
    "smdistributed": {
        "modelparallel": {
            "enabled":True,
            "parameters": {
                ...   # enter parameter key-value pairs here
            }
        },
    },
    "mpi": {
        "enabled" : True,
        ...           # enter parameter key-value pairs here
    }
}
```

```
estimator = Framework(
    ...,
    instance_count=2,
    instance_type="ml.p4d.24xlarge",
    distribution=distribution
)
```

Untuk mempelajari cara mengadaptasi skrip pelatihan Anda, mengonfigurasi parameter distribusi di `estimator` kelas, dan meluncurkan tugas pelatihan terdistribusi, lihat SageMakerpustaka paralelisme model (lihat juga API Pelatihan Terdistribusi dalam dokumentasi SageMaker Python SDK).

Gunakan kerangka kerja pelatihan terdistribusi open source

SageMaker juga mendukung opsi berikut untuk beroperasi `mpirun` dan `torchrun` di backend.

- Untuk menggunakan PyTorch DistributedDataParallel (DDP) SageMaker dengan `mpirun` backend, tambahkan `distribution={"pytorchddp": {"enabled": True}}` ke estimator Anda. PyTorch Untuk informasi selengkapnya, lihat juga `distribution` Argumen Pelatihan PyTorch Terdistribusi dan SageMaker PyTorch Estimator dalam dokumentasi SageMaker Python SDK.

  > **Note**
  >
  > Opsi ini tersedia untuk PyTorch 1.12.0 dan yang lebih baru.

  ```
  from sagemaker.pytorch import PyTorch

  estimator = PyTorch(
      ...,
      instance_count=2,
      instance_type="ml.p4d.24xlarge",
      distribution={"pytorchddp": {"enabled": True}}  # runs mpirun in the backend
  )
  ```

- SageMaker mendukung PyTorch `torchrun`peluncur untuk pelatihan terdistribusi pada instans Amazon EC2 berbasis GPU, seperti P3 dan P4, serta Trn1 yang didukung oleh perangkat Trainium. AWS

Untuk menggunakan PyTorch DistributedDataParallel (DDP) SageMaker dengan `torchrun` backend, tambahkan `distribution={"torch_distributed": {"enabled": True}}` ke estimator. PyTorch

> **Note**
>
> Opsi ini tersedia untuk PyTorch 1.13.0 dan yang lebih baru.

Cuplikan kode berikut menunjukkan contoh membangun SageMaker PyTorch estimator untuk menjalankan pelatihan terdistribusi pada dua `ml.p4d.24xlarge` instance dengan opsi distribusi. `torch_distributed`

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(
    ...,
    instance_count=2,
    instance_type="ml.p4d.24xlarge",
    distribution={"torch_distributed": {"enabled": True}}   # runs torchrun in the
 backend
)
```

Untuk informasi selengkapnya, lihat `distribution` Argumen PyTorch Pelatihan Terdistribusi dan SageMaker PyTorch Estimator dalam dokumentasi SageMaker Python SDK.

Catatan untuk pelatihan terdistribusi di Trn1

Sebuah instance Trn1 terdiri dari hingga 16 perangkat Trainium, dan setiap perangkat Trainium terdiri dari dua. NeuronCores Untuk spesifikasi perangkat AWS Trainium, lihat Arsitektur Trainium di Dokumentasi Neuron. AWS

Untuk melatih instance yang didukung Trainium, Anda hanya perlu menentukan kode instance Trn1`ml.trn1.*`, dalam string ke argumen kelas estimator. `instance_type` SageMaker PyTorch Untuk menemukan jenis instance Trn1 yang tersedia, lihat Arsitektur AWS Trn1 dalam dokumentasi Neuron. AWS

> **Note**
>
> SageMaker Pelatihan tentang instans Amazon EC2 Trn1 saat ini hanya tersedia untuk PyTorch kerangka kerja di AWS Deep Learning Containers for Neuron mulai v1.11.0. PyTorch Untuk menemukan daftar lengkap versi Neuron yang didukung, lihat PyTorch Neuron Containers di GitHub repositori AWS Deep Learning Containers.

Saat Anda meluncurkan pekerjaan pelatihan pada instance Trn1 menggunakan SageMaker Python SDK, SageMaker secara otomatis mengambil dan menjalankan container yang tepat dari Neuron Containers yang disediakan oleh Deep Learning Containers. AWS Neuron Container dikemas dengan pengaturan lingkungan pelatihan dan dependensi untuk memudahkan adaptasi pekerjaan pelatihan Anda ke platform Pelatihan dan instans Amazon EC2 SageMaker Trn1.

> **Note**
>
> Untuk menjalankan pekerjaan PyTorch pelatihan Anda pada instance Trn1 dengan SageMaker, Anda harus memodifikasi skrip pelatihan Anda untuk menginisialisasi grup proses dengan `xla` backend dan menggunakan /XLA. PyTorch Untuk mendukung proses adopsi XLA, Neuron SDK menyediakan AWS PyTorch Neuron yang menggunakan XLA untuk melakukan konversi PyTorch operasi ke instruksi Trainium. Untuk mempelajari cara memodifikasi skrip pelatihan Anda, lihat Panduan Pengembang untuk Pelatihan dengan PyTorch Neuron (`torch-neuronx`) di Dokumentasi AWS Neuron.

Untuk informasi lebih lanjut, lihat Pelatihan Terdistribusi dengan PyTorch Neuron pada instance Trn1 dan argumen SageMaker PyTorch Estimator *`distribution`* dalam dokumentasi Python SageMaker SDK.

- Untuk menggunakan MPI SageMaker, tambahkan `distribution={"mpi": {"enabled": True}}` ke estimator Anda. Opsi distribusi MPI tersedia untuk kerangka kerja berikut: MxNet,, dan. PyTorch TensorFlow
- Untuk menggunakan server parameter di SageMaker, tambahkan `distribution={"parameter_server": {"enabled": True}}` ke estimator Anda. Opsi server parameter tersedia untuk kerangka kerja berikut: PyTorch MxNet,, dan. TensorFlow

> **Tip**
>
> Untuk informasi lebih lanjut tentang penggunaan MPI dan opsi server parameter per kerangka kerja, gunakan tautan berikut ke dokumentasi SageMaker Python SDK.
>
> - Pelatihan Terdistribusi MXNet dan argumen MxNet SageMaker Estimator `distribution`
> - PyTorch Argumen Pelatihan dan SageMaker PyTorch Estimator`distribution` Terdistribusi
> - TensorFlow `distribution`Argumen Pelatihan dan SageMaker TensorFlow Estimator Terdistribusi.

Konsep pelatihan terdistribusi dasar

SageMakerPerpustakaan pelatihan terdistribusi menggunakan istilah dan fitur pelatihan terdistribusi berikut.

Dataset dan Batch

- Dataset Pelatihan: Semua data yang Anda gunakan untuk melatih model.
- Ukuran batch global: Jumlah record yang dipilih dari dataset pelatihan di setiap iterasi untuk dikirim ke GPU di cluster. Ini adalah jumlah catatan di mana gradien dihitung pada setiap iterasi. Jika paralelisme data digunakan, itu sama dengan jumlah total replika model dikalikan dengan ukuran batch per replika:. `global batch size = (the number of model replicas) * (per-replica batch size)` Satu batch ukuran batch global sering disebut sebagai mini-batch dalam literatur pembelajaran mesin.
- Ukuran batch per replika: Ketika paralelisme data digunakan, ini adalah jumlah catatan yang dikirim ke setiap replika model. Setiap replika model melakukan pass maju dan mundur dengan batch ini untuk menghitung pembaruan bobot. Pembaruan bobot yang dihasilkan disinkronkan (dirata-ratakan) di semua replika sebelum kumpulan batch per replika berikutnya diproses.
- Micro-batch: Subset dari mini-batch atau, jika model hibrida dan paralelisme data digunakan, itu adalah subset dari batch berukuran per replika. Saat Anda menggunakan SageMaker pustaka paralelisme model terdistribusi, setiap batch mikro dimasukkan ke dalam pipeline pelatihan one-by-one dan mengikuti jadwal eksekusi yang ditentukan oleh runtime pustaka.

Pelatihan

- Epoch: Satu siklus pelatihan melalui seluruh kumpulan data. Adalah umum untuk memiliki beberapa iterasi per zaman. Jumlah epoch yang Anda gunakan dalam pelatihan unik pada model dan kasus penggunaan Anda.
- Iterasi: Satu pass maju dan mundur dilakukan menggunakan batch data pelatihan berukuran batch global (batch mini). Jumlah iterasi yang dilakukan selama pelatihan ditentukan oleh ukuran batch global dan jumlah zaman yang digunakan untuk pelatihan. Misalnya, jika kumpulan data menyertakan 5.000 sampel, dan Anda menggunakan ukuran batch global 500, dibutuhkan 10 iterasi untuk menyelesaikan satu epoch.
- Tingkat pembelajaran: Variabel yang mempengaruhi jumlah bobot diubah sebagai respons terhadap kesalahan model yang dihitung. Tingkat pembelajaran memainkan peran penting dalam kemampuan model untuk konvergen serta kecepatan dan optimalitas konvergensi.

Instans dan GPU

- Instance: Instance komputasi pembelajaran AWS mesin. Ini juga disebut sebagai node.
- Ukuran klaster: Saat menggunakan SageMaker pustaka pelatihan terdistribusi, ini adalah jumlah instance yang dikalikan dengan jumlah GPU di setiap instance. Misalnya, jika Anda menggunakan dua instance ml.p3.8xlarge dalam pekerjaan pelatihan, yang masing-masing memiliki 4 GPU, ukuran klaster adalah 8. Sementara peningkatan ukuran cluster dapat menyebabkan waktu pelatihan yang lebih cepat, komunikasi antar instance harus dioptimalkan; Jika tidak, komunikasi antar node dapat menambah overhead dan menyebabkan waktu pelatihan lebih lambat. Pustaka pelatihan SageMaker terdistribusi dirancang untuk mengoptimalkan komunikasi antara instans komputasi Amazon EC2 MLL, yang mengarah ke pemanfaatan perangkat yang lebih tinggi dan waktu pelatihan yang lebih cepat.

Solusi Pelatihan Terdistribusi

- Paralelisme data: Strategi dalam pelatihan terdistribusi di mana kumpulan data pelatihan dibagi menjadi beberapa GPU dalam klaster komputasi, yang terdiri dari beberapa Instans Amazon EC2 ML2. Setiap GPU berisi replika model, menerima batch data pelatihan yang berbeda, melakukan pass maju dan mundur, dan berbagi pembaruan bobot dengan node lain untuk sinkronisasi sebelum pindah ke batch berikutnya dan pada akhirnya zaman lain.
- Paralelisme model: Strategi dalam pelatihan terdistribusi di mana model dipartisi di beberapa GPU dalam cluster komputasi, yang terdiri dari beberapa Instans Amazon EC2 ML. Modelnya mungkin

rumit dan memiliki sejumlah besar lapisan dan bobot tersembunyi, sehingga tidak dapat masuk ke dalam memori satu instance. Setiap GPU membawa subset dari model, di mana aliran data dan transformasi dibagikan dan dikompilasi. Efisiensi paralelisme model, dalam hal pemanfaatan GPU dan waktu pelatihan, sangat tergantung pada bagaimana model dipartisi dan jadwal eksekusi yang digunakan untuk melakukan pass maju dan mundur.

- Jadwal Eksekusi Pipeline (Pipelining): Jadwal eksekusi pipeline menentukan urutan pembuatan komputasi (batch mikro) dan data diproses di seluruh perangkat selama pelatihan model. Pipelining adalah teknik untuk mencapai paralelisasi sejati dalam paralelisme model dan mengatasi kehilangan kinerja karena komputasi sekuensial dengan membuat GPU menghitung secara bersamaan pada sampel data yang berbeda. Untuk mempelajari selengkapnya, lihat Jadwal Eksekusi Pipeline.

Konsep lanjutan

Praktisi Machine Learning (ML) biasanya menghadapi dua tantangan penskalaan saat model pelatihan: ukuran model penskalaan dan penskalaan data pelatihan. Meskipun ukuran dan kompleksitas model dapat menghasilkan akurasi yang lebih baik, ada batasan ukuran model yang dapat Anda masukkan ke dalam satu CPU atau GPU. Selain itu, ukuran model penskalaan dapat menghasilkan lebih banyak perhitungan dan waktu pelatihan yang lebih lama.

Tidak semua model menangani penskalaan data pelatihan dengan sama baiknya karena mereka perlu menelan semua data pelatihan dalam memori untuk pelatihan. Mereka hanya menskalakan secara vertikal, dan ke jenis instance yang lebih besar dan lebih besar. Dalam kebanyakan kasus, penskalaan data pelatihan menghasilkan waktu pelatihan yang lebih lama.

Deep Learning (DL) adalah keluarga spesifik algoritma ML yang terdiri dari beberapa lapisan jaringan saraf tiruan. Metode pelatihan yang paling umum adalah dengan mini-batch Stochastic Gradient Descent (SGD). Dalam mini-batch SGD, model dilatih dengan melakukan perubahan iteratif kecil dari koefisiennya ke arah yang mengurangi kesalahannya. Iterasi tersebut dilakukan pada subsampel berukuran sama dari kumpulan data pelatihan yang disebut mini-batch. Untuk setiap mini-batch, model dijalankan di setiap catatan mini-batch, kesalahannya diukur dan gradien kesalahan diperkirakan. Kemudian gradien rata-rata diukur di semua catatan mini-batch dan memberikan arah pembaruan untuk setiap koefisien model. Satu lintasan penuh atas kumpulan data pelatihan disebut zaman. Pelatihan model biasanya terdiri dari puluhan hingga ratusan zaman. Mini-batch SGD memiliki beberapa manfaat: Pertama, desain iteratifnya membuat waktu pelatihan secara teoritis linier dari ukuran dataset. Kedua, dalam mini-batch tertentu setiap rekaman diproses secara individual oleh

model tanpa perlu komunikasi antar-rekaman selain rata-rata gradien akhir. Pemrosesan batch mini akibatnya sangat cocok untuk paralelisasi dan distribusi.

Paralelisasi pelatihan SGD dengan mendistribusikan catatan mini-batch melalui perangkat komputasi yang berbeda disebut pelatihan terdistribusi paralel data, dan merupakan paradigma distribusi DL yang paling umum digunakan. Pelatihan paralel data adalah strategi distribusi yang relevan untuk menskalakan ukuran batch mini dan memproses setiap batch mini lebih cepat. Namun, pelatihan paralel data hadir dengan kompleksitas ekstra karena harus menghitung rata-rata gradien batch mini dengan gradien yang berasal dari semua pekerja dan mengkomunikasikannya ke semua pekerja, sebuah langkah yang disebut allreduce yang dapat mewakili overhead yang berkembang, karena cluster pelatihan diskalakan, dan itu juga dapat secara drastis menghukum waktu pelatihan jika diterapkan atau diimplementasikan secara tidak benar pada pengurangan perangkat keras yang tidak tepat.

Data parallel SGD masih mengharuskan pengembang untuk dapat menyesuaikan setidaknya model dan satu catatan dalam perangkat komputasi, seperti CPU tunggal atau GPU. Saat melatih model yang sangat besar seperti transformator besar di Natural Language Processing (NLP), atau model segmentasi pada gambar resolusi tinggi, mungkin ada situasi di mana hal ini tidak layak. Cara alternatif untuk memecah beban kerja adalah dengan mempartisi model melalui beberapa perangkat komputasi, sebuah pendekatan yang disebut pelatihan terdistribusi model-paralel.

Strategi

Pelatihan terdistribusi biasanya dibagi oleh dua pendekatan: data parallel dan model parallel. Data parallel adalah pendekatan yang paling umum untuk pelatihan terdistribusi: Anda memiliki banyak data, menggabungkannya, dan mengirim blok data ke beberapa CPU atau GPU (node) untuk diproses oleh jaringan saraf atau algoritma ML. kemudian menggabungkan hasilnya. Jaringan saraf sama pada setiap node. Pendekatan paralel model digunakan dengan model besar yang tidak akan muat dalam memori node dalam satu bagian; itu memecah model dan menempatkan bagian yang berbeda pada node yang berbeda. Dalam situasi ini, Anda perlu mengirim batch data Anda ke setiap node sehingga data diproses pada semua bagian model.

Istilah jaringan dan model sering digunakan secara bergantian: Model besar benar-benar jaringan besar dengan banyak lapisan dan parameter. Pelatihan dengan jaringan besar menghasilkan model yang besar, dan memuat model kembali ke jaringan dengan semua parameter Anda yang telah dilatih sebelumnya dan bobotnya memuat model besar ke dalam memori. Saat Anda memecah model untuk membaginya di seluruh node, Anda juga memecah jaringan yang mendasarinya.

Jaringan terdiri dari lapisan, dan untuk membagi jaringan, Anda menempatkan lapisan pada perangkat komputasi yang berbeda.

Perangkap umum dari pemisahan lapisan secara naif di seluruh perangkat adalah pemanfaatan GPU yang parah. Pelatihan secara inheren berurutan dalam lintasan maju dan mundur, dan pada waktu tertentu, hanya satu GPU yang dapat secara aktif menghitung, sementara yang lain menunggu aktivasi yang akan dikirim. Pustaka paralel model modern memecahkan masalah ini dengan menggunakan jadwal eksekusi pipeline untuk meningkatkan pemanfaatan perangkat. Namun, hanya perpustakaan paralel model SageMaker terdistribusi Amazon yang menyertakan pemisahan model otomatis. Dua fitur inti perpustakaan, pemisahan model otomatis dan penjadwalan eksekusi pipa, menyederhanakan proses penerapan paralelisme model dengan membuat keputusan otomatis yang mengarah pada pemanfaatan perangkat yang efisien.

Berlatih dengan data parallel dan model parallel

Jika Anda berlatih dengan kumpulan data besar, mulailah dengan pendekatan paralel data. Jika Anda kehabisan memori selama pelatihan, Anda mungkin ingin beralih ke pendekatan paralel model, atau mencoba model hibrida dan paralelisme data. Anda juga dapat mencoba yang berikut ini untuk meningkatkan kinerja dengan data parallel:

- Ubah hiperparameter model Anda.
- Kurangi ukuran batch.
- Terus kurangi ukuran batch hingga pas. Jika Anda mengurangi ukuran batch menjadi 1, dan masih kehabisan memori, maka Anda harus mencoba pelatihan model-paralel.

Coba kompresi gradien (FP16, INT8):

- Pada perangkat keras TensorCore yang dilengkapi NVIDIA, menggunakan pelatihan presisi campuran menciptakan pengurangan kecepatan dan konsumsi memori.
- SageMakerPustaka paralelisme data terdistribusi mendukung Automatic Mixed Precision (AMP) di luar kotak. Tidak diperlukan tindakan tambahan untuk mengaktifkan AMP selain modifikasi tingkat kerangka kerja pada skrip pelatihan Anda. Jika gradien berada di FP16, pustaka paralelisme SageMaker data menjalankan operasinya di FP16. `AllReduce` Untuk informasi selengkapnya tentang penerapan API AMP ke skrip pelatihan Anda, lihat sumber daya berikut:
  - Kerangka kerja - PyTorch dalam dokumentasi NVIDIA Deep Learning Performance
  - Kerangka kerja - TensorFlow dalam dokumentasi NVIDIA Deep Learning Performance
  - Presisi Campuran Otomatis untuk Pembelajaran Mendalam di Dokumen Pengembang NVIDIA

- Memperkenalkan presisi campuran PyTorch otomatis asli untuk pelatihan lebih cepat pada GPU NVIDIA di Blog PyTorch
- TensorFlow API presisi campuran dalam TensorFlowdokumentasi

Coba kurangi ukuran input:

- Kurangi panjang urutan NLP jika Anda menambah tautan urutan, perlu menyesuaikan ukuran batch ke bawah, atau menyesuaikan GPU untuk menyebarkan batch.
- Kurangi resolusi gambar.

Periksa apakah Anda menggunakan normalisasi batch, karena ini dapat memengaruhi konvergensi. Saat Anda menggunakan pelatihan terdistribusi, batch Anda dibagi menjadi GPU dan efek dari ukuran batch yang jauh lebih rendah bisa menjadi tingkat kesalahan yang lebih tinggi sehingga mengganggu model dari konvergen. Misalnya, jika Anda membuat prototipe jaringan pada satu GPU dengan ukuran batch 64, kemudian ditingkatkan hingga menggunakan empat p3dn.24xlarge, Anda sekarang memiliki 32 GPU dan ukuran batch per GPU Anda turun dari 64 menjadi 2. Ini kemungkinan akan mematahkan konvergensi yang Anda lihat dengan satu simpul.

Mulailah dengan pelatihan model-paralel ketika:

- Model Anda tidak muat pada satu perangkat.
- Karena ukuran model Anda, Anda menghadapi keterbatasan dalam memilih ukuran batch yang lebih besar, seperti jika bobot model Anda menghabiskan sebagian besar memori GPU Anda dan Anda terpaksa memilih ukuran batch yang lebih kecil dan kurang optimal.

Untuk mempelajari lebih lanjut tentang pustaka SageMaker terdistribusi, lihat yang berikut ini:

- Jalankan pelatihan terdistribusi dengan SageMaker pustaka paralelisme data terdistribusi
- (Diarsipkan) perpustakaan SageMaker paralelisme model v1.x

Optimalkan pelatihan terdistribusi

Sesuaikan hyperparameters untuk kasus penggunaan dan data Anda untuk mendapatkan efisiensi penskalaan terbaik. Dalam diskusi berikut, kami menyoroti beberapa variabel pelatihan yang paling berdampak dan memberikan referensi untuk state-of-the-art implementasi sehingga Anda

dapat mempelajari lebih lanjut tentang opsi Anda. Selain itu, kami menyarankan Anda merujuk ke dokumentasi pelatihan terdistribusi kerangka kerja pilihan Anda.

- Apache MXNet didistribusikan pelatihan
- PyTorch pelatihan terdistribusi
- TensorFlow pelatihan terdistribusi

Ukuran Batch

SageMaker toolkit terdistribusi umumnya memungkinkan Anda untuk berlatih pada batch yang lebih besar. Misalnya, jika model cocok dalam satu perangkat tetapi hanya dapat dilatih dengan ukuran batch kecil, menggunakan pelatihan paralel model atau pelatihan paralel data memungkinkan Anda bereksperimen dengan ukuran batch yang lebih besar.

Ketahuilah bahwa ukuran batch secara langsung memengaruhi akurasi model dengan mengontrol jumlah noise dalam pembaruan model pada setiap iterasi. Meningkatkan ukuran batch mengurangi jumlah noise dalam estimasi gradien, yang dapat bermanfaat ketika meningkat dari ukuran batch yang sangat kecil, tetapi dapat menghasilkan akurasi model yang terdegradasi karena ukuran batch meningkat ke nilai yang besar.

> **Tip**
>
> Sesuaikan hyperparameters Anda untuk memastikan bahwa model Anda berlatih ke konvergensi yang memuaskan saat Anda meningkatkan ukuran batch.

Sejumlah teknik telah dikembangkan untuk mempertahankan konvergensi model yang baik ketika batch ditingkatkan.

Ukuran mini-batch

Dalam SGD, ukuran mini-batch mengukur jumlah noise yang ada dalam estimasi gradien. Batch mini kecil menghasilkan gradien batch mini yang sangat bising, yang tidak mewakili gradien sebenarnya di atas kumpulan data. Batch mini yang besar menghasilkan gradien batch mini yang mendekati gradien sebenarnya di atas kumpulan data dan berpotensi tidak cukup berisik — kemungkinan akan tetap terkunci dalam minimum yang tidak relevan.

Untuk mempelajari lebih lanjut tentang teknik ini, lihat makalah berikut:

- Akurat, Minibatch Besar SGD: Pelatihan ImageNet dalam 1 Jam, Goya et al.
- PowerAI DDL, Cho dkk.
- Skala Keluar untuk Minibatch Besar SGD: Pelatihan Jaringan Sisa pada ImageNet -1K dengan Peningkatan Akurasi dan Pengurangan Waktu untuk Melatih, Codreanu et al.
- ImageNet Pelatihan dalam Menit, You et al.
- Pelatihan Batch Besar Jaringan Konvolusional, You et al.
- Optimalisasi Batch Besar untuk Pembelajaran Mendalam: Melatih BERT dalam 76 Menit, You et al.
- Optimalisasi Batch Besar yang Dipercepat dari PRETRAINING BERT dalam 54 menit, Zheng et al.
- Kompresi Gradien Dalam, Lin et al.

Skenario

Bagian berikut mencakup skenario di mana Anda mungkin ingin meningkatkan pelatihan, dan bagaimana Anda dapat melakukannya menggunakan AWS sumber daya.

Penskalaan dari GPU Tunggal ke Banyak GPU

Jumlah data atau ukuran model yang digunakan dalam pembelajaran mesin dapat menciptakan situasi di mana waktu untuk melatih model lebih lama sehingga Anda bersedia menunggu. Terkadang, pelatihan tidak berfungsi sama sekali karena model atau data pelatihan terlalu besar. Salah satu solusinya adalah meningkatkan jumlah GPU yang Anda gunakan untuk pelatihan. Pada instance dengan beberapa GPU, seperti `p3.16xlarge` yang memiliki delapan GPU, data dan pemrosesan dibagi menjadi delapan GPU. Saat Anda menggunakan pustaka pelatihan terdistribusi, ini dapat menghasilkan percepatan hampir linier dalam waktu yang diperlukan untuk melatih model Anda. Dibutuhkan sedikit lebih dari 1/8 waktu yang akan diambil `p3.2xlarge` dengan satu GPU.

Jenis instans	GPU
p3.2xlarge	1
p3.8xlarge	4
p3.16xlarge	8
p3dn.24xlarge	8

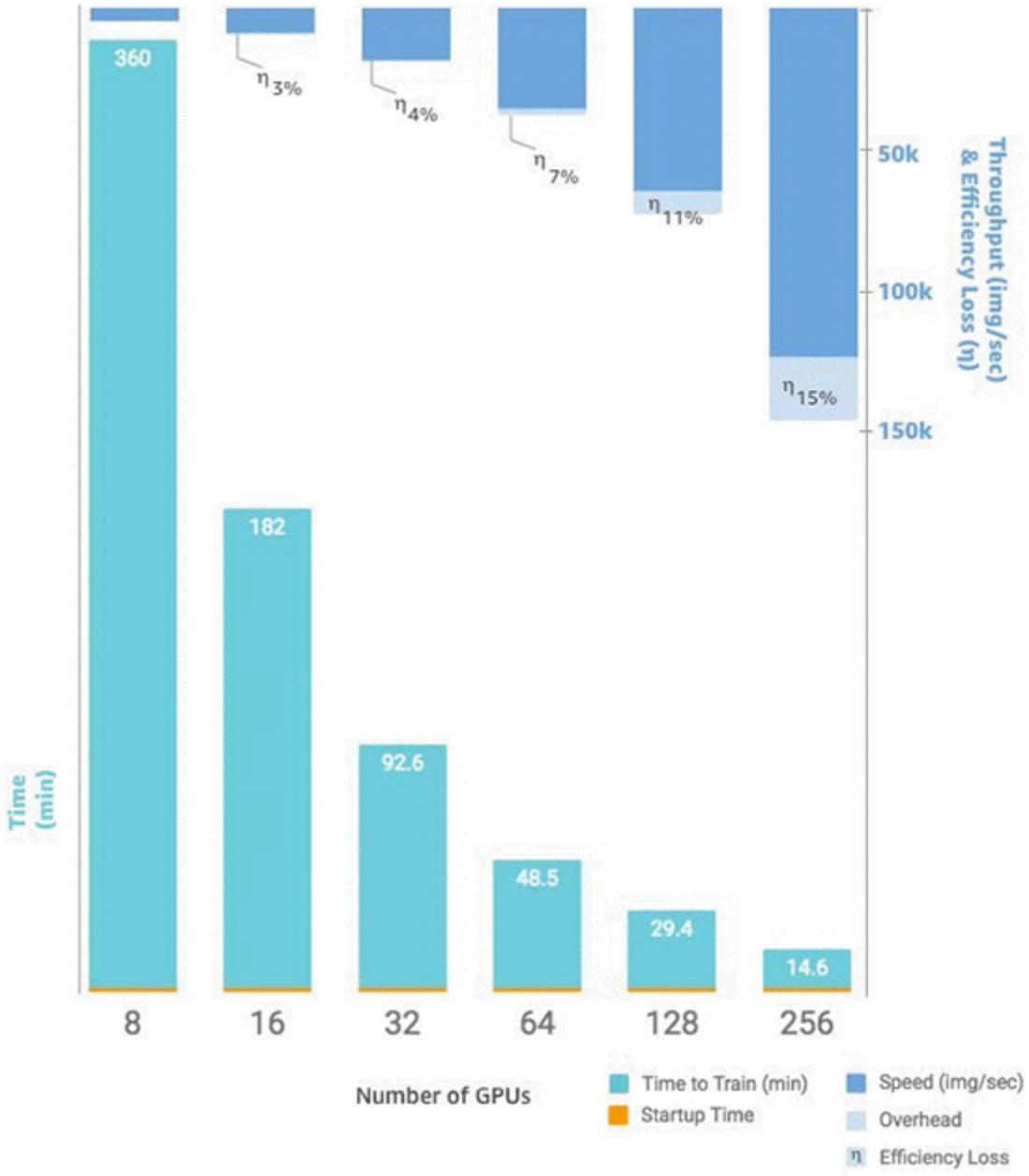
> **Note**
>
> Jenis instans mL yang digunakan oleh SageMaker pelatihan memiliki jumlah GPU yang sama dengan tipe instans p3 yang sesuai. Misalnya, `ml.p3.8xlarge` memiliki jumlah GPU yang sama dengan `p3.8xlarge` - 4.

Penskalaan dari satu instance ke beberapa instance

Jika Anda ingin meningkatkan skala pelatihan Anda lebih jauh, Anda dapat menggunakan lebih banyak contoh. Namun, Anda harus memilih jenis instans yang lebih besar sebelum menambahkan lebih banyak instance. Tinjau tabel sebelumnya untuk melihat berapa banyak GPU di setiap jenis instans p3.

Jika Anda telah melakukan lompatan dari satu GPU pada satu `p3.2xlarge` ke empat GPU pada ap`3.8xlarge`, tetapi memutuskan bahwa Anda memerlukan lebih banyak daya pemrosesan, Anda mungkin melihat kinerja yang lebih baik dan mengeluarkan biaya yang lebih rendah jika Anda memilih `p3.16xlarge` sebelum mencoba meningkatkan jumlah instans. Bergantung pada pustaka yang Anda gunakan, ketika Anda mempertahankan pelatihan Anda pada satu instance, kinerja lebih baik dan biaya lebih rendah daripada skenario di mana Anda menggunakan beberapa instance.

Ketika Anda siap untuk menskalakan jumlah instance, Anda dapat melakukan ini dengan fungsi SageMaker Python `estimator` SDK dengan menyetel. `instance_count` Misalnya, Anda dapat mengatur `instance_type = p3.16xlarge` dan`instance_count = 2`. Alih-alih delapan GPU pada satu`p3.16xlarge`, Anda memiliki 16 GPU di dua instance yang identik. Bagan berikut menunjukkan penskalaan dan throughput dimulai dengan delapan GPU pada satu instance dan meningkat menjadi 64 instance dengan total 256 GPU.



Skrip pelatihan khusus

Meskipun SageMaker membuatnya mudah untuk menyebarkan dan menskalakan jumlah instance dan GPU, tergantung pada kerangka kerja pilihan Anda, mengelola data dan hasil bisa sangat menantang, itulah sebabnya pustaka pendukung eksternal sering digunakan. Bentuk pelatihan terdistribusi yang paling dasar ini memerlukan modifikasi skrip pelatihan Anda untuk mengelola distribusi data.

SageMaker juga mendukung Horovod dan implementasi pelatihan terdistribusi asli untuk setiap kerangka pembelajaran mendalam utama. Jika Anda memilih untuk menggunakan contoh dari kerangka kerja ini, Anda dapat mengikuti SageMaker panduan kontainer untuk Deep Learning Containers, dan berbagai contoh notebook yang menunjukkan implementasi.

Jalankan pelatihan terdistribusi dengan SageMaker pustaka paralelisme data terdistribusi

Perpustakaan paralelisme data SageMaker terdistribusi (SMDDP) memperluas kemampuan SageMaker pelatihan pada model pembelajaran mendalam dengan efisiensi penskalaan hampir linier dengan menyediakan implementasi operasi komunikasi kolektif yang dioptimalkan untuk infrastruktur. AWS

Saat melatih model pembelajaran mesin besar (MLM), seperti model bahasa besar (LLM) dan model difusi, pada kumpulan data pelatihan yang besar, praktisi ML menggunakan kelompok akselerator dan teknik pelatihan terdistribusi untuk mengurangi waktu melatih atau menyelesaikan kendala memori untuk model yang tidak dapat muat di setiap memori GPU. Praktisi ML sering memulai dengan beberapa akselerator pada satu instance dan kemudian menskalakan ke kelompok instance saat persyaratan beban kerja mereka meningkat. Ketika ukuran cluster meningkat, demikian juga overhead komunikasi antara beberapa node, yang menyebabkan penurunan kinerja komputasi secara keseluruhan.

Untuk mengatasi masalah overhead dan memori seperti itu, perpustakaan SMDDP menawarkan yang berikut ini.

- Pustaka SMDDP mengoptimalkan pekerjaan pelatihan untuk infrastruktur AWS jaringan dan topologi instans Amazon SageMaker MLL.
- Perpustakaan SMDDP meningkatkan komunikasi antar node dengan implementasi `AllReduce` dan operasi komunikasi `AllGather` kolektif yang dioptimalkan untuk infrastruktur. AWS

Untuk mempelajari lebih lanjut tentang detail penawaran perpustakaan SMDDP, lanjutkan ke. the section called “Pengantar perpustakaan SMDDP”

Untuk informasi lebih lanjut tentang pelatihan dengan strategi model-paralel yang ditawarkan oleh SageMaker, lihat juga. (Diarsipkan) perpustakaan SageMaker paralelisme model v1.x

Topik

- Pengantar perpustakaan paralelisme data SageMaker terdistribusi

- Kerangka kerja yang didukung,Wilayah AWS, dan tipe instance
- Cara menjalankan pekerjaan pelatihan terdistribusi dengan pustaka paralelisme data SageMaker terdistribusi
- Kiat konfigurasi untuk pustaka paralelisme data SageMaker terdistribusi
- FAQ perpustakaan paralelisme data SageMaker terdistribusi Amazon
- Pemecahan masalah untuk pelatihan terdistribusi di Amazon SageMaker

Pengantar perpustakaan paralelisme data SageMaker terdistribusi

Pustaka paralelisme data SageMaker terdistribusi (SMDDP) adalah perpustakaan komunikasi kolektif yang meningkatkan kinerja komputasi pelatihan paralel data terdistribusi. Perpustakaan SMDDP menangani overhead komunikasi dari operasi komunikasi kolektif utama dengan menawarkan yang berikut ini.

1. Perpustakaan menawarkan `AllReduce` dioptimalkan untukAWS. `AllReduce`adalah operasi kunci yang digunakan untuk menyinkronkan gradien di seluruh GPU pada akhir setiap iterasi pelatihan selama pelatihan data terdistribusi.
2. Perpustakaan menawarkan `AllGather` dioptimalkan untukAWS. `AllGather`adalah operasi kunci lain yang digunakan dalam pelatihan paralelisme data sharded, yang merupakan teknik paralelisme data hemat memori yang ditawarkan oleh perpustakaan populer seperti perpustakaan SageMaker model paralelisme (SMP), DeepSpeed Zero Redundancy Optimizer (Zero), dan Fully Sharded Data Parallelism (FSDP). PyTorch
3. Pustaka melakukan node-to-node komunikasi yang dioptimalkan dengan sepenuhnya memanfaatkan infrastruktur AWS jaringan dan topologi instans Amazon EC2.

Pustaka SMDDP dapat meningkatkan kecepatan pelatihan dengan menawarkan peningkatan kinerja saat Anda menskalakan klaster pelatihan Anda, dengan efisiensi penskalaan hampir linier.

> **Note**
>
> Perpustakaan pelatihan SageMaker terdistribusi tersedia melalui wadah pembelajaran AWS mendalam untuk PyTorch dan Hugging Face dalam platform SageMaker Pelatihan. Untuk menggunakan library, Anda harus menggunakan SageMaker Python SDK atau API melalui SDK for SageMaker Python (Boto3) atau. AWS Command Line Interface Sepanjang

dokumentasi, instruksi dan contoh berfokus pada cara menggunakan pustaka pelatihan terdistribusi dengan SageMaker Python SDK.

Operasi komunikasi kolektif SMDDP dioptimalkan untuk sumber daya AWS komputasi dan infrastruktur jaringan

Perpustakaan SMDDP menyediakan implementasi operasi `AllGather` kolektif yang dioptimalkan untuk sumber daya AWS komputasi `AllReduce` dan infrastruktur jaringan.

Operasi kolektif SMDDP **`AllReduce`**

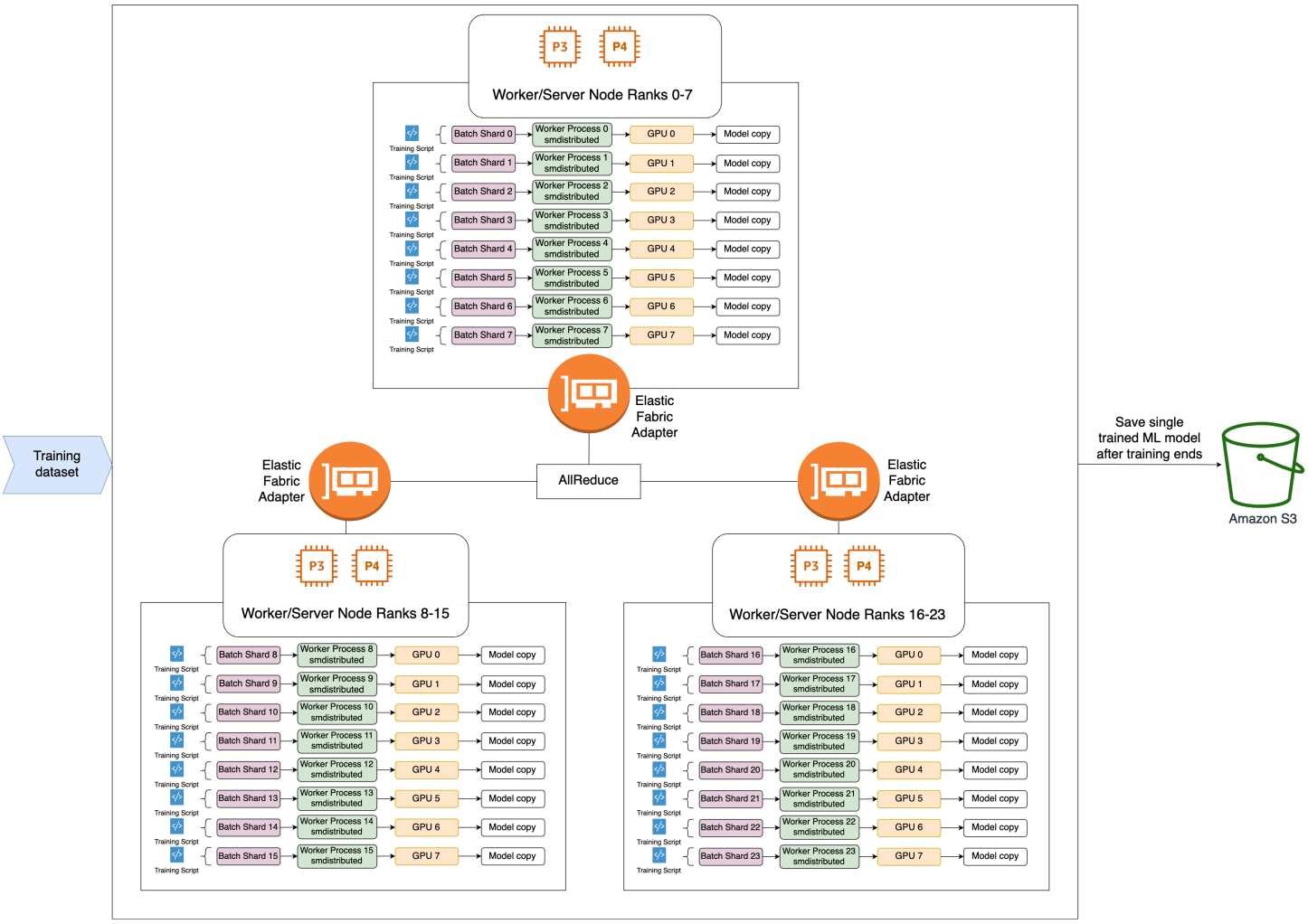
Pustaka SMDDP mencapai tumpang tindih `AllReduce` operasi yang optimal dengan backward pass, secara signifikan meningkatkan pemanfaatan GPU. Ini mencapai efisiensi penskalaan hampir linier dan kecepatan pelatihan yang lebih cepat dengan mengoptimalkan operasi kernel antara CPU dan GPU. Library bekerja `AllReduce` secara paralel saat GPU menghitung gradien tanpa menghilangkan siklus GPU tambahan, yang membuat perpustakaan mencapai pelatihan yang lebih cepat.

- Memanfaatkan CPU: Perpustakaan menggunakan CPU untuk `AllReduce` gradien, membongkar tugas ini dari GPU.
- Peningkatan penggunaan GPU: GPU cluster fokus pada gradien komputasi, meningkatkan pemanfaatannya selama pelatihan.

Berikut ini adalah alur kerja tingkat tinggi dari operasi `AllReduce` SMDDP.

1. Perpustakaan memberikan peringkat ke GPU (pekerja).
2. Pada setiap iterasi, perpustakaan membagi setiap batch global dengan jumlah total pekerja (ukuran dunia) dan memberikan batch kecil (pecahan batch) kepada pekerja.
   - Ukuran batch global adalah`(number of nodes in a cluster) * (number of GPUs per node) * (per batch shard)`.
   - Batch shard (batch kecil) adalah subset dari dataset yang ditetapkan untuk setiap GPU (pekerja) per iterasi.
3. Perpustakaan meluncurkan skrip pelatihan pada setiap pekerja.
4. Pustaka mengelola salinan bobot model dan gradien dari pekerja di akhir setiap iterasi.
5. Pustaka menyinkronkan bobot dan gradien model di seluruh pekerja untuk menggabungkan satu model terlatih.

Diagram arsitektur berikut menunjukkan contoh bagaimana perpustakaan mengatur paralelisme data untuk cluster 3 node.



Operasi kolektif SMDDP **`AllGather`**

`AllGather`adalah operasi kolektif di mana setiap pekerja memulai dengan buffer input, dan kemudian menggabungkan atau mengumpulkan buffer input dari semua pekerja lain ke dalam buffer output.

> **Note**
>
> Operasi `AllGather` kolektif SMDDP tersedia di AWS Deep Learning Containers (DLC) untuk PyTorch v2.0.1 `smdistributed-dataparallel>=2.0.1` dan yang lebih baru.

`AllGather`banyak digunakan dalam teknik pelatihan terdistribusi seperti paralelisme data sharded di mana setiap pekerja individu memegang sebagian kecil dari model, atau lapisan sharded. Para pekerja memanggil `AllGather` sebelum umpan maju dan mundur untuk merekonstruksi lapisan yang dipecah. Pass maju dan mundur terus berlanjut setelah semua parameter dikumpulkan. Selama backward pass, setiap pekerja juga memanggil `ReduceScatter` untuk mengumpulkan (mengurangi) gradien dan memecah (menyebarkan) mereka menjadi pecahan gradien untuk memperbarui lapisan sharded yang sesuai. Untuk detail lebih lanjut tentang peran operasi kolektif ini dalam paralelisme data sharded, lihat implementasi perpustakaan SMP tentang paralelisme data sharded, Zero dalam DeepSpeed dokumentasi, dan blog tentang Fully Sharded Data Parallelism. PyTorch

Karena operasi kolektif seperti AllGather dipanggil dalam setiap iterasi, mereka adalah kontributor utama untuk overhead komunikasi GPU. Perhitungan yang lebih cepat dari operasi kolektif ini secara langsung diterjemahkan ke waktu pelatihan yang lebih singkat tanpa efek samping pada konvergensi. Untuk mencapai hal ini, perpustakaan SMDDP menawarkan `AllGather` dioptimalkan untuk instance P4d.

SMDDP `AllGather` menggunakan teknik berikut untuk meningkatkan kinerja komputasi pada instance P4d.

1. Ini mentransfer data antar instance (antar-node) melalui jaringan Elastic Fabric Adapter (EFA) dengan topologi mesh. EFA adalah solusi jaringan AWS latensi rendah dan throughput tinggi. Topologi mesh untuk komunikasi jaringan antar simpul lebih disesuaikan dengan karakteristik EFA dan infrastruktur jaringan. AWS Dibandingkan dengan cincin NCCL atau topologi pohon yang melibatkan beberapa packet hop, SMDDP menghindari akumulasi latensi dari beberapa hop karena hanya membutuhkan satu hop. SMDDP mengimplementasikan algoritma kontrol laju jaringan yang menyeimbangkan beban kerja untuk setiap rekan komunikasi dalam topologi mesh dan mencapai throughput jaringan global yang lebih tinggi.
2. Ini mengadopsi perpustakaan salinan memori GPU latensi rendah berdasarkan teknologi NVIDIA GPUDirect RDMA (GDRCopy) untuk mengoordinasikan lalu lintas jaringan NVLink dan EFA lokal. GDRCopy, pustaka salinan memori GPU latensi rendah yang ditawarkan oleh NVIDIA, menyediakan komunikasi latensi rendah antara proses CPU dan kernel GPU CUDA. Dengan teknologi ini, perpustakaan SMDDP mampu menyalurkan pergerakan data intra-node dan antar-node.
3. Ini mengurangi penggunaan multiprosesor streaming GPU untuk meningkatkan daya komputasi untuk menjalankan kernel model. Instans P4d dan P4de dilengkapi dengan GPU NVIDIA A100, yang masing-masing memiliki 108 multiprosesor streaming. Sementara NCCL membutuhkan hingga 24 multiprosesor streaming untuk menjalankan operasi kolektif, SMDDP menggunakan

kurang dari 9 multiprosesor streaming. Kernel komputasi model mengambil multiprosesor streaming yang disimpan untuk komputasi yang lebih cepat.

Kerangka kerja yang didukung,Wilayah AWS, dan tipe instance

Sebelum menggunakan library parallelism data SageMaker terdistribusi (SMDDP), periksa kerangka kerja dan tipe instance yang didukung dan apakah ada cukup kuota di akun Anda dan. AWS Wilayah AWS

Kerangka kerja yang didukung

Tabel berikut menunjukkan kerangka pembelajaran mendalam dan versinya dan dukungan paralelisme data SageMaker terdistribusi. SageMaker Paralelisme data SageMaker terdistribusi tersedia dalam AWS Deep Learning Containers (DLC) atau dapat diunduh sebagai file biner.

> **Note**
>
> Untuk memeriksa pembaruan terbaru dan catatan rilis perpustakaan, lihat juga Catatan Rilis Paralel SageMaker Data dalam dokumentasi SageMaker Python SDK.

Topik

- PyTorch
- PyTorch Petir
- Trafo Hugging Face
- TensorFlow (usang)

PyTorch

PyTorch versi	SageMaker versi paralelisme data terdistribusi	**`smdistributed-dataparallel`** URI gambar terintegrasi	URL dari file biner**
v2.0.1	`smdistributed-dataparallel==v2.0.1`	763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:2.0.1-	https://smdataparallel.s3.amazonaws.com/binary/pytorch/2.0.1/cu118/2023-

PyTorch versi	SageMaker versi paralelisme data terdistribusi	**`smdistributed-dataparallel`** URI gambar terintegrasi	URL dari file biner**
		gpu-py310-cu118-ubuntu20.04-sagemaker	10-23/smdistributed_dataparallel-2.0.1-cp310-cp310-linux_x86_64.whl
v2.0.0	`smdistributed-dataparallel==v1.8.0`	763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:2.0.0-gpu-py310-cu118-ubuntu20.04-sagemaker	https://smdataparallel.s3.amazonaws.com/binary/pytorch/2.0.0/cu118/2023-03-20/smdistributed_dataparallel-1.8.0-cp310-cp310-linux_x86_64.whl
v1.13.1	`smdistributed-dataparallel==v1.7.0`	763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.13.1-gpu-py39-cu117-ubuntu20.04-sagemaker	https://smdataparallel.s3.amazonaws.com/binary/pytorch/1.13.1/cu117/2023-01-09/smdistributed_dataparallel-1.7.0-cp39-cp39-linux_x86_64.whl
v1.12.1	`smdistributed-dataparallel==v1.6.0`	763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.12.1-gpu-py38-cu113-ubuntu20.04-sagemaker	https://smdataparallel.s3.amazonaws.com/binary/pytorch/1.12.1/cu113/2022-12-05/smdistributed_dataparallel-1.6.0-cp38-cp38-linux_x86_64.whl

PyTorch versi	SageMaker versi paralelisme data terdistribusi	**`smdistributed-dataparallel`** URI gambar terintegrasi	URL dari file biner**
v1.12.0	`smdistributed-dataparallel==v1.5.0`	763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.12.0-gpu-py38-cu113-ubuntu20.04-sagemaker	https://smdataparallel.s3.amazonaws.com/binary/pytorch/1.12.0/cu113/2022-07-01/smdistributed_dataparallel-1.5.0-cp38-cp38-linux_x86_64.whl
v1.11.0	`smdistributed-dataparallel==v1.4.1`	763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.11.0-gpu-py38-cu113-ubuntu20.04-sagemaker	https://smdataparallel.s3.amazonaws.com/binary/pytorch/1.11.0/cu113/2022-04-14/smdistributed_dataparallel-1.4.1-cp38-cp38-linux_x86_64.whl

Note

Paralelisme data SageMaker terdistribusi v1.4.0 dan kemudian berfungsi sebagai backend paralelisme data terdistribusi (PyTorchtorch.distributed) (torch.parallel. DistributedDataParallel). Sesuai dengan perubahan, API smdistributed berikut untuk paket PyTorch terdistribusi tidak digunakan lagi.

- `smdistributed.dataparallel.torch.distributed`sudah usang. Gunakan paket torch.distributed sebagai gantinya.
- `smdistributed.dataparallel.torch.parallel.DistributedDataParallel`sudah usang. Gunakan torch.nn.parallel. DistributedDataParallelAPI sebagai gantinya.

Jika Anda perlu menggunakan versi pustaka sebelumnya (v1.3.0 atau sebelumnya), lihat dokumentasi paralelisme data SageMaker terdistribusi yang diarsipkan dalam dokumentasi Python SageMaker SDK.

** URL dari file biner adalah untuk menginstal paralelisme data SageMaker terdistribusi dalam wadah khusus. Untuk informasi selengkapnya, lihat Buat wadah Docker Anda sendiri dengan pustaka paralel data SageMaker terdistribusi.

PyTorch Petir

PyTorch Versi petir	PyTorch versi	SageMaker versi paralelisme data terdistribusi	**`smdistributed-dataparallel`** URI gambar terintegrasi	URL dari file biner**
1.7.2 1.7.0 1.6.4 1.6.3 1.5.10	1.12.0	`smdistributed-dataparallel==v1.5.0`	763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.12.0-gpu-py38-cu113-ubuntu20.04-sagemaker	https://smdataparallel.s3.amazonaws.com/binary/pytorch/1.12.0/cu113/2022-07-01/smdistributed_dataparallel-1.5.0-cp38-cp38-linux_x86_64.whl

Note

PyTorch Lightning dan perpustakaan utilitasnya seperti Lightning Bolts tidak diinstal sebelumnya di DLC. PyTorch Ketika Anda membangun SageMaker PyTorch estimator

dan mengajukan permintaan pekerjaan pelatihan di Langkah 2, Anda perlu menyediakan `requirements.txt` untuk menginstal `pytorch-lightning` dan `lightning-bolts` dalam wadah SageMaker PyTorch pelatihan.

```
# requirements.txt
pytorch-lightning
lightning-bolts
```

Untuk informasi selengkapnya tentang menentukan direktori sumber untuk menempatkan `requirements.txt` file bersama dengan skrip pelatihan dan pengiriman pekerjaan, lihat Menggunakan pustaka pihak ketiga dalam dokumentasi Amazon Python SageMaker SDK.

Trafo Hugging Face

AWSDeep Learning Containers untuk Hugging Face menggunakan SageMaker Wadah Pelatihan PyTorch untuk TensorFlow dan sebagai gambar dasarnya. Untuk mencari versi pustaka Hugging Face Transformers dan PyTorch dipasangkan TensorFlow dan versi, lihat Wadah Wajah Pelukan terbaru dan Versi Wadah Wajah Pelukan Sebelumnya.

TensorFlow (usang)

Important

Pustaka SMDDP menghentikan dukungan untuk TensorFlow dan tidak lagi tersedia di DLC selambat-lambatnya v2.11.0. TensorFlow Tabel berikut mencantumkan DLC sebelumnya TensorFlow dengan pustaka SMDDP diinstal.

TensorFlow versi	SageMaker versi paralelisme data terdistribusi
2.9.1, 2.10.1, 2.11.0	`smdistributed-dataparallel==v1.4.1`
2.8.3	`smdistributed-dataparallel==v1.3.0`

Wilayah AWS

Paralelisme data SageMaker terdistribusi tersedia di semua Wilayah AWS tempat AWSDeep Learning Containers SageMaker berada dalam layanan. Untuk informasi selengkapnya, lihat Gambar Deep Learning Containers yang Tersedia.

Tipe instans yang didukung

Paralelisme data SageMaker terdistribusi membutuhkan salah satu jenis contoh berikut.

Jenis instans		
`ml.p4d.24xlarge`		
`ml.p4de.24xlarge`		

> **Important**
>
> Pustaka SMDDP menghentikan dukungan untuk instance P3. Pustaka SMDDP mendukung jenis instans yang dilengkapi dengan GPU NVIDIA A100 dan EFA.

Untuk spesifikasi jenis instans, lihat bagian Komputasi Akselerasi di halaman Jenis Instans Amazon EC2. Untuk informasi tentang harga instans, lihat SageMaker Harga Amazon.

Jika Anda menemukan pesan galat yang mirip dengan berikut ini, ikuti petunjuk di Minta peningkatan kuota layanan untuk SageMaker sumber daya.

```
ResourceLimitExceeded: An error occurred (ResourceLimitExceeded) when calling
the CreateTrainingJob operation: The account-level service limit 'ml.p3dn.24xlarge
for training job usage' is 0 Instances, with current utilization of 0 Instances
and a request delta of 1 Instances.
Please contact AWS support to request an increase for this limit.
```

Cara menjalankan pekerjaan pelatihan terdistribusi dengan pustaka paralelisme data SageMaker terdistribusi

Pustaka paralelisme data SageMaker terdistribusi (SMDDP) dirancang untuk kemudahan penggunaan dan untuk memberikan integrasi yang mulus dengan. PyTorch

Saat melatih model pembelajaran mendalam dengan perpustakaan SMDDP SageMaker, Anda dapat fokus pada penulisan skrip pelatihan dan pelatihan model Anda.

Untuk memulai, impor perpustakaan SMDDP untuk menggunakan operasi kolektifnya yang dioptimalkan. AWS Topik berikut memberikan instruksi tentang apa yang harus ditambahkan ke skrip pelatihan Anda tergantung pada operasi kolektif mana yang ingin Anda optimalkan.

Topik

Langkah 1: Sesuaikan skrip pelatihan Anda untuk menggunakan operasi kolektif SMDDP

Contoh skrip pelatihan yang disediakan di bagian ini disederhanakan dan hanya menyoroti perubahan yang diperlukan untuk mengaktifkan perpustakaan paralelisme data SageMaker terdistribusi (SMDDP) dalam skrip pelatihan Anda. Untuk contoh end-to-end notebook Jupyter yang menunjukkan cara menjalankan pekerjaan pelatihan terdistribusi dengan pustaka SMDDP, lihat. Contoh Notebook Pelatihan SageMaker Terdistribusi Amazon

Topik

Gunakan perpustakaan SMDDP dalam skrip pelatihan Anda PyTorch

Mulai dari perpustakaan paralelisme data SageMaker terdistribusi (SMDDP) v1.4.0, Anda dapat menggunakan perpustakaan sebagai opsi backend untuk paket terdistribusi. PyTorch Untuk menggunakan SMDDP `AllReduce` dan operasi `AllGather` kolektif, Anda hanya perlu mengimpor perpustakaan SMDDP di awal skrip pelatihan Anda dan menetapkan SMDDP sebagai backend modul terdistribusi selama inisialisasi grup proses. PyTorch Dengan satu baris spesifikasi backend, Anda dapat menyimpan semua modul PyTorch terdistribusi asli dan seluruh skrip pelatihan tidak berubah. Cuplikan kode berikut menunjukkan cara menggunakan pustaka SMDDP sebagai backend paket pelatihan terdistribusi PyTorch berbasis: distributed PyTorch data parallel (DDP), PyTorch full sharded data parallelism (FSDP), dan Megatron-. DeepSpeedDeepSpeed

Untuk PyTorch DDP atau FSDP

Inisialisasi kelompok proses sebagai berikut.

```
import torch.distributed as dist
import smdistributed.dataparallel.torch.torch_smddp

dist.init_process_group(backend="smddp")
```

> **Note**
>
> (Hanya untuk pekerjaan PyTorch DDP) `smddp` Backend saat ini tidak mendukung pembuatan grup subproses dengan API. `torch.distributed.new_group()` Anda juga tidak dapat menggunakan `smddp` backend secara bersamaan dengan backend grup proses lainnya seperti dan. `NCCL` `Gloo`

Untuk DeepSpeed atau Megatron- DeepSpeed

Inisialisasi kelompok proses sebagai berikut.

```
import deepspeed
import smdistributed.dataparallel.torch.torch_smddp

deepspeed.init_distributed(dist_backend="smddp")
```

> **Note**
>
> Untuk menggunakan SMDDP `AllGather` dengan peluncur `mpirun` berbasis (`smdistributeddanpytorchddp`) dithe section called “Langkah 2: Luncurkan pekerjaan pelatihan terdistribusi”, Anda juga perlu mengatur variabel lingkungan berikut dalam skrip pelatihan Anda.
>
> ```
> export SMDATAPARALLEL_OPTIMIZE_SDP=true
> ```

Untuk panduan umum tentang menulis skrip pelatihan PyTorch FSDP, lihat Pelatihan Model Lanjutan dengan Paralel Data Berbagi Penuh (FSDP) dalam dokumentasi. PyTorch

Untuk panduan umum tentang menulis skrip pelatihan PyTorch DDP, lihat Memulai dengan data terdistribusi paralel dalam PyTorch dokumentasi.

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan keLangkah 2: Luncurkan pekerjaan pelatihan terdistribusi menggunakan SageMaker Python SDK.

Gunakan perpustakaan SMDDP dalam skrip pelatihan PyTorch Lightning Anda

Jika Anda ingin membawa skrip pelatihan PyTorchLightning dan menjalankan pekerjaan pelatihan paralel data terdistribusi SageMaker, Anda dapat menjalankan pekerjaan pelatihan dengan sedikit perubahan dalam skrip pelatihan Anda. Perubahan yang diperlukan meliputi: mengimpor PyTorch modul `smdistributed.dataparallel` perpustakaan, mengatur variabel lingkungan untuk PyTorch Lightning untuk menerima variabel SageMaker lingkungan yang telah ditetapkan oleh toolkit SageMaker pelatihan, dan mengaktifkan perpustakaan SMDDP dengan menyetel backend grup proses ke. `"smddp"` Untuk mempelajari lebih lanjut, ikuti instruksi berikut yang memecah langkah-langkah dengan contoh kode.

> **Note**
>
> Dukungan PyTorch Lightning tersedia di perpustakaan paralel SageMaker data v1.5.0 dan yang lebih baru.

PyTorch Petir == v2.1.0 dan == 2.0.1 PyTorch

1. Impor `pytorch_lightning` perpustakaan dan `smdistributed.dataparallel.torch` modul.

```
import lightning as pl
import smdistributed.dataparallel.torch.torch_smddp
```

2. Instantiate. LightningEnvironment

```
from lightning.fabric.plugins.environments.lightning import LightningEnvironment

env = LightningEnvironment()
env.world_size = lambda: int(os.environ["WORLD_SIZE"])
env.global_rank = lambda: int(os.environ["RANK"])
```

3. Untuk PyTorch DDP - Buat objek kelas DDPStrategy dengan `"smddp"` for `process_group_backend` dan `"gpu"` for`accelerator`, dan berikan itu ke kelas Trainer.

```
import lightning as pl
from lightning.pytorch.strategies import DDPStrategy

ddp = DDPStrategy(
    cluster_environment=env,
    process_group_backend="smddp",
    accelerator="gpu"
)

trainer = pl.Trainer(
    max_epochs=200,
    strategy=ddp,
    devices=num_gpus,
    num_nodes=num_nodes
)
```

Untuk PyTorch FSDP — Buat objek kelas FSDPStrategy (dengan kebijakan pembungkus pilihan) dengan "smddp" for process_group_backend dan "gpu" foraccelerator, dan teruskan ke kelas Trainer.

```
import lightning as pl
from lightning.pytorch.strategies import FSDPStrategy

from functools import partial
from torch.distributed.fsdp.wrap import size_based_auto_wrap_policy

policy = partial(
    size_based_auto_wrap_policy,
    min_num_params=10000
)

fsdp = FSDPStrategy(
    auto_wrap_policy=policy,
    process_group_backend="smddp",
    cluster_environment=env
)

trainer = pl.Trainer(
    max_epochs=200,
    strategy=fsdp,
    devices=num_gpus,
```

```
    num_nodes=num_nodes
)
```

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan keLangkah 2: Luncurkan pekerjaan pelatihan terdistribusi menggunakan SageMaker Python SDK.

> **Note**
>
> Ketika Anda membangun SageMaker PyTorch estimator dan mengajukan permintaan pekerjaan pelatihan dithe section called “Langkah 2: Luncurkan pekerjaan pelatihan terdistribusi”, Anda perlu menyediakan `requirements.txt` untuk menginstal `pytorch-lightning` dan `lightning-bolts` dalam wadah SageMaker PyTorch pelatihan.
>
> ```
> # requirements.txt
> pytorch-lightning
> lightning-bolts
> ```
>
> Untuk informasi selengkapnya tentang menentukan direktori sumber untuk menempatkan `requirements.txt` file bersama dengan skrip pelatihan dan pengiriman pekerjaan, lihat Menggunakan pustaka pihak ketiga dalam dokumentasi Amazon Python SageMaker SDK.

Gunakan pustaka SMDDP dalam skrip TensorFlow pelatihan Anda (tidak digunakan lagi)

> **Important**
>
> Pustaka SMDDP menghentikan dukungan untuk TensorFlow dan tidak lagi tersedia di DLC selambat-lambatnya v2.11.0. TensorFlow Untuk menemukan TensorFlow DLC sebelumnya dengan pustaka SMDDP diinstal, lihat. the section called “Kerangka kerja yang didukung”

Langkah-langkah berikut menunjukkan cara memodifikasi skrip TensorFlow pelatihan untuk memanfaatkan SageMaker perpustakaan paralel data terdistribusi.

API pustaka dirancang agar mirip dengan API Horovod. Untuk detail tambahan tentang setiap API yang ditawarkan library TensorFlow, lihat dokumentasi TensorFlow API parallel data SageMaker terdistribusi.

> **Note**
>
> SageMaker distributed data parallel dapat disesuaikan dengan skrip TensorFlow pelatihan yang terdiri dari modul `tf` inti kecuali `tf.keras` modul. SageMaker distributed data parallel tidak mendukung TensorFlow implementasi Keras.

> **Note**
>
> Pustaka paralelisme data SageMaker terdistribusi mendukung Automatic Mixed Precision (AMP) di luar kotak. Tidak diperlukan tindakan tambahan untuk mengaktifkan AMP selain modifikasi tingkat kerangka kerja pada skrip pelatihan Anda. Jika gradien berada di FP16, pustaka paralelisme SageMaker data menjalankan operasinya di FP16. `AllReduce` Untuk informasi selengkapnya tentang penerapan API AMP ke skrip pelatihan Anda, lihat sumber daya berikut:
>
> - Kerangka kerja - TensorFlow dalam dokumentasi NVIDIA Deep Learning Performance
> - Presisi Campuran Otomatis untuk Pembelajaran Mendalam di Dokumen Pengembang NVIDIA
> - TensorFlow API presisi campuran dalam TensorFlowdokumentasi

1. Impor TensorFlow klien perpustakaan dan inisialisasi.

```
import smdistributed.dataparallel.tensorflow as sdp
sdp.init()
```

2. Sematkan setiap GPU ke satu `smdistributed.dataparallel` proses dengan `local_rank` —ini mengacu pada peringkat relatif proses dalam node tertentu. `sdp.tensorflow.local_rank()`API memberi Anda peringkat lokal perangkat. Node pemimpin adalah peringkat 0, dan node pekerja adalah peringkat 1, 2, 3, dan seterusnya. Ini dipanggil dalam blok kode berikut sebagai`sdp.local_rank()`. `set_memory_growth`tidak terkait langsung dengan SageMaker didistribusikan, tetapi harus ditetapkan untuk pelatihan terdistribusi dengan TensorFlow.

```
gpus = tf.config.experimental.list_physical_devices('GPU')
for gpu in gpus:
    tf.config.experimental.set_memory_growth(gpu, True)
```

```
if gpus:
    tf.config.experimental.set_visible_devices(gpus[sdp.local_rank()], 'GPU')
```

3. Skala tingkat pembelajaran dengan jumlah pekerja. `sdp.tensorflow.size()`API memberi Anda jumlah pekerja di cluster. Ini dipanggil dalam blok kode berikut sebagai`sdp.size()`.

```
learning_rate = learning_rate * sdp.size()
```

4. Gunakan perpustakaan `DistributedGradientTape` untuk mengoptimalkan `AllReduce` operasi selama pelatihan. Ini membungkus`tf.GradientTape`.

```
with tf.GradientTape() as tape:
      output = model(input)
      loss_value = loss(label, output)

# SageMaker data parallel: Wrap tf.GradientTape with the library's
 DistributedGradientTape
tape = sdp.DistributedGradientTape(tape)
```

5. Siarkan variabel model awal dari node pemimpin (peringkat 0) ke semua node pekerja (peringkat 1 hingga n). Ini diperlukan untuk memastikan inisialisasi yang konsisten di semua peringkat pekerja. Gunakan `sdp.tensorflow.broadcast_variables` API setelah variabel model dan pengoptimal diinisialisasi. Ini dipanggil dalam blok kode berikut sebagai`sdp.broadcast_variables()`.

```
sdp.broadcast_variables(model.variables, root_rank=0)
sdp.broadcast_variables(opt.variables(), root_rank=0)
```

6. Terakhir, modifikasi skrip Anda untuk menyimpan pos pemeriksaan hanya pada node pemimpin. Node pemimpin memiliki model yang disinkronkan. Ini juga menghindari node pekerja yang menimpa pos pemeriksaan dan mungkin merusak pos pemeriksaan.

```
if sdp.rank() == 0:
    checkpoint.save(checkpoint_dir)
```

Berikut ini adalah contoh skrip TensorFlow pelatihan untuk pelatihan terdistribusi dengan perpustakaan.

```
import tensorflow as tf
```

```
# SageMaker data parallel: Import the library TF API
import smdistributed.dataparallel.tensorflow as sdp

# SageMaker data parallel: Initialize the library
sdp.init()

gpus = tf.config.experimental.list_physical_devices('GPU')
for gpu in gpus:
    tf.config.experimental.set_memory_growth(gpu, True)
if gpus:
    # SageMaker data parallel: Pin GPUs to a single library process
    tf.config.experimental.set_visible_devices(gpus[sdp.local_rank()], 'GPU')

# Prepare Dataset
dataset = tf.data.Dataset.from_tensor_slices(...)

# Define Model
mnist_model = tf.keras.Sequential(...)
loss = tf.losses.SparseCategoricalCrossentropy()

# SageMaker data parallel: Scale Learning Rate
# LR for 8 node run : 0.000125
# LR for single node run : 0.001
opt = tf.optimizers.Adam(0.000125 * sdp.size())

@tf.function
def training_step(images, labels, first_batch):
    with tf.GradientTape() as tape:
        probs = mnist_model(images, training=True)
        loss_value = loss(labels, probs)

    # SageMaker data parallel: Wrap tf.GradientTape with the library's
 DistributedGradientTape
    tape = sdp.DistributedGradientTape(tape)

    grads = tape.gradient(loss_value, mnist_model.trainable_variables)
    opt.apply_gradients(zip(grads, mnist_model.trainable_variables))

    if first_batch:
       # SageMaker data parallel: Broadcast model and optimizer variables
       sdp.broadcast_variables(mnist_model.variables, root_rank=0)
       sdp.broadcast_variables(opt.variables(), root_rank=0)

    return loss_value
```

```
...

# SageMaker data parallel: Save checkpoints only from master node.
if sdp.rank() == 0:
    checkpoint.save(checkpoint_dir)
```

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan keLangkah 2: Luncurkan pekerjaan pelatihan terdistribusi menggunakan SageMaker Python SDK.

Langkah 2: Luncurkan pekerjaan pelatihan terdistribusi menggunakan SageMaker Python SDK

Untuk menjalankan pekerjaan pelatihan terdistribusi dengan skrip adaptasi Anda darithe section called “Langkah 1: Sesuaikan skrip pelatihan Anda untuk menggunakan operasi kolektif SMDDP”, gunakan kerangka kerja SageMaker Python SDK atau estimator generik dengan menentukan skrip pelatihan yang disiapkan sebagai skrip titik masuk dan konfigurasi pelatihan terdistribusi.

Halaman ini memandu Anda melalui cara menggunakan SageMaker Python SDK dalam dua cara.

- Jika Anda ingin mencapai adopsi cepat dari pekerjaan pelatihan terdistribusi Anda di SageMaker, konfigurasikan kelas estimator SageMaker PyTorchatau TensorFlowkerangka kerja. Estimator framework mengambil skrip latihan Anda dan secara otomatis mencocokkan URI gambar yang tepat dari Deep Learning Containers (DLC) yang sudah dibuat sebelumnya PyTorch atau TensorFlow Deep Learning Containers (DLC), dengan nilai yang ditentukan pada parameter. `framework_version`
- Jika Anda ingin memperluas salah satu container yang sudah dibuat sebelumnya atau membuat container kustom untuk membuat lingkungan MLmu sendiri SageMaker, gunakan `Estimator` class SageMaker generik dan tentukan URI image dari container Docker kustom yang dihosting di Amazon Elastic Container Registry (Amazon ECR).

Kumpulan data pelatihan Anda harus disimpan di Amazon S3 atau Amazon FSx for Lustre Wilayah AWS di mana Anda meluncurkan pekerjaan pelatihan Anda. Jika Anda menggunakan notebook Jupyter, Anda harus memiliki instance SageMaker notebook atau aplikasi SageMaker Studio Classic yang berjalan dalam hal yang sama. Wilayah AWS Untuk informasi selengkapnya tentang menyimpan data latihan, lihat dokumentasi SageMaker input data SDK Python.

> **Tip**
>
> Kami sangat menyarankan Anda menggunakan Amazon FSx for Lustre alih-alih Amazon S3 untuk meningkatkan kinerja pelatihan. Amazon FSx memiliki throughput yang lebih tinggi dan latensi yang lebih rendah daripada Amazon S3.

Pilih salah satu topik berikut untuk instruksi tentang cara menjalankan pekerjaan pelatihan terdistribusi dari skrip pelatihan Anda. Setelah Anda meluncurkan pekerjaan pelatihan, Anda dapat memantau pemanfaatan sistem dan kinerja model menggunakan Gunakan Amazon SageMaker Debugger untuk men-debug dan meningkatkan kinerja model atau Amazon CloudWatch.

Saat Anda mengikuti petunjuk dalam topik berikut untuk mempelajari lebih lanjut tentang detail teknis, kami juga menyarankan Anda mencoba Contoh Notebook Pelatihan SageMaker Terdistribusi Amazon untuk memulai.

Topik

- Menggunakan estimator kerangka kerja di SageMaker Python SDK
- Menggunakan estimator SageMaker generik untuk memperluas wadah bawaan
- Buat wadah Docker Anda sendiri dengan pustaka paralel data SageMaker terdistribusi

Menggunakan estimator kerangka kerja di SageMaker Python SDK

Anda dapat meluncurkan pelatihan terdistribusi dengan menambahkan `distribution` argumen ke estimator SageMaker kerangka kerja, `PyTorch`atau `TensorFlow`. Untuk detail selengkapnya, pilih salah satu kerangka kerja yang didukung oleh pustaka paralelisme data SageMaker terdistribusi (SMDDP) dari pilihan berikut.

PyTorch

Opsi peluncur berikut tersedia untuk meluncurkan pelatihan PyTorch terdistribusi.

- `pytorchddp`— Opsi ini menjalankan `mpirun` dan mengatur variabel lingkungan yang diperlukan untuk menjalankan pelatihan PyTorch terdistribusi SageMaker. Untuk menggunakan opsi ini, teruskan kamus berikut ke `distribution` parameter.

```
{ "pytorchddp": { "enabled": True } }
```

- `torch_distributed`— Opsi ini menjalankan `torchrun` dan mengatur variabel lingkungan yang diperlukan untuk menjalankan pelatihan PyTorch terdistribusi SageMaker. Untuk menggunakan opsi ini, teruskan kamus berikut ke `distribution` parameter.

```
{ "torch_distributed": { "enabled": True } }
```

- `smdistributed`— Opsi ini juga berjalan `mpirun` tetapi dengan `smddprun` itu mengatur variabel lingkungan yang diperlukan untuk menjalankan pelatihan PyTorch terdistribusi SageMaker.

```
{ "smdistributed": { "dataparallel": { "enabled": True } } }
```

Jika Anda memilih untuk mengganti NCCL `AllGather` ke SMDDP`AllGather`, Anda dapat menggunakan ketiga opsi. Pilih satu opsi yang sesuai dengan kasus penggunaan Anda.

Jika Anda memilih untuk mengganti NCCL `AllReduce` dengan SMDDP`AllReduce`, Anda harus memilih salah satu opsi `mpirun` berbasis: atau. `smdistributed` `pytorchddp` Anda juga dapat menambahkan opsi MPI tambahan sebagai berikut.

```
{
    "pytorchddp": {
        "enabled": True,
        "custom_mpi_options": "-verbose -x NCCL_DEBUG=VERSION"
    }
}
```

```
{
    "smdistributed": {
        "dataparallel": {
            "enabled": True,
            "custom_mpi_options": "-verbose -x NCCL_DEBUG=VERSION"
        }
    }
}
```

Contoh kode berikut menunjukkan struktur dasar PyTorch estimator dengan opsi pelatihan terdistribusi.

```
from sagemaker.pytorch import PyTorch
```

```
pt_estimator = PyTorch(
    base_job_name="training_job_name_prefix",
    source_dir="subdirectory-to-your-code",
    entry_point="adapted-training-script.py",
    role="SageMakerRole",
    py_version="py310",
    framework_version="2.0.1",

    # For running a multi-node distributed training job, specify a value greater
 than 1
    # Example: 2,3,4,..8
    instance_count=2,

    # Instance types supported by the SageMaker data parallel library:
    # ml.p4d.24xlarge, ml.p4de.24xlarge
    instance_type="ml.p4d.24xlarge",

    # Activate distributed training with SMDDP
    distribution={ "pytorchddp": { "enabled": True } }  # mpirun, activates SMDDP
 AllReduce OR AllGather
    # distribution={ "torch_distributed": { "enabled": True } }  # torchrun,
 activates SMDDP AllGather
    # distribution={ "smdistributed": { "dataparallel": { "enabled": True } } }  #
 mpirun, activates SMDDP AllReduce OR AllGather
)

pt_estimator.fit("s3://bucket/path/to/training/data")
```

Note

PyTorch Lightning dan perpustakaan utilitasnya seperti Lightning Bolts tidak diinstal sebelumnya di DLC. SageMaker PyTorch Buat `requirements.txt` file berikut dan simpan di direktori sumber tempat Anda menyimpan skrip pelatihan.

```
# requirements.txt
pytorch-lightning
lightning-bolts
```

Misalnya, direktori terstruktur pohon akan terlihat seperti berikut ini.

```
### pytorch_training_launcher_jupyter_notebook.ipynb
```

```
### sub-folder-for-your-code
    ###  adapted-training-script.py
    ###  requirements.txt
```

Untuk informasi selengkapnya tentang menentukan direktori sumber untuk menempatkan `requirements.txt` file bersama dengan skrip pelatihan dan pengiriman pekerjaan, lihat Menggunakan pustaka pihak ketiga dalam dokumentasi Amazon Python SageMaker SDK.

Pertimbangan untuk mengaktifkan operasi kolektif SMDDP dan menggunakan opsi peluncur pelatihan terdistribusi yang tepat

- SMDDP `AllReduce` dan SMDDP tidak saling `AllGather` kompatibel saat ini.
- SMDDP `AllReduce` diaktifkan secara default saat menggunakan `smdistributed` ataupytorchddp, yang `mpirun` berbasis peluncur, dan NCCL digunakan. `AllGather`
- SMDDP `AllGather` diaktifkan secara default saat menggunakan `torch_distributed` peluncur, dan kembali ke `AllReduce` NCCL.
- SMDDP juga `AllGather` dapat diaktifkan saat menggunakan peluncur `mpirun` berbasis dengan variabel lingkungan tambahan yang ditetapkan sebagai berikut.

```
export SMDATAPARALLEL_OPTIMIZE_SDP=true
```

TensorFlow

Important

Pustaka SMDDP menghentikan dukungan untuk TensorFlow dan tidak lagi tersedia di DLC selambat-lambatnya v2.11.0. TensorFlow Untuk menemukan TensorFlow DLC sebelumnya dengan pustaka SMDDP diinstal, lihat. the section called “TensorFlow (usang)”

```
from sagemaker.tensorflow import TensorFlow

tf_estimator = TensorFlow(
    base_job_name = "training_job_name_prefix",
```

```
    entry_point="adapted-training-script.py",
    role="SageMakerRole",
    framework_version="2.11.0",
    py_version="py38",

    # For running a multi-node distributed training job, specify a value greater
 than 1
    # Example: 2,3,4,..8
    instance_count=2,

    # Instance types supported by the SageMaker data parallel library:
    # ml.p4d.24xlarge, ml.p3dn.24xlarge, and ml.p3.16xlarge
    instance_type="ml.p3.16xlarge",

    # Training using the SageMaker data parallel distributed training strategy
    distribution={ "smdistributed": { "dataparallel": { "enabled": True } } }
)

tf_estimator.fit("s3://bucket/path/to/training/data")
```

Menggunakan estimator SageMaker generik untuk memperluas wadah bawaan

Anda dapat menyesuaikan kontainer SageMaker bawaan atau memperluasnya untuk menangani persyaratan fungsional tambahan apa pun untuk algoritme atau model Anda yang tidak didukung oleh image SageMaker Docker bawaan. Untuk contoh bagaimana Anda dapat memperluas kontainer yang sudah dibuat sebelumnya, lihat Memperluas Kontainer Prebuilt.

Untuk memperluas wadah bawaan atau menyesuaikan wadah Anda sendiri untuk menggunakan pustaka, Anda harus menggunakan salah satu gambar yang tercantum di dalamnyaKerangka kerja yang didukung.

> **Important**
>
> Dari TensorFlow 2.4.1 dan PyTorch 1.8.1, kerangka DLC mendukung tipe instans berkemampuan EFA (,). `ml.p3dn.24xlarge` `ml.p4d.24xlarge` Kami menyarankan Anda menggunakan gambar DLC yang berisi TensorFlow 2.4.1 atau yang lebih baru dan PyTorch 1.8.1 atau yang lebih baru.

Misalnya, jika Anda menggunakan PyTorch, Dockerfile Anda harus berisi FROM pernyataan yang mirip dengan berikut ini:

```
# SageMaker PyTorch image
FROM 763104351884.dkr.ecr.<aws-region>.amazonaws.com/pytorch-training:<image-tag>

ENV PATH="/opt/ml/code:${PATH}"

# this environment variable is used by the SageMaker PyTorch container to determine our
 user code directory.
ENV SAGEMAKER_SUBMIT_DIRECTORY /opt/ml/code

# /opt/ml and all subdirectories are utilized by SageMaker, use the /code subdirectory
 to store your user code.
COPY train.py /opt/ml/code/train.py

# Defines cifar10.py as script entrypoint
ENV SAGEMAKER_PROGRAM train.py
```

Anda dapat lebih lanjut menyesuaikan wadah Docker Anda sendiri untuk bekerja dengan SageMaker menggunakan toolkit SageMaker Pelatihan dan file biner dari perpustakaan paralel data SageMaker terdistribusi. Untuk mempelajari lebih lanjut, lihat petunjuk di bagian berikut.

Buat wadah Docker Anda sendiri dengan pustaka paralel data SageMaker terdistribusi

Untuk membangun wadah Docker Anda sendiri untuk pelatihan dan menggunakan pustaka paralel SageMaker data, Anda harus menyertakan dependensi yang benar dan file biner dari pustaka SageMaker paralel terdistribusi di Dockerfile Anda. Bagian ini memberikan petunjuk tentang cara membuat Dockerfile lengkap dengan kumpulan dependensi minimum untuk pelatihan terdistribusi dalam SageMaker menggunakan pustaka paralel data.

> **Note**
>
> Opsi Docker khusus ini dengan pustaka paralel SageMaker data sebagai biner hanya tersedia untuk PyTorch.

Untuk membuat Dockerfile dengan toolkit SageMaker pelatihan dan pustaka paralel data

1. Mulailah dengan gambar Docker dari NVIDIA CUDA. Gunakan versi pengembang cuDNN yang berisi runtime CUDA dan alat pengembangan (header dan pustaka) untuk membangun dari kode sumber. PyTorch

   ```
   FROM nvidia/cuda:11.3.1-cudnn8-devel-ubuntu20.04
   ```

   > **Tip**
   >
   > Gambar AWS Deep Learning Container (DLC) resmi dibuat dari gambar dasar NVIDIA CUDA. Jika Anda ingin menggunakan gambar DLC bawaan sebagai referensi sambil mengikuti instruksi lainnya, lihat AWSDeep Learning Containers untuk Dockerfiles. PyTorch

2. Tambahkan argumen berikut untuk menentukan versi PyTorch dan paket lainnya. Juga, tunjukkan jalur bucket Amazon S3 ke pustaka SageMaker paralel data dan perangkat lunak lain untuk menggunakan AWS sumber daya, seperti plug-in Amazon S3.

   Untuk menggunakan versi pustaka pihak ketiga selain yang disediakan dalam contoh kode berikut, kami sarankan Anda melihat ke Dockerfiles resmi AWS Deep Learning Container PyTorch untuk menemukan versi yang diuji, kompatibel, dan cocok untuk aplikasi Anda.

   Untuk menemukan URL `SMDATAPARALLEL_BINARY` argumen, lihat tabel pencarian di. Kerangka kerja yang didukung

   ```
   ARG PYTORCH_VERSION=1.10.2
   ARG PYTHON_SHORT_VERSION=3.8
   ARG EFA_VERSION=1.14.1
   ARG SMDATAPARALLEL_BINARY=https://smdataparallel.s3.amazonaws.com/binary/pytorch/
   ${PYTORCH_VERSION}/cu113/2022-02-18/smdistributed_dataparallel-1.4.0-cp38-cp38-
   linux_x86_64.whl
   ARG PT_S3_WHL_GPU=https://aws-s3-plugin.s3.us-west-2.amazonaws.com/
   binaries/0.0.1/1c3e69e/awsio-0.0.1-cp38-cp38-manylinux1_x86_64.whl
   ARG CONDA_PREFIX="/opt/conda"
   ARG BRANCH_OFI=1.1.3-aws
   ```

3. Atur variabel lingkungan berikut untuk membangun komponen SageMaker pelatihan dengan benar dan menjalankan pustaka paralel data. Anda menggunakan variabel ini untuk komponen dalam langkah selanjutnya.

```
# Set ENV variables required to build PyTorch
ENV TORCH_CUDA_ARCH_LIST="7.0+PTX 8.0"
ENV TORCH_NVCC_FLAGS="-Xfatbin -compress-all"
ENV NCCL_VERSION=2.10.3

# Add OpenMPI to the path.
ENV PATH /opt/amazon/openmpi/bin:$PATH

# Add Conda to path
ENV PATH $CONDA_PREFIX/bin:$PATH

# Set this enviroment variable for SageMaker to launch SMDDP correctly.
ENV SAGEMAKER_TRAINING_MODULE=sagemaker_pytorch_container.training:main

# Add enviroment variable for processes to be able to call fork()
ENV RDMAV_FORK_SAFE=1

# Indicate the container type
ENV DLC_CONTAINER_TYPE=training

# Add EFA and SMDDP to LD library path
ENV LD_LIBRARY_PATH="/opt/conda/lib/python${PYTHON_SHORT_VERSION}/site-packages/
smdistributed/dataparallel/lib:$LD_LIBRARY_PATH"
ENV LD_LIBRARY_PATH=/opt/amazon/efa/lib/:$LD_LIBRARY_PATH
```

4. Instal atau perbarui `curl`wget,, dan `git` unduh dan buat paket di langkah selanjutnya.

```
RUN --mount=type=cache,id=apt-final,target=/var/cache/apt \
    apt-get update && apt-get install -y  --no-install-recommends \
        curl \
        wget \
        git \
    && rm -rf /var/lib/apt/lists/*
```

5. Instal perangkat lunak Elastic Fabric Adapter (EFA) untuk komunikasi jaringan Amazon EC2.

```
RUN DEBIAN_FRONTEND=noninteractive apt-get update
RUN mkdir /tmp/efa \
    && cd /tmp/efa \
    && curl --silent -O https://efa-installer.amazonaws.com/aws-efa-installer-
${EFA_VERSION}.tar.gz \
    && tar -xf aws-efa-installer-${EFA_VERSION}.tar.gz \
```

```
    && cd aws-efa-installer \
    && ./efa_installer.sh -y --skip-kmod -g \
    && rm -rf /tmp/efa
```

6. Instal Conda untuk menangani manajemen paket.

```
RUN curl -fsSL -v -o ~/miniconda.sh -O  https://repo.anaconda.com/miniconda/
Miniconda3-latest-Linux-x86_64.sh  && \
    chmod +x ~/miniconda.sh && \
    ~/miniconda.sh -b -p $CONDA_PREFIX && \
    rm ~/miniconda.sh && \
    $CONDA_PREFIX/bin/conda install -y python=${PYTHON_SHORT_VERSION} conda-build
 pyyaml numpy ipython && \
    $CONDA_PREFIX/bin/conda clean -ya
```

7. Dapatkan, bangun, PyTorch dan instal serta dependensinya. Kami membangun PyTorch dari kode sumber karena kami perlu memiliki kontrol versi NCCL untuk menjamin kompatibilitas dengan plug-in AWSOFI NCCL.

   a. Mengikuti langkah-langkah di dockerfile PyTorch resmi, instal dependensi build dan atur ccache untuk mempercepat kompilasi ulang.

```
RUN DEBIAN_FRONTEND=noninteractive \
    apt-get install -y --no-install-recommends \
        build-essential \
        ca-certificates \
        ccache \
        cmake \
        git \
        libjpeg-dev \
        libpng-dev \
    && rm -rf /var/lib/apt/lists/*

# Setup ccache
RUN /usr/sbin/update-ccache-symlinks
RUN mkdir /opt/ccache && ccache --set-config=cache_dir=/opt/ccache
```

   b. Ketergantungan umum dan Linux InstallPyTorch.

```
# Common dependencies for PyTorch
RUN conda install astunparse numpy ninja pyyaml mkl mkl-include setuptools cmake
 cffi typing_extensions future six requests dataclasses
```

```
# Linux specific dependency for PyTorch
RUN conda install -c pytorch magma-cuda113
```

c. Kloning PyTorch GitHubrepositori.

```
RUN --mount=type=cache,target=/opt/ccache \
    cd / \
    && git clone --recursive https://github.com/pytorch/pytorch -b v
${PYTORCH_VERSION}
```

d. Instal dan buat versi NCCL tertentu. Untuk melakukan ini, ganti konten di folder NCCL default (`/pytorch/third_party/nccl`) dengan versi NCCL tertentu dari repositori NVIDIA. PyTorch Versi NCCL ditetapkan pada langkah 3 panduan ini.

```
RUN cd /pytorch/third_party/nccl \
    && rm -rf nccl \
    && git clone https://github.com/NVIDIA/nccl.git -b v${NCCL_VERSION}-1 \
    && cd nccl \
    && make -j64 src.build CUDA_HOME=/usr/local/cuda NVCC_GENCODE="-
gencode=arch=compute_70,code=sm_70 -gencode=arch=compute_80,code=sm_80" \
    && make pkg.txz.build \
    && tar -xvf build/pkg/txz/nccl_*.txz -C $CONDA_PREFIX --strip-components=1
```

e. Membangun dan menginstal PyTorch. Proses ini biasanya memakan waktu sedikit lebih dari 1 jam untuk menyelesaikannya. Itu dibangun menggunakan versi NCCL yang diunduh pada langkah sebelumnya.

```
RUN cd /pytorch \
    && CMAKE_PREFIX_PATH="$(dirname $(which conda))/../" \
    python setup.py install \
    && rm -rf /pytorch
```

8. Membangun dan menginstal AWSplugin OFI NCCL. Ini memungkinkan dukungan libfabric untuk pustaka paralel SageMaker data.

```
RUN DEBIAN_FRONTEND=noninteractive apt-get update \
    && apt-get install -y --no-install-recommends \
        autoconf \
        automake \
        libtool
RUN mkdir /tmp/efa-ofi-nccl \
    && cd /tmp/efa-ofi-nccl \
```

```
    && git clone https://github.com/aws/aws-ofi-nccl.git -b v${BRANCH_OFI} \
    && cd aws-ofi-nccl \
    && ./autogen.sh \
    && ./configure --with-libfabric=/opt/amazon/efa \
    --with-mpi=/opt/amazon/openmpi \
    --with-cuda=/usr/local/cuda \
    --with-nccl=$CONDA_PREFIX \
    && make \
    && make install \
    && rm -rf /tmp/efa-ofi-nccl
```

9. Membangun dan menginstal TorchVision.

```
RUN pip install --no-cache-dir -U \
    packaging \
    mpi4py==3.0.3
RUN cd /tmp \
    && git clone https://github.com/pytorch/vision.git -b v0.9.1 \
    && cd vision \
    && BUILD_VERSION="0.9.1+cu111" python setup.py install \
    && cd /tmp \
    && rm -rf vision
```

10. Instal dan konfigurasikan OpenSSH. OpenSSH diperlukan agar MPI dapat berkomunikasi antar kontainer. Izinkan OpenSSH untuk berbicara dengan kontainer tanpa meminta konfirmasi.

```
RUN apt-get update \
    && apt-get install -y  --allow-downgrades --allow-change-held-packages --no-
install-recommends \
    && apt-get install -y --no-install-recommends openssh-client openssh-server \
    && mkdir -p /var/run/sshd \
    && cat /etc/ssh/ssh_config | grep -v StrictHostKeyChecking > /etc/ssh/
ssh_config.new \
    && echo "    StrictHostKeyChecking no" >> /etc/ssh/ssh_config.new \
    && mv /etc/ssh/ssh_config.new /etc/ssh/ssh_config \
    && rm -rf /var/lib/apt/lists/*

# Configure OpenSSH so that nodes can communicate with each other
RUN mkdir -p /var/run/sshd && \
 sed 's@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g' -i /
etc/pam.d/sshd
RUN rm -rf /root/.ssh/ && \
 mkdir -p /root/.ssh/ && \
```

```
 ssh-keygen -q -t rsa -N '' -f /root/.ssh/id_rsa && \
 cp /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys \
 && printf "Host *\n StrictHostKeyChecking no\n" >> /root/.ssh/config
```

11. Instal plug-in PT S3 untuk mengakses kumpulan data secara efisien di Amazon S3.

```
RUN pip install --no-cache-dir -U ${PT_S3_WHL_GPU}
RUN mkdir -p /etc/pki/tls/certs && cp /etc/ssl/certs/ca-certificates.crt /etc/pki/
tls/certs/ca-bundle.crt
```

12. Instal perpustakaan libboost. Paket ini diperlukan untuk jaringan fungsi IO asinkron dari pustaka SageMaker paralel data.

```
WORKDIR /
RUN wget https://sourceforge.net/projects/boost/files/boost/1.73.0/
boost_1_73_0.tar.gz/download -O boost_1_73_0.tar.gz \
    && tar -xzf boost_1_73_0.tar.gz \
    && cd boost_1_73_0 \
    && ./bootstrap.sh \
    && ./b2 threading=multi --prefix=${CONDA_PREFIX} -j 64 cxxflags=-fPIC cflags=-
fPIC install || true \
    && cd .. \
    && rm -rf boost_1_73_0.tar.gz \
    && rm -rf boost_1_73_0 \
    && cd ${CONDA_PREFIX}/include/boost
```

13. Instal SageMaker alat-alat berikut untuk PyTorch pelatihan.

```
WORKDIR /root
RUN pip install --no-cache-dir -U \
    smclarify \
    "sagemaker>=2,<3" \
    sagemaker-experiments==0.* \
    sagemaker-pytorch-training
```

14. Terakhir, instal SageMaker data parallel binary dan dependensi yang tersisa.

```
RUN --mount=type=cache,id=apt-final,target=/var/cache/apt \
  apt-get update && apt-get install -y  --no-install-recommends \
  jq \
  libhwloc-dev \
  libnuma1 \
  libnuma-dev \
```

```
  libssl1.1 \
  libtool \
  hwloc \
  && rm -rf /var/lib/apt/lists/*

RUN SMDATAPARALLEL_PT=1 pip install --no-cache-dir ${SMDATAPARALLEL_BINARY}
```

15. Setelah Anda selesai membuat Dockerfile, lihat Mengadaptasi Wadah Pelatihan Anda Sendiri untuk mempelajari cara membuat container Docker, menghostingnya di Amazon ECR, dan menjalankan tugas pelatihan menggunakan Python SDK. SageMaker

Kode contoh berikut menunjukkan Dockerfile lengkap setelah menggabungkan semua blok kode sebelumnya.

```
# This file creates a docker image with minimum dependencies to run SageMaker data
 parallel training
FROM nvidia/cuda:11.3.1-cudnn8-devel-ubuntu20.04

# Set appropiate versions and location for components
ARG PYTORCH_VERSION=1.10.2
ARG PYTHON_SHORT_VERSION=3.8
ARG EFA_VERSION=1.14.1
ARG SMDATAPARALLEL_BINARY=https://smdataparallel.s3.amazonaws.com/binary/pytorch/
${PYTORCH_VERSION}/cu113/2022-02-18/smdistributed_dataparallel-1.4.0-cp38-cp38-
linux_x86_64.whl
ARG PT_S3_WHL_GPU=https://aws-s3-plugin.s3.us-west-2.amazonaws.com/
binaries/0.0.1/1c3e69e/awsio-0.0.1-cp38-cp38-manylinux1_x86_64.whl
ARG CONDA_PREFIX="/opt/conda"
ARG BRANCH_OFI=1.1.3-aws

# Set ENV variables required to build PyTorch
ENV TORCH_CUDA_ARCH_LIST="3.7 5.0 7.0+PTX 8.0"
ENV TORCH_NVCC_FLAGS="-Xfatbin -compress-all"
ENV NCCL_VERSION=2.10.3

# Add OpenMPI to the path.
ENV PATH /opt/amazon/openmpi/bin:$PATH

# Add Conda to path
ENV PATH $CONDA_PREFIX/bin:$PATH

# Set this enviroment variable for SageMaker to launch SMDDP correctly.
```

```
ENV SAGEMAKER_TRAINING_MODULE=sagemaker_pytorch_container.training:main

# Add enviroment variable for processes to be able to call fork()
ENV RDMAV_FORK_SAFE=1

# Indicate the container type
ENV DLC_CONTAINER_TYPE=training

# Add EFA and SMDDP to LD library path
ENV LD_LIBRARY_PATH="/opt/conda/lib/python${PYTHON_SHORT_VERSION}/site-packages/
smdistributed/dataparallel/lib:$LD_LIBRARY_PATH"
ENV LD_LIBRARY_PATH=/opt/amazon/efa/lib/:$LD_LIBRARY_PATH

# Install basic dependencies to download and build other dependencies
RUN --mount=type=cache,id=apt-final,target=/var/cache/apt \
  apt-get update && apt-get install -y  --no-install-recommends \
  curl \
  wget \
  git \
  && rm -rf /var/lib/apt/lists/*

# Install EFA.
# This is required for SMDDP backend communication
RUN DEBIAN_FRONTEND=noninteractive apt-get update
RUN mkdir /tmp/efa \
    && cd /tmp/efa \
    && curl --silent -O https://efa-installer.amazonaws.com/aws-efa-installer-
${EFA_VERSION}.tar.gz \
    && tar -xf aws-efa-installer-${EFA_VERSION}.tar.gz \
    && cd aws-efa-installer \
    && ./efa_installer.sh -y --skip-kmod -g \
    && rm -rf /tmp/efa

# Install Conda
RUN curl -fsSL -v -o ~/miniconda.sh -O  https://repo.anaconda.com/miniconda/Miniconda3-
latest-Linux-x86_64.sh  && \
    chmod +x ~/miniconda.sh && \
    ~/miniconda.sh -b -p $CONDA_PREFIX && \
    rm ~/miniconda.sh && \
    $CONDA_PREFIX/bin/conda install -y python=${PYTHON_SHORT_VERSION} conda-build
 pyyaml numpy ipython && \
    $CONDA_PREFIX/bin/conda clean -ya

# Install PyTorch.
```

```
# Start with dependencies listed in official PyTorch dockerfile
# https://github.com/pytorch/pytorch/blob/master/Dockerfile
RUN DEBIAN_FRONTEND=noninteractive \
    apt-get install -y --no-install-recommends \
        build-essential \
        ca-certificates \
        ccache \
        cmake \
        git \
        libjpeg-dev \
        libpng-dev && \
    rm -rf /var/lib/apt/lists/*

# Setup ccache
RUN /usr/sbin/update-ccache-symlinks
RUN mkdir /opt/ccache && ccache --set-config=cache_dir=/opt/ccache

# Common dependencies for PyTorch
RUN conda install astunparse numpy ninja pyyaml mkl mkl-include setuptools cmake cffi
 typing_extensions future six requests dataclasses

# Linux specific dependency for PyTorch
RUN conda install -c pytorch magma-cuda113

# Clone PyTorch
RUN --mount=type=cache,target=/opt/ccache \
    cd / \
    && git clone --recursive https://github.com/pytorch/pytorch -b v${PYTORCH_VERSION}
# Note that we need to use the same NCCL version for PyTorch and OFI plugin.
# To enforce that, install NCCL from source before building PT and OFI plugin.

# Install NCCL.
# Required for building OFI plugin (OFI requires NCCL's header files and library)
RUN cd /pytorch/third_party/nccl \
    && rm -rf nccl \
    && git clone https://github.com/NVIDIA/nccl.git -b v${NCCL_VERSION}-1 \
    && cd nccl \
    && make -j64 src.build CUDA_HOME=/usr/local/cuda NVCC_GENCODE="-
gencode=arch=compute_70,code=sm_70 -gencode=arch=compute_80,code=sm_80" \
    && make pkg.txz.build \
    && tar -xvf build/pkg/txz/nccl_*.txz -C $CONDA_PREFIX --strip-components=1

# Build and install PyTorch.
RUN cd /pytorch \
```

```
    && CMAKE_PREFIX_PATH="$(dirname $(which conda))/../" \
    python setup.py install \
    && rm -rf /pytorch

RUN ccache -C

# Build and install OFI plugin. \
# It is required to use libfabric.
RUN DEBIAN_FRONTEND=noninteractive apt-get update \
    && apt-get install -y --no-install-recommends \
        autoconf \
        automake \
        libtool
RUN mkdir /tmp/efa-ofi-nccl \
    && cd /tmp/efa-ofi-nccl \
    && git clone https://github.com/aws/aws-ofi-nccl.git -b v${BRANCH_OFI} \
    && cd aws-ofi-nccl \
    && ./autogen.sh \
    && ./configure --with-libfabric=/opt/amazon/efa \
        --with-mpi=/opt/amazon/openmpi \
        --with-cuda=/usr/local/cuda \
        --with-nccl=$CONDA_PREFIX \
    && make \
    && make install \
    && rm -rf /tmp/efa-ofi-nccl

# Build and install Torchvision
RUN pip install --no-cache-dir -U \
    packaging \
    mpi4py==3.0.3
RUN cd /tmp \
    && git clone https://github.com/pytorch/vision.git -b v0.9.1 \
    && cd vision \
    && BUILD_VERSION="0.9.1+cu111" python setup.py install \
    && cd /tmp \
    && rm -rf vision

# Install OpenSSH.
# Required for MPI to communicate between containers, allow OpenSSH to talk to
 containers without asking for confirmation
RUN apt-get update \
    && apt-get install -y  --allow-downgrades --allow-change-held-packages --no-
install-recommends \
    && apt-get install -y --no-install-recommends openssh-client openssh-server \
```

```
    && mkdir -p /var/run/sshd \
    && cat /etc/ssh/ssh_config | grep -v StrictHostKeyChecking > /etc/ssh/
ssh_config.new \
    && echo "    StrictHostKeyChecking no" >> /etc/ssh/ssh_config.new \
    && mv /etc/ssh/ssh_config.new /etc/ssh/ssh_config \
    && rm -rf /var/lib/apt/lists/*
# Configure OpenSSH so that nodes can communicate with each other
RUN mkdir -p /var/run/sshd && \
    sed 's@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g' -
i /etc/pam.d/sshd
RUN rm -rf /root/.ssh/ && \
    mkdir -p /root/.ssh/ && \
    ssh-keygen -q -t rsa -N '' -f /root/.ssh/id_rsa && \
    cp /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys \
    && printf "Host *\n StrictHostKeyChecking no\n" >> /root/.ssh/config

# Install PT S3 plugin.
# Required to efficiently access datasets in Amazon S3
RUN pip install --no-cache-dir -U ${PT_S3_WHL_GPU}
RUN mkdir -p /etc/pki/tls/certs && cp /etc/ssl/certs/ca-certificates.crt /etc/pki/tls/
certs/ca-bundle.crt

# Install libboost from source.
# This package is needed for smdataparallel functionality (for networking asynchronous
 IO).
WORKDIR /
RUN wget https://sourceforge.net/projects/boost/files/boost/1.73.0/boost_1_73_0.tar.gz/
download -O boost_1_73_0.tar.gz \
    && tar -xzf boost_1_73_0.tar.gz \
    && cd boost_1_73_0 \
    && ./bootstrap.sh \
    && ./b2 threading=multi --prefix=${CONDA_PREFIX} -j 64 cxxflags=-fPIC cflags=-fPIC
 install || true \
    && cd .. \
    && rm -rf boost_1_73_0.tar.gz \
    && rm -rf boost_1_73_0 \
    && cd ${CONDA_PREFIX}/include/boost

# Install SageMaker PyTorch training.
WORKDIR /root
RUN pip install --no-cache-dir -U \
    smclarify \
    "sagemaker>=2,<3" \
    sagemaker-experiments==0.* \
```

```
    sagemaker-pytorch-training

# Install SageMaker data parallel binary (SMDDP)
# Start with dependencies
RUN --mount=type=cache,id=apt-final,target=/var/cache/apt \
    apt-get update && apt-get install -y  --no-install-recommends \
        jq \
        libhwloc-dev \
        libnuma1 \
        libnuma-dev \
        libssl1.1 \
        libtool \
        hwloc \
    && rm -rf /var/lib/apt/lists/*

# Install SMDDP
RUN SMDATAPARALLEL_PT=1 pip install --no-cache-dir ${SMDATAPARALLEL_BINARY}
```

> **Tip**
>
> Untuk informasi lebih umum tentang membuat Dockerfile kustom untuk pelatihan SageMaker, lihat Menggunakan Algoritma Pelatihan Anda Sendiri.

> **Tip**
>
> Jika Anda ingin memperluas Dockerfile khusus untuk menggabungkan pustaka SageMaker paralel model, lihat. Buat Container Docker Anda Sendiri dengan Perpustakaan Paralel Model SageMaker Terdistribusi

Kiat konfigurasi untuk pustaka paralelisme data SageMaker terdistribusi

Tinjau tips berikut sebelum menggunakan perpustakaan paralelisme data SageMaker terdistribusi (SMDDP). Daftar ini mencakup tips yang berlaku di seluruh kerangka kerja.

Topik

- Prapemrosesan data
- Tunggal versus beberapa node
- Efisiensi penskalaan debug dengan Debugger

- Ukuran batch
- Opsi MPI kustom
- Gunakan Amazon FSx dan siapkan kapasitas penyimpanan dan throughput yang optimal

Prapemrosesan data

Jika Anda melakukan pra-proses data selama pelatihan menggunakan pustaka eksternal yang menggunakan CPU, Anda mungkin mengalami kemacetan CPU karena SageMaker data terdistribusi paralel menggunakan CPU untuk operasi. `AllReduce` Anda mungkin dapat meningkatkan waktu pelatihan dengan memindahkan langkah-langkah pra-pemrosesan ke perpustakaan yang menggunakan GPU atau dengan menyelesaikan semua pra-pemrosesan sebelum pelatihan.

Tunggal versus beberapa node

Kami menyarankan Anda menggunakan perpustakaan ini dengan beberapa node. Pustaka dapat digunakan dengan satu host, penyiapan multi-perangkat (misalnya, satu instance komputasi ML dengan beberapa GPU); Namun, ketika Anda menggunakan dua atau lebih node, `AllReduce` operasi perpustakaan memberi Anda peningkatan kinerja yang signifikan. Juga, pada satu host, NVLink sudah berkontribusi pada efisiensi `AllReduce` in-node.

Efisiensi penskalaan debug dengan Debugger

Anda dapat menggunakan Amazon SageMaker Debugger untuk memantau dan memvisualisasikan pemanfaatan CPU dan GPU serta metrik lain yang menarik selama pelatihan. Anda dapat menggunakan aturan bawaan Debugger untuk memantau masalah kinerja komputasi, seperti`CPUBottleneck`,, dan. `LoadBalancing` `LowGPUUtilization` Anda dapat menentukan aturan ini dengan konfigurasi Debugger saat menentukan estimator Amazon Python SageMaker SDK. Jika Anda menggunakan AWS CLI dan AWS SDK for Python (Boto3) untuk pelatihan SageMaker, Anda dapat mengaktifkan Debugger seperti yang ditunjukkan pada Konfigurasi SageMaker Debugger Menggunakan Amazon API. SageMaker

Untuk melihat contoh menggunakan Debugger dalam pekerjaan SageMaker pelatihan, Anda dapat mereferensikan salah satu contoh buku catatan di repositori Contoh SageMaker GitHub Notebook. Untuk mempelajari lebih lanjut tentang Debugger, lihat Amazon SageMaker Debugger.

Ukuran batch

Dalam pelatihan terdistribusi, karena lebih banyak node ditambahkan, ukuran batch harus meningkat secara proporsional. Untuk meningkatkan kecepatan konvergensi saat Anda menambahkan lebih

banyak node ke pekerjaan pelatihan Anda dan meningkatkan ukuran batch global, tingkatkan tingkat pembelajaran.

Salah satu cara untuk mencapai hal ini adalah dengan menggunakan pemanasan tingkat pembelajaran bertahap di mana tingkat pembelajaran ditingkatkan dari nilai kecil ke nilai besar saat pekerjaan pelatihan berlangsung. Jalan ini menghindari peningkatan kecepatan belajar yang tiba-tiba, memungkinkan konvergensi yang sehat pada awal pelatihan. Misalnya, Anda dapat menggunakan Aturan Penskalaan Linear di mana setiap kali ukuran mini-batch dikalikan dengan k, tingkat pembelajaran juga dikalikan dengan k. Untuk mempelajari lebih lanjut tentang teknik ini, lihat paper penelitian, Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour, Section 2 dan 3.

Opsi MPI kustom

Pustaka paralel data SageMaker terdistribusi menggunakan Message Passing Interface (MPI), standar populer untuk mengelola komunikasi antar node dalam cluster berkinerja tinggi, dan menggunakan perpustakaan NCCL NVIDIA untuk komunikasi tingkat GPU. Saat Anda menggunakan pustaka paralel data dengan TensorFlow atau Pytorch`Estimator`, wadah masing-masing mengatur lingkungan MPI dan menjalankan `mpirun` perintah untuk memulai pekerjaan di node cluster.

Anda dapat mengatur operasi MPI kustom menggunakan `custom_mpi_options` parameter di. `Estimator` `mpirun`Bendera apa pun yang diteruskan di bidang ini ditambahkan ke `mpirun` perintah dan dijalankan oleh SageMaker untuk pelatihan. Misalnya, Anda dapat menentukan `distribution` parameter dari `Estimator` menggunakan berikut ini untuk menggunakan NCCL_DEBUGvariabel untuk mencetak versi NCCL di awal program:

```
distribution = {'smdistributed':{'dataparallel':{'enabled': True, "custom_mpi_options":
 "-verbose -x NCCL_DEBUG=VERSION"}}}
```

Gunakan Amazon FSx dan siapkan kapasitas penyimpanan dan throughput yang optimal

Saat melatih model pada beberapa node dengan paralelisme data terdistribusi, sangat disarankan untuk menggunakan FSx for Lustre. Amazon FSx adalah layanan penyimpanan yang dapat diskalakan dan berkinerja tinggi yang mendukung penyimpanan file bersama dengan throughput yang lebih cepat. Menggunakan penyimpanan Amazon FSx dalam skala besar, Anda dapat mencapai kecepatan pemuatan data yang lebih cepat di seluruh node komputasi.

Biasanya, dengan paralelisme data terdistribusi, Anda akan mengharapkan bahwa total throughput pelatihan berskala mendekati linier dengan jumlah GPU. Namun, jika Anda menggunakan penyimpanan Amazon FSx yang kurang optimal, kinerja pelatihan mungkin melambat karena throughput Amazon FSx yang rendah.

Misalnya, jika Anda menggunakan tipe penyebaran SCRATCH_2 dari sistem file Amazon FSx dengan kapasitas penyimpanan minimum 1,2 TiB, kapasitas throughput I/O adalah 240 MB/s. Penyimpanan Amazon FSx berfungsi sedemikian rupa sehingga Anda dapat menetapkan perangkat penyimpanan fisik, dan semakin banyak perangkat yang ditetapkan, semakin besar throughput yang Anda dapatkan. Peningkatan penyimpanan terkecil untuk tipe SRATCH_2 adalah 1,2 TiB, dan gain throughput yang sesuai adalah 240 MB/s.

Asumsikan bahwa Anda memiliki model untuk dilatih pada cluster 4-node lebih dari kumpulan data 100 GB. Dengan ukuran batch tertentu yang dioptimalkan ke cluster, asumsikan bahwa model dapat menyelesaikan satu epoch dalam waktu sekitar 30 detik. Dalam hal ini, kecepatan I/O minimum yang diperlukan adalah sekitar 3 GB/s (100 GB/30 detik). Ini tampaknya merupakan persyaratan throughput yang jauh lebih tinggi daripada 240 MB/s. Dengan kapasitas Amazon FSx yang terbatas, menskalakan pekerjaan pelatihan terdistribusi Anda hingga cluster yang lebih besar dapat memperburuk masalah bottleneck I/O; throughput pelatihan model mungkin meningkat di masa mendatang saat cache meningkat, tetapi throughput Amazon FSx masih bisa menjadi hambatan.

Untuk mengatasi masalah bottleneck I/O seperti itu, Anda harus meningkatkan ukuran penyimpanan Amazon FSx untuk mendapatkan kapasitas throughput yang lebih tinggi. Biasanya, untuk menemukan throughput I/O yang optimal, Anda dapat bereksperimen dengan kapasitas throughput Amazon FSx yang berbeda, menetapkan throughput yang sama atau sedikit lebih rendah dari perkiraan Anda, hingga Anda menemukan bahwa itu cukup untuk menyelesaikan masalah bottleneck I/O. Dalam kasus contoh yang disebutkan di atas, penyimpanan Amazon FSx dengan throughput 2, 4 GB/s dan cache RAM 67 GB sudah cukup. Jika sistem file memiliki throughput yang optimal, throughput pelatihan model harus mencapai maksimum baik segera atau setelah epoch pertama karena cache telah dibangun.

Untuk mempelajari selengkapnya tentang cara meningkatkan jenis penyimpanan dan penyebaran Amazon FSx, lihat halaman berikut di dokumentasi Amazon FSx for Lustre:

- Cara meningkatkan kapasitas penyimpanan
- Kinerja sistem file agregat

FAQ perpustakaan paralelisme data SageMaker terdistribusi Amazon

Gunakan yang berikut ini untuk menemukan jawaban atas pertanyaan umum tentang perpustakaan SMDDP.

T: Saat menggunakan pustaka, bagaimana instans CPU **`allreduce`** -pendukung dikelola? Apakah saya harus membuat cluster CPU-GPU yang heterogen, atau apakah SageMaker layanan membuat C5 tambahan untuk pekerjaan yang menggunakan perpustakaan SMDDP?

Pustaka SMDDP hanya mendukung instans GPU, lebih spesifik, instans P4d dan P4de dengan GPU NVIDIA A100 dan EFA. Tidak ada instance C5 atau CPU tambahan yang diluncurkan; jika tugas SageMaker pelatihan Anda berada di cluster P4d 8-node, hanya 8 instance yang digunakan. `ml.p4d.24xlarge` Tidak ada instance tambahan yang disediakan.

T: Saya memiliki pekerjaan pelatihan yang memakan waktu 5 hari pada satu **`ml.p3.24xlarge`** contoh dengan satu set hiperparameter H1 (tingkat pembelajaran, ukuran batch, pengoptimal, dll). Apakah perpustakaan paralelisme data using SageMaker dan cluster lima kali lebih besar cukup untuk mencapai perkiraan percepatan lima kali? Atau apakah saya harus meninjau kembali hyperparameters pelatihannya setelah mengaktifkan perpustakaan SMDDP?

Pustaka mengubah ukuran batch keseluruhan. Ukuran batch keseluruhan yang baru diskalakan secara linier dengan jumlah instans pelatihan yang digunakan. Sebagai akibatnya, hiperparameter, seperti tingkat pembelajaran, harus diubah untuk memastikan konvergensi.

T: Apakah perpustakaan SMDDP mendukung Spot?

Ya. Anda dapat menggunakan pelatihan spot terkelola. Anda menentukan jalur ke file pos pemeriksaan dalam pekerjaan SageMaker pelatihan. Anda menyimpan dan memulihkan pos pemeriksaan dalam skrip pelatihan mereka seperti yang disebutkan dalam langkah terakhir the section called “TensorFlow (usang)” danthe section called “PyTorch”.

T: Apakah pustaka SMDDP relevan dalam penyiapan multi-perangkat host tunggal?

Pustaka dapat digunakan dalam pelatihan multi-perangkat host tunggal tetapi perpustakaan hanya menawarkan peningkatan kinerja dalam pelatihan multi-host.

T: Di mana dataset pelatihan harus disimpan?

Dataset pelatihan dapat disimpan dalam bucket Amazon S3 atau di drive Amazon FSx. Lihat dokumen ini untuk berbagai sistem file input yang didukung untuk pekerjaan pelatihan.

T: Saat menggunakan perpustakaan SMDDP, apakah wajib memiliki data pelatihan di FSx for Lustre? Bisakah Amazon EFS dan Amazon S3 digunakan?

Kami biasanya menyarankan Anda menggunakan Amazon FSx karena latensi yang lebih rendah dan throughput yang lebih tinggi. Jika mau, Anda dapat menggunakan Amazon EFS atau Amazon S3.

T: Dapatkah perpustakaan digunakan dengan node CPU?

Tidak. Untuk menemukan jenis instance yang didukung oleh pustaka SMDDP, lihat. the section called "Tipe instans yang didukung"

T: Kerangka kerja dan versi kerangka kerja apa yang saat ini didukung oleh pustaka SMDDP saat diluncurkan?

perpustakaan SMDDP saat ini mendukung PyTorch v1.6.0 atau yang lebih baru dan TensorFlow v2.3.0 atau yang lebih baru. Itu tidak mendukung TensorFlow 1.x. Untuk informasi selengkapnya tentang versi pustaka SMDDP mana yang dikemas dalam wadah pembelajaran AWS mendalam, lihat Catatan Rilis untuk Deep Learning Containers.

T: Apakah pustaka mendukung AMP?

Ya, pustaka SMDDP mendukung Automatic Mixed Precision (AMP) di luar kotak. Tidak diperlukan tindakan tambahan untuk menggunakan AMP selain modifikasi tingkat kerangka kerja pada skrip latihan Anda. Jika gradien berada di FP16, pustaka paralelisme SageMaker data menjalankan operasinya di FP16. `AllReduce` Untuk informasi selengkapnya tentang penerapan API AMP ke skrip pelatihan Anda, lihat sumber daya berikut:

- Kerangka kerja - PyTorch dalam dokumentasi NVIDIA Deep Learning Performace
- Kerangka kerja - TensorFlow dalam dokumentasi NVIDIA Deep Learning Performace
- Presisi Campuran Otomatis untuk Pembelajaran Mendalam di Dokumen Pengembang NVIDIA
- Memperkenalkan presisi campuran PyTorch otomatis asli untuk pelatihan lebih cepat pada GPU NVIDIA di Blog PyTorch
- TensorFlow API presisi campuran dalam TensorFlow dokumentasi

T: Bagaimana cara mengidentifikasi jika pekerjaan pelatihan terdistribusi saya melambat karena kemacetan I/O?

Dengan cluster yang lebih besar, pekerjaan pelatihan membutuhkan lebih banyak throughput I/O, dan oleh karena itu throughput pelatihan mungkin membutuhkan waktu lebih lama (lebih banyak zaman) untuk meningkatkan kinerja maksimum. Ini menunjukkan bahwa I/O sedang macet dan cache lebih sulit untuk dibangun saat Anda meningkatkan skala node (persyaratan throughput yang lebih tinggi dan topologi jaringan yang lebih kompleks). Untuk informasi selengkapnya tentang memantau throughput Amazon FSx CloudWatch, lihat Memantau FSx for Lustre di Panduan Pengguna FSx for Lustre.

T: Bagaimana cara mengatasi kemacetan I/O saat menjalankan pekerjaan pelatihan terdistribusi dengan paralelisme data?

Kami sangat menyarankan Anda menggunakan Amazon FSx sebagai saluran data Anda jika Anda menggunakan Amazon S3. Jika Anda sudah menggunakan Amazon FSx tetapi masih mengalami masalah bottleneck I/O, Anda mungkin telah menyiapkan sistem file Amazon FSx Anda dengan throughput I/O rendah dan kapasitas penyimpanan yang kecil. Untuk informasi selengkapnya tentang cara memperkirakan dan memilih ukuran kapasitas throughput I/O yang tepat, lihat. Gunakan Amazon FSx dan siapkan kapasitas penyimpanan dan throughput yang optimal

T: (Untuk perpustakaan v1.4.0 atau yang lebih baru) Bagaimana cara mengatasi **`Invalid backend`** kesalahan saat menginisialisasi grup proses.

Jika Anda menemukan pesan kesalahan `ValueError: Invalid backend: 'smddp'` saat meneleponinit_process_group, ini disebabkan oleh perubahan yang melanggar di perpustakaan SMDDP v1.4.0 dan yang lebih baru. Anda harus mengimpor PyTorch klien perpustakaan,`smdistributed.dataparallel.torch.torch_smddp`, yang mendaftar `smddp` sebagai backend untuk. PyTorch Untuk mempelajari selengkapnya, lihat the section called “PyTorch”.

T: (Untuk perpustakaan SMDDP v1.4.0 atau yang lebih baru) Saya ingin memanggil primitif kolektif antarmuka. **`torch.distributed`** Primitif mana yang **`smddp`** didukung backend?

Di v1.4.0, perpustakaan SMDDP mendukung`all_reduce`,`broadcast`,`reduce`,`all_gather`, dan `barrier` antarmuka. `torch.distributed`

T: (Untuk perpustakaan SMDDP v1.4.0 atau yang lebih baru) Apakah API baru ini berfungsi dengan kelas atau pustaka DDP khusus lainnya seperti Apex DDP?

Pustaka SMDDP diuji dengan library paralel data terdistribusi pihak ketiga lainnya dan implementasi kerangka kerja yang menggunakan modul. `torch.distribtued` Menggunakan perpustakaan SMDDP dengan kelas DDP kustom bekerja selama operasi kolektif yang digunakan oleh kelas DDP kustom didukung oleh perpustakaan SMDDP. Lihat pertanyaan sebelumnya untuk daftar kolektif yang didukung. Jika Anda memiliki kasus penggunaan ini dan membutuhkan dukungan lebih lanjut, hubungi SageMaker tim melalui Pusat AWS Dukungan atau Forum AWS Pengembang untuk Amazon SageMaker.

T: Apakah perpustakaan SMDDP mendukung opsi bring-your-own-container (BYOC)? Jika demikian, bagaimana cara menginstal perpustakaan dan menjalankan pekerjaan pelatihan terdistribusi dengan menulis Dockerfile khusus?

Jika Anda ingin mengintegrasikan perpustakaan SMDDP dan dependensi minimumnya ke dalam wadah Docker Anda sendiri, BYOC adalah pendekatan yang tepat. Anda dapat membangun wadah Anda sendiri menggunakan file biner perpustakaan. Proses yang disarankan adalah menulis Dockerfile khusus dengan pustaka dan dependensinya, membangun wadah Docker, menghostingnya di Amazon ECR, dan menggunakan URI gambar ECR untuk meluncurkan pekerjaan pelatihan menggunakan kelas estimator generik. SageMaker Untuk petunjuk lebih lanjut tentang cara menyiapkan Dockerfile khusus untuk pelatihan terdistribusi SageMaker dengan perpustakaan SMDDP, lihat. Buat wadah Docker Anda sendiri dengan pustaka paralel data SageMaker terdistribusi

Pemecahan masalah untuk pelatihan terdistribusi di Amazon SageMaker

Jika Anda memiliki masalah dalam menjalankan pekerjaan pelatihan saat Anda menggunakan pustaka, gunakan daftar berikut untuk mencoba memecahkan masalah. Jika Anda memerlukan dukungan lebih lanjut, hubungi SageMaker tim melalui Pusat AWS Dukungan atau Forum AWS Pengembang untuk Amazon Amazon SageMaker.

Topik

- Menggunakan data SageMaker terdistribusi paralel dengan Amazon SageMaker Debugger dan pos pemeriksaan
- Awalan tak terduga yang dilampirkan ke kunci parameter model
- SageMaker mengulur-ulur pekerjaan pelatihan terdistribusi selama inisialisasi
- SageMaker terdistribusi pekerjaan pelatihan mengulur-ulur di akhir pelatihan
- Mengamati degradasi efisiensi penskalaan karena kemacetan throughput Amazon FSx
- SageMaker pekerjaan pelatihan terdistribusi dengan PyTorch mengembalikan peringatan penghentian

Menggunakan data SageMaker terdistribusi paralel dengan Amazon SageMaker Debugger dan pos pemeriksaan

Untuk memantau kemacetan sistem, operasi kerangka kerja profil, dan tensor keluaran model debug untuk tugas pelatihan dengan SageMaker paralel data terdistribusi, gunakan Amazon Debugger. SageMaker

Namun, ketika Anda menggunakan SageMaker Debugger, SageMaker distributed data parallel, dan SageMaker checkpoints, Anda mungkin melihat error yang terlihat seperti contoh berikut.

```
SMDebug Does Not Currently Support Distributed Training Jobs With Checkpointing Enabled
```

Hal ini disebabkan oleh kesalahan internal antara Debugger dan checkpoints, yang terjadi ketika Anda mengaktifkan distributed SageMaker data parallel.

- Jika Anda mengaktifkan ketiga fitur, SageMaker Python SDK secara otomatis mematikan Debugger dengan meneruskan`debugger_hook_config=False`, yang setara dengan contoh kerangka kerja berikut. `estimator`

```
bucket=sagemaker.Session().default_bucket()
base_job_name="sagemaker-checkpoint-test"
checkpoint_in_bucket="checkpoints"

# The S3 URI to store the checkpoints
checkpoint_s3_bucket="s3://{}/{}/{}".format(bucket, base_job_name,
 checkpoint_in_bucket)

estimator = TensorFlow(
    ...

    distribution={"smdistributed": {"dataparallel": { "enabled": True }}},
    checkpoint_s3_uri=checkpoint_s3_bucket,
    checkpoint_local_path="/opt/ml/checkpoints",
    debugger_hook_config=False
)
```

- Jika Anda ingin tetap menggunakan data parallel dan SageMaker Debugger SageMaker terdistribusi, solusinya adalah menambahkan fungsi checkpointing secara manual ke skrip pelatihan Anda alih-alih menentukan parameter dan dari estimator. `checkpoint_s3_uri` `checkpoint_local_path` Untuk informasi selengkapnya tentang menyiapkan pos pemeriksaan manual dalam skrip pelatihan, lihat. Menyimpan Pos Pemeriksaan

Awalan tak terduga yang dilampirkan ke kunci parameter model

Untuk pekerjaan pelatihan PyTorch terdistribusi, awalan yang tidak terduga (`model`misalnya) mungkin dilampirkan ke `state_dict` kunci (parameter model). Pustaka paralel SageMaker data tidak secara langsung mengubah atau menambahkan nama parameter model apa pun saat pekerjaan PyTorch pelatihan menyimpan artefak model. Pelatihan PyTorch terdistribusi mengubah nama dalam `state_dict` untuk melewati jaringan, mendahului awalan. Jika Anda mengalami masalah kegagalan model karena nama parameter yang berbeda saat Anda menggunakan pustaka paralel SageMaker data dan pos pemeriksaan untuk PyTorch pelatihan, sesuaikan kode contoh

berikut untuk menghapus awalan pada langkah Anda memuat pos pemeriksaan dalam skrip pelatihan Anda.

```
state_dict = {k.partition('model.')[2]:state_dict[k] for k in state_dict.keys()}
```

Ini mengambil setiap `state_dict` kunci sebagai nilai string, memisahkan string pada kemunculan pertama`'model.'`, dan mengambil item daftar ketiga (dengan indeks 2) dari string yang dipartisi.

Untuk informasi selengkapnya tentang masalah awalan, lihat utas diskusi di nama parameter Awalan dalam model tersimpan jika dilatih oleh multi-GPU? di forum PyTorch diskusi.

Untuk informasi selengkapnya tentang PyTorch metode penyimpanan dan pemuatan model, lihat Menyimpan & Memuat Model di Seluruh Perangkat dalam PyTorchdokumentasi.

SageMaker mengulur-ulur pekerjaan pelatihan terdistribusi selama inisialisasi

Jika pekerjaan pelatihan paralel data SageMaker terdistribusi Anda terhenti selama inisialisasi saat menggunakan instans berkemampuan EFA (`ml.p3dn.24xlarge`dan`ml.p4d.24xlarge`), ini mungkin disebabkan oleh kesalahan konfigurasi dalam grup keamanan subnet VPC yang digunakan untuk pekerjaan pelatihan. EFA membutuhkan konfigurasi grup keamanan yang tepat untuk mengaktifkan lalu lintas antar node.

Untuk mengonfigurasi aturan masuk dan keluar untuk grup keamanan

1. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pilih Grup Keamanan di panel navigasi kiri.
3. Pilih grup keamanan yang terkait dengan subnet VPC yang Anda gunakan untuk pelatihan.
4. Di bagian Detail, salin ID grup Keamanan.
5. Pada tab Inbound rules (Aturan ke dalam), pilih Edit inbound rules (Edit aturan ke dalam).
6. Pada halaman Edit aturan ke dalam, lakukan hal berikut ini:
   a. Pilih Add rule (Tambahkan aturan).
   b. Untuk Tipe, pilih Semua lalu lintas.
   c. Untuk Sumber, pilih Kustom, tempel ID grup keamanan ke dalam kotak pencarian, dan pilih grup keamanan yang muncul.
7. Pilih Simpan aturan untuk menyelesaikan konfigurasi aturan masuk untuk grup keamanan.
8. Pada tab Aturan keluar, pilih Edit aturan keluar.

9. Ulangi langkah 6 dan 7 untuk menambahkan aturan yang sama dengan aturan keluar.

Setelah Anda menyelesaikan langkah-langkah sebelumnya untuk mengonfigurasi grup keamanan dengan aturan masuk dan keluar, jalankan kembali tugas pelatihan dan verifikasi apakah masalah penghentian teratasi.

Untuk informasi selengkapnya tentang mengonfigurasi grup keamanan untuk VPC dan EFA, lihat Grup keamanan untuk VPC dan Adaptor Kain Elastis.

SageMaker terdistribusi pekerjaan pelatihan mengulur-ulur di akhir pelatihan

Salah satu akar penyebab masalah mengulur-ulur di akhir pelatihan adalah ketidakcocokan dalam jumlah batch yang diproses per zaman di berbagai peringkat. Semua pekerja (GPU) menyinkronkan gradien lokal mereka di lintasan mundur untuk memastikan mereka semua memiliki salinan model yang sama di akhir iterasi batch. Jika ukuran batch ditetapkan secara tidak merata ke kelompok pekerja yang berbeda selama zaman terakhir pelatihan, pekerjaan pelatihan berhenti. Misalnya, sementara sekelompok pekerja (grup A) selesai memproses semua batch dan keluar dari loop pelatihan, kelompok pekerja lain (grup B) mulai memproses batch lain dan masih mengharapkan komunikasi dari grup A untuk menyinkronkan gradien. Hal ini menyebabkan grup B menunggu grup A, yang sudah menyelesaikan pelatihan dan tidak memiliki gradien untuk disinkronkan.

Oleh karena itu, saat menyiapkan kumpulan data pelatihan Anda, penting bahwa setiap pekerja mendapatkan jumlah sampel data yang sama sehingga setiap pekerja melewati jumlah batch yang sama saat pelatihan. Pastikan setiap peringkat mendapatkan jumlah batch yang sama untuk menghindari masalah stalling ini.

Mengamati degradasi efisiensi penskalaan karena kemacetan throughput Amazon FSx

Salah satu penyebab potensial penurunan efisiensi penskalaan adalah batas throughput FSx. Jika Anda mengamati penurunan efisiensi penskalaan yang tiba-tiba saat beralih ke cluster pelatihan yang lebih besar, coba gunakan sistem file FSx for Lustre yang lebih besar dengan batas throughput yang lebih tinggi. Untuk informasi selengkapnya, lihat Menggabungkan kinerja sistem file dan Mengelola kapasitas penyimpanan dan throughput di Panduan Pengguna Amazon FSx for Lustre.

SageMaker pekerjaan pelatihan terdistribusi dengan PyTorch mengembalikan peringatan penghentian

Sejak v1.4.0, pustaka paralelisme data SageMaker terdistribusi berfungsi sebagai backend terdistribusi. PyTorch Karena perubahan yang melanggar penggunaan pustaka dengan PyTorch,

Anda mungkin menemukan pesan peringatan bahwa `smdistributed` API untuk paket PyTorch terdistribusi tidak digunakan lagi. Pesan peringatan harus mirip dengan yang berikut ini:

```
smdistributed.dataparallel.torch.dist is deprecated in the SageMaker distributed data
 parallel library v1.4.0+.
Please use torch.distributed and specify 'smddp' as a backend when initializing process
 group as follows:
torch.distributed.init_process_group(backend='smddp')
For more information, see the library's API documentation at
https://docs.aws.amazon.com/sagemaker/latest/dg/data-parallel-modify-sdp-pt.html
```

Di v1.4.0 dan yang lebih baru, perpustakaan hanya perlu diimpor sekali di bagian atas skrip pelatihan Anda dan ditetapkan sebagai backend selama inisialisasi terdistribusi. PyTorch Dengan satu baris spesifikasi backend, Anda dapat menjaga skrip PyTorch pelatihan Anda tidak berubah dan langsung menggunakan modul terdistribusi. PyTorch Lihat Gunakan perpustakaan SMDDP dalam skrip pelatihan Anda PyTorch untuk mempelajari tentang perubahan yang melanggar dan cara baru untuk menggunakan perpustakaan PyTorch.

SageMaker perpustakaan paralelisme model v2

> **Note**
>
> Sejak rilis pustaka SageMaker model paralelisme (SMP) v2.0.0 pada 19 Desember 2023, dokumentasi ini diperbarui untuk perpustakaan SMP v2. Untuk versi pustaka SMP sebelumnya, lihatthe section called “(Diarsipkan) perpustakaan SageMaker paralelisme model v1.x”.

Pustaka paralelisme SageMaker model Amazon adalah kemampuan yang memungkinkan kinerja tinggi dan pelatihan skala besar SageMaker yang dioptimalkan pada instans komputasi yang SageMaker dipercepat. the section called “Fitur inti dari SMP v2”Termasuk teknik dan pengoptimalan untuk mempercepat dan menyederhanakan pelatihan model besar, seperti paralelisme data sharded hybrid, paralelisme tensor, checkpointing aktivasi, dan pembongkaran aktivasi. Anda dapat menggunakan perpustakaan SMP untuk mempercepat pelatihan dan penyempurnaan model bahasa besar (LLM), model visi besar (LVM), dan model fondasi (MM) dengan ratusan miliar parameter.

Pustaka paralelisme SageMaker model v2 (SMP v2) menyelaraskan API dan metode perpustakaan dengan open source PyTorch Fully Sharded Data Parallelism (FSDP), yang memberi Anda manfaat pengoptimalan kinerja SMP dengan perubahan kode minimal. Dengan SMP v2, Anda dapat

meningkatkan kinerja komputasi dari pelatihan model state-of-the-art besar SageMaker dengan membawa skrip pelatihan PyTorch FSDP Anda. SageMaker

Anda dapat menggunakan SMP v2 untuk pekerjaan SageMaker Pelatihan umum dan beban kerja pelatihan yang didistribusikan pada the section called “SageMaker HyperPod” cluster.

Topik

Pengantar paralelisme model

Paralelisme model adalah metode pelatihan terdistribusi di mana model pembelajaran mendalam (DL) dipartisi di beberapa GPU dan instance. SageMaker Model parallel library v2 (SMP v2) kompatibel dengan PyTorch API dan kemampuan asli. Ini memudahkan Anda untuk menyesuaikan skrip pelatihan PyTorch Fully Sharded Data Parallel (FSDP) Anda ke platform SageMaker Pelatihan dan memanfaatkan peningkatan kinerja yang disediakan SMP v2.

Halaman pengantar ini memberikan gambaran tingkat tinggi tentang paralelisme model dan deskripsi tentang bagaimana hal itu dapat membantu mengatasi masalah yang muncul saat melatih model pembelajaran mendalam (DL) yang biasanya berukuran sangat besar. Ini juga memberikan contoh apa yang ditawarkan perpustakaan paralel SageMaker model untuk membantu mengelola strategi paralel model dan konsumsi memori.

Apa itu paralelisme model?

Meningkatkan ukuran model pembelajaran mendalam (lapisan dan parameter) menghasilkan akurasi yang lebih baik untuk tugas-tugas kompleks seperti visi komputer dan pemrosesan bahasa alami. Namun, ada batasan untuk ukuran model maksimum yang dapat Anda muat dalam memori satu GPU. Saat melatih model DL, batasan memori GPU dapat menjadi hambatan dengan cara berikut:

- Mereka membatasi ukuran model yang dapat Anda latih, karena jejak memori model berskala proporsional dengan jumlah parameter.
- Mereka membatasi ukuran batch per GPU selama pelatihan, menurunkan pemanfaatan GPU dan efisiensi pelatihan.

Untuk mengatasi keterbatasan yang terkait dengan pelatihan model pada satu GPU, SageMaker sediakan pustaka paralel model untuk membantu mendistribusikan dan melatih model DL secara efisien pada beberapa node komputasi. Selain itu, dengan perpustakaan, Anda dapat mencapai pelatihan terdistribusi yang dioptimalkan menggunakan perangkat yang didukung EFA, yang meningkatkan kinerja komunikasi antar-node dengan latensi rendah, throughput tinggi, dan bypass OS.

Perkirakan kebutuhan memori sebelum menggunakan paralelisme model

Sebelum Anda menggunakan perpustakaan paralel SageMaker model, pertimbangkan hal berikut untuk memahami persyaratan memori untuk melatih model DL besar.

Untuk pekerjaan pelatihan yang menggunakan presisi campuran otomatis seperti `float16` (FP16) atau `bfloat16` (BF16) dan pengoptimal Adam, memori GPU yang diperlukan per parameter adalah sekitar 20 byte, yang dapat kita uraikan sebagai berikut:

- Parameter FP16 atau BF16 ~ 2 byte
- Gradien FP16 atau BF16 ~ 2 byte
- Status pengoptimal FP32 ~ 8 byte berdasarkan pengoptimal Adam
- Salinan parameter FP32 ~ 4 byte (diperlukan untuk operasi (OA`optimizer apply`))
- Salinan gradien FP32 ~ 4 byte (diperlukan untuk operasi OA)

Bahkan untuk model DL yang relatif kecil dengan 10 miliar parameter, dapat memerlukan setidaknya 200GB memori, yang jauh lebih besar daripada memori GPU biasa (misalnya, NVIDIA A100 dengan memori 40GB/80GB) yang tersedia pada satu GPU. Di atas persyaratan memori untuk status model dan pengoptimal, ada konsumen memori lain seperti aktivasi yang dihasilkan dalam pass maju. Memori yang dibutuhkan bisa jauh lebih besar dari 200GB.

Untuk pelatihan terdistribusi, kami menyarankan Anda menggunakan instans Amazon EC2 P4 dan P5 yang masing-masing memiliki GPU NVIDIA A100 dan H100 Tensor Core. Untuk detail selengkapnya tentang spesifikasi seperti inti CPU, RAM, volume penyimpanan terlampir, dan

bandwidth jaringan, lihat bagian Komputasi Akselerasi di halaman Jenis Instans Amazon EC2. Misalnya jenis yang didukung SMP v2, lihatthe section called “Tipe instans yang didukung”.

Bahkan dengan instans komputasi yang dipercepat, model dengan sekitar 10 miliar parameter seperti Megatron-LM dan T5, dan bahkan model yang lebih besar dengan ratusan miliar parameter seperti GPT-3, tidak dapat memuat replika model di setiap perangkat GPU.

Bagaimana perpustakaan menggunakan paralelisme model dan teknik penghematan memori

Pustaka terdiri dari berbagai jenis fitur paralelisme model dan fitur hemat memori seperti sharding status pengoptimal, checkpointing aktivasi, dan pembongkaran aktivasi. Semua teknik ini dapat digabungkan untuk melatih model besar secara efisien yang terdiri dari ratusan miliar parameter.

Topik

- Paralelisme data yang dibagikan
- Paralelisme tensor
- Titik pemeriksaan aktivasi dan pembongkaran
- Memilih teknik yang tepat untuk model Anda

Paralelisme data yang dibagikan

Paralelisme data sharded adalah teknik pelatihan terdistribusi hemat memori yang membagi status model (parameter model, gradien, dan status pengoptimal) di seluruh GPU dalam grup data-paralel.

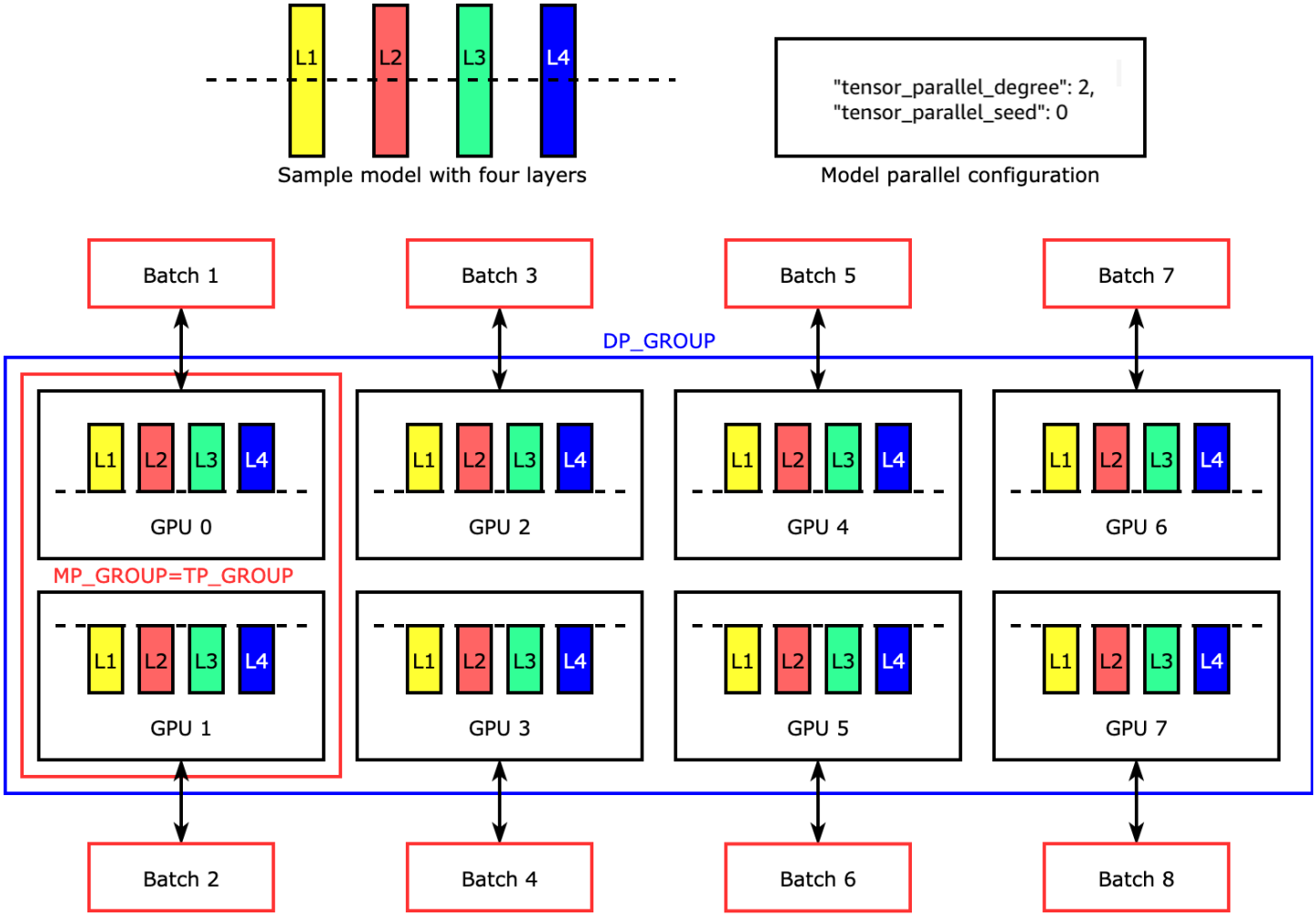
SageMaker mengimplementasikan paralelisme data sharded melalui FSDP, dan memperluasnya untuk mengimplementasikan strategi sharding hybrid sadar skala yang dibahas dalam posting blog penskalaan near-linear dari pelatihan model raksasa pada. AWS

Anda dapat menerapkan paralelisme data sharded ke model Anda sebagai strategi mandiri. Selain itu, jika Anda menggunakan instans GPU paling berkinerja yang dilengkapi dengan GPU NVIDIA A100 Tensor Core, `ml.p4d.24xlarge` dan`ml.p4de.24xlarge`, Anda dapat memanfaatkan peningkatan kecepatan pelatihan dari `AllGather` operasi yang ditawarkan oleh perpustakaan paralelisme SageMaker data (SMDDP).

Untuk menyelam jauh ke dalam paralelisme data sharded dan mempelajari cara mengaturnya atau menggunakan kombinasi paralelisme data sharded dengan teknik lain seperti paralelisme tensor dan pelatihan presisi campuran, lihat. the section called “Paralelisme data sharded hibrida”

Paralelisme tensor

Paralelisme tensor membagi lapisan individu, ataunn`.Modules`, di seluruh perangkat untuk berjalan secara paralel. Gambar berikut menunjukkan contoh paling sederhana tentang bagaimana perpustakaan SMP membagi model dengan empat lapisan untuk mencapai paralelisme tensor dua arah (). `"tensor_parallel_degree": 2` Pada gambar berikut, notasi untuk grup paralel model, grup paralel tensor, dan grup paralel data adalah`MP_GROUP`,`TP_GROUP`, dan `DP_GROUP` masing-masing. Lapisan setiap replika model dibagi dua dan didistribusikan menjadi dua GPU. Perpustakaan mengelola komunikasi di seluruh replika model terdistribusi tensor.



Untuk menyelam jauh ke dalam paralelisme tensor dan fitur hemat memori lainnya untuk PyTorch, dan untuk mempelajari cara mengatur kombinasi fitur inti, lihat. the section called “Paralelisme tensor”

Titik pemeriksaan aktivasi dan pembongkaran

Untuk menyimpan memori GPU, pustaka mendukung checkpointing aktivasi untuk menghindari penyimpanan aktivasi internal dalam memori GPU untuk modul yang ditentukan pengguna selama

forward pass. Pustaka menghitung ulang aktivasi ini selama pass mundur. Selain itu, dengan pembongkaran aktivasi, ia menurunkan aktivasi yang disimpan ke memori CPU dan mengambilnya kembali ke GPU selama lintasan mundur untuk lebih mengurangi jejak memori aktivasi. Untuk informasi selengkapnya tentang cara menggunakan fitur ini, lihat the section called “Pos pemeriksaan aktivasi” danthe section called “Pembongkaran aktivasi”.

Memilih teknik yang tepat untuk model Anda

Untuk informasi selengkapnya tentang memilih teknik dan konfigurasi yang tepat, lihatthe section called “Praktik terbaik”.

Kerangka kerja yang didukung dan Wilayah AWS

Sebelum menggunakan pustaka paralelisme SageMaker model v2 (SMP v2), periksa kerangka kerja dan jenis instance yang didukung dan tentukan apakah ada cukup kuota di akun Anda dan. AWS Wilayah AWS

> **Note**
>
> Untuk memeriksa pembaruan terbaru dan catatan rilis perpustakaan, lihatthe section called “Catatan rilis”.

Kerangka kerja yang didukung

SMP v2 mendukung kerangka pembelajaran mendalam berikut dan tersedia melalui wadah SMP Docker dan saluran SMP Conda. Saat Anda menggunakan kelas estimator kerangka kerja di SageMaker Python SDK dan menentukan konfigurasi distribusi untuk menggunakan SMP v2 SageMaker , secara otomatis mengambil kontainer SMP Docker. Untuk menggunakan SMP v2, kami menyarankan agar Anda selalu memperbarui SDK SageMaker Python di lingkungan pengembangan Anda.

PyTorch versi yang didukung oleh SageMaker pustaka paralelisme model

PyTorch versi	SageMaker model versi perpustakaan paralelisme	URI gambar SMP Docker
v2.1.2	`smdistributed-modelparallel==v2.1.0`	`658645717510.dkr.ecr.us-west-2.amazo`

PyTorch versi	SageMaker model versi perpustakaan paralelisme	URI gambar SMP Docker
		`naws.com/smdistrib uted-modelparallel :2.1.2-gpu-py310-c u121`
v2.0.1	`smdistributed-mode lparallel==v2.0.0`	`658645717510.dkr.e cr.us-west-2.amazo naws.com/smdistrib uted-modelparallel :2.0.1-gpu-py310-c u121`

Saluran SMP Conda

Bucket S3 berikut adalah saluran Conda publik yang diselenggarakan oleh tim layanan SMP. Jika Anda ingin menginstal perpustakaan SMP v2 di lingkungan seperti SageMaker HyperPod cluster, gunakan saluran Conda ini untuk menginstal pustaka SMP dengan benar.

- `https://sagemaker-distributed-model-parallel.s3.us-west-2.amazonaws.com/smp-v2/`

Untuk informasi selengkapnya tentang saluran Conda secara umum, lihat Saluran di dokumentasi Conda.

> **Note**
>
> Untuk menemukan versi sebelumnya dari perpustakaan SMP v1.x dan DLC pra-paket, lihat the section called “Kerangka Kerja yang Didukung” di dokumentasi SMP v1.

Gunakan SMP v2 dengan pustaka open source

Pustaka SMP v2 bekerja dengan perpustakaan open source PyTorch berbasis lainnya seperti PyTorch Lightning, Hugging Face Transformers, dan Hugging Face Accelerate, karena SMP v2 kompatibel dengan API FSDP. PyTorch Jika Anda memiliki pertanyaan lebih lanjut tentang

penggunaan perpustakaan SMP dengan pustaka pihak ketiga lainnya, hubungi tim layanan SMP di. `sm-model-parallel-feedback@amazon.com`

Wilayah AWS

SMP v2 tersedia di semua Wilayah AWS tempat AWSDeep Learning Containers SageMaker berada dalam layanan. Untuk informasi selengkapnya, lihat Gambar Deep Learning Containers yang Tersedia.

Tipe instans yang didukung

SMP v2 membutuhkan salah satu dari jenis contoh MS berikut.

Jenis instans
`ml.p4d.24xlarge`
`ml.p4de.24xlarge`
`ml.p5.48xlarge`

Untuk spesifikasi jenis instans pembelajaran SageMaker mesin secara umum, lihat bagian Komputasi Akselerasi di halaman Jenis Instans Amazon EC2. Untuk informasi tentang harga instans, lihat SageMakerHarga Amazon.

Jika Anda menemukan pesan galat yang serupa dengan berikut ini, ikuti petunjuk di Meminta peningkatan kuota dalam Panduan Pengguna Service AWS Quotas.

```
ResourceLimitExceeded: An error occurred (ResourceLimitExceeded) when calling
    the CreateTrainingJob operation: The account-level service limit 'ml.p3dn.24xlarge
    for training job usage' is 0 Instances, with current utilization of 0 Instances
    and a request delta of 1 Instances.
    Please contact AWS support to request an increase for this limit.
```

Memulai dengan SageMaker pustaka paralelisme model v2

Di halaman ini, Anda akan mempelajari cara menggunakan SageMaker model parallelism library v2 API dan memulai menjalankan tugas pelatihan PyTorch Fully Sharded Data Parallel (FSDP) di platform Pelatihan atau di SageMaker cluster. SageMaker HyperPod

Ada berbagai skenario untuk menjalankan pekerjaan PyTorch pelatihan dengan SMP v2.

1. Untuk SageMaker pelatihan, gunakan salah satu SageMaker Framework Container yang sudah dibuat sebelumnya untuk PyTorch v2.0.1 dan yang lebih baru, yang sudah dikemas sebelumnya dengan SMP v2.
2. Gunakan file biner SMP v2 untuk menyiapkan lingkungan Conda untuk menjalankan beban kerja pelatihan terdistribusi di cluster. SageMaker HyperPod
3. Perluas SageMaker Framework Container yang sudah dibuat sebelumnya untuk PyTorch v2.0.1 dan yang lebih baru untuk menginstal persyaratan fungsional tambahan apa pun untuk kasus penggunaan Anda. Untuk mempelajari cara memperluas wadah yang sudah dibuat sebelumnya, lihatMemperluas Kontainer Pra-dibangun.
4. Anda juga dapat membawa wadah Docker Anda sendiri dan secara manual mengatur semua lingkungan SageMaker Pelatihan menggunakan toolkit SageMaker Pelatihan dan menginstal file biner SMP v2. Ini adalah opsi yang paling tidak direkomendasikan karena kompleksitas dependensi. Untuk mempelajari cara menjalankan container Docker Anda sendiri, lihat Mengadaptasi Wadah Pelatihan Anda Sendiri.

Panduan memulai ini mencakup dua skenario pertama.

Topik

- Langkah 1: Sesuaikan skrip pelatihan PyTorch FSDP Anda
- Langkah 2: Luncurkan pekerjaan pelatihan

Langkah 1: Sesuaikan skrip pelatihan PyTorch FSDP Anda

Untuk mengaktifkan dan mengkonfigurasi perpustakaan SMP v2, mulailah dengan mengimpor dan menambahkan `torch.sagemaker.init()` modul di bagian atas skrip. Modul ini mengambil kamus konfigurasi SMP the section called “Parameter konfigurasi fitur inti SMP v2” yang akan Anda persiapkan. the section called “Langkah 2: Luncurkan pekerjaan pelatihan” Selain itu, untuk menggunakan berbagai fitur inti yang ditawarkan oleh SMP v2, Anda mungkin perlu membuat beberapa perubahan lagi untuk menyesuaikan skrip pelatihan Anda. Petunjuk lebih rinci tentang mengadaptasi skrip pelatihan Anda untuk menggunakan fitur inti SMP v2 disediakan di. the section called “Fitur inti dari SMP v2”

SageMaker Training

Dalam skrip pelatihan Anda, tambahkan dua baris kode berikut, yang merupakan persyaratan minimal untuk memulai pelatihan dengan SMP v2. Dithe section called “Langkah 2: Luncurkan

pekerjaan pelatihan", Anda akan menyiapkan objek kelas SageMaker `PyTorch` estimator dengan kamus konfigurasi SMP melalui `distribution` argumen kelas estimator.

```
import torch.sagemaker as tsm
tsm.init()
```

> **Note**
>
> Anda juga dapat langsung meneruskan kamus konfigurasi the section called "Parameter konfigurasi fitur inti SMP v2" ke `torch.sagemaker.init()` modul. Namun, parameter diteruskan ke PyTorch estimator di the section called "Langkah 2: Luncurkan pekerjaan pelatihan" take priority dan mengesampingkan yang ditentukan ke modul. `torch.sagemaker.init()`

SageMaker HyperPod

Dalam skrip pelatihan Anda, tambahkan dua baris kode berikut. Dithe section called "Langkah 2: Luncurkan pekerjaan pelatihan", Anda akan menyiapkan `smp_config.json` file untuk menyiapkan konfigurasi SMP dalam format JSON, dan mengunggahnya ke penyimpanan atau sistem file yang dipetakan dengan cluster Anda. SageMaker HyperPod Kami menyarankan Anda menyimpan file konfigurasi di bawah direktori yang sama tempat Anda mengunggah skrip pelatihan Anda.

```
import torch.sagemaker as tsm
tsm.init("/dir_to_training_files/smp_config.json")
```

> **Note**
>
> Anda juga dapat langsung meneruskan kamus konfigurasi the section called "Parameter konfigurasi fitur inti SMP v2" ke dalam `torch.sagemaker.init()` modul.

Langkah 2: Luncurkan pekerjaan pelatihan

Pelajari cara mengonfigurasi opsi distribusi SMP untuk meluncurkan pekerjaan pelatihan PyTorch FSDP dengan fitur inti SMP.

SageMaker Training

Saat Anda menyiapkan objek peluncur pekerjaan pelatihan dari kelas estimator PyTorch kerangka kerja di SageMaker Python SDK, konfigurasikan the section called “Parameter konfigurasi fitur inti SMP v2” melalui argumen sebagai berikut. `distribution`

> **Note**
>
> `distribution`Konfigurasi untuk SMP v2 terintegrasi dalam SDK SageMaker Python mulai dari v2.200. Pastikan Anda menggunakan SageMaker Python SDK v2.200 atau yang lebih baru.

> **Note**
>
> Di SMP v2, Anda harus mengkonfigurasi `smdistributed` dengan `torch_distributed` `distribution` argumen SageMaker `PyTorch` estimator. Dengan`torch_distributed`, SageMaker run`torchrun`, yang merupakan peluncur pekerjaan multi-node default dari PyTorch Distributed.

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(
    # Pass the training script you adapted with SMP from Step 1
    entry_point="your-training-script.py",
    ... # Configure other required and optional parameters
    distribution={
        "torch_distributed": { "enabled": True },
        "smdistributed": {
            "modelparallel": {
                "enabled": True,
                "parameters": {
                    "hybrid_shard_degree": Integer,
                    "sm_activation_offloading": Boolean,
                    "activation_loading_horizon": Integer,
                    "fsdp_cache_flush_warnings": Boolean,
                    "allow_empty_shards": Boolean,
                    "tensor_parallel_degree": Integer,
                    "tensor_parallel_seed": Integer
```

```
                }
            }
        }
    }
)
```

SageMaker HyperPod

Sebelum Anda mulai, pastikan apakah prasyarat berikut terpenuhi.

- Direktori bersama Amazon FSx mount (`/fsx`) ke klaster Anda HyperPod .
- Conda diinstal di direktori bersama FSx. Untuk mempelajari cara menginstal Conda, gunakan petunjuk di Instalasi di Linux di Panduan Pengguna Conda.
- `cuda11.8`atau `cuda12.1` diinstal pada kepala HyperPod cluster dan node komputasi Anda.

Jika semua prasyarat terpenuhi, lanjutkan ke instruksi berikut tentang meluncurkan beban kerja dengan SMP v2 pada sebuah cluster. HyperPod

1. Siapkan `smp_config.json` file yang berisi kamusthe section called “Parameter konfigurasi fitur inti SMP v2”. Pastikan Anda mengunggah file JSON ini ke tempat Anda menyimpan skrip pelatihan Anda, atau jalur yang Anda tentukan ke `torch.sagemaker.init()` modul di Langkah 1. Jika Anda telah melewati kamus konfigurasi ke `torch.sagemaker.init()` modul dalam skrip pelatihan di Langkah 1, Anda dapat melewati langkah ini.

   ```
   // smp_config.json
   {
       "hybrid_shard_degree": Integer,
       "sm_activation_offloading": Boolean,
       "activation_loading_horizon": Integer,
       "fsdp_cache_flush_warnings": Boolean,
       "allow_empty_shards": Boolean,
       "tensor_parallel_degree": Integer,
       "tensor_parallel_seed": Integer
   }
   ```

2. Unggah `smp_config.json` file ke direktori di sistem file Anda. Jalur direktori harus cocok dengan jalur yang Anda tentukan di Langkah 1. Jika Anda telah melewati kamus konfigurasi ke `torch.sagemaker.init()` modul dalam skrip pelatihan, Anda dapat melewati langkah ini.
3. Pada node komputasi HyperPod instance Anda, mulai sesi terminal dengan perintah berikut.

```
sudo su -l ubuntu
```

4. Buat lingkungan Conda pada node komputasi.

```
# Run on compute node
SMP_CUDA_VER=<11.8 or 12.1>

source /fsx/<path to miniconda>/miniconda3/bin/activate

export ENV_PATH=/fsx/<path to miniconda>/miniconda3/envs/<ENV NAME>
conda create -p ${ENV_PATH} python=3.10

conda activate ${ENV_PATH}

# Verify aws-cli is installed: Expect something like "aws-cli/2.15.0*"
aws --version
# Install aws-cli if not already installed
# https://docs.aws.amazon.com/cli/latest/userguide/getting-started-
install.html#cliv2-linux-install

conda install pytorch="2.0.1=sm_py3.10_cuda${SMP_CUDA_VER}*" packaging --override-
channels \
  -c https://sagemaker-distributed-model-parallel.s3.us-west-2.amazonaws.com/
smp-2.0.0-pt-2.0.1/2023-12-11/smp-v2/ \
  -c pytorch -c numba/label/dev \
  -c nvidia -c conda-forge

# Install dependencies of the script as below
python -m pip install packaging transformers==4.31.0 accelerate ninja tensorboard
 h5py datasets \
    && python -m pip install expecttest hypothesis \
    && python -m pip install "flash-attn>=2.0.4" --no-build-isolation

# Install SMDDP wheel (only run for cuda11.8)
SMDDP_WHL="smdistributed_dataparallel-2.0.2-cp310-cp310-linux_x86_64.whl" \
  && wget -q https://smdataparallel.s3.amazonaws.com/binary/pytorch/2.0.1/
cu118/2023-12-07/${SMDDP_WHL} \
  && pip install --force ${SMDDP_WHL} \
  && rm ${SMDDP_WHL}

# cuDNN installation for TransformerEngine installation for cuda11.8
# Please download from below link, you need to agree to terms
```

```
# https://developer.nvidia.com/downloads/compute/cudnn/secure/8.9.5/
local_installers/11.x/cudnn-linux-x86_64-8.9.5.30_cuda11-archive.tar.xz

tar xf cudnn-linux-x86_64-8.9.5.30_cuda11-archive.tar.xz \
    && rm -rf /usr/local/cuda-$SMP_CUDA_VER/include/cudnn* /usr/local/cuda-
$SMP_CUDA_VER/lib/cudnn* \
    && cp ./cudnn-linux-x86_64-8.9.5.30_cuda11-archive/include/* /usr/local/cuda-
$SMP_CUDA_VER/include/ \
    && cp ./cudnn-linux-x86_64-8.9.5.30_cuda11-archive/lib/* /usr/local/cuda-
$SMP_CUDA_VER/lib/ \
    && rm -rf cudnn-linux-x86_64-8.9.5.30_cuda11-archive.tar.xz \
    && rm -rf cudnn-linux-x86_64-8.9.5.30_cuda11-archive/

# Please download from below link, you need to agree to terms
# https://developer.download.nvidia.com/compute/cudnn/secure/8.9.7/
local_installers/12.x/cudnn-linux-x86_64-8.9.7.29_cuda12-archive.tar.xz \
# cuDNN installation for TransformerEngine installation for cuda12.1
tar xf cudnn-linux-x86_64-8.9.7.29_cuda12-archive.tar.xz \
    && rm -rf /usr/local/cuda-$SMP_CUDA_VER/include/cudnn* /usr/local/cuda-
$SMP_CUDA_VER/lib/cudnn* \
    && cp ./cudnn-linux-x86_64-8.9.7.29_cuda12-archive/include/* /usr/local/cuda-
$SMP_CUDA_VER/include/ \
    && cp ./cudnn-linux-x86_64-8.9.7.29_cuda12-archive/lib/* /usr/local/cuda-
$SMP_CUDA_VER/lib/ \
    && rm -rf cudnn-linux-x86_64-8.9.7.29_cuda12-archive.tar.xz \
    && rm -rf cudnn-linux-x86_64-8.9.7.29_cuda12-archive/

# TransformerEngine installation
export CUDA_HOME=/usr/local/cuda-$SMP_CUDA_VER
export CUDNN_PATH=/usr/local/cuda-$SMP_CUDA_VER/lib
export CUDNN_LIBRARY=/usr/local/cuda-$SMP_CUDA_VER/lib
export CUDNN_INCLUDE_DIR=/usr/local/cuda-$SMP_CUDA_VER/include
export PATH=/usr/local/cuda-$SMP_CUDA_VER/bin:$PATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-$SMP_CUDA_VER/lib

python -m pip install --no-build-isolation git+https://github.com/NVIDIA/
TransformerEngine.git@v1.0
```

5. Jalankan pekerjaan pelatihan tes.

   a. Di sistem file bersama (`/fsx`), kloning GitHub repositori Pelatihan Terdistribusi Awsome, dan buka folder. `3.test_cases/11.modelparallel`

```
git clone https://github.com/aws-samples/awsome-distributed-training/
```

```
cd awsome-distributed-training/3.test_cases/11.modelparallel
```

b. Kirim pekerjaan menggunakan `sbatch` sebagai berikut.

```
conda activate <ENV_PATH>
sbatch -N 16 conda_launch.sh
```

Jika pengiriman pekerjaan berhasil, pesan output dari `sbatch` perintah ini harus mirip `Submitted batch job ABCDEF` dengan.

c. Periksa file log di direktori saat ini di bawah`logs/`.

```
tail -f ./logs/fsdp_smp_ABCDEF.out
```

Fitur inti dari perpustakaan paralelisme SageMaker model v2

Perpustakaan paralelisme SageMaker model Amazon v2 (SMP v2) menawarkan strategi distribusi dan teknik penghematan memori, seperti paralelisme data sharded, paralelisme tensor, dan pos pemeriksaan. Strategi dan teknik paralelisme model yang ditawarkan oleh SMP v2 membantu mendistribusikan model besar di beberapa perangkat sambil mengoptimalkan kecepatan pelatihan dan konsumsi memori. SMP v2 juga menyediakan `torch.sagemaker` paket Python untuk membantu menyesuaikan skrip pelatihan Anda dengan beberapa baris perubahan kode.

Panduan ini mengikuti aliran dua langkah dasar yang diperkenalkan dithe section called “Memulai dengan SMP v2”. Untuk menyelami fitur inti SMP v2 dan cara menggunakannya, lihat topik berikut.

> **Note**
>
> Fitur inti ini tersedia di SMP v2.0.0 dan yang lebih baru dan SageMaker Python SDK v2.200.0 dan yang lebih baru, dan berfungsi untuk v2.0.1 dan yang lebih baru. PyTorch Untuk memeriksa versi paket, lihatthe section called “Kerangka kerja yang didukung dan Wilayah AWS”.

Topik

- Paralelisme data sharded hibrida
- Kompatibilitas dengan perpustakaan SMDDP yang dioptimalkan untuk infrastruktur AWS
- Pelatihan presisi campuran

- Inisialisasi parameter tertunda
- Pos pemeriksaan aktivasi
- Pembongkaran aktivasi
- Paralelisme tensor
- Penyetelan halus
- FlashAttention
- Simpan dan muat pos pemeriksaan saat menggunakan SMP

Paralelisme data sharded hibrida

Paralelisme data sharded adalah teknik pelatihan terdistribusi hemat memori yang membagi status model (parameter model, gradien, dan status pengoptimal) di seluruh perangkat. Ini membantu Anda menyesuaikan model yang lebih besar atau meningkatkan ukuran batch menggunakan memori GPU yang dibebaskan. Pustaka SMP menawarkan kemampuan menjalankan paralelisme data sharded dengan PyTorch Fully Sharded Data Parallel (FSDP). PyTorch FSDP secara default pecahan di seluruh rangkaian GPU yang digunakan. Di SMP v2, perpustakaan menawarkan paralelisme data sharded ini di atas PyTorch FSDP dengan memperluas sharding PyTorch hybrid (`HYBRID_SHARD`), yang merupakan salah satu strategi sharding yang disediakan oleh FSDP:,,,. PyTorch `FULL_SHARD` `SHARD_GRAD_OP` `HYBRID_SHARD` `_HYBRID_SHARD_ZERO2` Memperluas sharding hibrida dengan cara ini membantu mengimplementasikan scale-aware-sharding seperti yang dijelaskan dalam blog Penskalaan linier dekat pelatihan model raksasa untuk FSDP. AWS PyTorch

Pustaka SMP membuatnya mudah digunakan `HYBRID_SHARD` dan `_HYBRID_SHARD_ZERO2` di sejumlah GPU yang dapat dikonfigurasi, memperluas PyTorch FSDP asli yang mendukung sharding di satu node () atau semua GPU ()`HYBRID_SHARD`. `FULL_SHARD` PyTorch Panggilan FSDP dapat tetap apa adanya, dan Anda hanya perlu menambahkan `hybrid_shard_degree` argumen ke konfigurasi SMP, seperti yang ditunjukkan pada contoh kode berikut. Anda tidak perlu mengubah nilai `sharding_strategy` argumen dalam pembungkus PyTorch FSDP di sekitar model Anda. PyTorch Anda dapat lulus `ShardingStrategy.HYBRID_SHARD` sebagai nilainya. Atau, pustaka SMP mengganti strategi dalam skrip dan menyetelnya `ShardingStrategy.HYBRID_SHARD` jika Anda menentukan nilai yang sama dengan atau lebih besar dari 2 ke parameter. `hybrid_shard_degree`

Cuplikan kode berikut menunjukkan cara menambahkan modul inisialisasi SMP `torch.sagemaker.init()` ke skrip pelatihan Anda dan mengatur kamus konfigurasi SMP dalam format JSON untuk peluncur pekerjaan pelatihan sambil mengikuti proses dua langkah yang diperkenalkan. the section called “Memulai dengan SMP v2” Anda tidak perlu membuat perubahan

apa pun pada PyTorch model atau konfigurasi PyTorch FSDP Anda. Untuk informasi tentang parameter `hybrid_shard_degree`, lihat the section called “Parameter konfigurasi fitur inti SMP v2”.

Kamus konfigurasi SMP

```
{ "hybrid_shard_degree": 16 }
```

Dalam naskah pelatihan

```
import torch.sagemaker as tsm
tsm.init()

# Set up a PyTorch model
model = ...

# Wrap the PyTorch model using the PyTorch FSDP module
model = FSDP(
    model,
    ...
)

# Optimizer needs to be created after FSDP wrapper
optimizer = ...
```

Kompatibilitas dengan perpustakaan SMDDP yang dioptimalkan untuk infrastruktur AWS

Anda dapat menggunakan perpustakaan paralelisme SageMaker model v2 (SMP v2) bersama dengan perpustakaan paralelisme data SageMaker terdistribusi (SMDDP) yang menawarkan operasi komunikasi kolektif yang dioptimalkan untuk infrastruktur. `AllGather` AWS Dalam pelatihan terdistribusi, operasi komunikasi kolektif dirancang untuk menyinkronkan beberapa pekerja GPU dan bertukar informasi di antara mereka. `AllGather`adalah salah satu operasi komunikasi kolektif inti yang biasanya digunakan dalam paralelisme data sharded. Untuk mempelajari lebih lanjut tentang operasi SMDDP, lihat the section called “Operasi kolektif SMDDP `AllGather`” Mengoptimalkan `AllGather` operasi komunikasi kolektif semacam itu akan secara langsung berkontribusi pada end-to-end pelatihan yang lebih cepat tanpa efek samping pada konvergensi.

> **Note**
>
> Pustaka SMDDP mendukung instance P4 dan P4DE (lihat juga the section called “Kerangka kerja yang didukung,Wilayah AWS, dan tipe instance” oleh perpustakaan SMDDP).

Pustaka SMDDP terintegrasi secara native dengan PyTorch melalui lapisan grup proses. Untuk menggunakan perpustakaan SMDDP, Anda hanya perlu menambahkan dua baris kode ke skrip pelatihan Anda. Ini mendukung kerangka kerja pelatihan seperti SageMaker Model Parallelism Library, PyTorch FSDP, dan. DeepSpeed

Untuk mengaktifkan SMDDP dan menggunakan `AllGather` operasinya, Anda perlu menambahkan dua baris kode ke skrip pelatihan Anda sebagai bagian dari. the section called “Langkah 1: Sesuaikan skrip pelatihan PyTorch FSDP Anda” Perhatikan bahwa Anda perlu menginisialisasi PyTorch Distributed dengan backend SMDDP terlebih dahulu, dan kemudian menjalankan inisialisasi SMP.

```
import torch.distributed as dist

# Initialize with SMDDP
import smdistributed.dataparallel.torch.torch_smddp
dist.init_process_group(backend="smddp") # Replacing "nccl"

 # Initialize with SMP
import torch.sagemaker as tsm
tsm.init()
```

SageMaker Framework Containers untuk PyTorch (lihat juga the section called “Kerangka kerja yang didukung dan Wilayah AWS” oleh SMP v2 dan the section called “Kerangka kerja yang didukung,Wilayah AWS, dan tipe instance” oleh perpustakaan SMDDP) sudah dikemas sebelumnya dengan biner SMP dan biner SMDDP. Untuk mempelajari lebih lanjut tentang perpustakaan SMDDP, lihat. the section called “SageMaker perpustakaan paralelisme data terdistribusi”

Pelatihan presisi campuran

SMP v2 mendukung PyTorch `MixedPrecisionFSDP`. PyTorch FSDP menyediakan berbagai konfigurasi untuk presisi campuran untuk peningkatan kinerja dan pengurangan memori.
Cara standar untuk mengonfigurasi model untuk presisi campuran adalah dengan membuat model`float32`, dan kemudian mengizinkan FSDP untuk mentransmisikan parameter ke `float16` atau dengan cepat dengan meneruskan `MixedPrecision` kebijakan seperti yang ditunjukkan `bfloat16` pada cuplikan kode berikut. Untuk informasi selengkapnya tentang opsi untuk mengubah parameter, reduksi, atau buffer untuk presisi campuran PyTorch, lihat PyTorch FSDP *`MixedPrecision`* API dalam dokumentasi. *`dtype`* PyTorch

```
# Native PyTorch API
```

```
from torch.distributed.fsdp import MixedPrecision

dtype = torch.bfloat16
mixed_precision_policy = MixedPrecision(
    param_dtype=dtype, reduce_dtype=dtype, buffer_dtype=dtype
)

model = FSDP(
    model,
    ...,
    mixed_precision=mixed_precision_policy
)
```

Perhatikan bahwa model tertentu (seperti model Hugging Face Transformers Llama) mengharapkan buffer sebagai. `float32` Untuk menggunakan`float32`, ganti `torch.bfloat16` dengan `torch.float32` di baris yang mendefinisikan `dtype` objek.

Inisialisasi parameter tertunda

Inisialisasi model besar untuk pelatihan tidak selalu dimungkinkan dengan memori GPU yang terbatas. Untuk mengatasi masalah memori GPU yang tidak mencukupi ini, Anda dapat menginisialisasi model pada memori CPU. Namun, untuk model yang lebih besar dengan lebih dari 20 atau 40 miliar parameter, bahkan memori CPU mungkin tidak cukup. Untuk kasus seperti itu, kami menyarankan Anda menginisialisasi model pada apa yang PyTorch disebut perangkat meta, yang memungkinkan pembuatan tensor tanpa data apa pun yang melekat padanya. Tensor pada perangkat meta hanya membutuhkan informasi bentuk, dan ini memungkinkan untuk membuat model besar dengan parameternya pada perangkat meta. Hugging Face Accelerate menyediakan `init_empty_weights` manajer konteks untuk membantu membuat model seperti itu pada perangkat meta sambil menginisialisasi buffer pada perangkat biasa. Sebelum pelatihan dimulai, PyTorch FSDP menginisialisasi parameter model. Fitur inisialisasi parameter tertunda SMP v2 ini menunda pembuatan parameter model ini terjadi setelah PyTorch FSDP melakukan sharding parameter. PyTorch FSDP menerima fungsi inisialisasi parameter (`param_init_fn`) saat sharding modul, dan memanggil setiap modul. `param_init_fn` `param_init_fn`API mengambil modul sebagai argumen dan menginisialisasi semua parameter di dalamnya, tidak termasuk parameter modul anak mana pun. Perhatikan bahwa perilaku ini berbeda dari PyTorch v2.0.1 asli yang memiliki bug yang menyebabkan parameter diinisialisasi beberapa kali.

SMP v2 menyediakan the section called “`torch.sagemaker.delayed_param.DelayedParamIniter`” API untuk menerapkan inisialisasi parameter tertunda.

Cuplikan kode berikut menunjukkan cara menerapkan `torch.sagemaker.delayed_param.DelayedParamIniter` API ke skrip pelatihan Anda.

Asumsikan bahwa Anda memiliki skrip pelatihan Pytorch FSDP sebagai berikut.

```
# Creation of model on meta device
from accelerate import init_empty_weights
with init_empty_weights():
    model = create_model()

# Define a param init fn, below is an example for Hugging Face GPTNeoX.
def init_weights(module):
    d = torch.cuda.current_device()
    # Note that below doesn't work if you have buffers in the model
    # buffers will need to reinitialized after this call
    module.to_empty(device=d, recurse=False)
    if isinstance(module, (nn.Linear, Conv1D)):
        module.weight.data.normal_(mean=0.0, std=args.initializer_range)
        if module.bias:
            module.bias.data.zero_()
    elif isinstance(module, nn.Embedding):
        module.weight.data.normal_(mean=0.0, std=args.initializer_range)
        if module.padding_idx:
            module.weight.data[module.padding_idx].zero_()
    elif isinstance(module, nn.LayerNorm):
        module.bias.data.zero_()
        module.weight.data.fill_(1.0)

# Changes to FSDP wrapper.
model = FSDP(
    model,
    ...,
    param_init_fn=init_weights
)

# At this point model is initialized and sharded for sharded data parallelism.
```

Perhatikan bahwa pendekatan inisialisasi parameter tertunda bukanlah model agnostik. Untuk mengatasi masalah ini, Anda perlu menulis `init_weights` fungsi seperti yang ditunjukkan pada contoh sebelumnya agar sesuai dengan inisialisasi dalam definisi model asli, dan itu harus mencakup semua parameter model. Untuk menyederhanakan proses persiapan `init_weights` fungsi tersebut, SMP v2 mengimplementasikan fungsi inisialisasi ini untuk

model berikut: GPT-2, GPT-J, GPTNeox, dan Llama dari Hugging Face Transformers. `torch.sagemaker.delayed_param.DelayedParamIniter`API juga berfungsi dengan implementasi paralel tensor SMP, `torch.sagemaker.tensor_parallel.transformer.TransformerLMHead` model, yang dapat Anda panggil setelah panggilan the section called “`torch.sagemaker.transform`” API.

Menggunakan `torch.sagemaker.delayed_param.DelayedParamIniter` API, Anda dapat menyesuaikan skrip PyTorch FSDP Anda sebagai berikut. Setelah membuat model dengan bobot kosong, daftarkan `torch.sagemaker.delayed_param.DelayedParamIniter` API ke model, dan tentukan objeknya. Lewati objek ke `param_init_fn` kelas PyTorch FSDP.

```
from torch.sagemaker.delayed_param import DelayedParamIniter
from accelerate import init_empty_weights

with init_empty_weights():
    model = create_model()

delayed_initer = DelayedParamIniter(model)

with delayed_initer.validate_params_and_buffers_inited():
    model = FSDP(
        model,
        ...,
        param_init_fn=delayed_initer.get_param_init_fn()
    )
```

Catatan tentang bobot yang diikat

Saat melatih model dengan bobot terikat, kita perlu berhati-hati untuk mengikat bobot setelah menginisialisasi bobot dengan inisialisasi parameter yang tertunda. PyTorchFSDP tidak memiliki mekanisme untuk mengikat bobot setelah menginisialisasi mereka menggunakan seperti di atas. `param_init_fn` Untuk mengatasi kasus seperti itu, kami menambahkan API untuk mengizinkan a`post_init_hook_fn`, yang dapat digunakan untuk mengikat bobot. Anda dapat meneruskan fungsi apa pun di sana yang menerima modul sebagai argumen, tetapi kami juga memiliki standar yang `post_param_init_fn` ditentukan di `DelayedParamIniter` mana `tie_weights` metode panggilan modul jika ada. Perhatikan bahwa aman untuk selalu masuk `post_param_init_fn` meskipun tidak ada `tie_weights` metode untuk modul.

```
with delayed_initer.validate_params_and_buffers_inited():
    model = FSDP(
```

```
        model,
        ...,
        param_init_fn=delayed_initer.get_param_init_fn(),
        post_param_init_fn=delayed_initer.get_post_param_init_fn()
    )
```

Pos pemeriksaan aktivasi

Checkpointing aktivasi adalah teknik untuk mengurangi penggunaan memori dengan membersihkan aktivasi lapisan tertentu dan mengkomputernya kembali selama backward pass. Secara efektif, ini memperdagangkan waktu komputasi ekstra untuk mengurangi penggunaan memori. Jika modul diperiksa, di akhir pass maju, hanya input awal ke modul dan output akhir dari modul yang tetap berada di memori. PyTorch melepaskan tensor perantara apa pun yang merupakan bagian dari perhitungan di dalam modul itu selama pass maju. Selama lintasan mundur modul checkpoint, PyTorch hitung ulang tensor ini. Pada titik ini, lapisan di luar modul checkpointed ini telah menyelesaikan backward pass mereka, sehingga penggunaan memori puncak dengan checkpointing menjadi lebih rendah.

SMP v2 mendukung modul checkpointing PyTorch aktivasi,. [apply_activation_checkpointing](#) Berikut ini adalah contoh checkpointing aktivasi model Hugging Face GPT-Neox.

Lapisan Checkpointing Transformer dari model Hugging Face GPT-Neox

```
from transformers.models.gpt_neox import GPTNeoXLayer
from torch.distributed.algorithms._checkpoint.checkpoint_wrapper import (
    apply_activation_checkpointing
)

# check_fn receives a module as the arg,
# and it needs to return whether the module is to be checkpointed
def is_transformer_layer(module):
    from transformers.models.gpt_neox import GPTNeoXLayer
    return isinstance(submodule, GPTNeoXLayer)

apply_activation_checkpointing(model, check_fn=is_transformer_layer)
```

Checkpointing setiap lapisan Transformer lainnya dari model Hugging Face GPT-Neox

```
# check_fn receives a module as arg,
# and it needs to return whether the module is to be checkpointed
```

```
# here we define that function based on global variable (transformer_layers)
from transformers.models.gpt_neox import GPTNeoXLayer
from torch.distributed.algorithms._checkpoint.checkpoint_wrapper import (
    apply_activation_checkpointing
)

transformer_layers = [
    m for m model.modules() if isinstance(m, GPTNeoXLayer)
]

def is_odd_transformer_layer(module):
    return transformer_layers.index(module) % 2 == 0

apply_activation_checkpointing(model, check_fn=is_odd_transformer_layer)
```

Atau, PyTorch juga memiliki `torch.utils.checkpoint` modul untuk checkpointing, yang digunakan oleh subset model Hugging Face Transformers. Modul ini juga bekerja dengan SMP v2. Namun, ini mengharuskan Anda untuk memiliki akses ke definisi model untuk menambahkan pembungkus pos pemeriksaan. Karena itu, kami sarankan Anda untuk menggunakan `apply_activation_checkpointing` metode ini.

Pembongkaran aktivasi

Biasanya, pass maju menghitung aktivasi pada setiap lapisan dan menyimpannya dalam memori GPU sampai pass mundur untuk lapisan yang sesuai selesai. Membongkar tensor ini ke memori CPU setelah forward pass dan mengambilnya kembali ke GPU saat dibutuhkan untuk backward pass lapisan dapat menghemat penggunaan memori GPU yang substanal. PyTorchmendukung aktivasi pembongkaran, tetapi implementasinya menyebabkan GPU menjadi idle sementara aktivasi diambil kembali dari CPU selama backward pass. Hal ini menyebabkan penurunan kinerja yang besar saat menggunakan pembongkaran aktivasi.

SMP v2 meningkatkan pembongkaran aktivasi ini, dan melakukan pra-pengambilan aktivasi sebelumnya sebelum aktivasi diperlukan agar GPU mulai meneruskan aktivasi tersebut secara mundur. Fitur pra-pengambilan ini membantu kemajuan pelatihan berjalan lebih efisien tanpa GPU idle, yang menghasilkan menawarkan manfaat dari penggunaan memori yang lebih rendah tanpa penurunan kinerja.

Anda dapat menyimpan PyTorch modul asli untuk pembongkaran aktivasi dalam skrip pelatihan Anda. Berikut ini adalah contoh struktur penerapan fitur pembongkaran aktivasi SMP di skrip Anda. Perhatikan bahwa pembongkaran aktivasi hanya berlaku jika digunakan bersama denganthe

section called “Pos pemeriksaan aktivasi”. Untuk mempelajari lebih lanjut tentang alat PyTorch pos pemeriksaan asli untuk pembongkaran aktivasi, lihat juga checkpoint_wrapper.py di PyTorch GitHubrepositori dan Activation Checkpointing di PyTorch blog Scaling Multi-modal Foundation Models in with Distributed. TorchMultimodal PyTorch

Untuk menerapkan fitur pembongkaran aktivasi SMP pada pos pemeriksaan PyTorch aktivasi, tambahkan `sm_activation_offloading` dan `activation_loading_horizon` parameter ke kamus konfigurasi SMP selama. the section called “Langkah 2: Luncurkan pekerjaan pelatihan”

Cuplikan kode berikut menunjukkan cara menambahkan modul inisialisasi SMP `torch.sagemaker.init()` ke skrip pelatihan Anda dan mengatur kamus konfigurasi SMP dalam format JSON untuk peluncur pekerjaan pelatihan sambil mengikuti proses dua langkah yang diperkenalkan. the section called “Memulai dengan SMP v2” Anda tidak perlu membuat perubahan apa pun pada PyTorch model atau konfigurasi PyTorch FSDP Anda. Untuk informasi tentang parameter `sm_activation_offloading` dan `activation_loading_horizon`, lihat the section called “Parameter konfigurasi fitur inti SMP v2”.

Konfigurasi SMP

```
{
    "activation_loading_horizon": 2,
    "sm_activation_offloading": True
}
```

Dalam naskah pelatihan

> **Note**
>
> Saat mengaktifkan fitur pembongkaran aktivasi SMP, pastikan Anda juga menggunakan PyTorch `offload_wrapper` fungsi tersebut dan menerapkannya ke modul root. Fitur pembongkaran aktivasi SMP menggunakan modul root untuk menentukan kapan pass maju dilakukan untuk memulai pra-pengambilan.

```
import torch.sagemaker as tsm
tsm.init()

# Native PyTorch module for activation offloading
from torch.distributed.algorithms._checkpoint.checkpoint_wrapper import (
```

```
    apply_activation_checkpointing,
    offload_wrapper,
)

model = FSDP(...)

# Activation offloading requires activation checkpointing.
apply_activation_checkpointing(
    model,
    check_fn=checkpoint_transformer_layers_policy,
)

model = offload_wrapper(model)
```

Paralelisme tensor

Paralelisme tensor adalah jenis paralelisme model di mana bobot model tertentu, gradien, dan status pengoptimal dibagi di seluruh perangkat. Berbeda dengan paralelisme pipa, yang menjaga bobot individu tetap utuh tetapi mempartisi set bobot, gradien, atau pengoptimal di seluruh perangkat, paralelisme tensor memecah bobot individu. Ini biasanya melibatkan komputasi terdistribusi dari operasi tertentu, modul, atau lapisan model.

Paralelisme tensor diperlukan dalam kasus di mana satu parameter menghabiskan sebagian besar memori GPU (seperti tabel penyematan besar dengan ukuran kosakata besar atau lapisan softmax besar dengan sejumlah besar kelas). Dalam hal ini, memperlakukan tensor atau operasi besar ini sebagai unit atom tidak efisien dan menghambat keseimbangan beban memori.

SMP v2 terintegrasi dengan Transformer Engine untuk implementasi paralelisme tensor, dan berjalan di atas API FSDP. PyTorch Anda dapat mengaktifkan paralelisme tensor PyTorch FSDP dan SMP secara bersamaan, dan menentukan paralelisme model terbaik untuk kinerja terbaik. Perhatikan bahwa SMP v2 saat ini tidak mendukung tipe `float8` data karena ini adalah fitur yang hilang di PyTorch FSDP.

Dalam praktiknya, paralelisme tensor sangat membantu dalam skenario berikut.

- Saat berlatih dengan panjang konteks yang panjang karena itu mengarah ke memori aktivasi tinggi dengan FSDP saja.
- Saat berlatih dengan cluster yang sangat besar di mana ukuran batch global melebihi batas yang diinginkan.

Model Hugging Face Transformer kompatibel dengan paralelisme tensor SMP

SMP v2 saat ini menawarkan dukungan paralelisme tensor untuk model Hugging Face Transformer berikut.

- GPT-Neox
- Llama 2

Untuk konfigurasi referensi untuk menerapkan paralelisme tensor pada model ini, lihat. the section called “Kiat konfigurasi”

Konfigurasikan paralelisme tensor

Untuk`tensor_parallel_degree`, Anda memilih nilai untuk tingkat paralelisme tensor. Nilai harus membagi jumlah GPU di cluster Anda secara merata. Misalnya, untuk menghancurkan model Anda saat menggunakan instance dengan 8 GPU, pilih 2, 4, atau 8. Kami menyarankan Anda memulai dengan jumlah kecil, dan secara bertahap meningkatkannya hingga model sesuai dengan memori GPU.

SMP v2 saat ini tidak mendukung tipe data Float8 karena ini adalah fitur yang hilang di FSDP. PyTorch

Cuplikan kode berikut menunjukkan cara menambahkan modul inisialisasi SMP `torch.sagemaker.init()` ke skrip pelatihan Anda dan mengatur kamus konfigurasi SMP dalam format JSON untuk peluncur pekerjaan pelatihan sambil mengikuti proses dua langkah yang diperkenalkan. the section called “Memulai dengan SMP v2” Anda tidak perlu membuat perubahan apa pun pada PyTorch model atau konfigurasi PyTorch FSDP Anda. Untuk informasi tentang parameter `tensor_parallel_degree` dan `tensor_parallel_seed`, lihat the section called “Parameter konfigurasi fitur inti SMP v2”.

Konfigurasi SMP

```
{
    "tensor_parallel_degree": 8,
    "tensor_parallel_seed": 0
}
```

Dalam skrip pelatihan Anda

Inisialisasi dengan `torch.sagemaker.init()` untuk mengaktifkan SMP v2 dan bungkus model Anda dengan API. the section called “`torch.sagemaker.transform`”

```
import torch.sagemaker as tsm
tsm.init()

from transformers import AutoModelForCausalLM
model = AutoModelForCausalLM.from_config(..)
model = tsm.transform(model)
```

Menyimpan dan memuat pos pemeriksaan Hugging Face Transformer

Setelah perpustakaan SMP mengubah model, ia mengubah state dictionary (`state_dict`) model. Ini berarti bahwa model menjadi tidak kompatibel dengan fungsi checkpointing Hugging Face Transformer asli. Untuk menangani hal ini, pustaka SMP menyediakan API untuk menyimpan pos pemeriksaan dari model yang diubah dalam representasi Hugging Face Transformer, dan `torch.sagemaker.transform` API untuk memuat pos pemeriksaan model Hugging Face Transformer untuk fine-tuning.

Untuk informasi lebih lanjut tentang menyimpan pos pemeriksaan saat menggunakan fitur paralelisme tensor SMP v2, lihat. the section called “Simpan dan muat pos pemeriksaan saat menggunakan SMP”

Untuk informasi lebih lanjut tentang menyempurnakan model yang menerapkan fitur paralelisme tensor SMP v2, lihat. the section called “Penyetelan halus”

Penyetelan halus

Fine-tuning adalah proses pelatihan terus menerus model pra-terlatih untuk meningkatkan kinerja untuk kasus penggunaan tertentu.

Menyesuaikan model kecil yang pas sepenuhnya pada satu GPU, atau yang sesuai dengan 8 salinan model sepenuhnya pada CPU sangatlah mudah. Tidak memerlukan perubahan khusus pada pelatihan FSDP reguler. Di ranah model yang lebih besar dari ini, Anda perlu mempertimbangkan untuk menggunakan fungsionalitas inisialisasi parameter tertunda, yang bisa jadi rumit.

Untuk mengatasi hal ini, pustaka SMP memuat model lengkap di salah satu peringkat sementara peringkat lainnya membuat model dengan bobot kosong pada perangkat meta. Kemudian, PyTorch FSDP menginisialisasi bobot pada peringkat bukan nol menggunakan `init_weights` fungsi, dan menyinkronkan bobot pada semua peringkat ke bobot pada peringkat ke-0 dengan set ke.

`sync_module_states` `True` Cuplikan kode berikut menunjukkan bagaimana Anda harus mengaturnya dalam skrip pelatihan Anda.

```
import torch.distributed as dist
from transformers import AutoModelForCasalLM
from accelerate import init_empty_weights
from torch.sagemaker.delayed_param import DelayedParamIniter

if dist.get_rank() == 0:
    model = AutoModelForCasalLM.from_pretrained(..., low_cpu_mem_usage=True)
else:
    with init_empty_weights():
        model = AutoModelForCasalLM.from_config(AutoConfig.from_pretrained(...))
    delayed_initer = DelayedParamIniter(model)

model = FSDP(
    model,
    ...,
    sync_module_states=True,
    param_init_fn=delayed_initer.get_param_init_fn() if dist.get_rank() > 0 else None
)
```

Menyetel model Hugging Face Transformer yang telah dilatih sebelumnya dengan paralelisme tensor SMP

Bagian ini membahas pemuatan model Transformer untuk dua kasus penggunaan: fine-tuning model Transformer kecil dan fine-tuning model Transformer besar. Untuk model yang lebih kecil tanpa inisialisasi parameter yang tertunda, bungkus model dengan `torch.sagemaker.transform` API sebelum membungkusnya dengan PyTorch FSDP.

```
import functools
from transformers import AutoModelForCausalLM
from torch.distributed.fsdp import FullyShardedDataParallel as FSDP
from torch.distributed.fsdp.wrap import transformer_auto_wrap_policy
from torch.sagemaker import transform

model = AutoModelForCausalLM.from_pretrained("meta-llama/Llama-2-7b-hf",
 low_cpu_mem_usage=True)

# Transform model while loading state dictionary from rank 0.
tp_model = transform(model, load_state_dict_from_rank0=True)
```

```
# Wrap with FSDP.
model = FSDP(
    tp_model,
    ...
    sync_module_states=True,
)
```

Untuk model yang lebih besar, pendekatan sebelumnya menyebabkan kehabisan memori CPU. Kami menyarankan Anda menggunakan inisialisasi parameter tertunda untuk menghindari masalah memori CPU tersebut. Dalam hal ini, Anda dapat menerapkan `torch.sagemaker.transform` API dan `torch.sagemaker.delayed_param.DelayedParamIniter` API seperti yang ditunjukkan pada contoh kode berikut.

```
from transformers import AutoModelForCausalLM
from torch.sagemaker import transform
from torch.sagemaker.delayed_param import DelayedParamIniter

# Create one instance of model without delayed param
# on CPU, on one rank.
if dist.get_rank() == 0:
    model = AutoModelForCasalLM.from_pretrained(...,low_cpu_mem_usage=True)
else:
    with init_empty_weights():
        model = AutoModelForCasalLM.from_config(AutoConfig.from_pretrained(...))

# Transform model while loading state dictionary from rank 0
model = transform(model, load_state_dict_from_rank0=True)

if dist.get_rank() != 0: # For fine-tuning, delayed parameter on non-zero ranks
    delayed_initer = DelayedParamIniter(model)
else:
    delayed_initer = None

with (
    delayed_initer.validate_params_and_buffers_inited() if delayed_initer else
 nullcontext()
):
    # Wrap the model with FSDP
    model = FSDP(
        model,
        ...,
        sync_module_states=True,
        param_init_fn=delayed_initer.get_param_init_fn() if delayed_initer else None
```

```
    )
```

FlashAttention

SMP v2 mendukung FlashAttentionkernel dan membuatnya mudah untuk menerapkannya ke berbagai skenario untuk model Hugging Face Transformer. Perhatikan bahwa jika Anda menggunakan FlashAttention paket v2.0 atau yang lebih baru, SMP menggunakan FlashAttention v2; Namun, perhatian flash Triton default ke kernel perhatian flash di FlashAttention v1.x, membuatnya didukung secara eksklusif di v1. FlashAttention

Module (`nn.Module`) adalah API tingkat rendah yang mendefinisikan lapisan perhatian model. Ini harus diterapkan tepat setelah pembuatan model, dari `AutoModelForCausalLM.from_config()` API misalnya, dan sebelum model diubah atau dibungkus dengan FSDP.

Gunakan FlashAttention kernel untuk perhatian diri

Cuplikan kode berikut menunjukkan cara menggunakan the section called "`torch.sagemaker.nn.attn.FlashSelfAttention`" API yang disediakan oleh SMP v2.

```
def new_attn(self, q, k, v, attention_mask=None, head_mask=None):
    return (
        self.flashmod((q, k, v), causal=True, cast_dtype=torch.bfloat16, layout="b h s
 d"),
        None,
    )

for layer in model.gpt_neox.layers:
    layer.attention.flash_mod = torch.sagemaker.nn.attn.FlashSelfAttention()
    layer.attention._attn = functools.partial(new_attn, layer.attention)
```

Gunakan FlashAttention kernel untuk perhatian kueri yang dikelompokkan

SMP v2 juga mendukung FlashAttentionkernel untuk grouped-query attention (GQA) dan membuatnya mudah untuk menerapkannya ke berbagai skenario untuk model Hugging Face Transformer. Berbeda dari arsitektur perhatian asli, GQA sama-sama mempartisi kepala kueri ke dalam grup, dan kepala kueri dalam grup yang sama berbagi kunci dan kepala nilai yang sama. Oleh karena itu, kepala q dan kv diteruskan ke panggilan maju secara terpisah. Catatan: Jumlah kepala q harus habis dibagi dengan jumlah kepala kv.

Contoh penggunaan FlashGroupedQueryAttention

Cuplikan kode berikut menunjukkan cara menggunakan the section called "`torch.sagemaker.nn.attn.FlashGroupedQueryAttention`" API yang disediakan oleh SMP v2.

```
from transformers.models.llama.modeling_llama import LlamaAttention
from torch.sagemaker.nn.attn import FlashGroupedQueryAttention

class LlamaFlashAttention(LlamaAttention):
    def __init__(self, config: LlamaConfig):
        super().__init__(config)

        self.flash_attn = FlashGroupedQueryAttention(
            attention_dropout_prob=0.0,
        )

    def forward(
        self,
        hidden_states: torch.Tensor,
        attention_mask: Optional[torch.Tensor] = None,
        position_ids: Optional[torch.LongTensor] = None,
        ...
    ):
        query_states = self.q_proj(hidden_states)
        key_states = self.k_proj(hidden_states)
        value_states = self.v_proj(hidden_states)
        ...
        kv = (key_states, value_states)
        attn_output = self.flash_attn(
            query_states,
            kv,
            attn_mask=attention_mask,
            causal=True,
            layout="b h s d",
        )
        ...
        attn_output = self.o_proj(attn_output)
        ...
        return attn_output
```

Pustaka SMP juga menyediakanthe section called "`torch.sagemaker.nn.huggingface.llama_flashattn.LlamaFlashAttention`", yang menggunakan the section called

"torch.sagemaker.nn.attn.FlashGroupedQueryAttention" API pada tingkat rendah. Hugging Face Transformers memiliki LlamaFlashAttention2implementasi serupa yang disebut dari v4.36.0. Cuplikan kode berikut menunjukkan cara menggunakan API SMP v2 atau Transformers `LlamaFlashAttention` `LlamaFlashAttention2` API untuk mengganti lapisan perhatian model Llama yang ada.

```
from torch.sagemaker.nn.huggingface.llama_flashattn import LlamaFlashAttention
from transformers.models.llama.modeling_llama import LlamaFlashAttention2

flash_attn_class = LlamaFlashAttention # or flash_attn_class = LlamaFlashAttention2

attn_name = "self_attn"
for layer in model.model.layers:
    prev_layer = getattr(layer, attn_name)
    setattr(layer, attn_name, flash_attn_class(model.config))
```

Simpan dan muat pos pemeriksaan saat menggunakan SMP

Pustaka SMP mendukung PyTorch API untuk pos pemeriksaan, dan menyediakan API yang membantu pos pemeriksaan dengan benar saat menggunakan pustaka SMP.

PyTorch FSDP mendukung tiga jenis pos pemeriksaan: penuh, sharded dan lokal. Ini melayani tujuan yang berbeda. Pos pemeriksaan penuh idealnya hanya digunakan saat mengekspor model setelah pelatihan selesai, karena mahal untuk menghasilkan pos pemeriksaan penuh. Pos pemeriksaan sharded adalah pendekatan yang direkomendasikan untuk menyimpan dan memuat pos pemeriksaan selama pelatihan. Menggunakan pos pemeriksaan sharded, Anda juga dapat mengubah ukuran cluster saat melanjutkan pelatihan. Pos pemeriksaan lokal lebih ketat. Dengan pos pemeriksaan lokal, Anda perlu melanjutkan pelatihan dengan jumlah GPU yang sama dan saat ini tidak didukung saat menggunakan paralelisme tensor dengan SMP. Perhatikan bahwa pos pemeriksaan oleh FSDP memerlukan penulisan ke sistem file jaringan bersama, seperti FSx.

Pos pemeriksaan sharded

Prosedur berikut menyoroti apa yang perlu Anda lakukan untuk menyesuaikan skrip pelatihan Anda untuk menyimpan dan memuat pos pemeriksaan sharded dengan atau tanpa fitur paralelisme tensor SMP.

1. Impor `torch.sagemaker` paket SMP.

```
import torch.sagemaker as tsm
```

2. Siapkan variabel tambahan untuk menyimpan dan memuat pos pemeriksaan.

   a. Siapkan peringkat koordinator untuk melakukan operasi kolektif komunikatif seperti. `AllReduce`

   ```
   coordinator_rank: int = min(dist.get_process_group_ranks(model.process_group))
   ```

   b. Menggunakan `torch.sagemaker.state` enumerasi, atur peringkat tindakan untuk menentukan apakah akan membiarkan peringkat mengambil bagian dalam checkpointing. Dan tambahkan pernyataan if untuk menyimpan pos pemeriksaan tergantung pada penggunaan paralelisme tensor SMP v2.

   ```
   action_rank: bool = global_rank < (tsm.state.hybrid_shard_degree *
    tsm.state.tp_size)

   if tsm.state.tp_size > 1:
       # Tensor parallel groups will have their own sub directories.
       sub_dir = f"tp{tsm.state.tp_size}-{tsm.state.tp_rank}"
   else:
       sub_dir = ""
   ```

3. Tetap gunakan API pos pemeriksaan PyTorch FSDP apa adanya.

Contoh kode berikut menunjukkan skrip pelatihan PyTorch FSDP lengkap dengan API pos pemeriksaan FSDP.

```
import torch.distributed as dist
from torch.distributed.checkpoint.optimizer import (
    load_sharded_optimizer_state_dict
)
from torch.distributed.fsdp import (
    FullyShardedDataParallel as FSDP,
    StateDictType
)
import torch.sagemaker as tsm

sharding_strategy, state_dict_type = ..., ...
global_rank = dist.get_rank()

# 0. Auxiliary variables to save and load checkpoints.

# Used when performing comm collectives such as allreduce.
```

```
coordinator_rank: int = min(dist.get_process_group_ranks(model.process_group))

# To determine whether to take part in checkpointing.
action_rank: bool = global_rank < (tsm.state.hybrid_shard_degree * tsm.state.tp_size)

if tsm.state.tp_size > 1:
    # Tensor parallel groups will have their own sub directories.
    sub_dir = f"tp{tsm.state.tp_size}-{tsm.state.tp_rank}"
else:
    sub_dir = ""

# 1. Save checkpoints.
with FSDP.state_dict_type(model, StateDictType.SHARDED_STATE_DICT):
    state_dict = {
       "model": model.state_dict(),
       "optimizer": FSDP.optim_state_dict(model, optimizer),
        # Potentially add more customized state dicts.
    }

 # Save from one single replication group.
 if action_rank:
    dist.checkpoint.save_state_dict(
        state_dict=state_dict,
        storage_writer=dist.checkpoint.FileSystemWriter(os.path.join(save_dir,
 sub_dir)),
        process_group=model.process_group,
        coordinator_rank=coordinator_rank,
    )

# 2. Load checkpoints.
with FSDP.state_dict_type(model, StateDictType.SHARDED_STATE_DICT):
    # 2.1 Load model and everything else except the optimizer.
    state_dict = {
        # All states except optimizer state can be passed here.
        "model": model.state_dict()
    }

    dist.checkpoint.load_state_dict(
        state_dict=state_dict,
        storage_reader=dist.checkpoint.FileSystemReader(os.path.join(load_dir,
 sub_dir)),
        process_group=model.process_group,
        coordinator_rank=coordinator_rank,
    )
```

```
    model.load_state_dict(state_dict["model"])
    # Potentially process more customized and non-optimizer dict states.

    # 2.2 Load optimizer.
    optim_state = load_sharded_optimizer_state_dict(
        model_state_dict=state_dict["model"],
        optimizer_key="optimizer",
        storage_reader=dist.checkpoint.FileSystemReader(os.path.join(load_dir,
sub_dir)),
        process_group=model.process_group,
    )
    flattened_optimizer_state = FSDP.optim_state_dict_to_load(
        optim_state["optimizer"], model, optimizer, group=model.process_group,
    )
    optimizer.load_state_dict(flattened_optimizer_state)
```

Pos pemeriksaan model lengkap

Di akhir pelatihan, Anda dapat menyimpan pos pemeriksaan lengkap yang menggabungkan semua pecahan model ke dalam satu file pos pemeriksaan model. Pustaka SMP sepenuhnya mendukung API pos pemeriksaan model PyTorch lengkap, jadi Anda tidak perlu melakukan perubahan apa pun.

Perhatikan bahwa jika Anda menggunakan SMPthe section called “Paralelisme tensor”, perpustakaan SMP mengubah model. Saat memeriksa model lengkap dalam kasus ini, pustaka SMP menerjemahkan model kembali ke format pos pemeriksaan Hugging Face Transformers secara default.

Jika Anda berlatih dengan paralelisme tensor SMP dan mematikan proses penerjemahan SMP, Anda dapat menggunakan `translate_on_save` argumen PyTorch `FullStateDictConfig` API untuk mengaktifkan atau menonaktifkan terjemahan otomatis SMP sesuai kebutuhan. Misalnya, jika Anda berfokus pada pelatihan model, Anda tidak perlu menambahkan proses terjemahan yang menambahkan overhead. Dalam hal ini, kami sarankan Anda untuk mengatur`translate_on_save=False`. Juga, jika Anda berencana untuk tetap menggunakan terjemahan SMP model untuk pelatihan lebih lanjut di masa depan, Anda juga dapat mematikannya untuk menyimpan terjemahan SMP model untuk digunakan nanti. Menerjemahkan model kembali ke format pos pemeriksaan model Hugging Face Transformers diperlukan saat Anda menyelesaikan pelatihan model Anda dan menggunakannya untuk inferensi.

```
from torch.distributed.fsdp import FullyShardedDataParallel as FSDP
from torch.distributed.fsdp import FullStateDictConfig
import torch.sagemaker as tsm
```

```
# Save checkpoints.
with FSDP.state_dict_type(
    model,
    StateDictType.FULL_STATE_DICT,
    FullStateDictConfig(
        rank0_only=True, offload_to_cpu=True,
        # Default value is to translate back to Hugging Face Transformers format,
        # when saving full checkpoints for models trained with SMP tensor parallelism.
        # translate_on_save=True
    ),
):
    state_dict = model.state_dict()
    if dist.get_rank() == 0:
        logger.info("Processed state dict to save. Starting write to disk now.")
        os.makedirs(save_dir, exist_ok=True)
        # This name is needed for HF from_pretrained API to work.
        torch.save(state_dict, os.path.join(save_dir, "pytorch_model.bin"))
        hf_model_config.save_pretrained(save_dir)
    dist.barrier()
```

Perhatikan bahwa pilihannya `FullStateDictConfig(rank0_only=True, offload_to_cpu=True)` adalah mengumpulkan model pada CPU perangkat peringkat 0 untuk menghemat memori saat melatih model besar.

Untuk memuat kembali model untuk inferensi, Anda melakukannya seperti yang ditunjukkan pada contoh kode berikut. Perhatikan bahwa kelas `AutoModelForCausalLM` mungkin berubah ke kelas pembuat faktor lain di Hugging Face Transformers, `AutoModelForSeq2SeqLM` seperti, tergantung pada model Anda. Untuk informasi selengkapnya, lihat dokumentasi Hugging Face Transformers.

```
from transformers import AutoModelForCausalLM
model = AutoModelForCausalLM.from_pretrained(save_dir)
```

SageMaker praktik terbaik paralelisme model terdistribusi

Gunakan panduan berikut saat Anda menjalankan pekerjaan pelatihan terdistribusi dengan SageMaker model parallel library v2 (SMP v2).

Menyiapkan konfigurasi yang tepat untuk pelatihan terdistribusi

Untuk memperkirakan dan menemukan titik awal terbaik untuk menerapkan teknik pelatihan terdistribusi yang disediakan SMP v2, tinjau daftar berikut. Setiap item daftar membahas keuntungan menggunakan the section called “Fitur inti dari SMP v2” bersama dengan potensi pengorbanan.

Kiat konfigurasi

Bagian ini memberikan panduan tentang cara memutuskan konfigurasi model terbaik untuk throughput optimal dengan persyaratan ukuran batch global.

Pertama, kami merekomendasikan pengaturan berikut terlepas dari ukuran model Anda.

1. Gunakan jenis instans paling kuat yang dapat Anda gunakan.
2. Nyalakan presisi campuran sepanjang waktu, karena memberikan manfaat besar untuk kinerja dan pengurangan memori. Kami menyarankan Anda untuk menggunakannya `bfloat16` karena lebih tepat daripada`float16`.
3. Aktifkan pustaka paralelisme data SageMaker terdistribusi (bukan menggunakan NCCL) kapan pun itu berlaku, seperti yang ditunjukkan pada. the section called “Kompatibilitas dengan perpustakaan SMDDP” Satu pengecualian adalah untuk kasus tensor-parallelism-only penggunaan (`hybrid_shard_degree = 1`dan`tensor_paralle_degree > 1`).
4. Jika model Anda memiliki lebih dari sekitar 60 miliar parameter, sebaiknya gunakanthe section called “Inisialisasi parameter tertunda”. Anda juga dapat menggunakan inisialisasi parameter tertunda untuk mempercepat inisialisasi untuk model apa pun.
5. Kami menyarankan Anda untuk mengaktifkanthe section called “Pos pemeriksaan aktivasi”.

Tergantung pada ukuran model Anda, kami sarankan Anda memulai dengan panduan berikut.

1. Gunakan paralelisme data sharded.
   a. Bergantung pada ukuran batch yang ingin Anda muat dalam memori GPU, pilih derajat paralel data sharded yang sesuai. Biasanya, Anda harus mulai dengan tingkat terendah agar sesuai dengan model Anda dalam memori GPU sambil meminimalkan overhead dari komunikasi jaringan. Jika Anda melihat peringatan bahwa cache flushes sedang terjadi, kami sarankan Anda meningkatkan derajat sharding.
   b. Tentukan `world_size` berdasarkan ukuran batch lokal maksimum dan ukuran batch global yang diperlukan, jika ada.

c. Anda dapat bereksperimen dengan pembongkaran aktivasi. Tergantung pada skenario, itu dapat memenuhi kebutuhan memori Anda tanpa harus meningkatkan derajat sharding, yang berarti lebih sedikit komunikasi.

2. Gunakan paralelisme data sharded dari PyTorch FSDP dan paralelisme tensor SMP v2 secara bersamaan, seperti yang diperkenalkan pada. the section called “Paralelisme tensor”

   a. Saat berlatih pada cluster besar, dengan FSDP saja ukuran batch global bisa menjadi terlalu besar, menyebabkan masalah konvergensi untuk model. Biasanya, sebagian besar pekerjaan penelitian membuat ukuran batch di bawah 4 juta token. Dalam hal ini, Anda dapat menyelesaikan masalah dengan menyusun PyTorch FSDP dengan paralelisme tensor SMP v2 untuk mengurangi ukuran batch.

      Misalnya, jika Anda memiliki 256 node dan panjang urutan 4096, bahkan ukuran batch 1 per GPU mengarah ke ukuran batch global 8M token. Namun, ketika Anda menggunakan paralelisme tensor dengan derajat 2 dan ukuran batch 1 per grup paralel tensor, ini menjadi 1/2 ukuran batch per GPU, yang diterjemahkan menjadi 4 juta token.

   b. Saat berlatih dengan panjang konteks yang panjang seperti 8k, memori aktivasi 16k bisa menjadi sangat tinggi. FSDP tidak memecah aktivasi, dan aktivasi dapat menyebabkan GPU kehabisan memori. Dalam skenario seperti itu, Anda dapat berlatih secara efisien dengan menyusun PyTorch FSDP dengan paralelisme tensor SMP v2.

Konfigurasi referensi

Tim pelatihan paralelisme SageMaker model menyediakan titik referensi berikut berdasarkan eksperimen dengan model Llama 2 pada `ml.p4d.24xlarge` contoh, panjang urutan 4096, dan ditransformasikan ke model SMP dengan. the section called “`torch.sagemaker.transform`”

Model	Ukuran model (jumlah parameter model)	Jumlah contoh	Derajat paralel data sharded	Derajat paralel tensor	Pos pemeriksaan aktivasi	Pembongkaran aktivasi	Ukuran batch
Llama 2	7B	1	8	1	BETUL	SALAH	4
	70B	32	256	1	BETUL	SALAH	2
	175B	64	128	4	BETUL	BETUL	6

Anda dapat mengekstrapolasi dari konfigurasi sebelumnya untuk memperkirakan penggunaan memori GPU untuk konfigurasi model Anda. Misalnya, jika Anda menambah panjang urutan untuk model 10 miliar parameter atau meningkatkan ukuran model menjadi 20 miliar, Anda mungkin ingin menurunkan ukuran batch terlebih dahulu. Jika model masih tidak cocok, coba tingkatkan derajat paralelisme tensor.

Memantau dan mencatat pekerjaan pelatihan menggunakan SageMaker konsol dan Amazon CloudWatch

Untuk memantau metrik tingkat sistem seperti pemanfaatan memori CPU, pemanfaatan memori GPU, dan pemanfaatan GPU, gunakan visualisasi yang disediakan melalui konsol. SageMaker

1. Di panel navigasi kiri, pilih Pelatihan.
2. Pilih pekerjaan Pelatihan.
3. Di panel utama, pilih nama pekerjaan pelatihan yang ingin Anda lihat lebih jelasnya.
4. Jelajahi panel utama dan temukan bagian Monitor untuk melihat visualisasi otomatis.
5. Untuk melihat log pekerjaan pelatihan, pilih Lihat log di bagian Monitor. Anda dapat mengakses log pekerjaan pelatihan terdistribusi dari pekerjaan pelatihan di CloudWatch. Jika Anda meluncurkan pelatihan terdistribusi multi-node, Anda akan melihat beberapa aliran log dengan tag dalam format algo-n-1234567890. Aliran log algo-1 melacak log pelatihan dari simpul utama (ke-0).

Untuk informasi selengkapnya, lihat Memantau dan Menganalisis Pekerjaan Pelatihan Menggunakan Amazon CloudWatch Metrics.

Izin

Untuk menjalankan pekerjaan SageMaker pelatihan dengan paralelisme model, pastikan Anda memiliki izin yang tepat dalam peran IAM Anda, seperti berikut ini:

- Untuk menggunakan FSx for Lustre, tambahkan. `AmazonFSxFullAccess`
- Untuk menggunakan Amazon S3 sebagai saluran data, tambahkan. `AmazonS3FullAccess`
- Untuk menggunakan Docker, buat wadah Anda sendiri, dan dorong ke Amazon ECR, tambahkan. `AmazonEC2ContainerRegistryFullAccess`
- Untuk memiliki akses penuh untuk menggunakan seluruh rangkaian SageMaker fitur, tambahkan `AmazonSageMakerFullAccess`.

Referensi SageMaker model parallel library v2

Berikut ini adalah referensi untuk SageMaker model parallel library v2 (SMP v2).

Topik

Parameter konfigurasi fitur inti SMP v2

Berikut ini adalah daftar lengkap parameter untuk mengaktifkan dan mengkonfigurasithe section called “Fitur inti dari SMP v2”. Ini harus ditulis dalam format JSON dan diteruskan ke PyTorch estimator di SDK SageMaker Python atau disimpan sebagai file JSON untuk. SageMaker HyperPod

```
{
    "hybrid_shard_degree": Integer,
    "sm_activation_offloading": Boolean,
    "activation_loading_horizon": Integer,
    "fsdp_cache_flush_warnings": Boolean,
    "allow_empty_shards": Boolean,
    "tensor_parallel_degree": Integer,
    "tensor_parallel_seed": Integer
}
```

- `hybrid_shard_degree`(Integer) - Menentukan derajat paralelisme sharded. Nilai harus berupa bilangan bulat antara `0` dan`world_size`. Nilai default-nya adalah `0`.
  - Jika disetel ke`0`, itu kembali ke PyTorch implementasi asli dan API dalam skrip saat `tensor_parallel_degree` 1. Jika tidak, itu menghitung kemungkinan terbesar `hybrid_shard_degree` berdasarkan `tensor_parallel_degree` dan`world_size`. Saat kembali ke kasus penggunaan PyTorch FSDP asli, jika `FULL_SHARD` strategi yang Anda gunakan, itu akan terbagi di seluruh cluster GPU. Jika `HYBRID_SHARD` atau `_HYBRID_SHARD_ZERO2` apakah strateginya, itu setara `hybrid_shard_degree` dengan 8. Ketika paralelisme tensor diaktifkan, itu pecah berdasarkan revisi. `hybrid_shard_degree`
  - Jika disetel ke`1`, itu kembali ke PyTorch implementasi asli dan API untuk `NO_SHARD` dalam skrip saat `tensor_parallel_degree` 1. Jika tidak, itu setara dengan `NO_SHARD` dalam grup paralel tensor tertentu.

  - Jika disetel ke bilangan bulat antara 2 dan`world_size`, sharding terjadi di seluruh jumlah GPU yang ditentukan. Jika Anda tidak mengatur `sharding_strategy` skrip FSDP, itu akan diganti. `HYBRID_SHARD` Jika Anda mengatur`_HYBRID_SHARD_ZERO2`, yang `sharding_strategy` Anda tentukan digunakan.

- `sm_activation_offloading`(Boolean) - Menentukan apakah akan mengaktifkan implementasi pembongkaran aktivasi SMP. Jika`False`, pembongkaran menggunakan PyTorch implementasi asli. Jika`True`, ia menggunakan implementasi pembongkaran aktivasi SMP. Anda juga perlu menggunakan PyTorch aktivasi offload wrapper (`torch.distributed.algorithms._checkpoint.checkpoint_wrapper.offload_wrapper`) dalam skrip Anda. Untuk mempelajari selengkapnya, lihat the section called “Pembongkaran aktivasi”. Nilai default-nya adalah `True`.
- `activation_loading_horizon`(Integer) - Bilangan bulat yang menentukan jenis horizon pembongkaran aktivasi untuk FSDP. Ini adalah jumlah maksimum lapisan checkpoint atau offloaded yang inputnya dapat berada di memori GPU secara bersamaan. Untuk mempelajari selengkapnya, lihat the section called “Pembongkaran aktivasi”. Nilai masukan harus berupa bilangan bulat positif. Nilai default-nya adalah 2.
- `fsdp_cache_flush_warnings`(Boolean) - Mendeteksi dan memperingatkan jika cache flush terjadi di manajer PyTorch memori, karena mereka dapat menurunkan kinerja komputasi. Nilai default-nya adalah `True`.
- `allow_empty_shards`(Boolean) — Apakah akan mengizinkan pecahan kosong saat sharding tensor jika tensor tidak habis dibagi. Ini adalah perbaikan eksperimental untuk crash selama checkpointing dalam skenario tertentu. Menonaktifkan ini kembali ke perilaku aslinya PyTorch . Nilai default-nya adalah `False`.
- `tensor_parallel_degree`(Integer) - Menentukan derajat paralelisme tensor. Nilai harus antara 1 dan`world_size`. Nilai default-nya adalah 1. Melewati nilai yang lebih besar dari 1 tidak memungkinkan paralelisme tensor secara otomatis. Anda juga perlu menggunakan the section called “`torch.sagemaker.transform`” API untuk membungkus model dalam skrip pelatihan Anda. Untuk mempelajari selengkapnya, lihat the section called “Paralelisme tensor”.
- `tensor_parallel_seed`(Integer) — Nomor benih untuk operasi acak dalam modul terdistribusi paralel tensor. Benih ini akan ditambahkan ke peringkat paralel tensor untuk menetapkan benih aktual untuk setiap peringkat. Ini unik untuk setiap peringkat paralel tensor. SMP v2 memastikan bahwa pembuatan bilangan acak di seluruh peringkat paralel tensor cocok dengan kasus paralelisme non-tensor. Untuk mempelajari selengkapnya, lihat the section called “Paralelisme tensor”.

Referensi untuk paket SMP v2 **`torch.sagemaker`**

Bagian ini adalah referensi untuk `torch.sagemaker` paket yang disediakan oleh SMP v2.

Topik

`torch.sagemaker.delayed_param.DelayedParamIniter`

API untuk diterapkan the section called “Inisialisasi parameter tertunda” ke PyTorch model.

```
class torch.sagemaker.delayed_param.DelayedParamIniter(
    model: nn.Module,
    init_method_using_config : Callable = None,
    verbose: bool = False,
)
```

Parameter

- `model`(`nn.Module`) — PyTorch Model untuk membungkus dan menerapkan fungsionalitas inisialisasi parameter tertunda SMP v2.
- `init_method_using_config`(Dapat dipanggil) — Jika Anda menggunakan implementasi paralel tensor SMP v2 atau didukungthe section called “Model Hugging Face Transformer kompatibel dengan paralelisme tensor SMP”, pertahankan parameter ini pada nilai default, yaitu. `None` Secara default, `DelayedParamIniter` API mengetahui cara menginisialisasi model yang diberikan dengan benar. Untuk model lain, Anda perlu membuat fungsi inisialisasi parameter khusus dan menambahkannya ke skrip Anda. Cuplikan kode berikut adalah `init_method_using_config` fungsi default yang diterapkan SMP v2 untuk. the section called “Model Hugging Face Transformer kompatibel dengan paralelisme tensor SMP” Gunakan cuplikan kode berikut sebagai referensi untuk membuat fungsi konfigurasi inisialisasi Anda sendiri, menambahkannya ke skrip Anda, dan meneruskannya ke `init_method_using_config` parameter API SMP. `DelayedParamIniter`

```
from torch.sagemaker.utils.module_utils import empty_module_params,
 move_buffers_to_device

# Define a custom init config function.
def custom_init_method_using_config(module):
    d = torch.cuda.current_device()
    empty_module_params(module, device=d)
    if isinstance(module, (nn.Linear, Conv1D)):
        module.weight.data.normal_(mean=0.0, std=config.initializer_range)
        if module.bias is not None:
            module.bias.data.zero_()
    elif isinstance(module, nn.Embedding):
        module.weight.data.normal_(mean=0.0, std=config.initializer_range)
        if module.padding_idx is not None:
            module.weight.data[module.padding_idx].zero_()
    elif isinstance(module, nn.LayerNorm):
        module.weight.data.fill_(1.0)
        module.bias.data.zero_()
    elif isinstance(module, LlamaRMSNorm):
        module.weight.data.fill_(1.0)
    move_buffers_to_device(module, device=d)

delayed_initer = DelayedParamIniter(model,
 init_method_using_config=custom_init_method_using_config)
```

Untuk informasi selengkapnya tentang `torch.sagemaker.module_util` fungsi dalam cuplikan kode sebelumnya, lihat. the section called “`torch.sagemaker`fungsi dan properti util”

- `verbose`(Boolean) - Apakah akan mengaktifkan logging yang lebih rinci selama inisialisasi dan validasi. Nilai default-nya adalah `False`.

Metode

- `get_param_init_fn()`— Mengembalikan fungsi inisialisasi parameter yang dapat Anda lewatkan ke `param_init_fn` argumen kelas pembungkus PyTorch FSDP.
- `get_post_param_init_fn()`— Mengembalikan fungsi inisialisasi parameter yang dapat Anda lewatkan ke `post_param_init_fn` argumen kelas pembungkus PyTorch FSDP. Ini diperlukan ketika Anda telah mengikat bobot dalam model. Model harus menerapkan metode`tie_weights`. Untuk informasi selengkapnya, lihat Catatan tentang bobot terikatthe section called “Inisialisasi parameter tertunda”.

- `count_num_params(module: nn.Module, *args: Tuple[nn.Parameter])` — Melacak berapa banyak parameter yang diinisialisasi oleh fungsi inisialisasi parameter. Ini membantu menerapkan `validate_params_and_buffers_inited` metode berikut. Anda biasanya tidak perlu memanggil fungsi ini secara eksplisit, karena metode ini secara implisit memanggil `validate_params_and_buffers_inited` metode ini di backend.
- `validate_params_and_buffers_inited(enabled: bool=True)` — Ini adalah manajer konteks yang membantu memvalidasi bahwa jumlah parameter yang diinisialisasi cocok dengan jumlah total parameter dalam model. Ini juga memvalidasi bahwa semua parameter dan buffer sekarang ada di perangkat GPU, bukan perangkat meta. Ini memunculkan `AssertionErrors` jika kondisi ini tidak terpenuhi. Pengelola konteks ini hanya opsional dan Anda tidak diharuskan menggunakan pengelola konteks ini untuk menginisialisasi parameter.

`torch.sagemaker.nn.attn.FlashSelfAttention`

API untuk digunakan the section called “FlashAttention” dengan SMP v2.

```
class torch.sagemaker.nn.attn.FlashSelfAttention(
   attention_dropout_prob: float = 0.0,
   scale: Optional[float] = None,
   triton_flash_attention: bool = False,
   use_alibi: bool = False,
)
```

Parameter

- `attention_dropout_prob`(float) — Probabilitas putus sekolah untuk diterapkan pada perhatian. Nilai default-nya adalah `0.0`.
- `scale`(float) — Jika lulus, faktor skala ini akan diterapkan untuk softmax. Jika disetel ke `None` (yang juga merupakan nilai default), faktor skalanya adalah`1 / sqrt(attention_head_size)`. Nilai default-nya adalah `None`.
- `triton_flash_attention`(bool) — Jika lulus, implementasi Triton perhatian flash akan digunakan. Ini diperlukan untuk mendukung Perhatian dengan Bias Linear (ALiBi) (lihat `use_alibi` parameter berikut). Versi kernel ini tidak mendukung putus sekolah. Nilai default-nya adalah `False`.
- `use_alibi`(bool) — Jika diteruskan, ini memungkinkan Perhatian dengan Bias Linear (ALiBi) menggunakan topeng yang disediakan. Saat menggunakan ALiBi, perlu masker perhatian yang disiapkan sebagai berikut. Nilai default-nya adalah `False`.

```
def generate_alibi_attn_mask(attention_mask, batch_size, seq_length,
    num_attention_heads, alibi_bias_max=8):
    device, dtype = attention_mask.device, attention_mask.dtype
    alibi_attention_mask = torch.zeros(
        1, num_attention_heads, 1, seq_length, dtype=dtype, device=device
    )

    alibi_bias = torch.arange(1 - seq_length, 1, dtype=dtype, device=device).view(
        1, 1, 1, seq_length
    )
    m = torch.arange(1, num_attention_heads + 1, dtype=dtype, device=device)
    m.mul_(alibi_bias_max / num_attention_heads)
    alibi_bias = alibi_bias * (1.0 / (2 ** m.view(1, num_attention_heads, 1, 1)))

    alibi_attention_mask.add_(alibi_bias)
    alibi_attention_mask = alibi_attention_mask[..., :seq_length, :seq_length]
    if attention_mask is not None and attention_mask.bool().any():
        alibi_attention_mask.masked_fill(
            attention_mask.bool().view(batch_size, 1, 1, seq_length), float("-inf")
        )

    return alibi_attention_mask
```

Metode

- `forward(self, qkv, attn_mask=None, causal=False, cast_dtype=None, layout="b h s d")`- Fungsi PyTorch modul reguler. Ketika a `module(x)` dipanggil, SMP menjalankan fungsi ini secara otomatis.
  - `qkv`— `torch.Tensor` dari bentuk berikut: `(batch_size x seqlen x (3 x num_heads) x head_size)` atau`(batch_size, (3 x num_heads) x seqlen x head_size)`, tupel yang `torch.Tensors` masing-masing mungkin berbentuk`(batch_size x seqlen x num_heads x head_size)`, atau`(batch_size x num_heads x seqlen x head_size)`. Arg tata letak yang sesuai harus diteruskan berdasarkan bentuknya.
  - `attn_masktorch.Tensor`dari bentuk berikut`(batch_size x 1 x 1 x seqlen)`. Untuk mengaktifkan parameter topeng perhatian ini, diperlukan `triton_flash_attention=True` dan`use_alibi=True`. Untuk mempelajari cara membuat topeng perhatian menggunakan metode ini, lihat contoh kode dithe section called “FlashAttention”. Nilai default-nya adalah None.

- `causal`— Ketika diatur ke`False`, yang merupakan nilai default dari argumen, tidak ada topeng diterapkan. Ketika diatur ke`True`, `forward` metode ini menggunakan topeng segitiga bawah standar. Nilai default-nya adalah `False`.
- `cast_dtype`— Ketika diatur ke tertentu`dtype`, itu melemparkan `qkv` tensor ke itu sebelumnya. `dtype` `attn` Ini berguna untuk implementasi seperti model Hugging Face Transformer GPT-Neox, yang memiliki dan dengan after rotary embeddings. `q` `k` `fp32` Jika disetel ke`None`, tidak ada pemeran yang diterapkan. Nilai default-nya adalah `None`.
- `layout`(string) - Nilai yang tersedia adalah `b h s d` ataub `s h d`. Ini harus diatur ke tata letak `qkv` tensor yang dilewati, sehingga transformasi yang tepat dapat diterapkan. `attn` Nilai default-nya adalah `b h s d`.

Pengembalian

Satu `torch.Tensor` dengan bentuk`(batch_size x num_heads x seq_len x head_size)`.

`torch.sagemaker.nn.attn.FlashGroupedQueryAttention`

API untuk digunakan `FlashGroupedQueryAttention` dengan SMP v2. Untuk mempelajari lebih lanjut tentang penggunaan API ini, lihatthe section called “Gunakan FlashAttention kernel untuk perhatian kueri yang dikelompokkan”.

```
class torch.sagemaker.nn.attn.FlashGroupedQueryAttention(
    attention_dropout_prob: float = 0.0,
    scale: Optional[float] = None,
)
```

Parameter

- `attention_dropout_prob`(float) — Probabilitas putus sekolah untuk diterapkan pada perhatian. Nilai default-nya adalah `0.0`.
- `scale`(float) — Jika lulus, faktor skala ini diterapkan untuk softmax. Jika diatur ke`None`, `1 / sqrt(attention_head_size)` digunakan sebagai faktor skala. Nilai default-nya adalah `None`.

Metode

- `forward(self, q, kv, causal=False, cast_dtype=None, layout="b s h d")`- Fungsi PyTorch modul reguler. Ketika a `module(x)` dipanggil, SMP menjalankan fungsi ini secara otomatis.

- `qtorch.Tensor`dari bentuk berikut `(batch_size x seqlen x num_heads x head_size)` atau`(batch_size x num_heads x seqlen x head_size)`. Arg tata letak yang sesuai harus dilewatkan berdasarkan bentuknya.
- `kv`— `torch.Tensor` dari bentuk berikut `(batch_size x seqlen x (2 x num_heads) x head_size)` atau`(batch_size, (2 x num_heads) x seqlen x head_size)`, atau tupel dua `torch.Tensor` s, yang masing-masing mungkin berbentuk `(batch_size x seqlen x num_heads x head_size)` atau`(batch_size x num_heads x seqlen x head_size)`. `layout`Argumen yang tepat juga harus diteruskan berdasarkan bentuknya.
- `causal`— Ketika diatur ke`False`, yang merupakan nilai default dari argumen, tidak ada topeng diterapkan. Ketika diatur ke`True`, `forward` metode ini menggunakan topeng segitiga bawah standar. Nilai default-nya adalah `False`.
- `cast_dtype`— Ketika diatur ke dtype tertentu, itu melemparkan `qkv` tensor ke dtype itu sebelumnya. `attn` Ini berguna untuk implementasi seperti Hugging Face Transformers GPT-Neox, yang memiliki dengan after rotary embeddings. `q,k` `fp32` Jika disetel ke`None`, tidak ada pemeran yang diterapkan. Nilai default-nya adalah `None`.
- layout (string) - Nilai yang tersedia adalah `"b h s d"` atau`"b s h d"`. Ini harus diatur ke tata letak `qkv` tensor yang dilewati, sehingga transformasi yang tepat dapat diterapkan. `attn` Nilai default-nya adalah `"b h s d"`.

Pengembalian

Mengembalikan satu `torch.Tensor (batch_size x num_heads x seq_len x head_size)` yang mewakili output dari perhitungan perhatian.

`torch.sagemaker.nn.huggingface.llama_flashattn.LlamaFlashAttention`

API yang mendukung FlashAttention model Llama. API ini menggunakan the section called "`torch.sagemaker.nn.attn.FlashGroupedQueryAttention`" API pada level rendah. Untuk mempelajari cara menggunakan ini, lihatthe section called "Gunakan FlashAttention kernel untuk perhatian kueri yang dikelompokkan".

```
class torch.sagemaker.nn.huggingface.llama_flashattn.LlamaFlashAttention(
    config: LlamaConfig
)
```

Parameter

- `config`— FlashAttention Konfigurasi untuk model Llama.

Metode

- `forward(self, hidden_states, attention_mask, position_ids, past_key_value, output_attentions, use_cache)`
  - `hidden_states`(`torch.Tensor`) — Keadaan tersembunyi dari tensor dalam bentuk. `(batch_size x seq_len x num_heads x head_size)`
  - `attention_mask`(`torch.LongTensor`) — Topeng untuk menghindari perhatian pada indeks token padding dalam bentuk. `(batch_size x seqlen)` Nilai default-nya adalah `None`.
  - `position_ids`(`torch.LongTensor`) — Ketika tidak`None`, itu dalam bentuk`(batch_size x seqlen)`, menunjukkan indeks posisi setiap token urutan input dalam embeddings posisi. Nilai default-nya adalah `None`.
  - `past_key_value`(Cache) — Keadaan tersembunyi yang telah dihitung sebelumnya (kunci dan nilai di blok perhatian diri dan di blok perhatian silang). Nilai default-nya adalah `None`.
  - `output_attentions`(bool) — Menunjukkan apakah akan mengembalikan tensor perhatian dari semua lapisan perhatian. Nilai default-nya adalah `False`.
  - `use_cache`(bool) - Menunjukkan apakah akan mengembalikan status nilai `past_key_values` kunci. Nilai default-nya adalah `False`.

Pengembalian

Mengembalikan satu `torch.Tensor (batch_size x num_heads x seq_len x head_size)` yang mewakili output dari perhitungan perhatian.

`torch.sagemaker.transform`

SMP v2 menyediakan `torch.sagemaker.transform()` API ini untuk mengubah model Hugging Face Transformer menjadi implementasi model SMP dan mengaktifkan paralelisme tensor SMP.

```
torch.sagemaker.transform(
    model: nn.Module,
    device: Optional[torch.device] = None,
    dtype: Optional[torch.dtype] = None,
    config: Optional[Dict] = None,
    load_state_dict_from_rank0: bool = False
)
```

SMP v2 mempertahankan kebijakan transformasi untuk the section called “Model Hugging Face Transformer kompatibel dengan paralelisme tensor SMP” dengan mengubah konfigurasi model Hugging Face Transformer ke konfigurasi transformator SMP.

Parameter

- `model`(`torch.nn.Module`) — Model dari the section called “Model Hugging Face Transformer kompatibel dengan paralelisme tensor SMP” untuk mengubah dan menerapkan fitur paralelisme tensor dari perpustakaan SMP.
- `device`(`torch.device`) — Jika diteruskan, model baru dibuat pada perangkat ini. Jika modul asli memiliki parameter apa pun pada perangkat meta (lihatthe section called “Inisialisasi parameter tertunda”), maka modul yang diubah juga akan dibuat pada perangkat meta, mengabaikan argumen yang diteruskan di sini. Nilai default-nya adalah `None`.
- `dtype`(`torch.dtype`) — Jika diteruskan, tetapkan ini sebagai pengelola konteks dtype untuk pembuatan model dan buat model dengan dtype ini. Ini biasanya tidak perlu, karena kita ingin membuat model dengan `fp32` saat menggunakan`MixedPrecision`, dan `fp32` merupakan dtype default di PyTorch. Nilai default-nya adalah `None`.
- `config`(dict) - Ini adalah kamus untuk mengkonfigurasi transformator SMP. Berikut ini adalah kunci yang tersedia. Nilai default-nya adalah `None`.
  - `sequence_parallel`(Boolean) — Sebuah Boolean menentukan apakah akan menggunakan paralelisme urutan. Paralelisme urutan partisi masukan urutan sepanjang dimensi urutan untuk meningkatkan efisiensi memori. Nilai default-nya adalah `True`.
- `load_state_dict_from_rank0`(Boolean) - Secara default, modul ini membuat instance baru dari model dengan bobot baru. Ketika argumen ini disetel ke`True`, SMP mencoba memuat kamus status PyTorch model asli dari peringkat 0 menjadi model transformasi untuk grup paralel tensor yang menjadi bagian dari peringkat ke-0. Ketika ini disetel ke`True`, peringkat 0 tidak dapat memiliki parameter apa pun di perangkat meta. Hanya grup paralel tensor pertama yang mengisi bobot dari peringkat ke-0 setelah panggilan transformasi ini. Anda perlu mengatur `sync_module_states` ke `True` dalam pembungkus FSDP untuk mendapatkan bobot ini dari grup paralel tensor pertama ke semua proses lainnya. Dengan ini diaktifkan, perpustakaan SMP memuat kamus status dari model aslinya. Pustaka SMP mengambil model sebelum mentransformasi, mengubahnya agar sesuai dengan struktur model yang ditransformasikan, memecahnya untuk setiap peringkat paralel tensor, mengkomunikasikan keadaan ini dari peringkat ke-0 ke peringkat lain dalam kelompok paralel tensor yang peringkat ke-0 adalah bagian dari, dan memuatnya. `state_dict` Nilai default-nya adalah `False`.

Pengembalian

Mengembalikan model yang diubah yang dapat Anda bungkus dengan PyTorch FSDP. Kapan `load_state_dict_from_rank0` diatur ke`True`, grup paralel tensor yang melibatkan peringkat 0 memiliki bobot yang dimuat dari kamus status asli pada peringkat 0. Saat menggunakan the section called “Inisialisasi parameter tertunda” pada model asli, hanya peringkat ini yang memiliki tensor aktual pada CPU untuk parameter dan buffer model yang diubah. Sisa peringkat terus memiliki parameter dan buffer pada perangkat meta untuk menghemat memori.

`torch.sagemaker`fungsi dan properti util

fungsi obor.sagemaker util

- `torch.sagemaker.init(config: Optional[Union[str, Dict[str, Any]]] = None) -> None`— Menginisialisasi pekerjaan PyTorch pelatihan dengan SMP.
- `torch.sagemaker.is_initialized() -> bool`— Memeriksa apakah pekerjaan pelatihan diinisialisasi dengan SMP. Ketika kembali ke asli PyTorch saat pekerjaan diinisialisasi dengan SMP, beberapa properti tidak relevan dan menjadi`None`, seperti yang ditunjukkan dalam daftar Properti berikut.
- `torch.sagemaker.utils.module_utils.empty_module_params(module: nn.Module, device: Optional[torch.device] = None, recurse: bool = False) -> nn.Module`— Membuat parameter kosong pada yang diberikan `device` jika ada, dan dapat bersifat rekursif untuk semua modul bersarang jika ditentukan.
- `torch.sagemaker.utils.module_utils.move_buffers_to_device(module: nn.Module, device: torch.device, recurse: bool = False) -> nn.Module`— Memindahkan buffer modul ke yang diberikan`device`, dan dapat bersifat rekursif untuk semua modul bersarang jika ditentukan.

Properti

`torch.sagemaker.state`memegang beberapa properti yang berguna setelah inisialisasi SMP dengan. `torch.sagemaker.init`

- `torch.sagemaker.state.hybrid_shard_degree`(int) — Tingkat paralelisme data sharded, salinan dari input pengguna dalam konfigurasi SMP diteruskan ke. `torch.sagemaker.init()` Untuk mempelajari selengkapnya, lihat the section called “Memulai dengan SMP v2”.
- `torch.sagemaker.state.rank`(int) — Peringkat global untuk perangkat, dalam kisaran`[0, world_size)`.

- `torch.sagemaker.state.rep_rank_process_group(torch.distributed.ProcessGroup)` — Grup proses termasuk semua perangkat dengan peringkat replikasi yang sama. Perhatikan perbedaan halus namun mendasar dengan`torch.sagemaker.state.tp_process_group`. Ketika jatuh kembali ke asli PyTorch, ia kembali`None`.
- `torch.sagemaker.state.tensor_parallel_degree`(int) — Derajat paralelisme tensor, salinan dari input pengguna dalam konfigurasi SMP diteruskan ke. `torch.sagemaker.init()` Untuk mempelajari selengkapnya, lihat the section called “Memulai dengan SMP v2”.
- `torch.sagemaker.state.tp_size`(int) — Sebuah alias untuk`torch.sagemaker.state.tensor_parallel_degree`.
- `torch.sagemaker.state.tp_rank`(int) — Peringkat paralelisme tensor untuk perangkat dalam kisaran`[0, tp_size)`, ditentukan oleh derajat paralelisme tensor dan mekanisme peringkat.
- `torch.sagemaker.state.tp_process_group(torch.distributed.ProcessGroup)` — Kelompok proses paralel tensor termasuk semua perangkat dengan peringkat yang sama di dimensi lain (misalnya, paralelisme dan replikasi data sharded) tetapi peringkat paralel tensor yang unik. Ketika jatuh kembali ke asli PyTorch, ia kembali`None`.
- `torch.sagemaker.state.world_size`(int) — Jumlah total perangkat yang digunakan dalam pelatihan.

Tingkatkan dari SMP v1 ke SMP v2

Untuk berpindah dari SMP v1 ke SMP v2, Anda harus membuat perubahan skrip untuk menghapus API SMP v1 dan menerapkan API SMP v2. Alih-alih memulai dari skrip SMP v1 Anda, kami sarankan Anda memulai dari skrip PyTorch FSDP, dan ikuti instruksi di. the section called “Memulai dengan SMP v2”

Untuk membawa model SMP v1 ke SMP v2, di SMP v1 Anda harus mengumpulkan kamus status model lengkap dan menerapkan fungsi terjemahan pada kamus status model untuk mengubahnya menjadi format pos pemeriksaan model Hugging Face Transformers. Kemudian di SMP v2, seperti yang dibahas dithe section called “Simpan dan muat pos pemeriksaan saat menggunakan SMP”, Anda dapat memuat pos pemeriksaan model Hugging Face Transformers, dan kemudian melanjutkan dengan menggunakan API pos pemeriksaan dengan SMP v2. PyTorch Untuk menggunakan SMP dengan model PyTorch FSDP Anda, pastikan Anda pindah ke SMP v2 dan membuat perubahan pada skrip pelatihan Anda untuk menggunakan PyTorch FSDP dan fitur terbaru lainnya.

```
import smdistributed.modelparallel.torch as smp
```

```
# Create model
model = ...
model = smp.DistributedModel(model)

# Run training
...

# Save v1 full checkpoint
if smp.rdp_rank() == 0:
    model_dict = model.state_dict(gather_to_rank0=True) # save the full model
    # Get the corresponding translation function in smp v1 and translate
    if model_type == "gpt_neox":
        from smdistributed.modelparallel.torch.nn.huggingface.gptneox import
 translate_state_dict_to_hf_gptneox
        translated_state_dict = translate_state_dict_to_hf_gptneox(state_dict,
 max_seq_len=None)

    # Save the checkpoint
    checkpoint_path = "checkpoint.pt"
    if smp.rank() == 0:
        smp.save(
            {"model_state_dict": translated_state_dict},
            checkpoint_path,
            partial=False,
        )
```

Untuk menemukan fungsi terjemahan yang tersedia di SMP v1, lihat. the section called “Support untuk Model Trafo Hugging Face”

Untuk instruksi tentang penyimpanan dan pemuatan pos pemeriksaan model di SMP v2, lihat. the section called “Simpan dan muat pos pemeriksaan saat menggunakan SMP”

Catatan rilis untuk perpustakaan paralelisme SageMaker model

Lihat catatan rilis berikut untuk melacak pembaruan terbaru untuk pustaka SageMaker model paralelisme (SMP). Jika Anda memiliki pertanyaan lebih lanjut tentang perpustakaan SMP, hubungi tim layanan SMP di. `sm-model-parallel-feedback@amazon.com`

Pustaka paralelisme SageMaker model v2.1.0

Tanggal: 6 Februari 2024

Pembaruan Mata Uang

- Ditambahkan dukungan untuk PyTorch v2.1.2.

penghentian

- Dukungan yang dihentikan untuk Hugging Face Transformers v4.31.0.

Masalah yang diketahui

- Masalah ditemukan bahwa fine-tuning model Hugging Face Llama 2 `attn_implementation=flash_attention_2` dengan dan FSDP menyebabkan model menyimpang. Untuk referensi, lihat tiket terbitan di repositori Hugging Face GitHub Transformers. Untuk menghindari masalah divergensi, gunakan`attn_implementation=sdpa`. Atau, gunakan implementasi model transformator SMP dengan menyiapkan`use_smp_implementation=True`.

Wadah SMP Docker

Mulai dari rilis ini, tim perpustakaan SMP mendistribusikan kontainer Docker sebagai pengganti wadah kerangka kerja. SageMaker PyTorch Jika Anda menggunakan kelas PyTorch estimator di SageMaker Python SDK dan menentukan konfigurasi distribusi untuk menggunakan SMP v2 SageMaker , secara otomatis mengambil kontainer SMP Docker. Untuk menggunakan rilis SMP v2 ini, tingkatkan SDK SageMaker Python Anda ke v2.207.0 atau yang lebih baru.

- Wadah SMP Docker untuk PyTorch v2.1.2 dengan CUDA v12.1

```
658645717510.dkr.ecr.us-west-2.amazonaws.com/smdistributed-modelparallel:2.1.2-gpu-
py310-cu121
```

  - Tersedia untuk instans P4d dan P5d
  - Paket pra-instal dalam wadah Docker ini
    - Perpustakaan SMDDP v2.1.0
    - CUDNN v8.9.5.29
    - FlashAttention v2.3.3
    - TransformerEngine v1.2.1
    - Trafo Hugging Face v4.37.1

- Pustaka Kumpulan Data Hugging Face v2.16.1
- EFA v1.30.0

Saluran SMP Conda

Bucket S3 berikut adalah saluran Conda publik yang diselenggarakan oleh tim layanan SMP. Jika Anda ingin menginstal perpustakaan SMP v2 di lingkungan seperti SageMaker HyperPod cluster, gunakan saluran Conda ini untuk menginstal pustaka SMP dengan benar.

- `https://sagemaker-distributed-model-parallel.s3.us-west-2.amazonaws.com/smp-v2/`

Untuk informasi selengkapnya tentang saluran Conda secara umum, lihat Saluran di dokumentasi Conda.

Pustaka paralelisme SageMaker model v2.0.0

Tanggal: 19 Desember 2023

Fitur baru

Merilis perpustakaan SageMaker model paralelisme (SMP) v2.0.0 dengan penawaran baru berikut.

- `torch.sagemaker`Paket baru, sepenuhnya dirubah dari `smdistributed.modelparallel.torch` paket sebelumnya di SMP v1.x.
- Support untuk PyTorch 2.0.1.
- Support untuk PyTorch FSDP.
- Implementasi paralelisme tensor dengan mengintegrasikan dengan perpustakaan Transformer Engine.
- Support untuk SageMaker Training dan SageMaker HyperPod.

Melanggar perubahan

- SMP v2 mengubah API sepenuhnya dan menyediakan paket. `torch.sagemaker` Sebagian besar, Anda hanya perlu menginisialisasi dengan `torch.sagemaker.init()` modul dan melewati parameter konfigurasi paralel model. Dengan paket baru ini, Anda dapat secara signifikan menyederhanakan modifikasi kode dalam skrip pelatihan Anda. Untuk mempelajari lebih lanjut

tentang mengadaptasi skrip pelatihan Anda untuk menggunakan SMP v2, lihat. the section called "Memulai dengan SMP v2"

- Jika Anda telah menggunakan SMP v1 untuk melatih model Hugging Face Transformer dan ingin menggunakan kembali model di SMP v2, lihat. the section called "Tingkatkan dari SMP v1 ke SMP v2"
- Untuk pelatihan PyTorch FSDP, Anda harus menggunakan SMP v2.

Masalah yang diketahui

- Checkpointing aktivasi saat ini hanya berfungsi dengan kebijakan pembungkus berikut dengan FSDP.
  - `auto_wrap_policy = functools.partial(transformer_auto_wrap_policy, ...)`
- Untuk menggunakanthe section called "Pembongkaran aktivasi", jenis checkpointing aktivasi FSDP harus REENTRANT.
- Saat menjalankan dengan tensor parallel diaktifkan dengan derajat paralel data sharded yang disetel ke1, Anda harus menggunakan. `backend = nccl` Opsi `smddp` backend tidak didukung dalam skenario ini.
- Mesin Transformer diperlukan untuk digunakan PyTorch dengan perpustakaan SMP bahkan ketika tidak menggunakan paralelisme tensor.

Perubahan lainnya

- Mulai dari rilis ini, dokumentasi untuk pustaka paralelisme SageMaker model sepenuhnya tersedia di Panduan SageMaker Pengembang Amazon ini. Untuk mendukung panduan pengembang lengkap ini untuk SMP v2 di Panduan SageMaker Pengembang Amazon, referensi tambahan untuk SMP v1.x dalam dokumentasi SageMaker Python SDK tidak digunakan lagi. Jika Anda masih memerlukan dokumentasi untuk SMP v1.x, panduan pengembang untuk SMP v1.x tersedia dithe section called "(Diarsipkan) perpustakaan SageMaker paralelisme model v1.x", dan referensi v1.x perpustakaan SMP Python tersedia di dokumentasi Python SDK v2.199.0. SageMaker

penghentian

- Dukungan yang dihentikan untuk TensorFlow.
- Tidak ada dukungan paralelisme pipa di SMP v2.
- Tidak ada dukungan untuk DeepSpeed perpustakaan yang mendukung PyTorch FSDP asli.

Migrasi ke AWS Deep Learning Containers

Versi ini lulus pengujian benchmark dan dimigrasikan ke AWSDeep Learning Containers berikut.

- SageMaker Wadah Kerangka untuk PyTorch v2.0.1

(Diarsipkan) perpustakaan SageMaker paralelisme model v1.x

> **Important**
>
> Pada 19 Desember 2023, perpustakaan SageMaker model paralelisme (SMP) v2 dirilis. Untuk mendukung pustaka SMP v2, kemampuan SMP v1 tidak lagi didukung di rilis mendatang. Bagian dan topik berikut diarsipkan dan khusus untuk menggunakan perpustakaan SMP v1. Untuk informasi tentang menggunakan pustaka SMP v2, lihatthe section called “SageMaker perpustakaan paralelisme model v2”.

Gunakan perpustakaan paralel model Amazon SageMaker untuk melatih model deep learning (DL) besar yang sulit dilatih karena keterbatasan memori GPU. Pustaka secara otomatis dan efisien membagi model di beberapa GPU dan instance. Dengan menggunakan perpustakaan, Anda dapat mencapai akurasi prediksi target lebih cepat dengan melatih model DL yang lebih besar secara efisien dengan miliaran atau triliunan parameter.

Anda dapat menggunakan perpustakaan untuk secara otomatis mempartisi sendiri TensorFlow dan PyTorch model Anda di beberapa GPU dan beberapa node dengan perubahan kode minimal. Anda dapat mengakses API perpustakaan melalui SageMaker Python SDK.

Gunakan bagian berikut untuk mempelajari selengkapnya tentang paralelisme model dan pustaka SageMaker paralel model. Dokumentasi API pustaka ini terletak di API Pelatihan Terdistribusi dalam dokumentasi SageMaker Python SDK v2.199.0.

Topik

- Pengantar Model Paralelisme
- Kerangka Kerja yang Didukung dan Wilayah AWS
- Fitur Inti dari Perpustakaan Paralelisme SageMaker Model
- Jalankan Job Pelatihan SageMaker Terdistribusi dengan Paralelisme Model
- Checkpointing dan Fine-Tuning Model dengan Paralelisme Model

- SageMaker Praktik Terbaik Paralelisme Model Terdistribusi
- Tips dan Jebakan Konfigurasi Perpustakaan Paralelisme Model SageMaker Terdistribusi
- Pemecahan Masalah Paralel Model

Pengantar Model Paralelisme

Paralelisme model adalah metode pelatihan terdistribusi di mana model pembelajaran mendalam dipartisi di beberapa perangkat, di dalam atau di seluruh instance. Halaman pengantar ini memberikan gambaran tingkat tinggi tentang paralelisme model, deskripsi tentang bagaimana hal itu dapat membantu mengatasi masalah yang muncul saat melatih model DL yang biasanya berukuran sangat besar, dan contoh apa yang ditawarkan perpustakaan paralel SageMaker model untuk membantu mengelola strategi paralel model serta konsumsi memori.

Apa itu Model Paralelisme?

Meningkatkan ukuran model pembelajaran mendalam (lapisan dan parameter) menghasilkan akurasi yang lebih baik untuk tugas-tugas kompleks seperti visi komputer dan pemrosesan bahasa alami. Namun, ada batas ukuran model maksimum yang dapat Anda muat dalam memori GPU tunggal. Saat melatih model DL, keterbatasan memori GPU dapat menjadi hambatan dengan cara-cara berikut:

- Mereka membatasi ukuran model yang dapat Anda latih, karena jejak memori model skala proporsional dengan jumlah parameter.
- Mereka membatasi ukuran batch per-GPU selama pelatihan, menurunkan pemanfaatan GPU dan efisiensi pelatihan.

Untuk mengatasi keterbatasan yang terkait dengan pelatihan model pada GPU tunggal, SageMaker menyediakan perpustakaan paralel model untuk membantu mendistribusikan dan melatih model DL secara efisien pada beberapa node komputasi. Selain itu, dengan perpustakaan, Anda dapat mencapai pelatihan terdistribusi yang paling optimal menggunakan perangkat yang didukung EFA, yang meningkatkan kinerja komunikasi antar-node dengan latensi rendah, throughput tinggi, dan bypass OS.

Perkirakan Persyaratan Memori Sebelum Menggunakan Model Paralelisme

Sebelum Anda menggunakan perpustakaan paralel SageMaker model, pertimbangkan hal berikut untuk mendapatkan rasa persyaratan memori pelatihan model DL besar.

Untuk pekerjaan pelatihan yang menggunakan pengoptimal AMP (FP16) dan Adam, memori GPU yang diperlukan per parameter adalah sekitar 20 byte, yang dapat kita uraikan sebagai berikut:

- Parameter FP16 ~ 2 byte
- Gradien FP16 ~ 2 byte
- Sebuah negara optimizer FP32 ~ 8 byte berdasarkan optimizers Adam
- Salinan parameter FP32 ~ 4 byte (diperlukan untuk operasi (OA`optimizer apply`))
- Salinan FP32 gradien ~ 4 byte (diperlukan untuk operasi OA)

Bahkan untuk model DL yang relatif kecil dengan 10 miliar parameter, dapat memerlukan setidaknya 200GB memori, yang jauh lebih besar daripada memori GPU biasa (misalnya, NVIDIA A100 dengan memori 40GB/80GB dan V100 dengan 16/32 GB) tersedia pada satu GPU. Perhatikan bahwa di atas persyaratan memori untuk status model dan pengoptimal, ada konsumen memori lain seperti aktivasi yang dihasilkan di pass maju. Memori yang dibutuhkan bisa jauh lebih besar dari 200GB.

Untuk pelatihan terdistribusi, sebaiknya gunakan instans Amazon EC2 P3 dan P4 yang masing-masing memiliki GPU NVIDIA V100 dan A100 Tensor Core. Untuk detail selengkapnya tentang spesifikasi seperti inti CPU, RAM, volume penyimpanan terlampir, dan bandwidth jaringan, lihat bagian Accelerated Computing di halaman Jenis Instans Amazon EC2.

Bahkan dengan instance komputasi yang dipercepat, jelas bahwa model dengan sekitar 10 miliar parameter seperti Megatron-LM dan T5 dan bahkan model yang lebih besar dengan ratusan miliar parameter seperti GPT-3 tidak dapat menyesuaikan replika model di setiap perangkat GPU.

Bagaimana Perpustakaan Mempekerjakan Model Paralelisme dan Teknik Menyimpan Memori

Perpustakaan terdiri dari berbagai jenis fitur paralelisme model dan fitur hemat memori seperti sharding status optimizer, checkpointing aktivasi, dan pembongkaran aktivasi. Semua teknik ini dapat dikombinasikan untuk melatih model besar secara efisien yang terdiri dari ratusan miliar parameter.

Topik

- Paralelisme data yang dipecah (tersedia untuk) PyTorch
- Paralelisme pipa (tersedia untuk PyTorch dan) TensorFlow
- Paralelisme tensor (tersedia untuk) PyTorch
- Sharding status pengoptimal (tersedia untuk) PyTorch
- Aktivasi pembongkaran dan checkpointing (tersedia untuk) PyTorch
- Memilih teknik yang tepat untuk model Anda

Paralelisme data yang dipecah (tersedia untuk) PyTorch

Paralelisme data yang dipecah adalah teknik pelatihan terdistribusi hemat memori yang membagi keadaan model (parameter model, gradien, dan status pengoptimal) di seluruh GPU dalam grup paralel data.

SageMakermengimplementasikan data sharded paralelisme melalui pelaksanaan MIC, yang merupakan perpustakaan yang mi nimizes c ommunication s cale dan dibahas dalam posting blog dekat-linear skala pelatihan raksasa model pada. AWS

Anda dapat menerapkan paralelisme data sharded ke model Anda sebagai strategi yang berdiri sendiri. Selain itu, jika Anda menggunakan instans GPU paling berkinerja yang dilengkapi dengan GPU NVIDIA A100 Tensor Core`ml.p4d.24xlarge`, Anda dapat memanfaatkan kecepatan latihan yang ditingkatkan dari operasi yang ditawarkan oleh SMDDP Collectives. `AllGather`

Untuk menyelami paralelisme data yang dipecah dan mempelajari cara mengaturnya atau menggunakan kombinasi paralelisme data yang dipecah dengan teknik lain seperti paralelisme tensor dan pelatihan FP16, lihat. the section called “Paralelisme Data Sharded”

Paralelisme pipa (tersedia untuk PyTorch dan) TensorFlow

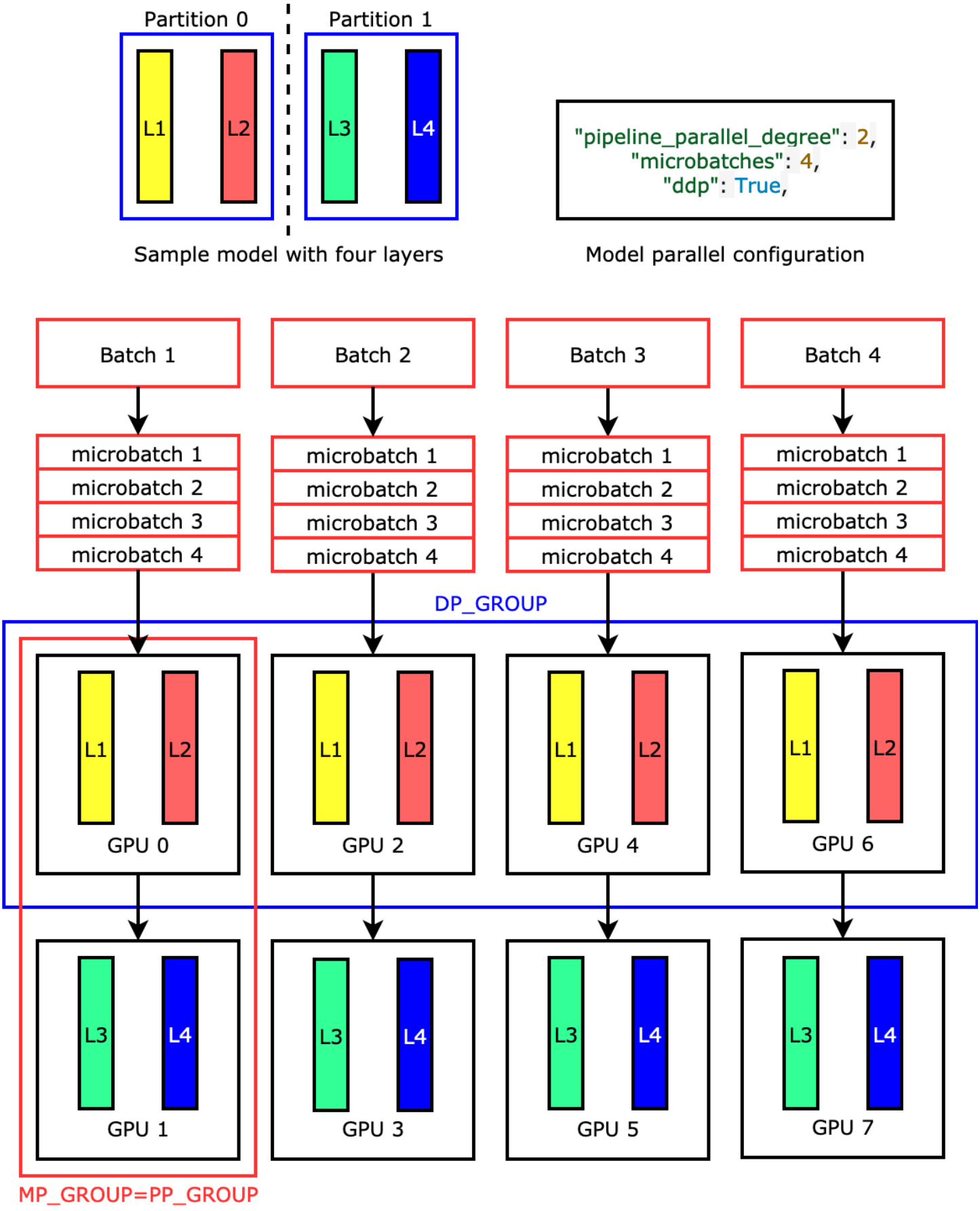
Paralelisme pipa partisi set lapisan atau operasi di seluruh set perangkat, meninggalkan setiap operasi utuh. Ketika Anda menentukan nilai untuk jumlah partisi model (`pipeline_parallel_degree`), jumlah total GPU (`processes_per_host`) harus dibagi dengan jumlah partisi model. Untuk mengatur ini dengan benar, Anda harus menentukan nilai yang benar untuk `pipeline_parallel_degree` dan `processes_per_host` parameter. Matematika sederhana adalah sebagai berikut:

```
(pipeline_parallel_degree) x (data_parallel_degree) = processes_per_host
```

Perpustakaan menangani menghitung jumlah replika model (juga disebut`data_parallel_degree`) mengingat dua parameter input yang Anda berikan.

Misalnya, jika Anda menyetel `"pipeline_parallel_degree": 2` dan `"processes_per_host": 8` menggunakan instans MLdengan delapan pekerja GPU seperti`ml.p3.16xlarge`, library secara otomatis menyiapkan model terdistribusi di seluruh GPU dan paralelisme data empat arah. Gambar berikut menggambarkan bagaimana model didistribusikan di delapan GPU yang mencapai paralelisme data empat arah dan paralelisme pipa dua arah. Setiap replika model, di mana kita mendefinisikannya sebagai grup paralel pipa dan memberi label sebagai`PP_GR0UP`, dipartisi di dua GPU. Setiap partisi model ditetapkan ke empat GPU, di mana

keempat replika partisi berada dalam grup paralel data dan diberi label sebagai. DP_GROUP Tanpa paralelisme tensor, kelompok paralel pipa pada dasarnya adalah kelompok paralel model.

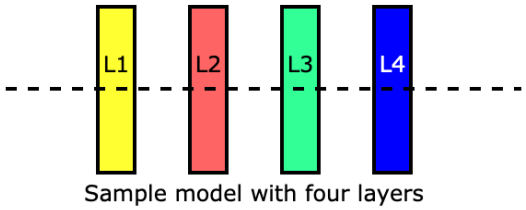

Partition 0
Partition 1
L1
L2
L3
L4
Sample model with four layers
"pipeline_parallel_degree": 2,
"microbatches": 4,
"ddp": True,
Model parallel configuration
Batch 1
Batch 2
Batch 3
Batch 4
microbatch 1
microbatch 2
microbatch 3
microbatch 4
DP_GROUP
GPU 0
GPU 2
GPU 4
GPU 6
GPU 1
GPU 3
GPU 5
GPU 7
MP_GROUP=PP_GROUP

Untuk menyelam jauh ke dalam paralelisme pipa, lihat. Fitur Inti dari Perpustakaan Paralelisme SageMaker Model

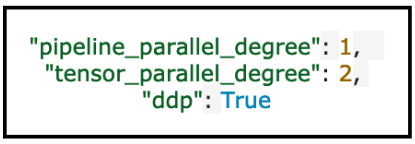
Untuk memulai menjalankan model Anda menggunakan paralelisme pipeline, lihat Menjalankan Pekerjaan Pelatihan SageMaker Terdistribusi dengan Pustaka Paralel SageMaker Model.

Paralelisme tensor (tersedia untuk) PyTorch

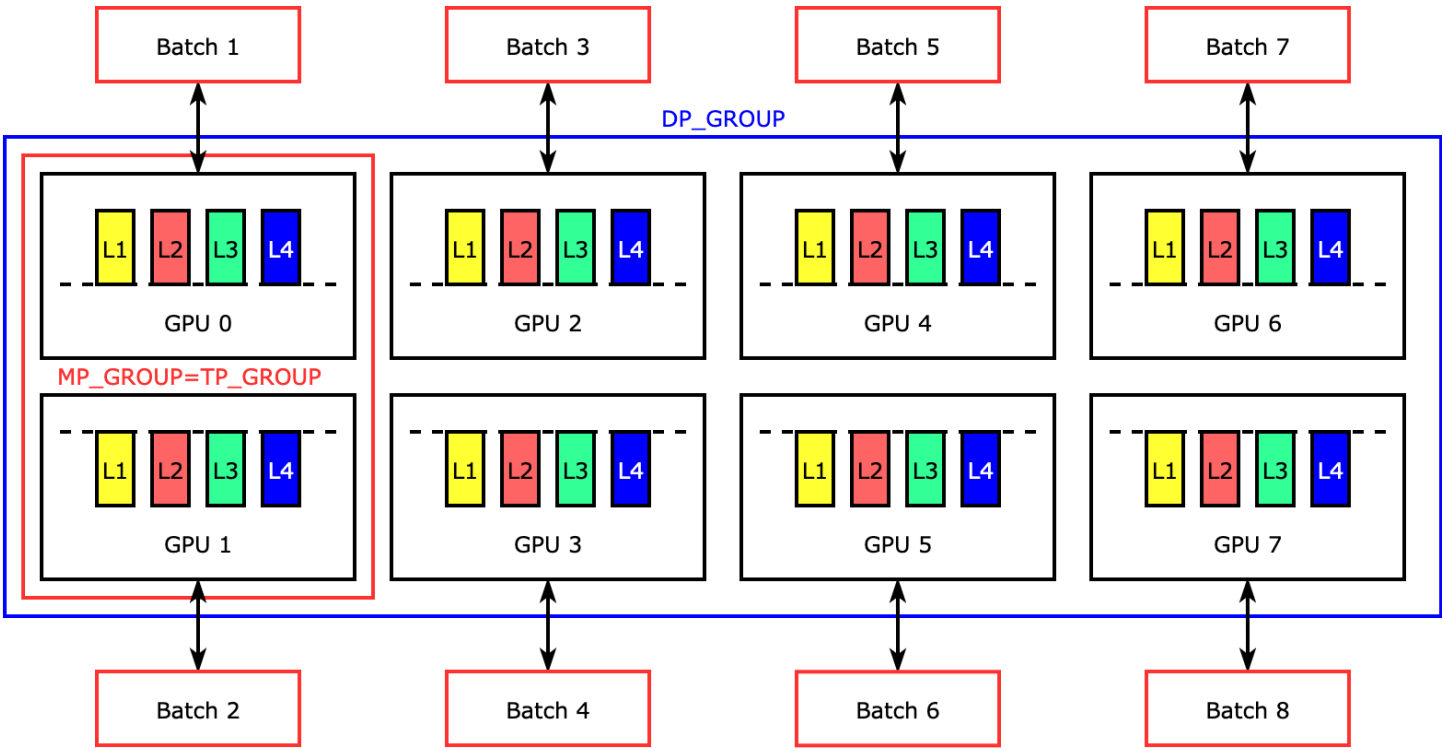
Paralelisme tensor membagi lapisan individu, atau, di seluruh perangkatnn `.Modules`, untuk dijalankan secara paralel. Gambar berikut menunjukkan contoh paling sederhana tentang bagaimana pustaka membagi model dengan empat lapisan untuk mencapai paralelisme tensor dua arah (). `"tensor_parallel_degree": 2` Lapisan masing-masing replika model dibelah dua dan didistribusikan menjadi dua GPU. Dalam contoh kasus ini, konfigurasi paralel model juga mencakup `"pipeline_parallel_degree": 1` dan `"ddp": True` (menggunakan PyTorch DistributedDataParallel paket di latar belakang), sehingga tingkat paralelisme data menjadi delapan. Pustaka mengelola komunikasi di seluruh replika model terdistribusi tensor.



Sample model with four layers



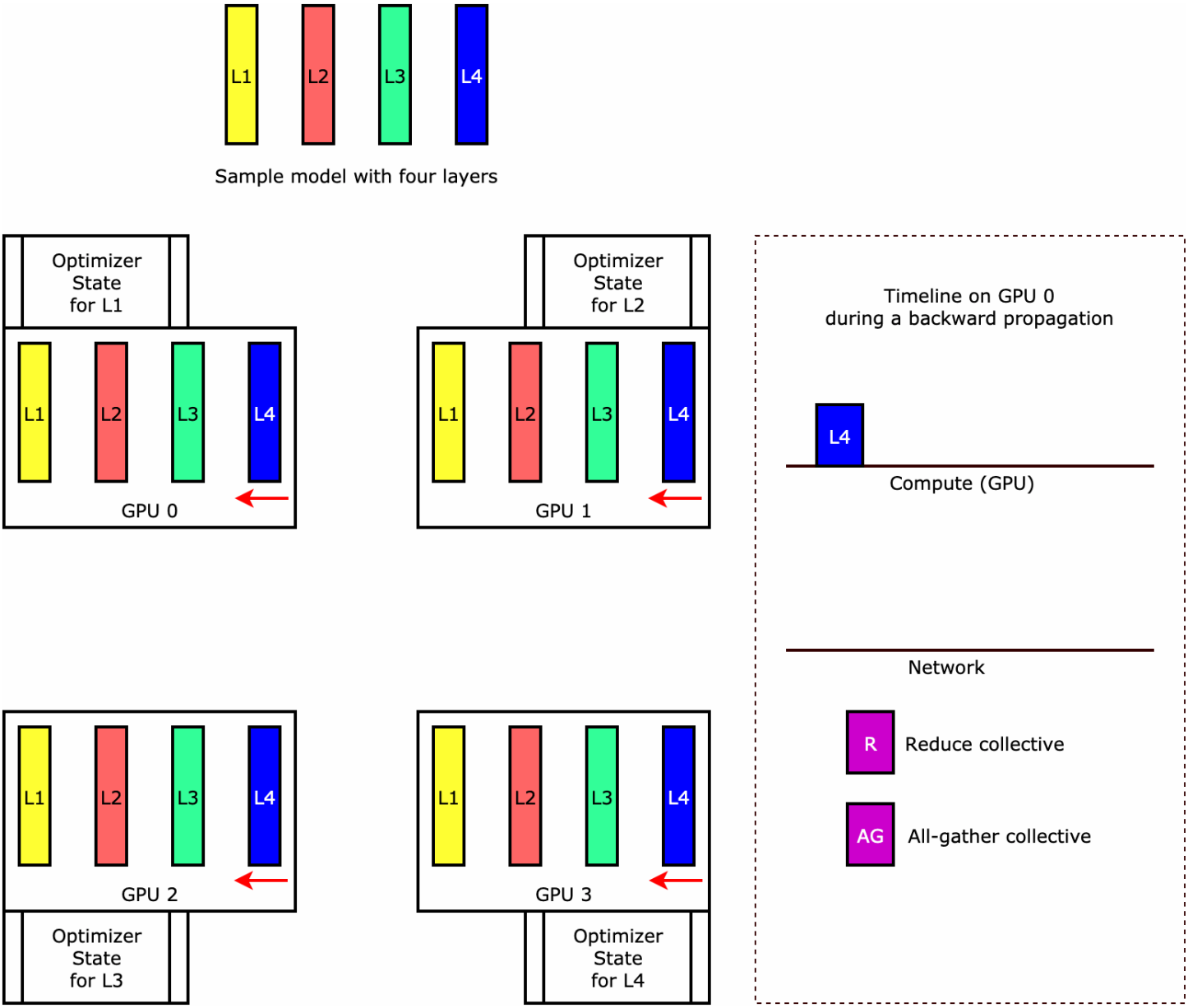
Model parallel configuration



Kegunaan fitur ini adalah kenyataan bahwa Anda dapat memilih lapisan tertentu atau subset lapisan untuk menerapkan paralelisme tensor. Untuk menyelami paralelisme tensor dan fitur hemat memori lainnyaPyTorch, dan mempelajari cara mengatur kombinasi paralelisme pipeline dan tensor, lihat. Paralelisme Tensor

Sharding status pengoptimal (tersedia untuk) PyTorch

Untuk memahami bagaimana library melakukan sharding state optimizer, pertimbangkan model contoh sederhana dengan empat layer. Ide utama dalam mengoptimalkan sharding status adalah Anda tidak perlu mereplikasi status pengoptimal Anda di semua GPU Anda. Sebagai gantinya, satu replika status pengoptimal dipecah di peringkat paralel data, tanpa redundansi di seluruh perangkat. Misalnya, GPU 0 memegang status optimizer untuk layer satu, GPU 1 berikutnya memegang status optimizer untuk L2, dan seterusnya. Gambar animasi berikut menunjukkan propagasi mundur dengan teknik sharding negara optimizer. Pada akhir propagasi mundur, ada waktu komputasi dan jaringan untuk operasi `optimizer apply` (OA) untuk memperbarui status pengoptimal dan operasi `all-gather` (AG) untuk memperbarui parameter model untuk iterasi berikutnya. Yang terpenting, `reduce` operasi dapat tumpang tindih dengan komputasi pada GPU 0, menghasilkan propagasi mundur yang lebih efisien memori dan lebih cepat. Dalam implementasi saat ini, operasi AG dan OA tidak tumpang tindih dengan`compute`. Hal ini dapat mengakibatkan perhitungan diperpanjang selama operasi AG, sehingga mungkin ada tradeoff.



Untuk informasi selengkapnya tentang cara menggunakan fitur ini, lihat Optimizer State Sharding.

Aktivasi pembongkaran dan checkpointing (tersedia untuk) PyTorch

Untuk menyimpan memori GPU, pustaka mendukung checkpointing aktivasi untuk menghindari penyimpanan aktivasi internal dalam memori GPU untuk modul yang ditentukan pengguna selama pass forward. Library menghitung ulang aktivasi ini selama pass mundur. Selain itu, fitur pembongkaran aktivasi membongkar aktivasi yang tersimpan ke memori CPU dan mengambil kembali ke GPU selama pass mundur untuk lebih mengurangi jejak memori aktivasi. Untuk informasi selengkapnya tentang cara menggunakan fitur ini, lihat Activation Checkpointing dan Activation Offloading.

Memilih teknik yang tepat untuk model Anda

Untuk informasi selengkapnya tentang memilih teknik dan konfigurasi yang tepat, lihat SageMakerDistributed Model Parallel Best Practices and Configuration Tips and Pitfalls.

Kerangka Kerja yang Didukung dan Wilayah AWS

Sebelum menggunakan pustaka paralelisme SageMaker model, periksa kerangka kerja dan jenis instance yang didukung, dan tentukan apakah ada cukup kuota di akun Anda dan. AWS Wilayah AWS

> **Note**
>
> Untuk memeriksa pembaruan terbaru dan catatan rilis perpustakaan, lihat Catatan Rilis Paralel SageMaker Model dalam dokumentasi SageMaker Python SDK.

Kerangka Kerja yang Didukung

Pustaka paralelisme SageMaker model mendukung kerangka pembelajaran mendalam berikut dan tersedia dalam AWS Deep Learning Containers (DLC) atau dapat diunduh sebagai file biner.

PyTorch versi yang didukung oleh SageMaker dan pustaka paralelisme SageMaker model

PyTorch versi	SageMaker model versi perpustakaan paralelisme	**`smdistributed-modelparallel`** URI gambar DLC terintegrasi	URL dari file biner**
v2.0.0	`smdistributed-modelparallel==v1.15.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:2.0.0-gpu-py310-cu118-ubuntu20.04-sagemaker`	https://. s3.us-west-2.amazonaws.com sagemaker-distributed-model-parallel / pytorch-2.0.0/build-artifacts/2023-04-14-20-14/smdistributed_modelparallel-1.15.0-cp310-cp310-linux_x86_64.whl

PyTorch versi	SageMaker model versi perpustakaan paralelisme	**smdistributed-modelparallel** URI gambar DLC terintegrasi	URL dari file biner**
v1.13.1	`smdistributed-modelparallel==v1.15.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.13.1-gpu-py39-cu117-ubuntu20.04-sagemaker`	https://. s3.us-west-2.amazonaws.com sagemaker-distributed-model-parallel / pytorch-1.13.1/build-artifacts/2023-04-17-15-49/smdistributed_modelparallel-1.15.0-cp39-cp39-linux_x86_64.whl
v1.12.1	`smdistributed-modelparallel==v1.13.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.12.1-gpu-py38-cu113-ubuntu20.04-sagemaker`	https://. s3.us-west-2.amazonaws.com sagemaker-distributed-model-parallel / pytorch-1.12.1/build-artifacts/2022-12-08-21-34/smdistributed_modelparallel-1.13.0-cp38-cp38-linux_x86_64.whl
v1.12.0	`smdistributed-modelparallel==v1.11.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.12.0-gpu-py38-cu113-ubuntu20.04-sagemaker`	https://. s3.us-west-2.amazonaws.com sagemaker-distributed-model-parallel / pytorch-1.12.0/build-artifacts/2022-08-12-16-58/smdistributed_modelparallel-1.11.0-cp38-cp38-linux_x86_64.whl

PyTorch versi	SageMaker model versi perpustakaan paralelisme	**smdistributed-modelparallel** URI gambar DLC terintegrasi	URL dari file biner**
v1.11.0	`smdistributed-modelparallel==v1.10.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.11.0-gpu-py38-cu113-ubuntu20.04-sagemaker`	https://. s3.us-west-2.amazonaws.com sagemaker-distributed-model-parallel / pytorch-1.11.0/build-artifacts/2022-07-11-19-23/smdistributed_modelparallel-1.10.0-cp38-cp38-linux_x86_64.whl
v1.10.2	`smdistributed-modelparallel==v1.7.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.10.2-gpu-py38-cu113-ubuntu20.04-sagemaker`	-
v1.10.0	`smdistributed-modelparallel==v1.5.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.10.0-gpu-py38-cu113-ubuntu20.04-sagemaker`	-

PyTorch versi	SageMaker model versi perpustakaan paralelisme	**smdistributed-modelparallel** URI gambar DLC terintegrasi	URL dari file biner**
v1.9.1	`smdistributed-modelparallel==v1.4.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.9.1-gpu-py38-cu111-ubuntu20.04`	-
v1.8.1*	`smdistributed-modelparallel==v1.6.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-training:1.8.1-gpu-py36-cu111-ubuntu18.04`	-

> **Note**
>
> Pustaka paralelisme SageMaker model v1.6.0 dan yang lebih baru menyediakan fitur tambahan untuk. PyTorch Untuk informasi selengkapnya, lihat Fitur Inti dari Perpustakaan Paralelisme SageMaker Model.

** URL file biner adalah untuk menginstal pustaka paralelisme SageMaker model dalam wadah khusus. Untuk informasi selengkapnya, lihat the section called “Buat Container Docker Anda Sendiri dengan Library”.

TensorFlow versi yang didukung oleh SageMaker dan pustaka paralelisme SageMaker model

TensorFlow versi	SageMaker model versi perpustakaan paralelisme	**`smdistributed-modelparallel`** URI gambar DLC terintegrasi
v2.6.0	`smdistributed-modelparallel==v1.4.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/tensorflow-training:2.6.0-gpu-py38-cu112-ubuntu20.04`
v2.5.1	`smdistributed-modelparallel==v1.4.0`	`763104351884.dkr.ecr.<region>.amazonaws.com/tensorflow-training:2.5.1-gpu-py37-cu112-ubuntu18.04`

Versi Hugging Face Transformers didukung SageMaker oleh dan perpustakaan paralel data SageMaker terdistribusi

AWSDeep Learning Containers untuk Hugging Face menggunakan SageMaker Wadah Pelatihan PyTorch untuk TensorFlow dan sebagai gambar dasarnya. Untuk mencari versi pustaka Hugging Face Transformers dan PyTorch dipasangkan TensorFlow dan versi, lihat Wadah Wajah Pelukan terbaru dan Versi Wadah Wajah Pelukan Sebelumnya.

Wilayah AWS

Pustaka paralel SageMaker data tersedia di semua Wilayah AWS tempat AWSDeep Learning Containers SageMaker berada dalam layanan. Untuk informasi selengkapnya, lihat Gambar Deep Learning Containers yang Tersedia.

Tipe Instans Yang Didukung

Pustaka paralelisme SageMaker model memerlukan salah satu jenis instance ML berikut.

Jenis instans
ml.g4dn.12xlarge
ml.p3.16xlarge
ml.p3dn.24xlarge
ml.p4d.24xlarge
ml.p4de.24xlarge

Untuk spesifikasi jenis instans, lihat bagian Komputasi Akselerasi di halaman Jenis Instans Amazon EC2. Untuk informasi tentang harga instans, lihat SageMakerHarga Amazon.

Jika Anda menemukan pesan galat yang mirip dengan berikut ini, ikuti petunjuk di Minta peningkatan kuota layanan untuk SageMaker sumber daya.

```
ResourceLimitExceeded: An error occurred (ResourceLimitExceeded) when calling
    the CreateTrainingJob operation: The account-level service limit 'ml.p3dn.24xlarge
    for training job usage' is 0 Instances, with current utilization of 0 Instances
    and a request delta of 1 Instances.
    Please contact AWS support to request an increase for this limit.
```

Fitur Inti dari Perpustakaan Paralelisme SageMaker Model

Pustaka paralelisme model Amazon SageMaker menawarkan strategi distribusi dan teknik penghematan memori, seperti paralelisme data sharded, paralelisme tensor, partisi model berdasarkan lapisan untuk penjadwalan pipa, dan checkpointing. Strategi dan teknik paralelisme model membantu mendistribusikan model besar di beberapa perangkat sambil mengoptimalkan kecepatan pelatihan dan konsumsi memori. Pustaka juga menyediakan fungsi pembantu Python, manajer konteks, dan fungsi pembungkus untuk mengadaptasi skrip pelatihan Anda untuk partisi otomatis atau manual model Anda.

Saat Anda menerapkan paralelisme model ke pekerjaan pelatihan Anda, Anda mempertahankan alur kerja dua langkah yang sama yang ditunjukkan di bagian Jalankan Pekerjaan SageMaker Pelatihan Terdistribusi dengan Paralelisme Model. Untuk mengadaptasi skrip pelatihan Anda, Anda akan menambahkan nol atau beberapa baris kode tambahan ke skrip pelatihan Anda. Untuk meluncurkan

tugas pelatihan skrip pelatihan yang disesuaikan, Anda harus mengatur parameter konfigurasi distribusi untuk mengaktifkan fitur hemat memori atau meneruskan nilai untuk tingkat paralelisme.

Untuk memulai dengan contoh, lihat buku catatan Jupyter berikut yang menunjukkan cara menggunakan pustaka paralelisme SageMaker model.

- PyTorch contoh notebook
- TensorFlow contoh notebook

Untuk menyelam jauh ke dalam fitur inti perpustakaan, lihat topik berikut.

> **Note**
>
> Perpustakaan pelatihan SageMaker terdistribusi tersedia melalui wadah pembelajaran AWS mendalam untuk PyTorch, Hugging Face, dan TensorFlow dalam platform SageMaker Pelatihan. Untuk memanfaatkan fitur perpustakaan pelatihan terdistribusi, kami sarankan Anda menggunakan Python SageMaker SDK. Anda juga dapat mengonfigurasi secara manual dalam sintaks permintaan JSON jika Anda menggunakan SageMaker API melalui SDK for Python (Boto3) atau. AWS Command Line Interface Sepanjang dokumentasi, instruksi dan contoh berfokus pada cara menggunakan pustaka pelatihan terdistribusi dengan SageMaker Python SDK.

> **Important**
>
> Pustaka paralelisme SageMaker model mendukung semua fitur inti untuk PyTorch, dan mendukung paralelisme pipa untuk. TensorFlow

Topik

- Paralelisme Data Sharded
- Pipelining Model
- Paralelisme Tensor
- Sharding Status Optimizer
- Aktivasi Checkpointing
- Pembongkaran Aktivasi

- Pelatihan FP16 dengan Paralelisme Model
- Support untuk FlashAttention

Paralelisme Data Sharded

Paralelisme data sharded adalah teknik pelatihan terdistribusi hemat memori yang membagi status model (parameter model, gradien, dan status pengoptimal) di seluruh GPU dalam grup paralel data.

> **Note**
>
> Paralelisme data sharded tersedia untuk PyTorch di pustaka paralelisme SageMaker model v1.11.0 dan yang lebih baru.

Saat meningkatkan pekerjaan pelatihan Anda ke cluster GPU yang besar, Anda dapat mengurangi jejak memori per-GPU model dengan membagi status pelatihan model melalui beberapa GPU. Ini mengembalikan dua manfaat: Anda dapat memasukkan model yang lebih besar, yang jika tidak akan kehabisan memori dengan paralelisme data standar, atau Anda dapat meningkatkan ukuran batch menggunakan memori GPU yang dibebaskan.

Teknik paralelisme data standar mereplikasi status pelatihan di seluruh GPU dalam grup paralel data, dan melakukan agregasi gradien berdasarkan operasi. `AllReduce` Paralelisme data sharded memodifikasi prosedur pelatihan terdistribusi data-paralel standar untuk memperhitungkan sifat sharded dari status pengoptimal. Sekelompok peringkat di mana status model dan pengoptimal dipecah disebut grup sharding. Teknik paralelisme data sharded memecah parameter model yang dapat dilatih dan gradien serta status pengoptimal yang sesuai di seluruh GPU dalam grup sharding.

SageMaker mencapai paralelisme data yang dibagikan melalui implementasi MIC, yang dibahas dalam posting AWS blog Penskalaan hampir linier dari pelatihan model raksasa pada. AWS Dalam implementasi ini, Anda dapat mengatur derajat sharding sebagai parameter yang dapat dikonfigurasi, yang harus kurang dari tingkat paralelisme data. Selama setiap lintasan maju dan mundur, MIC untuk sementara menggabungkan kembali parameter model di semua GPU melalui operasi. `AllGather` Setelah pass maju atau mundur dari setiap lapisan, MIC memecah parameter lagi untuk menghemat memori GPU. Selama lintasan mundur, MIC mengurangi gradien dan secara bersamaan memecahnya di seluruh GPU melalui operasi. `ReduceScatter` Terakhir, MIC menerapkan gradien lokal tereduksi dan sharded ke pecahan parameter lokal yang sesuai, menggunakan pecahan lokal status pengoptimal. Untuk menurunkan overhead komunikasi, pustaka paralelisme SageMaker model

mengambil lapisan yang akan datang di pass maju atau mundur, dan tumpang tindih komunikasi jaringan dengan komputasi.

Status pelatihan model direplikasi di seluruh kelompok sharding. Ini berarti bahwa sebelum gradien diterapkan pada parameter, `AllReduce` operasi harus dilakukan di seluruh kelompok sharding, selain `ReduceScatter` operasi yang terjadi dalam grup sharding.

Akibatnya, paralelisme data sharded memperkenalkan tradeoff antara overhead komunikasi dan efisiensi memori GPU. Menggunakan paralelisme data sharded meningkatkan biaya komunikasi, tetapi jejak memori per GPU (tidak termasuk penggunaan memori karena aktivasi) dibagi dengan tingkat paralelisme data sharded, sehingga model yang lebih besar dapat dimasukkan ke dalam cluster GPU.

Memilih tingkat paralelisme data sharded

Ketika Anda memilih nilai untuk tingkat paralelisme data sharded, nilai harus membagi derajat paralelisme data secara merata. Misalnya, untuk pekerjaan paralelisme data 8 arah, pilih 2, 4, atau 8 untuk tingkat paralelisme data sharded. Saat memilih tingkat paralelisme data yang dipecah, kami menyarankan Anda memulai dengan jumlah kecil, dan secara bertahap meningkatkannya hingga model sesuai dengan memori bersama dengan ukuran batch yang diinginkan.

Memilih ukuran batch

Setelah menyiapkan paralelisme data sharded, pastikan Anda menemukan konfigurasi pelatihan paling optimal yang berhasil dijalankan di cluster GPU. Untuk melatih model bahasa besar (LLM), mulai dari ukuran batch 1, dan secara bertahap tingkatkan hingga Anda mencapai titik untuk menerima kesalahan out-of-memory (OOM). Jika Anda menemukan kesalahan OOM bahkan dengan ukuran batch terkecil, terapkan tingkat paralelisme data sharded yang lebih tinggi atau kombinasi paralelisme data sharded dan paralelisme tensor.

Topik

Cara menerapkan paralelisme data sharded ke pekerjaan pelatihan Anda

Untuk memulai paralelisme data sharded, terapkan modifikasi yang diperlukan pada skrip pelatihan Anda, dan atur SageMaker PyTorch estimator dengan parameter. sharded-data-parallelism-specific Pertimbangkan juga untuk mengambil nilai referensi dan contoh notebook sebagai titik awal.

Sesuaikan skrip PyTorch pelatihan Anda

Ikuti instruksi di Langkah 1: Ubah Skrip PyTorch Pelatihan untuk membungkus objek model dan pengoptimal dengan `smdistributed.modelparallel.torch` pembungkus modul `torch.nn.parallel` dan`torch.distributed`.

(Opsional) Modifikasi tambahan untuk mendaftarkan parameter model eksternal

Jika model Anda dibangun dengan `torch.nn.Module` dan menggunakan parameter yang tidak ditentukan dalam kelas modul, Anda harus mendaftarkannya ke modul secara manual untuk SMP untuk mengumpulkan parameter penuh sementara. Untuk mendaftarkan parameter ke modul, gunakan`smp.register_parameter(module, parameter)`.

```
class Module(torch.nn.Module):
    def __init__(self, *args):
        super().__init__(self, *args)
        self.layer1 = Layer1()
        self.layer2 = Layer2()
        smp.register_parameter(self, self.layer1.weight)

    def forward(self, input):
        x = self.layer1(input)
        # self.layer1.weight is required by self.layer2.forward
        y = self.layer2(x, self.layer1.weight)
        return y
```

Siapkan SageMaker PyTorch estimator

Saat mengonfigurasi SageMaker PyTorch estimatorthe section called “Langkah 2: Luncurkan Training Job”, tambahkan parameter untuk paralelisme data sharded.

Untuk mengaktifkan paralelisme data sharded, tambahkan `sharded_data_parallel_degree` parameter ke Estimator. SageMaker PyTorch Parameter ini menentukan jumlah GPU di mana status pelatihan dipecah. Nilai untuk `sharded_data_parallel_degree` harus berupa bilangan bulat antara satu dan derajat paralelisme data dan harus membagi derajat paralelisme data secara merata.

Perhatikan bahwa pustaka secara otomatis mendeteksi jumlah GPU sehingga tingkat paralel data. Parameter tambahan berikut tersedia untuk mengonfigurasi paralelisme data sharded.

- `"sdp_reduce_bucket_size"`(int, default: 5e8) - Menentukan ukuran ember gradien PyTorch DDP dalam jumlah elemen dtype default.
- `"sdp_param_persistence_threshold"`(int, default: 1e6) — Menentukan ukuran tensor parameter dalam jumlah elemen yang dapat bertahan di setiap GPU. Paralelisme data sharded membagi setiap tensor parameter di seluruh GPU dari grup paralel data. Jika jumlah elemen dalam tensor parameter lebih kecil dari ambang batas ini, tensor parameter tidak terbelah; ini membantu mengurangi overhead komunikasi karena tensor parameter direplikasi di seluruh GPU paralel data.
- `"sdp_max_live_parameters"`(int, default: 1e9) — Menentukan jumlah maksimum parameter yang secara bersamaan dapat berada dalam keadaan pelatihan rekombinasi selama pass maju dan mundur. Pengambilan parameter dengan `AllGather` operasi berhenti ketika jumlah parameter aktif mencapai ambang batas yang diberikan. Perhatikan bahwa meningkatkan parameter ini meningkatkan jejak memori.
- `"sdp_hierarchical_allgather"`(bool, default: True) - Jika disetel ke`True`, `AllGather` operasi berjalan secara hierarkis: berjalan di dalam setiap node terlebih dahulu, dan kemudian berjalan melintasi node. Untuk pekerjaan pelatihan terdistribusi multi-node, `AllGather` operasi hierarkis diaktifkan secara otomatis.
- `"sdp_gradient_clipping"`(float, default: 1.0) — Menentukan ambang batas untuk gradien memotong norma L2 dari gradien sebelum menyebarkannya mundur melalui parameter model. Ketika paralelisme data sharded diaktifkan, kliping gradien juga diaktifkan. Ambang batas default adalah`1.0`. Sesuaikan parameter ini jika Anda memiliki masalah gradien yang meledak.

Kode berikut menunjukkan contoh cara mengkonfigurasi paralelisme data sharded.

```
import sagemaker
from sagemaker.pytorch import PyTorch

smp_options = {
    "enabled": True,
    "parameters": {
        # "pipeline_parallel_degree": 1,    # Optional, default is 1
        # "tensor_parallel_degree": 1,      # Optional, default is 1
        "ddp": True,
        # parameters for sharded data parallelism
        "sharded_data_parallel_degree": 2,              # Add this to activate sharded
 data parallelism
```

```
        "sdp_reduce_bucket_size": int(5e8),              # Optional
        "sdp_param_persistence_threshold": int(1e6),     # Optional
        "sdp_max_live_parameters": int(1e9),             # Optional
        "sdp_hierarchical_allgather": True,              # Optional
        "sdp_gradient_clipping": 1.0                     # Optional
    }
}

mpi_options = {
    "enabled" : True,                      # Required
    "processes_per_host" : 8               # Required
}

smp_estimator = PyTorch(
    entry_point="your_training_script.py", # Specify your train script
    role=sagemaker.get_execution_role(),
    instance_count=1,
    instance_type='ml.p3.16xlarge',
    framework_version='1.13.1',
    py_version='py3',
    distribution={
        "smdistributed": {"modelparallel": smp_options},
        "mpi": mpi_options
    },
    base_job_name="sharded-data-parallel-job"
)

smp_estimator.fit('s3://my_bucket/my_training_data/')
```

Konfigurasi referensi

Tim pelatihan SageMaker terdistribusi menyediakan konfigurasi referensi berikut yang dapat Anda gunakan sebagai titik awal. Anda dapat mengekstrapolasi dari konfigurasi berikut untuk bereksperimen dan memperkirakan penggunaan memori GPU untuk konfigurasi model Anda.

Paralelisme data sharded dengan SMDDP Collectives

Model/ jumlah parameter	Contoh num	Jenis instans	Panjang urutan	Ukuran batch global	Ukuran batch mini	Derajat paralel data sharded
GPT-NEOX-20B	2	ml.p4d.24 xlarge	2048	64	4	16
GPT-NEOX-20B	8	ml.p4d.24 xlarge	2048	768	12	32

Misalnya, jika Anda menambah panjang urutan untuk model 20 miliar parameter atau menambah ukuran model menjadi 65 miliar parameter, Anda perlu mencoba mengurangi ukuran batch terlebih dahulu. Jika model masih tidak sesuai dengan ukuran batch terkecil (ukuran batch 1), coba tingkatkan derajat paralelisme model.

Paralelisme data sharded dengan paralelisme tensor dan NCCL Collectives

Model/ jumlah parameter	Contoh num	Jenis instans	Panjang urutan	Ukuran batch global	Ukuran batch mini	Derajat paralel data sharded	Derajat paralel tensor	Pembongkaran aktivasi
GPT-NEOX-65B	64	ml.p4d.24 xlarge	2048	512	8	16	8	Y
GPT-NEOX-65B	64	ml.p4d.24 xlarge	4096	512	2	64	2	Y

Penggunaan gabungan paralelisme data sharded dan paralelisme tensor berguna ketika Anda ingin memasukkan model bahasa besar (LLM) ke dalam cluster skala besar saat menggunakan data teks dengan panjang urutan yang lebih panjang, yang mengarah pada penggunaan ukuran batch yang lebih kecil, dan akibatnya menangani penggunaan memori GPU untuk melatih LLM terhadap urutan

teks yang lebih panjang. Untuk mempelajari selengkapnya, lihat the section called “Paralelisme data sharded dengan paralelisme tensor”.

Untuk studi kasus, tolok ukur, dan contoh konfigurasi lainnya, lihat posting blog Peningkatan kinerja baru di perpustakaan paralel SageMaker model Amazon.

Paralelisme data sharded dengan SMDDP Collectives

Perpustakaan paralelisme SageMaker data menawarkan primitif komunikasi kolektif (kolektif SMDDP) yang dioptimalkan untuk infrastruktur. AWS Ini mencapai optimasi dengan mengadopsi pola all-to-all-type komunikasi dengan memanfaatkan Elastic Fabric Adapter (EFA), menghasilkan throughput tinggi dan kolektif yang kurang sensitif terhadap latensi, menurunkan pemrosesan terkait komunikasi ke CPU, dan membebaskan siklus GPU untuk komputasi. Pada cluster besar, SMDDP Collectives dapat menawarkan peningkatan kinerja pelatihan terdistribusi hingga 40% dibandingkan dengan NCCL. Untuk studi kasus dan hasil benchmark, lihat blog Peningkatan kinerja baru di perpustakaan paralelisme SageMaker model Amazon.

> **Note**
>
> Paralelisme data sharded dengan SMDDP Collectives tersedia di perpustakaan paralelisme SageMaker model v1.13.0 dan yang lebih baru, dan perpustakaan paralelisme data v1.6.0 dan yang lebih baru. SageMaker Lihat juga Supported configurations untuk menggunakan paralelisme data sharded dengan SMDDP Collectives.

Dalam paralelisme data sharded, yang merupakan teknik yang umum digunakan dalam pelatihan terdistribusi skala besar, `AllGather` kolektif digunakan untuk menyusun kembali parameter lapisan sharded untuk komputasi pass maju dan mundur, secara paralel dengan komputasi GPU. Untuk model besar, melakukan `AllGather` operasi secara efisien sangat penting untuk menghindari masalah kemacetan GPU dan memperlambat kecepatan pelatihan. Ketika paralelisme data sharded diaktifkan, SMDDP Collectives masuk ke dalam kolektif kritis kinerja `AllGather` ini, meningkatkan throughput pelatihan.

Berlatih dengan SMDDP Collectives

Ketika pekerjaan pelatihan Anda telah berbagi paralelisme data diaktifkan dan memenuhiSupported configurations, SMDDP Collectives diaktifkan secara otomatis. Secara internal, SMDDP Collectives mengoptimalkan `AllGather` kolektif untuk berkinerja pada AWS infrastruktur dan kembali ke NCCL

untuk semua kolektif lainnya. Selanjutnya, di bawah konfigurasi yang tidak didukung, semua kolektif, termasuk`AllGather`, secara otomatis menggunakan backend NCCL.

Karena pustaka paralelisme SageMaker model versi 1.13.0, `"ddp_dist_backend"` parameter ditambahkan ke opsi. `modelparallel` Nilai default untuk parameter konfigurasi ini adalah`"auto"`, yang menggunakan SMDDP Collectives bila memungkinkan, dan kembali ke NCCL sebaliknya. Untuk memaksa perpustakaan untuk selalu menggunakan NCCL, tentukan `"nccl"` ke parameter `"ddp_dist_backend"` konfigurasi.

Contoh kode berikut menunjukkan cara mengatur PyTorch estimator menggunakan paralelisme data sharded dengan `"ddp_dist_backend"` parameter, yang diatur ke default dan, `"auto"` oleh karena itu, opsional untuk ditambahkan.

```
import sagemaker
from sagemaker.pytorch import PyTorch

smp_options = {
    "enabled":True,
    "parameters": {
        "partitions": 1,
        "ddp": True,
        "sharded_data_parallel_degree": 64
        "bf16": True,
        "ddp_dist_backend": "auto"  # Specify "nccl" to force to use NCCL.
    }
}

mpi_options = {
    "enabled" : True,                      # Required
    "processes_per_host" : 8               # Required
}

smd_mp_estimator = PyTorch(
    entry_point="your_training_script.py", # Specify your train script
    source_dir="location_to_your_script",
    role=sagemaker.get_execution_role(),
    instance_count=8,
    instance_type='ml.p4d.24xlarge',
    framework_version='1.13.1',
    py_version='py3',
    distribution={
        "smdistributed": {"modelparallel": smp_options},
```

```
        "mpi": mpi_options
    },
    base_job_name="sharded-data-parallel-demo",
)

smd_mp_estimator.fit('s3://my_bucket/my_training_data/')
```

Konfigurasi yang didukung

`AllGather`Operasi dengan SMDDP Collectives diaktifkan dalam pekerjaan pelatihan ketika semua persyaratan konfigurasi berikut terpenuhi.

- Tingkat paralelisme data sharded lebih besar dari 1
- `Instance_count`lebih besar dari 1
- `Instance_type`sama dengan `ml.p4d.24xlarge`
- SageMaker wadah pelatihan untuk PyTorch v1.12.1 atau yang lebih baru
- Pustaka paralelisme SageMaker data v1.6.0 atau yang lebih baru
- Pustaka paralelisme SageMaker model v1.13.0 atau yang lebih baru

Kinerja dan penyetelan memori

SMDDP Collectives menggunakan memori GPU tambahan. Ada dua variabel lingkungan untuk mengonfigurasi penggunaan memori GPU tergantung pada kasus penggunaan pelatihan model yang berbeda.

- `SMDDP_AG_SCRATCH_BUFFER_SIZE_BYTES`— Selama `AllGather` operasi SMDDP, buffer `AllGather` input disalin ke buffer sementara untuk komunikasi antar simpul. `SMDDP_AG_SCRATCH_BUFFER_SIZE_BYTES`Variabel mengontrol ukuran (dalam byte) dari buffer sementara ini. Jika ukuran buffer sementara lebih kecil dari ukuran buffer `AllGather` input, `AllGather` kolektif akan kembali menggunakan NCCL.
  - Nilai default: 16 * 1024 * 1024 (16 MB)
  - Nilai yang dapat diterima: kelipatan 8192

- `SMDDP_AG_SORT_BUFFER_SIZE_BYTES``SMDDP_AG_SORT_BUFFER_SIZE_BYTES`Variabelnya adalah untuk mengukur buffer sementara (dalam byte) untuk menyimpan data yang dikumpulkan dari komunikasi antar simpul. Jika ukuran buffer sementara ini lebih kecil dari`1/8 *`

`sharded_data_parallel_degree * AllGather input size`, `AllGather` kolektif akan kembali menggunakan NCCL.

- Nilai default: 128 * 1024 * 1024 (128 MB)
- Nilai yang dapat diterima: kelipatan 8192

Panduan penyetelan pada variabel ukuran buffer

Nilai default untuk variabel lingkungan harus berfungsi dengan baik untuk sebagian besar kasus penggunaan. Kami merekomendasikan untuk menyetel variabel-variabel ini hanya jika pelatihan mengalami kesalahan out-of-memory (OOM).

Daftar berikut membahas beberapa tips penyetelan untuk mengurangi jejak memori GPU SMDDP Collectives sambil mempertahankan keuntungan kinerja dari mereka.

- Penyetelan `SMDDP_AG_SCRATCH_BUFFER_SIZE_BYTES`
  - Ukuran buffer `AllGather` input lebih kecil untuk model yang lebih kecil. Oleh karena itu, ukuran yang diperlukan untuk `SMDDP_AG_SCRATCH_BUFFER_SIZE_BYTES` bisa lebih kecil untuk model dengan parameter lebih sedikit.
  - Ukuran buffer `AllGather` input berkurang seiring `sharded_data_parallel_degree` bertambahnya, karena model akan dibagi di lebih banyak GPU. Oleh karena itu, ukuran yang diperlukan untuk `SMDDP_AG_SCRATCH_BUFFER_SIZE_BYTES` bisa lebih kecil untuk pekerjaan pelatihan dengan nilai besar untuk`sharded_data_parallel_degree`.
- Penyetelan `SMDDP_AG_SORT_BUFFER_SIZE_BYTES`
  - Jumlah data yang dikumpulkan dari komunikasi antar simpul lebih sedikit untuk model dengan parameter yang lebih sedikit. Oleh karena itu, ukuran yang diperlukan untuk `SMDDP_AG_SORT_BUFFER_SIZE_BYTES` bisa lebih kecil untuk model tersebut dengan jumlah parameter yang lebih sedikit.

Beberapa kolektif mungkin kembali menggunakan NCCL; karenanya, Anda mungkin tidak mendapatkan keuntungan kinerja dari kolektif SMDDP yang dioptimalkan. Jika memori GPU tambahan tersedia untuk digunakan, Anda dapat mempertimbangkan untuk meningkatkan nilai `SMDDP_AG_SCRATCH_BUFFER_SIZE_BYTES` dan `SMDDP_AG_SORT_BUFFER_SIZE_BYTES` untuk mendapatkan keuntungan dari peningkatan kinerja.

Kode berikut menunjukkan bagaimana Anda dapat mengkonfigurasi variabel lingkungan dengan menambahkannya ke `mpi_options` dalam parameter distribusi untuk PyTorch estimator.

```
import sagemaker
from sagemaker.pytorch import PyTorch

smp_options = {
    .... # All modelparallel configuration options go here
}

mpi_options = {
    "enabled" : True,                       # Required
    "processes_per_host" : 8                # Required
}

# Use the following two lines to tune values of the environment variables for buffer
mpioptions += " -x SMDDP_AG_SCRATCH_BUFFER_SIZE_BYTES=8192"
mpioptions += " -x SMDDP_AG_SORT_BUFFER_SIZE_BYTES=8192"

smd_mp_estimator = PyTorch(
    entry_point="your_training_script.py", # Specify your train script
    source_dir="location_to_your_script",
    role=sagemaker.get_execution_role(),
    instance_count=8,
    instance_type='ml.p4d.24xlarge',
    framework_version='1.13.1',
    py_version='py3',
    distribution={
        "smdistributed": {"modelparallel": smp_options},
        "mpi": mpi_options
    },
    base_job_name="sharded-data-parallel-demo-with-tuning",
)

smd_mp_estimator.fit('s3://my_bucket/my_training_data/')
```

Pelatihan presisi campuran dengan paralelisme data sharded

Untuk lebih menghemat memori GPU dengan angka floating point setengah presisi dan paralelisme data sharded, Anda dapat mengaktifkan format floating point 16-bit (FP16) atau Brain floating point format (BF16) dengan menambahkan satu parameter tambahan ke konfigurasi pelatihan terdistribusi.

> **Note**
>
> Pelatihan presisi campuran dengan paralelisme data sharded tersedia di perpustakaan paralelisme SageMaker model v1.11.0 dan yang lebih baru.

Untuk Pelatihan FP16 dengan Paralelisme Data Sharded

Untuk menjalankan pelatihan FP16 dengan paralelisme data sharded, tambahkan ke `"fp16": True"` kamus konfigurasi. `smp_options` Dalam skrip pelatihan Anda, Anda dapat memilih antara opsi penskalaan kerugian statis dan dinamis melalui `smp.DistributedOptimizer` modul. Untuk informasi selengkapnya, lihat the section called “Pelatihan FP16 dengan Paralelisme Model”.

```
smp_options = {
    "enabled": True,
    "parameters": {
        "ddp": True,
        "sharded_data_parallel_degree": 2,
        "fp16": True
    }
}
```

Untuk Pelatihan BF16 dengan Paralelisme Data Sharded

Fitur paralelisme data sharded SageMaker mendukung pelatihan dalam tipe data BF16. Tipe data BF16 menggunakan 8 bit untuk mewakili eksponen nomor floating point, sedangkan tipe data FP16 menggunakan 5 bit. Mempertahankan 8 bit untuk eksponen memungkinkan untuk mempertahankan representasi yang sama dari eksponen nomor floating point presisi tunggal 32-bit (FP32). Hal ini membuat konversi antara FP32 dan BF16 lebih sederhana dan secara signifikan kurang rentan menyebabkan masalah overflow dan underflow yang sering muncul dalam pelatihan FP16, terutama saat melatih model yang lebih besar. Sementara kedua tipe data menggunakan total 16 bit, rentang representasi yang meningkat untuk eksponen dalam format BF16 ini mengorbankan presisi yang berkurang. Untuk melatih model besar, presisi yang berkurang ini sering dianggap sebagai trade-off yang dapat diterima untuk jangkauan dan stabilitas pelatihan.

> **Note**
>
> Saat ini, pelatihan BF16 hanya berfungsi ketika paralelisme data sharded diaktifkan.

Untuk menjalankan pelatihan BF16 dengan paralelisme data sharded, tambahkan ke `"bf16": True` kamus konfigurasi. `smp_options`

```
smp_options = {
    "enabled": True,
    "parameters": {
        "ddp": True,
        "sharded_data_parallel_degree": 2,
        "bf16": True
    }
}
```

Paralelisme data sharded dengan paralelisme tensor

Jika Anda menggunakan paralelisme data sharded dan juga perlu mengurangi ukuran batch global, pertimbangkan untuk menggunakan paralelisme tensor dengan paralelisme data sharded. Saat melatih model besar dengan paralelisme data sharded pada cluster komputasi yang sangat besar (biasanya 128 node atau lebih), bahkan ukuran batch kecil per GPU menghasilkan ukuran batch global yang sangat besar. Ini mungkin menyebabkan masalah konvergensi atau masalah kinerja komputasi yang rendah. Mengurangi ukuran batch per GPU terkadang tidak dimungkinkan dengan paralelisme data sharded saja ketika satu batch sudah besar dan tidak dapat dikurangi lebih lanjut. Dalam kasus seperti itu, menggunakan paralelisme data sharded dalam kombinasi dengan paralelisme tensor membantu mengurangi ukuran batch global.

Memilih paralel data sharded dan derajat paralel tensor yang optimal tergantung pada skala model, jenis instance, dan ukuran batch global yang masuk akal untuk model untuk bertemu. Kami menyarankan Anda memulai dari derajat paralel tensor rendah agar sesuai dengan ukuran batch global ke dalam cluster komputasi untuk mengatasi out-of-memory kesalahan CUDA dan mencapai kinerja terbaik. Lihat dua contoh kasus berikut untuk mempelajari bagaimana kombinasi paralelisme tensor dan paralelisme data sharded membantu Anda menyesuaikan ukuran batch global dengan mengelompokkan GPU untuk paralelisme model, menghasilkan jumlah replika model yang lebih rendah dan ukuran batch global yang lebih kecil.

> **Note**
>
> Fitur ini tersedia dari pustaka paralelisme SageMaker model v1.15, dan mendukung v1.13.1. PyTorch

> **Note**
>
> Fitur ini tersedia untuk model yang didukung oleh fungsionalitas paralelisme tensor perpustakaan. Untuk menemukan daftar model yang didukung, lihat Support for Hugging Face Transformer Models. Perhatikan juga bahwa Anda harus `tensor_parallelism=True` meneruskan `smp.model_creation` argumen sambil memodifikasi skrip pelatihan Anda. Untuk mempelajari lebih lanjut, lihat skrip pelatihan `train_gpt_simple.py`di GitHub repositori SageMaker Contoh.

Contoh 1

Asumsikan bahwa kita ingin melatih model di atas sekelompok 1536 GPU (192 node dengan masing-masing 8 GPU), mengatur tingkat paralelisme data sharded menjadi 32 (`sharded_data_parallel_degree=32`) dan ukuran batch per GPU menjadi 1, di mana setiap batch memiliki panjang urutan 4096 token. Dalam hal ini, ada 1536 replika model, ukuran batch global menjadi 1536, dan setiap batch global berisi sekitar 6 juta token.

```
(1536 GPUs) * (1 batch per GPU) = (1536 global batches)
(1536 batches) * (4096 tokens per batch) = (6,291,456 tokens)
```

Menambahkan paralelisme tensor ke dalamnya dapat menurunkan ukuran batch global. Salah satu contoh konfigurasi dapat mengatur derajat paralel tensor ke 8 dan ukuran batch per GPU menjadi 4. Ini membentuk 192 kelompok paralel tensor atau 192 replika model, di mana setiap replika model didistribusikan di 8 GPU. Ukuran batch 4 adalah jumlah data pelatihan per iterasi dan per kelompok paralel tensor; yaitu, setiap replika model mengkonsumsi 4 batch per iterasi. Dalam hal ini, ukuran batch global menjadi 768, dan setiap batch global berisi sekitar 3 juta token. Oleh karena itu, ukuran batch global berkurang setengahnya dibandingkan dengan kasus sebelumnya dengan paralelisme data sharded saja.

```
(1536 GPUs) / (8 tensor parallel degree) = (192 tensor parallelism groups)
(192 tensor parallelism groups) * (4 batches per tensor parallelism group) = (768
 global batches)
(768 batches) * (4096 tokens per batch) = (3,145,728 tokens)
```

Contoh 2

Ketika paralelisme data sharded dan paralelisme tensor diaktifkan, perpustakaan pertama-tama menerapkan paralelisme tensor dan memecah model di seluruh dimensi ini. Untuk setiap peringkat paralel tensor, paralelisme data diterapkan sesuai. `sharded_data_parallel_degree`

Misalnya, asumsikan bahwa kita ingin mengatur 32 GPU dengan derajat paralel tensor 4 (membentuk kelompok 4 GPU), derajat paralel data sharded 4, berakhir dengan derajat replikasi 2. Penugasan tersebut membuat delapan grup GPU berdasarkan derajat paralel tensor sebagai berikut`(0,1,2,3)`:`(4,5,6,7)`,,,,`(8,9,10,11)`,`(12,13,14,15)`, `(16,17,18,19)(20,21,22,23)`,`(24,25,26,27)`. `(28,29,30,31)` Artinya, empat GPU membentuk satu kelompok paralel tensor. Dalam hal ini, grup paralel data yang dikurangi untuk GPU peringkat 0 dari kelompok paralel tensor adalah. `(0,4,8,12,16,20,24,28)` Kelompok paralel data tereduksi dipecah berdasarkan derajat paralel data sharded 4, menghasilkan dua kelompok replikasi untuk paralelisme data. GPU `(0,4,8,12)` membentuk satu grup sharding, yang secara kolektif menyimpan salinan lengkap semua parameter untuk peringkat paralel tensor ke-0, dan `(16,20,24,28)` GPU membentuk kelompok lain seperti itu. Peringkat paralel tensor lainnya juga memiliki kelompok sharding dan replikasi yang serupa.

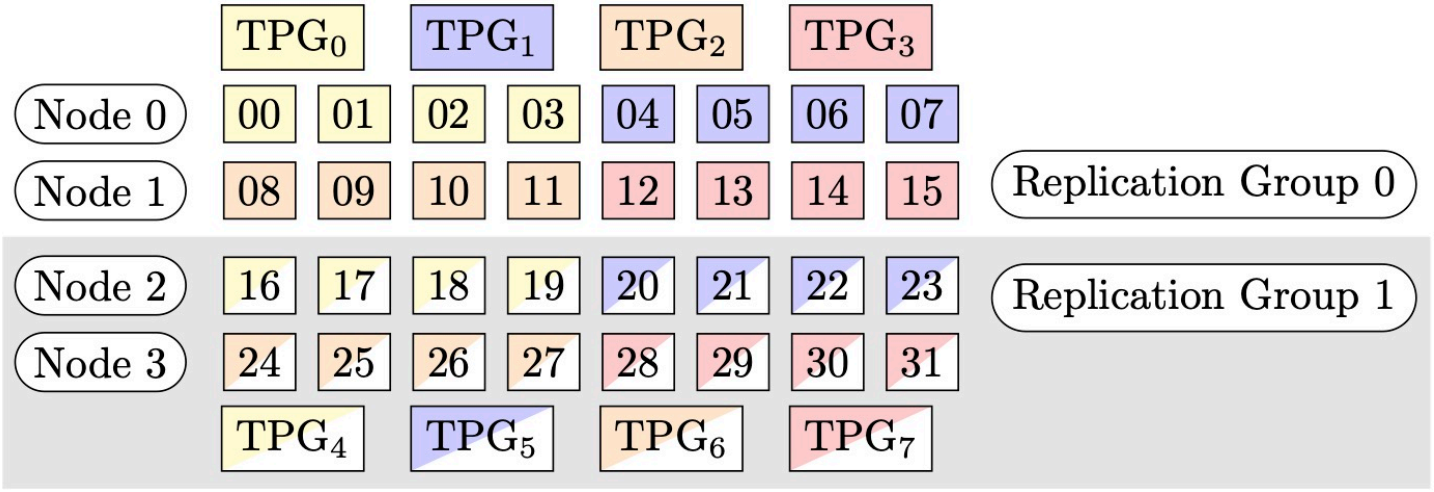


Figure 1: Tensor parallelism groups for (nodes, sharded data parallel degree, tensor parallel degree) = (4, 4, 4), where each rectangle represents a GPU with indices from 0 to 31. The GPUs form tensor parallelism groups from TPG_0 to TPG_7. Replication groups are ({TPG_0, TPG_4}, {TPG_1, TPG_5}, {TPG_2, TPG_6} and {TPG_3, TPG_7}); each replication group pair shares the same color but filled differently.

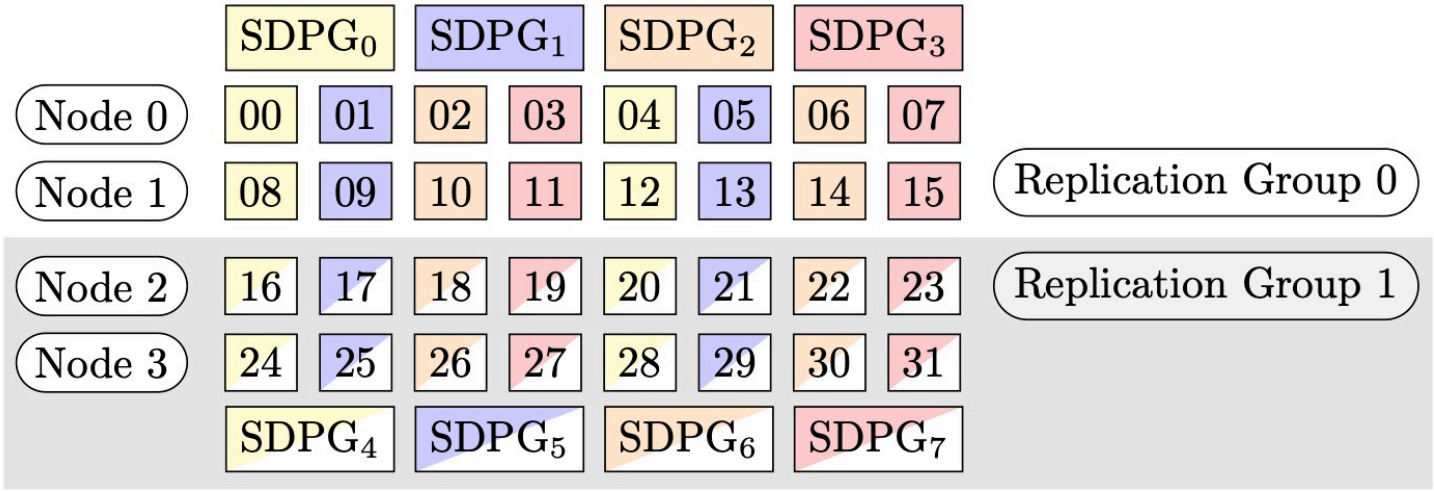


Figure 2: Sharded data parallelism groups for (nodes, sharded data parallel degree, tensor parallel degree) = (4, 4, 4), where each rectangle represents a GPU with indices from 0 to 31. The GPUs form sharded data parallelism groups from $SDPG_0$ to $SDPG_7$. Replication groups are ({$SDPG_0$, $SDPG_4$}, {$SDPG_1$, $SDPG_5$}, {$SDPG_2$, $SDPG_6$} and {$SDPG_3$, $SDPG_7$}); each replication group pair shares the same color but filled differently.

Cara mengaktifkan paralelisme data sharded dengan paralelisme tensor

Untuk menggunakan paralelisme data sharded dengan paralelisme tensor, Anda perlu mengatur keduanya `sharded_data_parallel_degree` dan `tensor_parallel_degree` dalam konfigurasi `distribution` saat membuat objek dari kelas estimator. SageMaker PyTorch

Anda juga perlu mengaktifkan`prescaled_batch`. Ini berarti bahwa, alih-alih setiap GPU membaca kumpulan datanya sendiri, setiap grup paralel tensor secara kolektif membaca kumpulan gabungan dari ukuran batch yang dipilih. Secara efektif, alih-alih membagi dataset menjadi bagian-bagian yang sama dengan jumlah GPU (atau ukuran paralel data,`smp.dp_size()`), ia membagi menjadi bagian-bagian yang sama dengan jumlah GPU dibagi dengan `tensor_parallel_degree` (juga disebut ukuran paralel data yang dikurangi,). `smp.rdp_size()` Untuk detail selengkapnya tentang batch pra-skala, lihat Batch Prescaled dalam dokumentasi SageMaker Python SDK. Lihat juga contoh skrip pelatihan *`train_gpt_simple.py`*untuk GPT-2 di repositori Contoh. SageMaker GitHub

Cuplikan kode berikut menunjukkan contoh pembuatan objek PyTorch estimator berdasarkan skenario yang disebutkan di atas. the section called “Contoh 2”

```
mpi_options = "-verbose --mca orte_base_help_aggregate 0 "
smp_parameters = {
    "ddp": True,
    "fp16": True,
```

```
    "prescaled_batch": True,
    "sharded_data_parallel_degree": 4,
    "tensor_parallel_degree": 4
}

pytorch_estimator = PyTorch(
    entry_point="your_training_script.py",
    role=role,
    instance_type="ml.p4d.24xlarge",
    volume_size=200,
    instance_count=4,
    sagemaker_session=sagemaker_session,
    py_version="py3",
    framework_version="1.13.1",
    distribution={
        "smdistributed": {
            "modelparallel": {
                "enabled": True,
                "parameters": smp_parameters,
            }
        },
        "mpi": {
            "enabled": True,
            "processes_per_host": 8,
            "custom_mpi_options": mpi_options,
        },
    },
    source_dir="source_directory_of_your_code",
    output_path=s3_output_location
)
```

Kiat dan pertimbangan untuk menggunakan paralelisme data sharded

Pertimbangkan hal berikut saat menggunakan paralelisme data sharded pustaka paralelisme SageMaker model.

- Paralelisme data sharded kompatibel dengan pelatihan FP16. Untuk menjalankan pelatihan FP16, lihat bagian. the section called “Pelatihan FP16 dengan Paralelisme Model”
- Paralelisme data sharded kompatibel dengan paralelisme tensor. Item berikut adalah apa yang mungkin perlu Anda pertimbangkan untuk menggunakan paralelisme data sharded dengan paralelisme tensor.

  - Saat menggunakan paralelisme data sharded dengan paralelisme tensor, lapisan penyematan juga didistribusikan secara otomatis di seluruh grup paralel tensor. Dengan kata lain, `distribute_embedding` parameter secara otomatis diatur ke`True`. Untuk informasi lebih lanjut tentang paralelisme tensor, lihat. the section called “Paralelisme Tensor”
  - Perhatikan bahwa paralelisme data sharded dengan paralelisme tensor saat ini menggunakan kolektif NCCL sebagai backend dari strategi pelatihan terdistribusi.

  Untuk mempelajari lebih lanjut, lihat the section called “Paralelisme data sharded dengan paralelisme tensor” bagian.

- Paralelisme data sharded saat ini tidak kompatibel dengan paralelisme pipa atau sharding status pengoptimal. Untuk mengaktifkan paralelisme data sharded, matikan sharding status pengoptimal dan atur derajat paralel pipeline ke 1.
- Fitur checkpointing aktivasi dan pembongkaran aktivasi kompatibel dengan paralelisme data sharded.
- Untuk menggunakan paralelisme data sharded dengan akumulasi gradien, setel `backward_passes_per_step` argumen ke jumlah langkah akumulasi sambil membungkus model Anda dengan modul. `smdistributed.modelparallel.torch.DistributedModel` Ini memastikan bahwa `AllReduce` operasi gradien di seluruh grup replikasi model (grup sharding) berlangsung pada batas akumulasi gradien.
- Anda dapat memeriksa model Anda yang dilatih dengan paralelisme data sharded menggunakan API checkpointing perpustakaan, dan. `smp.save_checkpoint` `smp.resume_from_checkpoint` Untuk informasi selengkapnya, lihat the section called “Checkpointing PyTorch model terdistribusi (untuk pustaka paralelisme SageMaker model v1.10.0 dan yang lebih baru)”.
- Perilaku parameter `delayed_parameter_initialization`konfigurasi berubah di bawah paralelisme data sharded. Ketika kedua fitur ini diaktifkan secara bersamaan, parameter segera diinisialisasi pada pembuatan model secara sharded alih-alih menunda inisialisasi parameter, sehingga setiap peringkat menginisialisasi dan menyimpan pecahan parameternya sendiri.
- Saat paralelisme data sharded diaktifkan, pustaka melakukan kliping gradien secara internal saat panggilan berjalan. `optimizer.step()` Anda tidak perlu menggunakan API utilitas untuk kliping gradien, seperti. `torch.nn.utils.clip_grad_norm_()` Untuk menyesuaikan nilai ambang batas untuk kliping gradien, Anda dapat mengaturnya melalui `sdp_gradient_clipping` parameter untuk konfigurasi parameter distribusi saat Anda membuat SageMaker PyTorch estimator, seperti yang ditunjukkan di bagian. the section called “Cara menerapkan paralelisme data sharded ke pekerjaan pelatihan Anda”

Pipelining Model

Salah satu fitur inti dari SageMaker perpustakaan paralelisme model adalah paralelisme pipa, yang menentukan urutan perhitungan dibuat dan data diproses di seluruh perangkat selama pelatihan model. Pipelining adalah teknik untuk mencapai paralelisasi sejati dalam paralelisme model, dengan meminta GPU menghitung secara bersamaan pada sampel data yang berbeda, dan untuk mengatasi kehilangan kinerja karena komputasi berurutan. Saat Anda menggunakan paralelisme pipeline, pekerjaan pelatihan dijalankan secara pipelined melalui microbatch untuk memaksimalkan penggunaan GPU.

> **Note**
>
> Paralelisme pipa, juga disebut partisi model, tersedia untuk keduanya dan. PyTorch TensorFlow Untuk versi kerangka kerja yang didukung, lihatthe section called “Kerangka Kerja yang Didukung dan Wilayah AWS”.

Jadwal Eksekusi Pipeline

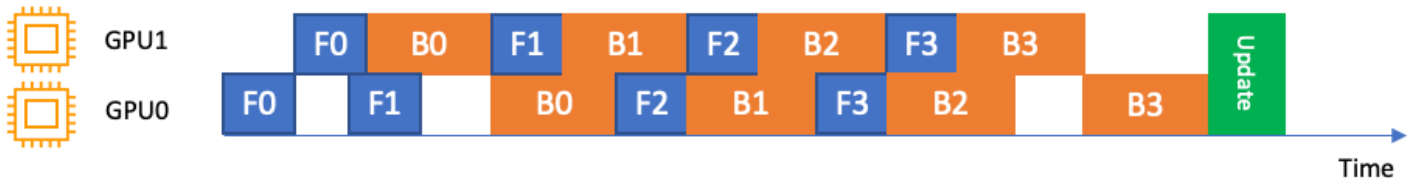
Pipelining didasarkan pada pemisahan mini-batch menjadi microbatch, yang dimasukkan ke dalam pipeline pelatihan one-by-one dan mengikuti jadwal eksekusi yang ditentukan oleh runtime perpustakaan. Microbatch adalah subset yang lebih kecil dari mini-batch pelatihan yang diberikan. Jadwal pipa menentukan microbatch mana yang dijalankan oleh perangkat mana untuk setiap slot waktu.

Misalnya, tergantung pada jadwal pipeline dan partisi model, GPU `i` mungkin melakukan komputasi (maju atau mundur) pada microbatch b sementara GPU `i+1` melakukan komputasi pada microbatchb `+1`, sehingga menjaga kedua GPU tetap aktif pada saat yang bersamaan. Selama pass maju atau mundur tunggal, alur eksekusi untuk satu microbatch mungkin mengunjungi perangkat yang sama beberapa kali, tergantung pada keputusan partisi. Misalnya, operasi yang ada di awal model dapat ditempatkan pada perangkat yang sama dengan operasi di akhir model, sementara operasi di antaranya berada pada perangkat yang berbeda, yang berarti perangkat ini dikunjungi dua kali.

Pustaka menawarkan dua jadwal pipeline yang berbeda, sederhana dan interleaved, yang dapat dikonfigurasi menggunakan `pipeline` parameter di Python SageMaker SDK. Dalam kebanyakan kasus, pipa interleaved dapat mencapai kinerja yang lebih baik dengan memanfaatkan GPU secara lebih efisien.

Pipa Interleaved

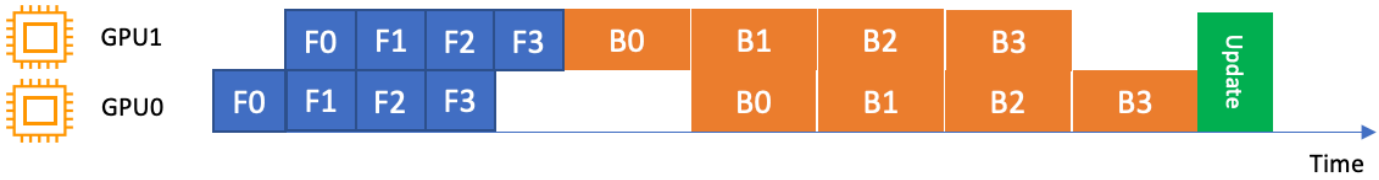
Dalam pipa yang disisipkan, eksekusi mundur dari microbatch diprioritaskan bila memungkinkan. Hal ini memungkinkan pelepasan lebih cepat dari memori yang digunakan untuk aktivasi, menggunakan memori lebih efisien. Ini juga memungkinkan untuk menskalakan jumlah microbatch yang lebih tinggi, mengurangi waktu idle GPU. Pada kondisi tunak, setiap perangkat bergantian antara berjalan maju dan mundur. Ini berarti bahwa lintasan mundur dari satu microbatch dapat berjalan sebelum pass maju dari microbatch lain selesai.



Gambar sebelumnya menggambarkan contoh jadwal eksekusi untuk pipeline interleaved lebih dari 2 GPU. Pada gambar, F0 mewakili pass maju untuk microbatch 0, dan B1 mewakili pass mundur untuk microbatch 1. Pembaruan mewakili pembaruan pengoptimal parameter. GPU0 selalu memprioritaskan pass mundur bila memungkinkan (misalnya, mengeksekusi B0 sebelum F2), yang memungkinkan untuk membersihkan memori yang digunakan untuk aktivasi sebelumnya.

Alur Sederhana

Pipeline sederhana, sebaliknya, selesai menjalankan pass maju untuk setiap microbatch sebelum memulai lintasan mundur. Ini berarti bahwa itu hanya menyalurkan tahapan pass maju dan mundur di dalam diri mereka sendiri. Gambar berikut mengilustrasikan contoh cara kerjanya, lebih dari 2 GPU.

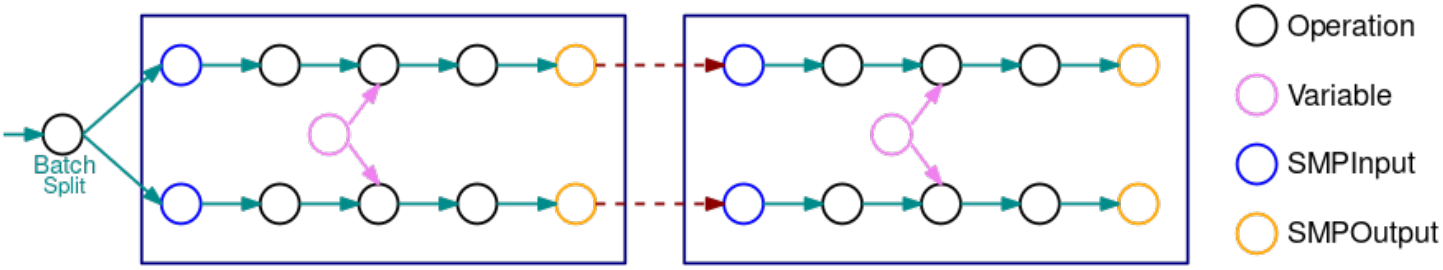


Eksekusi Pipelining dalam Kerangka Kerja Tertentu

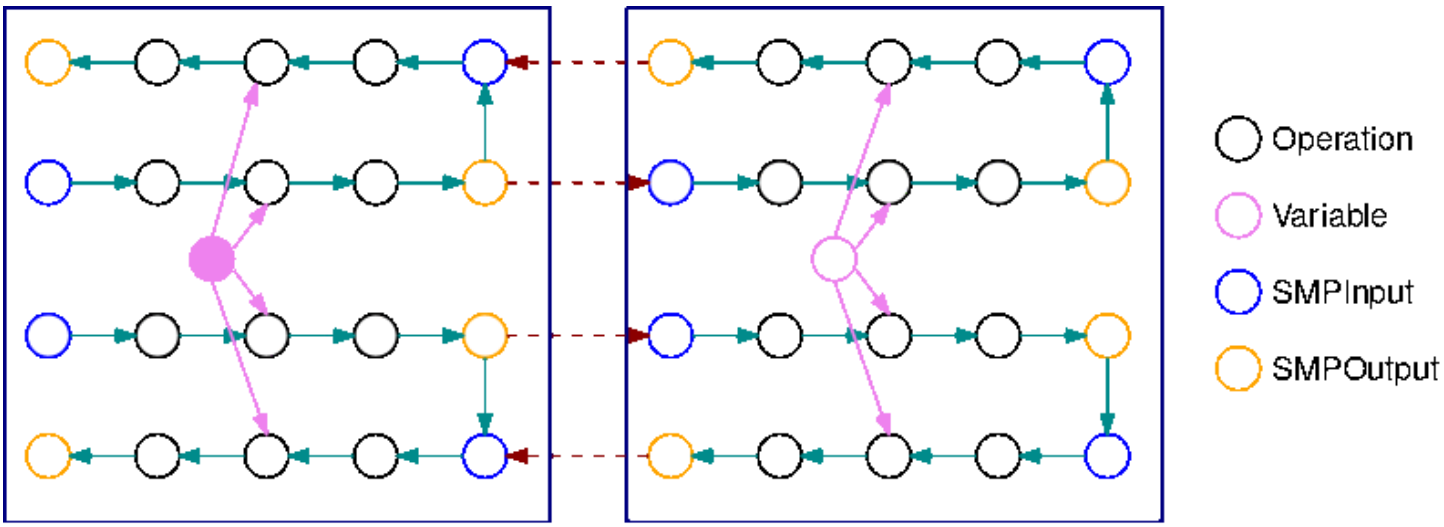
Gunakan bagian berikut untuk mempelajari tentang keputusan penjadwalan pipeline khusus kerangka kerja yang dibuat oleh pustaka SageMaker paralelisme model untuk dan. TensorFlow PyTorch

Eksekusi Pipeline dengan TensorFlow

Gambar berikut adalah contoh TensorFlow grafik yang dipartisi oleh pustaka paralelisme model, menggunakan pemisahan model otomatis. Ketika grafik dibagi, setiap subgraf yang dihasilkan direplikasi B kali (kecuali untuk variabel), di mana B adalah jumlah microbatch. Pada gambar ini, setiap subgraf direplikasi 2 kali (B=2). `SMPInput`Operasi dimasukkan pada setiap input subgraf, dan `SMPOutput` operasi dimasukkan pada setiap output. Operasi ini berkomunikasi dengan backend perpustakaan untuk mentransfer tensor ke dan dari satu sama lain.



Gambar berikut adalah contoh dari 2 subgraf yang dibagi dengan B = 2 dengan operasi gradien ditambahkan. Gradien `SMPInput` op adalah `SMPOutput` op, dan sebaliknya. Ini memungkinkan gradien mengalir mundur selama propagasi balik.



GIF ini menunjukkan contoh jadwal eksekusi pipeline interleaved dengan B = 2 microbatch dan 2 subgraf. Setiap perangkat secara berurutan mengeksekusi salah satu replika subgraf untuk meningkatkan pemanfaatan GPU. Saat B tumbuh lebih besar, fraksi slot waktu idle menjadi nol. Kapan pun waktunya untuk melakukan perhitungan (maju atau mundur) pada replika subgraf

tertentu, lapisan pipa memberi sinyal ke operasi biru `SMPInput` yang sesuai untuk mulai mengeksekusi.

Setelah gradien dari semua microbatch dalam satu mini-batch dihitung, perpustakaan menggabungkan gradien di seluruh microbatch, yang kemudian dapat diterapkan ke parameter.

Eksekusi Pipeline dengan PyTorch

Secara konseptual, pipelining mengikuti ide serupa di. PyTorch Namun, karena PyTorch tidak melibatkan grafik statis sehingga PyTorch fitur perpustakaan paralelisme model menggunakan paradigma pipelining yang lebih dinamis.

Seperti dalam TensorFlow, setiap batch dibagi menjadi beberapa microbatch, yang dijalankan satu per satu pada setiap perangkat. Namun, jadwal eksekusi ditangani melalui server eksekusi yang diluncurkan pada setiap perangkat. Setiap kali output submodul yang ditempatkan pada perangkat lain diperlukan pada perangkat saat ini, permintaan eksekusi dikirim ke server eksekusi perangkat jarak jauh bersama dengan tensor input ke submodul. Server kemudian mengeksekusi modul ini dengan input yang diberikan dan mengembalikan respons ke perangkat saat ini.

Karena perangkat saat ini menganggur selama eksekusi submodul jarak jauh, eksekusi lokal untuk microbatch saat ini berhenti, dan runtime perpustakaan mengalihkan eksekusi ke microbatch lain yang dapat dikerjakan secara aktif oleh perangkat saat ini. Prioritas microbatch ditentukan oleh jadwal pipa yang dipilih. Untuk jadwal pipa yang disisipkan, microbatch yang berada dalam tahap mundur komputasi diprioritaskan bila memungkinkan.

Paralelisme Tensor

Paralelisme tensor adalah jenis paralelisme model di mana bobot model tertentu, gradien, dan status pengoptimal dibagi di seluruh perangkat. Berbeda dengan paralelisme pipa, yang menjaga bobot individu tetap utuh tetapi mempartisi set bobot, paralelisme tensor membagi bobot individu. Ini biasanya melibatkan komputasi terdistribusi dari operasi tertentu, modul, atau lapisan model.

Paralelisme tensor diperlukan dalam kasus di mana satu parameter menghabiskan sebagian besar memori GPU (seperti tabel penyematan besar dengan ukuran kosakata besar atau lapisan softmax besar dengan sejumlah besar kelas). Dalam hal ini, memperlakukan tensor atau operasi besar ini sebagai unit atom tidak efisien dan menghambat keseimbangan beban memori.

Paralelisme tensor juga berguna untuk model yang sangat besar di mana pipelining murni tidak cukup. Misalnya, dengan model skala GPT-3 yang memerlukan partisi lebih dari puluhan instance,

pipelining mikrobatch murni tidak efisien karena kedalaman pipa menjadi terlalu tinggi dan overhead menjadi sangat besar.

> **Note**
>
> Paralelisme tensor tersedia untuk perpustakaan paralelisme SageMaker model PyTorch v1.6.0 dan yang lebih baru.

Topik

Bagaimana Paralelisme Tensor Bekerja

Paralelisme tensor terjadi pada tingkat`nn.Modules`; itu mempartisi modul tertentu dalam model di seluruh peringkat paralel tensor. Ini merupakan tambahan dari partisi yang ada dari kumpulan modul yang digunakan dalam paralelisme pipa.

Ketika modul dipartisi melalui paralelisme tensor, propagasi maju dan mundur didistribusikan. Pustaka menangani komunikasi yang diperlukan di seluruh perangkat untuk mengimplementasikan eksekusi terdistribusi modul-modul ini. Modul dipartisi di beberapa peringkat paralel data. Berlawanan dengan distribusi beban kerja tradisional, setiap peringkat paralel data tidak memiliki replika model lengkap saat paralelisme tensor perpustakaan digunakan. Sebaliknya, setiap peringkat paralel data mungkin hanya memiliki partisi dari modul terdistribusi, di samping keseluruhan modul yang tidak didistribusikan.

Contoh: Pertimbangkan paralelisme tensor di seluruh peringkat paralel data, di mana tingkat paralelisme data adalah 4 dan derajat paralelisme tensor adalah 2. Asumsikan bahwa Anda memiliki grup paralel data yang memegang pohon modul berikut, setelah mempartisi kumpulan modul.

```
A
### B
|   ### E
```

```
|   ### F
### C
### D
    ### G
    ### H
```

Asumsikan bahwa paralelisme tensor didukung untuk modul B, G, dan H. Salah satu kemungkinan hasil partisi paralel tensor model ini adalah:

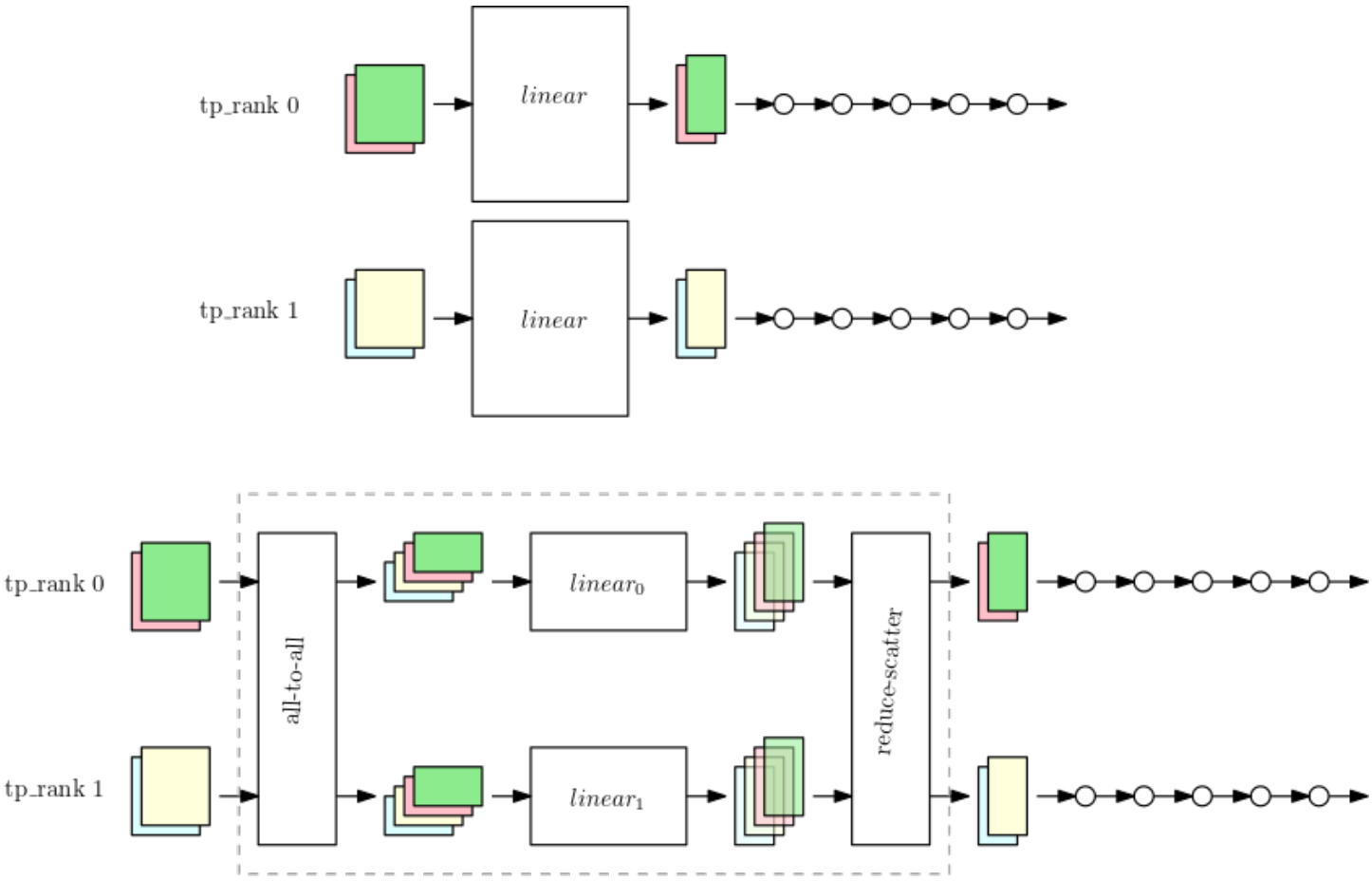
```
dp_rank 0 (tensor parallel rank 0): A, B:0, C, D, G:0, H
dp_rank 1 (tensor parallel rank 1): A, B:1, C, D, G:1, H
dp_rank 2 (tensor parallel rank 0): A, B:0, C, D, G:0, H
dp_rank 3 (tensor parallel rank 1): A, B:1, C, D, G:1, H
```

Setiap baris mewakili kumpulan modul yang disimpan di dalamnyadp_rank, dan notasi X:y mewakili fraksi y ke-modul. X Perhatikan hal berikut:

1. Partisi terjadi di seluruh himpunan bagian dari peringkat paralel data, yang kita sebutTP_GROUP, bukan keseluruhanDP_GROUP, sehingga partisi model yang tepat direplikasi di dp_rank 0 dan dp_rank 2, dan juga di dp_rank 1 dan 3. dp_rank
2. Modul E dan F tidak lagi menjadi bagian dari model, karena modul induknya B dipartisi, dan setiap eksekusi yang biasanya merupakan bagian dari E dan F berlangsung di dalam modul (dipartisi). B
3. Meskipun H didukung untuk paralelisme tensor, dalam contoh ini tidak dipartisi, yang menyoroti bahwa apakah akan mempartisi modul bergantung pada input pengguna. Fakta bahwa modul didukung untuk paralelisme tensor tidak selalu berarti itu dipartisi.

Bagaimana perpustakaan menyesuaikan paralelisme tensor ke modul PyTorch **nn.Linear**

Ketika paralelisme tensor dilakukan melalui peringkat paralel data, subset dari parameter, gradien, dan status pengoptimal dipartisi di seluruh perangkat paralel tensor untuk modul yang dipartisi. Untuk modul lainnya, perangkat paralel tensor beroperasi secara paralel data reguler. Untuk menjalankan modul yang dipartisi, perangkat pertama-tama mengumpulkan bagian yang diperlukan dari semua sampel data di seluruh perangkat rekan dalam grup paralelisme tensor yang sama. Perangkat kemudian menjalankan fraksi lokal modul pada semua sampel data ini, diikuti oleh putaran sinkronisasi lain yang menggabungkan bagian-bagian output untuk setiap sampel data dan mengembalikan sampel data gabungan ke GPU dari mana sampel data pertama berasal. Gambar berikut menunjukkan contoh proses ini melalui modul yang dipartisinn.Linear.



Gambar pertama menunjukkan model kecil dengan `nn.Linear` modul besar dengan paralelisme data di atas dua peringkat paralelisme tensor. `nn.Linear`Modul direplikasi ke dalam dua peringkat paralel.

Gambar kedua menunjukkan paralelisme tensor diterapkan pada model yang lebih besar saat membelah modul. `nn.Linear` Masing-masing `tp_rank` memegang setengah modul linier, dan keseluruhan operasi lainnya. Sementara modul linier berjalan, masing-masing `tp_rank` mengumpulkan setengah yang relevan dari semua sampel data dan meneruskannya melalui setengah dari `nn.Linear` modul mereka. Hasilnya perlu direduksi tersebar (dengan penjumlahan sebagai operasi reduksi) sehingga setiap peringkat memiliki keluaran linier akhir untuk sampel datanya sendiri. Sisa model berjalan dengan cara paralel data yang khas.

Jalankan Job Pelatihan Paralel Model SageMaker Terdistribusi dengan Paralelisme Tensor

Di bagian ini, Anda belajar:

- Cara mengonfigurasi SageMaker PyTorch estimator dan opsi paralelisme SageMaker model untuk menggunakan paralelisme tensor.

- Cara mengadaptasi skrip pelatihan Anda menggunakan `smdistributed.modelparallel` modul yang diperluas untuk paralelisme tensor.

Untuk mempelajari lebih lanjut tentang `smdistributed.modelparallel` modul, lihat API paralel SageMaker model dalam dokumentasi SageMaker Python SDK.

Topik

Paralelisme tensor saja

Berikut ini adalah contoh opsi pelatihan terdistribusi untuk mengaktifkan paralelisme tensor saja, tanpa paralelisme pipa. Konfigurasikan `mpi_options` dan `smp_options` kamus untuk menentukan opsi pelatihan terdistribusi ke estimator. SageMaker `PyTorch`

> **Note**
>
> Fitur hemat memori yang diperluas tersedia melalui Deep Learning Containers for PyTorch, yang mengimplementasikan pustaka paralelisme SageMaker model v1.6.0 atau yang lebih baru.

Konfigurasikan SageMaker PyTorch estimator

```
mpi_options = {
    "enabled" : True,
    "processes_per_host" : 8,               # 8 processes
    "custom_mpi_options" : "--mca btl_vader_single_copy_mechanism none "
}

smp_options = {
    "enabled":True,
    "parameters": {
        "pipeline_parallel_degree": 1,    # alias for "partitions"
        "placement_strategy": "cluster",
        "tensor_parallel_degree": 4,      # tp over 4 devices
        "ddp": True
    }
```

```
}

smp_estimator = PyTorch(
    entry_point='your_training_script.py', # Specify
    role=role,
    instance_type='ml.p3.16xlarge',
    sagemaker_session=sagemaker_session,
    framework_version='1.13.1',
    py_version='py36',
    instance_count=1,
    distribution={
        "smdistributed": {"modelparallel": smp_options},
        "mpi": mpi_options
    },
    base_job_name="SMD-MP-demo",
)

smp_estimator.fit('s3://my_bucket/my_training_data/')
```

> **Tip**
>
> Untuk menemukan daftar lengkap parameter`distribution`, lihat Parameter Konfigurasi untuk Paralelisme Model dalam dokumentasi SageMaker Python SDK.

Sesuaikan skrip PyTorch pelatihan Anda

Contoh skrip pelatihan berikut menunjukkan bagaimana mengadaptasi perpustakaan paralelisme SageMaker model ke skrip pelatihan. Dalam contoh ini, diasumsikan bahwa skrip diberi nama`your_training_script.py`.

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchnet.dataset import SplitDataset
from torchvision import datasets

import smdistributed.modelparallel.torch as smp

class Net(nn.Module):
    def __init__(self):
```

```
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 3, 1)
        self.conv2 = nn.Conv2d(32, 64, 3, 1)
        self.fc1 = nn.Linear(9216, 128)
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = self.conv1(x)
        x = F.relu(x)
        x = self.conv2(x)
        x = F.relu(x)
        x = F.max_pool2d(x, 2)
        x = torch.flatten(x, 1)
        x = self.fc1(x)
        x = F.relu(x)
        x = self.fc2(x)
        return F.log_softmax(x, 1)

def train(model, device, train_loader, optimizer):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        # smdistributed: Move input tensors to the GPU ID used by
        # the current process, based on the set_device call.
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        output = model(data)
        loss = F.nll_loss(output, target, reduction="mean")
        loss.backward()
        optimizer.step()

# smdistributed: Initialize the backend
smp.init()

# smdistributed: Set the device to the GPU ID used by the current process.
# Input tensors should be transferred to this device.
torch.cuda.set_device(smp.local_rank())
device = torch.device("cuda")

# smdistributed: Download only on a single process per instance.
# When this is not present, the file is corrupted by multiple processes trying
# to download and extract at the same time
if smp.local_rank() == 0:
    dataset = datasets.MNIST("../data", train=True, download=False)
smp.barrier()
```

```
# smdistributed: Shard the dataset based on data parallel ranks
if smp.dp_size() > 1:
    partitions_dict = {f"{i}": 1 / smp.dp_size() for i in range(smp.dp_size())}
    dataset = SplitDataset(dataset, partitions=partitions_dict)
    dataset.select(f"{smp.dp_rank()}")

train_loader = torch.utils.data.DataLoader(dataset, batch_size=64)

# smdistributed: Enable tensor parallelism for all supported modules in the model
# i.e., nn.Linear in this case. Alternatively, we can use
# smp.set_tensor_parallelism(model.fc1, True)
# to enable it only for model.fc1
with smp.tensor_parallelism():
    model = Net()

# smdistributed: Use the DistributedModel wrapper to distribute the
# modules for which tensor parallelism is enabled
model = smp.DistributedModel(model)

optimizer = optim.AdaDelta(model.parameters(), lr=4.0)
optimizer = smp.DistributedOptimizer(optimizer)

train(model, device, train_loader, optimizer)
```

Paralelisme tensor dikombinasikan dengan paralelisme pipa

Berikut ini adalah contoh opsi pelatihan terdistribusi yang memungkinkan paralelisme tensor dikombinasikan dengan paralelisme pipa. Siapkan `smp_options` parameter `mpi_options` dan untuk menentukan opsi paralel model dengan paralelisme tensor saat Anda mengonfigurasi estimator. SageMaker `PyTorch`

> **Note**
>
> Fitur hemat memori yang diperluas tersedia melalui Deep Learning Containers for PyTorch, yang mengimplementasikan pustaka paralelisme SageMaker model v1.6.0 atau yang lebih baru.

Konfigurasikan SageMaker PyTorch estimator

```
mpi_options = {
```

```
    "enabled" : True,
    "processes_per_host" : 8,               # 8 processes
    "custom_mpi_options" : "--mca btl_vader_single_copy_mechanism none "
}

smp_options = {
    "enabled":True,
    "parameters": {
    "microbatches": 4,
        "pipeline_parallel_degree": 2,    # alias for "partitions"
        "placement_strategy": "cluster",
        "tensor_parallel_degree": 2,      # tp over 2 devices
        "ddp": True
    }
}

smp_estimator = PyTorch(
    entry_point='your_training_script.py', # Specify
    role=role,
    instance_type='ml.p3.16xlarge',
    sagemaker_session=sagemaker_session,
    framework_version='1.13.1',
    py_version='py36',
    instance_count=1,
    distribution={
        "smdistributed": {"modelparallel": smp_options},
        "mpi": mpi_options
    },
    base_job_name="SMD-MP-demo",
)

smp_estimator.fit('s3://my_bucket/my_training_data/')
```

Sesuaikan skrip PyTorch pelatihan Anda

Contoh skrip pelatihan berikut menunjukkan bagaimana mengadaptasi perpustakaan paralelisme SageMaker model ke skrip pelatihan. Perhatikan bahwa skrip pelatihan sekarang menyertakan `smp.step` dekorator:

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
```

```
from torchnet.dataset import SplitDataset
from torchvision import datasets

import smdistributed.modelparallel.torch as smp

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 3, 1)
        self.conv2 = nn.Conv2d(32, 64, 3, 1)
        self.fc1 = nn.Linear(9216, 128)
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = self.conv1(x)
        x = F.relu(x)
        x = self.conv2(x)
        x = F.relu(x)
        x = F.max_pool2d(x, 2)
        x = torch.flatten(x, 1)
        x = self.fc1(x)
        x = F.relu(x)
        x = self.fc2(x)
        return F.log_softmax(x, 1)


# smdistributed: Define smp.step. Return any tensors needed outside.
@smp.step
def train_step(model, data, target):
    output = model(data)
    loss = F.nll_loss(output, target, reduction="mean")
    model.backward(loss)
    return output, loss

def train(model, device, train_loader, optimizer):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        # smdistributed: Move input tensors to the GPU ID used by
        # the current process, based on the set_device call.
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        # Return value, loss_mb is a StepOutput object
        _, loss_mb = train_step(model, data, target)
```

```
        # smdistributed: Average the loss across microbatches.
        loss = loss_mb.reduce_mean()

        optimizer.step()

# smdistributed: Initialize the backend
smp.init()

# smdistributed: Set the device to the GPU ID used by the current process.
# Input tensors should be transferred to this device.
torch.cuda.set_device(smp.local_rank())
device = torch.device("cuda")

# smdistributed: Download only on a single process per instance.
# When this is not present, the file is corrupted by multiple processes trying
# to download and extract at the same time
if smp.local_rank() == 0:
    dataset = datasets.MNIST("../data", train=True, download=False)
smp.barrier()

# smdistributed: Shard the dataset based on data parallel ranks
if smp.dp_size() > 1:
    partitions_dict = {f"{i}": 1 / smp.dp_size() for i in range(smp.dp_size())}
    dataset = SplitDataset(dataset, partitions=partitions_dict)
    dataset.select(f"{smp.dp_rank()}")

# smdistributed: Set drop_last=True to ensure that batch size is always divisible
# by the number of microbatches
train_loader = torch.utils.data.DataLoader(dataset, batch_size=64, drop_last=True)

model = Net()

# smdistributed: enable tensor parallelism only for model.fc1
smp.set_tensor_parallelism(model.fc1, True)

# smdistributed: Use the DistributedModel container to provide the model
# to be partitioned across different ranks. For the rest of the script,
# the returned DistributedModel object should be used in place of
# the model provided for DistributedModel class instantiation.
model = smp.DistributedModel(model)

optimizer = optim.AdaDelta(model.parameters(), lr=4.0)
optimizer = smp.DistributedOptimizer(optimizer)
```

```
train(model, device, train_loader, optimizer)
```

Support untuk Model Trafo Hugging Face

Paralelisme tensor perpustakaan paralelisme SageMaker model menawarkan out-of-the-box dukungan untuk model Hugging Face Transformer berikut:

- GPT-2, BERT, dan RoberTA (Tersedia di perpustakaan paralelisme SageMaker model v1.7.0 dan yang lebih baru)
- GPT-J (Tersedia di perpustakaan paralelisme SageMaker model v1.8.0 dan yang lebih baru)
- GPT-neo (Tersedia di perpustakaan paralelisme SageMaker model v1.10.0 dan yang lebih baru)

> **Note**
>
> Untuk model Transformers lainnya, Anda perlu menggunakan smdistributed.modelparallel.torch.tp_register_with_module () API untuk menerapkan paralelisme tensor.

> **Note**
>
> Untuk menggunakan paralelisme tensor untuk melatih model Hugging Face Transformer, pastikan Anda menggunakan Hugging Face Deep Learning Containers untuk yang SageMaker memiliki pustaka paralelisme model PyTorch v1.7.0 dan yang lebih baru. Untuk informasi selengkapnya, lihat catatan rilis perpustakaan paralelisme SageMaker model.

Model yang Didukung Di Luar Kotak

Untuk model transformator Hugging Face yang didukung oleh pustaka di luar kotak, Anda tidak perlu mengimplementasikan kait secara manual untuk menerjemahkan API `smdistributed` Transformer ke lapisan transformator. Anda dapat mengaktifkan paralelisme tensor dengan menggunakan manajer konteks smdistributed.modelparallel.torch.tensor_parallelism () dan membungkus model dengan smdistributed.modelparallel.torch. DistributedModel(). Anda tidak perlu mendaftarkan kait secara manual untuk paralelisme tensor menggunakan API. `smp.tp_register`

Fungsi `state_dict` terjemahan antara Hugging Face Transformers `smdistributed.modelparallel` dan dapat diakses sebagai berikut.

- `smdistributed.modelparallel.torch.nn.huggingface.gpt2.translate_state_dict_to_h max_seq_len=None)`
- `smdistributed.modelparallel.torch.nn.huggingface.gpt2.translate_hf_state_dict_t`
- `smdistributed.modelparallel.torch.nn.huggingface.bert.translate_state_dict_to_h max_seq_len=None)`
- `smdistributed.modelparallel.torch.nn.huggingface.bert.translate_hf_state_dict_t`
- `smdistributed.modelparallel.torch.nn.huggingface.roberta.translate_state_dict_t max_seq_len=None)`
- `smdistributed.modelparallel.torch.nn.huggingface.roberta.translate_hf_state_dic`
- `smdistributed.modelparallel.torch.nn.huggingface.gptj.translate_state_dict_to_h max_seq_len=None)`(Tersedia di perpustakaan paralelisme SageMaker model v1.8.0 dan yang lebih baru)
- `smdistributed.modelparallel.torch.nn.huggingface.gptj.translate_hf_gptj_state_d` di perpustakaan paralelisme SageMaker model v1.8.0 dan yang lebih baru)
- `smdistributed.modelparallel.torch.nn.huggingface.gptneo.translate_state_dict_to max_seq_len=None)`(Tersedia di perpustakaan paralelisme SageMaker model v1.10.0 dan yang lebih baru)
- `smdistributed.modelparallel.torch.nn.huggingface.gptneo.translate_hf_state_dict` di perpustakaan paralelisme SageMaker model v1.10.0 dan yang lebih baru)

Contoh penggunaan fungsi terjemahan GPT-2

Mulailah dengan membungkus model seperti yang ditunjukkan pada kode berikut.

```
from transformers import AutoModelForCausalLM

with smp.tensor_parallelism():
    model = AutoModelForCausalLM.from_config(hf_gpt2_config)

model = smp.DistributedModel(model)
```

Diberikan `state_dict` dari `DistributedModel` objek, Anda dapat memuat bobot ke dalam model GPT-2 Hugging Face asli menggunakan `translate_state_dict_to_hf_gpt2` fungsi seperti yang ditunjukkan pada kode berikut.

```
from smdistributed.modelparallel.torch.nn.huggingface.gpt2 \
```

```
                                        import translate_state_dict_to_hf_gpt2
max_seq_len = 1024

# [... code block for training ...]

if smp.rdp_rank() == 0:
    state_dict = dist_model.state_dict()
    hf_state_dict = translate_state_dict_to_hf_gpt2(state_dict, max_seq_len)

    # can now call model.load_state_dict(hf_state_dict) to the original HF model
```

Contoh penggunaan fungsi terjemahan RoberTA

Demikian pula, dengan HuggingFace model yang didukung`state_dict`, Anda dapat menggunakan `translate_hf_state_dict_to_smdistributed` fungsi untuk mengubahnya menjadi format yang dapat dibaca oleh`smp.DistributedModel`. Ini dapat berguna dalam kasus penggunaan pembelajaran transfer, di mana model yang telah dilatih sebelumnya dimuat ke dalam fine-tuning paralel `smp.DistributedModel` untuk model-paralel:

```
from smdistributed.modelparallel.torch.nn.huggingface.roberta \
                                        import translate_state_dict_to_smdistributed

model = AutoModelForMaskedLM.from_config(roberta_config)
model = smp.DistributedModel(model)

pretrained_model = AutoModelForMaskedLM.from_pretrained("roberta-large")
translated_state_dict =
        translate_state_dict_to_smdistributed(pretrained_model.state_dict())

# load the translated pretrained weights into the smp.DistributedModel
model.load_state_dict(translated_state_dict)

# start fine-tuning...
```

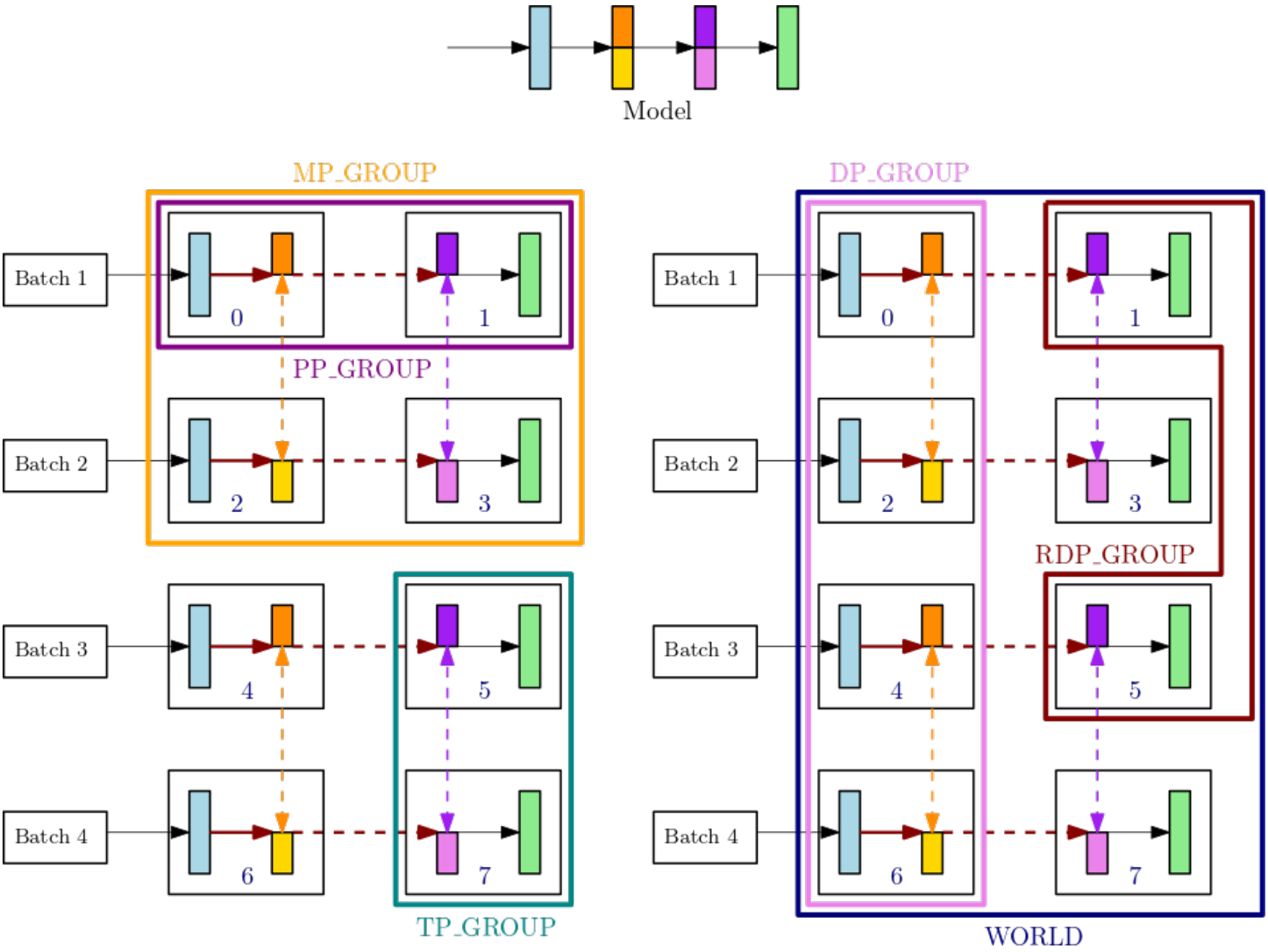
Mekanisme Pemeringkatan Saat Menggunakan Kombinasi Paralelisme Pipa dan Paralelisme Tensor

Bagian ini menjelaskan bagaimana mekanisme peringkat paralelisme model bekerja dengan paralelisme tensor. Ini diperpanjang dari Dasar-Dasar Ranking untukFitur Inti dari Perpustakaan Paralelisme SageMaker Model. Dengan paralelisme tensor, pustaka memperkenalkan tiga jenis API peringkat dan grup proses: untuk peringkat paralel `smp.tp_rank()` tensor, untuk peringkat paralel `smp.pp_rank()` pipeline, dan untuk `smp.rdp_rank()` peringkat paralel data yang dikurangi.

Kelompok proses komunikasi yang sesuai adalah tensor parallel group (TP_GROUP), pipeline parallel group (PP_GROUP), dan reduced-data parallel group RDP_GROUP (). Kelompok-kelompok ini didefinisikan sebagai berikut:

- Gugus paralel tensor (TP_GROUP) adalah subset yang dapat dibagi secara merata dari kelompok paralel data, di mana distribusi paralel tensor modul berlangsung. Ketika derajat paralelisme pipa adalah 1, TP_GROUP sama dengan model parallel group ()MP_GROUP.
- Pipeline parallel group (PP_GROUP) adalah kelompok proses di mana paralelisme pipa terjadi. Ketika derajat paralelisme tensor adalah 1, PP_GROUP sama dengan. MP_GROUP
- Reduced-data parallel group (RDP_GROUP) adalah serangkaian proses yang memegang partisi paralelisme pipa yang sama dan partisi paralelisme tensor yang sama, dan melakukan paralelisme data di antara mereka sendiri. Ini disebut grup paralelisme data tereduksi karena merupakan bagian dari seluruh kelompok paralelisme data,. DP_GROUP Untuk parameter model yang didistribusikan di dalamTP_GROUP, `allreduce` operasi gradien dilakukan hanya untuk grup paralel data tereduksi, sedangkan untuk parameter yang tidak didistribusikan, `allreduce` gradien berlangsung di seluruh. DP_GROUP
- Model parallel group (MP_GROUP) mengacu pada sekelompok proses yang secara kolektif menyimpan seluruh model. Ini terdiri dari penyatuan PP_GROUP s dari semua peringkat yang ada dalam TP_GROUP proses saat ini. Ketika derajat paralelisme tensor adalah 1, MP_GROUP setara dengan. PP_GROUP Hal ini juga konsisten dengan definisi yang ada dari MP_GROUP dari `smdistributed` rilis sebelumnya. Perhatikan bahwa arus TP_GROUP adalah bagian dari arus DP_GROUP dan arus. MP_GROUP

Untuk mempelajari lebih lanjut tentang API proses komunikasi di pustaka paralelisme SageMaker model, lihat Common API dan API PyTorch -specific dalam dokumentasi Python SageMaker SDK.



This figure shows ranking mechanism, parameter distribution, and associated AllReduce operations of tensor parallelism.

Misalnya, pertimbangkan grup proses untuk satu node dengan 8 GPU, di mana derajat paralelisme tensor adalah 2, derajat paralelisme pipa adalah 2, dan tingkat paralelisme data adalah 4. Bagian tengah atas dari gambar sebelumnya menunjukkan contoh model dengan 4 lapisan. Bagian kiri bawah dan kanan bawah gambar menggambarkan model 4-lapisan yang didistribusikan di 4 GPU menggunakan paralelisme pipa dan paralelisme tensor, di mana paralelisme tensor digunakan untuk dua lapisan tengah. Kedua angka yang lebih rendah ini adalah salinan sederhana untuk menggambarkan garis batas kelompok yang berbeda. Model yang dipartisi direplikasi untuk paralelisme data di seluruh GPU 0-3 dan 4-7. Gambar kiri bawah menunjukkan definisi`MP_GROUP`,`PP_GROUP`, dan`TP_GROUP`. Angka kanan bawah menunjukkan`RDP_GROUP`,`DP_GROUP`, dan `WORLD` di atas set GPU yang sama. Gradien untuk lapisan dan irisan lapisan yang memiliki warna yang sama adalah `allreduce` d bersama-

sama untuk paralelisme data. Misalnya, lapisan pertama (biru muda) mendapatkan `allreduce` operasi`DP_GROUP`, sedangkan irisan oranye gelap di lapisan kedua hanya mendapatkan `allreduce` operasi dalam prosesnya`RDP_GROUP`. Panah merah tua yang berani mewakili tensor dengan batch keseluruhannya. `TP_GROUP`

```
GPU0: pp_rank 0, tp_rank 0, rdp_rank 0, dp_rank 0, mp_rank 0
GPU1: pp_rank 1, tp_rank 0, rdp_rank 0, dp_rank 0, mp_rank 1
GPU2: pp_rank 0, tp_rank 1, rdp_rank 0, dp_rank 1, mp_rank 2
GPU3: pp_rank 1, tp_rank 1, rdp_rank 0, dp_rank 1, mp_rank 3
GPU4: pp_rank 0, tp_rank 0, rdp_rank 1, dp_rank 2, mp_rank 0
GPU5: pp_rank 1, tp_rank 0, rdp_rank 1, dp_rank 2, mp_rank 1
GPU6: pp_rank 0, tp_rank 1, rdp_rank 1, dp_rank 3, mp_rank 2
GPU7: pp_rank 1, tp_rank 1, rdp_rank 1, dp_rank 3, mp_rank 3
```

Dalam contoh ini, paralelisme pipa terjadi di seluruh pasangan GPU (0,1); (2,3); (4,5) dan (6,7). Selain itu, paralelisme data (`allreduce`) terjadi di seluruh GPU 0, 2, 4, 6, dan secara independen melalui GPU 1, 3, 5, 7. Paralelisme tensor terjadi pada himpunan bagian dari `DP_GROUP` s, di seluruh pasangan GPU (0,2); (1,3); (4,6) dan (5,7).

Sharding Status Optimizer

Sharding status pengoptimal adalah teknik hemat memori yang berguna yang memecah status pengoptimal (kumpulan bobot yang menjelaskan status pengoptimal) di seluruh grup perangkat paralel data. Anda dapat menggunakan sharding status pengoptimal setiap kali Anda menggunakan pengoptimal stateful (seperti Adam) atau pengoptimal FP16 (yang menyimpan salinan parameter FP16 dan FP32).

> **Note**
>
> Sharding status pengoptimal tersedia PyTorch di pustaka paralelisme SageMaker model v1.6.0 dan yang lebih baru.

Cara Menggunakan Optimizer State Sharding

Anda dapat mengaktifkan sharding status pengoptimal dengan mengatur konfigurasi`"shard_optimizer_state": True`. `modelparallel`

Saat fitur ini dihidupkan, pustaka mempartisi kumpulan parameter model berdasarkan tingkat paralelisme data. Gradien yang sesuai dengan partisi `i` th berkurang hanya pada peringkat paralel

data `i` th. Pada akhir panggilan pertama ke fungsi `smp.step` dekorator, pengoptimal yang dibungkus dengan `smp.DistributedOptimizer` mendefinisikan ulang parameternya hanya terbatas pada parameter yang sesuai dengan partisi peringkat paralel data saat ini. Parameter yang didefinisikan ulang disebut parameter virtual dan berbagi penyimpanan yang mendasarinya dengan parameter asli. Selama panggilan pertama ke`optimizer.step`, status pengoptimal dibuat berdasarkan parameter yang didefinisikan ulang ini, yang di-sharded karena partisi asli. Setelah pembaruan pengoptimal, AllGather operasi (sebagai bagian dari `optimizer.step` panggilan) berjalan di seluruh peringkat paralel data untuk mencapai status parameter yang konsisten.

> **Tip**
>
> Sharding status pengoptimal dapat berguna ketika tingkat paralelisme data lebih besar dari 1 dan model memiliki lebih dari satu miliar parameter.
>
> Tingkat paralelisme data dihitung oleh`(processes_per_host * instance_count / pipeline_parallel_degree)`, dan `smp.dp_size()` fungsi menangani ukuran di latar belakang.

Konfigurasikan SageMaker PyTorch estimator

```
mpi_options = {
    "enabled" : True,
    "processes_per_host" : 8,               # 8 processes
    "custom_mpi_options" : "--mca btl_vader_single_copy_mechanism none "
}

smp_options = {
    "enabled":True,
    "parameters": {
        "microbatches": 4,
        "pipeline_parallel_degree": 2,    # alias for "partitions"
        "placement_strategy": "cluster",
        "tensor_parallel_degree": 2,      # tp over 2 devices
        "ddp": True,
        "shard_optimizer_state": True
    }
}
```

Sesuaikan skrip PyTorch pelatihan Anda

Lihat Menyesuaikan skrip PyTorch pelatihan Anda di bagian paralelisme Tensor yang dikombinasikan dengan paralelisme pipa. Tidak ada modifikasi tambahan yang diperlukan untuk skrip.

Aktivasi Checkpointing

Activation checkpointing (atau gradient checkpointing) adalah teknik untuk mengurangi penggunaan memori dengan membersihkan aktivasi lapisan tertentu dan mengkomputernya kembali selama backward pass. Secara efektif, ini memperdagangkan waktu komputasi ekstra untuk mengurangi penggunaan memori. Jika modul diperiksa, di akhir pass maju, input ke dan output dari modul tetap berada di memori. Tensor perantara apa pun yang akan menjadi bagian dari perhitungan di dalam modul itu dibebaskan selama pass maju. Selama lintasan mundur modul checkpoint, tensor ini dihitung ulang. Pada titik ini, lapisan di luar modul checkpointed ini telah menyelesaikan backward pass mereka, sehingga penggunaan memori puncak dengan checkpointing bisa lebih rendah.

> **Note**
>
> Fitur ini tersedia untuk PyTorch di pustaka paralelisme SageMaker model v1.6.0 dan yang lebih baru.

Cara Menggunakan Checkpointing Aktivasi

Dengan`smdistributed.modelparallel`, Anda dapat menggunakan pos pemeriksaan aktivasi pada perincian modul. Untuk semua `torch.nn` modul kecuali`torch.nn.Sequential`, Anda hanya dapat memeriksa pohon modul jika terletak dalam satu partisi dari perspektif paralelisme pipa. Dalam kasus `torch.nn.Sequential` modul, setiap pohon modul di dalam modul sekuensial harus terletak sepenuhnya dalam satu partisi agar pos pemeriksaan aktivasi berfungsi. Saat Anda menggunakan partisi manual, perhatikan batasan ini.

Saat Anda menggunakan partisi model otomatis, Anda dapat menemukan log tugas partisi yang dimulai dengan `Partition assignments:` di log pekerjaan pelatihan. Jika modul dipartisi di beberapa peringkat (misalnya, dengan satu keturunan pada satu peringkat dan keturunan lain pada peringkat yang berbeda), perpustakaan mengabaikan upaya untuk memeriksa modul dan memunculkan pesan peringatan bahwa modul tidak akan diperiksa.

> **Note**
>
> Pustaka paralelisme SageMaker model mendukung operasi yang tumpang tindih dan tidak tumpang tindih dalam kombinasi dengan pos `allreduce` pemeriksaan.

> **Note**
>
> PyTorchAPI checkpointing asli tidak kompatibel dengan. `smdistributed.modelparallel`

Contoh 1: Kode contoh berikut menunjukkan cara menggunakan checkpointing aktivasi ketika Anda memiliki definisi model dalam skrip Anda.

```
import torch.nn as nn
import torch.nn.functional as F

from smdistributed.modelparallel.torch.patches.checkpoint import checkpoint

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 3, 1)
        self.conv2 = nn.Conv2d(32, 64, 3, 1)
        self.fc1 = nn.Linear(9216, 128)
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = self.conv1(x)
        x = self.conv2(x)
        x = F.max_pool2d(x, 2)
        x = torch.flatten(x, 1)
        # This call of fc1 will be checkpointed
        x = checkpoint(self.fc1, x)
        x = self.fc2(x)
        return F.log_softmax(x, 1)
```

Contoh 2: Kode contoh berikut menunjukkan cara menggunakan checkpointing aktivasi ketika Anda memiliki model sekuensial dalam skrip Anda.

```
import torch.nn as nn
from smdistributed.modelparallel.torch.patches.checkpoint import checkpoint_sequential

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.seq = nn.Sequential(
            nn.Conv2d(1,20,5),
```

```
            nn.ReLU(),
            nn.Conv2d(20,64,5),
            nn.ReLU()
        )

    def forward(self, x):
        # This call of self.seq will be checkpointed
        x = checkpoint_sequential(self.seq, x)
        return F.log_softmax(x, 1)
```

Contoh 3: Contoh kode berikut menunjukkan cara menggunakan checkpointing aktivasi saat Anda mengimpor model bawaan dari pustaka, seperti dan PyTorch Hugging Face Transformers. Apakah Anda memeriksa modul sekuensial atau tidak, lakukan hal berikut:

1. Bungkus model dengan`smp.DistributedModel()`.
2. Tentukan objek untuk lapisan berurutan.
3. Bungkus objek layer sekuensial dengan`smp.set_activation_checkpointig()`.

```
import smdistributed.modelparallel.torch as smp
from transformers import AutoModelForCausalLM

smp.init()
model = AutoModelForCausalLM(*args, **kwargs)
model = smp.DistributedModel(model)

# Call set_activation_checkpointing API
transformer_layers = model.module.module.module.transformer.seq_layers
smp.set_activation_checkpointing(
    transformer_layers, pack_args_as_tuple=True, strategy='each')
```

Pembongkaran Aktivasi

Ketika checkpointing aktivasi dan paralelisme pipa dihidupkan dan jumlah microbatch lebih besar dari satu, pembongkaran aktivasi adalah fitur tambahan yang selanjutnya dapat mengurangi penggunaan memori. Pembongkaran aktivasi secara asinkron memindahkan aktivasi checkpoint yang sesuai dengan microbatch mereka yang saat ini tidak berjalan di CPU. Tepat sebelum GPU membutuhkan aktivasi untuk backward pass microbatch, fungsi ini mengambil kembali aktivasi yang diturunkan dari CPU.

> **Note**
>
> Fitur ini tersedia untuk PyTorch di pustaka paralelisme SageMaker model v1.6.0 dan yang lebih baru.

Cara Menggunakan Pembongkaran Aktivasi

Gunakan pembongkaran aktivasi untuk mengurangi penggunaan memori ketika jumlah microbatch lebih besar dari 1, dan pos pemeriksaan aktivasi diaktifkan (lihat). Aktivasi Checkpointing Ketika checkpointing aktivasi tidak digunakan, pembongkaran aktivasi tidak berpengaruh. Ketika digunakan hanya dengan satu microbatch, itu tidak menghemat memori.

Untuk menggunakan pembongkaran aktivasi, atur `"offload_activations": True` dalam `modelparallel` konfigurasi.

Pembongkaran aktivasi memindahkan aktivasi checkpoint dalam `nn.Sequential` modul ke CPU secara asinkron. Transfer data melalui tautan PCIe tumpang tindih dengan komputasi GPU. Pembongkaran terjadi segera, segera setelah pass maju untuk lapisan checkpoint tertentu dihitung. Aktivasi dimuat kembali ke GPU sesaat sebelum diperlukan untuk pass mundur dari microbatch tertentu. Transfer CPU-GPU juga tumpang tindih dengan komputasi.

Untuk menyesuaikan seberapa awal aktivasi dimuat kembali ke GPU, Anda dapat menggunakan parameter konfigurasi `"activation_loading_horizon"` (default diatur ke 4, harus `int` lebih besar dari 0). Cakrawala pemuatan aktivasi yang lebih besar akan menyebabkan aktivasi dimuat kembali ke GPU sebelumnya. Jika cakrawala terlalu besar, dampak penghematan memori dari pembongkaran aktivasi mungkin berkurang. Jika cakrawala terlalu kecil, aktivasi mungkin tidak dimuat kembali ke masa lalu, mengurangi jumlah tumpang tindih dan menurunkan kinerja.

> **Tip**
>
> Pembongkaran aktivasi dapat berguna untuk model besar dengan lebih dari seratus miliar parameter.

Konfigurasikan SageMaker PyTorch estimator

```
mpi_options = {
    "enabled" : True,
```

```
    "processes_per_host" : 8,               # 8 processes
    "custom_mpi_options" : "--mca btl_vader_single_copy_mechanism none "
}

smp_options = {
    "enabled":True,
    "parameters": {
        "microbatches": 4,
        "pipeline_parallel_degree": 2,    # alias for "partitions"
        "placement_strategy": "cluster",
        "tensor_parallel_degree": 2,      # tp over 2 devices
        "ddp": True,
        "offload_activations": True,
        "activation_loading_horizon": 4   # optional. default is 4.
    }
}
```

Pelatihan FP16 dengan Paralelisme Model

Untuk pelatihan FP16, terapkan modifikasi berikut pada skrip pelatihan dan estimator Anda.

> **Note**
>
> Fitur ini tersedia untuk PyTorch di pustaka paralelisme SageMaker model v1.10.0 dan yang lebih baru.

Sesuaikan skrip PyTorch pelatihan Anda

1. Bungkus model Anda menggunakan pengelola konteks smdistributed.modelparallel.torch.model_creation ().

```
# fp16_training_script.py

import torch
import smdistributed.modelparallel.torch as smp

with smp.model_creation(
    dtype=torch.float16 if args.fp16 else torch.get_default_dtype()
):
    model = ...
```

> **Tip**
>
> Jika Anda menggunakan paralelisme tensor, tambahkan `tensor_parallelism=smp.tp_size() > 1` ke manajer konteks. `smp.model_creation` Menambahkan baris ini juga membantu mendeteksi secara otomatis apakah paralelisme tensor diaktifkan atau tidak.
>
> ```
> with smp.model_creation(
>     ... ,
>     tensor_parallelism=smp.tp_size() > 1
> ):
>     model = ...
> ```

2. Saat Anda membungkus pengoptimal dengan`smdistributed.modelparallel.torch.DistributedOptimizer`, atur `dynamic_loss_scaling` argumen `static_loss_scaling` atau. Secara default, `static_loss_scaling` diatur ke `1.0`, dan `dynamic_loss_scaling` diatur ke `False`. Jika Anda mengaturdynamic_loss_scale=True, Anda dapat memasukkan opsi penskalaan kerugian dinamis sebagai kamus melalui `dynamic_loss_args` argumen. Dalam kebanyakan kasus, kami sarankan Anda menggunakan penskalaan kerugian dinamis dengan opsi default. Untuk informasi selengkapnya, opsi, dan contoh fungsi pembungkus pengoptimal, lihat smdistributed.modelparallel.torch. DistributedOptimizerAPI.

   Kode berikut adalah contoh pembungkus objek `Adadelta` pengoptimal dengan penskalaan kerugian dinamis untuk pelatihan FP16.

   ```
   optimizer = torch.optim.Adadelta(...)
   optimizer = smp.DistributedOptimizer(
       optimizer,
       static_loss_scale=None,
       dynamic_loss_scale=True,
       dynamic_loss_args={
           "scale_window": 1000,
           "min_scale": 1,
           "delayed_shift": 2
       }
   )
   ```

Konfigurasikan SageMaker PyTorch estimator

Tambahkan parameter FP16 (`"fp16"`) ke konfigurasi distribusi untuk paralelisme model saat membuat objek estimator. SageMaker PyTorch Untuk daftar lengkap parameter konfigurasi paralelisme model, lihat Parameter untuk. `smdistributed`

```
from sagemaker.pytorch import PyTorch

smp_options = {
    "enabled": True,
    "parameters":  {
        "microbatches":  4,
        "pipeline_parallel_degree":  2,
        "tensor_parallel_degree":  2,
        ...,

        "fp16": True
    }
}

fp16_estimator = PyTorch(
    entry_point="fp16_training_script.py", # Specify your train script
    ...,

    distribution={
        "smdistributed": {"modelparallel": smp_options},
        "mpi": {...}
    }
)

fp16_estimator.fit(...)
```

Ketika pelatihan FP16 dimulai, model dan pengoptimal dibungkus oleh `FP16_Module` dan `FP16_Optimizer` masing-masing, yang merupakan `smdistributed` versi modifikasi dari utilitas Apex. `FP16_Module`mengubah model ke FP16 dtype dan berurusan dengan pass maju di FP16.

> **Tip**
>
> Anda dapat menerapkan kliping gradien dengan menelepon `clip_master_grads` sebelumnya. `optimizer.step`

```
optimizer.clip_master_grads(max_norm)     # max_norm(float or int): max norm of
 the gradients
```

> **Tip**
>
> Saat menggunakan `torch.optim.lr_scheduler` dan pelatihan FP16, Anda harus meneruskan `optimizer.optimizer` ke penjadwal LR daripada pengoptimal. Lihat contoh kode berikut.
>
> ```
> from torch.optim.lr_scheduler import StepLR
>
> scheduler = StepLR(
>     optimizer.optimizer if smp.state.cfg.fp16 else optimizer,
>     step_size=1,
>     gamma=args.gamma
> )
> ```

Support untuk FlashAttention

Support for FlashAttention adalah fitur perpustakaan yang hanya berlaku untuk model transformator terdistribusi, yang merupakan model Transformer yang dibungkus oleh `smp.DistributedModel()`untuk pelatihan model-paralel. Fitur ini juga kompatibel denganthe section called “Paralelisme Tensor”.

FlashAttentionPustaka hanya mendukung model ketika `attention_head_size` disetel ke nilai yang kelipatan 8 dan kurang dari 128. Oleh karena itu, ketika Anda melatih transformator terdistribusi dan memastikannya FlashAttention berfungsi dengan baik, Anda harus menyesuaikan parameter untuk membuat ukuran kepala perhatian memenuhi persyaratan. Untuk informasi selengkapnya, lihat juga Instalasi dan fitur di FlashAttention GitHubrepositori.

Misalnya, asumsikan bahwa Anda mengonfigurasi model Transformer dengan `hidden_width=864` dan`num_heads=48`. Ukuran kepala FlashAttention dihitung sebagai`attention_head_size = hidden_width / num_heads = 864 / 48 = 18`. Untuk mengaktifkan FlashAttention, Anda perlu menyesuaikan `num_heads` parameter ke`54`, sehingga`attention_head_size = hidden_width / num_heads = 864 / 54 = 16`, yang merupakan kelipatan dari 8.

Jalankan Job Pelatihan SageMaker Terdistribusi dengan Paralelisme Model

Pelajari cara menjalankan tugas pelatihan model-paralel dari skrip pelatihan Anda sendiri menggunakan SageMaker Python SDK dengan pustaka paralelisme model. SageMaker

Ada tiga skenario kasus penggunaan untuk menjalankan pekerjaan SageMaker pelatihan.

1. Anda dapat menggunakan salah satu AWS Deep Learning Container yang sudah dibuat sebelumnya untuk TensorFlow dan PyTorch. Opsi ini disarankan jika ini adalah pertama kalinya bagi Anda untuk menggunakan perpustakaan paralel model. Untuk menemukan tutorial tentang cara menjalankan pekerjaan pelatihan paralel SageMaker model, lihat contoh notebook saat PyTorch pelatihan dengan perpustakaan SageMaker paralelisme model Amazon.
2. Anda dapat memperluas kontainer pra-bangun untuk menangani persyaratan fungsional tambahan apa pun untuk algoritme atau model Anda yang tidak didukung oleh image SageMaker Docker yang sudah dibuat sebelumnya. Untuk menemukan contoh bagaimana Anda dapat memperluas kontainer yang sudah dibuat sebelumnya, lihatMemperluas Kontainer Pra-dibangun.
3. Anda dapat menyesuaikan wadah Docker Anda sendiri untuk bekerja dengan SageMaker menggunakan toolkit SageMaker Pelatihan. Sebagai contoh, lihat Mengadaptasi Wadah Pelatihan Anda Sendiri.

Untuk opsi 2 dan 3 di daftar sebelumnya, lihat Perluas Container Docker Pra-built yang Berisi Perpustakaan Paralel Model SageMaker Terdistribusi untuk mempelajari cara menginstal pustaka paralel model dalam wadah Docker yang diperluas atau disesuaikan.

Dalam semua kasus, Anda meluncurkan tugas pelatihan dengan mengonfigurasi `PyTorch` estimator SageMaker `TensorFlow` atau untuk mengaktifkan pustaka. Untuk mempelajari lebih lanjut, lihat topik berikut.

Topik

- Langkah 1: Ubah Skrip Pelatihan Anda Sendiri Menggunakan SageMaker Perpustakaan Paralel Model Terdistribusi
- Langkah 2: Luncurkan Training Job Menggunakan SageMaker Python SDK

Langkah 1: Ubah Skrip Pelatihan Anda Sendiri Menggunakan SageMaker Perpustakaan Paralel Model Terdistribusi

Gunakan bagian ini untuk mempelajari cara menyesuaikan skrip pelatihan Anda untuk menggunakan fitur inti dari perpustakaan paralelisme SageMaker model Amazon. Untuk menggunakan fungsi dan

parameter API khusus perpustakaan, kami sarankan Anda menggunakan dokumentasi ini bersama API library SageMaker paralel model dalam dokumentasi Python SageMaker SDK.

Contoh skrip pelatihan yang disediakan di bagian ini disederhanakan dan dirancang untuk menyoroti perubahan yang diperlukan yang harus Anda lakukan untuk menggunakan perpustakaan. Untuk end-to-end contoh buku catatan yang dapat dijalankan yang menunjukkan cara menggunakan skrip TensorFlow atau PyTorch pelatihan dengan pustaka paralelisme SageMaker model, lihat. Contoh Notebook Pelatihan SageMaker Terdistribusi Amazon

Topik

- Pisahkan model skrip pelatihan Anda menggunakan pustaka paralelisme SageMaker model
- Memodifikasi skrip TensorFlow pelatihan
- Memodifikasi Skrip PyTorch Pelatihan

Pisahkan model skrip pelatihan Anda menggunakan pustaka paralelisme SageMaker model

Ada dua cara untuk memodifikasi skrip pelatihan Anda untuk mengatur pemisahan model: pemisahan otomatis atau pemisahan manual.

Pemisahan model otomatis

Saat Anda menggunakan SageMaker pustaka paralelisme model, Anda dapat memanfaatkan pemisahan model otomatis, juga disebut sebagai partisi model otomatis. Pustaka menggunakan algoritma partisi yang menyeimbangkan memori, meminimalkan komunikasi antar perangkat, dan mengoptimalkan kinerja. Anda dapat mengonfigurasi algoritma partisi otomatis untuk mengoptimalkan kecepatan atau memori.

Atau, Anda dapat menggunakan pemisahan model manual. Kami merekomendasikan pemisahan model otomatis, kecuali jika Anda sangat akrab dengan arsitektur model dan memiliki ide bagus tentang cara mempartisi model Anda secara efisien.

Cara kerjanya

Partisi otomatis terjadi selama langkah pelatihan pertama, ketika fungsi `smp.step` -decorated pertama kali dipanggil. Selama panggilan ini, perpustakaan pertama-tama membuat versi model pada RAM CPU (untuk menghindari keterbatasan memori GPU), dan kemudian menganalisis grafik model dan membuat keputusan partisi. Berdasarkan keputusan ini, setiap partisi model dimuat pada GPU, dan baru kemudian langkah pertama dijalankan. Karena langkah-langkah analisis dan partisi ini, langkah pelatihan pertama mungkin memakan waktu lebih lama.

Dalam kedua kerangka kerja, perpustakaan mengelola komunikasi antar perangkat melalui backend sendiri, yang dioptimalkan untuk AWS infrastruktur.

Desain partisi otomatis menyesuaikan dengan karakteristik kerangka kerja, dan perpustakaan melakukan partisi pada tingkat granularitas yang lebih alami di setiap kerangka kerja. Misalnya, di TensorFlow, setiap operasi tertentu dapat ditugaskan ke perangkat yang berbeda, sedangkan pada PyTorch, penugasan dilakukan pada tingkat modul, di mana setiap modul terdiri dari beberapa operasi. Bagian berikut mengulas spesifikasi desain di setiap kerangka kerja.

Pemisahan model otomatis dengan PyTorch

Selama langkah pelatihan pertama, perpustakaan paralelisme model secara internal menjalankan langkah penelusuran yang dimaksudkan untuk membangun grafik model dan menentukan bentuk tensor dan parameter. Setelah langkah penelusuran ini, perpustakaan membangun pohon, yang terdiri dari `nn.Module` objek bersarang dalam model, serta data tambahan yang dikumpulkan dari penelusuran, seperti jumlah yang disimpan`nn.Parameters`, dan waktu eksekusi untuk masing-masing. `nn.Module`

Selanjutnya, perpustakaan melintasi pohon ini dari root dan menjalankan algoritma partisi yang menetapkan masing-masing `nn.Module` ke perangkat, yang menyeimbangkan beban komputasi (diukur dengan waktu eksekusi modul) dan penggunaan memori (diukur dengan total ukuran dan aktivasi yang disimpan). `nn.Parameter` Jika beberapa `nn.Modules` berbagi yang sama`nn.Parameter`, maka modul ini ditempatkan pada perangkat yang sama untuk menghindari mempertahankan beberapa versi dari parameter yang sama. Setelah keputusan partisi dibuat, modul dan bobot yang ditetapkan dimuat ke perangkat mereka.

Untuk petunjuk tentang cara mendaftarkan `smp.step` dekorator ke skrip PyTorch pelatihan Anda, lihatthe section called “Pemisahan otomatis dengan PyTorch”.

Pemisahan model otomatis dengan TensorFlow

Pustaka paralelisme model menganalisis ukuran variabel yang dapat dilatih dan struktur grafik, dan secara internal menggunakan algoritma partisi grafik. Algoritma ini hadir dengan penugasan perangkat untuk setiap operasi, dengan tujuan meminimalkan jumlah komunikasi yang dibutuhkan di seluruh perangkat, tunduk pada dua kendala:

- Menyeimbangkan jumlah variabel yang disimpan di setiap perangkat
- Menyeimbangkan jumlah operasi yang dijalankan di setiap perangkat

Jika Anda menentukan `speed` for `optimize` (dalam parameter paralelisme model di SDK Python), pustaka mencoba menyeimbangkan jumlah operasi dan `tf.Variable` objek di setiap perangkat. Jika tidak, ia mencoba menyeimbangkan ukuran total`tf.Variables`.

Setelah keputusan partisi dibuat, pustaka akan membuat representasi serial dari subgraf yang perlu dijalankan oleh setiap perangkat dan mengimpornya ke setiap perangkat. Saat mempartisi, perpustakaan menempatkan operasi yang menggunakan yang sama `tf.Variable` dan operasi yang merupakan bagian dari lapisan Keras yang sama ke perangkat yang sama. Ini juga menghormati kendala kolokasi yang diberlakukan oleh. TensorFlow Ini berarti bahwa, misalnya, jika ada dua lapisan Keras yang berbagi `atf.Variable`, maka semua operasi yang merupakan bagian dari lapisan ini ditempatkan pada satu perangkat.

Untuk petunjuk tentang cara mendaftarkan `smp.step` dekorator ke skrip PyTorch pelatihan Anda, lihatthe section called “Pemisahan otomatis dengan TensorFlow”.

Perbandingan pemisahan model otomatis antar kerangka kerja

Dalam TensorFlow, unit dasar komputasi adalah `atf.Operation`, dan TensorFlow mewakili model sebagai grafik asiklik terarah (DAG) dari `tf.Operation` s, dan oleh karena itu perpustakaan paralelisme model mempartisi DAG ini sehingga setiap node masuk ke satu perangkat. Yang terpenting, `tf.Operation` objek cukup kaya dengan atribut yang dapat disesuaikan, dan bersifat universal dalam arti bahwa setiap model dijamin terdiri dari grafik objek tersebut.

PyTorch di sisi lain, tidak memiliki gagasan operasi yang setara yang cukup kaya dan universal. Unit komputasi terdekat PyTorch yang memiliki karakteristik ini adalahnn`.Module`, yang berada pada tingkat granularitas yang jauh lebih tinggi, dan inilah mengapa perpustakaan melakukan partisi pada tingkat ini di. PyTorch

Pemisahan Model Manual

Jika Anda ingin menentukan secara manual cara mempartisi model Anda di seluruh perangkat, gunakan pengelola `smp.partition` konteks. Untuk petunjuk tentang cara mengatur manajer konteks untuk partisi manual, lihat halaman berikut.

- the section called “Pemisahan manual dengan TensorFlow”
- the section called “Pemisahan manual dengan PyTorch”

Untuk menggunakan opsi ini setelah melakukan modifikasi, pada Langkah 2, Anda harus mengatur `auto_partition` ke`False`, dan menentukan kelas estimator kerangka kerja dari SageMaker

Python SDK. `default_partition` Setiap operasi yang tidak secara eksplisit ditempatkan pada partisi melalui manajer `smp.partition` konteks dijalankan pada file. `default_partition` Dalam hal ini, logika pemisahan otomatis dilewati, dan setiap operasi ditempatkan berdasarkan spesifikasi Anda. Berdasarkan struktur grafik yang dihasilkan, pustaka paralelisme model membuat jadwal eksekusi pipelined secara otomatis.

Memodifikasi skrip TensorFlow pelatihan

Di bagian ini, Anda mempelajari cara memodifikasi skrip TensorFlow pelatihan untuk mengonfigurasi pustaka paralelisme SageMaker model untuk partisi otomatis dan partisi manual. Pemilihan contoh ini juga mencakup contoh yang terintegrasi dengan Horovod untuk model hibrida dan paralelisme data.

> **Note**
>
> Untuk menemukan TensorFlow versi mana yang didukung oleh perpustakaan, lihatthe section called “Kerangka Kerja yang Didukung dan Wilayah AWS”.

Modifikasi yang diperlukan yang harus Anda lakukan pada skrip pelatihan Anda untuk menggunakan perpustakaan tercantum di dalamnyaPemisahan otomatis dengan TensorFlow.

Untuk mempelajari cara memodifikasi skrip pelatihan Anda untuk menggunakan model hibrida dan paralelisme data dengan Horovod, lihat. Pemisahan otomatis dengan TensorFlow dan Horovod untuk model hibrida dan paralelisme data

Jika Anda ingin menggunakan partisi manual, tinjau juga. Pemisahan manual dengan TensorFlow

> **Tip**
>
> Untuk contoh end-to-end buku catatan yang menunjukkan cara menggunakan skrip TensorFlow pelatihan dengan pustaka paralelisme SageMaker model, lihat. TensorFlowContoh

Topik berikut menunjukkan contoh skrip pelatihan yang dapat Anda gunakan untuk mengonfigurasi SageMaker pustaka paralelisme model untuk model partisi otomatis dan partisi manual. TensorFlow

> **Note**
>
> Partisi otomatis diaktifkan secara default. Kecuali ditentukan lain, skrip contoh menggunakan partisi otomatis.

Topik

Pemisahan otomatis dengan TensorFlow

Perubahan skrip pelatihan berikut diperlukan untuk menjalankan TensorFlow model dengan pustaka SageMaker paralelisme model:

1. Impor dan inisialisasi perpustakaan dengan `smp.init()`.
2. Mendefinisikan model Keras dengan mewarisi dari `smp.DistributedModel`bukan kelas Model Keras. Kembalikan output model dari metode panggilan `smp.DistributedModel` objek. Perhatikan bahwa setiap tensor yang dikembalikan dari metode panggilan akan disiarkan di seluruh perangkat paralel model, yang menimbulkan overhead komunikasi, jadi tensor apa pun yang tidak diperlukan di luar metode panggilan (seperti aktivasi perantara) tidak boleh dikembalikan.
3. Ditetapkan `drop_remainder=True` dalam `tf.Dataset.batch()` metode. Ini untuk memastikan bahwa ukuran batch selalu habis dibagi dengan jumlah microbatch.
4. Benih operasi acak dalam pipa data menggunakan`smp.dp_rank()`, misalnya, `shuffle(ds, seed=smp.dp_rank())` untuk memastikan konsistensi sampel data di seluruh GPU yang memegang partisi model yang berbeda.
5. Letakkan logika maju dan mundur dalam fungsi langkah dan hiasi dengan`smp.step`.
6. Lakukan pasca-pemrosesan pada output di seluruh microbatch menggunakan `StepOutput`metode seperti. `reduce_mean` `smp.step`Fungsi harus memiliki nilai kembali yang tergantung pada output dari`smp.DistributedModel`.
7. Jika ada langkah evaluasi, tempatkan logika penerusan di dalam fungsi `smp.step` -decorated dan pasca-proses output menggunakan API. `StepOutput`

Untuk mempelajari lebih lanjut tentang API pustaka paralelisme model, lihat dokumentasi API. SageMaker

Skrip Python berikut adalah contoh skrip pelatihan setelah perubahan dilakukan.

```
import tensorflow as tf

# smdistributed: Import TF2.x API
import smdistributed.modelparallel.tensorflow as smp

# smdistributed: Initialize
smp.init()

# Download and load MNIST dataset.
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data(
    "MNIST-data-%d" % smp.rank()
)
x_train, x_test = x_train / 255.0, x_test / 255.0

# Add a channels dimension
x_train = x_train[..., tf.newaxis]
x_test = x_test[..., tf.newaxis]

# smdistributed: If needed, seed the shuffle with smp.dp_rank(), and drop_remainder
# in batching to make sure batch size is always divisible by number of microbatches
train_ds = (
    tf.data.Dataset.from_tensor_slices((x_train, y_train))
    .shuffle(10000, seed=smp.dp_rank())
    .batch(256, drop_remainder=True)
)

# smdistributed: Define smp.DistributedModel the same way as Keras sub-classing API
class MyModel(smp.DistributedModel):
    def __init__(self):
        super(MyModel, self).__init__()
        # define layers

    def call(self, x, training=None):
        # define forward pass and return the model output

model = MyModel()

loss_object = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
optimizer = tf.keras.optimizers.Adam()
```

```
train_accuracy = tf.keras.metrics.SparseCategoricalAccuracy(name="train_accuracy")

# smdistributed: Define smp.step. Return any tensors needed outside
@smp.step
def get_grads(images, labels):
    predictions = model(images, training=True)
    loss = loss_object(labels, predictions)

    grads = optimizer.get_gradients(loss, model.trainable_variables)
    return grads, loss, predictions


@tf.function
def train_step(images, labels):
    gradients, loss, predictions = get_grads(images, labels)

    # smdistributed: Accumulate the gradients across microbatches
    gradients = [g.accumulate() for g in gradients]
    optimizer.apply_gradients(zip(gradients, model.trainable_variables))

    # smdistributed: Merge predictions and average losses across microbatches
    train_accuracy(labels, predictions.merge())
    return loss.reduce_mean()


for epoch in range(5):
    # Reset the metrics at the start of the next epoch
    train_accuracy.reset_states()
    for images, labels in train_ds:
        loss = train_step(images, labels)
    accuracy = train_accuracy.result()
```

Jika Anda selesai mempersiapkan skrip pelatihan Anda, lanjutkan keLangkah 2: Luncurkan Training Job Menggunakan SageMaker Python SDK. Jika Anda ingin menjalankan model hybrid dan pekerjaan pelatihan paralel data, lanjutkan ke bagian berikutnya.

Pemisahan otomatis dengan TensorFlow dan Horovod untuk model hibrida dan paralelisme data

Anda dapat menggunakan perpustakaan paralelisme SageMaker model dengan Horovod untuk model hibrida dan paralelisme data. Untuk membaca lebih lanjut tentang bagaimana perpustakaan membagi model untuk paralelisme hibrida, lihat. Paralelisme pipa (tersedia untuk PyTorch dan) TensorFlow

Pada langkah ini, kami fokus pada cara memodifikasi skrip pelatihan Anda untuk mengadaptasi perpustakaan paralelisme SageMaker model.

Untuk mengatur skrip pelatihan dengan benar untuk mengambil konfigurasi paralelisme hibrida yang akan Anda aturLangkah 2: Luncurkan Training Job Menggunakan SageMaker Python SDK, gunakan fungsi pembantu perpustakaan, `smp.dp_rank()` dan`smp.mp_rank()`, yang secara otomatis mendeteksi peringkat paralel data dan peringkat paralel model masing-masing.

Untuk menemukan semua primitif MPI yang didukung perpustakaan, lihat Dasar MPI dalam dokumentasi Python SageMaker SDK.

Perubahan yang diperlukan dalam skrip adalah:

- Menambahkan `hvd.allreduce`
- Variabel penyiaran setelah batch pertama, seperti yang dipersyaratkan oleh Horovod
- Operasi pengocokan pembibitan dan/atau sharding dalam pipa data dengan. `smp.dp_rank()`

> **Note**
>
> Saat Anda menggunakan Horovod, Anda tidak boleh langsung memanggil skrip `hvd.init` pelatihan Anda. Sebagai gantinya, Anda harus mengatur `"horovod"` ke `True` dalam parameter SDK SageMaker `modelparallel` Python di. Langkah 2: Luncurkan Training Job Menggunakan SageMaker Python SDK Hal ini memungkinkan perpustakaan untuk menginisialisasi Horovod secara internal berdasarkan penetapan perangkat partisi model. Memanggil `hvd.init()` langsung dalam skrip pelatihan Anda dapat menyebabkan masalah.

> **Note**
>
> Menggunakan `hvd.DistributedOptimizer` API secara langsung di skrip pelatihan Anda dapat mengakibatkan kinerja dan kecepatan pelatihan yang buruk, karena API secara implisit menempatkan `AllReduce` operasi di dalamnya. `smp.step` Kami menyarankan Anda untuk menggunakan perpustakaan paralelisme model dengan Horovod dengan langsung memanggil `hvd.allreduce` setelah memanggil `accumulate()` atau `reduce_mean()` pada gradien yang dikembalikan dari`smp.step`, seperti yang akan ditunjukkan pada contoh berikut.

Untuk mempelajari lebih lanjut tentang API pustaka paralelisme model, lihat dokumentasi API. SageMaker

```
import tensorflow as tf
import horovod.tensorflow as hvd

# smdistributed: Import TF2.x API
import smdistributed.modelparallel.tensorflow as smp

# smdistributed: Initialize
smp.init()

# Download and load MNIST dataset.
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data(
    "MNIST-data-%d" % smp.rank()
)
x_train, x_test = x_train / 255.0, x_test / 255.0

# Add a channels dimension
x_train = x_train[..., tf.newaxis]
x_test = x_test[..., tf.newaxis]

# smdistributed: Seed the shuffle with smp.dp_rank(), and drop_remainder
# in batching to make sure batch size is always divisible by number of microbatches
train_ds = (
    tf.data.Dataset.from_tensor_slices((x_train, y_train))
    .shuffle(10000, seed=smp.dp_rank())
    .batch(256, drop_remainder=True)
)

# smdistributed: Define smp.DistributedModel the same way as Keras sub-classing API
class MyModel(smp.DistributedModel):
    def __init__(self):
        super(MyModel, self).__init__()
        # define layers

    def call(self, x, training=None):
        # define forward pass and return model outputs


model = MyModel()

loss_object = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

```
optimizer = tf.keras.optimizers.Adam()
train_accuracy = tf.keras.metrics.SparseCategoricalAccuracy(name="train_accuracy")

# smdistributed: Define smp.step. Return any tensors needed outside
@smp.step
def get_grads(images, labels):
    predictions = model(images, training=True)
    loss = loss_object(labels, predictions)

    grads = optimizer.get_gradients(loss, model.trainable_variables)
    return grads, loss, predictions


@tf.function
def train_step(images, labels, first_batch):
    gradients, loss, predictions = get_grads(images, labels)

    # smdistributed: Accumulate the gradients across microbatches
    # Horovod: AllReduce the accumulated gradients
    gradients = [hvd.allreduce(g.accumulate()) for g in gradients]
    optimizer.apply_gradients(zip(gradients, model.trainable_variables))

    # Horovod: Broadcast the variables after first batch
    if first_batch:
        hvd.broadcast_variables(model.variables, root_rank=0)
        hvd.broadcast_variables(optimizer.variables(), root_rank=0)

    # smdistributed: Merge predictions across microbatches
    train_accuracy(labels, predictions.merge())
    return loss.reduce_mean()


for epoch in range(5):
    # Reset the metrics at the start of the next epoch
    train_accuracy.reset_states()

    for batch, (images, labels) in enumerate(train_ds):
        loss = train_step(images, labels, tf.constant(batch == 0))
```

Pemisahan manual dengan TensorFlow

Gunakan manajer `smp.partition` konteks untuk menempatkan operasi di partisi tertentu. Setiap operasi yang tidak ditempatkan dalam `smp.partition` konteks apa pun ditempatkan di.

`default_partition` Untuk mempelajari lebih lanjut tentang API pustaka paralelisme model, lihat dokumentasi API. SageMaker

```
import tensorflow as tf

# smdistributed: Import TF2.x API.
import smdistributed.modelparallel.tensorflow as smp

# smdistributed: Initialize
smp.init()

# Download and load MNIST dataset.
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data(
    "MNIST-data-%d" % smp.rank()
)
x_train, x_test = x_train / 255.0, x_test / 255.0

# Add a channels dimension
x_train = x_train[..., tf.newaxis]
x_test = x_test[..., tf.newaxis]

# smdistributed: If needed, seed the shuffle with smp.dp_rank(), and drop_remainder
# in batching to make sure batch size is always divisible by number of microbatches.
train_ds = (
    tf.data.Dataset.from_tensor_slices((x_train, y_train))
    .shuffle(10000, seed=smp.dp_rank())
    .batch(256, drop_remainder=True)
)

# smdistributed: Define smp.DistributedModel the same way as Keras sub-classing API.
class MyModel(smp.DistributedModel):
    def __init__(self):
         # define layers

    def call(self, x):
        with smp.partition(0):
            x = self.layer0(x)
        with smp.partition(1):
            return self.layer1(x)


model = MyModel()
```

```
loss_object = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
optimizer = tf.keras.optimizers.Adam()
train_accuracy = tf.keras.metrics.SparseCategoricalAccuracy(name="train_accuracy")

# smdistributed: Define smp.step. Return any tensors needed outside
@smp.step
def get_grads(images, labels):
    predictions = model(images, training=True)
    loss = loss_object(labels, predictions)

    grads = optimizer.get_gradients(loss, model.trainable_variables)
    return grads, loss, predictions


@tf.function
def train_step(images, labels):
    gradients, loss, predictions = get_grads(images, labels)

    # smdistributed: Accumulate the gradients across microbatches
    gradients = [g.accumulate() for g in gradients]
    optimizer.apply_gradients(zip(gradients, model.trainable_variables))

    # smdistributed: Merge predictions and average losses across microbatches
    train_accuracy(labels, predictions.merge())
    return loss.reduce_mean()


for epoch in range(5):
    # Reset the metrics at the start of the next epoch
    train_accuracy.reset_states()
    for images, labels in train_ds:
        loss = train_step(images, labels)
    accuracy = train_accuracy.result()
```

Fitur kerangka kerja yang tidak didukung

TensorFlow Fitur-fitur berikut tidak didukung oleh pustaka:

- `tf.GradientTape()`saat ini tidak didukung. Anda dapat menggunakan `Optimizer.get_gradients()` atau `Optimizer.compute_gradients()` sebagai gantinya untuk menghitung gradien.
- `tf.train.Checkpoint.restore()`API saat ini tidak didukung. Untuk checkpointing, gunakan `smp.CheckpointManager` sebagai gantinya, yang menyediakan API dan fungsionalitas yang

sama. Perhatikan bahwa pemulihan pos pemeriksaan dengan `smp.CheckpointManager` harus dilakukan setelah langkah pertama.

Memodifikasi Skrip PyTorch Pelatihan

Di bagian ini, Anda mempelajari cara memodifikasi skrip PyTorch pelatihan untuk mengonfigurasi pustaka paralelisme SageMaker model untuk partisi otomatis dan partisi manual.

> **Note**
>
> Untuk menemukan PyTorch versi mana yang didukung oleh perpustakaan, lihatthe section called “Kerangka Kerja yang Didukung dan Wilayah AWS”.

> **Tip**
>
> Untuk contoh end-to-end buku catatan yang menunjukkan cara menggunakan skrip PyTorch pelatihan dengan pustaka paralelisme SageMaker model, lihat. PyTorchContoh

Perhatikan bahwa partisi otomatis diaktifkan secara default. Kecuali ditentukan lain, skrip berikut menggunakan partisi otomatis.

Topik

- Pemisahan otomatis dengan PyTorch
- Pemisahan manual dengan PyTorch
- Pertimbangan
- Fitur kerangka kerja yang tidak didukung

Pemisahan otomatis dengan PyTorch

Perubahan skrip pelatihan berikut diperlukan untuk menjalankan skrip PyTorch pelatihan dengan pustaka SageMaker paralelisme model:

1. Impor dan inisialisasi perpustakaan dengan `smdistributed.modelparallel.torch.init()`.
2. Bungkus model dengan `smdistributed.modelparallel.torch.DistributedModel`. Berhati-hatilah bahwa setiap tensor yang dikembalikan dari `forward` metode `nn.Module` objek

yang mendasarinya akan disiarkan di seluruh perangkat model-paralel, menimbulkan overhead komunikasi, jadi tensor apa pun yang tidak diperlukan di luar metode panggilan (seperti aktivasi perantara) tidak boleh dikembalikan.

> **Note**
>
> Untuk pelatihan FP16, Anda perlu menggunakan pengelola konteks smdistributed.modelparallel.torch.model_creation () untuk membungkus model. Untuk informasi selengkapnya, lihat Pelatihan FP16 dengan Paralelisme Model.

3. Bungkus pengoptimal dengan `smdistributed.modelparallel.torch.DistributedOptimizer`.

> **Note**
>
> Untuk pelatihan FP16, Anda perlu mengatur penskalaan kerugian statis atau dinamis. Untuk informasi selengkapnya, lihat Pelatihan FP16 dengan Paralelisme Model.

4. Gunakan `DistributedModel` objek yang dikembalikan alih-alih model pengguna.
5. Letakkan logika maju dan mundur dalam fungsi langkah dan hiasi dengan `smdistributed.modelparallel.torch.step`.
6. Batasi setiap proses ke perangkatnya sendiri melalui`torch.cuda.set_device(smp.local_rank())`.
7. Pindahkan tensor input ke GPU menggunakan `.to()` API sebelum `smp.step` panggilan (lihat contoh di bawah).
8. Ganti `torch.Tensor.backward` dan `torch.autograd.backward` dengan`DistributedModel.backward`.
9. Lakukan pasca-pemrosesan pada output di seluruh microbatch menggunakan `StepOutput`metode seperti. `reduce_mean`
10. Jika ada langkah evaluasi, tempatkan logika penerusan di dalam fungsi `smp.step` -decorated dan pasca-proses output menggunakan API. `StepOutput`
11. Ditetapkan `drop_last=True` di`DataLoader`. Atau, lewati batch secara manual dalam loop pelatihan jika ukuran batch tidak habis dibagi dengan jumlah microbatch.

Untuk mempelajari lebih lanjut tentang API pustaka paralelisme model, lihat dokumentasi API. SageMaker

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchnet.dataset import SplitDataset
from torchvision import datasets

import smdistributed.modelparallel.torch as smp

class GroupedNet(nn.Module):
    def __init__(self):
        super(GroupedNet, self).__init__()
        # define layers

    def forward(self, x):
        # define forward pass and return model outputs


# smdistributed: Define smp.step. Return any tensors needed outside.
@smp.step
def train_step(model, data, target):
    output = model(data)
    loss = F.nll_loss(output, target, reduction="mean")
    model.backward(loss)
    return output, loss


def train(model, device, train_loader, optimizer):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        # smdistributed: Move input tensors to the GPU ID used by the current process,
        # based on the set_device call.
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        # Return value, loss_mb is a StepOutput object
        _, loss_mb = train_step(model, data, target)

        # smdistributed: Average the loss across microbatches.
        loss = loss_mb.reduce_mean()

        optimizer.step()

# smdistributed: initialize the backend
```

```
smp.init()

# smdistributed: Set the device to the GPU ID used by the current process.
# Input tensors should be transferred to this device.
torch.cuda.set_device(smp.local_rank())
device = torch.device("cuda")

# smdistributed: Download only on a single process per instance.
# When this is not present, the file is corrupted by multiple processes trying
# to download and extract at the same time
dataset = datasets.MNIST("../data", train=True, download=False)

# smdistributed: Shard the dataset based on data-parallel ranks
if smp.dp_size() > 1:
    partitions_dict = {f"{i}": 1 / smp.dp_size() for i in range(smp.dp_size())}
    dataset = SplitDataset(dataset, partitions=partitions_dict)
    dataset.select(f"{smp.dp_rank()}")

# smdistributed: Set drop_last=True to ensure that batch size is always divisible
# by the number of microbatches
train_loader = torch.utils.data.DataLoader(dataset, batch_size=64, drop_last=True)

model = GroupedNet()
optimizer = optim.Adadelta(model.parameters(), lr=4.0)

# smdistributed: Use the DistributedModel container to provide the model
# to be partitioned across different ranks. For the rest of the script,
# the returned DistributedModel object should be used in place of
# the model provided for DistributedModel class instantiation.
model = smp.DistributedModel(model)
optimizer = smp.DistributedOptimizer(optimizer)

train(model, device, train_loader, optimizer)
```

Pemisahan manual dengan PyTorch

Gunakan manajer `smp.partition`konteks untuk menempatkan modul di perangkat tertentu. Modul apa pun yang tidak ditempatkan dalam `smp.partition` konteks apa pun ditempatkan di. `default_partition` `default_partition`Kebutuhan yang harus `auto_partition` disediakan jika diatur ke`False`. Modul yang dibuat dalam `smp.partition` konteks tertentu ditempatkan pada partisi yang sesuai.

Untuk mempelajari lebih lanjut tentang API pustaka paralelisme model, lihat dokumentasi API. SageMaker

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchnet.dataset import SplitDataset
from torchvision import datasets

import smdistributed.modelparallel.torch as smp

class GroupedNet(nn.Module):
    def __init__(self):
        super(GroupedNet, self).__init__()
        with smp.partition(0):
            # define child modules on device 0
        with smp.partition(1):
            # define child modules on device 1

    def forward(self, x):
        # define forward pass and return model outputs


# smdistributed: Define smp.step. Return any tensors needed outside.
@smp.step
def train_step(model, data, target):
    output = model(data)
    loss = F.nll_loss(output, target, reduction="mean")
    model.backward(loss)
    return output, loss


def train(model, device, train_loader, optimizer):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        # smdistributed: Move input tensors to the GPU ID used by the current process,
        # based on the set_device call.
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        # Return value, loss_mb is a StepOutput object
        _, loss_mb = train_step(model, data, target)
```

```
        # smdistributed: Average the loss across microbatches.
        loss = loss_mb.reduce_mean()

        optimizer.step()

# smdistributed: initialize the backend
smp.init()

# smdistributed: Set the device to the GPU ID used by the current process.
# Input tensors should be transferred to this device.
torch.cuda.set_device(smp.local_rank())
device = torch.device("cuda")

# smdistributed: Download only on a single process per instance.
# When this is not present, the file is corrupted by multiple processes trying
# to download and extract at the same time
dataset = datasets.MNIST("../data", train=True, download=False)

# smdistributed: Shard the dataset based on data-parallel ranks
if smp.dp_size() > 1:
    partitions_dict = {f"{i}": 1 / smp.dp_size() for i in range(smp.dp_size())}
    dataset = SplitDataset(dataset, partitions=partitions_dict)
    dataset.select(f"{smp.dp_rank()}")

# smdistributed: Set drop_last=True to ensure that batch size is always divisible
# by the number of microbatches
train_loader = torch.utils.data.DataLoader(dataset, batch_size=64, drop_last=True)

model = GroupedNet()
optimizer = optim.Adadelta(model.parameters(), lr=4.0)

# smdistributed: Use the DistributedModel container to provide the model
# to be partitioned across different ranks. For the rest of the script,
# the returned DistributedModel object should be used in place of
# the model provided for DistributedModel class instantiation.
model = smp.DistributedModel(model)
optimizer = smp.DistributedOptimizer(optimizer)

train(model, device, train_loader, optimizer)
```

Pertimbangan

Saat Anda mengonfigurasi skrip PyTorch pelatihan menggunakan SageMaker pustaka paralelisme model, Anda harus mengetahui hal berikut:

- Jika Anda menggunakan teknik optimasi yang bergantung pada norma gradien global, misalnya norma gradien dari seluruh model, seperti beberapa varian pengoptimal LAMB atau kliping gradien global, Anda perlu mengumpulkan semua norma di seluruh partisi model untuk kebenaran. Anda dapat menggunakan tipe data dasar komunikasi perpustakaan untuk melakukan hal ini.
- Semua `torch.Tensor` argumen untuk metode penerusan `nn.Modules` dalam model Anda harus digunakan dalam perhitungan output modul. Dengan kata lain, perpustakaan tidak mendukung kasus di mana ada `torch.Tensor` argumen ke modul di mana output modul tidak bergantung.
- Argumen untuk `smp.DistributedModel.backward()` panggilan harus bergantung pada semua output model. Dengan kata lain, tidak mungkin ada output dari `smp.DistributedModel.forward` panggilan yang tidak digunakan dalam perhitungan tensor yang dimasukkan ke dalam panggilan. `smp.DistributedModel.backward`
- Jika ada `torch.cuda.synchronize()` panggilan dalam kode Anda, Anda mungkin perlu menelepon `torch.cuda.set_device(smp.local_rank())` segera sebelum panggilan sinkronisasi. Jika tidak, konteks CUDA yang tidak perlu dapat dibuat di perangkat 0, yang akan menghabiskan memori dengan sia-sia.
- Karena perpustakaan ditempatkan `nn.Modules` pada perangkat yang berbeda, modul dalam model tidak boleh bergantung pada keadaan global apa pun yang dimodifikasi di dalamnya`smp.step`. Setiap keadaan yang tetap selama pelatihan, atau yang dimodifikasi `smp.step` di luar dengan cara yang terlihat oleh semua proses, diperbolehkan.
- Anda tidak perlu memindahkan model ke GPU (misalnya, menggunakan`model.to(device)`) saat menggunakan perpustakaan. Jika Anda mencoba memindahkan model ke GPU sebelum model dipartisi (sebelum `smp.step` panggilan pertama), panggilan pindah akan diabaikan. Pustaka secara otomatis memindahkan bagian model yang ditetapkan ke peringkat ke GPU-nya. Setelah pelatihan dengan perpustakaan dimulai, jangan pindahkan model ke CPU dan gunakan, karena tidak akan memiliki parameter yang benar untuk modul yang tidak ditetapkan ke partisi yang dipegang oleh proses. Jika Anda ingin melatih ulang model atau menggunakannya untuk inferensi tanpa perpustakaan setelah dilatih menggunakan pustaka paralelisme model, cara yang disarankan adalah menyimpan model lengkap menggunakan API pos pemeriksaan kami dan memuatnya kembali ke Modul biasa. PyTorch
- Jika Anda memiliki daftar modul sehingga output dari satu umpan ke yang lain, mengganti daftar itu dengan `nn.Sequential` dapat secara signifikan meningkatkan kinerja.

- Pembaruan bobot (`optimizer.step()`) perlu terjadi di luar `smp.step` karena saat itulah seluruh backward pass selesai dan gradien sudah siap. Saat menggunakan model hybrid dengan model dan paralelisme data, pada titik ini, AllReduce gradien juga dijamin selesai.
- Saat menggunakan library dalam kombinasi dengan paralelisme data, pastikan jumlah batch pada semua data parallel rank sama sehingga AllReduce tidak hang menunggu rank yang tidak berpartisipasi dalam langkah tersebut.
- Jika Anda meluncurkan pekerjaan pelatihan menggunakan jenis instans ml.p4d (seperti ml.p4d.24xlarge), Anda harus menyetel variabel pemuat data. `num_workers=0` Misalnya, Anda dapat mendefinisikan `DataLoader` sebagai berikut:

```
dataloader = torch.utils.data.DataLoader(
            data,
            batch_size=batch_size,
            num_workers=0,
            pin_memory=True,
            drop_last=True,
            shuffle=shuffle,
        )
```

- Input `smp.step` harus menjadi input model yang dihasilkan oleh. `DataLoader` Ini karena `smp.step` secara internal membagi tensor input di sepanjang dimensi batch dan menyalurkannya. Ini berarti bahwa meneruskan `DataLoader` dirinya ke `smp.step` fungsi untuk menghasilkan input model di dalamnya tidak berfungsi.

  Misalnya, jika Anda mendefinisikan `DataLoader` sebagai berikut:

```
train_loader = torch.utils.data.DataLoader(dataset, batch_size=64, drop_last=True)
```

  Anda harus mengakses input model yang dihasilkan oleh `train_loader` dan meneruskannya ke fungsi yang `smp.step` didekorasi. Jangan `train_loader` langsung lolos ke `smp.step` fungsi.

```
def train(model, device, train_loader, optimizer):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        ...
        _, loss_mb = train_step(model, data, target)
        ...

@smp.step
def train_step(model, data, target):
```

```
    ...
    return output, loss
```

- Tensor input `smp.step` harus dipindahkan ke perangkat saat ini menggunakan `.to()` API, yang harus dilakukan setelah panggilan. `torch.cuda.set_device(local_rank())`

  Misalnya, Anda dapat mendefinisikan `train` fungsi sebagai berikut. Fungsi ini menambahkan `data` dan `target` ke perangkat saat ini menggunakan `.to()` API sebelum menggunakan tensor input tersebut untuk memanggil. `train_step`

```
def train(model, device, train_loader, optimizer):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        # smdistributed: Move input tensors to the GPU ID used by the current
 process,
        # based on the set_device call.
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        # Return value, loss_mb is a StepOutput object
        _, loss_mb = train_step(model, data, target)

        # smdistributed: Average the loss across microbatches.
        loss = loss_mb.reduce_mean()

        optimizer.step()
```

  Tensor input ke fungsi yang `smp.set` didekorasi ini telah dipindahkan ke perangkat saat ini dalam `train` fungsi di atas. Model tidak perlu dipindahkan ke perangkat saat ini. Pustaka secara otomatis memindahkan bagian model yang ditetapkan ke peringkat ke GPU-nya.

```
@smp.step
def train_step(model, data, target):
    output = model(data)
    loss = F.nll_loss(output, target, reduction="mean")
    model.backward(loss)
    return output, loss
```

Fitur kerangka kerja yang tidak didukung

PyTorch Fitur-fitur berikut tidak didukung oleh pustaka SageMaker paralelisme model:

- Jika Anda menggunakan paralelisme data dengan PyTorch DDP asli, modul `torch.nn.parallel.DistributedDataParallel`pembungkus tidak didukung oleh perpustakaan. Pustaka secara internal mengelola integrasi dengan PyTorch DDP, termasuk siaran parameter dan gradien. AllReduce Saat menggunakan perpustakaan, buffer modul hanya disiarkan sekali pada awal pelatihan. Jika model Anda memiliki buffer modul yang perlu disinkronkan di seluruh grup paralel data pada setiap langkah, Anda dapat melakukannya melalui `torch.distributed` API, menggunakan grup proses yang dapat diperoleh melalui. `smp.get_dp_process_group()`
- Untuk pelatihan presisi campuran, `apex.amp` modul tidak didukung. Cara yang disarankan untuk menggunakan perpustakaan dengan presisi campuran otomatis adalah dengan menggunakan`torch.cuda.amp`, dengan pengecualian menggunakan `smp.amp.GradScaler` alih-alih implementasi dalam obor.
- `torch.jit.ScriptModules`atau `ScriptFunctions` tidak didukung oleh`smp.DistributedModel`.
- `apex`:`FusedLayerNorm`,`FusedAdam`,`FusedLAMB`, dan `FusedNovoGrad` dari `apex` tidak didukung. Anda dapat menggunakan implementasi pustaka ini melalui `smp.optimizers` dan `smp.nn` API sebagai gantinya.

Langkah 2: Luncurkan Training Job Menggunakan SageMaker Python SDK

SageMaker Python SDK mendukung pelatihan terkelola model dengan kerangka kerja ML seperti dan. TensorFlow PyTorch Untuk meluncurkan pekerjaan pelatihan menggunakan salah satu kerangka kerja ini, Anda mendefinisikan estimator, SageMaker TensorFlow estimator, atau SageMaker PyTorch Estimator SageMaker generik untuk menggunakan skrip pelatihan yang dimodifikasi dan konfigurasi paralelisme model.

Topik

- Menggunakan SageMaker TensorFlow dan PyTorch Estimator
- Perluas Container Docker Pra-built yang Berisi Perpustakaan Paralel Model SageMaker Terdistribusi
- Buat Container Docker Anda Sendiri dengan Perpustakaan Paralel Model SageMaker Terdistribusi

Menggunakan SageMaker TensorFlow dan PyTorch Estimator

Kelas TensorFlow dan PyTorch estimator berisi `distribution` parameter, yang dapat Anda gunakan untuk menentukan parameter konfigurasi untuk menggunakan kerangka pelatihan

terdistribusi. Pustaka paralel SageMaker model secara internal menggunakan MPI untuk data hibrida dan paralelisme model, jadi Anda harus menggunakan opsi MPI dengan perpustakaan.

Template berikut dari TensorFlow atau PyTorch estimator menunjukkan cara mengkonfigurasi `distribution` parameter untuk menggunakan SageMaker model parallel library dengan MPI.

Using the SageMaker TensorFlow estimator

```
import sagemaker
from sagemaker.tensorflow import TensorFlow

smp_options = {
    "enabled":True,              # Required
    "parameters": {
        "partitions": 2,         # Required
        "microbatches": 4,
        "placement_strategy": "spread",
        "pipeline": "interleaved",
        "optimize": "speed",
        "horovod": True,         # Use this for hybrid model and data parallelism
    }
}

mpi_options = {
    "enabled" : True,            # Required
    "processes_per_host" : 8,    # Required
    # "custom_mpi_options" : "--mca btl_vader_single_copy_mechanism none"
}

smd_mp_estimator = TensorFlow(
    entry_point="your_training_script.py", # Specify your train script
    source_dir="location_to_your_script",
    role=sagemaker.get_execution_role(),
    instance_count=1,
    instance_type='ml.p3.16xlarge',
    framework_version='2.6.3',
    py_version='py38',
    distribution={
        "smdistributed": {"modelparallel": smp_options},
        "mpi": mpi_options
    },
    base_job_name="SMD-MP-demo",
)
```

```
smd_mp_estimator.fit('s3://my_bucket/my_training_data/')
```

Using the SageMaker PyTorch estimator

```
import sagemaker
from sagemaker.pytorch import PyTorch

smp_options = {
    "enabled":True,
    "parameters": {                        # Required
        "pipeline_parallel_degree": 2,     # Required
        "microbatches": 4,
        "placement_strategy": "spread",
        "pipeline": "interleaved",
        "optimize": "speed",
        "ddp": True,
    }
}

mpi_options = {
    "enabled" : True,                      # Required
    "processes_per_host" : 8,              # Required
    # "custom_mpi_options" : "--mca btl_vader_single_copy_mechanism none"
}

smd_mp_estimator = PyTorch(
    entry_point="your_training_script.py", # Specify your train script
    source_dir="location_to_your_script",
    role=sagemaker.get_execution_role(),
    instance_count=1,
    instance_type='ml.p3.16xlarge',
    framework_version='1.13.1',
    py_version='py38',
    distribution={
        "smdistributed": {"modelparallel": smp_options},
        "mpi": mpi_options
    },
    base_job_name="SMD-MP-demo",
)

smd_mp_estimator.fit('s3://my_bucket/my_training_data/')
```

Untuk mengaktifkan pustaka, Anda harus meneruskan kamus konfigurasi ke `"mpi"` kunci `"smdistributed"` dan melalui `distribution` argumen konstruktor SageMaker estimator.

Parameter konfigurasi untuk SageMaker paralelisme model

- Untuk `"smdistributed"` kuncinya, berikan kamus dengan `"modelparallel"` kunci dan kamus batin berikut.

> **Note**
>
> Menggunakan `"modelparallel"` dan `"dataparallel"` dalam satu pekerjaan pelatihan tidak didukung.

  - `"enabled"` – Wajib diisi. Untuk mengaktifkan paralelisme model, atur. `"enabled": True`
  - `"parameters"` – Wajib diisi. Tentukan satu set parameter untuk SageMaker paralelisme model.
    - Untuk daftar lengkap parameter umum, lihat Parameter untuk `smdistributed` dalam dokumentasi SageMaker Python SDK.

      Untuk TensorFlow, lihat TensorFlow-Parameter spesifik.

      Untuk PyTorch, lihat PyTorch-Parameter spesifik.
    - `"pipeline_parallel_degree"`(atau `"partitions"` dismdistributed-modelparallel<`v1.6.0`) - Diperlukan. Di antara parameter untuk `smdistributed`, parameter ini diperlukan untuk menentukan berapa banyak partisi model yang ingin Anda bagi.

> **Important**
>
> Ada perubahan besar pada nama parameter. `"pipeline_parallel_degree"`Parameter menggantikan `"partitions"` sejak `smdistributed-modelparallel` v1.6.0. Untuk informasi selengkapnya, lihat Parameter Umum untuk konfigurasi paralelisme SageMaker model dan Catatan Rilis Paralel Model SageMaker Terdistribusi dalam dokumentasi SageMaker Python SDK.

- Untuk `"mpi"` kuncinya, berikan kamus yang berisi berikut ini:
  - `"enabled"` – Wajib diisi. Ditetapkan `True` untuk meluncurkan pekerjaan pelatihan terdistribusi dengan MPI.

- `"processes_per_host"` – Wajib diisi. Tentukan jumlah proses yang harus diluncurkan MPI pada setiap host. Di SageMaker, host adalah satu instans Amazon EC2 ML. SageMaker Python SDK mempertahankan one-to-one pemetaan antara proses dan GPU di seluruh model dan paralelisme data. Ini berarti bahwa SageMaker menjadwalkan setiap proses pada satu GPU terpisah dan tidak ada GPU yang berisi lebih dari satu proses. Jika Anda menggunakan PyTorch, Anda harus membatasi setiap proses ke perangkatnya sendiri. `torch.cuda.set_device(smp.local_rank())` Untuk mempelajari selengkapnya, lihat Pemisahan otomatis dengan PyTorch.

  > **Important**
  >
  > `process_per_host`tidak boleh lebih besar dari jumlah GPU per instance dan biasanya akan sama dengan jumlah GPU per instance.

- `"custom_mpi_options"`(opsional) - Gunakan kunci ini untuk meneruskan opsi MPI khusus yang mungkin Anda perlukan. Jika Anda tidak meneruskan opsi kustom MPI ke kunci, opsi MPI diatur secara default ke bendera berikut.

  ```
  --mca btl_vader_single_copy_mechanism none
  ```

  > **Note**
  >
  > Anda tidak perlu secara eksplisit menentukan bendera default ini ke kunci. Jika Anda secara eksplisit menentukannya, pekerjaan pelatihan paralel model terdistribusi Anda mungkin gagal dengan kesalahan berikut:
  >
  > ```
  > The following MCA parameter has been listed multiple times on the command
   line:
  > MCA param: btl_vader_single_copy_mechanism MCA parameters can only be listed
   once
  > on a command line to ensure there is no ambiguity as to its value.
  > Please correct the situation and try again.
  > ```

> **Tip**
>
> Jika Anda meluncurkan tugas pelatihan menggunakan jenis instans berkemampuan EFA, seperti `ml.p4d.24xlarge` dan`ml.p3dn.24xlarge`, gunakan flag berikut untuk performa terbaik:
>
> ```
> -x FI_EFA_USE_DEVICE_RDMA=1 -x FI_PROVIDER=efa -x RDMAV_FORK_SAFE=1
> ```

Untuk meluncurkan pekerjaan pelatihan menggunakan estimator dan skrip pelatihan yang dikonfigurasi paralel SageMaker model Anda, jalankan `estimator.fit()` fungsinya.

Gunakan sumber daya berikut untuk mempelajari selengkapnya tentang penggunaan fitur paralelisme model di Python SageMaker SDK:

- Gunakan TensorFlow dengan SageMaker Python SDK
- Gunakan PyTorch dengan SageMaker Python SDK

- Kami menyarankan Anda menggunakan instance SageMaker notebook jika Anda adalah pengguna baru. Untuk melihat contoh bagaimana Anda dapat meluncurkan pekerjaan pelatihan menggunakan instance SageMaker notebook, lihatContoh Notebook Pelatihan SageMaker Terdistribusi Amazon.
- Anda juga dapat mengirimkan pekerjaan pelatihan terdistribusi dari mesin Anda menggunakanAWS CLI. Untuk mengatur AWS CLI di mesin Anda, lihat mengatur AWS kredensyal Anda dan Wilayah untuk pengembangan.

Perluas Container Docker Pra-built yang Berisi Perpustakaan Paralel Model SageMaker Terdistribusi

Untuk memperluas wadah pra-bangun dan menggunakan SageMaker pustaka paralelisme model, Anda harus menggunakan salah satu gambar AWS Deep Learning Containers (DLC) yang tersedia untuk atau. PyTorch TensorFlow Pustaka paralelisme SageMaker model termasuk dalam gambar DLC TensorFlow (2.3.0 dan yang lebih baru) dan PyTorch (1.6.0 dan yang lebih baru) dengan CUDA (). `cuxyz` Untuk daftar lengkap gambar DLC, lihat Gambar Deep Learning Containers yang Tersedia di repositori AWSDeep Learning Containers GitHub .

> **Tip**
>
> Kami menyarankan Anda menggunakan gambar yang berisi versi terbaru TensorFlow atau PyTorch untuk mengakses sebagian besar up-to-date versi pustaka paralelisme SageMaker model.

Misalnya, Dockerfile Anda harus berisi `FROM` pernyataan yang mirip dengan berikut ini:

```
# Use the SageMaker DLC image URI for TensorFlow or PyTorch
FROM aws-dlc-account-id.dkr.ecr.aws-region.amazonaws.com/framework-training:{framework-version-tag}

# Add your dependencies here
RUN ...

ENV PATH="/opt/ml/code:${PATH}"

# this environment variable is used by the SageMaker container to determine our user
 code directory.
ENV SAGEMAKER_SUBMIT_DIRECTORY /opt/ml/code
```

Selain itu, ketika Anda mendefinisikan PyTorch atau TensorFlow estimator, Anda harus menentukan bahwa `entry_point` untuk skrip pelatihan Anda. Ini harus menjadi jalur yang sama yang diidentifikasi `ENV SAGEMAKER_SUBMIT_DIRECTORY` di Dockerfile Anda.

> **Tip**
>
> Anda harus mendorong wadah Docker ini ke Amazon Elastic Container Registry (Amazon ECR) dan menggunakan URI image (`image_uri`) untuk menentukan estimator untuk pelatihan. SageMaker Untuk informasi selengkapnya, lihat Memperluas Kontainer Pra-dibangun.

Setelah Anda selesai menghosting wadah Docker dan mengambil URI gambar wadah, buat objek SageMaker `PyTorch` estimator sebagai berikut. Contoh ini mengasumsikan bahwa Anda telah mendefinisikan `smp_options` dan`mpi_options`.

```
smd_mp_estimator = Estimator(
    entry_point="your_training_script.py",
```

```
    role=sagemaker.get_execution_role(),
    instance_type='ml.p3.16xlarge',
    sagemaker_session=sagemaker_session,
    image_uri='your_aws_account_id.dkr.ecr.region.amazonaws.com/name:tag'
    instance_count=1,
    distribution={
        "smdistributed": smp_options,
        "mpi": mpi_options
    },
    base_job_name="SMD-MP-demo",
)

smd_mp_estimator.fit('s3://my_bucket/my_training_data/')
```

Buat Container Docker Anda Sendiri dengan Perpustakaan Paralel Model SageMaker Terdistribusi

Untuk membangun wadah Docker Anda sendiri untuk pelatihan dan menggunakan pustaka paralel SageMaker model, Anda harus menyertakan dependensi yang benar dan file biner dari pustaka SageMaker paralel terdistribusi di Dockerfile Anda. Bagian ini menyediakan kumpulan blok kode minimum yang harus Anda sertakan untuk mempersiapkan lingkungan SageMaker pelatihan dengan benar dan pustaka paralel model di wadah Docker Anda sendiri.

> **Note**
>
> Opsi Docker khusus ini dengan pustaka paralel SageMaker model sebagai biner hanya tersedia untuk PyTorch.

Untuk membuat Dockerfile dengan toolkit SageMaker pelatihan dan perpustakaan paralel model

1. Mulailah dengan salah satu gambar dasar NVIDIA CUDA.

   ```
   FROM <cuda-cudnn-base-image>
   ```

   > **Tip**
   >
   > Gambar AWS Deep Learning Container (DLC) resmi dibuat dari gambar dasar NVIDIA CUDA. Kami sarankan Anda melihat ke Dockerfiles resmi dari AWS Deep Learning Container PyTorch untuk menemukan versi pustaka mana yang perlu Anda instal dan cara mengonfigurasinya. Dockerfiles resmi lengkap, benchmark diuji, dan dikelola oleh

tim layanan Deep Learning Container SageMaker dan Deep Learning. Di tautan yang disediakan, pilih PyTorch versi yang Anda gunakan, pilih folder CUDA (`cuxyz`), dan pilih Dockerfile yang diakhiri dengan atau. `.gpu` `.sagemaker.gpu`

2. Untuk mengatur lingkungan pelatihan terdistribusi, Anda perlu menginstal perangkat lunak untuk perangkat komunikasi dan jaringan, seperti Elastic Fabric Adapter (EFA), NVIDIA Collective Communications Library (NCCL), dan Open MPI. Bergantung pada versi PyTorch dan CUDA yang Anda pilih, Anda harus menginstal versi pustaka yang kompatibel.

   > **Important**
   >
   > Karena pustaka paralel SageMaker model memerlukan pustaka paralel SageMaker data pada langkah berikutnya, kami sangat menyarankan Anda mengikuti instruksi di Buat wadah Docker Anda sendiri dengan pustaka paralel data SageMaker terdistribusi untuk mengatur lingkungan SageMaker pelatihan dengan benar untuk pelatihan terdistribusi.

   Untuk informasi lebih lanjut tentang pengaturan EFA dengan NCCL dan Open MPI, lihat Memulai dengan EFA dan MPI dan Memulai dengan EFA dan NCCL.

3. Tambahkan argumen berikut untuk menentukan URL paket pelatihan SageMaker terdistribusi untuk PyTorch. Pustaka paralel SageMaker model membutuhkan pustaka paralel SageMaker data untuk menggunakan cross-node Remote Direct Memory Access (RDMA).

```
ARG SMD_MODEL_PARALLEL_URL=https://sagemaker-distributed-model-parallel.s3.us-
west-2.amazonaws.com/pytorch-1.10.0/build-artifacts/2022-02-21-19-26/
smdistributed_modelparallel-1.7.0-cp38-cp38-linux_x86_64.whl
ARG SMDATAPARALLEL_BINARY=https://smdataparallel.s3.amazonaws.com/binary/
pytorch/1.10.2/cu113/2022-02-18/smdistributed_dataparallel-1.4.0-cp38-cp38-
linux_x86_64.whl
```

4. Instal dependensi yang dibutuhkan oleh pustaka SageMaker paralel model.

   a. Instal perpustakaan METIS.

```
ARG METIS=metis-5.1.0

RUN rm /etc/apt/sources.list.d/* \
  && wget -nv http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/${METIS}.tar.gz \
  && gunzip -f ${METIS}.tar.gz \
  && tar -xvf ${METIS}.tar \
```

```
  && cd ${METIS} \
  && apt-get update \
  && make config shared=1 \
  && make install \
  && cd .. \
  && rm -rf ${METIS}.tar* \
  && rm -rf ${METIS} \
  && rm -rf /var/lib/apt/lists/* \
  && apt-get clean
```

b. Instal perpustakaan RAPIDS Memory Manager. Ini membutuhkan CMake 3.14 atau yang lebih baru.

```
ARG RMM_VERSION=0.15.0

RUN  wget -nv https://github.com/rapidsai/rmm/archive/v${RMM_VERSION}.tar.gz \
  && tar -xvf v${RMM_VERSION}.tar.gz \
  && cd rmm-${RMM_VERSION} \
  && INSTALL_PREFIX=/usr/local ./build.sh librmm \
  && cd .. \
  && rm -rf v${RMM_VERSION}.tar* \
  && rm -rf rmm-${RMM_VERSION}
```

5. Instal perpustakaan paralel SageMaker model.

```
RUN pip install --no-cache-dir -U ${SMD_MODEL_PARALLEL_URL}
```

6. Instal perpustakaan paralel SageMaker data.

```
RUN SMDATAPARALLEL_PT=1 pip install --no-cache-dir ${SMDATAPARALLEL_BINARY}
```

7. Instal toolkit pelatihan sagemaker. Toolkit berisi fungsionalitas umum yang diperlukan untuk membuat wadah yang kompatibel dengan platform SageMaker pelatihan dan SageMaker Python SDK.

```
RUN pip install sagemaker-training
```

8. Setelah Anda selesai membuat Dockerfile, lihat Mengadaptasi Wadah Pelatihan Anda Sendiri untuk mempelajari cara membuat wadah Docker dan menghostingnya di Amazon ECR.

> **Tip**
>
> Untuk informasi lebih umum tentang membuat Dockerfile kustom untuk pelatihan SageMaker, lihat Menggunakan Algoritma Pelatihan Anda Sendiri.

Checkpointing dan Fine-Tuning Model dengan Paralelisme Model

Pustaka paralelisme SageMaker model menyediakan API pos pemeriksaan untuk menyimpan status model dan status pengoptimal yang dibagi oleh berbagai strategi paralelisme model, dan untuk memuat pos pemeriksaan untuk pelatihan berkelanjutan dari tempat Anda ingin memulai ulang pelatihan dan menyempurnakan. API juga mendukung opsi untuk menyimpan status model dan pengoptimal sebagian atau seluruhnya.

Topik

- Checkpointing model terdistribusi
- Menyetel model terdistribusi

Checkpointing model terdistribusi

Pilih salah satu topik berikut tergantung pada kerangka kerja antara PyTorch dan TensorFlow dan versi pustaka paralelisme SageMaker model yang Anda gunakan.

Topik

- Checkpointing PyTorch model terdistribusi (untuk pustaka paralelisme SageMaker model v1.10.0 dan yang lebih baru)
- Checkpointing PyTorch model terdistribusi (untuk pustaka paralelisme SageMaker model antara v1.6.0 dan v1.9.0)
- Checkpointing model terdistribusi TensorFlow

Checkpointing PyTorch model terdistribusi (untuk pustaka paralelisme SageMaker model v1.10.0 dan yang lebih baru)

Pustaka paralelisme SageMaker model menyediakan API pos pemeriksaan untuk menyimpan dan memuat pos pemeriksaan penuh atau sebagian dari status model terdistribusi dan status pengoptimalnya.

> **Note**
>
> Metode checkpointing ini direkomendasikan jika Anda menggunakan PyTorch dan pustaka paralelisme SageMaker model v1.10.0 atau yang lebih baru.

Pos pemeriksaan sebagian

Untuk menyimpan pos pemeriksaan model yang dilatih dengan paralelisme model, gunakan `smdistributed.modelparallel.torch.save_checkpoint`API dengan opsi checkpointing sebagian yang disetel ke true (). `partial=True` Ini menghemat setiap partisi model satu per satu. Selain model dan status pengoptimal, Anda juga dapat menyimpan data kustom tambahan melalui `user_content` argumen. Model checkpoint, optimizer, dan konten pengguna disimpan sebagai file terpisah. Panggilan `save_checkpoint` API membuat folder pos pemeriksaan dalam struktur berikut.

```
- path
  - ${tag}_partial (folder for partial checkpoints)
    - model_rankinfo.pt
    - optimizer_rankinfo.pt
    - fp16_states_rankinfo.pt
    - user_content.pt
  - $tag (checkpoint file for full checkpoints)
  - user_content_$tag (user_content file for full checkpoints)
  - newest (a file that indicates the newest checkpoint)
```

Untuk melanjutkan pelatihan dari pos pemeriksaan sebagian, gunakan `smdistributed.modelparallel.torch.resume_from_checkpoint`API dengan`partial=True`, dan tentukan direktori pos pemeriksaan dan tag yang digunakan saat menyimpan pos pemeriksaan sebagian. Perhatikan bahwa pemuatan bobot model yang sebenarnya terjadi setelah partisi model, selama menjalankan pertama fungsi langkah pelatihan `smdistributed.modelparallel.torch.step` yang didekorasi.

Saat menyimpan pos pemeriksaan sebagian, perpustakaan juga menyimpan keputusan partisi model sebagai file dengan ekstensi `.pt` file. Sebaliknya, ketika melanjutkan dari pos pemeriksaan sebagian, perpustakaan memuat file keputusan partisi bersama-sama. Setelah keputusan partisi dimuat, Anda tidak dapat mengubah partisi.

Cuplikan kode berikut menunjukkan cara mengatur API pos pemeriksaan dalam skrip pelatihan. PyTorch

```
import smdistributed.modelparallel.torch as smp

model = ...
model = smp.DistributedModel(model)
optimizer = ...
optimizer = smp.DistributedOptimizer(optimizer)
user_content = ...     # additional custom data
checkpoint_path = "/opt/ml/checkpoint/model_parallel"

# Save a checkpoint.
smp.save_checkpoint(
    path=checkpoint_path,
    tag=f"total_steps{total_steps}",
    partial=True,
    model=model,
    optimizer=optimizer,
    user_content=user_content
    num_kept_partial_checkpoints=5
)

# Load a checkpoint.
# This automatically loads the most recently saved checkpoint.
smp_checkpoint = smp.resume_from_checkpoint(
    path=checkpoint_path,
    partial=True
)
```

Checkpointing penuh

Untuk menyimpan artefak model akhir untuk tujuan inferensi, gunakan `smdistributed.modelparallel.torch.save_checkpoint` API dengan`partial=False`, yang menggabungkan partisi model untuk membuat artefak model tunggal. Perhatikan bahwa ini tidak menggabungkan status pengoptimal.

Untuk menginisialisasi pelatihan dengan bobot tertentu, dengan pos pemeriksaan model lengkap, Anda dapat menggunakan API dengan`smdistributed.modelparallel.torch.resume_from_checkpoint`. `partial=False` Perhatikan bahwa ini tidak memuat status pengoptimal.

> **Note**
>
> Dengan paralelisme tensor, secara umum, `state_dict` harus diterjemahkan antara implementasi model asli dan implementasi. `DistributedModel` Secara opsional, Anda dapat memberikan fungsi `state_dict` terjemahan sebagai argumen untuk. `smdistributed.modelparallel.torch.resume_from_checkpoint` Namun, untukthe section called “Model yang Didukung Di Luar Kotak”, perpustakaan menangani terjemahan ini secara otomatis.

Kode berikut menunjukkan contoh cara menggunakan API pos pemeriksaan untuk memeriksa sepenuhnya model yang dilatih dengan PyTorch paralelisme model.

```
import smdistributed.modelparallel.torch as smp

model = ...
model = smp.DistributedModel(model)
optimizer = ...
optimizer = smp.DistributedOptimizer(optimizer)
user_content = ...     # additional custom data
checkpoint_path = "/opt/ml/checkpoint/model_parallel"

# Save a checkpoint.
smp.save_checkpoint(
    path=checkpoint_path,
    tag=f"total_steps{total_steps}",
    partial=False,
    model=model,
    optimizer=optimizer,
    user_content=user_content
    num_kept_partial_checkpoints=5
)

# Load a checkpoint.
# This automatically loads the most recently saved checkpoint.
smp_checkpoint = smp.resume_from_checkpoint(
    path=checkpoint_path,
    partial=False
)
```

Checkpointing PyTorch model terdistribusi (untuk pustaka paralelisme SageMaker model antara v1.6.0 dan v1.9.0)

Pustaka paralelisme SageMaker model menyediakan fungsi Python untuk menyimpan pos pemeriksaan sebagian atau penuh untuk pekerjaan pelatihan dengan paralelisme tensor. Prosedur berikut menunjukkan cara menggunakan `smp.save()`dan menyimpan dan `smp.load()`memuat pos pemeriksaan saat Anda menggunakan paralelisme tensor.

> **Note**
>
> Metode checkpointing ini direkomendasikan jika Anda menggunakan PyTorch,the section called “Paralelisme Tensor”, dan pustaka paralelisme SageMaker model antara v1.6.0 dan v1.9.0.

1. Siapkan objek model dan bungkus dengan fungsi `smp.DistributedModel()` pembungkus perpustakaan.

   ```
   model = MyModel(...)
   model = smp.DistributedModel(model)
   ```

2. Siapkan pengoptimal untuk model. Satu set parameter model adalah argumen iterable yang diperlukan oleh fungsi pengoptimal. Untuk menyiapkan satu set parameter model, Anda harus memproses `model.parameters()` untuk menetapkan ID unik ke parameter model individual.

   Jika ada parameter dengan ID duplikat dalam parameter model yang dapat diulang, memuat status pengoptimal checkpoint gagal. Untuk membuat parameter model yang dapat diulang dengan ID unik untuk pengoptimal Anda, lihat yang berikut ini:

   ```
   unique_params = []
   unique_params_set = set()
   for p in model.parameters():
     if p not in unique_params_set:
       unique_params.append(p)
       unique_params_set.add(p)
   del unique_params_set

   optimizer = MyOpt(unique_params, ...)
   ```

3. Bungkus pengoptimal menggunakan fungsi pembungkus perpustakaan. `smp.DistributedOptimizer()`

```
optimizer = smp.DistributedOptimizer(optimizer)
```

4. Simpan model dan status pengoptimal menggunakan `smp.save()`. Bergantung pada bagaimana Anda ingin menyimpan pos pemeriksaan, pilih salah satu dari dua opsi berikut:

   - Opsi 1: Simpan sebagian model pada masing-masing `mp_rank` untuk satu`MP_GROUP`.

```
model_dict = model.local_state_dict() # save a partial model
opt_dict = optimizer.local_state_dict() # save a partial optimizer state
# Save the dictionaries at rdp_rank 0 as a checkpoint
if smp.rdp_rank() == 0:
    smp.save(
        {"model_state_dict": model_dict, "optimizer_state_dict": opt_dict},
        f"/checkpoint.pt",
        partial=True,
    )
```

   Dengan paralelisme tensor, perpustakaan menyimpan file yang ditunjuk periksa yang dinamai dalam format berikut:. `checkpoint.pt_{pp_rank}_{tp_rank}`

> **Note**
>
> Dengan paralelisme tensor, pastikan Anda mengatur pernyataan if sebagai `if smp.rdp_rank() == 0` pengganti. `if smp.dp_rank() == 0` Saat status pengoptimal dibagi dengan paralelisme tensor, semua peringkat paralel data tereduksi harus menyimpan partisi mereka sendiri dari status pengoptimal. Menggunakan pernyataan if yang salah untuk checkpointing dapat mengakibatkan pekerjaan pelatihan yang terhenti. Untuk informasi selengkapnya tentang penggunaan paralelisme *`if smp.dp_rank() == 0`* tanpa tensor, lihat Instruksi Umum untuk Menyimpan dan Memuat dalam dokumentasi Python SageMaker SDK.

   - Opsi 2: Simpan model lengkap.

```
if smp.rdp_rank() == 0:
    model_dict = model.state_dict(gather_to_rank0=True) # save the full model
    if smp.rank() == 0:
        smp.save(
```

```
            {"model_state_dict": model_dict},
            "/checkpoint.pt",
            partial=False,
        )
```

> **Note**
>
> Pertimbangkan hal berikut untuk pemeriksaan lengkap:
>
> - Jika Anda mengaturgather_to_rank0=True, semua peringkat selain 0 mengembalikan kamus kosong.
> - Untuk pos pemeriksaan penuh, Anda hanya dapat memeriksa model. Pemeriksaan penuh status pengoptimal saat ini tidak didukung.
> - Model lengkap hanya perlu disimpan dismp.rank() == 0.

5. Muat pos pemeriksaan menggunakan smp.load(). Bergantung pada bagaimana Anda memeriksa pada langkah sebelumnya, pilih salah satu dari dua opsi berikut:

   - Opsi 1: Muat pos pemeriksaan sebagian.

     ```
     checkpoint = smp.load("/checkpoint.pt", partial=True)
     model.load_state_dict(checkpoint["model_state_dict"], same_partition_load=False)
     optimizer.load_state_dict(checkpoint["optimizer_state_dict"])
     ```

     Anda dapat mengatur same_partition_load=True model.load_state_dict() untuk beban yang lebih cepat, jika Anda tahu bahwa partisi tidak akan berubah.

   - Opsi 2: Muat pos pemeriksaan penuh.

     ```
     if smp.rdp_rank() == 0:
         checkpoint = smp.load("/checkpoint.pt", partial=False)
         model.load_state_dict(checkpoint["model_state_dict"])
     ```

     if smp.rdp_rank() == 0Kondisi ini tidak diperlukan, tetapi dapat membantu menghindari pemuatan berlebihan di antara s yang berbedaMP_GROUP. Dict status pengoptimal pos pemeriksaan penuh saat ini tidak didukung dengan paralelisme tensor.

Checkpointing model terdistribusi TensorFlow

Untuk menyimpan TensorFlow model saat berlatih dengan paralelisme model, gunakan fungsi berikut yang disediakan oleh perpustakaan paralelisme SageMaker model.

- `smdistributed.modelparallel.tensorflow.DistributedModel.save_model`
- `smdistributed.modelparallel.tensorflow.CheckpointManager`

Menyetel model terdistribusi

Penyetelan halus perlu dikonfigurasi dalam skrip pelatihan Anda. Cuplikan kode berikut menunjukkan contoh struktur skrip pelatihan menggunakan kelas AutoModelForCausalLM dari Hugging Face Transformers dengan modifikasi untuk mendaftarkan `smdistributed.model.parallel.torch` modul dan pengaturan untuk fine-tuning.

> **Note**
>
> Menyetel transformator terdistribusi (model Transformer yang dibungkus oleh`smp.DistributedModel()`) dengan fungsi smp.delayed_param_initialization diaktifkan memerlukan pekerjaan fine-tuning untuk dikonfigurasi dengan sistem file FSx for Lustre. Dalam kasus di mana Anda ingin menyempurnakan model skala besar dengan opsi inisialisasi parameter tertunda, Anda harus menyiapkan sistem file FSx for Lustre.

```
import argparse
from transformers import AutoModelForCausalLM
import smdistributed.modelparallel
import smdistributed.modelparallel.torch as smp

def parse_args():

    parser = argparse.ArgumentParser()

    # set an arg group for model
    model_grp = parser.add_argument_group(
        title="model", description="arguments to describe model configuration"
    )

    ... # set up numerous args to parse from the configuration dictionary to the script
 for training
```

```
    # add arg for activating fine-tuning
    model_grp.add_argument(
        "--fine_tune",
        type=int,
        default=0,
        help="Fine-tune model from checkpoint or pretrained model",
    )

def main():
    """Main function to train GPT."""
    args = parse_args()

    ... # parse numerous args

    if args.fine_tune > 0 and args.delayed_param > 0 and smp.rank() == 0:
        pretrained_model = AutoModelForCausalLM.from_pretrained(
            args.model_name or args.model_dir
        )
        model_state_dict = pretrained_model.state_dict()
        path = os.path.join(args.model_dir, "fullmodel.pt")
        torch.save(model_state_dict, path)

    # create a Transformer model and wrap by smp.model_creation()
    # with options to configure model parallelism parameters offered by SageMaker
    with smp.model_creation(
        tensor_parallelism=smp.tp_size() > 1 or args.use_distributed_transformer > 0,
        zero_init=args.use_distributed_transformer == 0,
        dtype=dtype,
        distribute_embedding=args.sharded_data_parallel_degree > 1 and smp.tp_size() >
 1,
        use_alibi=args.alibi > 0,
        attention_in_fp32=args.attention_in_fp32 > 0,
        fp32_residual_addition=args.residual_addition_in_fp32 > 0,
        query_key_layer_scaling=args.query_key_layer_scaling > 0 and args.bf16 < 1,
        fused_softmax=args.fused_softmax > 0,
        fused_dropout=args.fused_dropout > 0,
        fused_bias_gelu=args.fused_bias_gelu > 0,
        flash_attention=args.flash_attention > 0,
    ):
        if args.fine_tune > 0 and args.delayed_param == 0:
            model = AutoModelForCausalLM.from_pretrained(
                args.model_name or args.model_dir
            )
```

```
        else:
            model = AutoModelForCausalLM.from_config(model_config)

    # wrap the model by smp.DistributedModel() to apply SageMaker model parallelism
    model = smp.DistributedModel(
        model, trace_device="gpu", backward_passes_per_step=args.gradient_accumulation
    )

    # wrap the optimizer by smp.DistributedOptimizer() to apply SageMaker model
parallelism
    optimizer= ... # define an optimizer
    optimizer = smp.DistributedOptimizer(
        optimizer,
        static_loss_scale=None,
        dynamic_loss_scale=True,
        dynamic_loss_args={"scale_window": 1000, "min_scale": 1, "delayed_shift": 2},
    )

    # for fine-tuning, use smp.resume_from_checkpoint() to load a pre-trained model
    if args.fine_tune > 0 and args.delayed_param > 0:
        smp.resume_from_checkpoint(args.model_dir, tag="fullmodel.pt", partial=False)
```

Untuk contoh lengkap skrip pelatihan dan buku catatan Jupyter, lihat contoh GPT-2 di repositori Contoh. PyTorch SageMaker GitHub

SageMaker Praktik Terbaik Paralelisme Model Terdistribusi

Gunakan panduan berikut saat Anda menjalankan pekerjaan pelatihan terdistribusi dengan pustaka paralel SageMaker model.

Menyiapkan Konfigurasi yang Tepat untuk Model yang Diberikan

Saat meningkatkan model, kami sarankan Anda untuk memeriksa daftar berikut secara berurutan. Setiap item daftar membahas keuntungan menggunakan teknik perpustakaan bersama dengan pengorbanan yang mungkin muncul.

> **Tip**
>
> Jika model dapat cocok dengan baik menggunakan subset fitur perpustakaan, menambahkan lebih banyak paralelisme model atau fitur penyimpanan memori biasanya tidak meningkatkan kinerja.

Menggunakan tipe instans GPU besar

- Dalam bidang paralelisme model, yang terbaik adalah menggunakan instance yang kuat dengan memori GPU besar untuk menangani overhead dari operasi paralelisme model seperti mempartisi model di beberapa GPU. Kami merekomendasikan penggunaan `ml.p4d` atau `ml.p3dn` contoh untuk melatih model DL besar. Instans ini juga dilengkapi dengan Elastic Fabric Adapter (EFA), yang menyediakan bandwidth jaringan yang lebih tinggi dan memungkinkan pelatihan skala besar dengan paralelisme model.

Status pengoptimal sharding

- Dampak status sharding optimizer tergantung pada jumlah peringkat paralel data. Biasanya, tingkat paralelisme data yang lebih tinggi (sebanding dengan ukuran node komputasi) dapat meningkatkan efisiensi penggunaan memori.

  Saat Anda ingin mengurangi ukuran cluster, pastikan Anda memeriksa konfigurasi sharding status pengoptimal. Misalnya, model DL besar dengan sharding status pengoptimal yang cocok pada cluster komputasi dengan 16 GPU (misalnya, dua instance P4d atau P4DE) mungkin tidak selalu muat pada node dengan 8 GPU (misalnya, satu instance P4d atau P4de). Ini karena memori gabungan 8 GPU lebih rendah daripada memori gabungan 16 GPU, dan memori yang diperlukan per GPU untuk sharding lebih dari 8 GPU juga lebih tinggi daripada memori per GPU untuk sharding selama skenario 16-GPU. Akibatnya, peningkatan kebutuhan memori mungkin tidak cocok dengan cluster yang lebih kecil.

  Untuk informasi selengkapnya, lihat Sharding Status Optimizer.

Pos pemeriksaan aktivasi

- Efisiensi memori dapat ditingkatkan dengan menggunakan checkpointing aktivasi untuk sekelompok modul. Semakin banyak Anda mengelompokkan modul, semakin efisien penggunaan memori. Saat memeriksa modul sekuensial untuk lapisan, `strategy` argumen `smp.set_activation_checkpointing` fungsi mengelompokkan lapisan bersama-sama untuk pemeriksaan. Misalnya, mengelompokkan dua atau lebih lapisan bersama-sama untuk checkpointing lebih efisien memori daripada checkpointing satu lapisan pada satu waktu, dan ini memperdagangkan waktu komputasi ekstra untuk mengurangi penggunaan memori.

  Untuk informasi selengkapnya, lihat Aktivasi Checkpointing.

Paralelisme tensor

- Derajat paralelisme tensor harus berupa kekuatan dua (2, 4, 8,..., 2^n), di mana derajat maksimum harus sama dengan jumlah GPU per node. Misalnya, jika Anda menggunakan node dengan 8 GPU, angka yang mungkin untuk derajat paralelisme tensor adalah 2, 4, dan 8. Kami tidak merekomendasikan angka arbitrer (seperti 3, 5, 6, dan 7) untuk tingkat paralelisme tensor. Saat Anda menggunakan beberapa node, salah mengonfigurasi tingkat paralelisme tensor dapat mengakibatkan menjalankan paralelisme tensor di seluruh node; ini menambahkan overhead yang signifikan dari komunikasi aktivasi di seluruh node dan dapat menjadi mahal secara komputasi.

  Untuk informasi selengkapnya, lihat Paralelisme Tensor.

Paralelisme pipa di seluruh node

- Anda dapat menjalankan paralelisme pipeline baik dalam satu node maupun di beberapa node. Saat Anda menggunakan paralelisme pipeline dalam kombinasi dengan paralelisme tensor, kami merekomendasikan menjalankan paralelisme pipeline di beberapa node dan menjaga paralelisme tensor dalam masing-masing node.
- Paralelisme pipa dilengkapi dengan tiga tombol berikut:`microbatches`,, `active_microbatches` dan. `prescaled_batch`
  - Saat Anda menggunakan paralelisme tensor dengan paralelisme pipa, kami sarankan untuk mengaktifkan `prescaled_batch` sehingga ukuran batch per grup paralel model dapat ditingkatkan untuk pipelining yang efisien. Dengan `prescaled_batch` diaktifkan, ukuran batch yang ditetapkan dalam skrip pelatihan menjadi `tp_size` kali ukuran batch yang ditetapkan untuk setiap peringkat tanpa`prescaled_batch`.
  - Meningkatkan jumlah `microbatches` membantu mencapai pipelining yang efisien dan kinerja yang lebih baik. Perhatikan bahwa ukuran microbatch efektif adalah ukuran batch dibagi dengan jumlah microbatch. Jika Anda menambah jumlah microbatch sambil menjaga ukuran batch konstan, setiap microbatch memproses lebih sedikit sampel.
  - Jumlah `active_microbatches` adalah jumlah maksimum microbatch yang secara bersamaan dalam proses selama pipelining. Untuk setiap microbatch aktif dalam proses, aktivasi dan gradiennya mengambil memori GPU. Oleh karena itu, peningkatan `active_microbatches` membutuhkan lebih banyak memori GPU.
- Jika memori GPU dan GPU kurang dimanfaatkan, tingkatkan paralelisasi yang lebih baik `active_microbatches` selama pipelining.

- Untuk informasi lebih lanjut tentang cara menggunakan paralelisme tensor dengan paralelisme pipa, lihat. Paralelisme tensor dikombinasikan dengan paralelisme pipa
- Untuk menemukan deskripsi parameter yang disebutkan di atas, lihat Parameter untuk `smdistributed` dalam dokumentasi SageMaker Python SDK.

Bongkar aktivasi ke CPU

- Pastikan bahwa ini digunakan dalam kombinasi dengan checkpointing aktivasi dan paralelisme pipa. Untuk memastikan bahwa pembongkaran dan pramuat terjadi di latar belakang, tentukan nilai yang lebih besar dari 1 ke parameter microbatches.
- Saat membongkar aktivasi, Anda mungkin dapat meningkatkan `active_microbatches` dan terkadang cocok dengan jumlah total microbatch. Ini tergantung pada modul mana yang diperiksa dan bagaimana model dipartisi.

  Untuk informasi selengkapnya, lihat Pembongkaran Aktivasi.

Konfigurasi referensi

Tim pelatihan paralelisme SageMaker model menyediakan titik referensi berikut berdasarkan eksperimen dengan model GPT-2, panjang urutan 512, dan ukuran kosakata 50.000.

Jumlah parameter model	Jenis instans	Paralelisme pipa	Paralelisme tensor	Sharding status pengoptimal	Pos pemeriksaan aktivasi	Batch berskala	Ukuran batch
10 miliar	16 `ml.p4d.24xlarge`	1	4	Benar	Setiap lapisan transformator	Benar	`batch_size=40`
30 miliar	16 `ml.p4d.24xlarge`	1	8	Benar	Setiap lapisan transformator	Benar	`batch_size=32`

Jumlah parameter model	Jenis instans	Paralelisme pipa	Paralelisme tensor	Sharding status pengoptimal	Pos pemeriksaan aktivasi	Batch berskala	Ukuran batch
60 miliar	32 `ml.p4d.24xlarge`	2	8	Benar	Setiap lapisan transformator	Benar	`batch_size=56` , `microbatches=4` , `active_microbatches=2`

Anda dapat mengekstrapolasi dari konfigurasi sebelumnya untuk memperkirakan penggunaan memori GPU untuk konfigurasi model Anda. Misalnya, jika Anda menambah panjang urutan untuk model 10 miliar parameter atau meningkatkan ukuran model menjadi 20 miliar, Anda mungkin ingin menurunkan ukuran batch terlebih dahulu. Jika model masih tidak cocok, coba tingkatkan derajat paralelisme tensor.

Memodifikasi Skrip Pelatihan Anda

- Sebelum Anda menggunakan fitur pustaka paralel SageMaker model dalam skrip pelatihan Anda, tinjauTips dan Jebakan Konfigurasi Perpustakaan Paralelisme Model SageMaker Terdistribusi.
- Untuk meluncurkan pekerjaan pelatihan lebih cepat, gunakan mode SageMaker lokal. Ini membantu Anda dengan cepat menjalankan pekerjaan pelatihan secara lokal pada instance SageMaker notebook. Bergantung pada skala instans ML tempat instans SageMaker notebook Anda berjalan, Anda mungkin perlu menyesuaikan ukuran model Anda dengan mengubah konfigurasi model, seperti lebar tersembunyi, jumlah lapisan transformator, dan kepala perhatian. Validasi jika model yang dikurangi berjalan dengan baik pada instance notebook sebelum menggunakan cluster besar untuk melatih model lengkap.

Memantau dan Mencatat Job Pelatihan Menggunakan SageMaker Konsol dan Amazon CloudWatch

Untuk memantau metrik tingkat sistem seperti pemanfaatan memori CPU, pemanfaatan memori GPU, dan pemanfaatan GPU, gunakan visualisasi yang disediakan melalui konsol. SageMaker

1. Di panel navigasi kiri, pilih Pelatihan.

2. Pilih pekerjaan Pelatihan.
3. Di panel utama, pilih nama pekerjaan pelatihan yang ingin Anda lihat lebih detail.
4. Jelajahi panel utama dan temukan bagian Monitor untuk melihat visualisasi otomatis.
5. Untuk melihat log pekerjaan pelatihan, pilih Lihat log di bagian Monitor. Anda dapat mengakses log pekerjaan pelatihan terdistribusi dari pekerjaan pelatihan di CloudWatch. Jika Anda meluncurkan pelatihan terdistribusi multi-node, Anda akan melihat beberapa aliran log dengan tag dalam format algo-n-1234567890. Aliran log algo-1 melacak log pelatihan dari simpul utama (ke-0).

Untuk informasi selengkapnya, lihat Memantau dan Menganalisis Pekerjaan Pelatihan Menggunakan Amazon CloudWatch Metrics.

Izin

Untuk menjalankan pekerjaan SageMaker pelatihan dengan paralelisme model atau buku catatan contoh pelatihan SageMaker terdistribusi, pastikan Anda memiliki izin yang tepat dalam peran IAM Anda, seperti berikut ini:

- Untuk menggunakan FSx for Lustre, tambahkan. `AmazonFSxFullAccess`
- Untuk menggunakan Amazon S3 sebagai saluran data, tambahkan. `AmazonS3FullAccess`
- Untuk menggunakan Docker, buat wadah Anda sendiri, dan dorong ke Amazon ECR, tambahkan. `AmazonEC2ContainerRegistryFullAccess`
- Untuk memiliki akses penuh untuk menggunakan seluruh rangkaian SageMaker fitur, tambahkan `AmazonSageMakerFullAccess`.

Tips dan Jebakan Konfigurasi Perpustakaan Paralelisme Model SageMaker Terdistribusi

Tinjau tips dan jebakan berikut sebelum menggunakan perpustakaan paralelisme model Amazon SageMaker. Daftar ini mencakup tips yang berlaku di seluruh kerangka kerja. Untuk TensorFlow dan tips PyTorch khusus, lihat Memodifikasi skrip TensorFlow pelatihan danMemodifikasi Skrip PyTorch Pelatihan, masing-masing.

Ukuran Batch dan Jumlah Microbatch

- Pustaka paling efisien ketika ukuran batch ditingkatkan. Untuk kasus penggunaan di mana model cocok dalam satu perangkat, tetapi hanya dapat dilatih dengan ukuran batch kecil, ukuran batch dapat dan harus ditingkatkan setelah perpustakaan terintegrasi. Paralelisme model menghemat

memori untuk model besar, memungkinkan Anda berlatih menggunakan ukuran batch yang sebelumnya tidak sesuai dengan memori.

- Memilih sejumlah microbatch yang terlalu kecil atau terlalu besar dapat menurunkan kinerja. Pustaka mengeksekusi setiap microbatch secara berurutan di setiap perangkat, sehingga ukuran microbatch (ukuran batch dibagi dengan jumlah microbatch) harus cukup besar untuk sepenuhnya memanfaatkan setiap GPU. Pada saat yang sama, efisiensi pipa meningkat dengan jumlah microbatch, jadi penting untuk mencapai keseimbangan yang tepat. Biasanya, titik awal yang baik adalah mencoba 2 atau 4 microbatch, meningkatkan ukuran batch ke batas memori, dan kemudian bereksperimen dengan ukuran batch yang lebih besar dan jumlah microbatch. Karena jumlah microbatch meningkat, ukuran batch yang lebih besar mungkin menjadi layak jika pipa interleaved digunakan.
- Ukuran batch Anda harus selalu habis dibagi dengan jumlah microbatch. Perhatikan bahwa tergantung pada ukuran kumpulan data, terkadang batch terakhir dari setiap epoch dapat berukuran lebih kecil daripada yang lain, dan batch yang lebih kecil ini perlu dibagi dengan jumlah microbatch juga. Jika tidak, Anda dapat mengatur `drop_remainder=True` `tf.Dataset.batch()` panggilan (in TensorFlow), atau mengatur `DataLoader` (`drop_last=True`in PyTorch), sehingga batch kecil terakhir ini tidak digunakan. Jika Anda menggunakan API yang berbeda untuk pipeline data, Anda mungkin perlu melewati batch terakhir secara manual setiap kali tidak habis dibagi dengan jumlah microbatch.

Partisi Manual

- Jika Anda menggunakan partisi manual, perhatikan parameter yang digunakan oleh beberapa operasi dan modul dalam model Anda, seperti tabel penyematan dalam arsitektur transformator. Modul yang berbagi parameter yang sama harus ditempatkan di perangkat yang sama untuk kebenaran. Saat partisi otomatis digunakan, pustaka secara otomatis memberlakukan batasan ini.

Persiapan Data

- Jika model mengambil beberapa input, pastikan Anda menyemai operasi acak dalam pipeline data Anda (misalnya, pengocokan) dengan. `smp.dp_rank()` Jika kumpulan data sedang dipecah secara deterministik di seluruh perangkat paralel data, pastikan pecahan diindeks oleh. `smp.dp_rank()` Ini untuk memastikan bahwa urutan data yang terlihat pada semua peringkat yang membentuk partisi model konsisten.

Mengembalikan Tensor dari `smp.DistributedModel`

- Setiap tensor yang dikembalikan dari fungsi `smp.DistributedModel.call` (for TensorFlow) atau `smp.DistributedModel.forward` (for PyTorch) disiarkan ke semua peringkat lain, dari peringkat yang menghitung tensor tertentu. Akibatnya, tensor apa pun yang tidak diperlukan di luar metode panggilan dan penerusan (aktivasi menengah, misalnya) tidak boleh dikembalikan, karena ini menyebabkan komunikasi yang tidak perlu dan overhead memori dan merusak kinerja.

`@smp.step`Dekorator

- Jika fungsi `smp.step` -decorated memiliki argumen tensor yang tidak memiliki dimensi batch, nama argumen harus disediakan dalam `non_split_inputs` daftar saat memanggil. `smp.step` Ini mencegah perpustakaan mencoba membagi tensor menjadi microbatch. Untuk informasi selengkapnya lihat `smp.step`di dokumentasi API.

Menunda Inisialisasi Parameter

Untuk model yang sangat besar lebih dari 100 miliar parameter, inisialisasi bobot melalui memori CPU dapat mengakibatkan out-of-memory kesalahan. Untuk menyiasatinya, perpustakaan menawarkan manajer `smp.delay_param_initialization` konteks. Ini menunda alokasi fisik parameter sampai mereka pindah ke GPU selama eksekusi pertama dari fungsi -decorated. `smp.step` Ini menghindari penggunaan memori CPU yang tidak perlu selama inisialisasi pelatihan. Gunakan manajer konteks saat Anda membuat objek model seperti yang ditunjukkan dalam kode berikut.

```
with smp.delay_param_initialization(enabled=True):
    model = MyModel()
```

Paralelisme Tensor untuk PyTorch

- Jika Anda menggunakan benih untuk hasil deterministik, atur benih berdasarkan `smp.dp_rank()` (misalnya,`torch.manual_seed(42 + smp.dp_rank())`). Jika Anda tidak melakukan ini, partisi yang berbeda dari an `nn.Parameter` diinisialisasi dengan cara yang sama, memengaruhi konvergensi.
- SageMakerperpustakaan paralelisme model menggunakan NCCL untuk mengimplementasikan kolektif yang diperlukan untuk distribusi modul. Khusus untuk model yang lebih kecil, jika terlalu banyak panggilan NCCL dijadwalkan pada GPU pada saat yang sama, penggunaan memori

mungkin meningkat karena ruang tambahan yang digunakan oleh NCCL. Untuk mengatasi hal ini, smp membatasi panggilan NCCL sehingga jumlah operasi NCCL yang sedang berlangsung pada waktu tertentu kurang dari atau sama dengan batas yang diberikan. Batas default adalah 8, tetapi ini dapat disesuaikan menggunakan variabel lingkunganSMP_NCCL_THROTTLE_LIMIT. Jika Anda mengamati lebih banyak penggunaan memori daripada yang Anda harapkan saat menggunakan paralelisme tensor, Anda dapat mencoba mengurangi batas ini. Namun, memilih batas yang terlalu kecil dapat menyebabkan kerugian throughput. Untuk menonaktifkan throttling sama sekali, Anda dapat mengatur. SMP_NCCL_THROTTLE_LIMIT=-1

- Identitas berikut, yang berlaku ketika derajat paralelisme tensor adalah 1, tidak berlaku ketika derajat paralelisme tensor lebih besar dari 1:. smp.mp_size() * smp.dp_size() == smp.size() Ini karena gugus paralel tensor merupakan bagian dari kelompok paralelisme model dan kelompok paralelisme data. Jika kode Anda memiliki referensi kemp_rank,mp_size,MP_GROUP, dan sebagainya, dan jika Anda ingin bekerja hanya dengan grup paralel pipeline, Anda mungkin perlu mengganti referensi dengansmp.pp_size(). Identitas berikut selalu benar:
  - smp.mp_size() * smp.rdp_size() == smp.size()
  - smp.pp_size() * smp.dp_size() == smp.size()
  - smp.pp_size() * smp.tp_size() * smp.rdp_size() == smp.size()
- Karena smp.DistributedModel pembungkus memodifikasi parameter model saat paralelisme tensor diaktifkan, pengoptimal harus dibuat setelah memanggilsmp.DistributedModel, dengan parameter terdistribusi. Misalnya, berikut ini tidak berfungsi:

  ```
  ## WRONG
  model = MyModel()
  optimizer = SomeOptimizer(model.parameters())
  model = smp.DistributedModel(model)  # optimizer now has outdated parameters!
  ```

  Sebagai gantinya, pengoptimal harus dibuat dengan parameter smp.DistributedModel sebagai berikut:

  ```
  ## CORRECT
  model = smp.DistributedModel(MyModel())
  optimizer = SomeOptimizer(model.optimizers())
  ```

- Ketika modul diganti dengan mitra terdistribusi melalui paralelisme tensor, modul terdistribusi tidak mewarisi bobotnya dari modul asli, dan menginisialisasi bobot baru. Ini berarti bahwa, misalnya, jika bobot perlu diinisialisasi dalam panggilan tertentu (misalnya, melalui load_state_dict

panggilan), ini perlu terjadi setelah `smp.DistributedModel` panggilan, setelah distribusi modul berlangsung.

- Saat mengakses parameter modul terdistribusi secara langsung, perhatikan bahwa bobotnya tidak memiliki bentuk yang sama dengan modul aslinya. Sebagai contoh,

```
with smp.tensor_parallelism():
    linear = nn.Linear(60, 60)

# will pass
assert tuple(linear.weight.shape) == (60, 60)

distributed_linear = smp.DistributedModel(linear)

# will fail. the number of input channels will have been divided by smp.tp_size()
assert tuple(distributed_linear.module.weight.shape) == (60, 60)
```

- Penggunaan sangat `torch.utils.data.distributed.DistributedSampler` disarankan untuk paralelisme tensor. Ini memastikan bahwa setiap peringkat paralel data menerima jumlah sampel data yang sama, yang mencegah hang yang mungkin dihasilkan dari `dp_rank` s yang berbeda mengambil sejumlah langkah yang berbeda.
- Jika Anda menggunakan `join` API `DistributedDataParallel` kelas untuk menangani kasus di mana peringkat paralel data yang berbeda memiliki jumlah batch yang berbeda, Anda masih perlu memastikan bahwa peringkat yang sama `TP_GROUP` memiliki jumlah batch yang sama; jika tidak, kolektif komunikasi yang digunakan dalam eksekusi terdistribusi modul dapat hang. PyTorch Peringkat yang berada di `TP_GROUP` s berbeda dapat memiliki jumlah batch yang berbeda, selama `join` API digunakan.
- Jika Anda ingin memeriksa model Anda dan menggunakan paralelisme tensor, pertimbangkan hal berikut:
  - Untuk menghindari kondisi macet dan balapan saat menyimpan dan memuat model saat Anda menggunakan paralelisme tensor, pastikan Anda memanggil fungsi yang sesuai dari model berikut dan status pengoptimal di dalam peringkat paralelisme data yang dikurangi.
  - Jika Anda mentransisikan skrip paralel pipeline yang ada dan mengaktifkan tensor parallel untuk skrip, pastikan Anda memodifikasi `if smp.dp_rank() == 0` blok apa pun yang digunakan untuk menyimpan dan memuat dengan blok. `if smp.rdp_rank() == 0` Jika tidak, itu mungkin menyebabkan pekerjaan pelatihan Anda terhenti.

  Untuk informasi lebih lanjut tentang checkpointing model dengan paralelisme tensor, lihat. the section called “Checkpointing model terdistribusi”

Pemecahan Masalah Paralel Model

Jika Anda mengalami kesalahan, Anda dapat menggunakan daftar berikut untuk mencoba memecahkan masalah pekerjaan pelatihan Anda. Jika masalah berlanjut, hubungi AWSSupport.

Topik

Pertimbangan untuk Menggunakan SageMaker Debugger dengan Model Parallelism Library SageMaker

SageMaker Debugger tidak tersedia untuk pustaka SageMaker paralelisme model. Debugger diaktifkan secara default untuk semua pekerjaan SageMaker TensorFlow dan PyTorch pelatihan, dan Anda mungkin melihat kesalahan yang terlihat seperti berikut:

```
FileNotFoundError: [Errno 2] No such file or directory: '/opt/ml/checkpoints/
metadata.json.sagemaker-uploading
```

Untuk memperbaiki masalah ini, nonaktifkan Debugger dengan meneruskan `debugger_hook_config=False` saat membuat kerangka kerja `estimator` seperti yang ditunjukkan pada contoh berikut.

```
bucket=sagemaker.Session().default_bucket()
base_job_name="sagemaker-checkpoint-test"
checkpoint_in_bucket="checkpoints"

# The S3 URI to store the checkpoints
checkpoint_s3_bucket="s3://{}/{}/{}".format(bucket, base_job_name,
 checkpoint_in_bucket)

estimator = TensorFlow(
    ...
```

```
    distribution={"smdistributed": {"modelparallel": { "enabled": True }}},
    checkpoint_s3_uri=checkpoint_s3_bucket,
    checkpoint_local_path="/opt/ml/checkpoints",
    debugger_hook_config=False
)
```

Menyimpan Pos Pemeriksaan

Anda mungkin mengalami kesalahan berikut saat menyimpan pos pemeriksaan model besar di SageMaker:

```
InternalServerError: We encountered an internal error. Please try again
```

Ini bisa disebabkan oleh SageMaker batasan saat mengunggah pos pemeriksaan lokal ke Amazon S3 selama pelatihan. Untuk menonaktifkan pos pemeriksaan SageMaker, gunakan contoh berikut untuk mengunggah pos pemeriksaan secara eksplisit.

Jika Anda mengalami kesalahan sebelumnya, jangan gunakan `checkpoint_s3_uri` dengan panggilan. SageMaker `estimator` Saat menyimpan pos pemeriksaan untuk model yang lebih besar, kami sarankan untuk menyimpan pos pemeriksaan ke direktori khusus dan meneruskan yang sama ke fungsi pembantu (sebagai `local_path` argumen).

```
import os

def aws_s3_sync(source, destination):
    """aws s3 sync in quiet mode and time profile"""
    import time, subprocess
    cmd = ["aws", "s3", "sync", "--quiet", source, destination]
    print(f"Syncing files from {source} to {destination}")
    start_time = time.time()
    p = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    p.wait()
    end_time = time.time()
    print("Time Taken to Sync: ", (end_time-start_time))
    return

def sync_local_checkpoints_to_s3(local_path="/opt/ml/checkpoints",
 s3_uri=os.path.dirname(os.path.dirname(os.getenv('SM_MODULE_DIR', '')))+'/
checkpoints'):
    """ sample function to sync checkpoints from local path to s3 """

    import boto3
```

```
    #check if local path exists
    if not os.path.exists(local_path):
        raise RuntimeError("Provided local path {local_path} does not exist. Please
 check")

    #check if s3 bucket exists
    s3 = boto3.resource('s3')
    if not s3_uri.startswith("s3://"):
        raise ValueError(f"Provided s3 uri {s3_uri} is not valid.")

    s3_bucket = s3_uri.replace('s3://','').split('/')[0]
    print(f"S3 Bucket: {s3_bucket}")
    try:
        s3.meta.client.head_bucket(Bucket=s3_bucket)
    except Exception as e:
        raise e
    aws_s3_sync(local_path, s3_uri)
    return

def sync_s3_checkpoints_to_local(local_path="/opt/ml/checkpoints",
 s3_uri=os.path.dirname(os.path.dirname(os.getenv('SM_MODULE_DIR', '')))+'/
checkpoints'):
    """ sample function to sync checkpoints from s3 to local path """

    import boto3
    #try to create local path if it does not exist
    if not os.path.exists(local_path):
        print(f"Provided local path {local_path} does not exist. Creating...")
        try:
            os.makedirs(local_path)
        except Exception as e:
            raise RuntimeError(f"Failed to create {local_path}")

    #check if s3 bucket exists
    s3 = boto3.resource('s3')
    if not s3_uri.startswith("s3://"):
        raise ValueError(f"Provided s3 uri {s3_uri} is not valid.")

    s3_bucket = s3_uri.replace('s3://','').split('/')[0]
    print(f"S3 Bucket: {s3_bucket}")
    try:
        s3.meta.client.head_bucket(Bucket=s3_bucket)
    except Exception as e:
        raise e
```

```
    aws_s3_sync(s3_uri, local_path)
    return
```

Penggunaan fungsi pembantu:

```
#base_s3_uri - user input s3 uri or save to model directory (default)
#curr_host - to save checkpoints of current host
#iteration - current step/epoch during which checkpoint is saved

# save checkpoints on every node using local_rank
if smp.local_rank() == 0:
    base_s3_uri = os.path.dirname(os.path.dirname(os.getenv('SM_MODULE_DIR', '')))
    curr_host = os.environ['SM_CURRENT_HOST']
    full_s3_uri = f'{base_s3_uri}/checkpoints/{curr_host}/{iteration}'
    sync_local_checkpoints_to_s3(local_path=checkpoint_dir, s3_uri=full_s3_uri)
```

Konvergensi Menggunakan Model Paralel dan TensorFlow

Saat Anda menggunakan pelatihan SageMaker multi-node dengan TensorFlow dan pustaka paralelisme model, kerugian mungkin tidak menyatu seperti yang diharapkan karena urutan file input pelatihan mungkin berbeda pada setiap node. Ini dapat menyebabkan peringkat yang berbeda dalam grup paralel model yang sama bekerja pada file input yang berbeda, menyebabkan inkonsistensi. Untuk mencegah hal ini, pastikan file input diurutkan dengan cara yang sama di semua peringkat sebelum dikonversi ke TensorFlow kumpulan data. Salah satu cara untuk mencapai ini adalah dengan mengurutkan nama file input dalam skrip pelatihan.

Menghentikan atau Menghancurkan Pekerjaan Pelatihan Terdistribusi

Jika pekerjaan pelatihan Anda mengalami masalah macet, mogok, atau tidak merespons, baca item pemecahan masalah berikut untuk mengidentifikasi penyebab masalah tersebut. Jika Anda membutuhkan dukungan lebih lanjut, hubungi tim pelatihan SageMaker terdistribusi melalui AWSSupport.

- Jika Anda melihat pekerjaan pelatihan terdistribusi terhenti pada langkah inisialisasi NCCL, pertimbangkan hal berikut:
  - Jika Anda menggunakan salah satu instans (`ml.p4d`atau `ml.p3dn` instance) yang mendukung EFA dengan VPC kustom dan subnetnya, pastikan bahwa grup keamanan yang digunakan memiliki koneksi masuk dan keluar untuk semua port ke dan dari SG yang sama. Anda juga umumnya memerlukan koneksi keluar ke IP apa pun sebagai aturan terpisah (untuk akses internet). Untuk menemukan petunjuk tentang cara menambahkan aturan masuk dan keluar

untuk komunikasi EFA, lihat. SageMaker mengulur-ulur pekerjaan pelatihan terdistribusi selama inisialisasi

- Jika Anda melihat pekerjaan pelatihan terdistribusi terhenti saat memeriksa model lengkap, ini mungkin karena `state_dict()` panggilan pada model atau pengoptimal tidak dilakukan di semua peringkat dengan `rdp_rank()==0` (saat menggunakan paralelisme tensor) atau `dp_rank()==0` (saat hanya menggunakan paralelisme pipa). Peringkat ini perlu berkomunikasi untuk membangun pos pemeriksaan untuk diselamatkan. Masalah penghentian serupa juga dapat terjadi saat checkpointing pengoptimal sebagian jika diaktifkan. `shard_optimizer_state`

  Untuk informasi selengkapnya tentang checkpointing model dengan paralelisme model, lihat Instruksi Umum untuk Menyimpan dan Memuat dan. Checkpointing PyTorch model terdistribusi (untuk pustaka paralelisme SageMaker model antara v1.6.0 dan v1.9.0)

- Jika pekerjaan pelatihan macet dengan kesalahan CUDA Out of Memory, ini berarti bahwa konfigurasi pelatihan terdistribusi perlu disesuaikan agar sesuai dengan model pada cluster GPU. Untuk informasi lebih lanjut dan praktik terbaik, lihatMenyiapkan Konfigurasi yang Tepat untuk Model yang Diberikan.
- Jika pekerjaan pelatihan macet dengan kesalahan ECC yang tidak dapat diperbaiki, ini berarti bahwa salah satu GPU di cluster menjadi buruk. Jika Anda memerlukan dukungan teknis, bagikan pekerjaan ARN dengan AWS tim dan mulai ulang pekerjaan pelatihan Anda dari pos pemeriksaan jika memungkinkan.
- Dalam kasus yang jarang terjadi, konfigurasi pekerjaan yang bekerja sebelumnya tetapi mendekati batas memori GPU mungkin gagal nanti dengan cluster yang berbeda karena kesalahan CUDA Out of Memory. Ini bisa jadi karena beberapa GPU memiliki memori yang tersedia lebih rendah dari biasanya karena kesalahan ECC.
- Kerusakan batas waktu jaringan mungkin terjadi saat menjalankan pekerjaan multinode yang tidak menggunakan semua GPU di node. Untuk menyiasatinya, gunakan semua GPU pada node dengan memastikan bahwa `processes_per_host` parameter diatur ke jumlah GPU di setiap instance. Misalnya, ini `processes_per_host=8` untuk`ml.p3.16xlarge`,`ml.p3dn.24xlarge`, dan `ml.p4d.24xlarge` contoh.
- Jika Anda menemukan bahwa pekerjaan pelatihan Anda membutuhkan waktu lama selama tahap pengunduhan data, pastikan jalur Amazon S3 yang Anda berikan `checkpoint_s3_uri` untuk SageMaker `Estimator` kelas tersebut unik untuk pekerjaan pelatihan saat ini. Jika jalur ini digunakan kembali di beberapa pekerjaan pelatihan yang berjalan secara bersamaan, semua pos pemeriksaan tersebut diunggah dan diunduh ke jalur Amazon S3 yang sama dan mungkin secara signifikan meningkatkan waktu pemuatan pos pemeriksaan.

- Gunakan FSx for Lustre ketika Anda berurusan dengan data besar dan model.
  - Jika kumpulan data Anda besar dan mengambilnya membutuhkan waktu lama, kami sarankan untuk menyimpan kumpulan data Anda di FSx for Lustre.
  - Ketika model pelatihan melebihi 10 miliar parameter, kami sarankan menggunakan FSx for Lustre untuk pos pemeriksaan.
  - Setelah Anda membuat sistem file, pastikan untuk menunggu status tersedia sebelum memulai pekerjaan pelatihan menggunakannya.

Menerima Kesalahan NCCL untuk Training Job PyTorch

Jika Anda mengalami kesalahan berikut, mungkin karena proses kehabisan memori GPU.

```
NCCL error in: ../torch/lib/c10d/ProcessGroupNCCL.cpp:825, unhandled system error, NCCL
 version 2.7.8
ncclSystemError: System call (socket, malloc, munmap, etc) failed.
```

Anda dapat mengatasi ini dengan mengurangi ukuran batch atau`active_microbatches`. Jika partisi otomatis tidak menghasilkan partisi yang seimbang, Anda mungkin harus mempertimbangkan partisi manual. Untuk informasi selengkapnya, lihat Paralelisme pipa di seluruh node.

Menerima **`RecursionError`** PyTorch Training Job

Pustaka tidak mendukung pemanggilan `super.forward()` di dalam panggilan penerusan modul. Jika Anda menggunakan`super.forward()`, Anda mungkin menerima pesan galat berikut.

```
RecursionError: maximum recursion depth exceeded
```

Untuk memperbaiki kesalahan, alih-alih meneleponsuper.forward(), Anda harus meneleponsuper()._orig_forward().

Contoh Notebook Pelatihan SageMaker Terdistribusi Amazon

Studi kasus dan buku catatan berikut memberikan contoh penerapan perpustakaan pelatihan SageMaker terdistribusi untuk kerangka kerja pembelajaran mendalam yang didukung (PyTorch,, dan HuggingFace) dan model TensorFlow, seperti CNN dan MaskRCNN untuk visi, dan BERT untuk pemrosesan bahasa alami.

Notebook ini disediakan dalam SageMaker contoh GitHub repositori. Anda juga dapat menelusurinya di situs web SageMaker contoh.

Blog dan Studi Kasus

Blog-blog berikut membahas studi kasus tentang penggunaan perpustakaan pelatihan SageMaker terdistribusi.

SageMaker Pustaka paralelisme data

- Aktifkan pelatihan lebih cepat dengan library paralel SageMaker data Amazon, AWSMachine Learning Blog (05 Desember 2023)
- Bagaimana saya melatih 10TB untuk Difusi Stabil SageMaker di Sedang (November 29, 2022)
- Jalankan PyTorch Lightning dan PyTorch DDP asli di Amazon SageMaker Training, menampilkan Amazon Search, AWSMachine Learning Blog (18 Agustus 2022)
- Pelatihan YoloV5 AWS dengan PyTorch dan perpustakaan paralel data SageMaker terdistribusi, Medium (6 Mei 2022)
- Mempercepat pelatihan EfficientNet model SageMaker dengan PyTorch dan perpustakaan paralel data SageMaker terdistribusi, Sedang (Maret 21, 2022)
- Percepat EfficientNet pelatihan AWS dengan perpustakaan paralel data SageMaker terdistribusi, Menuju Ilmu Data (12 Januari 2022)
- Hyundai mengurangi waktu pelatihan model ML untuk model mengemudi otonom menggunakan Amazon SageMaker, AWSMachine Learning Blog (25 Juni 2021)
- Pelatihan Terdistribusi: Latih BART/T5 untuk Ringkasan menggunakan Transformers dan Amazon, situs web SageMaker Hugging Face (8 April 2021)

Perpustakaan paralelisme SageMaker model

- Peningkatan kinerja baru di perpustakaan paralelisme SageMaker model Amazon, Blog AWS Machine Learning (16 Desember 2022)
- Latih model raksasa dengan penskalaan hampir linier menggunakan paralelisme data sharded di Amazon SageMaker, AWSMachine Learning Blog (31 Oktober 2022)

PyTorchContoh

SageMaker Pustaka paralelisme data

- CNN dengan PyTorch dan perpustakaan SageMaker paralelisme data
- BERT dengan PyTorch dan perpustakaan paralelisme SageMaker data

Perpustakaan paralelisme SageMaker model

- Latih GPT-2 dengan penskalaan linier dekat menggunakan teknik paralelisme data sharded di perpustakaan paralelisme model SageMaker
- Sempurnakan GPT-2 dengan penskalaan hampir linier menggunakan teknik paralelisme data sharded di perpustakaan paralelisme model SageMaker
- Latih GPT-Neox-20b dengan penskalaan linier dekat menggunakan teknik paralelisme data sharded di perpustakaan paralelisme model SageMaker
- Latih GPT-J 6B menggunakan paralelisme data sharded dan teknik paralelisme tensor di perpustakaan paralelisme model SageMaker
- Latih FLAN-T5 dengan penskalaan linier dekat menggunakan teknik paralelisme data sharded di perpustakaan paralelisme model SageMaker
- Latih Falcon dengan penskalaan hampir linier menggunakan teknik paralelisme data sharded di perpustakaan paralelisme model SageMaker

TensorFlowContoh

SageMaker Pustaka paralelisme data

- CNN dengan TensorFlow 2.3.1 dan perpustakaan paralelisme SageMaker data
- BERT dengan TensorFlow 2.3.1 dan perpustakaan SageMaker paralelisme data

Perpustakaan paralelisme SageMaker model

- CNN dengan TensorFlow 2.3.1 dan perpustakaan paralelisme SageMaker model

HuggingFaceContoh

Berikut ini HuggingFace pada SageMaker contoh tersedia di HuggingFacerepositori notebook.

SageMaker Pustaka paralelisme data

- HuggingFace Pelatihan Paralel Data Terdistribusi dalam PyTorch Menjawab Pertanyaan Terdistribusi SageMaker
- HuggingFace Pelatihan Paralel Data SageMaker Terdistribusi dalam PyTorch Ringkasan Teks Terdistribusi

- HuggingFace Pelatihan Paralel Data Terdistribusi TensorFlow di SageMaker

Perpustakaan paralelisme SageMaker model

- HuggingFace dengan Perpustakaan paralelisme model TensorFlow terdistribusi Pelatihan tentang SageMaker

Cara Mengakses atau Mengunduh Contoh Notebook Pelatihan SageMaker Terdistribusi

Ikuti petunjuk untuk mengakses atau mengunduh buku catatan contoh pelatihan SageMaker terdistribusi.

Opsi 1: Gunakan contoh SageMaker notebook

Untuk menggunakan contoh yang disebutkan di atas, kami sarankan Anda menggunakan instance SageMaker notebook Amazon. Instans notebook menjalankan Jupyter Notebook dan JupyterServer aplikasi di instans Amazon EC2, yang dioptimalkan untuk pembelajaran mesin. Jika Anda tidak memiliki instance notebook aktif, ikuti petunjuk Membuat Instance Notebook di panduan SageMaker pengembang untuk membuatnya.

Setelah Anda membuat instance, di halaman instance Notebook SageMaker konsol, lakukan hal berikut:

1. Buka JupyterLab.
2. Pilih ikon contoh
   ()
   di baki kiri.
3. Jelajahi contoh untuk Pelatihan dan cari buku catatan berjudul Distributed Data Parallel atau Distributed Model Parallel.

Opsi 2: Kloning repositori SageMaker contoh ke SageMaker Studio atau instance notebook

Untuk mengunduh dan menggunakan contoh notebook yang disebutkan di atas, lakukan hal berikut untuk mengkloning repositori contoh: GitHub

1. Buka terminal.

2. Di baris perintah, navigasikan ke SageMaker folder.

```
cd SageMaker
```

3. Kloning SageMaker GitHub repositori contoh.

```
git clone https://github.com/aws/amazon-sagemaker-examples.git
```

> **Note**
>
> Untuk mengunduh HuggingFace contoh notebook, kloning repositori HuggingFace notebook GitHub :
>
> ```
> git clone https://github.com/huggingface/notebooks huggingface-notebooks
> ```

4. Di JupyterLab antarmuka, navigasikan ke `amazon-sagemaker-examples` folder.
5. Di `training/distributed_training` folder, ada folder untuk kerangka kerja, dan di masing-masing, ada folder untuk `data_parallel` dan`model_parallel`. Pilih contoh pilihan Anda dan ikuti instruksi untuk meluncurkan pelatihan terdistribusi dengan perpustakaan pelatihan SageMaker terdistribusi.

Komputasi terdistribusi dengan praktik SageMaker terbaik

Halaman praktik terbaik ini menyajikan berbagai ragam pekerjaan komputasi terdistribusi untuk pembelajaran mesin (ML) secara umum. Istilah komputasi terdistribusi di halaman ini mencakup pelatihan terdistribusi untuk tugas pembelajaran mesin dan komputasi paralel untuk pemrosesan data, pembuatan data, rekayasa fitur, dan pembelajaran penguatan. Di halaman ini, kami membahas tentang tantangan umum dalam komputasi terdistribusi, dan opsi yang tersedia dalam SageMaker Pelatihan dan SageMaker Pemrosesan. Untuk materi bacaan tambahan tentang komputasi terdistribusi, lihat Apa Itu Komputasi Terdistribusi? .

Anda dapat mengonfigurasi tugas ML untuk dijalankan secara terdistribusi di beberapa node (instance), akselerator (GPU NVIDIA, chip AWS Trainium), dan inti vCPU. Dengan menjalankan komputasi terdistribusi, Anda dapat mencapai berbagai tujuan seperti operasi komputasi lebih cepat, menangani kumpulan data besar, atau melatih model ML yang besar.

Daftar berikut mencakup tantangan umum yang mungkin Anda hadapi ketika Anda menjalankan pekerjaan pelatihan ML dalam skala besar.

- Anda perlu membuat keputusan tentang cara mendistribusikan komputasi tergantung pada tugas ML, pustaka perangkat lunak yang ingin Anda gunakan, dan sumber daya komputasi.
- Tidak semua tugas ML mudah didistribusikan. Selain itu, tidak semua pustaka ML mendukung komputasi terdistribusi.
- Komputasi terdistribusi mungkin tidak selalu menghasilkan peningkatan linear dalam efisiensi komputasi. Secara khusus, Anda perlu mengidentifikasi apakah data I/O dan komunikasi antar-GPU memiliki kemacetan atau menyebabkan overhead.
- Komputasi terdistribusi dapat mengganggu proses numerik dan mengubah akurasi model. Khusus untuk pelatihan jaringan saraf paralel data, saat Anda mengubah ukuran batch global sambil meningkatkan skala ke cluster komputasi yang lebih besar, Anda juga perlu menyesuaikan tingkat pembelajaran yang sesuai.

SageMaker menyediakan solusi pelatihan terdistribusi untuk meringankan tantangan tersebut untuk berbagai kasus penggunaan. Pilih salah satu opsi berikut yang paling sesuai dengan kasus penggunaan Anda.

Topik

Opsi 1: Gunakan algoritma SageMaker bawaan yang mendukung pelatihan terdistribusi

SageMaker menyediakan algoritma bawaan yang dapat Anda gunakan di luar kotak melalui SageMaker konsol atau SageMaker Python SDK. Menggunakan algoritme bawaan, Anda tidak perlu menghabiskan waktu untuk kustomisasi kode, memahami ilmu di balik model, atau menjalankan Docker pada instans Amazon EC2 yang disediakan.

Subset dari algoritma SageMaker bawaan mendukung pelatihan terdistribusi. Untuk memeriksa apakah algoritme pilihan Anda mendukung pelatihan terdistribusi, lihat kolom Parallelizable di

tabel Common Information About Built-in Algorithms. Beberapa algoritme mendukung pelatihan terdistribusi multi-instance, sementara algoritme paralelisasi lainnya mendukung paralelisasi di beberapa GPU dalam satu instance, seperti yang ditunjukkan dalam kolom Parallelizable.

Opsi 2: Jalankan kode HTML kustom di lingkungan pelatihan atau pemrosesan SageMaker terkelola

SageMaker pekerjaan dapat membuat instance lingkungan pelatihan terdistribusi untuk kasus penggunaan dan kerangka kerja tertentu. Lingkungan ini bertindak sebagai ready-to-use papan tulis, di mana Anda dapat membawa dan menjalankan kode ML Anda sendiri.

Jika kode ML Anda menggunakan kerangka pembelajaran mendalam

Anda dapat meluncurkan pekerjaan pelatihan terdistribusi menggunakan Deep Learning Containers (DLC) for SageMaker Training, yang dapat Anda orkestrasi baik melalui modul Python khusus di Python SDK, atau melalui API SageMaker dengan,. SageMaker AWS CLIAWS SDK for Python (Boto3) SageMaker menyediakan wadah pelatihan untuk kerangka kerja pembelajaran mesin, termasuk,, Hugging Face Transformers PyTorchTensorFlow, dan Apache MXNet. Anda memiliki dua opsi untuk menulis kode pembelajaran mendalam untuk pelatihan terdistribusi.

- Perpustakaan pelatihan yang SageMaker didistribusikan

  Perpustakaan pelatihan SageMaker terdistribusi mengusulkan kode yang AWS dikelola untuk paralelisme data jaringan saraf dan paralelisme model. SageMaker pelatihan terdistribusi juga dilengkapi dengan klien peluncur yang dibangun ke dalam SageMaker Python SDK, dan Anda tidak perlu membuat kode peluncuran paralel. Untuk mempelajari lebih lanjut, lihat SageMaker perpustakaan paralelisme data dan SageMaker pustaka paralelisme model.
- Perpustakaan pelatihan terdistribusi sumber terbuka

  Kerangka kerja open source memiliki mekanisme distribusi sendiri seperti DistributedDataParallelism (DDP) di dalam PyTorch atau `tf.distribute` modul di. TensorFlow Anda dapat memilih untuk menjalankan kerangka kerja pelatihan terdistribusi ini di wadah kerangka kerja yang SageMaker dikelola. Misalnya, kode sampel untuk pelatihan MaskRCNN dalam SageMaker menunjukkan cara menggunakan kedua PyTorch DDP dalam wadah kerangka kerja dan Horovod dalam wadah SageMaker PyTorch kerangka kerja. SageMaker TensorFlow

SageMaker Kontainer HTML juga dilengkapi dengan MPI yang sudah diinstal sebelumnya, sehingga Anda dapat memparalelkan skrip titik masuk Anda menggunakan mpi4py. Menggunakan wadah

pelatihan terintegrasi MPI adalah pilihan yang bagus saat Anda meluncurkan peluncur pelatihan terdistribusi pihak ketiga atau menulis kode paralel ad-hoc di lingkungan pelatihan SageMaker terkelola.

Catatan untuk pelatihan jaringan saraf paralel data pada GPU

- Skala ke multi-GPU dan paralelisme multi-mesin bila sesuai

  Kami sering menjalankan pekerjaan pelatihan jaringan saraf pada instance multi-CPU atau multiple-GPU. Setiap instance berbasis GPU biasanya berisi beberapa perangkat GPU. Akibatnya, komputasi GPU terdistribusi dapat terjadi baik dalam satu instance GPU dengan beberapa GPU (pelatihan multi-GPU simpul tunggal), atau di beberapa instance GPU dengan beberapa inti GPU di masing-masing (pelatihan multi-GPU multi-node). Pelatihan single-instance lebih mudah untuk menulis kode dan debug, dan throughput GPU-ke-GPU intra-node biasanya lebih cepat daripada throughput GPU-ke-GPU antar-node. Oleh karena itu, merupakan ide yang baik untuk menskalakan paralelisme data secara vertikal terlebih dahulu (gunakan satu instance GPU dengan beberapa GPU) dan memperluas ke beberapa instance GPU jika diperlukan. Ini mungkin tidak berlaku untuk kasus di mana anggaran CPU tinggi (misalnya, beban kerja besar untuk pra-pemrosesan data) dan ketika rasio CPU-ke-GPU dari instance multi-GPU terlalu rendah. Dalam semua kasus, Anda perlu bereksperimen dengan kombinasi tipe instans yang berbeda berdasarkan kebutuhan pelatihan dan beban kerja Anda sendiri.
- Pantau kualitas konvergensi

  Saat melatih jaringan saraf dengan paralelisme data, meningkatkan jumlah GPU sambil menjaga ukuran mini-batch per GPU konstan menyebabkan peningkatan ukuran mini-batch global untuk proses penurunan gradien stokastik mini-batch (MSGD). Ukuran mini-batch untuk MSGD diketahui berdampak pada kebisingan turun dan konvergensi. Untuk penskalaan yang benar sambil mempertahankan akurasi, Anda perlu menyesuaikan hiperparameter lain seperti tingkat pembelajaran [Goyal et al. (2017)].
- Pantau kemacetan I/O

  Saat Anda meningkatkan jumlah GPU, throughput untuk penyimpanan membaca dan menulis juga harus meningkat. Pastikan sumber data dan pipeline Anda tidak menjadi hambatan.
- Ubah skrip pelatihan Anda sesuai kebutuhan

  Skrip pelatihan yang ditulis untuk pelatihan GPU tunggal harus dimodifikasi untuk pelatihan multi-GPU multi-node. Di sebagian besar pustaka paralelisme data, modifikasi skrip diperlukan untuk melakukan hal berikut.

- Tetapkan batch data pelatihan untuk setiap GPU.
- Gunakan pengoptimal yang dapat menangani komputasi gradien dan pembaruan parameter di beberapa GPU.
- Tetapkan tanggung jawab checkpointing ke host dan GPU tertentu.

Jika kode ML Anda melibatkan pemrosesan data tabular

PySpark adalah frontend Python dari Apache Spark, yang merupakan kerangka kerja komputasi terdistribusi open-source. PySpark telah diadopsi secara luas untuk pemrosesan data tabular terdistribusi untuk beban kerja produksi skala besar. Jika Anda ingin menjalankan kode pemrosesan data tabular, pertimbangkan untuk menggunakan PySpark kontainer SageMaker Processing dan menjalankan pekerjaan paralel. Anda juga dapat menjalankan pekerjaan pemrosesan data secara paralel menggunakan API SageMaker Pelatihan dan SageMaker Pemrosesan di Amazon SageMaker Studio Classic, yang terintegrasi dengan Amazon EMR dan. AWS Glue

Opsi 3: Tulis kode pelatihan terdistribusi kustom Anda sendiri

Saat Anda mengirimkan tugas pelatihan atau pemrosesan ke SageMaker, API SageMaker Pelatihan dan SageMaker Pemrosesan meluncurkan instans komputasi Amazon EC2. Anda dapat menyesuaikan lingkungan pelatihan dan pemrosesan dalam instance dengan menjalankan container Docker Anda sendiri atau menginstal pustaka tambahan di container AWS terkelola. Untuk informasi selengkapnya tentang Docker with SageMaker Training, lihat Mengadaptasi container Docker Anda sendiri untuk dikerjakan SageMaker dan Membuat container dengan algoritme dan model Anda sendiri. Untuk informasi selengkapnya tentang Docker dengan SageMaker Processing, lihat Menggunakan Kode Pemrosesan Anda Sendiri.

Setiap lingkungan pekerjaan SageMaker pelatihan berisi file konfigurasi di`/opt/ml/input/config/resourceconfig.json`, dan setiap lingkungan pekerjaan SageMaker pemrosesan berisi file konfigurasi serupa di`/opt/ml/config/resourceconfig.json`. Kode Anda dapat membaca file ini untuk menemukan `hostnames` dan membangun komunikasi antar simpul. Untuk mempelajari selengkapnya, termasuk skema file JSON, lihat Konfigurasi Pelatihan Terdistribusi dan Cara SageMaker Pemrosesan Amazon Mengkonfigurasi Kontainer Pemrosesan Anda. Anda juga dapat menginstal dan menggunakan pustaka komputasi terdistribusi pihak ketiga seperti Ray atau DeepSpeed in SageMaker.

Anda juga dapat menggunakan SageMaker Pelatihan dan SageMaker Pemrosesan untuk menjalankan komputasi terdistribusi khusus yang tidak memerlukan komunikasi antar pekerja.

Dalam literatur komputasi, tugas-tugas itu sering digambarkan sebagai paralel yang memalukan atau tidak berbagi apa-apa. Contohnya termasuk pemrosesan paralel file data, model pelatihan secara paralel pada konfigurasi yang berbeda, atau menjalankan inferensi batch pada kumpulan catatan. Anda dapat secara sepele memparalelkan kasus penggunaan share-nothing seperti itu dengan Amazon. SageMaker Saat Anda meluncurkan pekerjaan SageMaker Pelatihan atau SageMaker Pemrosesan di klaster dengan beberapa node, secara SageMaker default mereplikasi dan meluncurkan kode pelatihan Anda (dengan Python atau Docker) di semua node. Tugas yang membutuhkan penyebaran data input secara acak di beberapa node tersebut dapat difasilitasi dengan mengatur `S3DataDistributionType=ShardedByS3Key` konfigurasi input data SageMaker `TrainingInput` API.

Opsi 4: Luncurkan beberapa pekerjaan secara paralel atau berurutan

Anda juga dapat mendistribusikan alur kerja komputasi ML ke tugas komputasi paralel atau sekuensial yang lebih kecil, masing-masing diwakili oleh pekerjaan SageMaker Pelatihan atau Pemrosesan sendiri. SageMaker Memisahkan tugas menjadi beberapa pekerjaan dapat bermanfaat untuk situasi atau tugas berikut:

- Bila Anda memiliki saluran data dan entri metadata tertentu (seperti hyperparameters, konfigurasi model, atau tipe instance) untuk setiap sub-tugas.
- Saat Anda menerapkan langkah-langkah coba lagi di tingkat sub-tugas.
- Ketika Anda memvariasikan konfigurasi sub-tugas selama beban kerja, seperti saat pelatihan tentang peningkatan ukuran batch.
- Saat Anda perlu menjalankan tugas ML yang membutuhkan waktu lebih lama dari waktu pelatihan maksimum yang diizinkan untuk satu pekerjaan pelatihan (maksimum 28 hari).
- Ketika langkah-langkah yang berbeda dari alur kerja komputasi memerlukan jenis instance yang berbeda.

Untuk kasus spesifik pencarian hyperparameter, gunakan Penyetelan Model SageMaker Otomatis. SageMaker Automated Model Tuning adalah orkestrator pencarian parameter tanpa server yang meluncurkan beberapa pekerjaan pelatihan atas nama Anda, sesuai dengan logika pencarian yang bisa acak, Bayesian, atau. HyperBand

Selain itu, untuk mengatur beberapa pekerjaan pelatihan, Anda juga dapat mempertimbangkan alat orkestrasi alur kerja, seperti Pipelines, SageMaker Step FunctionsAWS, dan Apache Airflow yang didukung oleh Amazon ManagedWorkflows for Apache Airflow (MWAA) dan Workflows. SageMaker

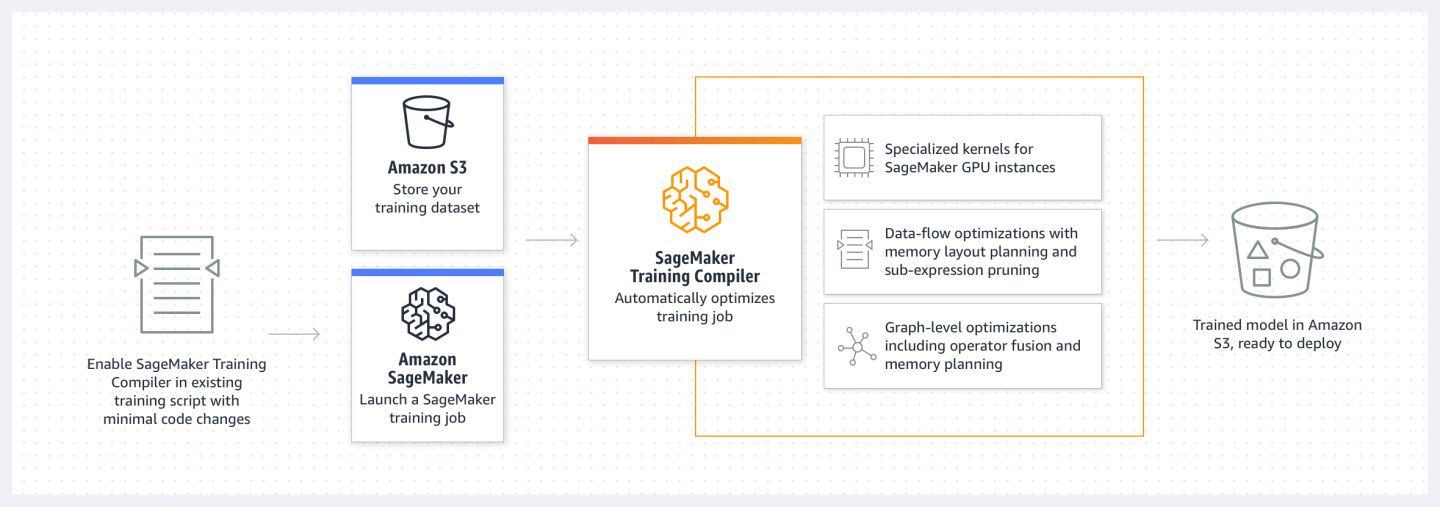
Kompiler SageMaker Pelatihan Amazon

Gunakan Amazon SageMaker Training Compiler untuk melatih model deep learning (DL) lebih cepat pada instans GPU skalabel yang dikelola oleh. SageMaker

Apa itu SageMaker Training Compiler?

Model S tate-of-the-art deep learning (DL) terdiri dari jaringan saraf berlapis-lapis yang kompleks dengan miliaran parameter yang dapat memakan waktu ribuan jam GPU untuk dilatih. Mengoptimalkan model seperti itu pada infrastruktur pelatihan membutuhkan pengetahuan luas tentang DL dan rekayasa sistem; ini menantang bahkan untuk kasus penggunaan yang sempit. Meskipun ada implementasi open-source dari kompiler yang mengoptimalkan proses pelatihan DL, mereka dapat kekurangan fleksibilitas untuk mengintegrasikan kerangka kerja DL dengan beberapa perangkat keras seperti instance GPU.

SageMaker Training Compiler adalah kemampuan SageMaker yang membuat hard-to-implement pengoptimalan ini untuk mengurangi waktu pelatihan pada instance GPU. Kompiler mengoptimalkan model DL untuk mempercepat pelatihan dengan lebih efisien menggunakan instance GPU SageMaker machine learning (ML). SageMaker Training Compiler tersedia tanpa biaya tambahan di dalamnya SageMaker dan dapat membantu mengurangi total waktu yang dapat ditagih karena mempercepat pelatihan.



SageMaker Training Compiler diintegrasikan ke dalam AWS Deep Learning Containers (DLC). Dengan menggunakan AWS DLC yang mendukung SageMaker Training Compiler, Anda dapat mengkompilasi dan mengoptimalkan pekerjaan pelatihan pada instans GPU dengan sedikit perubahan pada kode Anda. Bawa model deep learning Anda SageMaker dan aktifkan SageMaker

Training Compiler untuk mempercepat kecepatan pekerjaan pelatihan Anda pada instans SageMaker ML untuk komputasi yang dipercepat.

Cara Kerjanya

SageMaker Training Compiler mengonversi model DL dari representasi bahasa tingkat tinggi menjadi instruksi yang dioptimalkan perangkat keras. Secara khusus, SageMaker Training Compiler menerapkan pengoptimalan tingkat grafik, pengoptimalan tingkat rendah data, dan pengoptimalan backend untuk menghasilkan model yang dioptimalkan yang secara efisien menggunakan sumber daya perangkat keras. Hasilnya, Anda dapat melatih model Anda lebih cepat daripada saat Anda melatihnya tanpa kompilasi.

Ini adalah proses dua langkah untuk mengaktifkan SageMaker Training Compiler untuk pekerjaan pelatihan Anda:

1. Bawa skrip DL Anda sendiri dan, jika perlu, beradaptasi untuk mengkompilasi dan melatih dengan SageMaker Training Compiler. Untuk mempelajari selengkapnya, lihat Bawa Model Pembelajaran Mendalam Anda Sendiri.
2. Buat objek SageMaker estimator dengan parameter konfigurasi compiler menggunakan Python SageMaker SDK.
   a. Aktifkan SageMaker Training Compiler dengan menambahkan `compiler_config=TrainingCompilerConfig()` ke kelas SageMaker estimator.
   b. Sesuaikan hiperparameter (`batch_size`dan`learning_rate`) untuk memaksimalkan manfaat yang disediakan oleh SageMaker Training Compiler.

      Kompilasi melalui SageMaker Training Compiler mengubah jejak memori model. Paling umum, ini bermanifestasi sebagai pengurangan pemanfaatan memori dan konsekuensi peningkatan ukuran batch terbesar yang dapat muat pada GPU. Dalam beberapa kasus, kompiler cerdas mempromosikan caching yang mengarah pada penurunan ukuran batch terbesar yang dapat muat pada GPU. Perhatikan bahwa jika Anda ingin mengubah ukuran batch, Anda harus menyesuaikan tingkat pembelajaran dengan tepat.

      Untuk referensi untuk `batch_size` diuji untuk model populer, lihatModel yang Diuji.

      Saat Anda menyesuaikan ukuran batch, Anda juga harus menyesuaikannya `learning_rate` dengan tepat. Untuk praktik terbaik untuk menyesuaikan tingkat pembelajaran bersama dengan perubahan ukuran batch, lihatthe section called “Praktik dan Pertimbangan Terbaik”.

c. Dengan menjalankan metode `estimator.fit()` kelas, SageMaker mengkompilasi model Anda dan memulai pekerjaan pelatihan.

Untuk petunjuk tentang cara meluncurkan pekerjaan pelatihan, lihatAktifkan Kompiler SageMaker Pelatihan.

SageMaker Training Compiler tidak mengubah model terlatih akhir, sekaligus memungkinkan Anda untuk mempercepat pekerjaan pelatihan dengan lebih efisien menggunakan memori GPU dan menyesuaikan ukuran batch yang lebih besar per iterasi. Model terlatih terakhir dari pekerjaan pelatihan yang dipercepat kompiler identik dengan model dari pekerjaan pelatihan biasa.

> **Tip**
>
> SageMaker Training Compiler hanya mengompilasi model DL untuk pelatihan pada instance GPU yang didukung yang dikelola oleh. SageMaker Untuk mengkompilasi model Anda untuk inferensi dan menerapkannya untuk dijalankan di mana saja di cloud dan di tepi, gunakan SageMaker kompiler Neo.

Topik

Kerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji

Sebelum menggunakan SageMaker Training Compiler, periksa apakah kerangka kerja pilihan Anda didukung, jenis instans tersedia di AWS akun Anda, dan AWS akun Anda ada di salah satu yang didukungWilayah AWS.

> **Note**
>
> SageMaker Training Compiler tersedia di SageMaker Python SDK v2.70.0 atau yang lebih baru.

Kerangka Kerja yang Didukung

SageMaker Training Compiler mendukung kerangka pembelajaran mendalam berikut dan tersedia melalui AWS Deep Learning Containers.

Topik

- PyTorch
- TensorFlow

PyTorch

Kerangka Kerja	Versi kerangka	URI Wadah Pembelajaran Mendalam	Dapat diperpanjang untuk kustomisasi Docker
PyTorch	PyTorch v1.13.1	763104351884.dkr.ecr. <region>.amazonaws.com/:1.12.0-gpu-py38-cu113-ubuntu20.04-sagemaker pytorch-trcomp-training	Tidak
	PyTorch v1.12.0	763104351884.dkr.ecr. <region>.amazonaws.com/:1.13.1-gpu-py39-cu117-ubuntu20.04-sagemaker pytorch-trcomp-training	Tidak

Kerangka Kerja	Versi kerangka	URI Wadah Pembelajaran Mendalam	Dapat diperpanjang untuk kustomisasi Docker
PyTorch dengan Hugging Face Transformers	Transformer v4.21.1 PyTorch v1.11.0	763104351884.dkr.ecr. <region>. amazonaws.com/:1.11.0-transformers4.21.1-gpu-py38-cu113-ubuntu20.04 huggingface-pytorch-trcomp-training	Tidak
	Transformer v4.17.0 PyTorch v1.10.2	763104351884.dkr.ecr. <region>. amazonaws.com/:1.10.2-transformers4.17.0-gpu-py38-cu113-ubuntu20.04 huggingface-pytorch-trcomp-training	Tidak
	Transformer v4.11.0 PyTorch v1.9.0	763104351884.dkr.ecr. <region>. amazonaws.com/:1.9.0-transformers4.11.0-gpu-py38-cu111-ubuntu20.04 huggingface-pytorch-training-comp	Tidak

TensorFlow

Kerangka Kerja	Versi kerangka	URI Wadah Pembelajaran Mendalam	Dapat diperpanjang untuk kustomisasi Docker
TensorFlow	TensorFlow v2.11.0	763104351884.dkr.ecr. <region>.amazonaws.com/tensorflow-training:2.11.0-gpu-py39-cu112-ubuntu20.04-sagemaker	Ya
	TensorFlow v2.10.0	763104351884.dkr.ecr. <region>.amazonaws.com/tensorflow-training:2.10.0-gpu-py39-cu112-ubuntu20.04-sagemaker	Ya
	TensorFlow v2.9.1	763104351884.dkr.ecr. <region>.amazonaws.com/tensorflow-training:2.9.1-gpu-py39-cu112-ubuntu20.04-sagemaker	Ya
TensorFlow dengan Hugging Face Transformers	Transformer v4.17.0 TensorFlow v2.6.3	763104351884.dkr.ecr. <region>.amazonaws.com/:2.6.3-transformers4.17.0-gpu-py38-cu112-ubuntu20.04	Tidak

Kerangka Kerja	Versi kerangka	URI Wadah Pembelajaran Mendalam	Dapat diperpanjang untuk kustomisasi Docker
		huggingface-tensorflow-trcomp-training	
	Transformer v4.11.0 TensorFlow v2.5.1	763104351884.dkr.ecr.<region>.amazonaws.com/:2.5.1-transformers4.11.0-gpu-py37-cu112-ubuntu18.04 huggingface-tensorflow-training-comp	Tidak

Untuk informasi selengkapnya, lihat Gambar yang Tersedia di GitHub repositori AWS Deep Learning Containers.

Wilayah AWS

SageMaker Training Compiler Containers tersedia di Wilayah AWS tempat AWSDeep Learning Containers berada dalam layanan kecuali wilayah China.

Tipe Instans Yang Didukung

SageMaker Training Compiler diuji dan mendukung jenis instans ML berikut.

- Instans P4
- Instans P3
- Instans G4dn
- Instans G5

Untuk spesifikasi jenis instans, lihat bagian Komputasi Akselerasi di halaman Jenis Instans Amazon EC2. Untuk informasi tentang harga instans, lihat SageMaker Harga Amazon.

Jika Anda menemukan pesan galat yang mirip dengan berikut ini, ikuti petunjuk di Minta peningkatan kuota layanan untuk SageMaker sumber daya.

```
ResourceLimitExceeded: An error occurred (ResourceLimitExceeded) when calling
the CreateTrainingJob operation: The account-level service limit 'ml.p3dn.24xlarge
for training job usage' is 0 Instances, with current utilization of 0 Instances
and a request delta of 1 Instances.
Please contact AWS support to request an increase for this limit.
```

Model yang Diuji

Tabel berikut mencakup daftar model yang telah diuji dengan SageMaker Training Compiler. Sebagai referensi, ukuran batch terbesar yang dapat dimasukkan ke dalam memori juga disertakan bersama parameter pelatihan lainnya. SageMaker Training Compiler dapat mengubah jejak memori dari proses pelatihan model; sebagai hasilnya, ukuran batch yang lebih besar sering dapat digunakan selama proses pelatihan, yang selanjutnya mengurangi total waktu pelatihan. Dalam beberapa kasus, SageMaker Training Compiler secara cerdas mempromosikan caching yang mengarah pada penurunan ukuran batch terbesar yang dapat muat pada GPU. Anda harus menyetel ulang hyperparameters model Anda dan menemukan ukuran batch yang optimal untuk casing Anda. Untuk menghemat waktu, gunakan tabel referensi berikut untuk mencari ukuran batch yang bisa menjadi titik awal yang baik untuk kasus penggunaan Anda.

> **Note**
>
> Ukuran batch adalah ukuran batch lokal yang sesuai dengan masing-masing GPU individu dalam jenis instans masing-masing. Anda juga harus menyesuaikan tingkat pembelajaran saat mengubah ukuran batch.

PyTorch 1.13.1

Model pemrosesan bahasa alami (NLP)

Model-model berikut diuji untuk pekerjaan pelatihan untuk semua kombinasi node tunggal dan multi-node dengan core GPU tunggal atau multi dan Automatic Mixed Precision (AMP) seperti yang ditunjukkan.

Single-node/multi-node GPU/Multi-GPU Tunggal/Multi-GPU						
Model	Set data	Jenis instans	presisi	Panjang Urutan	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
albert-base-v2	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	128	80	192
albert-base-v2	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	128	332
albert-base-v2	wikitext-2-raw-v1	p3.2xlarge	mengapung16	128	80	224
bert-base-uncased	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	160	288
dasar camembert	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	160	280
distilbert-base-uncased	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	240	472
distilgpt2	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	128	77	128
distilgpt2	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	138	390
distilgpt2	wikitext-2-raw-v1	p3.2xlarge	mengapung16	128	96	256
distilroberta-basis	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	128	96	192

Single-node/multi-node GPU/Multi-GPU Tunggal/Multi-GPU						
Model	Set data	Jenis instans	presisi	Panjang Urutan	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
distilroberta-basis	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	171	380
distilroberta-basis	wikitext-2-raw-v1	p3.2xlarge	mengapung16	128	112	256
gpt2	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	128	52	152
gpt2	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	84	240
gpt2	wikitext-2-raw-v1	p3.2xlarge	mengapung16	128	58	164
microsoft/deberta-basis	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	128	48	128
microsoft/deberta-basis	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	84	207
microsoft/deberta-basis	wikitext-2-raw-v1	p3.2xlarge	mengapung16	128	53	133
roberta-basis	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	125	224

Single-node/multi-node GPU/Multi-GPU Tunggal/Multi-GPU						
Model	Set data	Jenis instans	presisi	Panjang Urutan	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
facebook/bart-base	xsum	g4dn.16xlarge	mengapung16	128	10	16
facebook/bart-base	xsum	g5.4xlarge	mengapung16	128	16	32
facebook/bart-besar	xsum	g5.4xlarge	mengapung16	128	5	8
facebook/bart-besar	xsum	p3.2xlarge	mengapung16	128	2	4
xlm-roberta-base	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	128	16	31
xlm-roberta-base	wikitext-2-raw-v1	p3.2xlarge	mengapung16	128	18	50
xlnet-base-cased	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	128	240
bert-base-uncased	wikitext-103-v1	g5.48xlarge	mengapung16	512	29	50
distilbert-base-uncased	wikitext-103-v1	g5.48xlarge	mengapung16	512	45	64
gpt2	wikitext-103-v1	g5.48xlarge	mengapung16	512	18	45

Single-node/multi-node GPU/Multi-GPU Tunggal/Multi-GPU						
Model	Set data	Jenis instans	presisi	Panjang Urutan	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
roberta-basis	wikitext-103-v1	g5.48xlarge	mengapung16	512	23	44
gpt2	wikitext-103-v1	p4d.24xlarge	mengapung16	512	36	64

Model Computer Vision (CV)

Diuji menggunakan TensorFlowModel Garden dengan Automatic Mixed Precision (AMP) seperti yang ditunjukkan.

Tunggal/multi-node tunggal/multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
ResNet152	makanan101	g4dn.16xlarge	mengapung16	128	144
ResNet152	makanan101	g5.4xlarge	mengapung16	128	192
ResNet152	makanan101	p3.2xlarge	mengapung16	152	156

Tunggal/multi-node tunggal/multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
Vlt	makanan101	g4dn.16xlarge	mengapung16	512	512
Vlt	makanan101	g5.4xlarge	mengapung16	992	768
Vlt	makanan101	p3.2xlarge	mengapung16	848	768

PyTorch 1.12.0

Model pemrosesan bahasa alami (NLP)

Model-model berikut diuji untuk pekerjaan pelatihan untuk semua kombinasi node tunggal dan multi-node dengan core GPU tunggal atau multi dan Automatic Mixed Precision (AMP) seperti yang ditunjukkan.

Single-node/multi-node GPU/Multi-GPU Tunggal/Multi-GPU						
Model	Set data	Jenis instans	presisi	Panjang Urutan	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
albert-base-v2	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	128	248

Single-node/multi-node GPU/Multi-GPU Tunggal/Multi-GPU						
Model	Set data	Jenis instans	presisi	Panjang Urutan	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
bert-base-uncased	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	160	288
dasar camembert	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	160	279
dasar camembert	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	128	105	164
distilgpt2	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	136	256
distilgpt2	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	128	80	118
gpt2	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	84	240
gpt2	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	128	80	119
microsoft/deberta-basis	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	93	197
microsoft/deberta-basis	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	128	113	130

Single-node/multi-node GPU/Multi-GPU Tunggal/Multi-GPU						
Model	Set data	Jenis instans	presisi	Panjang Urutan	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
roberta-base	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	125	224
roberta-base	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	128	78	112
xlnet-base-cased	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	138	240
bert-base-uncased	wikitext-103-v1	ml.p4d.24xlarge	mengapung16	512		52
distilbert-base-uncased	wikitext-103-v1	ml.p4d.24xlarge	mengapung16	512		160
gpt2	wikitext-103-v1	ml.p4d.24xlarge	mengapung16	512		25
roberta-base	wikitext-103-v1	ml.p4d.24xlarge	mengapung16	512		64

TensorFlow2.11.0

Model Computer Vision (CV)

Diuji menggunakan TensorFlowModel Garden dengan Automatic Mixed Precision (AMP) seperti yang ditunjukkan.

Tunggal/multi-node tunggal/multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
MaskRCNN-50-FPN ResNet	COCO-2017	ml.g5.2xbesar	mengapung 16	6	8
MaskRCNN-50-FPN ResNet	COCO-2017	ml.p3.2xlarge	mengapung 16	4	6
ResNet50	ImageNet	ml.g5.2xbesar	mengapung 16	192	256
ResNet50	ImageNet	ml.p3.2xlarge	mengapung 16	256	256
ResNet101	ImageNet	ml.g5.2xbesar	mengapung 16	128	256
ResNet101	ImageNet	ml.p3.2xlarge	mengapung 16	128	128
ResNet152	ImageNet	ml.g5.2xbesar	mengapung 16	128	224
ResNet152	ImageNet	ml.p3.2xlarge	mengapung 16	128	128
VisionTransformer	ImageNet	ml.g5.2xbesar	mengapung 16	112	144
VisionTransformer	ImageNet	ml.p3.2xlarge	mengapung 16	96	128

Model Natural Language Processing (NLP)

Diuji menggunakan model Transformer dengan `Sequence_Len=128` dan Automatic Mixed Precision (AMP) seperti yang ditunjukkan.

Tunggal/multi-node tunggal/multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
albert-base-v2	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	160	197
albert-base-v2	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	95	127
bert-base-uncased	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	160	128
bert-base-uncased	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	104	111
bert-large-uncased	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	65	48
bert-large-uncased	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	40	35
dasar camembert	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	162
dasar camembert	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	105	111

Tunggal/multi-node tunggal/multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
distilbert-base-uncased	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	256	264
distilbert-base-uncased	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	128	169
gpt2	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	120
gpt2	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	80	83
jplu/ tf-xlm-roberta-base	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	32	32
jplu/ tf-xlm-roberta-base	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	32	36
microsoft/ mpnet-basis	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	144	160
microsoft/ mpnet-basis	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	106	110
roberta-basis	wikitext-2-raw-v1	ml.g5.2xbesar	mengapung16	128	128
roberta-basis	wikitext-2-raw-v1	ml.p3.2xlarge	mengapung16	72	98

Tunggal/multi-node tunggal/multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
albert-base-v2	wikitext-2-raw-v1	ml.g5.48x besar	mengapung 16	128	192
albert-base-v2	wikitext-2-raw-v1	ml.p3.16x large	mengapung 16	95	96
distilbert-base-uncased	wikitext-2-raw-v1	ml.g5.48x besar	mengapung 16	256	256
distilbert-base-uncased	wikitext-2-raw-v1	ml.p3.16x large	mengapung 16	140	184
google/electra-small-discriminator	wikitext-2-raw-v1	ml.g5.48x besar	mengapung 16	256	384
google/electra-small-discriminator	wikitext-2-raw-v1	ml.p3.16x large	mengapung 16	256	268
gpt2	wikitext-2-raw-v1	ml.g5.48x besar	mengapung 16	116	116
gpt2	wikitext-2-raw-v1	ml.p3.16x large	mengapung 16	85	83
gpt2	wikitext-2-raw-v1	ml.p4d.24 xlarge	mengapung 16	94	110

Tunggal/multi-node tunggal/multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
microsoft/mpnet-basis	wikitext-2-raw-v1	ml.g5.48xbesar	mengapung16	187	164
microsoft/mpnet-basis	wikitext-2-raw-v1	ml.p3.16xlarge	mengapung16	106	111

TensorFlow2.10.0

Model Computer Vision (CV)

Diuji menggunakan TensorFlowModel Garden dengan Automatic Mixed Precision (AMP) seperti yang ditunjukkan.

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
Detection Transformer-ResNet 50	COCO-2017	ml.g4dn.2xbesar	mengapung32	2	4
Detection Transformer-ResNet 50	COCO-2017	ml.g5.2xbesar	mengapung32	3	6

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
Detection Transformer-ResNet 50	COCO-2017	ml.p3.2xlarge	mengapung 32	2	4
MaskRCNN-50-FPN ResNet	COCO-2017	ml.g4dn.2xbesar	mengapung 16	4	6
MaskRCNN-50-FPN ResNet	COCO-2017	ml.g5.2xbesar	mengapung 16	6	8
MaskRCNN-50-FPN ResNet	COCO-2017	ml.g5.48xbesar	mengapung 16	48	64
MaskRCNN-50-FPN ResNet	COCO-2017	ml.p3.2xlarge	mengapung 16	4	6
ResNet50	ImageNet	ml.g4dn.2xbesar	mengapung 16	224	256
ResNet50	ImageNet	ml.g5.2xbesar	mengapung 16	192	160
ResNet50	ImageNet	ml.g5.48xbesar	mengapung 16	2048	2048
ResNet50	ImageNet	ml.p3.2xlarge	mengapung 16	224	160

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
ResNet101	ImageNet	ml.g4dn.2xbesar	mengapung 16	160	128
ResNet101	ImageNet	ml.g5.2xbesar	mengapung 16	192	256
ResNet101	ImageNet	ml.g5.48xbesar	mengapung 16	2048	2048
ResNet101	ImageNet	ml.p3.2xlarge	mengapung 16	160	224
ResNet152	ImageNet	ml.g4dn.2xbesar	mengapung 16	128	128
ResNet152	ImageNet	ml.g5.2xbesar	mengapung 16	192	224
ResNet152	ImageNet	ml.g5.48xbesar	mengapung 16	1536	1792
ResNet152	ImageNet	ml.p3.2xlarge	mengapung 16	128	160
VisionTransformer	ImageNet	ml.g4dn.2xbesar	mengapung 16	80	128
VisionTransformer	ImageNet	ml.g5.2xbesar	mengapung 16	112	144

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
VisionTransformer	ImageNet	ml.g5.48x besar	mengapung 16	896	1152
VisionTransformer	ImageNet	ml.p3.2xlarge	mengapung 16	80	128

Model Natural Language Processing (NLP)

Diuji menggunakan model Transformer dengan `Sequence_Len=128` dan Automatic Mixed Precision (AMP) seperti yang ditunjukkan.

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
albert-base-v2	wikitext-2-raw-v1	g4dn.16xlarge	mengapung 16	128	112
albert-base-v2	wikitext-2-raw-v1	p3.2xlarge	mengapung 16	128	128
albert-base-v2	wikitext-2-raw-v1	p3.8xlarge	mengapung 16	128	135
albert-base-v2	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	128	191

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
bert-base-uncased	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	64	94
bert-base-uncased	wikitext-2-raw-v1	p3.2xlarge	mengapung16	96	101
bert-base-uncased	wikitext-2-raw-v1	p3.8xlarge	mengapung16	96	96
bert-base-uncased	wikitext-2-raw-v1	g5.4xlarge	mengapung16	128	128
bert-large-uncased	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	35	21
bert-large-uncased	wikitext-2-raw-v1	p3.2xlarge	mengapung16	39	26
bert-large-uncased	wikitext-2-raw-v1	g5.4xlarge	mengapung16	60	50
dasar camembert	wikitext-2-raw-v1	g4dn.16xlarge	mengapung16	96	90
dasar camembert	wikitext-2-raw-v1	p3.2xlarge	mengapung16	96	98
dasar camembert	wikitext-2-raw-v1	p3.8xlarge	mengapung16	96	96

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
dasar camembert	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	128	128
distilbert-base-uncased	wikitext-2-raw-v1	g4dn.16xlarge	mengapung 16	256	160
distilbert-base-uncased	wikitext-2-raw-v1	p3.2xlarge	mengapung 16	128	176
distilbert-base-uncased	wikitext-2-raw-v1	p3.8xlarge	mengapung 16	128	160
distilbert-base-uncased	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	256	258
google_electra-small-discriminator	wikitext-2-raw-v1	g4dn.16xlarge	mengapung 16	256	216
google_electra-small-discriminator	wikitext-2-raw-v1	p3.2xlarge	mengapung 16	256	230
google_electra-small-discriminator	wikitext-2-raw-v1	p3.8xlarge	mengapung 16	256	224

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
google_ electra-small-discriminator	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	256	320
gpt2	wikitext-2-raw-v1	g4dn.16xlarge	mengapung 16	80	64
gpt2	wikitext-2-raw-v1	p3.2xlarge	mengapung 16	80	77
gpt2	wikitext-2-raw-v1	p3.8xlarge	mengapung 16	80	72
gpt2	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	128	120
jplu_ tf-xlm-roberta-base	wikitext-2-raw-v1	g4dn.16xlarge	mengapung 16	28	24
jplu_ tf-xlm-roberta-base	wikitext-2-raw-v1	p3.2xlarge	mengapung 16	32	24
jplu_ tf-xlm-roberta-base	wikitext-2-raw-v1	p3.8xlarge	mengapung 16	32	26
jplu_ tf-xlm-roberta-base	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	66	52
microsoft _mpnet-basis	wikitext-2-raw-v1	g4dn.16xlarge	mengapung 16	96	92

Node tunggal GPU/Multi-GPU					
Model	Set data	Jenis instans	presisi	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
microsoft _mpnet-basis	wikitext-2-raw-v1	p3.2xlarge	mengapung 16	96	101
microsoft _mpnet-basis	wikitext-2-raw-v1	p3.8xlarge	mengapung 16	96	101
microsoft _mpnet-basis	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	128	152
roberta-basis	wikitext-2-raw-v1	g4dn.16xlarge	mengapung 16	64	72
roberta-basis	wikitext-2-raw-v1	p3.2xlarge	mengapung 16	64	84
roberta-basis	wikitext-2-raw-v1	p3.8xlarge	mengapung 16	64	86
roberta-basis	wikitext-2-raw-v1	g5.4xlarge	mengapung 16	128	128

TensorFlow2.9.1

Diuji menggunakan TensorFlowModel Garden dengan Automatic Mixed Precision (AMP).

Node tunggal GPU/Multi-GPU				
Model	Set data	Jenis instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
ResNet50	ImageNet	ml.g4dn.2xbesar	192	256*
ResNet101	ImageNet	ml.g4dn.2xbesar	128	160
		ml.g5.2xbesar	224	256*
		ml.p3.16xlarge	1536	1792
ResNet152	ImageNet	ml.g5.2xbesar	192	224
		ml.p3.2xlarge	160	160
		ml.p3.16xlarge	1024	1280
VisionTransformer	ImageNet	ml.g4dn.2xbesar	80	128*
		ml.g5.2xbesar	112	128*
		ml.p3.2xlarge	56	128*
		ml.p3.16xlarge	640	1024*
Detection Transformer-ResNet 50	COCO-2017	ml.g4dn.2xbesar	2	2
		ml.g5.2xbesar	3	6
		ml.p3.2xlarge	2	4
		ml.p3.16xlarge	8	32
MaskRCNN- 50-FPN ResNet	COCO-2017	ml.g4dn.2xbesar	4	4
		ml.g5.2xbesar	6	8

Node tunggal GPU/Multi-GPU				
Model	Set data	Jenis instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk SageMaker Training Compiler
		ml.p3.2xlarge	4	6

* Ukuran batch yang ditandai dengan simbol tanda bintang (*) menunjukkan ukuran batch terbesar yang diuji oleh tim pengembang SageMaker Training Compiler. Untuk sel yang ditandai, instance mungkin dapat memuat ukuran batch yang lebih besar dari yang ditunjukkan.

Transformers 4.21.1 dengan 1.11.0 PyTorch

Diuji dengan `Sequence_Len=512` dan Automatic Mixed Precision (AMP).

GPU tunggal simpul tunggal					
Model	Set data	Jenis instans	Jumlah instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
albert-base-v2	wikitext-2	ml.g4dn.2xbesar	1	14	28
		ml.g5.2xbesar	1	18	40
		ml.p3.2xlarge	1	14	32
bert-base-cased	wikitext-2	ml.g4dn.2xbesar	1	12	24
		ml.g5.2xbesar	1	28	44

GPU tunggal simpul tunggal					
Model	Set data	Jenis instans	Jumlah instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
		ml.p3.2xlarge	1	16	20
dasar camembert	wikitext-2	ml.g4dn.2xbesar	1	16	28
		ml.g5.2xbesar	1	24	40
		ml.p3.2xlarge	1	16	24
distilbert-base-uncased	wikitext-2	ml.g4dn.2xbesar	1	28	52
		ml.g5.2xbesar	1	40	76
		ml.p3.2xlarge	1	32	48
	wikitext-103-v1	ml.p4d.24xlarge	4	82	160
distilgpt2	wikitext-2	ml.g4dn.2xbesar	1	6	18
		ml.g5.2xbesar	1	12	28
		ml.p3.2xlarge	1	6	16
distilroberta-basis	wikitext-2	ml.g4dn.2xbesar	1	20	40

GPU tunggal simpul tunggal					
Model	Set data	Jenis instans	Jumlah instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
		ml.g5.2xbesar	1	28	56
		ml.p3.2xlarge	1	24	40
Eleuther ai/GPT-neo -125m	wikitext-2	ml.g4dn.2xbesar	1	4	8
		ml.g5.2xbesar	1	6	14
		ml.p3.2xlarge	1	4	10
gpt2	wikitext-2	ml.g4dn.2xbesar	1	4	8
		ml.g5.2xbesar	1	6	16
		ml.p3.2xlarge	1	4	10
	wikitext-103-v1	ml.p4d.24xlarge	4	13	25
roberta-basis	wikitext-2	ml.g4dn.2xbesar	1	12	20
		ml.g5.2xbesar	1	24	36
		ml.p3.2xlarge	1	12	20

GPU tunggal simpul tunggal					
Model	Set data	Jenis instans	Jumlah instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
	wikitext-103-v1	ml.p4d.24xlarge	4	36	64
xlnet-base-cased	wikitext-2	ml.g4dn.2xbesar	1	2	6
		ml.g5.2xbesar	1	2	10
		ml.p3.2xlarge	1	2	8
bert-base-uncased	wikitext-103-v1	ml.p4d.24xlarge	2	32	64
			4	32	64
			8	32	64
			16	32	64
roberta-besar	wikitext-103-v1	ml.p4d.24xlarge	4	16	24
microsoft/deberta-v3-basis	wikitext-103-v1	ml.p4d.24xlarge	16	9	23

Transformers 4.17.0 dengan 1.10.2 PyTorch

Diuji dengan `Sequence_Len=512` dan Automatic Mixed Precision (AMP).

GPU tunggal simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
albert-base-v2	ml.p3.2xlarge	14	28
	ml.g4dn.2xbesar	14	24
bert-base-cased	ml.p3.2xlarge	16	24
	ml.g4dn.2xbesar	12	24
bert-base-uncased	ml.p3.2xlarge	16	24
	ml.g4dn.2xbesar	12	28
dasar camembert	ml.p3.2xlarge	12	24
	ml.g4dn.2xbesar	12	28
distilbert-base-uncased	ml.p3.2xlarge	28	48
	ml.g4dn.2xbesar	24	52
distilgpt2	ml.p3.2xlarge	6	12
	ml.g4dn.2xbesar	6	14
distilroberta-basis	ml.p3.2xlarge	20	40
	ml.g4dn.2xbesar	12	40
Eleutherai/GPT-neo-125m	ml.p3.2xlarge	2	10
	ml.g4dn.2xbesar	2	8
facebook/bart-base	ml.p3.2xlarge	2	6
	ml.g4dn.2xbesar	2	6
gpt2	ml.p3.2xlarge	4	8

GPU tunggal simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
	ml.g4dn.2xbesar	2	8
roberta-basis	ml.p3.2xlarge	12	20
	ml.g4dn.2xbesar	12	20
xlnet-base-cased	ml.p3.2xlarge	2	8
	ml.g4dn.2xbesar	4	6

Transformers 4.11.0 dengan 1.9.0 PyTorch

Diuji dengan `Sequence_Len=512` dan Automatic Mixed Precision (AMP).

GPU tunggal simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk asli	Ukuran Batch untuk Training Compiler
albert-base-v2	ml.p3.2xlarge	12	32
bert-base-cased	ml.p3.2xlarge	14	24
bert-base-chinese	ml.p3.2xlarge	16	24
bert-base-multilingual-cased	ml.p3.2xlarge	4	16
bert-base-multilingual-uncased	ml.p3.2xlarge	8	16
bert-base-uncased	ml.p3.2xlarge	12	24

GPU tunggal simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk asli	Ukuran Batch untuk Training Compiler
cl-tohoku/ -kata-masking bert-base-japanese-whole	ml.p3.2xlarge	12	24
cl-tohoku/ bert-base-japanese	ml.p3.2xlarge	12	24
distilbert-base-un cased	ml.p3.2xlarge	28	32
distilbert-base-un cased-finetuned-sst-2-bahasa Inggris	ml.p3.2xlarge	28	32
distilgpt2	ml.p3.2xlarge	16	32
facebook/bart-base	ml.p3.2xlarge	4	8
gpt2	ml.p3.2xlarge	6	20
Nreimers/minilmv2-l6-h384-suling-dari-roberta-besar	ml.p3.2xlarge	20	32
roberta-basis	ml.p3.2xlarge	12	20

Multi-GPU simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk asli	Ukuran Batch untuk Training Compiler
bert-base-chinese	ml.p3.8xlarge	16	26

Multi-GPU simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk asli	Ukuran Batch untuk Training Compiler
bert-base-multilingual-cased	ml.p3.8xlarge	6	16
bert-base-multilingual-uncased	ml.p3.8xlarge	6	16
bert-base-uncased	ml.p3.8xlarge	14	24
distilbert-base-uncased	ml.p3.8xlarge	14	32
distilgpt2	ml.p3.8xlarge	6	32
facebook/bart-base	ml.p3.8xlarge	8	16
gpt2	ml.p3.8xlarge	8	20
roberta-basis	ml.p3.8xlarge	12	20

Transformers 4.17.0 dengan 2.6.3 TensorFlow

Diuji dengan `Sequence_Len=128` dan Automatic Mixed Precision (AMP).

Model	Jenis instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
albert-base-v2	ml.g4dn.16xlarge	136	208
albert-base-v2	ml.g5.4xbesar	219	312
albert-base-v2	ml.p3.2xlarge	152	208
albert-base-v2	ml.p3.8xlarge	152	192
bert-base-uncased	ml.g4dn.16xlarge	120	101

Model	Jenis instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
bert-base-uncased	ml.g5.4xbesar	184	160
bert-base-uncased	ml.p3.2xlarge	128	108
bert-large-uncased	ml.g4dn.16xlarge	37	28
bert-large-uncased	ml.g5.4xbesar	64	55
bert-large-uncased	ml.p3.2xlarge	40	32
dasar camembert	ml.g4dn.16xlarge	96	100
dasar camembert	ml.g5.4xbesar	190	160
dasar camembert	ml.p3.2xlarge	129	108
dasar camembert	ml.p3.8xlarge	128	104
distilbert-base-uncased	ml.g4dn.16xlarge	210	160
distilbert-base-uncased	ml.g5.4xbesar	327	288
distilbert-base-uncased	ml.p3.2xlarge	224	196
distilbert-base-uncased	ml.p3.8xlarge	192	182
google_ electra-small-discriminator	ml.g4dn.16xlarge	336	288
google_ electra-small-discriminator	ml.g5.4xbesar	504	384
google_ electra-small-discriminator	ml.p3.2xlarge	352	323

Model	Jenis instans	Ukuran Batch untuk kerangka kerja asli	Ukuran Batch untuk Training Compiler
gpt2	ml.g4dn.16xlarge	89	64
gpt2	ml.g5.4xbesar	140	146
gpt2	ml.p3.2xlarge	94	96
gpt2	ml.p3.8xlarge	96	88
jplu_ tf-xlm-roberta-base	ml.g4dn.16xlarge	52	16
jplu_ tf-xlm-roberta-base	ml.g5.4xbesar	64	44
microsoft_mpnet-basis	ml.g4dn.16xlarge	120	100
microsoft_mpnet-basis	ml.g5.4xbesar	192	160
microsoft_mpnet-basis	ml.p3.2xlarge	128	104
microsoft_mpnet-basis	ml.p3.8xlarge	130	92
roberta-basis	ml.g4dn.16xlarge	108	64
roberta-basis	ml.g5.4xbesar	176	142
roberta-basis	ml.p3.2xlarge	118	100
roberta-basis	ml.p3.8xlarge	112	88

Transformers 4.11.0 dengan 2.5.1 TensorFlow

Diuji dengan `Sequence_Len=128` dan Automatic Mixed Precision (AMP).

GPU tunggal simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk asli	Ukuran Batch untuk Training Compiler
albert-base-v2	ml.p3.2xlarge	128	128
bart-dasar	ml.p3.2xlarge	12	64
bart-besar	ml.p3.2xlarge	4	28
bert-base-cased	ml.p3.2xlarge	16	128
bert-base-chinese	ml.p3.2xlarge	16	128
bert-base-multilingual-cased	ml.p3.2xlarge	12	64
bert-base-multilingual-uncased	ml.p3.2xlarge	16	96
bert-base-uncased	ml.p3.2xlarge	16	96
bert-large-uncased	ml.p3.2xlarge	4	24
cl-tohoku/ bert-base-japanese	ml.p3.2xlarge	16	128
cl-tohoku/ -kata-masking bert-base-japanese-whole	ml.p3.2xlarge	16	128
distilbert-base-sst2	ml.p3.2xlarge	32	128
distilbert-base-un cased	ml.p3.2xlarge	32	128
distilgpt2	ml.p3.2xlarge	32	128
gpt2	ml.p3.2xlarge	12	64

GPU tunggal simpul tunggal			
Model	Jenis instans	Ukuran Batch untuk asli	Ukuran Batch untuk Training Compiler
gpt2-besar	ml.p3.2xlarge	2	24
jplu/ tf-xlm-roberta-base	ml.p3.2xlarge	12	32
roberta-basis	ml.p3.2xlarge	4	64
roberta-besar	ml.p3.2xlarge	4	64
t5-dasar	ml.p3.2xlarge	64	64
t5-kecil	ml.p3.2xlarge	128	128

Bawa Model Pembelajaran Mendalam Anda Sendiri

Panduan ini memandu Anda melalui cara mengadaptasi skrip pelatihan Anda untuk pekerjaan pelatihan yang dipercepat kompiler. Persiapan skrip pelatihan Anda tergantung pada hal-hal berikut:

- Pengaturan pelatihan seperti pelatihan inti tunggal atau terdistribusi.
- Kerangka kerja dan pustaka yang Anda gunakan untuk membuat skrip pelatihan.

Pilih salah satu topik berikut tergantung pada kerangka kerja yang Anda gunakan.

Topik

- PyTorch
- TensorFlow

> **Note**
>
> Setelah Anda selesai menyiapkan skrip pelatihan Anda, Anda dapat menjalankan pekerjaan SageMaker pelatihan menggunakan kelas estimator SageMaker kerangka kerja. Untuk informasi lebih lanjut, lihat topik sebelumnya diAktifkan Kompiler SageMaker Pelatihan.

PyTorch

Bawa PyTorch model Anda sendiriSageMaker, dan jalankan pekerjaan pelatihan dengan Kompiler SageMaker Pelatihan.

Topik

- PyTorchModel dengan Memeluk Transformers Wajah

PyTorchModel dengan Memeluk Transformers Wajah

PyTorchmodel dengan Hugging Face Transformers didasarkan pada API PyTorch Torch.nn.Module. Memeluk Wajah Transformers juga menyediakan Trainer dan kelas model terlatih PyTorch untuk membantu mengurangi upaya untuk mengkonfigurasi model pengolahan bahasa alami (NLP). Setelah menyiapkan skrip pelatihan Anda, Anda dapat meluncurkan pekerjaan pelatihan menggunakan SageMaker `PyTorch` atau `HuggingFace` estimator dengan konfigurasi Kompiler SageMaker Pelatihan ketika Anda akan melanjutkan ke topik berikutnya di. Aktifkan Kompiler SageMaker Pelatihan

> **Tip**
>
> Saat Anda membuat tokenizer untuk model NLP menggunakan Transformers dalam skrip latihan Anda, pastikan Anda menggunakan bentuk tensor input statis dengan menentukan. `padding='max_length'` Jangan gunakan `padding='longest'` karena padding ke urutan terpanjang dalam batch dapat mengubah bentuk tensor untuk setiap batch pelatihan. Bentuk input dinamis dapat memicu kompilasi ulang model dan dapat meningkatkan total waktu pelatihan. Untuk informasi lebih lanjut tentang opsi padding dari tokenizer Transformers, lihat Padding dan pemotongan dalam dokumentasi Hugging Face Transformers.

Topik

- Model Bahasa Besar Menggunakan Kelas Memeluk Wajah Transformers Trainer
- Model Bahasa Besar Menggunakan PyTorch Langsung (tanpa API Pelatih Transformers Wajah Memeluk)

Model Bahasa Besar Menggunakan Kelas Memeluk Wajah Transformers `Trainer`

Jika Anda menggunakan kelas Trainer library transformer, Anda tidak perlu membuat perubahan tambahan pada skrip latihan Anda. SageMakerTraining Compiler secara otomatis mengkompilasi model Trainer Anda jika Anda mengaktifkannya melalui kelas estimator. Kode berikut menunjukkan bentuk dasar skrip PyTorch pelatihan dengan API Hugging Face Trainer.

```
from transformers import Trainer, TrainingArguments

training_args=TrainingArguments(**kwargs)
trainer=Trainer(args=training_args, **kwargs)
```

Topik

- Untuk pelatihan GPU tunggal
- Untuk pelatihan terdistribusi
- Praktik Terbaik untuk Menggunakan Kompiler SageMaker Pelatihan dengan Trainer

Untuk pelatihan GPU tunggal

Anda tidak perlu mengubah kode Anda ketika Anda menggunakan `transformers.Trainer`kelas.

Untuk pelatihan terdistribusi

PyTorchv1.11.0 dan kemudian

Untuk menjalankan pelatihan terdistribusi dengan SageMaker Training Compiler, Anda harus menambahkan `_mp_fn()` fungsi berikut dalam skrip latihan Anda dan membungkus `main()` fungsinya. Ini mengalihkan panggilan `_mp_fn(index)` fungsi dari runtime SageMaker terdistribusi for PyTorch (`pytorchxla`) ke `main()` fungsi skrip pelatihan Anda.

```
def _mp_fn(index):
    main()
```

Fungsi ini menerima `index` argumen untuk menunjukkan peringkat GPU saat ini di cluster untuk pelatihan terdistribusi. Untuk menemukan lebih banyak contoh skrip, lihat skrip contoh pemodelan bahasa Hugging Face Transformers.

Untuk Transformers v4.17 dan sebelumnya dengan v1.10.2 dan sebelumnya PyTorch

SageMakerKompiler Pelatihan menggunakan mekanisme alternatif untuk meluncurkan pekerjaan pelatihan terdistribusi, dan Anda tidak perlu melakukan modifikasi apa pun dalam skrip pelatihan Anda. Sebagai gantinya, Kompiler SageMaker Pelatihan mengharuskan Anda untuk meneruskan skrip peluncur pelatihan SageMaker terdistribusi ke `entry_point` argumen dan meneruskan skrip pelatihan Anda ke `hyperparameters` argumen di penaksir Wajah SageMaker Pelukan.

Praktik Terbaik untuk Menggunakan Kompiler SageMaker Pelatihan dengan **`Trainer`**

- Pastikan Anda menggunakan SyncFree pengoptimal dengan mengatur `optim` argumen ke `adamw_torch_xla` saat menyiapkan transformator. TrainingArgument. Lihat juga Optimizer dalam dokumentasi Hugging Face Transformers.
- Pastikan throughput pipa pemrosesan data lebih tinggi daripada throughput pelatihan. Anda dapat men-tweak `dataloader_num_workers` dan `preprocessing_num_workers` argumen dari transformer. TrainingArgumentkelas untuk mencapai hal ini. Biasanya, ini harus lebih besar dari atau sama dengan jumlah GPU tetapi kurang dari jumlah CPU.

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan kethe section called “Jalankan Pekerjaan PyTorch Pelatihan dengan Kompiler Pelatihan”.

Model Bahasa Besar Menggunakan PyTorch Langsung (tanpa API Pelatih Transformers Wajah Memeluk)

Jika Anda memiliki skrip pelatihan yang menggunakan PyTorch secara langsung, Anda perlu membuat perubahan tambahan pada skrip PyTorch pelatihan Anda untuk mengimplementasikan PyTorch /XLA. Ikuti petunjuk untuk memodifikasi skrip Anda untuk mengatur primatif PyTorch /XLA dengan benar.

Topik

- Untuk pelatihan GPU tunggal
- Untuk pelatihan terdistribusi
- Praktik Terbaik untuk Menggunakan Kompiler SageMaker Pelatihan PyTorch dengan/XLA

Untuk pelatihan GPU tunggal

1. Impor perpustakaan optimasi.

```
import torch_xla
import torch_xla.core.xla_model as xm
```

2. Ubah perangkat target menjadi XLA, bukan `torch.device("cuda")`

```
device=xm.xla_device()
```

3. Jika Anda menggunakan PyTorch Automatic Mixed Precision (AMP), lakukan hal berikut:

   a. Ganti `torch.cuda.amp` dengan yang berikut ini:

   ```
   import torch_xla.amp
   ```

   b. Ganti `torch.optim.SGD` dan `torch.optim.Adam` dengan yang berikut ini:

   ```
   import torch_xla.amp.syncfree.Adam as adam
   import torch_xla.amp.syncfree.SGD as SGD
   ```

   c. Ganti `torch.cuda.amp.GradScaler` dengan yang berikut ini:

   ```
   import torch_xla.amp.GradScaler as grad_scaler
   ```

4. Jika Anda tidak menggunakan AMP, ganti `optimizer.step()` dengan yang berikut:

```
xm.optimizer_step(optimizer)
```

5. Jika Anda menggunakan dataloader terdistribusi, bungkus dataloader Anda di kelas /XLA: PyTorch `ParallelLoader`

```
import torch_xla.distributed.parallel_loader as pl
parallel_loader=pl.ParallelLoader(dataloader, [device]).per_device_loader(device)
```

6. Tambahkan `mark_step` di akhir loop latihan saat Anda tidak menggunakan`parallel_loader`:

```
xm.mark_step()
```

7. Untuk memeriksa pelatihan Anda, gunakan metode pos pemeriksaan model PyTorch /XLA:

```
xm.save(model.state_dict(), path_to_save)
```

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan kethe section called “Jalankan Pekerjaan PyTorch Pelatihan dengan Kompiler Pelatihan”.

Untuk pelatihan terdistribusi

Selain perubahan yang tercantum di Untuk pelatihan GPU tunggal bagian sebelumnya, tambahkan perubahan berikut untuk mendistribusikan beban kerja dengan benar di seluruh GPU.

1. Jika Anda menggunakan AMP, tambahkan `all_reduce` setelah`scaler.scale(loss).backward()`:

```
gradients=xm._fetch_gradients(optimizer)
xm.all_reduce('sum', gradients, scale=1.0/xm.xrt_world_size())
```

2. Jika Anda perlu mengatur variabel untuk `local_ranks` dan`world_size`, gunakan kode serupa dengan yang berikut:

```
local_rank=xm.get_local_ordinal()
world_size=xm.xrt_world_size()
```

3. Untuk setiap `world_size` (`num_gpus_per_node*num_nodes`) lebih besar dari1, Anda harus menentukan sampler kereta api yang seharusnya terlihat mirip dengan berikut ini:

```
import torch_xla.core.xla_model as xm

if xm.xrt_world_size() > 1:
    train_sampler=torch.utils.data.distributed.DistributedSampler(
        train_dataset,
        num_replicas=xm.xrt_world_size(),
        rank=xm.get_ordinal(),
        shuffle=True
    )

train_loader=torch.utils.data.DataLoader(
    train_dataset,
    batch_size=args.batch_size,
    sampler=train_sampler,
    drop_last=args.drop_last,
```

```
    shuffle=False if train_sampler else True,
    num_workers=args.num_workers
)
```

4. Buat perubahan berikut untuk memastikan Anda menggunakan yang `parallel_loader` disediakan oleh `torch_xla distributed` modul.

```
import torch_xla.distributed.parallel_loader as pl
train_device_loader=pl.MpDeviceLoader(train_loader, device)
```

`train_device_loader`Fungsi seperti PyTorch loader biasa sebagai berikut:

```
for step, (data, target) in enumerate(train_device_loader):
    optimizer.zero_grad()
    output=model(data)
    loss=torch.nn.NLLLoss(output, target)
    loss.backward()
```

Dengan semua perubahan ini, Anda harus dapat meluncurkan pelatihan terdistribusi dengan PyTorch model apa pun tanpa Transformer Trainer API. Perhatikan bahwa petunjuk ini dapat digunakan untuk multi-GPU single-node dan multi-node multi-GPU.

5. Untuk PyTorch v1.11.0 dan yang lebih baru

Untuk menjalankan pelatihan terdistribusi dengan SageMaker Training Compiler, Anda harus menambahkan `_mp_fn()` fungsi berikut dalam skrip latihan Anda dan membungkus `main()` fungsinya. Ini mengalihkan panggilan `_mp_fn(index)` fungsi dari runtime SageMaker terdistribusi for PyTorch (`pytorchxla`) ke `main()` fungsi skrip pelatihan Anda.

```
def _mp_fn(index):
    main()
```

Fungsi ini menerima `index` argumen untuk menunjukkan peringkat GPU saat ini di cluster untuk pelatihan terdistribusi. Untuk menemukan lebih banyak contoh skrip, lihat skrip contoh pemodelan bahasa Hugging Face Transformers.

Untuk Transformers v4.17 dan sebelumnya dengan v1.10.2 dan sebelumnya PyTorch

SageMakerKompiler Pelatihan menggunakan mekanisme alternatif untuk meluncurkan pekerjaan pelatihan terdistribusi dan mengharuskan Anda untuk meneruskan skrip peluncur pelatihan

SageMaker terdistribusi ke argumen dan meneruskan skrip pelatihan Anda ke `entry_point` argumen dalam `hyperparameters` penaksir Wajah Pelukan. SageMaker

Setelah Anda selesai mengadaptasi skrip pelatihan Anda, lanjutkan kethe section called “Jalankan Pekerjaan PyTorch Pelatihan dengan Kompiler Pelatihan”.

Praktik Terbaik untuk Menggunakan Kompiler SageMaker Pelatihan PyTorch dengan/XLA

Jika Anda ingin memanfaatkan Kompiler SageMaker Pelatihan pada skrip PyTorch pelatihan asli Anda, Anda mungkin ingin terlebih dahulu membiasakan diri dengan perangkat PyTorchXLA. Bagian berikut mencantumkan beberapa praktik terbaik untuk mengaktifkan XLA. PyTorch

> **Note**
>
> Bagian ini untuk praktik terbaik mengasumsikan bahwa Anda menggunakan modul PyTorch / XLA berikut:
>
> ```
> import torch_xla.core.xla_model as xm
> import torch_xla.distributed.parallel_loader as pl
> ```

Memahami mode malas PyTorch di/XLA

Satu perbedaan yang signifikan antara PyTorch /XLA dan native PyTorch adalah bahwa PyTorch sistem/XLA berjalan dalam mode malas sementara native PyTorch berjalan dalam mode eager. Tensor dalam mode malas adalah placeholder untuk membangun grafik komputasi sampai terwujud setelah kompilasi dan evaluasi selesai. Sistem PyTorch /XLA membuat grafik komputasi dengan cepat saat Anda memanggil PyTorch API untuk membuat komputasi menggunakan tensor dan operator. Grafik komputasi dikompilasi dan dijalankan saat `xm.mark_step()` dipanggil secara eksplisit atau implisit olehpl.`MpDeviceLoader`/`pl.ParallelLoader`, atau ketika Anda secara eksplisit meminta nilai tensor seperti dengan menelepon atau. `loss.item()` `print(loss)`

Minimalkan jumlah compilation-and-executionspenggunaan **`pl.MpDeviceLoader/ pl.ParallelLoader`** dan **`xm.step_closure`**

Untuk kinerja terbaik, Anda harus diingat cara yang mungkin untuk memulai compilation-and-executionsseperti yang dijelaskan dalam Memahami mode malas PyTorch di/XLA dan harus mencoba untuk meminimalkan jumlahcompilation-and-executions. Idealnya, hanya satu compilation-and-execution yang diperlukan per iterasi pelatihan dan dimulai secara otomatis oleh.

`pl.MpDeviceLoader/pl.ParallelLoader` `MpDeviceLoader`Dioptimalkan untuk XLA dan harus selalu digunakan jika memungkinkan untuk kinerja terbaik. Selama pelatihan, Anda mungkin ingin memeriksa beberapa hasil menengah seperti nilai kerugian. Dalam hal seperti itu, pencetakan tensor malas harus dibungkus menggunakan `xm.add_step_closure()` untuk menghindari yang tidak perlu. compilation-and-executions

Menggunakan AMP dan **`syncfree`** pengoptimal

Pelatihan dalam mode Automatic Mixed Precision (AMP) secara signifikan mempercepat kecepatan latihan Anda dengan memanfaatkan inti Tensor GPU NVIDIA. SageMakerTraining Compiler menyediakan `syncfree` pengoptimal yang dioptimalkan untuk XLA guna meningkatkan kinerja AMP. Saat ini, tiga `syncfree` pengoptimal berikut tersedia dan harus digunakan jika memungkinkan untuk kinerja terbaik.

```
torch_xla.amp.syncfree.SGD
torch_xla.amp.syncfree.Adam
torch_xla.amp.syncfree.AdamW
```

`syncfree`Pengoptimal ini harus dipasangkan dengan `torch_xla.amp.GradScaler` untuk penskalaan gradien/unscaling.

> **Tip**
>
> Mulai PyTorch 1.13.1, SageMaker Training Compiler meningkatkan kinerja dengan PyTorch membiarkan/XLA untuk secara otomatis menimpa pengoptimal (seperti SGD, Adam, ADAMW) di `torch.optim` atau `transformers.optimization` dengan versi syncfree dari mereka di (seperti,,). `torch_xla.amp.syncfree` `torch_xla.amp.syncfree.SGD` `torch_xla.amp.syncfree.Adam` `torch_xla.amp.syncfree.AdamW` Anda tidak perlu mengubah baris kode di mana Anda menentukan pengoptimal dalam skrip pelatihan Anda.

TensorFlow

Bawa TensorFlow model Anda sendiriSageMaker, dan jalankan pekerjaan pelatihan dengan SageMaker Training Compiler.

TensorFlowModel

SageMakerTraining Compiler secara otomatis mengoptimalkan beban kerja pelatihan model yang dibangun di atas TensorFlow API asli atau API Keras tingkat tinggi.

> **Tip**
>
> Untuk preprocessing dataset input Anda, pastikan bahwa Anda menggunakan bentuk input statis. Bentuk input dinamis dapat memulai kompilasi ulang model dan dapat meningkatkan total waktu pelatihan.

Menggunakan Keras (Direkomendasikan)

Untuk akselerasi kompiler terbaik, sebaiknya gunakan model yang merupakan subclass TensorFlow Keras (tf.keras.model).

Untuk pelatihan GPU tunggal

Tidak ada perubahan tambahan yang perlu Anda buat dalam skrip pelatihan.

Tanpa Keras

SageMakerPelatihan Compiler tidak mendukung eksekusi bersemangat di. TensorFlow Dengan demikian, Anda harus membungkus model dan loop pelatihan Anda dengan dekorator TensorFlow fungsi (`@tf.function`) untuk memanfaatkan akselerasi kompiler.

SageMakerTraining Compiler melakukan optimasi tingkat grafik, dan menggunakan dekorator untuk memastikan TensorFlow fungsi Anda diatur untuk berjalan dalam mode grafik.

Untuk pelatihan GPU tunggal

TensorFlow2.0 atau yang lebih baru memiliki eksekusi yang bersemangat secara default, jadi Anda harus menambahkan `@tf.function` dekorator di depan setiap fungsi yang Anda gunakan untuk membuat model. TensorFlow

TensorFlowModel dengan Memeluk Transformers Wajah

TensorFlowmodel dengan Memeluk Transformers Wajah didasarkan pada tf.keras.Model APITensorFlow. Memeluk Wajah Transformers juga menyediakan kelas model terlatih TensorFlow untuk membantu mengurangi upaya untuk mengkonfigurasi pengolahan bahasa alami (NLP) model. Setelah membuat skrip pelatihan Anda sendiri menggunakan perpustakaan Transformers, Anda dapat menjalankan skrip pelatihan menggunakan SageMaker `HuggingFace` estimator dengan kelas konfigurasi SageMaker Training Compiler seperti yang ditunjukkan pada topik sebelumnya di. Jalankan Pekerjaan TensorFlow Pelatihan dengan Kompiler SageMaker Pelatihan

SageMakerTraining Compiler secara otomatis mengoptimalkan beban kerja pelatihan model yang dibangun di atas TensorFlow API asli atau API Keras tingkat tinggi, seperti model transformator. TensorFlow

> **Tip**
>
> Saat Anda membuat tokenizer untuk model NLP menggunakan Transformers dalam skrip latihan Anda, pastikan Anda menggunakan bentuk tensor input statis dengan menentukan. `padding='max_length'` Jangan gunakan `padding='longest'` karena padding ke urutan terpanjang dalam batch dapat mengubah bentuk tensor untuk setiap batch pelatihan. Bentuk input dinamis dapat memulai kompilasi ulang model dan dapat meningkatkan total waktu pelatihan. Untuk informasi lebih lanjut tentang opsi padding dari tokenizer Transformers, lihat Padding dan pemotongan dalam dokumentasi Hugging Face Transformers.

Topik

- Menggunakan Keras
- Tanpa Keras

Menggunakan Keras

Untuk akselerasi kompiler terbaik, sebaiknya gunakan model yang merupakan subclass TensorFlow Keras (tf.keras.model). Seperti yang tercantum di halaman tur Cepat dalam dokumentasi Hugging Face Transformers, Anda dapat menggunakan model sebagai model Keras biasaTensorFlow.

Untuk pelatihan GPU tunggal

Tidak ada perubahan tambahan yang perlu Anda buat dalam skrip pelatihan.

Untuk pelatihan terdistribusi

SageMakerAkselerasi Training Compiler bekerja secara transparan untuk beban kerja multi-GPU saat model dibuat dan dilatih menggunakan API Keras dalam lingkup panggilan. `tf.distribute.Strategy.scope()`

1. Pilih strategi pelatihan terdistribusi yang tepat.
   a. Untuk single-node multi-GPU, gunakan `tf.distribute.MirroredStrategy` untuk mengatur strategi.

```
strategy = tf.distribute.MirroredStrategy()
```

b. Untuk multi-node multi-GPU, tambahkan kode berikut untuk mengatur konfigurasi pelatihan TensorFlow terdistribusi dengan benar sebelum membuat strategi.

```
def set_sm_dist_config():
    DEFAULT_PORT = '8890'
    DEFAULT_CONFIG_FILE = '/opt/ml/input/config/resourceconfig.json'
    with open(DEFAULT_CONFIG_FILE) as f:
        config = json.loads(f.read())
        current_host = config['current_host']
    tf_config = {
        'cluster': {
            'worker': []
        },
        'task': {'type': 'worker', 'index': -1}
    }
    for i, host in enumerate(config['hosts']):
        tf_config['cluster']['worker'].append("%s:%s" % (host, DEFAULT_PORT))
        if current_host == host:
            tf_config['task']['index'] = i
    os.environ['TF_CONFIG'] = json.dumps(tf_config)

set_sm_dist_config()
```

Gunakan `tf.distribute.MultiWorkerMirroredStrategy` untuk mengatur strategi.

```
strategy = tf.distribute.MultiWorkerMirroredStrategy()
```

2. Menggunakan strategi pilihan Anda, bungkus model.

```
with strategy.scope():
    # create a model and do fit
```

Tanpa Keras

Jika Anda ingin membawa model khusus dengan loop pelatihan khusus menggunakan TensorFlow tanpa Keras, Anda harus membungkus model dan loop pelatihan dengan dekorator TensorFlow fungsi (`@tf.function`) untuk memanfaatkan akselerasi kompiler.

SageMakerTraining Compiler melakukan optimasi tingkat grafik, dan menggunakan dekorator untuk memastikan TensorFlow fungsi Anda diatur untuk berjalan dalam mode grafik.

Untuk pelatihan GPU tunggal

TensorFlow2.0 atau yang lebih baru memiliki eksekusi yang bersemangat secara default, jadi Anda harus menambahkan `@tf.function` dekorator di depan setiap fungsi yang Anda gunakan untuk membuat model. TensorFlow

Untuk pelatihan terdistribusi

Selain perubahan yang diperlukan untuk Menggunakan Keras untuk pelatihan terdistribusi, Anda perlu memastikan bahwa fungsi yang akan dijalankan pada setiap GPU diberi anotasi`@tf.function`, sementara fungsi komunikasi lintas GPU tidak diberi anotasi. Contoh kode pelatihan akan terlihat seperti berikut ini:

```
@tf.function()
def compiled_step(inputs, outputs):
    with tf.GradientTape() as tape:
        pred=model(inputs, training=True)
        total_loss=loss_object(outputs, pred)/args.batch_size
    gradients=tape.gradient(total_loss, model.trainable_variables)
    return total_loss, pred, gradients

def train_step(inputs, outputs):
    total_loss, pred, gradients=compiled_step(inputs, outputs)
    if args.weight_decay > 0.:
        gradients=[g+v*args.weight_decay for g,v in zip(gradients,
 model.trainable_variables)]

    optimizer.apply_gradients(zip(gradients, model.trainable_variables))

    train_loss.update_state(total_loss)
    train_accuracy.update_state(outputs, pred)

@tf.function()
def train_step_dist(inputs, outputs):
    strategy.run(train_step, args= (inputs, outputs))
```

Perhatikan bahwa instruksi ini dapat digunakan untuk multi-GPU single-node dan multi-node multi-GPU.

Aktifkan Kompiler SageMaker Pelatihan

SageMaker Training Compiler dibangun ke dalam SageMaker Python SDK dan Deep Learning AWS Containers sehingga Anda tidak perlu mengubah alur kerja untuk mengaktifkan Training Compiler. Pilih salah satu topik berikut yang cocok dengan kasus penggunaan Anda.

Topik

Jalankan Pekerjaan PyTorch Pelatihan dengan Kompiler SageMaker Pelatihan

Anda dapat menggunakan salah satu SageMaker antarmuka untuk menjalankan pekerjaan pelatihan dengan SageMaker Training Compiler: Amazon SageMaker Studio Classic, instans SageMaker notebook Amazon, AWS SDK for Python (Boto3) dan. AWS Command Line Interface

Topik

Menggunakan SageMaker Python SDK

SageMaker Training Compiler untuk PyTorch tersedia melalui kelas estimator SageMaker `PyTorch`dan `HuggingFace`framework. Untuk mengaktifkan SageMaker Training Compiler, tambahkan `compiler_config` parameter ke SageMaker estimator. Impor `TrainingCompilerConfig` kelas dan berikan instance ke `compiler_config` parameter. Contoh kode berikut menunjukkan struktur kelas SageMaker estimator dengan SageMaker Training Compiler diaktifkan.

> **Tip**
>
> Untuk memulai dengan model prebuilt yang disediakan oleh PyTorch atau Transformers, coba gunakan ukuran batch yang disediakan dalam tabel referensi di. Model yang Diuji

> **Note**
>
> PyTorch Dukungan asli tersedia di SageMaker Python SDK v2.121.0 dan yang lebih baru. Pastikan Anda memperbarui SDK SageMaker Python yang sesuai.

> **Note**
>
> Mulai PyTorch v1.12.0, wadah SageMaker Training Compiler untuk PyTorch tersedia. Perhatikan bahwa wadah SageMaker Training Compiler untuk tidak PyTorch dikemas dengan Hugging Face Transformers. Jika Anda perlu menginstal pustaka dalam wadah, pastikan Anda menambahkan `requirements.txt` file di bawah direktori sumber saat mengirimkan pekerjaan pelatihan.
> Untuk PyTorch v1.11.0 dan sebelumnya, gunakan versi sebelumnya dari wadah SageMaker Training Compiler untuk Hugging Face dan. PyTorch
> Untuk daftar lengkap versi kerangka kerja dan informasi kontainer yang sesuai, lihatthe section called “Kerangka Kerja yang Didukung”.

Untuk informasi yang sesuai dengan kasus penggunaan Anda, lihat salah satu opsi berikut.

Untuk pelatihan GPU tunggal

PyTorch v1.12.0 and later

Untuk mengkompilasi dan melatih PyTorch model, konfigurasikan SageMaker PyTorch estimator dengan SageMaker Training Compiler seperti yang ditunjukkan pada contoh kode berikut.

> **Note**
>
> PyTorch Dukungan asli ini tersedia di SageMaker Python SDK v2.120.0 dan yang lebih baru. Pastikan Anda memperbarui SDK SageMaker Python.

```
from sagemaker.pytorch import PyTorch, TrainingCompilerConfig

# the original max batch size that can fit into GPU memory without compiler
batch_size_native=12
learning_rate_native=float('5e-5')
```

```
# an updated max batch size that can fit into GPU memory with compiler
batch_size=64

# update learning rate
learning_rate=learning_rate_native/batch_size_native*batch_size

hyperparameters={
    "n_gpus": 1,
    "batch_size": batch_size,
    "learning_rate": learning_rate
}

pytorch_estimator=PyTorch(
    entry_point='train.py',
    source_dir='path-to-requirements-file', # Optional. Add this if need to install
 additional packages.
    instance_count=1,
    instance_type='ml.p3.2xlarge',
    framework_version='1.13.1',
    py_version='py3',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    disable_profiler=True,
    debugger_hook_config=False
)

pytorch_estimator.fit()
```

Hugging Face Transformers with PyTorch v1.11.0 and before

Untuk mengkompilasi dan melatih model transformator PyTorch, konfigurasikan estimator SageMaker Hugging Face dengan SageMaker Training Compiler seperti yang ditunjukkan pada contoh kode berikut.

```
from sagemaker.huggingface import HuggingFace, TrainingCompilerConfig

# the original max batch size that can fit into GPU memory without compiler
batch_size_native=12
learning_rate_native=float('5e-5')

# an updated max batch size that can fit into GPU memory with compiler
batch_size=64
```

```
# update learning rate
learning_rate=learning_rate_native/batch_size_native*batch_size

hyperparameters={
    "n_gpus": 1,
    "batch_size": batch_size,
    "learning_rate": learning_rate
}

pytorch_huggingface_estimator=HuggingFace(
    entry_point='train.py',
    instance_count=1,
    instance_type='ml.p3.2xlarge',
    transformers_version='4.21.1',
    pytorch_version='1.11.0',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    disable_profiler=True,
    debugger_hook_config=False
)

pytorch_huggingface_estimator.fit()
```

Untuk menyiapkan skrip pelatihan Anda, lihat halaman-halaman berikut.

- Untuk pelatihan GPU tunggal PyTorch model yang menggunakan API Trainer Hugging Face Transformers
- Untuk pelatihan GPU tunggal PyTorch model tanpa API Trainer Hugging Face Transformers

Untuk menemukan end-to-end contoh, lihat buku catatan berikut:

- Kompilasi dan Latih Model Trainer Trainer Hugging Face Transformers untuk Pertanyaan dan Jawaban dengan dataset sQUAD
- Kompilasi dan Latih BERT Model Trafo Hugging Face dengan Dataset SST menggunakan Training Compiler SageMaker
- Kompilasi dan Latih Model Pelatih Klasifikasi Biner dengan Dataset SST2 untuk Pelatihan GPU Tunggal Node Tunggal

Untuk pelatihan terdistribusi

PyTorch v1.12

Untuk PyTorch v1.12, Anda dapat menjalankan pelatihan terdistribusi dengan SageMaker Training Compiler dengan menambahkan `pytorch_xla` opsi yang ditentukan ke `distribution` parameter kelas SageMaker PyTorch estimator.

> **Note**
>
> PyTorch Dukungan asli ini tersedia di SageMaker Python SDK v2.121.0 dan yang lebih baru. Pastikan Anda memperbarui SDK SageMaker Python.

```
from sagemaker.pytorch import PyTorch, TrainingCompilerConfig

# choose an instance type, specify the number of instances you want to use,
# and set the num_gpus variable the number of GPUs per instance.
instance_count=1
instance_type='ml.p3.8xlarge'
num_gpus=4

# the original max batch size that can fit to GPU memory without compiler
batch_size_native=16
learning_rate_native=float('5e-5')

# an updated max batch size that can fit to GPU memory with compiler
batch_size=26

# update learning rate
learning_rate=learning_rate_native/
batch_size_native*batch_size*num_gpus*instance_count

hyperparameters={
    "n_gpus": num_gpus,
    "batch_size": batch_size,
    "learning_rate": learning_rate
}

pytorch_estimator=PyTorch(
    entry_point='your_training_script.py',
```

```
    source_dir='path-to-requirements-file', # Optional. Add this if need to install
 additional packages.
    instance_count=instance_count,
    instance_type=instance_type,
    framework_version='1.13.1',
    py_version='py3',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    distribution ={'pytorchxla' : { 'enabled': True }},
    disable_profiler=True,
    debugger_hook_config=False
)

pytorch_estimator.fit()
```

> **Tip**
>
> Untuk mempersiapkan naskah pelatihan Anda, lihat PyTorch

Transformers v4.21 with PyTorch v1.11

Untuk PyTorch v1.11 dan yang lebih baru, SageMaker Training Compiler tersedia untuk pelatihan terdistribusi dengan `pytorch_xla` opsi yang ditentukan untuk parameter. `distribution`

```
from sagemaker.huggingface import HuggingFace, TrainingCompilerConfig

# choose an instance type, specify the number of instances you want to use,
# and set the num_gpus variable the number of GPUs per instance.
instance_count=1
instance_type='ml.p3.8xlarge'
num_gpus=4

# the original max batch size that can fit to GPU memory without compiler
batch_size_native=16
learning_rate_native=float('5e-5')

# an updated max batch size that can fit to GPU memory with compiler
batch_size=26

# update learning rate
```

```
learning_rate=learning_rate_native/
batch_size_native*batch_size*num_gpus*instance_count

hyperparameters={
    "n_gpus": num_gpus,
    "batch_size": batch_size,
    "learning_rate": learning_rate
}

pytorch_huggingface_estimator=HuggingFace(
    entry_point='your_training_script.py',
    instance_count=instance_count,
    instance_type=instance_type,
    transformers_version='4.21.1',
    pytorch_version='1.11.0',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    distribution ={'pytorchxla' : { 'enabled': True }},
    disable_profiler=True,
    debugger_hook_config=False
)

pytorch_huggingface_estimator.fit()
```

Tip

Untuk menyiapkan skrip pelatihan Anda, lihat halaman-halaman berikut.

- Untuk pelatihan terdistribusi PyTorch model yang menggunakan API Trainer Hugging Face Transformers
- Untuk pelatihan terdistribusi PyTorch model tanpa API Trainer Hugging Face Transformers

Transformers v4.17 with PyTorch v1.10.2 and before

Untuk versi PyTorch v1.10.2 dan sebelumnya yang didukung, SageMaker Training Compiler memerlukan mekanisme alternatif untuk meluncurkan pekerjaan pelatihan terdistribusi. Untuk menjalankan pelatihan terdistribusi, SageMaker Training Compiler mengharuskan Anda untuk meneruskan skrip peluncur pelatihan SageMaker terdistribusi ke `entry_point` argumen, dan meneruskan skrip pelatihan Anda ke argumen. `hyperparameters` Contoh kode berikut

menunjukkan cara mengonfigurasi estimator SageMaker Hugging Face yang menerapkan perubahan yang diperlukan.

```
from sagemaker.huggingface import HuggingFace, TrainingCompilerConfig

# choose an instance type, specify the number of instances you want to use,
# and set the num_gpus variable the number of GPUs per instance.
instance_count=1
instance_type='ml.p3.8xlarge'
num_gpus=4

# the original max batch size that can fit to GPU memory without compiler
batch_size_native=16
learning_rate_native=float('5e-5')

# an updated max batch size that can fit to GPU memory with compiler
batch_size=26

# update learning rate
learning_rate=learning_rate_native/
batch_size_native*batch_size*num_gpus*instance_count

training_script="your_training_script.py"

hyperparameters={
    "n_gpus": num_gpus,
    "batch_size": batch_size,
    "learning_rate": learning_rate,
    "training_script": training_script     # Specify the file name of your training
 script.
}

pytorch_huggingface_estimator=HuggingFace(
    entry_point='distributed_training_launcher.py',    # Specify the distributed
 training launcher script.
    instance_count=instance_count,
    instance_type=instance_type,
    transformers_version='4.17.0',
    pytorch_version='1.10.2',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    disable_profiler=True,
    debugger_hook_config=False
```

```
)

pytorch_huggingface_estimator.fit()
```

Skrip peluncur akan terlihat seperti berikut ini. Ini membungkus skrip pelatihan Anda dan mengonfigurasi lingkungan pelatihan terdistribusi tergantung pada ukuran contoh pelatihan pilihan Anda.

```
# distributed_training_launcher.py

#!/bin/python

import subprocess
import sys

if __name__ == "__main__":
    arguments_command = " ".join([arg for arg in sys.argv[1:]])
    """
    The following line takes care of setting up an inter-node communication
    as well as managing intra-node workers for each GPU.
    """
    subprocess.check_call("python -m torch_xla.distributed.sm_dist " +
 arguments_command, shell=True)
```

Tip

Untuk menyiapkan skrip pelatihan Anda, lihat halaman-halaman berikut.

- Untuk pelatihan terdistribusi PyTorch model yang menggunakan API Trainer Hugging Face Transformers
- Untuk pelatihan terdistribusi PyTorch model tanpa API Trainer Hugging Face Transformers

Tip

Untuk menemukan end-to-end contoh, lihat buku catatan berikut:

- Kompilasi dan Latih Model GPT2 menggunakan Transformers Trainer API dengan Dataset SST2 untuk Pelatihan Multi-GPU Single-Node

- Kompilasi dan Latih Model GPT2 menggunakan Transformers Trainer API dengan Dataset SST2 untuk Pelatihan Multi-GPU Multi-Node

Daftar berikut adalah set minimal parameter yang diperlukan untuk menjalankan pekerjaan SageMaker pelatihan dengan compiler.

> **Note**
>
> Saat menggunakan estimator SageMaker Hugging Face, Anda harus menentukan`transformers_version`,, `pytorch_version`hyperparameters, `compiler_config` dan parameter untuk SageMaker mengaktifkan Training Compiler. Anda tidak dapat menggunakan `image_uri` untuk secara manual menentukan Deep Learning Containers terintegrasi Training Compiler yang terdaftar diKerangka Kerja yang Didukung.

- `entry_point`(str) - Diperlukan. Tentukan nama file skrip pelatihan Anda.

  > **Note**
  >
  > Untuk menjalankan pelatihan terdistribusi dengan SageMaker Training Compiler dan PyTorch v1.10.2 dan sebelumnya, tentukan nama file skrip peluncur ke parameter ini. Skrip peluncur harus disiapkan untuk membungkus skrip pelatihan Anda dan mengonfigurasi lingkungan pelatihan terdistribusi. Untuk informasi selengkapnya, lihat contoh buku catatan berikut:
  >
  > - Kompilasi dan Latih Model GPT2 menggunakan Transformers Trainer API dengan Dataset SST2 untuk Pelatihan Multi-GPU Single-Node
  > - Kompilasi dan Latih Model GPT2 menggunakan Transformers Trainer API dengan Dataset SST2 untuk Pelatihan Multi-GPU Multi-Node

- `source_dir`(str) - Opsional. Tambahkan ini jika perlu menginstal paket tambahan. Untuk menginstal paket, Anda perlu prapare `requirements.txt` file di bawah direktori ini.
- `instance_count`(int) - Diperlukan. Tentukan jumlah instans.
- `instance_type`(str) - Diperlukan. Tentukan jenis instance.
- `transformers_version`(str) - Diperlukan hanya saat menggunakan estimator SageMaker Hugging Face. Tentukan versi pustaka Hugging Face Transformers yang didukung oleh

SageMaker Training Compiler. Untuk menemukan versi yang tersedia, lihatKerangka Kerja yang Didukung.

- `framework_version`atau `pytorch_version` (str) - Diperlukan. Tentukan PyTorch versi yang didukung oleh SageMaker Training Compiler. Untuk menemukan versi yang tersedia, lihatKerangka Kerja yang Didukung.

  > **Note**
  >
  > Saat menggunakan estimator SageMaker Hugging Face, Anda harus menentukan `transformers_version` keduanya dan. `pytorch_version`

- `hyperparameters`(dict) — Opsional. Tentukan hyperparameters untuk pekerjaan pelatihan, sepertin_gpus,`batch_size`, dan`learning_rate`. Saat Anda mengaktifkan SageMaker Training Compiler, coba ukuran batch yang lebih besar dan sesuaikan tingkat pembelajaran yang sesuai. Untuk menemukan studi kasus penggunaan kompiler dan ukuran batch yang disesuaikan untuk meningkatkan kecepatan pelatihan, lihat the section called “Model yang Diuji” danSageMaker Contoh Kompiler Pelatihan Notebook dan Blog.

  > **Note**
  >
  > Untuk menjalankan pelatihan terdistribusi dengan SageMaker Training Compiler dan PyTorch v1.10.2 dan sebelumnya, Anda perlu menambahkan parameter tambahan,`"training_script"`, untuk menentukan skrip pelatihan Anda, seperti yang ditunjukkan pada contoh kode sebelumnya.

- `compiler_config`(TrainingCompilerConfig objek) - Diperlukan untuk mengaktifkan SageMaker Training Compiler. Sertakan parameter ini untuk mengaktifkan SageMaker Training Compiler. Berikut ini adalah parameter untuk `TrainingCompilerConfig` kelas.
  - `enabled`(bool) — Opsional. Tentukan `True` atau `False` untuk mengaktifkan atau mematikan SageMaker Training Compiler. Nilai default-nya adalah `True`.
  - `debug`(bool) — Opsional. Untuk menerima log pelatihan yang lebih mendetail dari pekerjaan pelatihan yang dipercepat kompiler, ubah ke. `True` Namun, pencatatan tambahan mungkin menambah overhead dan memperlambat pekerjaan pelatihan yang dikompilasi. Nilai default-nya adalah `False`.

- `distribution`(dict) — Opsional. Untuk menjalankan pekerjaan pelatihan terdistribusi dengan SageMaker Training Compiler, tambahkan`distribution = { 'pytorchxla' : { 'enabled': True }}`.

> **Warning**
>
> Jika Anda mengaktifkan SageMaker Debugger, itu mungkin memengaruhi kinerja SageMaker Training Compiler. Kami menyarankan Anda mematikan Debugger saat menjalankan SageMaker Training Compiler untuk memastikan tidak ada dampak pada kinerja. Untuk informasi selengkapnya, lihat the section called “Pertimbangan”. Untuk menonaktifkan fungsionalitas Debugger, tambahkan dua argumen berikut ke estimator:
>
> ```
> disable_profiler=True,
> debugger_hook_config=False
> ```

Jika pekerjaan pelatihan dengan kompiler berhasil diluncurkan, Anda menerima log berikut selama fase inisialisasi pekerjaan:

- dengan `TrainingCompilerConfig(debug=False)`

```
Found configuration for Training Compiler
Configuring SM Training Compiler...
```

- dengan `TrainingCompilerConfig(debug=True)`

```
Found configuration for Training Compiler
Configuring SM Training Compiler...
Training Compiler set to debug mode
```

Menggunakan Operasi SageMaker **`CreateTrainingJob`** API

SageMaker Opsi konfigurasi Training Compiler harus ditentukan melalui `HyperParameters` kolom `AlgorithmSpecification` and dalam sintaks permintaan untuk operasi `CreateTrainingJob`API.

```
"AlgorithmSpecification": {
    "TrainingImage": "<sagemaker-training-compiler-enabled-dlc-image>"
```

```
},

"HyperParameters": {
    "sagemaker_training_compiler_enabled": "true",
    "sagemaker_training_compiler_debug_mode": "false",
    "sagemaker_pytorch_xla_multi_worker_enabled": "false"    // set to "true" for
 distributed training
}
```

Untuk menemukan daftar lengkap URI image container pembelajaran mendalam yang telah diimplementasikan SageMaker Training Compiler, lihat. Kerangka Kerja yang Didukung

Jalankan Pekerjaan TensorFlow Pelatihan dengan Kompiler SageMaker Pelatihan

Anda dapat menggunakan salah satu SageMaker antarmuka untuk menjalankan pekerjaan pelatihan dengan SageMaker Training Compiler: Amazon SageMaker Studio Classic, instans SageMaker notebook Amazon, AWS SDK for Python (Boto3) dan. AWS Command Line Interface

Topik

- Menggunakan SageMaker Python SDK
- Menggunakan SageMaker Python SDK dan Extending SageMaker Framework Deep Learning Containers
- Aktifkan Kompiler SageMaker Pelatihan Menggunakan Operasi SageMaker CreateTrainingJob API

Menggunakan SageMaker Python SDK

Untuk mengaktifkan SageMaker Training Compiler, tambahkan `compiler_config` parameter ke estimator SageMaker TensorFlow atau Hugging Face. Impor `TrainingCompilerConfig` kelas dan berikan instance ke `compiler_config` parameter. Contoh kode berikut menunjukkan struktur kelas SageMaker estimator dengan SageMaker Training Compiler diaktifkan.

> **Tip**
>
> Untuk memulai dengan model prebuilt yang disediakan oleh library TensorFlow and Transformers, coba gunakan ukuran batch yang disediakan dalam tabel referensi di. Model yang Diuji

> **Note**
>
> SageMaker Training Compiler untuk TensorFlow tersedia melalui estimator kerangka SageMaker TensorFlowkerja dan Hugging Face.

Untuk informasi yang sesuai dengan kasus penggunaan Anda, lihat salah satu opsi berikut.

Untuk pelatihan GPU tunggal

TensorFlow

```
from sagemaker.tensorflow import TensorFlow, TrainingCompilerConfig

# the original max batch size that can fit into GPU memory without compiler
batch_size_native=12
learning_rate_native=float('5e-5')

# an updated max batch size that can fit into GPU memory with compiler
batch_size=64

# update the global learning rate
learning_rate=learning_rate_native/batch_size_native*batch_size

hyperparameters={
    "n_gpus": 1,
    "batch_size": batch_size,
    "learning_rate": learning_rate
}

tensorflow_estimator=TensorFlow(
    entry_point='train.py',
    instance_count=1,
    instance_type='ml.p3.2xlarge',
    framework_version='2.9.1',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    disable_profiler=True,
    debugger_hook_config=False
)

tensorflow_estimator.fit()
```

Untuk menyiapkan skrip pelatihan Anda, lihat halaman-halaman berikut.

- Untuk pelatihan GPU tunggaldari model yang dibangun menggunakan TensorFlow Keras (`tf.keras.*`).
- Untuk pelatihan GPU tunggaldari model yang dibangun menggunakan TensorFlow modul (`tf.*`tidak termasuk modul TensorFlow Keras).

Hugging Face Estimator with TensorFlow

```
from sagemaker.huggingface import HuggingFace, TrainingCompilerConfig

# the original max batch size that can fit into GPU memory without compiler
batch_size_native=12
learning_rate_native=float('5e-5')

# an updated max batch size that can fit into GPU memory with compiler
batch_size=64

# update the global learning rate
learning_rate=learning_rate_native/batch_size_native*batch_size

hyperparameters={
    "n_gpus": 1,
    "batch_size": batch_size,
    "learning_rate": learning_rate
}

tensorflow_huggingface_estimator=HuggingFace(
    entry_point='train.py',
    instance_count=1,
    instance_type='ml.p3.2xlarge',
    transformers_version='4.21.1',
    tensorflow_version='2.6.3',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    disable_profiler=True,
    debugger_hook_config=False
)

tensorflow_huggingface_estimator.fit()
```

Untuk menyiapkan skrip pelatihan Anda, lihat halaman-halaman berikut.

- Untuk pelatihan GPU tunggalmodel TensorFlow Keras dengan Hugging Face Transformers
- Untuk pelatihan GPU tunggal TensorFlow model dengan Hugging Face Transformers

Untuk pelatihan terdistribusi

Hugging Face Estimator with TensorFlow

```
from sagemaker.huggingface import HuggingFace, TrainingCompilerConfig

# choose an instance type, specify the number of instances you want to use,
# and set the num_gpus variable the number of GPUs per instance.
instance_count=1
instance_type='ml.p3.8xlarge'
num_gpus=4

# the original max batch size that can fit to GPU memory without compiler
batch_size_native=16
learning_rate_native=float('5e-5')

# an updated max batch size that can fit to GPU memory with compiler
batch_size=26

# update learning rate
learning_rate=learning_rate_native/
batch_size_native*batch_size*num_gpus*instance_count

hyperparameters={
    "n_gpus": num_gpus,
    "batch_size": batch_size,
    "learning_rate": learning_rate
}

tensorflow_huggingface_estimator=HuggingFace(
    entry_point='train.py',
    instance_count=instance_count,
    instance_type=instance_type,
    transformers_version='4.21.1',
    tensorflow_version='2.6.3',
    hyperparameters=hyperparameters,
    compiler_config=TrainingCompilerConfig(),
    disable_profiler=True,
    debugger_hook_config=False
```

```
)

tensorflow_huggingface_estimator.fit()
```

> **Tip**
>
> Untuk menyiapkan skrip pelatihan Anda, lihat halaman-halaman berikut.
>
> - Untuk pelatihan terdistribusimodel TensorFlow Keras dengan Hugging Face Transformers
> - Untuk pelatihan terdistribusi TensorFlow model dengan Hugging Face Transformers

Daftar berikut adalah set minimal parameter yang diperlukan untuk menjalankan pekerjaan SageMaker pelatihan dengan compiler.

> **Note**
>
> Saat menggunakan estimator SageMaker Hugging Face, Anda harus menentukan`transformers_version`,, `tensorflow_version`hyperparameters, `compiler_config` dan parameter untuk SageMaker mengaktifkan Training Compiler. Anda tidak dapat menggunakan `image_uri` untuk secara manual menentukan Deep Learning Containers terintegrasi Training Compiler yang terdaftar diKerangka Kerja yang Didukung.

- `entry_point`(str) - Diperlukan. Tentukan nama file skrip pelatihan Anda.
- `instance_count`(int) - Diperlukan. Tentukan jumlah instans.
- `instance_type`(str) - Diperlukan. Tentukan jenis instance.
- `transformers_version`(str) - Diperlukan hanya saat menggunakan estimator SageMaker Hugging Face. Tentukan versi pustaka Hugging Face Transformers yang didukung oleh SageMaker Training Compiler. Untuk menemukan versi yang tersedia, lihatKerangka Kerja yang Didukung.
- `framework_version`atau `tensorflow_version` (str) - Diperlukan. Tentukan TensorFlow versi yang didukung oleh SageMaker Training Compiler. Untuk menemukan versi yang tersedia, lihatKerangka Kerja yang Didukung.

> **Note**
>
> Saat menggunakan SageMaker TensorFlow estimator, Anda harus menentukan`framework_version`.
> Saat menggunakan estimator SageMaker Hugging Face, Anda harus menentukan `transformers_version` keduanya dan. `tensorflow_version`

- `hyperparameters`(dict) — Opsional. Tentukan hyperparameters untuk pekerjaan pelatihan, sepertin_gpus,`batch_size`, dan`learning_rate`. Saat Anda mengaktifkan SageMaker Training Compiler, coba ukuran batch yang lebih besar dan sesuaikan tingkat pembelajaran yang sesuai. Untuk menemukan studi kasus penggunaan kompiler dan ukuran batch yang disesuaikan untuk meningkatkan kecepatan pelatihan, lihat the section called “Model yang Diuji” danSageMaker Contoh Kompiler Pelatihan Notebook dan Blog.
- `compiler_config`(TrainingCompilerConfig objek) - Diperlukan. Sertakan parameter ini untuk mengaktifkan SageMaker Training Compiler. Berikut ini adalah parameter untuk `TrainingCompilerConfig` kelas.
  - `enabled`(bool) — Opsional. Tentukan `True` atau `False` untuk mengaktifkan atau mematikan SageMaker Training Compiler. Nilai default-nya adalah `True`.
  - `debug`(bool) — Opsional. Untuk menerima log pelatihan yang lebih mendetail dari pekerjaan pelatihan yang dipercepat kompiler, ubah ke. `True` Namun, pencatatan tambahan mungkin menambah overhead dan memperlambat pekerjaan pelatihan yang dikompilasi. Nilai default-nya adalah `False`.

> **Warning**
>
> Jika Anda mengaktifkan SageMaker Debugger, itu mungkin memengaruhi kinerja SageMaker Training Compiler. Kami menyarankan Anda mematikan Debugger saat menjalankan SageMaker Training Compiler untuk memastikan tidak ada dampak pada kinerja. Untuk informasi selengkapnya, lihat the section called “Pertimbangan”. Untuk menonaktifkan fungsionalitas Debugger, tambahkan dua argumen berikut ke estimator:
>
> ```
> disable_profiler=True,
> debugger_hook_config=False
> ```

Jika pekerjaan pelatihan dengan kompiler berhasil diluncurkan, Anda menerima log berikut selama fase inisialisasi pekerjaan:

- dengan `TrainingCompilerConfig(debug=False)`

```
Found configuration for Training Compiler
Configuring SM Training Compiler...
```

- dengan `TrainingCompilerConfig(debug=True)`

```
Found configuration for Training Compiler
Configuring SM Training Compiler...
Training Compiler set to debug mode
```

Menggunakan SageMaker Python SDK dan Extending SageMaker Framework Deep Learning Containers

AWSDeep Learning Containers (DLC) untuk TensorFlow menggunakan versi adaptasi TensorFlow yang menyertakan perubahan di atas kerangka open source TensorFlow . SageMaker Framework Deep Learning Containers dioptimalkan untuk AWS infrastruktur dasar dan Amazon SageMaker. Dengan keuntungan menggunakan DLC, integrasi SageMaker Training Compiler menambahkan lebih banyak peningkatan kinerja dibandingkan yang asli. TensorFlow Selanjutnya, Anda dapat membuat wadah pelatihan khusus dengan memperluas gambar DLC.

> **Note**
>
> Fitur kustomisasi Docker ini saat ini hanya tersedia untuk TensorFlow.

Untuk memperluas dan menyesuaikan SageMaker TensorFlow DLC untuk kasus penggunaan Anda, gunakan petunjuk berikut.

Buat Dockerfile

Gunakan template Dockerfile berikut untuk memperluas DLC. SageMaker TensorFlow Anda harus menggunakan gambar SageMaker TensorFlow DLC sebagai gambar dasar wadah Docker Anda. Untuk menemukan URI image SageMaker TensorFlow DLC, lihat Kerangka Kerja yang Didukung.

```
# SageMaker TensorFlow Deep Learning Container image
```

```
FROM 763104351884.dkr.ecr.<aws-region>.amazonaws.com/tensorflow-training:<image-tag>

ENV PATH="/opt/ml/code:${PATH}"

# This environment variable is used by the SageMaker container
# to determine user code directory.
ENV SAGEMAKER_SUBMIT_DIRECTORY /opt/ml/code

# Add more code lines to customize for your use-case
...
```

Untuk informasi selengkapnya, lihat Langkah 2: Membuat dan mengunggah skrip pelatihan Dockerfile dan Python.

Pertimbangkan jebakan berikut saat memperluas SageMaker Framework DLC:

- Jangan secara eksplisit menghapus atau mengubah versi TensorFlow paket dalam wadah. SageMaker Melakukan hal itu menyebabkan TensorFlow paket yang AWS dioptimalkan ditimpa oleh TensorFlow paket sumber terbuka, yang dapat mengakibatkan penurunan kinerja.
- Hati-hati dengan paket yang memiliki TensorFlow versi atau rasa tertentu sebagai ketergantungan. Paket-paket ini mungkin secara implisit menghapus instalasi paket sumber terbuka yang AWS dioptimalkan TensorFlow dan menginstal. TensorFlow

Misalnya, ada masalah yang diketahui bahwa pustaka tensorflow/models dan tensorflow/text selalu mencoba menginstal ulang open source. TensorFlow Jika Anda perlu menginstal pustaka ini untuk memilih versi tertentu untuk kasus penggunaan Anda, kami sarankan Anda melihat ke SageMaker TensorFlow DLC Dockerfiles untuk v2.9 atau yang lebih baru. Jalur ke Dockerfiles biasanya dalam format berikut:. `tensorflow/training/docker/<tensorflow-version>/py3/<cuda-version>/Dockerfile.gpu` Di Dockerfiles, Anda harus menemukan baris kode untuk menginstal ulang TensorFlow biner AWS terkelola (ditentukan ke variabel `TF_URL` lingkungan) dan dependensi lainnya secara berurutan. Bagian instalasi ulang akan terlihat seperti contoh berikut:

```
# tf-models does not respect existing installations of TensorFlow
# and always installs open source TensorFlow

RUN pip3 install --no-cache-dir -U \
    tf-models-official==x.y.z

RUN pip3 uninstall -y tensorflow tensorflow-gpu \
  ; pip3 install --no-cache-dir -U \
```

```
    ${TF_URL} \
    tensorflow-io==x.y.z \
    tensorflow-datasets==x.y.z
```

Bangun dan dorong ke ECR

Untuk membangun dan mendorong container Docker Anda ke Amazon ECR, ikuti petunjuk di tautan berikut:

- Langkah 3: Bangun wadah
- Langkah 4: Uji wadahnya
- Langkah 5: Dorong wadah ke Amazon ECR

Jalankan menggunakan SageMaker Python SDK Estimator

Gunakan estimator SageMaker TensorFlow kerangka kerja seperti biasa. Anda harus menentukan `image_uri` untuk menggunakan wadah baru yang Anda host di Amazon ECR.

```
import sagemaker, boto3
from sagemaker import get_execution_role
from sagemaker.tensorflow import TensorFlow, TrainingCompilerConfig

account_id = boto3.client('sts').get_caller_identity().get('Account')
ecr_repository = 'tf-custom-container-test'
tag = ':latest'

region = boto3.session.Session().region_name

uri_suffix = 'amazonaws.com'

byoc_image_uri = '{}.dkr.ecr.{}.{}/{}'.format(
    account_id, region, uri_suffix, ecr_repository + tag
)

byoc_image_uri
# This should return something like
# 111122223333.dkr.ecr.us-east-2.amazonaws.com/tf-custom-container-test:latest

estimator = TensorFlow(
    image_uri=image_uri,
    role=get_execution_role(),
```

```
    base_job_name='tf-custom-container-test-job',
    instance_count=1,
    instance_type='ml.p3.8xlarge'
    compiler_config=TrainingCompilerConfig(),
    disable_profiler=True,
    debugger_hook_config=False
)

# Start training
estimator.fit()
```

Aktifkan Kompiler SageMaker Pelatihan Menggunakan Operasi SageMaker **`CreateTrainingJob`** API

SageMaker Opsi konfigurasi Training Compiler harus ditentukan melalui `HyperParameters` kolom `AlgorithmSpecification` and dalam sintaks permintaan untuk operasi `CreateTrainingJob`API.

```
"AlgorithmSpecification": {
    "TrainingImage": "<sagemaker-training-compiler-enabled-dlc-image>"
},

"HyperParameters": {
    "sagemaker_training_compiler_enabled": "true",
    "sagemaker_training_compiler_debug_mode": "false"
}
```

Untuk menemukan daftar lengkap URI image container pembelajaran mendalam yang telah diimplementasikan SageMaker Training Compiler, lihat. Kerangka Kerja yang Didukung

SageMaker Contoh Kompiler Pelatihan Notebook dan Blog

Blog, studi kasus, dan notebook berikut memberikan contoh bagaimana menerapkan SageMaker Training Compiler.

Contoh notebook disediakan di GitHub repositori SageMaker contoh, dan Anda juga dapat menjelajahinya di situs web contoh. SageMaker

Blog dan Studi Kasus

Blog-blog berikut membahas studi kasus tentang penggunaan SageMaker Training Compiler.

- Baru - Memperkenalkan Kompiler SageMaker Pelatihan
- Hugging Face Transformers BERT fine-tuning menggunakan Amazon Training Compiler SageMaker
- Percepat Pekerjaan AWS Pelatihan Hugging Face hingga 50% SageMaker dengan Training Compiler

Contoh Notebook

Untuk menemukan contoh penggunaan SageMaker Training Compiler, lihat halaman Training Compiler di Amazon SageMaker Example Read the Docs website.

SageMaker Praktik dan Pertimbangan Terbaik Kompiler Pelatihan

Tinjau praktik dan pertimbangan terbaik berikut saat menggunakan SageMaker Training Compiler.

Praktik Terbaik

Gunakan panduan berikut untuk mencapai hasil terbaik saat Anda menjalankan pekerjaan pelatihan dengan SageMaker Training Compiler.

Praktik Terbaik Umum

- Pastikan Anda menggunakan salah satu Tipe Instans Yang Didukung danModel yang Diuji.
- Saat Anda membuat tokenizer untuk model NLP menggunakan pustaka Hugging Face Transformers dalam skrip pelatihan Anda, pastikan Anda menggunakan bentuk tensor input statis dengan menentukan. `padding='max_length'` Jangan gunakan `padding='longest'` karena padding ke urutan terpanjang dalam batch dapat mengubah bentuk tensor untuk setiap batch pelatihan. Bentuk input dinamis dapat memulai kompilasi ulang model dan dapat meningkatkan total waktu pelatihan. Untuk informasi selengkapnya tentang opsi padding tokenizer Transformers, lihat Padding dan pemotongan dalam dokumentasi Hugging Face Transformers.
- Ukur pemanfaatan memori GPU untuk memastikan bahwa Anda menggunakan ukuran batch maksimum yang dapat masuk ke dalam memori GPU. Amazon SageMaker Training Compiler mengurangi jejak memori model Anda selama pelatihan, yang biasanya memungkinkan Anda memasukkan memori GPU yang lebih besar`batch_size`. Menggunakan `batch_size` hasil yang lebih besar dalam pemanfaatan GPU yang lebih baik dan mengurangi total waktu pelatihan.

  Saat Anda menyesuaikan ukuran batch, Anda juga harus menyesuaikannya `learning_rate` dengan tepat. Misalnya, jika Anda meningkatkan ukuran batch dengan faktork, Anda perlu

menyesuaikan secara `learning_rate` linier (perkalian sederhana dengank) atau kalikan dengan akar kuadrat dari. k Ini untuk mencapai perilaku konvergensi yang sama atau serupa dalam waktu pelatihan yang berkurang. Untuk referensi `batch_size` diuji untuk model populer, lihatModel yang Diuji.

- Untuk men-debug tugas pelatihan yang dipercepat kompiler, aktifkan `debug` flag di parameter. `compiler_config` Hal ini memungkinkan SageMaker untuk menempatkan log debugging ke dalam log pekerjaan SageMaker pelatihan.

```
huggingface_estimator=HuggingFace(
    ...
    compiler_config=TrainingCompilerConfig(debug=True)
)
```

Perhatikan bahwa jika Anda mengaktifkan debugging penuh dari pekerjaan pelatihan dengan kompiler, ini mungkin menambahkan beberapa overhead.

Praktik Terbaik untuk PyTorch

- Jika Anda membawa PyTorch model dan ingin memeriksanya, pastikan Anda menggunakan fungsi penyimpanan PyTorch model/XLA untuk memeriksa model Anda dengan benar. Untuk informasi selengkapnya tentang fungsi ini, lihat `torch_xla.core.xla_model.savePyTorch` di dokumentasi Perangkat XLA.

  Untuk mempelajari cara menambahkan modifikasi ke PyTorch skrip Anda, lihatModel Bahasa Besar Menggunakan PyTorch Langsung (tanpa API Pelatih Transformers Wajah Memeluk).

  Untuk informasi lebih lanjut tentang aplikasi aktual menggunakan fungsi penyimpanan model, lihat Checkpoint Writing and Loading in the Hugging Face on PyTorch /XLA TPUs: Blog pelatihan yang lebih cepat dan lebih murah.
- Untuk mencapai waktu pelatihan paling optimal untuk pelatihan terdistribusi, pertimbangkan hal berikut.
  - Gunakan instance dengan beberapa GPU alih-alih menggunakan instance gpu tunggal. Misalnya, satu `ml.p3dn.24xlarge` instance memiliki waktu pelatihan yang lebih cepat dibandingkan dengan 8 x `ml.p3.2xlarge` instance.
  - Gunakan instance dengan dukungan EFA seperti `ml.p3dn.24xlarge` dan. `ml.p4d.24xlarge` Jenis contoh ini telah mempercepat kecepatan jaringan dan mengurangi waktu pelatihan.

- Setel `preprocessing_num_workers` parameter untuk kumpulan data, sehingga pelatihan model tidak tertunda oleh preprocessing yang lambat.

Pertimbangan

Pertimbangkan hal berikut saat menggunakan SageMaker Training Compiler.

Degradasi kinerja karena logging, checkpointing, dan profiling

- Hindari logging, checkpointing, dan profiling tensor model yang mengarah ke evaluasi eksplisit. Untuk memahami apa itu evaluasi eksplisit, pertimbangkan contoh kompilasi kode berikut.

```
a = b+c
e = a+d
```

Sebuah compiler menafsirkan kode sebagai berikut dan mengurangi footprint memori untuk variabel: a

```
e = b+c+d
```

Sekarang pertimbangkan kasus berikut di mana kode diubah untuk menambahkan fungsi cetak untuk variabela.

```
a = b+c
e = a+d
print(a)
```

Compiler membuat evaluasi eksplisit dari variabel a sebagai berikut.

```
e = b+c+d
a = b+c    # Explicit evaluation
print(a)
```

Misalnya PyTorch, hindari penggunaan torch.tensor.items (), yang mungkin memperkenalkan evaluasi eksplisit. Dalam pembelajaran mendalam, evaluasi eksplisit seperti itu dapat menyebabkan overhead karena mereka merusak operasi yang menyatu dalam grafik kompilasi model dan mengarah pada perhitungan ulang tensor.

Jika Anda masih ingin mengevaluasi model secara berkala selama pelatihan saat menggunakan SageMaker Training Compiler, kami merekomendasikan logging dan checkpointing pada frekuensi yang lebih rendah untuk mengurangi overhead karena evaluasi eksplisit. Misalnya, catat setiap 10 zaman, bukan setiap zaman.

- Kompilasi grafik berjalan selama beberapa langkah pertama pelatihan. Akibatnya, beberapa langkah pertama diharapkan sangat lambat. Namun, ini adalah biaya kompilasi satu kali dan dapat diamortisasi dengan pelatihan untuk durasi yang lebih lama karena kompilasi membuat langkah-langkah masa depan jauh lebih cepat. Overhead kompilasi awal tergantung pada ukuran model, ukuran tensor input, dan distribusi bentuk tensor input.

Penggunaan PyTorch /XLA API yang salah saat menggunakan secara langsung PyTorch

PyTorch/XLA mendefinisikan satu set API untuk menggantikan beberapa API pelatihan yang ada PyTorch. Gagal menggunakannya dengan benar menyebabkan PyTorch pelatihan gagal.

- Salah satu kesalahan paling umum saat mengkompilasi PyTorch model adalah karena jenis perangkat yang salah untuk operator dan tensor. Untuk mengkompilasi PyTorch model dengan benar, pastikan Anda menggunakan perangkat XLA (`xm.xla_device()`) alih-alih menggunakan CUDA atau mencampur perangkat CUDA dan perangkat XLA.
- `mark_step()`adalah penghalang hanya untuk XLA. Gagal mengaturnya dengan benar menyebabkan pekerjaan pelatihan terhenti.
- PyTorch/XLA menyediakan API pelatihan terdistribusi tambahan. Gagal memprogram API dengan benar menyebabkan gradien dikumpulkan secara tidak benar, yang menyebabkan kegagalan konvergensi pelatihan.

Untuk mengatur PyTorch skrip Anda dengan benar dan menghindari penggunaan API yang salah yang disebutkan di atas, lihatModel Bahasa Besar Menggunakan PyTorch Langsung (tanpa API Pelatih Transformers Wajah Memeluk).

SageMaker FAQ Kompiler Pelatihan

Gunakan item FAQ berikut untuk menemukan jawaban atas pertanyaan umum tentang SageMaker Training Compiler.

T. Bagaimana saya tahu SageMaker Training Compiler bekerja?

Jika Anda berhasil meluncurkan tugas pelatihan dengan SageMaker Training Compiler, Anda menerima pesan log berikut:

- dengan `TrainingCompilerConfig(debug=False)`

```
Found configuration for Training Compiler
Configuring SM Training Compiler...
```

- dengan `TrainingCompilerConfig(debug=True)`

```
Found configuration for Training Compiler
Configuring SM Training Compiler...
Training Compiler set to debug mode
```

T. Model mana yang dipercepat oleh SageMaker Training Compiler?

SageMaker Training Compiler mendukung model pembelajaran mendalam paling populer dari perpustakaan transformer Hugging Face. Dengan sebagian besar operator yang didukung kompiler, model ini dapat dilatih lebih cepat dengan SageMaker Training Compiler. Model yang dapat dikompilasi termasuk tetapi tidak terbatas pada hal-hal berikut:`bert-base-cased`,,,,,,,,`bert-base-chinese`,,`bert-base-uncased`,,`distilbert-base-uncased`,`distilbert-base-uncased-finetuned-sst-2-english`,,`gpt2`,,`roberta-base`,`roberta-large`,,`t5-base`,,, `xlm-roberta-base` Kompiler bekerja dengan sebagian besar operator DL dan struktur data dan dapat mempercepat banyak model DL lainnya di luar yang telah diuji.

T. Apa yang terjadi jika saya mengaktifkan SageMaker Training Compiler dengan model yang tidak diuji?

Untuk model yang belum teruji, Anda mungkin perlu memodifikasi skrip pelatihan terlebih dahulu agar kompatibel dengan SageMaker Training Compiler. Untuk informasi lebih lanjut, lihat Bawa Model Pembelajaran Mendalam Anda Sendiri dan ikuti petunjuk tentang cara menyiapkan skrip pelatihan Anda.

Setelah Anda memperbarui skrip pelatihan Anda, Anda dapat memulai pekerjaan pelatihan. Kompiler melanjutkan untuk mengkompilasi model. Namun, kecepatan pelatihan mungkin tidak meningkat dan bahkan mungkin menurun relatif terhadap baseline dengan model yang belum teruji. Anda mungkin perlu menyesuaikan kembali parameter pelatihan seperti `batch_size` dan `learning_rate` untuk mencapai manfaat percepatan apa pun.

Jika kompilasi model yang belum teruji gagal, kompiler mengembalikan kesalahan. Lihat SageMaker Pemecahan Masalah Kompiler Pelatihan untuk informasi rinci tentang jenis kegagalan dan pesan kesalahan.

T. Apakah saya akan selalu mendapatkan pekerjaan pelatihan yang lebih cepat dengan SageMaker Training Compiler?

Tidak, belum tentu. Pertama, SageMaker Training Compiler menambahkan beberapa overhead kompilasi sebelum proses pelatihan yang sedang berlangsung dapat dipercepat. Pekerjaan pelatihan yang dioptimalkan harus berjalan cukup lama untuk diamortisasi dan menebus overhead kompilasi tambahan ini di awal pekerjaan pelatihan.

Selain itu, seperti halnya proses pelatihan model lainnya, pelatihan dengan parameter suboptimal dapat meningkatkan waktu pelatihan. SageMaker Training Compiler dapat mengubah karakteristik pekerjaan pelatihan dengan, misalnya, mengubah jejak memori pekerjaan. Karena perbedaan ini, Anda mungkin perlu menyesuaikan kembali parameter pekerjaan pelatihan Anda untuk mempercepat pelatihan. Tabel referensi yang menentukan parameter berkinerja terbaik untuk pekerjaan pelatihan dengan jenis dan model instans yang berbeda dapat ditemukan diModel yang Diuji.

Akhirnya, beberapa kode dalam skrip pelatihan mungkin menambahkan overhead tambahan atau mengganggu grafik komputasi yang dikompilasi dan pelatihan yang lambat. Jika bekerja dengan model yang disesuaikan atau belum teruji, lihat instruksi diPraktik Terbaik untuk Menggunakan Kompiler SageMaker Pelatihan PyTorch dengan/XLA.

T. Dapatkah saya selalu menggunakan ukuran batch yang lebih besar dengan SageMaker Training Compiler?

Ukuran Batch meningkat di sebagian besar, tetapi tidak semua, kasus. Pengoptimalan yang dilakukan oleh SageMaker Training Compiler dapat mengubah karakteristik pekerjaan pelatihan Anda, seperti jejak memori. Biasanya, pekerjaan Training Compiler menempati lebih sedikit memori daripada pekerjaan pelatihan yang tidak dikompilasi dengan kerangka kerja asli, yang memungkinkan ukuran batch yang lebih besar selama pelatihan. Ukuran batch yang lebih besar, dan penyesuaian yang sesuai dengan tingkat pembelajaran, meningkatkan throughput pelatihan dan dapat mengurangi total waktu pelatihan.

Namun, mungkin ada kasus di mana SageMaker Training Compiler mungkin benar-benar meningkatkan jejak memori berdasarkan skema pengoptimalannya. Compiler menggunakan model biaya analitis untuk memprediksi jadwal eksekusi dengan biaya eksekusi terendah untuk setiap operator komputasi intensif. Model ini dapat menemukan jadwal optimal yang meningkatkan

penggunaan memori. Dalam hal ini, Anda tidak akan dapat meningkatkan ukuran batch, tetapi throughput sampel Anda masih lebih tinggi.

T. Apakah SageMaker Training Compiler bekerja dengan fitur SageMaker pelatihan lainnya, seperti perpustakaan pelatihan SageMaker terdistribusi dan Debugger? SageMaker

SageMaker Training Compiler saat ini tidak kompatibel dengan SageMaker pustaka pelatihan terdistribusi.

SageMaker Training Compiler kompatibel dengan SageMaker Debugger, tetapi Debugger mungkin menurunkan kinerja komputasi dengan menambahkan overhead.

T. Apakah SageMaker Training Compiler mendukung kontainer kustom (membawa kontainer Anda sendiri)?

SageMaker Training Compiler disediakan melalui AWS Deep Learning Containers, dan Anda dapat memperluas subset container untuk disesuaikan dengan kasus penggunaan Anda. Wadah yang diperluas dari AWS DLC didukung oleh SageMaker Training Compiler. Untuk informasi selengkapnya, lihat Kerangka Kerja yang Didukung danMenggunakan SageMaker Python SDK dan Extending SageMaker Framework Deep Learning Containers. Jika Anda membutuhkan dukungan lebih lanjut, hubungi SageMaker tim melalui Forum AWSDukungan atau AWS Pengembang untuk Amazon SageMaker.

SageMaker Pemecahan Masalah Kompiler Pelatihan

Jika Anda mengalami kesalahan, Anda dapat menggunakan daftar berikut untuk mencoba memecahkan masalah pekerjaan pelatihan Anda. Jika Anda membutuhkan dukungan lebih lanjut, hubungi SageMaker tim melalui Forum AWSDukungan atau AWS Pengembang untuk Amazon SageMaker.

Pekerjaan pelatihan tidak konvergen seperti yang diharapkan jika dibandingkan dengan pekerjaan pelatihan kerangka kerja asli

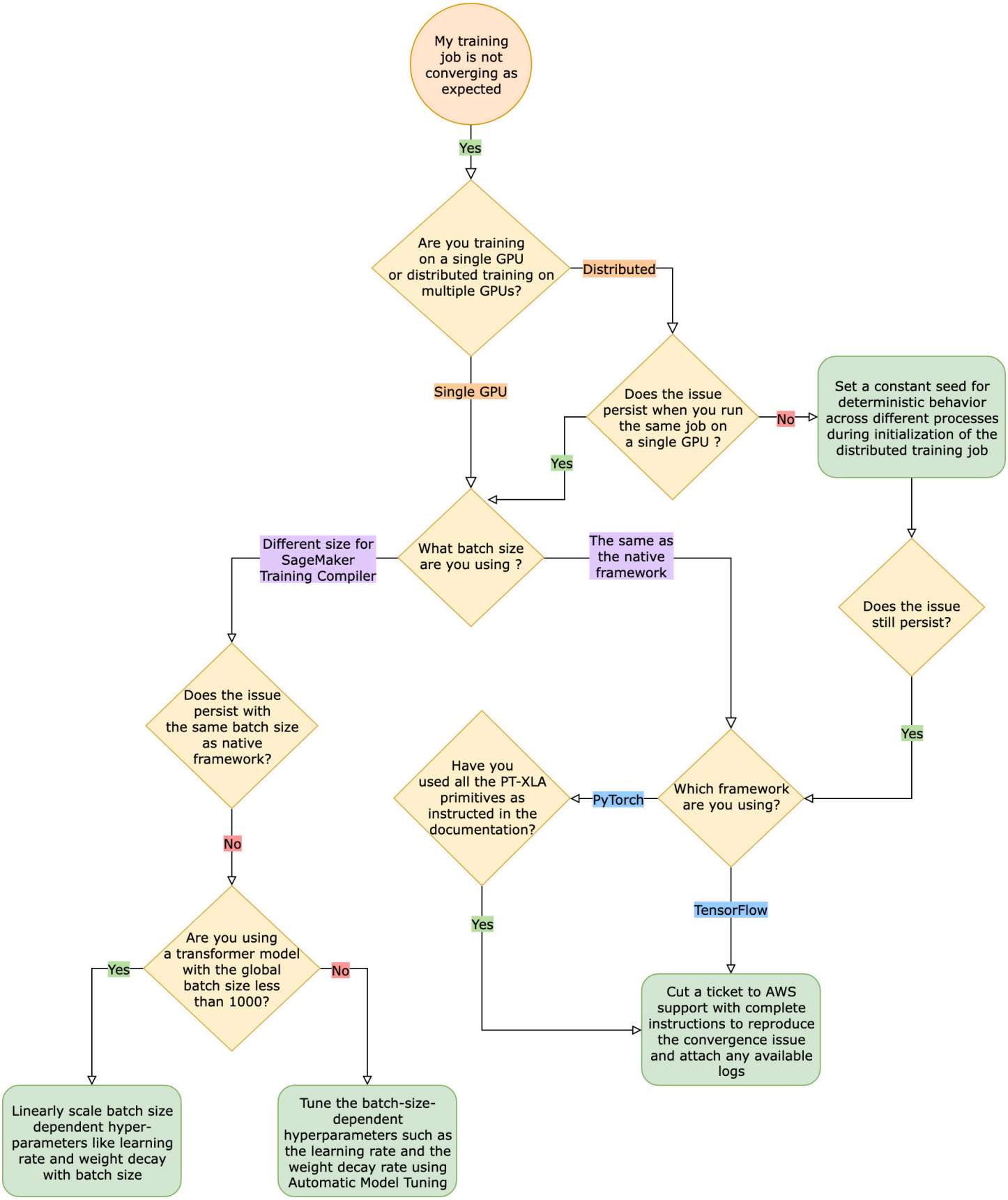
Masalah konvergensi berkisar dari “model tidak belajar saat SageMaker Training Compiler dihidupkan” hingga “model sedang belajar tetapi lebih lambat dari kerangka kerja asli”. Dalam panduan pemecahan masalah ini, kami menganggap konvergensi Anda baik-baik saja tanpa SageMaker Training Compiler (dalam kerangka kerja asli) dan menganggap ini sebagai baseline.

Ketika dihadapkan dengan masalah konvergensi seperti itu, langkah pertama adalah mengidentifikasi apakah masalah ini terbatas pada pelatihan terdistribusi atau berasal dari pelatihan GPU tunggal.

Pelatihan terdistribusi dengan SageMaker Training Compiler adalah perpanjangan dari pelatihan GPU tunggal dengan langkah-langkah tambahan.

1. Siapkan cluster dengan beberapa instance atau GPU.
2. Mendistribusikan data input ke semua pekerja.
3. Sinkronkan pembaruan model dari semua pekerja.

Oleh karena itu, setiap masalah konvergensi dalam pelatihan GPU tunggal menyebar ke pelatihan terdistribusi dengan banyak pekerja.


My training job is not converging as expected
Yes
Are you training on a single GPU or distributed training on multiple GPUs?
Distributed
Single GPU
Does the issue persist when you run the same job on a single GPU ?
No
Yes
Set a constant seed for deterministic behavior across different processes during initialization of the distributed training job
What batch size are you using ?
Different size for SageMaker Training Compiler
The same as the native framework
Does the issue still persist?
Does the issue persist with the same batch size as native framework?
Yes
Which framework are you using?
PyTorch
TensorFlow
Have you used all the PT-XLA primitives as instructed in the documentation?
No
Yes
Are you using a transformer model with the global batch size less than 1000?
Yes
No
Cut a ticket to AWS support with complete instructions to reproduce the convergence issue and attach any available logs
Linearly scale batch size dependent hyper-parameters like learning rate and weight decay with batch size
Tune the batch-size-dependent hyperparameters such as the learning rate and the weight decay rate using Automatic Model Tuning

A flow chart to troubleshoot convergence issues in training jobs when using SageMaker Training Compiler. Descriptions are in the following sections.

Masalah konvergensi yang terjadi dalam pelatihan GPU tunggal

Jika masalah konvergensi Anda berasal dari pelatihan GPU tunggal, ini kemungkinan disebabkan oleh pengaturan yang tidak tepat untuk hiperparameter atau API. `torch_xla`

Periksa hiperparameter

SageMaker Pelatihan dengan Training Compiler menyebabkan perubahan jejak memori model. Kompiler secara cerdas menengahi antara penggunaan kembali dan komputasi ulang yang mengarah ke peningkatan atau penurunan konsumsi memori yang sesuai. Untuk memanfaatkan ini, penting untuk menyetel ulang ukuran batch dan hyperparameter terkait saat memigrasikan pekerjaan pelatihan ke SageMaker Training Compiler. Namun, pengaturan hiperparameter yang salah sering menyebabkan osilasi dalam kehilangan pelatihan dan mungkin konvergensi yang lebih lambat sebagai hasilnya. Dalam kasus yang jarang terjadi, hiperparameter agresif dapat mengakibatkan model tidak belajar (metrik kehilangan pelatihan tidak berkurang atau kembali`NaN`). Untuk mengidentifikasi apakah masalah konvergensi disebabkan oleh hiperparameter, lakukan side-by-side pengujian dua pekerjaan pelatihan dengan dan tanpa SageMaker Training Compiler sambil menjaga semua hyperparameter tetap sama.

Periksa apakah **`torch_xla`** API telah diatur dengan benar untuk pelatihan GPU tunggal

Jika masalah konvergensi berlanjut dengan hiperparameter dasar, Anda perlu memeriksa apakah ada penggunaan `torch_xla` API yang tidak tepat, khususnya yang untuk memperbarui model. Pada dasarnya, `torch_xla` terus mengakumulasi instruksi (menunda eksekusi) dalam bentuk grafik hingga secara eksplisit diinstruksikan untuk menjalankan grafik akumulasi. `torch_xla.core.xla_model.mark_step()`Fungsi ini memfasilitasi pelaksanaan grafik yang terakumulasi. Eksekusi grafik harus disinkronkan menggunakan fungsi ini setelah setiap pembaruan model dan sebelum mencetak dan mencatat variabel apa pun. Jika tidak memiliki langkah sinkronisasi, model mungkin menggunakan nilai basi dari memori selama pencetakan, log, dan penerusan berikutnya, alih-alih menggunakan nilai terbaru yang harus disinkronkan setelah setiap iterasi dan pembaruan model.

Ini bisa menjadi lebih rumit saat menggunakan SageMaker Training Compiler dengan penskalaan gradien (mungkin dari penggunaan AMP) atau teknik kliping gradien. Urutan komputasi gradien yang sesuai dengan AMP adalah sebagai berikut.

1. Komputasi gradien dengan penskalaan

2. Gradien un-scaling, kliping gradien, dan kemudian penskalaan
3. Pembaruan model
4. Menyinkronkan eksekusi grafik dengan `mark_step()`

Untuk menemukan API yang tepat untuk operasi yang disebutkan dalam daftar, lihat panduan untuk memigrasikan skrip pelatihan Anda ke Kompiler SageMaker Pelatihan.

Pertimbangkan untuk menggunakan Penyetelan Model Otomatis

Jika masalah konvergensi muncul saat menyetel ulang ukuran batch dan hiperparameter terkait seperti kecepatan pembelajaran saat menggunakan Kompiler SageMaker Pelatihan, pertimbangkan untuk menggunakan Penyetelan Model Otomatis untuk menyetel hiperparameter Anda. Anda dapat merujuk ke contoh notebook tentang tuning hyperparameters dengan SageMaker Training Compiler.

Masalah konvergensi yang terjadi dalam pelatihan terdistribusi

Jika masalah konvergensi Anda berlanjut dalam pelatihan terdistribusi, hal ini kemungkinan disebabkan oleh pengaturan yang tidak tepat untuk inisialisasi bobot atau API. `torch_xla`

Periksa inisialisasi berat di seluruh pekerja

Jika masalah konvergensi muncul saat menjalankan pekerjaan pelatihan terdistribusi dengan banyak pekerja, pastikan ada perilaku deterministik yang seragam di semua pekerja dengan menetapkan benih konstan jika berlaku. Waspadalah terhadap teknik seperti inisialisasi berat badan, yang melibatkan pengacakan. Setiap pekerja mungkin akhirnya melatih model yang berbeda tanpa adanya benih yang konstan.

Periksa apakah **`torch_xla`** API telah disiapkan dengan benar untuk pelatihan terdistribusi

Jika masalah masih berlanjut, ini kemungkinan karena penggunaan `torch_xla` API yang tidak tepat untuk pelatihan terdistribusi. Pastikan Anda menambahkan berikut ini di estimator Anda untuk menyiapkan klaster untuk pelatihan terdistribusi dengan SageMaker Training Compiler.

```
distribution={'torchxla': {'enabled': True}}
```

Ini harus disertai dengan fungsi `_mp_fn(index)` dalam skrip pelatihan Anda, yang dipanggil sekali per pekerja. Tanpa `mp_fn(index)` fungsi, Anda mungkin akhirnya membiarkan setiap pekerja melatih model secara mandiri tanpa berbagi pembaruan model.

Selanjutnya, pastikan Anda menggunakan `torch_xla.distributed.parallel_loader.MpDeviceLoader` API bersama dengan sampler data terdistribusi, seperti yang dipandu dalam dokumentasi tentang memigrasikan skrip pelatihan Anda ke SageMaker Training Compiler, seperti pada contoh berikut.

```
torch.utils.data.distributed.DistributedSampler()
```

Ini memastikan bahwa data input didistribusikan dengan benar di semua pekerja.

Terakhir, untuk menyinkronkan pembaruan model dari semua pekerja, gunakan `torch_xla.core.xla_model._fetch_gradients` untuk mengumpulkan gradien dari semua pekerja dan `torch_xla.core.xla_model.all_reduce` untuk menggabungkan semua gradien yang dikumpulkan menjadi satu pembaruan.

Ini bisa menjadi lebih rumit saat menggunakan SageMaker Training Compiler dengan penskalaan gradien (mungkin dari penggunaan AMP) atau teknik kliping gradien. Urutan komputasi gradien yang sesuai dengan AMP adalah sebagai berikut.

1. Komputasi gradien dengan penskalaan
2. Sinkronisasi gradien di semua pekerja
3. Gradien un-scaling, kliping gradien, dan kemudian penskalaan gradien
4. Pembaruan model
5. Menyinkronkan eksekusi grafik dengan `mark_step()`

Perhatikan bahwa daftar periksa ini memiliki item tambahan untuk menyinkronkan semua pekerja, dibandingkan dengan daftar periksa untuk pelatihan GPU tunggal.

Pekerjaan pelatihan gagal karena konfigurasi PyTorcl /XLA yang hilang

Jika tugas pelatihan gagal dengan pesan `Missing XLA configuration` kesalahan, itu mungkin karena kesalahan konfigurasi dalam jumlah GPU per instance yang Anda gunakan.

XLA membutuhkan variabel lingkungan tambahan untuk mengkompilasi pekerjaan pelatihan. Variabel lingkungan hilang yang paling umum adalah`GPU_NUM_DEVICES`. Agar kompiler berfungsi dengan baik, Anda harus mengatur variabel lingkungan ini sama dengan jumlah GPU per instance.

Ada tiga pendekatan untuk mengatur variabel `GPU_NUM_DEVICES` lingkungan:

- Pendekatan 1 — Gunakan `environment` argumen kelas SageMaker estimator. Misalnya, jika Anda menggunakan `ml.p3.8xlarge` instance yang memiliki empat GPU, lakukan hal berikut:

```
# Using the SageMaker Python SDK's HuggingFace estimator

hf_estimator=HuggingFace(
    ...
    instance_type="ml.p3.8xlarge",
    hyperparameters={...},
    environment={
        ...
        "GPU_NUM_DEVICES": "4" # corresponds to number of GPUs on the specified
 instance
    },
)
```

- Pendekatan 2 — Gunakan `hyperparameters` argumen kelas SageMaker estimator dan uraikan dalam skrip pelatihan Anda.

  1. Untuk menentukan jumlah GPU, tambahkan pasangan kunci-nilai ke argumen. `hyperparameters`

     Misalnya, jika Anda menggunakan `ml.p3.8xlarge` instance yang memiliki empat GPU, lakukan hal berikut:

```
# Using the SageMaker Python SDK's HuggingFace estimator

hf_estimator=HuggingFace(
    ...
    entry_point = "train.py"
    instance_type= "ml.p3.8xlarge",
    hyperparameters = {
        ...
        "n_gpus": 4 # corresponds to number of GPUs on specified instance
    }
)
hf_estimator.fit()
```

  2. Dalam skrip pelatihan Anda, parse `n_gpus` hyperparameter dan tentukan sebagai input untuk variabel `GPU_NUM_DEVICES` lingkungan.

```
# train.py
import os, argparse
```

```
if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    ...
    # Data, model, and output directories
    parser.add_argument("--output_data_dir", type=str,
 default=os.environ["SM_OUTPUT_DATA_DIR"])
    parser.add_argument("--model_dir", type=str,
 default=os.environ["SM_MODEL_DIR"])
    parser.add_argument("--training_dir", type=str,
 default=os.environ["SM_CHANNEL_TRAIN"])
    parser.add_argument("--test_dir", type=str,
 default=os.environ["SM_CHANNEL_TEST"])
    parser.add_argument("--n_gpus", type=str, default=os.environ["SM_NUM_GPUS"])

    args, _ = parser.parse_known_args()

    os.environ["GPU_NUM_DEVICES"] = args.n_gpus
```

- Pendekatan 3 — Hard-code variabel `GPU_NUM_DEVICES` lingkungan dalam skrip pelatihan Anda. Misalnya, tambahkan berikut ini ke skrip Anda jika Anda menggunakan instance yang memiliki empat GPU.

```
# train.py

import os
os.environ["GPU_NUM_DEVICES"] = 4
```

> **Tip**
>
> Untuk mengetahui jumlah perangkat GPU pada instans pembelajaran mesin yang ingin Anda gunakan, lihat Komputasi Akselerasi di halaman Jenis Instans Amazon EC2.

SageMaker Training Compiler tidak mengurangi total waktu pelatihan

Jika total waktu pelatihan tidak berkurang dengan SageMaker Training Compiler, kami sangat menyarankan Anda untuk memeriksa SageMaker Praktik dan Pertimbangan Terbaik Kompiler Pelatihan halaman untuk memeriksa konfigurasi pelatihan Anda, strategi padding untuk bentuk tensor input, dan hyperparameters.

Catatan Rilis Kompiler SageMaker Pelatihan Amazon

Lihat catatan rilis berikut untuk melacak pembaruan terbaru untuk Amazon SageMaker Training Compiler.

SageMaker Catatan Rilis Training Compiler: 13 Februari 2023

Pembaruan Mata Uang

- Ditambahkan dukungan untuk PyTorch v1.13.1

Perbaikan Bug

- Memperbaiki masalah kondisi balapan pada GPU yang menyebabkan hilangnya NAN pada beberapa model seperti model transformator penglihatan (ViT).

Perubahan Lainnya

- SageMaker Training Compiler meningkatkan kinerja dengan PyTorch membiarkan/XLA untuk secara otomatis menimpa pengoptimal (seperti SGD, Adam, AdamW) di`torch.optim` atau`transformers.optimization` dengan versi syncfree dari mereka di`torch_xla.amp.syncfree` (seperti`torch_xla.amp.syncfree.SGD`,`torch_xla.amp.syncfree.Adam`,`torch_xla.amp.syncf` Anda tidak perlu mengubah baris kode di mana Anda menentukan pengoptimal dalam skrip pelatihan Anda.

Migrasi keAWS Deep Learning Containers

Rilis ini lulus pengujian benchmark dan dimigrasikan ke Kontainer PembelajaranAWS Mendalam berikut:

- PyTorch v1.13.1

  ```
  763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-trcomp-training:1.13.1-gpu-py39-cu117-ubuntu20.04-sagemaker
  ```

  Untuk menemukan daftar lengkap kontainer bawaan dengan Amazon SageMaker Training Compiler, lihatKerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji.

SageMaker Catatan Rilis Training Compiler: 9 Januari 2023

Melanggar Perubahan

- `tf.keras.optimizers.Optimizer`menunjuk ke optimizer baru di TensorFlow 2.11.0 dan kemudian. Pengoptimal lama dipindahkan ke`tf.keras.optimizers.legacy`. Anda mungkin mengalami kegagalan pekerjaan karena perubahan melanggar ketika Anda melakukan hal berikut.
  - Muat pos pemeriksaan dari pengoptimal lama. Kami menyarankan Anda untuk beralih menggunakan pengoptimal lama.
  - Gunakan TensorFlow v1. Kami menyarankan Anda untuk bermigrasi ke TensorFlow v2, atau beralih ke pengoptimal lama jika Anda perlu terus menggunakan TensorFlow v1.

  Untuk daftar perubahan yang lebih rinci dari perubahan pengoptimal, lihat catatan rilis TensorFlow v2.11.0 resmi di TensorFlow GitHub repositori.

Migrasi keAWS Deep Learning Containers

Rilis ini lulus pengujian benchmark dan dimigrasikan ke Kontainer PembelajaranAWS Mendalam berikut:

- TensorFlow v2.11.0

  ```
  763104351884.dkr.ecr.<region>.amazonaws.com/tensorflow-training:2.11.0-gpu-py39-
  cu112-ubuntu20.04-sagemaker
  ```

  Untuk menemukan daftar lengkap kontainer bawaan dengan Amazon SageMaker Training Compiler, lihatKerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji.

SageMaker Catatan Rilis Training Compiler: 8 Desember 2022

Perbaikan Bug

- Memperbaiki benih untuk pekerjaan PyTorch pelatihan mulai PyTorch v1.12 untuk memastikan bahwa tidak ada perbedaan dalam inisialisasi model di seluruh proses yang berbeda. Lihat juga PyTorchReproduktifitas.
- Memperbaiki masalah yang menyebabkan pekerjaan pelatihan PyTorch terdistribusi pada instans G4dn dan G5 tidak default untuk komunikasi melalui PCIe.

Masalah yang diketahui

- Penggunaan API PyTorch /XLA yang tidak tepat dalam transformator penglihatan Hugging Face dapat menyebabkan masalah konvergensi.

Perubahan Lainnya

- Saat menggunakan`Trainer` kelas Hugging Face Transformers, pastikan Anda menggunakan SyncFree pengoptimal dengan mengatur`optim` argumen `adamw_torch_xla`. Untuk informasi selengkapnya, lihat Model Bahasa Besar Menggunakan Kelas Memeluk Wajah Transformers `Trainer`. Lihat juga Optimizer dalam dokumentasi Hugging Face Transformers.

Migrasi keAWS Deep Learning Containers

Rilis ini lulus pengujian benchmark dan dimigrasikan ke Kontainer PembelajaranAWS Mendalam berikut:

- PyTorch v1.12.0

  ```
  763104351884.dkr.ecr.<region>.amazonaws.com/pytorch-trcomp-training:1.12.0-gpu-py38-cu113-ubuntu20.04-sagemaker
  ```

  Untuk menemukan daftar lengkap kontainer bawaan dengan Amazon SageMaker Training Compiler, lihatKerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji.

SageMaker Catatan Rilis Training Compiler: 4 Oktober 2022

Pembaruan Mata Uang

- Ditambahkan dukungan untuk TensorFlow v2.10.0.

Perubahan Lainnya

- Menambahkan model Hugging Face NLP menggunakan perpustakaan Transformers ke tes TensorFlow kerangka kerja. Untuk menemukan model Transformer yang diuji, lihatthe section called “Model yang Diuji”.

Migrasi keAWS Deep Learning Containers

Rilis ini lulus pengujian benchmark dan dimigrasikan ke Kontainer PembelajaranAWS Mendalam berikut:

- TensorFlow v2.10.0

```
763104351884.dkr.ecr.<region>.amazonaws.com/tensorflow-training:2.10.0-gpu-py39-
cu112-ubuntu20.04-sagemaker
```

  Untuk menemukan daftar lengkap kontainer bawaan dengan Amazon SageMaker Training Compiler, lihatKerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji.

SageMaker Catatan Rilis Training Compiler: 1 September 2022

Pembaruan Mata Uang

- Ditambahkan dukungan untuk Memeluk Wajah Transformers v4.21.1 dengan PyTorch v1.11.0.

Peningkatan

- Menerapkan mekanisme peluncur pelatihan terdistribusi baru untuk mengaktifkan SageMaker Training Compiler untuk model Hugging Face Transformer dengan PyTorch. Untuk mempelajari lebih lanjut, lihat Menjalankan Pekerjaan PyTorch SageMaker Pelatihan dengan Kompiler Pelatihan untuk Pelatihan Terdistribusi.
- Terintegrasi dengan EFA untuk meningkatkan komunikasi kolektif dalam pelatihan terdistribusi.
- Menambahkan dukungan untuk instans G5 untuk pekerjaan PyTorch pelatihan. Untuk informasi selengkapnya, lihat the section called “Kerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji”.

Migrasi keAWS Deep Learning Containers

Rilis ini lulus pengujian benchmark dan dimigrasikan ke Kontainer PembelajaranAWS Mendalam berikut:

- HuggingFace v4.21.1 dengan PyTorch v1.11.0

```
763104351884.dkr.ecr.us-west-2.amazonaws.com/huggingface-pytorch-trcomp-
training:1.11.0-transformers4.21.1-gpu-py38-cu113-ubuntu20.04
```

Untuk menemukan daftar lengkap kontainer bawaan dengan Amazon SageMaker Training Compiler, lihatKerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji.

SageMaker Catatan Rilis Training Compiler: 14 Juni 2022

Fitur Baru

- Ditambahkan dukungan untuk TensorFlow v2.9.1. SageMaker Training Compiler sepenuhnya mendukung TensorFlow modul kompilasi (`tf.*`) dan modul TensorFlow Keras (`tf.keras.*`).
- Menambahkan dukungan untuk wadah khusus yang dibuat dengan memperluasAWS Deep Learning Containers untuk TensorFlow. Untuk informasi selengkapnya, lihat Mengaktifkan Kompiler SageMaker Pelatihan Menggunakan SDK SageMaker Python dan Extending SageMaker Framework Deep Learning Containers.
- Menambahkan dukungan untuk instans G5 untuk pekerjaan TensorFlow pelatihan.

Migrasi keAWS Deep Learning Containers

Rilis ini lulus pengujian benchmark dan dimigrasikan ke Kontainer PembelajaranAWS Mendalam berikut:

- TensorFlow 2.9.1

```
763104351884.dkr.ecr.<region>.amazonaws.com/tensorflow-training:2.9.1-gpu-py39-cu112-
ubuntu20.04-sagemaker
```

Untuk menemukan daftar lengkap kontainer yang sudah dibuat sebelumnya dengan Amazon SageMaker Training Compiler, lihatKerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji.

SageMaker Catatan Rilis Training Compiler: 26 April 2022

Peningkatan

- Menambahkan dukungan untuk semuaWilayah AWS tempat AWSDeep Learning Containers berada dalam layanan kecuali wilayah China.

SageMaker Catatan Rilis Training Compiler: 12 April 2022

Pembaruan Mata Uang

- Ditambahkan dukungan untuk Memeluk Wajah Transformers v4.17.0 dengan TensorFlow v2.6.3 dan PyTorch v1.10.2.

SageMaker Catatan Rilis Training Compiler: 21 Februari 2022

Peningkatan

- Tes benchmark yang diselesaikan dan kecepatan pelatihan yang dikonfirmasi pada jenis`ml.g4dn` instans. Untuk menemukan daftar lengkap`ml` instance yang diuji, lihatTipe Instans Yang Didukung.

SageMaker Catatan Rilis Training Compiler: 01 Desember 2021

Fitur Baru

- Meluncurkan Kompiler SageMaker Pelatihan Amazon diAWS Re:Invent 2021.

Migrasi keAWS Deep Learning Containers

- Amazon SageMaker Training Compiler lulus pengujian benchmark dan dimigrasi keAWS Deep Learning Containers. Untuk menemukan daftar lengkap kontainer bawaan dengan Amazon SageMaker Training Compiler, lihatKerangka Kerja yang DidukungWilayah AWS, Jenis Instance, dan Model yang Diuji.

Akses Data Pelatihan

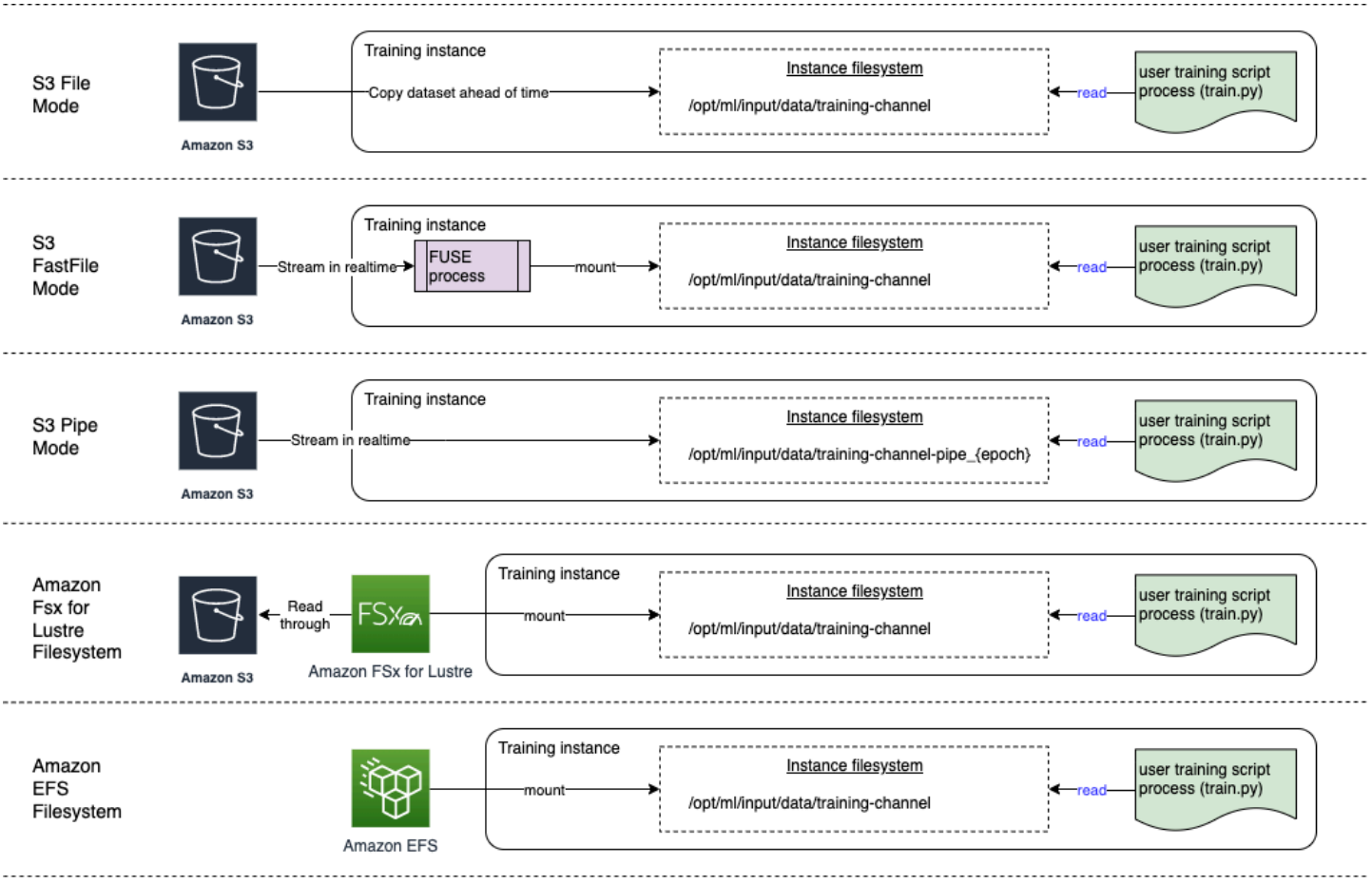
Saat Anda membuat pekerjaan pelatihan, Anda menentukan lokasi kumpulan data pelatihan dan mode input untuk mengakses kumpulan data. Untuk lokasi data, Amazon SageMaker mendukung

Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), Amazon Elastic File System (Amazon EFS), dan Amazon FSx for Lustre. Mode input menentukan apakah akan mengalirkan file data dari kumpulan data secara real time atau mengunduh seluruh kumpulan data pada awal pekerjaan pelatihan.

> **Note**
>
> Dataset input Anda harus Wilayah AWS sama dengan pekerjaan pelatihan Anda.

SageMaker Mode Input dan Penyimpanan AWS Cloud

Bagian ini merangkum mode SageMaker input untuk Amazon S3 dan sistem file di Amazon EFS dan Amazon FSx for Lustre.



- Mode file menyajikan tampilan sistem file dari kumpulan data ke wadah pelatihan. Ini adalah mode input default jika Anda tidak secara eksplisit menentukan salah satu dari dua opsi lainnya. Jika Anda menggunakan mode file, SageMaker unduh data pelatihan dari lokasi penyimpanan

ke direktori lokal di wadah Docker. Pelatihan dimulai setelah kumpulan data lengkap diunduh. Dalam mode file, instance pelatihan harus memiliki ruang penyimpanan yang cukup agar sesuai dengan seluruh kumpulan data. Kecepatan unduh mode file tergantung pada ukuran kumpulan data, ukuran rata-rata file, dan jumlah file. Anda dapat mengonfigurasi kumpulan data untuk mode file dengan menyediakan awalan Amazon S3, file manifes, atau file manifes tambahan. Anda harus menggunakan awalan S3 ketika semua file dataset Anda berada dalam awalan S3 umum. Mode file kompatibel dengan mode SageMaker lokal (memulai wadah SageMaker pelatihan secara interaktif dalam hitungan detik). Untuk pelatihan terdistribusi, Anda dapat memisahkan kumpulan data di beberapa instance dengan opsi. `ShardedByS3Key`

- Mode file cepat menyediakan akses sistem file ke sumber data Amazon S3 sambil memanfaatkan keunggulan kinerja mode pipa. Pada awal pelatihan, mode file cepat mengidentifikasi file data tetapi tidak mengunduhnya. Pelatihan dapat dimulai tanpa menunggu seluruh kumpulan data diunduh. Ini berarti bahwa startup pelatihan membutuhkan waktu lebih sedikit ketika ada lebih sedikit file di awalan Amazon S3 yang disediakan.

  Berbeda dengan mode pipa, mode file cepat bekerja dengan akses acak ke data. Namun, ini berfungsi paling baik ketika data dibaca secara berurutan. Mode file cepat tidak mendukung file manifes tambahan.

  Mode file cepat mengekspos objek S3 menggunakan antarmuka sistem file yang sesuai dengan POSIX, seolah-olah file tersedia di disk lokal instance pelatihan Anda. Ini mengalirkan konten S3 sesuai permintaan karena skrip pelatihan Anda mengkonsumsi data. Ini berarti bahwa kumpulan data Anda tidak perlu lagi masuk ke dalam ruang penyimpanan instans pelatihan secara keseluruhan, dan Anda tidak perlu menunggu dataset diunduh ke instans pelatihan sebelum pelatihan dimulai. File cepat saat ini hanya mendukung awalan S3 (tidak mendukung manifes dan augmented manifest). Mode file cepat kompatibel dengan mode SageMaker lokal.

- Mode pipa mengalirkan data langsung dari sumber data Amazon S3. Streaming dapat memberikan waktu mulai yang lebih cepat dan throughput yang lebih baik daripada mode file.

  Saat melakukan streaming data secara langsung, Anda dapat mengurangi ukuran volume Amazon EBS yang digunakan oleh instans pelatihan. Mode pipa hanya membutuhkan ruang disk yang cukup untuk menyimpan artefak model akhir.

  Ini adalah mode streaming lain yang sebagian besar digantikan oleh mode file yang lebih baru dan simpler-to-use cepat. Dalam mode pipa, data diambil sebelumnya dari Amazon S3 pada konkurensi dan throughput tinggi, dan dialirkan ke pipa bernama, yang juga dikenal sebagai pipa First-In-First-Out (FIFO) karena perilakunya. Setiap pipa hanya dapat dibaca dengan satu proses.

Ekstensi SageMaker khusus untuk mengintegrasikan mode Pipe dengan TensorFlow mudah ke pemuat TensorFlow data asli untuk streaming teks, TFRecords, atau format file Recordio. Mode pipa juga mendukung sharding dan shuffling data yang dikelola.

- Amazon S3 Express One Zone adalah kelas penyimpanan Availability Zone tunggal berkinerja tinggi yang dapat memberikan akses data milidetik satu digit yang konsisten untuk aplikasi yang paling sensitif terhadap latensi termasuk pelatihan model. SageMaker Amazon S3 Express One Zone memungkinkan pelanggan untuk mengumpulkan penyimpanan objek mereka dan menghitung sumber daya dalam satu AWS Availability Zone, mengoptimalkan kinerja komputasi dan biaya dengan peningkatan kecepatan pemrosesan data. Untuk lebih meningkatkan kecepatan akses dan mendukung ratusan ribu permintaan per detik, data disimpan dalam jenis bucket baru, bucket direktori Amazon S3.

  SageMaker pelatihan model mendukung bucket direktori Amazon S3 Express One Zone berkinerja tinggi sebagai lokasi input data untuk mode file, mode file cepat, dan mode pipa. Untuk menggunakan Amazon S3 Express One Zone, masukkan lokasi bucket direktori Amazon S3 Express One Zone, bukan bucket Amazon S3. Berikan ARN untuk peran IAM dengan kontrol akses dan kebijakan izin yang diperlukan. Lihat AmazonSageMakerFullAccesspolicyuntuk detailnya. Untuk informasi selengkapnya, lihat S3 Express One Zone.

- Amazon FSx for Lustre — FSx for Lustre dapat menskalakan hingga ratusan gigabyte throughput dan jutaan IOPS dengan pengambilan file latensi rendah. Saat memulai pekerjaan pelatihan, SageMaker pasang sistem file FSx for Lustre ke sistem file instance pelatihan, lalu mulai skrip pelatihan Anda. Pemasangan itu sendiri adalah operasi yang relatif cepat yang tidak bergantung pada ukuran kumpulan data yang disimpan di FSx for Lustre.

  Untuk mengakses FSx for Lustre, tugas pelatihan Anda harus terhubung ke Amazon Virtual Private Cloud (VPC), yang memerlukan penyiapan dan keterlibatan. DevOps Untuk menghindari biaya transfer data, sistem file menggunakan Availability Zone tunggal, dan Anda perlu menentukan subnet VPC yang memetakan ke ID Availability Zone ini saat menjalankan tugas pelatihan.

- Amazon EFS — Untuk menggunakan Amazon EFS sebagai sumber data, data harus sudah berada di Amazon EFS sebelum pelatihan. SageMaker memasang sistem file Amazon EFS yang ditentukan ke instance pelatihan, lalu mulai skrip latihan Anda. Pekerjaan pelatihan Anda harus terhubung ke VPC untuk mengakses Amazon EFS.

> **Tip**
>
> Untuk mempelajari lebih lanjut tentang cara menentukan konfigurasi VPC Anda ke SageMaker estimator, lihat Menggunakan Sistem File sebagai Input Pelatihan dalam dokumentasi Python SageMaker SDK.

Memilih Mode Input Data Menggunakan SageMaker Python SDK

SageMaker Python SDK menyediakan kelas Estimator generik dan variasinya untuk kerangka kerja ML untuk meluncurkan pekerjaan pelatihan. Anda dapat menentukan salah satu mode input data saat mengkonfigurasi SageMaker `Estimator` kelas atau `Estimator.fit` metode. Template kode berikut menunjukkan dua cara untuk menentukan mode input.

Untuk menentukan mode input menggunakan kelas Estimator

```
from sagemaker.estimator import Estimator
from sagemaker.inputs import TrainingInput

estimator = Estimator(
    checkpoint_s3_uri='s3://my-bucket/checkpoint-destination/',
    output_path='s3://my-bucket/output-path/',
    base_job_name='job-name',
    input_mode='File'  # Available options: File | Pipe | FastFile
    ...
)

# Run the training job
estimator.fit(
    inputs=TrainingInput(s3_data="s3://my-bucket/my-data/train")
)
```

Untuk informasi selengkapnya, lihat kelas SageMaker.Estimator.Estimator dalam dokumentasi Python SDK. SageMaker

Untuk menentukan mode input melalui metode Estimator fit

```
from sagemaker.estimator import Estimator
from sagemaker.inputs import TrainingInput
```

```
estimator = Estimator(
    checkpoint_s3_uri='s3://my-bucket/checkpoint-destination/',
    output_path='s3://my-bucket/output-path/',
    base_job_name='job-name',
    ...
)

# Run the training job
estimator.fit(
    inputs=TrainingInput(
        s3_data="s3://my-bucket/my-data/train",
        input_mode='File'  # Available options: File | Pipe | FastFile
    )
)
```

Untuk informasi selengkapnya, lihat metode kelas SageMaker.Estimator.Fit dan sagemaker.inputs.TrainingInputkelas dalam dokumentasi SageMaker Python SDK.

> **Tip**
>
> Untuk mempelajari lebih lanjut tentang cara mengonfigurasi Amazon FSx for Lustre atau Amazon EFS dengan SageMaker konfigurasi VPC menggunakan estimator SDK Python, lihat Menggunakan Sistem File sebagai Input Pelatihan dalam dokumentasi Python SDK. SageMaker

> **Tip**
>
> Integrasi mode input data dengan Amazon S3, Amazon EFS, dan FSx for Lustre adalah cara yang disarankan untuk mengonfigurasi sumber data secara optimal untuk praktik terbaik. Anda dapat meningkatkan kinerja pemuatan data secara strategis menggunakan opsi penyimpanan SageMaker terkelola dan mode input, tetapi tidak dibatasi secara ketat. Anda dapat menulis logika pembacaan data Anda sendiri langsung di wadah pelatihan Anda. Misalnya, Anda dapat mengatur untuk membaca dari sumber data yang berbeda, menulis kelas pemuat data S3 Anda sendiri, atau menggunakan fungsi pemuatan data kerangka kerja pihak ketiga dalam skrip pelatihan Anda. Namun, Anda harus memastikan bahwa Anda menentukan jalur yang benar yang SageMaker dapat mengenali.

> **Tip**
>
> Jika Anda menggunakan wadah pelatihan khusus, pastikan Anda menginstal toolkit SageMaker pelatihan yang membantu mengatur lingkungan untuk pekerjaan SageMaker pelatihan. Jika tidak, Anda harus menentukan variabel lingkungan secara eksplisit di Dockerfile Anda. Untuk informasi selengkapnya, lihat Membuat wadah dengan algoritme dan model Anda sendiri.

Untuk informasi selengkapnya tentang cara menyetel mode input data menggunakan SageMaker API tingkat rendahBagaimana Amazon SageMaker Menyediakan Informasi Pelatihan, lihat, `CreateTrainingJob`API, dan `TrainingInputMode` in `AlgorithmSpecification`.

Konfigurasikan Saluran Input Data untuk Menggunakan Amazon FSx for Lustre

Pelajari cara menggunakan Amazon FSx for Lustre sebagai sumber data Anda untuk throughput yang lebih tinggi dan pelatihan yang lebih cepat dengan mengurangi waktu pemuatan data.

Sinkronkan Amazon S3 dan Amazon FSx for Lustre

Untuk menautkan Amazon S3 Anda ke Amazon FSx for Lustre dan mengunggah kumpulan data pelatihan Anda, lakukan hal berikut.

1. Siapkan kumpulan data Anda dan unggah ke bucket Amazon S3. Misalnya, asumsikan bahwa jalur Amazon S3 untuk kumpulan data kereta api dan kumpulan data pengujian dalam format berikut.

   ```
   s3://my-bucket/data/train
   s3://my-bucket/data/test
   ```

2. Untuk membuat sistem file FSx for Lustre yang ditautkan dengan bucket Amazon S3 dengan data pelatihan, ikuti langkah-langkah di Menautkan sistem file Anda ke bucket Amazon S3 di Panduan Pengguna Amazon FSx for Lustre. Pastikan Anda menambahkan titik akhir ke VPC yang memungkinkan akses Amazon S3. Untuk informasi selengkapnya, lihat the section called “Buat VPC Endpoint Amazon S3”. Saat Anda menentukan jalur repositori Data, berikan URI bucket Amazon S3 dari folder yang berisi kumpulan data Anda. Misalnya, berdasarkan contoh jalur S3 di langkah 1, jalur repositori data harus sebagai berikut.

```
s3://my-bucket/data
```

3. Setelah sistem file FSx for Lustre dibuat, periksa informasi konfigurasi dengan menjalankan perintah berikut.

```
aws fsx describe-file-systems && \
aws fsx describe-data-repository-association
```

Perintah-perintah ini kembali `FileSystemId`,`MountName`,`FileSystemPath`,, dan`DataRepositoryPath`. Misalnya, outputnya akan terlihat seperti berikut ini.

```
# Output of aws fsx describe-file-systems
"FileSystemId": "fs-0123456789abcdef0"
"MountName": "1234abcd"

# Output of aws fsx describe-data-repository-association
"FileSystemPath": "/ns1",
"DataRepositoryPath": "s3://my-bucket/data/"
```

Setelah sinkronisasi antara Amazon S3 dan Amazon FSx selesai, kumpulan data Anda disimpan di Amazon FSx di direktori berikut.

```
/ns1/train  # synced with s3://my-bucket/data/train
/ns1/test   # synced with s3://my-bucket/data/test
```

Atur jalur sistem file Amazon FSx sebagai saluran input data untuk pelatihan SageMaker

Prosedur berikut memandu Anda melalui proses pengaturan sistem file Amazon FSx sebagai sumber data untuk pekerjaan SageMaker pelatihan.

Using the SageMaker Python SDK

Untuk mengatur sistem file Amazon FSx dengan benar sebagai sumber data, konfigurasikan kelas SageMaker estimator dan `FileSystemInput` gunakan instruksi berikut.

1. Konfigurasikan objek FileSystemInput kelas.

```
from sagemaker.inputs import FileSystemInput

train_fs = FileSystemInput(
    file_system_id="fs-0123456789abcdef0",
    file_system_type="FSxLustre",
    directory_path="/1234abcd/ns1/",
    file_system_access_mode="ro",
)
```

> **Tip**
>
> Saat Anda menentukan`directory_path`, pastikan Anda menyediakan jalur sistem file Amazon FSx yang dimulai dengan. `MountName`

2. Konfigurasikan SageMaker estimator dengan konfigurasi VPC yang digunakan untuk sistem file Amazon FSx.

```
from sagemaker.estimator import Estimator

estimator = Estimator(
    ...
    role="your-iam-role-with-access-to-your-fsx",
    subnets=["subnet-id"],  # Should be the same as the subnet used for Amazon FSx
    security_group_ids="security-group-id"
)
```

3. Luncurkan pekerjaan pelatihan dengan menjalankan metode estimator.fit dengan sistem file Amazon FSx.

```
estimator.fit(train_fs)
```

Untuk menemukan lebih banyak contoh kode, lihat Menggunakan Sistem File sebagai Input Pelatihan dalam dokumentasi SageMaker Python SDK.

Using the SageMaker CreateTrainingJob API

Sebagai bagian dari CreateTrainingJobpermintaan JSON, konfigurasikan `InputDataConfig` sebagai berikut.

```
"InputDataConfig": [
    {
        "ChannelName": "string",
        "DataSource": {
            "FileSystemDataSource": {
                "DirectoryPath": "/1234abcd/ns1/",
                "FileSystemAccessMode": "ro",
                "FileSystemId": "fs-0123456789abcdef0",
                "FileSystemType": "FSxLustre"
            }
        }
    }
],
```

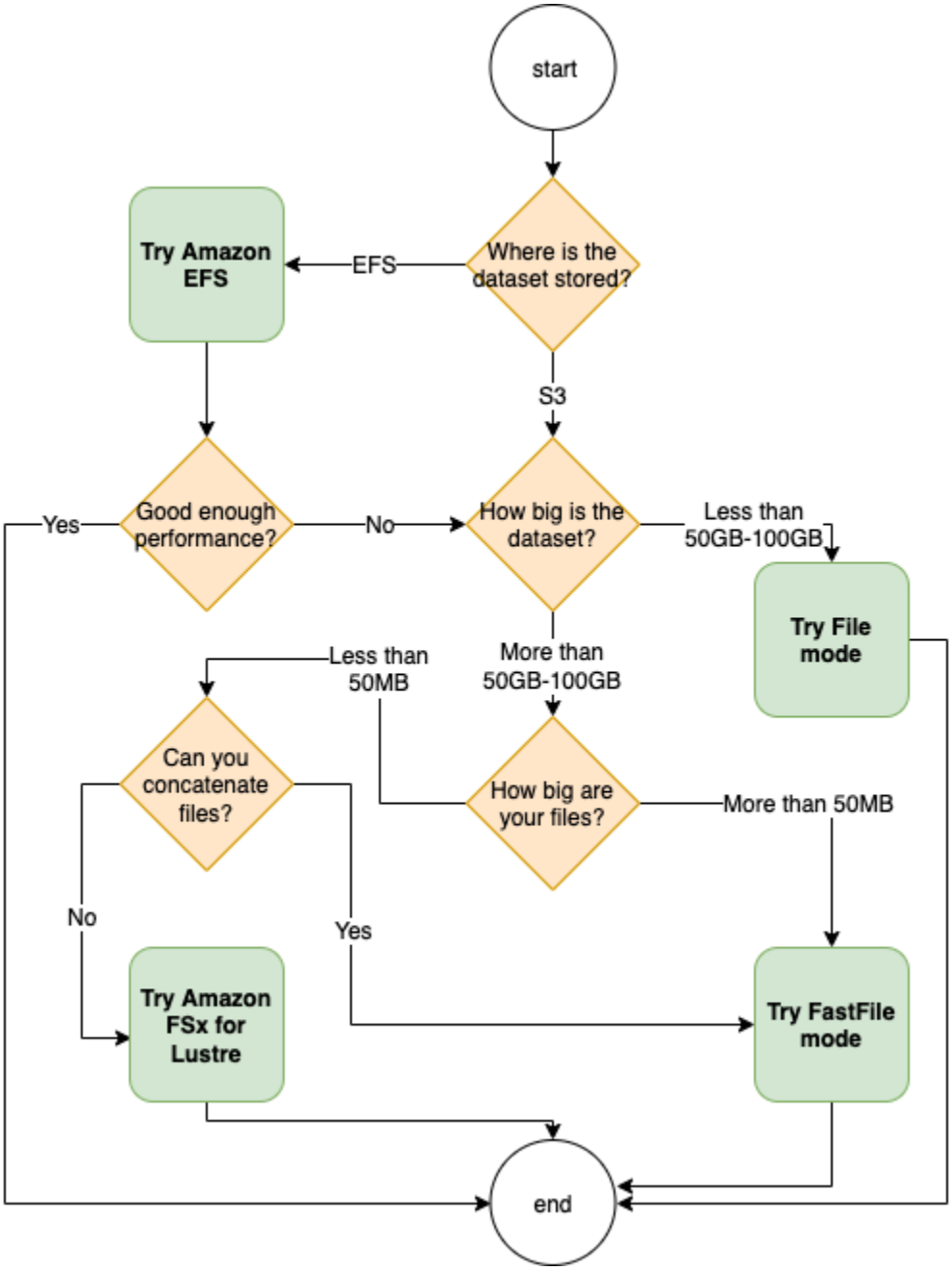
> **Tip**
>
> Saat Anda menentukan`DirectoryPath`, pastikan Anda menyediakan jalur sistem file Amazon FSx yang dimulai dengan. `MountName`

Tips dan Pertimbangan Saat Mengkonfigurasi FSx for Lustre

1. Saat Anda menggunakan instance berkemampuan EFA seperti P4d dan P3dn, pastikan Anda menetapkan aturan masuk dan output yang sesuai di grup keamanan. Khususnya, membuka port ini diperlukan SageMaker untuk mengakses sistem file Amazon FSx dalam pekerjaan pelatihan. Untuk mempelajari selengkapnya, lihat Kontrol Akses Sistem File dengan Amazon VPC.
2. Pastikan Peran IAM yang digunakan untuk meluncurkan pekerjaan SageMaker pelatihan memiliki akses ke Amazon FSx.

Praktik Terbaik untuk Memilih Sumber Data dan Mode Input

Sumber data terbaik untuk pekerjaan pelatihan Anda bergantung pada karakteristik beban kerja seperti ukuran kumpulan data, format file, ukuran rata-rata file, durasi pelatihan, pola baca pemuat data berurutan atau acak, dan seberapa cepat model Anda dapat mengkonsumsi data pelatihan. Praktik terbaik berikut memberikan panduan untuk memulai mode input dan penyimpanan data yang paling sesuai untuk kasus penggunaan Anda.



This flowchart summarizes and visualizes best practices of choosing the best storage as the data source and input file mode. All of the cases in the flowchart are described in the following sections.

Kapan menggunakan Amazon EFS

Jika kumpulan data Anda disimpan di Amazon Elastic File System, Anda mungkin memiliki aplikasi pra-pemrosesan atau anotasi yang menggunakan Amazon EFS untuk penyimpanan. Anda dapat menjalankan pekerjaan pelatihan yang dikonfigurasi dengan saluran data yang mengarah ke sistem file Amazon EFS. Untuk informasi selengkapnya, lihat Mempercepat pelatihan di Amazon SageMaker

menggunakan sistem file Amazon FSx for Lustre dan Amazon EFS. Jika Anda tidak dapat mencapai kinerja yang lebih baik, periksa opsi pengoptimalan Anda mengikuti panduan kinerja Amazon Elastic File System atau pertimbangkan untuk menggunakan mode input atau penyimpanan data yang berbeda.

Gunakan mode file untuk kumpulan data kecil

Jika kumpulan data disimpan di Amazon Simple Storage Service dan volume keseluruhannya relatif kecil (misalnya, kurang dari 50-100 GB), coba gunakan mode file. Overhead mengunduh dataset 50 GB dapat bervariasi berdasarkan jumlah total file. Misalnya, dibutuhkan sekitar 5 menit jika kumpulan data dibagi menjadi pecahan 100 MB. Apakah overhead startup ini dapat diterima terutama tergantung pada durasi keseluruhan pekerjaan pelatihan Anda, karena fase pelatihan yang lebih lama berarti fase pengunduhan yang lebih kecil secara proporsional.

Serialisasi banyak file kecil

Jika ukuran dataset Anda kecil (kurang dari 50-100 GB), tetapi terdiri dari banyak file kecil (kurang dari 50 MB per file), overhead unduhan mode file bertambah, karena setiap file perlu diunduh satu per satu dari Amazon Simple Storage Service ke volume instans pelatihan. Untuk mengurangi overhead dan waktu traversal data ini secara umum, pertimbangkan serialisasi grup file kecil tersebut ke dalam wadah file yang lebih kecil (seperti 150 MB per file) dengan menggunakan format file, seperti TFRecord for, for, dan PyTorch Recordio TensorFlow untuk _WebDatasetMxNet.

Kapan menggunakan mode file cepat

Untuk kumpulan data yang lebih besar dengan file yang lebih besar (lebih dari 50 MB per file), opsi pertama adalah mencoba mode file cepat, yang lebih mudah digunakan daripada FSx for Lustre karena tidak memerlukan pembuatan sistem file, atau menghubungkan ke VPC. Mode file cepat sangat ideal untuk wadah file besar (lebih dari 150 MB), dan mungkin juga cocok dengan file lebih dari 50 MB. Karena mode file cepat menyediakan antarmuka POSIX, ia mendukung pembacaan acak (membaca rentang byte non-sekuensial). Namun, ini bukan kasus penggunaan yang ideal, dan throughput Anda mungkin lebih rendah dibandingkan dengan pembacaan berurutan. Namun, jika Anda memiliki model HTML yang relatif besar dan intensif secara komputasi, mode file cepat mungkin masih dapat memenuhi bandwidth efektif dari pipa pelatihan dan tidak mengakibatkan kemacetan IO. Anda harus bereksperimen dan melihat. Untuk beralih dari mode file ke mode file cepat (dan kembali), cukup tambahkan (atau hapus) `input_mode='FastFile'` parameter saat menentukan saluran input Anda menggunakan SageMaker Python SDK:

```
sagemaker.inputs.TrainingInput(S3_INPUT_FOLDER,  input_mode = 'FastFile')
```

Kapan menggunakan Amazon FSx for Lustre

Jika dataset Anda terlalu besar untuk mode file, memiliki banyak file kecil yang tidak dapat Anda serialisasi dengan mudah, atau menggunakan pola akses baca acak, FSx for Lustre adalah pilihan yang baik untuk dipertimbangkan. Sistem filenya berskala hingga ratusan gigabyte per detik (GB/s) throughput dan jutaan IOPS, yang ideal ketika Anda memiliki banyak file kecil. Namun, perhatikan bahwa mungkin ada masalah cold start karena pemuatan lambat dan overhead pengaturan dan inisialisasi sistem file FSx for Lustre.

> **Tip**
>
> Untuk mempelajari lebih lanjut, lihat Memilih sumber data terbaik untuk pekerjaan SageMaker pelatihan Amazon Anda. Blog AWS machine learning ini lebih lanjut membahas studi kasus dan tolok ukur kinerja sumber data dan mode input.

Melatih Menggunakan Cluster Heterogen

Dengan menggunakan fitur klaster heterogen dari SageMaker Pelatihan, Anda dapat menjalankan tugas pelatihan dengan beberapa jenis instans MLuntuk penskalaan dan pemanfaatan sumber daya yang lebih baik untuk tugas dan tujuan pelatihan ML yang berbeda. Misalnya, jika pekerjaan pelatihan Anda di cluster dengan instans GPU mengalami pemanfaatan GPU yang rendah dan masalah bottleneck CPU karena tugas intensif CPU, menggunakan cluster heterogen dapat membantu membongkar tugas intensif CPU dengan menambahkan grup instans CPU yang lebih hemat biaya, menyelesaikan masalah bottleneck tersebut, dan mencapai pemanfaatan GPU yang lebih baik.

> **Note**
>
> Fitur ini tersedia di SageMaker Python SDK v2.98.0 dan versi lebih baru.

> **Note**
>
> Fitur ini tersedia melalui kelas SageMaker PyTorchdan TensorFlowkerangka estimator. Framework yang didukung adalah PyTorch v1.10 atau yang lebih baru dan TensorFlow v2.6 atau yang lebih baru.

Topik

Cara Mengkonfigurasi Cluster Heterogen

Bagian ini memberikan petunjuk tentang cara menjalankan pekerjaan pelatihan menggunakan klaster heterogen yang terdiri dari beberapa jenis instans.

Topik

Menggunakan SageMaker Python SDK

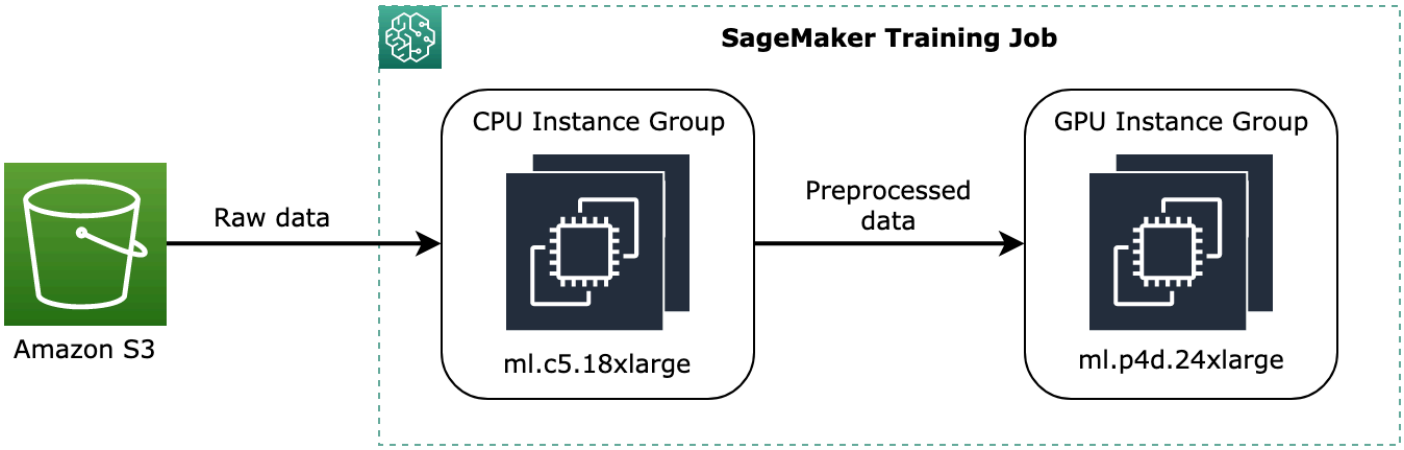
Ikuti petunjuk tentang cara mengonfigurasi grup instans untuk klaster heterogen menggunakan Python SDK. SageMaker

1. Untuk mengonfigurasi grup instance dari cluster heterogen untuk pekerjaan pelatihan, gunakan kelas. `sagemaker.instance_group.InstanceGroup` Anda dapat menentukan nama khusus untuk setiap grup instans, jenis instance, dan jumlah instance untuk setiap grup instans. Untuk informasi selengkapnya, lihat sagemaker.instance_group. InstanceGroupdalam dokumentasi SageMakerPython SDK.

   > **Note**
   >
   > Untuk informasi selengkapnya tentang jenis instans yang tersedia dan jumlah maksimum grup instans yang dapat Anda konfigurasikan dalam klaster heterogen, lihat referensi API. InstanceGroup

   Contoh kode berikut menunjukkan bagaimana untuk mengatur dua kelompok contoh yang terdiri dari dua `ml.c5.18xlarge` CPU-satunya contoh bernama `instance_group_1` dan satu contoh

`ml.p3dn.24xlarge` GPU bernama`instance_group_2`, seperti yang ditunjukkan dalam diagram berikut.



The preceding diagram shows a conceptual example of how pre-training processes, such as data preprocessing, can be assigned to the CPU instance group and stream the preprocessed data to the GPU instance group.

```
from sagemaker.instance_group import InstanceGroup

instance_group_1 = InstanceGroup(
    "instance_group_1", "ml.c5.18xlarge", 2
)
instance_group_2 = InstanceGroup(
    "instance_group_2", "ml.p3dn.24xlarge", 1
)
```

2. Menggunakan objek grup instance, atur saluran input pelatihan dan tetapkan grup instance ke saluran melalui `instance_group_names` argumen sagemaker.inputs. TrainingInputkelas. `instance_group_names`Argumen menerima daftar string nama grup instance.

   Contoh berikut menunjukkan cara mengatur dua saluran input pelatihan dan menetapkan grup instance yang dibuat dalam contoh langkah sebelumnya. Anda juga dapat menentukan jalur bucket Amazon S3 ke `s3_data` argumen untuk grup instans untuk memproses data untuk tujuan penggunaan Anda.

```
from sagemaker.inputs import TrainingInput

training_input_channel_1 = TrainingInput(
    s3_data_type='S3Prefix', # Available Options: S3Prefix | ManifestFile |
 AugmentedManifestFile
    s3_data='s3://your-training-data-storage/folder1',
```

```
    distribution='FullyReplicated', # Available Options: FullyReplicated |
 ShardedByS3Key
    input_mode='File', # Available Options: File | Pipe | FastFile
    instance_groups=["instance_group_1"]
)

training_input_channel_2 = TrainingInput(
    s3_data_type='S3Prefix',
    s3_data='s3://your-training-data-storage/folder2',
    distribution='FullyReplicated',
    input_mode='File',
    instance_groups=["instance_group_2"]
)
```

Untuk informasi lebih lanjut tentang argumen`TrainingInput`, lihat link berikut.

- The sagemaker.input. TrainingInputkelas dalam dokumentasi SageMakerPython SDK
- DataSourceAPI S3 dalam Referensi SageMakerAPI

3. Mengkonfigurasi SageMaker estimator dengan `instance_groups` argumen seperti yang ditunjukkan dalam contoh kode berikut. `instance_groups`Argumen menerima daftar objek. `InstanceGroup`

   PyTorch

   ```
   from sagemaker.pytorch import PyTorch

   estimator = PyTorch(
       ...
       entry_point='my-training-script.py',
       framework_version='x.y.z',    # 1.10.0 or later
       py_version='pyxy',
       job_name='my-training-job-with-heterogeneous-cluster',
       instance_groups=[instance_group_1, instance_group_2]
   )
   ```

   TensorFlow

   ```
   from sagemaker.tensorflow import TensorFlow

   estimator = TensorFlow(
       ...
       entry_point='my-training-script.py',
   ```

```
    framework_version='x.y.z', # 2.6.0 or later
    py_version='pyxy',
    job_name='my-training-job-with-heterogeneous-cluster',
    instance_groups=[instance_group_1, instance_group_2]
)
```

> **Note**
>
> Pasangan `instance_type` dan `instance_count` argumen dan `instance_groups` argumen kelas SageMaker estimator saling eksklusif. Untuk pelatihan cluster homogen, gunakan `instance_type` dan pasangan `instance_count` argumen. Untuk pelatihan cluster heterogen, gunakan. `instance_groups`

> **Note**
>
> Untuk menemukan daftar lengkap container framework, versi framework, dan versi Python yang tersedia, lihat SageMakerFramework Container di GitHub repositori AWS Deep Learning Container.

4. Konfigurasikan `estimator.fit` metode dengan saluran input pelatihan yang dikonfigurasi dengan grup instans dan mulai pekerjaan pelatihan.

```
estimator.fit(
    inputs={
        'training': training_input_channel_1,
        'dummy-input-channel': training_input_channel_2
    }
)
```

Menggunakan API Tingkat Rendah SageMaker

Jika Anda menggunakan AWS Command Line Interface atau AWS SDK for Python (Boto3) dan ingin menggunakan SageMaker API tingkat rendah untuk mengirimkan permintaan pekerjaan pelatihan dengan klaster heterogen, lihat referensi API berikut.

- CreateTrainingJob
- ResourceConfig

- InstanceGroup
- S3 DataSource

Pelatihan Terdistribusi dengan Cluster Heterogen

Melalui `distribution` argumen kelas SageMaker estimator, Anda dapat menetapkan grup instans tertentu untuk menjalankan pelatihan terdistribusi. Misalnya, asumsikan bahwa Anda memiliki dua grup instans berikut dan ingin menjalankan pelatihan multi-GPU pada salah satunya.

```
from sagemaker.instance_group import InstanceGroup

instance_group_1 = InstanceGroup("instance_group_1", "ml.c5.18xlarge", 1)
instance_group_2 = InstanceGroup("instance_group_2", "ml.p3dn.24xlarge", 2)
```

Anda dapat mengatur konfigurasi pelatihan terdistribusi untuk salah satu grup instans. Misalnya, contoh kode berikut menunjukkan cara menetapkan `training_group_2` dengan dua `ml.p3dn.24xlarge` instance ke konfigurasi pelatihan terdistribusi.

> **Note**
>
> Saat ini, hanya satu kelompok instance dari cluster heterogen yang dapat ditentukan ke konfigurasi distribusi.

Dengan MPI

PyTorch

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(
    ...
    instance_groups=[instance_group_1, instance_group_2],
    distribution={
        "mpi": {
            "enabled": True, "processes_per_host": 8
        },
        "instance_groups": [instance_group_2]
    }
```

```
)
```

TensorFlow

```
from sagemaker.tensorflow import TensorFlow

estimator = TensorFlow(
    ...
    instance_groups=[instance_group_1, instance_group_2],
    distribution={
        "mpi": {
            "enabled": True, "processes_per_host": 8
        },
        "instance_groups": [instance_group_2]
    }
)
```

Dengan pustaka paralel SageMaker data

PyTorch

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(
    ...
    instance_groups=[instance_group_1, instance_group_2],
    distribution={
        "smdistributed": {
            "dataparallel": {
                "enabled": True
            }
        },
        "instance_groups": [instance_group_2]
    }
)
```

TensorFlow

```
from sagemaker.tensorflow import TensorFlow

estimator = TensorFlow(
    ...
```

```
    instance_groups=[instance_group_1, instance_group_2],
    distribution={
        "smdistributed": {
            "dataparallel": {
                "enabled": True
            }
        },
        "instance_groups": [instance_group_2]
    }
)
```

> **Note**
>
> Saat menggunakan pustaka paralel SageMaker data, pastikan grup instance terdiri dari jenis instance yang didukung oleh pustaka.

Untuk informasi selengkapnya tentang pustaka paralel SageMaker data, lihat Pelatihan Paralel SageMaker Data.

Dengan perpustakaan paralel SageMaker model

PyTorch

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(
    ...
    instance_groups=[instance_group_1, instance_group_2],
    distribution={
        "smdistributed": {
            "modelparallel": {
                "enabled":True,
                "parameters": {
                    ...   # SageMaker model parallel parameters
                }
            }
        },
        "instance_groups": [instance_group_2]
    }
)
```

TensorFlow

```
from sagemaker.tensorflow import TensorFlow

estimator = TensorFlow(
    ...
    instance_groups=[instance_group_1, instance_group_2],
    distribution={
        "smdistributed": {
            "modelparallel": {
                "enabled":True,
                "parameters": {
                    ...   # SageMaker model parallel parameters
                }
            }
        },
        "instance_groups": [instance_group_2]
    }
)
```

Untuk informasi selengkapnya tentang pustaka paralel SageMaker model, lihat Pelatihan Paralel SageMaker Model.

Ubah Skrip Pelatihan Anda untuk Menetapkan Grup Instans

Dengan konfigurasi klaster heterogen di bagian sebelumnya, Anda telah menyiapkan lingkungan SageMaker pelatihan dan instance untuk pekerjaan pelatihan Anda. Untuk menetapkan grup instans lebih lanjut ke tugas pelatihan dan pemrosesan data tertentu, langkah selanjutnya adalah memodifikasi skrip pelatihan Anda. Secara default, pekerjaan pelatihan hanya membuat replika skrip pelatihan untuk semua node terlepas dari ukuran instans, dan ini dapat menyebabkan hilangnya kinerja.

Misalnya, jika Anda mencampur instance CPU dan instans GPU dalam klaster heterogen sambil meneruskan skrip pelatihan jaringan saraf dalam ke `entry_point` argumen SageMaker estimator, skrip direplikasi ke setiap instance. `entry_point` Ini berarti bahwa, tanpa tugas tugas yang tepat, instance CPU juga menjalankan seluruh skrip dan memulai pekerjaan pelatihan yang dirancang untuk pelatihan terdistribusi pada instance GPU. Oleh karena itu, Anda harus membuat perubahan dalam fungsi pemrosesan tertentu yang ingin Anda bongkar dan jalankan pada instance CPU. Anda dapat menggunakan variabel SageMaker lingkungan untuk mengambil informasi dari cluster heterogen dan membiarkan proses tertentu berjalan sesuai.

Kueri informasi grup instance selama fase inisialisasi pekerjaan SageMaker pelatihan

Saat pekerjaan latihan Anda dimulai, skrip latihan Anda membaca informasi lingkungan SageMaker pelatihan yang mencakup konfigurasi klaster heterogen. Konfigurasi berisi informasi seperti grup instance saat ini, host saat ini di setiap grup, dan di grup mana host saat ini berada.

Anda dapat mengambil informasi grup instance dengan cara berikut.

(Disarankan) Membaca informasi grup instance dengan SageMaker toolkit pelatihan

Gunakan modul Python lingkungan yang disediakan oleh library toolkit SageMaker pelatihan. Pustaka toolkit sudah diinstal sebelumnya dalam wadah SageMaker kerangka kerja untuk TensorFlow danPyTorch, sehingga Anda tidak memerlukan langkah penginstalan tambahan saat menggunakan kontainer bawaan. Ini adalah cara yang disarankan untuk mengambil variabel SageMaker lingkungan dengan lebih sedikit perubahan kode dalam skrip latihan Anda.

```
from sagemaker_training import environment

env = environment.Environment()
```

Variabel lingkungan yang terkait dengan SageMaker pelatihan umum dan cluster heterogen:

- `env.is_hetero`- Mengembalikan hasil Boolean apakah cluster heterogen dikonfigurasi atau tidak.
- `env.current_host`- Mengembalikan host saat ini.
- `env.current_instance_type`- Mengembalikan jenis instance dari host saat ini.
- `env.current_instance_group`- Mengembalikan nama kelompok contoh saat ini.
- `env.current_instance_group_hosts`- Mengembalikan daftar host dalam kelompok contoh saat ini.
- `env.instance_groups`- Mengembalikan daftar nama grup instance yang digunakan untuk pelatihan.
- `env.instance_groups_dict`- Mengembalikan seluruh konfigurasi cluster heterogen dari pekerjaan pelatihan.
- `env.distribution_instance_groups`- Mengembalikan daftar kelompok contoh ditugaskan untuk `distribution` parameter kelas SageMaker estimator.
- `env.distribution_hosts`- Mengembalikan daftar host milik kelompok contoh ditugaskan untuk `distribution` parameter kelas SageMaker estimator.

Misalnya, perhatikan contoh berikut dari cluster heterogen yang terdiri dari dua kelompok instance.

```
from sagemaker.instance_group import InstanceGroup

instance_group_1 = InstanceGroup(
    "instance_group_1", "ml.c5.18xlarge", 1)
instance_group_2 = InstanceGroup(
    "instance_group_2", "ml.p3dn.24xlarge", 2)
```

Output `env.instance_groups_dict` dari contoh cluster heterogen harus mirip dengan berikut ini.

```
{
    "instance_group_1": {
        "hosts": [
            "algo-2"
        ],
        "instance_group_name": "instance_group_1",
        "instance_type": "ml.c5.18xlarge"
    },
    "instance_group_2": {
        "hosts": [
            "algo-3",
            "algo-1"
        ],
        "instance_group_name": "instance_group_2",
        "instance_type": "ml.p3dn.24xlarge"
    }
}
```

(Opsional) Membaca informasi grup instance dari file JSON konfigurasi sumber daya

Jika Anda lebih suka mengambil variabel lingkungan dalam format JSON, Anda dapat langsung menggunakan file JSON konfigurasi sumber daya. File JSON dalam instance SageMaker pelatihan terletak secara `/opt/ml/input/config/resourceconfig.json` default.

```
file_path = '/opt/ml/input/config/resourceconfig.json'
config = read_file_as_json(file_path)
print(json.dumps(config, indent=4, sort_keys=True))
```

Pertimbangan-pertimbangan

Pertimbangkan item berikut saat menggunakan fitur klaster heterogen.

- Semua grup instans berbagi gambar Docker dan skrip pelatihan yang sama. Oleh karena itu, skrip latihan Anda harus dimodifikasi untuk mendeteksi grup instans mana yang menjadi miliknya dan eksekusi fork yang sesuai.
- Fitur klaster heterogen tidak didukung dalam SageMaker mode lokal.
- Aliran CloudWatch log Amazon dari tugas pelatihan klaster heterogen tidak dikelompokkan berdasarkan grup instans. Anda perlu mencari tahu dari log mana node berada di grup mana.
- Fitur cluster heterogen tersedia melalui kelas SageMaker PyTorchdan TensorFlowkerangka estimator. Framework yang didukung adalah PyTorch v1.10 atau yang lebih baru dan TensorFlow v2.6 atau yang lebih baru. Untuk menemukan daftar lengkap container framework, versi framework, dan versi Python yang tersedia, lihat SageMakerFramework Container di GitHub repositori AWS Deep Learning Container.
- Strategi pelatihan terdistribusi hanya dapat diterapkan pada satu grup instans.

Contoh, Blog, dan Studi Kasus

Blog berikut membahas studi kasus tentang penggunaan pelatihan cluster SageMaker heterogen.

- Tingkatkan kinerja harga pelatihan model Anda menggunakan klaster SageMaker heterogen Amazon (27 Okt 2022)

Menggunakan Pelatihan Inkremental di Amazon SageMaker

Seiring waktu, Anda mungkin menemukan bahwa model menghasilkan kesimpulan yang tidak sebaik mereka di masa lalu. Dengan pelatihan tambahan, Anda dapat menggunakan artefak dari model yang ada dan menggunakan kumpulan data yang diperluas untuk melatih model baru. Pelatihan tambahan menghemat waktu dan sumber daya.

Gunakan pelatihan tambahan untuk:

- Latih model baru menggunakan kumpulan data yang diperluas yang berisi pola dasar yang tidak diperhitungkan dalam pelatihan sebelumnya dan yang menghasilkan kinerja model yang buruk.
- Gunakan artefak model atau sebagian artefak model dari model populer yang tersedia untuk umum dalam pekerjaan pelatihan. Anda tidak perlu melatih model baru dari awal.
- Lanjutkan pekerjaan pelatihan yang dihentikan.
- Latih beberapa varian model, baik dengan pengaturan hyperparameter yang berbeda atau menggunakan kumpulan data yang berbeda.

Untuk informasi lebih lanjut tentang pekerjaan pelatihan, lihatLatih Model dengan Amazon SageMaker.

Anda dapat berlatih secara bertahap menggunakan SageMaker konsol atau Amazon SageMaker Python SDK.

> **Important**
>
> Hanya tiga algoritma bawaan yang saat ini mendukung pelatihan tambahan:Deteksi,, Klasifikasi Gambar - MXNet dan. Algoritme segmentasi semantik

Topik

- Lakukan Pelatihan Inkremental (Konsol)
- Lakukan Pelatihan Inkremental (API)

Lakukan Pelatihan Inkremental (Konsol)

Untuk menyelesaikan prosedur ini, Anda memerlukan:

- URI bucket Amazon Simple Storage Service (Amazon S3) tempat Anda menyimpan data latihan.
- URI bucket S3 tempat Anda ingin menyimpan output pekerjaan.
- Jalur Amazon Elastic Container Registry tempat kode pelatihan disimpan. Untuk informasi selengkapnya, lihat Jalur Registri Docker dan Kode Contoh.
- URL bucket S3 tempat Anda menyimpan artefak model yang ingin Anda gunakan dalam pelatihan tambahan. Untuk menemukan URL artefak model, lihat halaman detail pekerjaan pelatihan yang digunakan untuk membuat model. Untuk menemukan halaman detail, di SageMaker konsol, pilih Inferensi, pilih Model, lalu pilih model.

Untuk memulai ulang pekerjaan pelatihan yang dihentikan, gunakan URL ke artefak model yang disimpan di halaman detail seperti yang Anda lakukan dengan model atau pekerjaan pelatihan yang telah selesai.

Untuk melakukan pelatihan tambahan (konsol)

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilih Pelatihan, lalu pilih Pekerjaan latihan.

3. Pilih Buat pekerjaan pelatihan.
4. Berikan nama untuk pekerjaan pelatihan. Nama harus unik dalam suatu AWS Wilayah di AWS akun. Nama pekerjaan pelatihan harus memiliki 1 hingga 63 karakter. Karakter yang valid: a-z, A-Z, 0-9, dan.: + = @ _% - (tanda hubung).
5. Pilih algoritma yang ingin Anda gunakan. Untuk informasi tentang algoritma, lihat. Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih
6. (Opsional) Untuk konfigurasi Sumber Daya, tinggalkan nilai default atau tingkatkan konsumsi sumber daya untuk mengurangi waktu perhitungan.

    a. (Opsional) Untuk jenis Instans, pilih jenis instans komputasi yang ingin Anda gunakan. Dalam kebanyakan kasus, ml.m4.xlarge sudah cukup.
    b. Untuk hitungan Instance, gunakan default, 1.
    c. (Opsional) Untuk Volume tambahan per instans (GB), pilih ukuran volume penyimpanan yang ingin Anda sediakan. Dalam kebanyakan kasus, Anda dapat menggunakan default, 1. Jika Anda menggunakan dataset besar, gunakan ukuran yang lebih besar.
7. Berikan informasi tentang data input untuk kumpulan data pelatihan.

    a. Untuk nama Channel, biarkan default (**`train`**) atau masukkan nama yang lebih berarti untuk set data pelatihan, seperti. **`expanded-training-dataset`**
    b. Untuk InputMode, pilih File. Untuk pelatihan tambahan, Anda perlu menggunakan mode input file.
    c. Untuk tipe distribusi data S3, pilih FullyReplicated. Hal ini menyebabkan setiap instans komputasi ML menggunakan replika penuh dari kumpulan data yang diperluas saat berlatih secara bertahap.
    d. Jika kumpulan data yang diperluas tidak terkompresi, atur tipe Kompresi ke Tidak Ada. Jika kumpulan data yang diperluas dikompresi menggunakan Gzip, atur ke Gzip.
    e. (Opsional) Jika Anda menggunakan mode input File, biarkan Jenis konten kosong. Untuk mode input Pipa, tentukan tipe MIME yang sesuai. Jenis konten adalah multiguna internet mail extension (MIME) jenis data.
    f. Untuk Record wrapper, jika dataset disimpan dalam format RecordO, pilih Recordio. Jika kumpulan data Anda tidak disimpan sebagai file berformat RecordiO, pilih Tidak ada.
    g. Untuk tipe data S3, jika dataset disimpan sebagai satu file, pilih S3Prefix. Jika dataset disimpan sebagai beberapa file dalam folder, pilih Manifest.
    h. Untuk lokasi S3, berikan URL ke jalur tempat Anda menyimpan kumpulan data yang diperluas.

i. PilihSelesai.

8. Untuk menggunakan artefak model dalam pekerjaan pelatihan, Anda perlu menambahkan saluran baru dan memberikan informasi yang diperlukan tentang artefak model.

   a. Untuk Konfigurasi data input, pilih Tambahkan saluran.

   b. Untuk nama Channel, masukkan **`model`** untuk mengidentifikasi saluran ini sebagai sumber artefak model.

   c. Untuk InputMode, pilih File. Model artefak disimpan sebagai file.

   d. Untuk tipe distribusi data S3, pilih FullyReplicated. Hal ini menunjukkan bahwa setiap instans komputasi ML harus menggunakan semua artefak model untuk pelatihan.

   e. Untuk tipe Compression, pilih None karena kita menggunakan model untuk channel.

   f. Biarkan jenis Konten kosong. Jenis konten adalah multiguna internet mail extension (MIME) jenis data. Untuk artefak model, kami membiarkannya kosong.

   g. Atur Record wrapper ke None karena artefak model tidak disimpan dalam format RecordiO.

   h. Untuk tipe data S3, jika Anda menggunakan algoritma bawaan atau algoritma yang menyimpan model sebagai satu file, pilih S3Prefix. Jika Anda menggunakan algoritma yang menyimpan model sebagai beberapa file, pilih Manifest.

   i. Untuk lokasi S3, berikan URL ke jalur tempat Anda menyimpan artefak model. Biasanya, model disimpan dengan nama`model.tar.gz`. Untuk menemukan URL artefak model, di panel navigasi, pilih Inferensi, lalu pilih Model. Dari daftar model, pilih model untuk menampilkan halaman detailnya. URL untuk artefak model tercantum di bawah wadah Primer.

   j. PilihSelesai.

9. Untuk konfigurasi data Output, berikan informasi berikut:

   a. Untuk lokasi S3, ketik path ke bucket S3 tempat Anda ingin menyimpan data keluaran.

   b. (Opsional) Untuk kunci Enkripsi, Anda dapat menambahkan kunci enkripsi AWS Key Management Service (AWS KMS) untuk mengenkripsi data keluaran saat istirahat. Berikan ID kunci atau Nomor Sumber Daya Amazon (ARN). Untuk informasi selengkapnya, lihat Kunci Enkripsi yang Dikelola KMS.

10. (Opsional) Untuk Tag, tambahkan satu atau lebih tag ke pekerjaan pelatihan. Tag adalah metadata yang dapat Anda tentukan dan tetapkan ke AWS sumber daya. Dalam hal ini, Anda dapat menggunakan tag untuk membantu Anda mengelola pekerjaan pelatihan Anda. Tag terdiri dari kunci dan nilai, yang Anda tentukan. Misalnya, Anda mungkin ingin membuat tag dengan

`Project` sebagai kunci dan nilai yang mengacu pada proyek yang terkait dengan pekerjaan pelatihan, seperti**`Home value forecasts`**.

11. Pilih Buat pekerjaan pelatihan. SageMakermenciptakan dan menjalankan pekerjaan pelatihan.

Setelah pekerjaan pelatihan selesai, artefak model yang baru dilatih disimpan di bawah jalur keluaran S3 yang Anda berikan di bidang Konfigurasi data keluaran. Untuk menerapkan model untuk mendapatkan prediksi, lihat. Langkah 5: Menyebarkan Model ke Amazon EC2

Lakukan Pelatihan Inkremental (API)

Contoh ini menunjukkan cara menggunakan SageMaker API untuk melatih model menggunakan algoritma klasifikasi SageMaker gambar dan Dataset Gambar Caltech 256, lalu melatih model baru menggunakan yang pertama. Ini menggunakan Amazon S3 untuk sumber input dan output. Silakan lihat contoh notebook pelatihan tambahan untuk detail lebih lanjut tentang penggunaan pelatihan tambahan.

> **Note**
>
> Dalam contoh ini kami menggunakan kumpulan data asli dalam pelatihan tambahan, namun Anda dapat menggunakan kumpulan data yang berbeda, seperti kumpulan data yang berisi sampel yang baru ditambahkan. Unggah kumpulan data baru ke S3 dan buat penyesuaian pada `data_channels` variabel yang digunakan untuk melatih model baru.

Dapatkan peran AWS Identity and Access Management (IAM) yang memberikan izin yang diperlukan dan menginisialisasi variabel lingkungan:

```
import sagemaker
from sagemaker import get_execution_role

role = get_execution_role()
print(role)

sess = sagemaker.Session()

bucket=sess.default_bucket()
print(bucket)
prefix = 'ic-incr-training'
```

Dapatkan gambar pelatihan untuk algoritma klasifikasi gambar:

```
from sagemaker.amazon.amazon_estimator import get_image_uri

training_image = get_image_uri(sess.boto_region_name, 'image-classification',
 repo_version="latest")
#Display the training image
print (training_image)
```

Unduh set data pelatihan dan validasi, lalu unggah ke Amazon Simple Storage Service (Amazon S3):

```
import os
import urllib.request
import boto3

# Define a download function
def download(url):
    filename = url.split("/")[-1]
    if not os.path.exists(filename):
        urllib.request.urlretrieve(url, filename)

# Download the caltech-256 training and validation datasets
download('http://data.mxnet.io/data/caltech-256/caltech-256-60-train.rec')
download('http://data.mxnet.io/data/caltech-256/caltech-256-60-val.rec')

# Create four channels: train, validation, train_lst, and validation_lst
s3train = 's3://{}/{}/train/'.format(bucket, prefix)
s3validation = 's3://{}/{}/validation/'.format(bucket, prefix)

# Upload the first files to the train and validation channels
!aws s3 cp caltech-256-60-train.rec $s3train --quiet
!aws s3 cp caltech-256-60-val.rec $s3validation --quiet
```

Tentukan hyperparameter pelatihan:

```
# Define hyperparameters for the estimator
hyperparams = { "num_layers": "18",
                "resize": "32",
                "num_training_samples": "50000",
                "num_classes": "10",
                "image_shape": "3,28,28",
                "mini_batch_size": "128",
```

```
                "epochs": "3",
                "learning_rate": "0.1",
                "lr_scheduler_step": "2,3",
                "lr_scheduler_factor": "0.1",
                "augmentation_type": "crop_color",
                "optimizer": "sgd",
                "momentum": "0.9",
                "weight_decay": "0.0001",
                "beta_1": "0.9",
                "beta_2": "0.999",
                "gamma": "0.9",
                "eps": "1e-8",
                "top_k": "5",
                "checkpoint_frequency": "1",
                "use_pretrained_model": "0",
                "model_prefix": "" }
```

Buat objek estimator dan latih model pertama menggunakan set data pelatihan dan validasi:

```
# Fit the base estimator
s3_output_location = 's3://{}/{}/output'.format(bucket, prefix)
ic = sagemaker.estimator.Estimator(training_image,
                                   role,
                                   instance_count=1,
                                   instance_type='ml.p2.xlarge',
                                   volume_size=50,
                                   max_run=360000,
                                   input_mode='File',
                                   output_path=s3_output_location,
                                   sagemaker_session=sess,
                                   hyperparameters=hyperparams)

train_data = sagemaker.inputs.TrainingInput(s3train, distribution='FullyReplicated',
                                        content_type='application/x-recordio',
 s3_data_type='S3Prefix')
validation_data = sagemaker.inputs.TrainingInput(s3validation,
 distribution='FullyReplicated',
                                             content_type='application/x-recordio',
 s3_data_type='S3Prefix')

data_channels = {'train': train_data, 'validation': validation_data}

ic.fit(inputs=data_channels, logs=True)
```

Untuk menggunakan model untuk melatih model lain secara bertahap, buat objek estimator baru dan gunakan artefak model (`ic.model_data`, dalam contoh ini) untuk argumen masukan: `model_uri`

```
# Given the base estimator, create a new one for incremental training
incr_ic = sagemaker.estimator.Estimator(training_image,
                                        role,
                                        instance_count=1,
                                        instance_type='ml.p2.xlarge',
                                        volume_size=50,
                                        max_run=360000,
                                        input_mode='File',
                                        output_path=s3_output_location,
                                        sagemaker_session=sess,
                                        hyperparameters=hyperparams,
                                        model_uri=ic.model_data) # This parameter will
 ingest the previous job's model as a new channel
incr_ic.fit(inputs=data_channels, logs=True)
```

Setelah pekerjaan pelatihan selesai, artefak model yang baru dilatih disimpan di bawah `S3 output path` yang Anda berikan. `Output_path` Untuk menerapkan model untuk mendapatkan prediksi, lihat. Langkah 5: Menyebarkan Model ke Amazon EC2

Gunakan Pelatihan Spot Terkelola di Amazon SageMaker

Amazon SageMaker memudahkan untuk melatih model pembelajaran mesin menggunakan instans Amazon EC2 Spot terkelola. Pelatihan spot terkelola dapat mengoptimalkan biaya model pelatihan hingga 90% dibandingkan instans sesuai permintaan. SageMaker mengelola interupsi Spot atas nama Anda.

Pelatihan Spot Terkelola menggunakan instans Spot Amazon EC2 untuk menjalankan pekerjaan pelatihan, bukan instans sesuai permintaan. Anda dapat menentukan lowongan pelatihan mana yang menggunakan instans spot dan kondisi penghentian yang menentukan berapa lama SageMaker menunggu pekerjaan dijalankan menggunakan instans Amazon EC2 Spot. Metrik dan log yang dihasilkan selama pelatihan berjalan tersedia di CloudWatch.

Penyetelan model SageMaker otomatis Amazon, juga dikenal sebagai tuning hyperparameter, dapat menggunakan pelatihan spot terkelola. Untuk informasi lebih lanjut tentang penyetelan model otomatis, lihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Instans spot dapat terganggu, menyebabkan pekerjaan membutuhkan waktu lebih lama untuk memulai atau menyelesaikan. Anda dapat mengonfigurasi pekerjaan pelatihan spot terkelola untuk menggunakan pos pemeriksaan. SageMaker menyalin data pos pemeriksaan dari jalur lokal ke Amazon S3. Saat pekerjaan dimulai ulang, SageMaker salin data dari Amazon S3 kembali ke jalur lokal. Pekerjaan pelatihan kemudian dapat dilanjutkan dari pos pemeriksaan terakhir alih-alih memulai kembali. Untuk informasi selengkapnya tentang pos pemeriksaan, lihat. Menggunakan Pos Pemeriksaan di Amazon SageMaker

> **Note**
>
> Kecuali pekerjaan pelatihan Anda akan selesai dengan cepat, kami sarankan Anda menggunakan pos pemeriksaan dengan pelatihan spot terkelola. SageMaker algoritma bawaan dan algoritme pasar yang tidak memiliki pos pemeriksaan saat ini dibatasi hingga 3600 detik (60 menit). `MaxWaitTimeInSeconds`

Topik

Menggunakan Pelatihan Spot Terkelola

Untuk menggunakan pelatihan spot terkelola, buat pekerjaan pelatihan. Setel `EnableManagedSpotTraining` ke `True` dan tentukan`MaxWaitTimeInSeconds`. `MaxWaitTimeInSeconds`harus lebih besar dari`MaxRuntimeInSeconds`. Untuk informasi selengkapnya tentang membuat pekerjaan pelatihan, lihat DescribeTrainingJob.

Anda dapat menghitung penghematan dari menggunakan pelatihan spot terkelola menggunakan rumus`(1 - (BillableTimeInSeconds / TrainingTimeInSeconds)) * 100`. Misalnya, jika `BillableTimeInSeconds` 100 dan `TrainingTimeInSeconds` 500, ini berarti bahwa pekerjaan pelatihan Anda berjalan selama 500 detik, tetapi Anda ditagih hanya 100 detik. Tabungan Anda adalah (1 - (100/500)) * 100 = 80%.

Untuk mempelajari cara menjalankan pekerjaan pelatihan di instans SageMaker spot Amazon dan cara kerja pelatihan spot terkelola serta mengurangi waktu yang dapat ditagih, lihat contoh buku catatan berikut:

- Pelatihan Spot Terkelola dengan TensorFlow

- Pelatihan Spot Terkelola dengan PyTorch
- Pelatihan Spot Terkelola dengan XGBoost
- Pelatihan Spot Terkelola dengan MxNet
- GitHub Repositori Contoh Pelatihan Spot SageMaker Terkelola Amazon

Siklus Hidup Pelatihan Spot Terkelola

Anda dapat memantau pekerjaan pelatihan menggunakan `TrainingJobStatus` dan `SecondaryStatus` dikembalikan oleh DescribeTrainingJob. Daftar di bawah ini menunjukkan bagaimana `TrainingJobStatus` dan `SecondaryStatus` nilai berubah tergantung pada skenario pelatihan:

- Instans spot diperoleh tanpa gangguan selama pelatihan
  1. `InProgress`: `Starting` → `Downloading` → `Training` → `Uploading`
- Instance spot terputus sekali. Kemudian, cukup banyak contoh spot diperoleh untuk menyelesaikan pekerjaan pelatihan.
  1. `InProgress`: `Starting` → `Downloading` → `Training` → `Interrupted` → `Starting` → `Downloading` → `Training` → `Uploading`
- Instance spot terputus dua kali dan **`MaxWaitTimeInSeconds`** terlampaui.
  1. `InProgress`: `Starting` → `Downloading` → `Training` → `Interrupted` → `Starting` → `Downloading` → `Training` → `Interrupted` → `Downloading` → `Training`
  2. `Stopping`: `Stopping`
  3. `Stopped`: `MaxWaitTimeExceeded`
- Instans spot tidak pernah diluncurkan.
  1. `InProgress`: `Starting`
  2. `Stopping`: `Stopping`
  3. `Stopped`: `MaxWaitTimeExceeded`

Melatih Menggunakan Kolam Hangat yang SageMaker Dikelola

SageMakerkolam hangat yang dikelola memungkinkan Anda mempertahankan dan menggunakan kembali infrastruktur yang disediakan setelah menyelesaikan pekerjaan pelatihan untuk mengurangi latensi beban kerja berulang, seperti eksperimen berulang atau menjalankan banyak pekerjaan

secara berurutan. Pekerjaan pelatihan selanjutnya yang sesuai dengan parameter tertentu berjalan pada infrastruktur kolam hangat yang dipertahankan, yang mempercepat waktu mulai dengan mengurangi waktu yang dihabiskan untuk menyediakan sumber daya.

> **Important**
>
> SageMakerkolam hangat yang dikelola adalah sumber daya yang dapat ditagih. Untuk informasi selengkapnya, lihat Penagihan.

Topik

Cara kerjanya

Untuk menggunakan kolam hangat yang SageMaker dikelola dan mengurangi latensi antara pekerjaan pelatihan berturut-turut yang serupa, buat pekerjaan pelatihan yang `KeepAlivePeriodInSeconds` menentukan nilainya. `ResourceConfig` Nilai ini mewakili durasi waktu dalam hitungan detik untuk mempertahankan sumber daya yang dikonfigurasi di kolam hangat untuk pekerjaan pelatihan berikutnya. Jika Anda perlu menjalankan beberapa pekerjaan pelatihan menggunakan konfigurasi serupa, Anda dapat mengurangi latensi dan waktu yang dapat ditagih dengan menggunakan direktori cache persisten khusus untuk menyimpan dan menggunakan kembali informasi Anda dalam pekerjaan yang berbeda.

Topik

Siklus hidup kolam renang hangat

1. Buat pekerjaan pelatihan awal dengan `KeepAlivePeriodInSeconds` nilai lebih besar dari 0. Ketika Anda menjalankan pekerjaan pelatihan pertama ini, ini "cold-start" cluster dengan waktu startup yang khas.
2. Ketika pekerjaan pelatihan pertama selesai, sumber daya yang disediakan tetap hidup di kolam hangat untuk periode yang ditentukan dalam nilai. `KeepAlivePeriodInSeconds` Selama cluster sehat dan kolam hangat berada dalam yang ditentukan`KeepAlivePeriodInSeconds`, maka status kolam hangat adalah`Available`.
3. Kolam hangat tetap `Available` sampai mengidentifikasi pekerjaan pelatihan yang cocok untuk digunakan kembali atau melebihi yang ditentukan `KeepAlivePeriodInSeconds` dan dihentikan. Panjang maksimum waktu yang diizinkan untuk `KeepAlivePeriodInSeconds` adalah 3600 detik (60 menit). Jika status kolam hangat`Terminated`, maka ini adalah akhir dari siklus hidup kolam hangat.
4. Jika kolam hangat mengidentifikasi pekerjaan pelatihan kedua dengan spesifikasi yang cocok seperti jumlah instans atau jenis instans, maka kolam hangat bergerak dari pekerjaan pelatihan pertama ke pekerjaan pelatihan kedua untuk digunakan kembali. Status pekerjaan pelatihan pertama kolam hangat menjadi`Reused`. Ini adalah akhir dari siklus hidup kolam hangat untuk pekerjaan pelatihan pertama.
5. Status pekerjaan pelatihan kedua yang menggunakan kembali kolam hangat menjadi`InUse`. Setelah pekerjaan pelatihan kedua selesai, kolam hangat adalah `Available` untuk `KeepAlivePeriodInSeconds` durasi yang ditentukan dalam pekerjaan pelatihan kedua. Kolam yang hangat dapat terus berpindah ke pekerjaan pelatihan pencocokan berikutnya selama maksimal 28 hari.
6. Jika kolam hangat tidak lagi tersedia untuk digunakan kembali, status kolam hangat adalah`Terminated`. Kolam hangat tidak lagi tersedia jika dihentikan oleh pengguna, untuk pembaruan patch, atau untuk melebihi yang ditentukan. `KeepAlivePeriodInSeconds`

Untuk informasi selengkapnya tentang opsi status kolam hangat, lihat WarmPoolStatusdi Referensi SageMaker API Amazon.

Penciptaan kolam hangat

Jika pekerjaan pelatihan awal berhasil diselesaikan dan memiliki `KeepAlivePeriodInSeconds` nilai lebih besar dari 0, ini menciptakan kolam hangat. Jika Anda menghentikan pekerjaan pelatihan setelah cluster sudah diluncurkan, kolam hangat masih dipertahankan. Jika pekerjaan pelatihan

gagal karena algoritma atau kesalahan klien, kolam hangat masih dipertahankan. Jika pekerjaan pelatihan gagal karena alasan lain yang mungkin membahayakan kesehatan cluster, maka kolam hangat tidak dibuat.

Untuk memverifikasi keberhasilan pembuatan kolam hangat, periksa status kolam hangat dari pekerjaan pelatihan Anda. Jika kolam hangat berhasil ketentuan, status kolam hangat adalah`Available`. Jika kolam hangat gagal untuk penyediaan, status kolam hangat adalah`Terminated`.

Lowongan kerja Matching training

Agar kolam hangat bertahan, ia harus menemukan pekerjaan pelatihan yang cocok dalam waktu yang ditentukan dalam `KeepAlivePeriodInSeconds` nilai. Pekerjaan pelatihan berikutnya adalah kecocokan jika nilai berikut identik:

- `RoleArn`
- `ResourceConfig`nilai:
  - `InstanceCount`
  - `InstanceType`
  - `VolumeKmsKeyId`
  - `VolumeSizeInGB`
- `VpcConfig`nilai:
  - `SecurityGroupIds`
  - `Subnets`
- `EnableInterContainerTrafficEncryption`
- `EnableNetworkIsolation`

Semua nilai ini harus sama untuk kolam hangat untuk pindah ke pekerjaan pelatihan berikutnya untuk digunakan kembali.

Durasi maksimum kolam hangat

Maksimum `KeepAlivePeriodInSeconds` untuk satu pekerjaan pelatihan adalah 3600 detik (60 menit) dan durasi maksimum cluster kolam hangat dapat terus menjalankan pekerjaan pelatihan berturut-turut adalah 28 hari.

Setiap pekerjaan pelatihan berikutnya juga harus menentukan `KeepAlivePeriodInSeconds` nilai. Ketika kolam hangat bergerak ke pekerjaan pelatihan berikutnya, itu mewarisi `KeepAlivePeriodInSeconds` nilai baru yang ditentukan dalam pekerjaan pelatihan itu`ResourceConfig`. Dengan cara ini, Anda dapat menyimpan kolam hangat bergerak dari pekerjaan pelatihan ke pekerjaan pelatihan selama maksimal 28 hari.

Jika tidak `KeepAlivePeriodInSeconds` ada yang ditentukan, maka kolam hangat berputar ke bawah setelah pekerjaan pelatihan selesai.

Menggunakan cache persisten

Saat Anda membuat kolam hangat, SageMaker pasang direktori khusus pada volume yang akan bertahan sepanjang siklus hidup kolam hangat. Direktori ini juga dapat digunakan untuk menyimpan informasi yang ingin Anda gunakan kembali di pekerjaan lain.

Menggunakan cache persisten dapat mengurangi latensi dan waktu yang dapat ditagih menggunakan kolam hangat saja untuk pekerjaan yang memerlukan hal berikut:

- beberapa interaksi dengan konfigurasi serupa
- pekerjaan inkremental training
- optimasi hiperparameter

Misalnya, Anda dapat menghindari mengunduh dependensi Python yang sama pada proses berulang dengan menyiapkan direktori cache pip di dalam direktori cache persisten. Anda bertanggung jawab penuh untuk mengelola isi direktori ini. Berikut ini adalah contoh jenis informasi yang dapat Anda masukkan ke dalam cache persisten untuk membantu mengurangi latensi dan waktu yang dapat ditagih.

- Dependensi dikelola oleh pip.
- Dependensi dikelola oleh conda.
- Informasi pos pemeriksaan.
- Informasi tambahan apa pun yang dihasilkan selama pelatihan.

Lokasi cache persisten adalah`/opt/ml/sagemaker/warmpoolcache`. Variabel lingkungan `SAGEMAKER_MANAGED_WARMPOOL_CACHE_DIRECTORY` menunjuk ke lokasi direktori cache persisten.

Contoh kode berikut menunjukkan cara mengatur kolam hangat dan menggunakan cache persisten untuk menyimpan dependensi pip Anda untuk digunakan dalam pekerjaan berikutnya. Pekerjaan berikutnya harus berjalan dalam jangka waktu yang diberikan oleh parameter`keep_alive_period_in_seconds`.

```
import sagemakerfrom sagemaker import get_execution_rolefrom sagemaker.tensorflow
 import TensorFlow
# Creates a SageMaker session and gets execution role
session = sagemaker.Session()
role = get_execution_role()
# Creates an example estimator
estimator = TensorFlow(
    ...
    entry_point='my-training-script.py',
    source_dir='code',
    role=role,
    model_dir='model_dir',
    framework_version='2.2',
    py_version='py37',
    job_name='my-training-job-1',
    instance_type='ml.g4dn.xlarge',
    instance_count=1,
    volume_size=250,
    hyperparameters={
"batch-size": 512,
        "epochs": 1,
        "learning-rate": 1e-3,
        "beta_1": 0.9,
        "beta_2": 0.999,
    },
    keep_alive_period_in_seconds=1800,
    environment={"PIP_CACHE_DIR": "/opt/ml/sagemaker/warmpoolcache/pip"}
)
```

Dalam contoh kode sebelumnya, menggunakan parameter lingkungan ekspor variabel lingkungan `PIP_CACHE_DIRECTORY` untuk menunjuk ke direktori`/opt/ml/sagemaker/warmpoolcache/pip`. Mengekspor variabel lingkungan ini akan berubah di mana pip menyimpan cache ke lokasi baru. Direktori apa pun, termasuk direktori bersarang, yang Anda buat di dalam direktori cache persisten akan tersedia untuk digunakan kembali selama pelatihan berikutnya. Dalam contoh kode sebelumnya, direktori yang `pip` disebut diubah menjadi lokasi default untuk cache setiap dependensi diinstal menggunakan pip.

Lokasi cache persisten juga dapat diakses dari dalam skrip pelatihan Python Anda menggunakan variabel lingkungan seperti yang ditunjukkan dalam contoh kode berikut.

```
import os
import shutil
if __name__ == '__main__':
    PERSISTED_DIR = os.environ["SAGEMAKER_MANAGED_WARMPOOL_CACHE_DIRECTORY"]

    # create a file to be persisted
    open(os.path.join(PERSISTED_DIR, "test.txt"), 'a').close()
    # create a directory to be persisted
    os.mkdir(os.path.join(PERSISTED_DIR, "test_dir"))

    # Move a file to be persisted
    shutil.move("path/of/your/file.txt", PERSISTED_DIR)
```

Penagihan

SageMakerkolam hangat yang dikelola adalah sumber daya yang dapat ditagih. Ambil status kolam hangat untuk pekerjaan pelatihan Anda untuk memeriksa waktu yang dapat ditagih untuk kolam hangat Anda. Anda dapat memeriksa status kolam hangat baik melalui Menggunakan SageMaker konsol Amazon atau langsung melalui perintah DescribeTrainingJobAPI. Untuk informasi lebih lanjut, lihat WarmPoolStatus dalam Amazon SageMaker Referensi API.

> **Note**
>
> Setelah waktu yang ditentukan oleh parameter `KeepAlivePeriodInSeconds` berakhir, kolam hangat dan cache persisten akan mati, dan isinya akan dihapus.

Batas sumber daya kolam yang hangat

Untuk memulai, Anda harus terlebih dahulu meminta peningkatan batas layanan untuk kolam hangat yang SageMaker dikelola. Batas sumber daya default untuk kolam hangat adalah 0.

Jika pekerjaan pelatihan dibuat dengan `KeepAlivePeriodInSeconds` ditentukan, tetapi Anda tidak meminta kenaikan batas kolam yang hangat, maka kolam hangat tidak dipertahankan setelah selesainya pekerjaan pelatihan. Kolam hangat hanya dibuat jika batas kolam hangat Anda memiliki sumber daya yang cukup. Setelah kolam hangat dibuat, sumber daya dilepaskan ketika mereka

pindah ke pekerjaan pelatihan yang cocok atau jika `KeepAlivePeriodInSeconds` kedaluwarsa (jika status kolam hangat adalah `Reused` atau`Terminated`).

Minta kenaikan kuota kolam hangat

Minta peningkatan kuota pool hangat menggunakan konsol Kuota AWS Layanan.

> **Note**
>
> Semua penggunaan instans kolam hangat dihitung terhadap batas sumber daya SageMaker latihan Anda. Meningkatkan batas sumber daya kolam hangat Anda tidak meningkatkan batas instans Anda, tetapi mengalokasikan subset dari batas sumber daya Anda untuk latihan kolam hangat.

1. Buka konsol Kuota AWS Layanan.
2. Di panel navigasi sebelah kiri, pilih AWS layanan.
3. Cari dan pilih Amazon SageMaker.
4. Cari kata kunci **`warm pool`** untuk melihat semua kuota layanan kolam hangat yang tersedia.
5. Temukan jenis instans yang ingin Anda tingkatkan kuota kolam hangat Anda, pilih kuota layanan kolam hangat untuk jenis instans tersebut, dan pilih Request quota increase.
6. Masukkan nomor limit instans yang Anda minta di bawah Ubah nilai kuota. Nilai baru harus lebih besar dari nilai kuota Terapan saat ini.
7. Pilih Permintaan.

Ada batasan jumlah instans yang dapat Anda pertahankan untuk setiap akun, yang ditentukan oleh jenis instans. Anda dapat memeriksa batas sumber daya Anda di konsol Kuota AWS Layanan atau langsung menggunakan perintah list-service-quotasAWSCLI. Untuk informasi lebih lanjut tentang Kuota AWS Layanan, lihat Meminta peningkatan kuota dalam Panduan Pengguna Kuota Layanan.

Anda juga dapat menggunakan AWSSupport Center untuk meminta kenaikan kuota kolam hangat. Untuk daftar jenis instans yang tersedia menurut Wilayah, lihat SageMakerHarga Amazon dan pilih Pelatihan di tabel Harga Sesuai Permintaan.

Cara menggunakan kolam hangat yang SageMaker dikelola

Anda dapat menggunakan kumpulan hangat SageMaker terkelola melalui SageMaker Python SDK, SageMaker konsol Amazon, atau melalui API tingkat rendah. Administrator secara opsional dapat menggunakan kunci `sagemaker:KeepAlivePeriod` kondisi untuk membatasi `KeepAlivePeriodInSeconds` batasan pengguna atau grup tertentu.

Topik

Menggunakan SageMaker Python SDK

Buat, perbarui, atau hentikan kumpulan hangat menggunakan SageMaker Python SDK.

> **Note**
>
> Fitur ini tersedia di SageMaker Python SDK v2.110.0 dan versi lebih baru.

Topik

Buat kolam yang hangat

Untuk membuat kolam hangat, gunakan SageMaker Python SDK untuk membuat estimator dengan `keep_alive_period_in_seconds` nilai lebih besar dari 0 dan panggil. `fit()` Saat pekerjaan pelatihan selesai, kolam hangat dipertahankan. Untuk informasi selengkapnya tentang skrip pelatihan dan estimator, lihat Melatih Model dengan SageMaker Python SDK. Jika skrip Anda tidak membuat kolam hangat, lihat penjelasan Penciptaan kolam hangat yang mungkin.

```
import sagemaker
from sagemaker import get_execution_role
```

```
from sagemaker.tensorflow import TensorFlow

# Creates a SageMaker session and gets execution role
session = sagemaker.Session()
role = get_execution_role()

# Creates an example estimator
estimator = TensorFlow(
    ...
    entry_point='my-training-script.py',
    source_dir='code',
    role=role,
    model_dir='model_dir',
    framework_version='2.2',
    py_version='py37',
    job_name='my-training-job-1',
    instance_type='ml.g4dn.xlarge',
    instance_count=1,
    volume_size=250,
    hyperparameters={
        "batch-size": 512,
        "epochs": 1,
        "learning-rate": 1e-3,
        "beta_1": 0.9,
        "beta_2": 0.999,
    },
    keep_alive_period_in_seconds=1800,
)

# Starts a SageMaker training job and waits until completion
estimator.fit('s3://my_bucket/my_training_data/')
```

Selanjutnya, buat pekerjaan pelatihan pencocokan kedua. Dalam contoh ini, kita membuat`my-training-job-2`, yang memiliki semua atribut yang diperlukan untuk mencocokkan dengan`my-training-job-1`, tetapi memiliki hyperparameter yang berbeda untuk eksperimen. Pekerjaan pelatihan kedua menggunakan kembali kolam hangat dan memulai lebih cepat daripada pekerjaan pelatihan pertama. Contoh kode berikut menggunakan estimator Tensorflow. Fitur kolam hangat dapat digunakan dengan algoritme pelatihan apa pun yang berjalan di AmazonSageMaker. Untuk informasi lebih lanjut tentang atribut mana yang harus dicocokkan, lihat[Lowongan kerja Matching training](#).

```
# Creates an example estimator
```

```
estimator = TensorFlow(
    ...
    entry_point='my-training-script.py',
    source_dir='code',
    role=role,
    model_dir='model_dir',
    framework_version='py37',
    py_version='pyxy',
    job_name='my-training-job-2',
    instance_type='ml.g4dn.xlarge',
    instance_count=1,
    volume_size=250,
    hyperparameters={
        "batch-size": 512,
        "epochs": 2,
        "learning-rate": 1e-3,
        "beta_1": 0.9,
        "beta_2": 0.999,
    },
    keep_alive_period_in_seconds=1800,
)

# Starts a SageMaker training job and waits until completion
estimator.fit('s3://my_bucket/my_training_data/')
```

Periksa status kolam hangat dari kedua pekerjaan pelatihan untuk memastikan bahwa kolam hangat adalah `Reused` untuk `my-training-job-1` dan `InUse` untuk`my-training-job-2`.

> **Note**
>
> Nama pekerjaan pelatihan memiliki akhiran tanggal/waktu. Contoh nama pekerjaan pelatihan `my-training-job-1` dan `my-training-job-2` harus diganti dengan nama pekerjaan pelatihan yang sebenarnya. Anda dapat menggunakan `estimator.latest_training_job.job_name` perintah untuk mengambil nama pekerjaan pelatihan yang sebenarnya.

```
session.describe_training_job('my-training-job-1')
session.describe_training_job('my-training-job-2')
```

Hasil `describe_training_job` memberikan semua rincian tentang pekerjaan pelatihan yang diberikan. Temukan `WarmPoolStatus` atribut untuk memeriksa informasi tentang kolam hangat pekerjaan pelatihan. Output Anda akan terlihat mirip dengan contoh berikut:

```
# Warm pool status for training-job-1
...
'WarmPoolStatus': {'Status': 'Reused',
  'ResourceRetainedBillableTimeInSeconds': 1000,
  'ReusedByName': my-training-job-2}
...

# Warm pool status for training-job-2
...
'WarmPoolStatus': {'Status': 'InUse'}
...
```

Perbarui kolam hangat

Ketika pekerjaan pelatihan selesai dan status kolam hangat`Available`, maka Anda dapat memperbarui `KeepAlivePeriodInSeconds` nilainya.

```
session.update_training_job(job_name,
 resource_config={"KeepAlivePeriodInSeconds":3600})
```

Hentikan kolam hangat

Untuk mengakhiri kolam hangat secara manual, atur `KeepAlivePeriodInSeconds` nilainya ke 0.

```
session.update_training_job(job_name, resource_config={"KeepAlivePeriodInSeconds":0})
```

Kolam hangat secara otomatis berakhir ketika melebihi `KeepAlivePeriodInSeconds` nilai yang ditentukan atau jika ada pembaruan patch untuk klaster.

Menggunakan SageMaker konsol Amazon

Melalui konsol, Anda dapat membuat kolam yang hangat, melepaskan kolam hangat, atau memeriksa status kolam hangat dan waktu yang dapat ditagih dari pekerjaan pelatihan tertentu. Anda juga dapat melihat pekerjaan pelatihan yang cocok digunakan kembali kolam hangat.

1. Buka SageMakerkonsol Amazon dan pilih Pekerjaan pelatihan dari panel navigasi. Jika berlaku, status kolam hangat dari setiap pekerjaan pelatihan terlihat di kolom status kolam hangat dan waktu yang tersisa untuk kolam hangat aktif terlihat di kolom Waktu kiri.
2. Untuk membuat pekerjaan pelatihan yang menggunakan kolam hangat dari konsol, pilih Buat pekerjaan pelatihan. Kemudian, pastikan untuk menentukan nilai untuk bidang Keep alive period saat mengonfigurasi sumber daya pekerjaan pelatihan Anda. Nilai ini harus berupa bilangan bulat antara 1 dan 3600, yang mewakili durasi waktu dalam hitungan detik.
3. Untuk melepaskan kolam hangat dari konsol, pilih pekerjaan latihan tertentu dan pilih Klaster rilis dari menu tarik-turun Tindakan.
4. Untuk melihat informasi lebih lanjut tentang kolam hangat, pilih nama pekerjaan pelatihan. Di halaman detail pekerjaan, gulir ke bawah ke bagian status kolam hangat untuk menemukan status kolam hangat, waktu yang tersisa jika status kolam hangat`Available`, detik billable pool hangat, dan nama pekerjaan pelatihan yang menggunakan kembali kolam hangat jika status kolam hangat adalah. `Reused`

Menggunakan API tingkat rendah SageMaker

Gunakan kolam hangat SageMaker terkelola dengan SageMaker API atau AWS CLI.

API SageMaker

Siapkan kumpulan hangat SageMaker terkelola menggunakan SageMaker API dengan perintah berikut:

- CreateTrainingJob
- UpdateTrainingJob
- ListTrainingJobs
- DescribeTrainingJob

CLI AWS

Siapkan kolam hangat SageMaker terkelola menggunakan AWS CLI dengan perintah berikut:

- create-training-job
- update-training-job
- list-training-jobs

- describe-training-job

Kunci kondisi IAM

Administrator secara opsional dapat menggunakan kunci `sagemaker:KeepAlivePeriod` kondisi untuk membatasi `KeepAlivePeriodInSeconds` batasan pengguna atau grup tertentu. SageMakerkolam hangat yang dikelola dibatasi hingga `KeepAlivePeriodInSeconds` nilai 3600 detik (60 menit), tetapi administrator dapat menurunkan batas ini jika diperlukan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnforceKeepAlivePeriodLimit",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob"
            ],
            "Resource": "*",
            "Condition": {
                "NumericLessThanIfExists": {
                    "sagemaker:KeepAlivePeriod": 1800
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Kunci kondisi untuk Amazon SageMaker di Referensi Otorisasi Layanan.

Pertimbangan-pertimbangan

Pertimbangkan item berikut saat menggunakan kolam hangat yang SageMaker dikelola.

- SageMakerkolam hangat yang dikelola tidak dapat digunakan dengan pelatihan cluster heterogen.
- SageMakerkolam hangat terkelola tidak dapat digunakan dengan instans spot.
- SageMakerkolam hangat yang dikelola dibatasi hingga `KeepAlivePeriodInSeconds` nilai 3600 detik (60 menit).

- Jika kolam hangat terus berhasil mencocokkan pekerjaan pelatihan dalam `KeepAlivePeriodInSeconds` nilai yang ditentukan, klaster hanya dapat terus berjalan selama maksimal 28 hari.

Memantau dan Menganalisis Pekerjaan Pelatihan Menggunakan Amazon CloudWatch Metrics

Pekerjaan SageMaker pelatihan Amazon adalah proses berulang yang mengajarkan model untuk membuat prediksi dengan menyajikan contoh dari kumpulan data pelatihan. Biasanya, algoritma pelatihan menghitung beberapa metrik, seperti kesalahan pelatihan dan akurasi prediksi. Metrik ini membantu mendiagnosis apakah model tersebut belajar dengan baik dan akan menggeneralisasi dengan baik untuk membuat prediksi pada data yang tidak terlihat. Algoritma pelatihan menulis nilai metrik ini ke log, yang SageMaker memantau dan mengirim ke Amazon CloudWatch secara real time. Untuk menganalisis kinerja pekerjaan pelatihan Anda, Anda dapat melihat grafik metrik ini di. CloudWatch Ketika pekerjaan pelatihan telah selesai, Anda juga bisa mendapatkan daftar nilai metrik yang dihitung dalam iterasi terakhirnya dengan memanggil operasi. `DescribeTrainingJob`

> **Note**
>
> Amazon CloudWatch mendukung metrik kustom resolusi tinggi, dan resolusi terbaiknya adalah 1 detik. Namun, semakin halus resolusinya, semakin pendek umur metrik. CloudWatch Untuk resolusi frekuensi 1 detik, CloudWatch metrik tersedia selama 3 jam. Untuk informasi selengkapnya tentang resolusi dan umur CloudWatch metrik, lihat GetMetricStatisticsdi Referensi Amazon CloudWatch API.

> **Tip**
>
> Jika Anda ingin membuat profil pekerjaan pelatihan Anda dengan resolusi yang lebih baik hingga perincian 100 milidetik (0,1 detik) dan menyimpan metrik pelatihan tanpa batas waktu di Amazon S3 untuk analisis khusus kapan saja, pertimbangkan untuk menggunakan Amazon Debugger. SageMaker SageMaker Debugger menyediakan aturan bawaan untuk secara otomatis mendeteksi masalah pelatihan umum; ia mendeteksi masalah pemanfaatan sumber daya perangkat keras (seperti kemacetan CPU, GPU, dan I/O) dan masalah model non-konvergen (seperti overfit, gradien menghilang, dan tensor yang meledak). SageMaker Debugger juga menyediakan visualisasi melalui Studio Classic dan laporan profilingnya.

Untuk menjelajahi visualisasi Debugger, lihat Panduan Dasbor Wawasan SageMaker Debugger, Panduan Laporan Profil Debugger, dan Menganalisis Data Menggunakan Pustaka Klien SMDebug.

Topik

- Mendefinisikan Metrik Pelatihan
- Monitoring Training Job Metrics (CloudWatch Konsol)
- Monitoring Training Job Metrics (SageMakerKonsol)
- Contoh: Melihat Kurva Pelatihan dan Validasi

Mendefinisikan Metrik Pelatihan

SageMaker secara otomatis mem-parsing log pekerjaan pelatihan dan mengirimkan metrik pelatihan ke. CloudWatch Secara default, SageMaker mengirimkan metrik pemanfaatan sumber daya sistem yang tercantum dalam SageMaker Lowongan dan Metrik Titik Akhir. Jika Anda SageMaker ingin mengurai log dan mengirim metrik khusus dari pekerjaan pelatihan algoritme Anda sendiri CloudWatch, Anda perlu menentukan definisi metrik dengan meneruskan nama metrik dan ekspresi reguler saat mengonfigurasi permintaan pekerjaan pelatihan. SageMaker

Anda dapat menentukan metrik yang ingin dilacak menggunakan SageMaker konsol, SageMaker Python SDK, atau API tingkat rendah. SageMaker

Jika Anda menggunakan algoritma Anda sendiri, lakukan hal berikut:

- Pastikan algoritme menulis metrik yang ingin Anda tangkap ke log.
- Tentukan ekspresi reguler yang secara akurat mencari log untuk menangkap nilai metrik yang ingin Anda kirim. CloudWatch

Misalnya, algoritme Anda memancarkan metrik berikut untuk kesalahan pelatihan dan kesalahan validasi:

```
Train_error=0.138318;  Valid_error=0.324557;
```

Jika Anda ingin memantau kedua metrik tersebut CloudWatch, kamus untuk definisi metrik akan terlihat seperti contoh berikut:

```
[
    {
        "Name": "train:error",
        "Regex": "Train_error=(.*?);"
    },
    {
        "Name": "validation:error",
        "Regex": "Valid_error=(.*?);"
    }
]
```

Dalam regex untuk `train:error` metrik yang ditentukan dalam contoh sebelumnya, bagian pertama dari regex menemukan teks yang tepat “Train_error=”, dan ekspresi `(.*?);` menangkap karakter apa pun hingga karakter titik koma pertama muncul. Dalam ungkapan ini, tanda kurung memberi tahu regex untuk menangkap apa yang ada di dalamnya, `.` berarti karakter apa pun, `*` berarti nol atau lebih, dan `?` berarti menangkap hanya sampai contoh pertama karakter. `;`

Tentukan Metrik Menggunakan SageMaker Python SDK

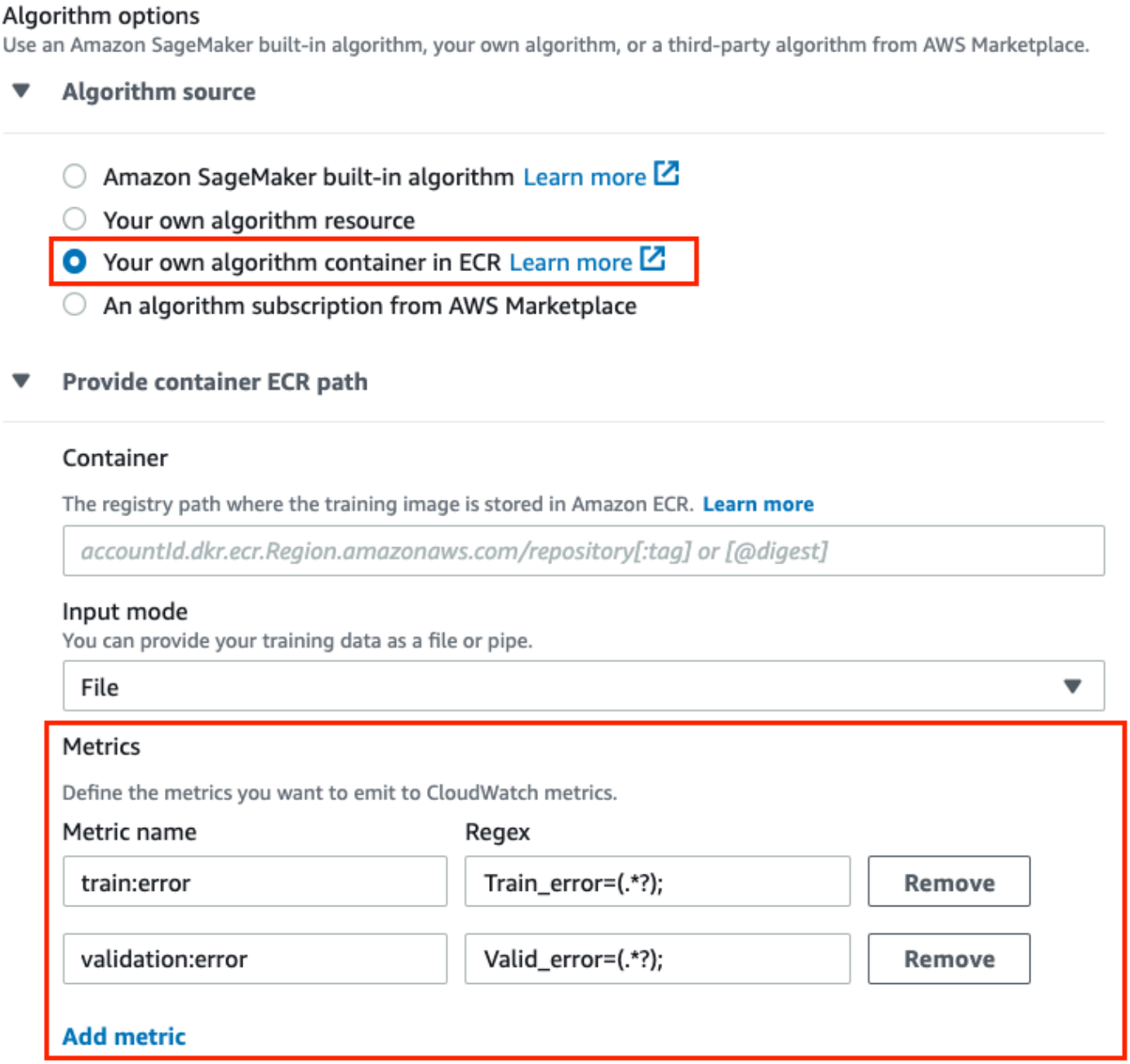
Tentukan metrik yang ingin Anda kirim CloudWatch dengan menentukan daftar nama metrik dan ekspresi reguler sebagai `metric_definitions` argumen saat Anda menginisialisasi objek. `Estimator` Misalnya, jika Anda ingin memantau metrik `train:error` dan `validation:error` metrik CloudWatch, `Estimator` inisialisasi Anda akan terlihat seperti contoh berikut:

```
import sagemaker
from sagemaker.estimator import Estimator

estimator = Estimator(
    image_uri="your-own-image-uri",
    role=sagemaker.get_execution_role(),
    sagemaker_session=sagemaker.Session(),
    instance_count=1,
    instance_type='ml.c4.xlarge',
    metric_definitions=[
       {'Name': 'train:error', 'Regex': 'Train_error=(.*?);'},
       {'Name': 'validation:error', 'Regex': 'Valid_error=(.*?);'}
    ]
)
```

Untuk informasi selengkapnya tentang pelatihan menggunakan estimator Amazon SageMaker Python SDK, lihat Sagemaker Python SDK on. GitHub

Tentukan Metrik Menggunakan Konsol SageMaker

Jika Anda memilih wadah algoritme Anda sendiri di opsi ECR sebagai sumber algoritme Anda di SageMaker konsol saat Anda membuat pekerjaan pelatihan, tambahkan definisi metrik di bagian Metrik. Tangkapan layar berikut menunjukkan bagaimana seharusnya terlihat setelah Anda menambahkan contoh nama metrik dan ekspresi reguler yang sesuai.



Tentukan Metrik Menggunakan API Tingkat Rendah SageMaker

Tentukan metrik yang ingin Anda kirim CloudWatch dengan menentukan daftar nama metrik dan ekspresi reguler di `MetricDefinitions` bidang parameter `AlgorithmSpecification`input yang Anda berikan ke operasi. `CreateTrainingJob` Misalnya, jika Anda ingin memantau metrik `train:error` dan `validation:error` metrik CloudWatch, Anda `AlgorithmSpecification` akan terlihat seperti contoh berikut:

```
"AlgorithmSpecification": {
    "TrainingImage": your-own-image-uri,
    "TrainingInputMode": "File",
    "MetricDefinitions" : [
        {
            "Name": "train:error",
            "Regex": "Train_error=(.*?);"
        },
        {
            "Name": "validation:error",
            "Regex": "Valid_error=(.*?);"
        }
    ]
}
```

Untuk informasi selengkapnya tentang mendefinisikan dan menjalankan tugas pelatihan menggunakan SageMaker API tingkat rendah, lihat. `CreateTrainingJob`

Monitoring Training Job Metrics (CloudWatch Konsol)

Anda dapat memantau metrik yang dipancarkan pekerjaan pelatihan secara real time di konsol. CloudWatch

Untuk memantau metrik pekerjaan pelatihan (CloudWatch konsol)

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch.
2. Pilih Metrik, lalu pilih /aws/sagemaker/ TrainingJobs.
3. Pilih TrainingJobName.
4. Pada tab Semua metrik, pilih nama metrik pelatihan yang ingin Anda pantau.
5. Pada tab Metrik grafik, konfigurasikan opsi grafik. Untuk informasi selengkapnya tentang penggunaan CloudWatch grafik, lihat Metrik Grafik di CloudWatch Panduan Pengguna Amazon.

Monitoring Training Job Metrics (SageMakerKonsol)

Anda dapat memantau metrik yang dipancarkan pekerjaan pelatihan secara real time dengan menggunakan konsol. SageMaker

Untuk memantau metrik pekerjaan pelatihan (SageMaker konsol)

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker.
2. Pilih pekerjaan Pelatihan, lalu pilih pekerjaan pelatihan yang metriknya ingin Anda lihat.
3. Pilih TrainingJobName.
4. Di bagian Monitor, Anda dapat meninjau grafik pemanfaatan instance dan metrik algoritme.


Monitor
Access logs for debugging and progress reporting. View metrics to set alarms, send notifications, or take actions. Learn more
View algorithm metrics
View logs
View instance metrics
2019-01-24 (10:33:57) - 2019-01-24 (16:10:45)
MemoryUtilization
CPUUtilization
DiskUtilization
GPUUtilization
GPUMemoryUtilization
validation:accuracy
train:progress
train:throughput
train:accuracy
validation:cross_entropy
train:cross_entropy

Contoh: Melihat Kurva Pelatihan dan Validasi

Biasanya, Anda membagi data tempat Anda melatih model menjadi kumpulan data pelatihan dan validasi. Anda menggunakan set pelatihan untuk melatih parameter model yang digunakan untuk membuat prediksi pada kumpulan data pelatihan. Kemudian Anda menguji seberapa baik model membuat prediksi dengan menghitung prediksi untuk set validasi. Untuk menganalisis kinerja pekerjaan pelatihan, Anda biasanya merencanakan kurva pelatihan terhadap kurva validasi.

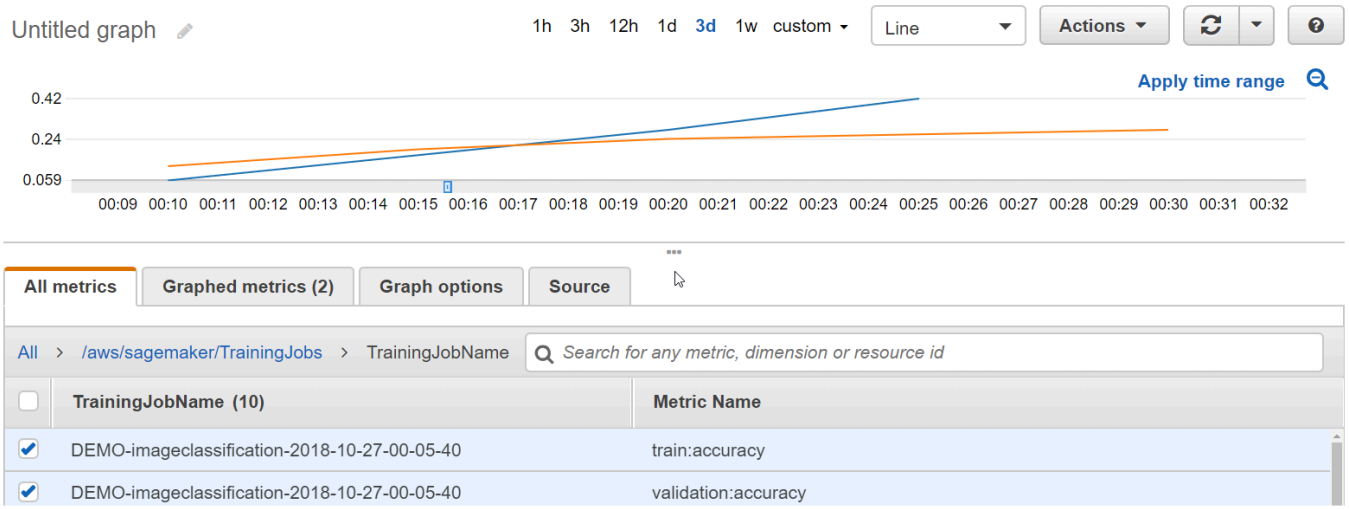
Melihat grafik yang menunjukkan keakuratan untuk set pelatihan dan validasi dari waktu ke waktu dapat membantu Anda meningkatkan kinerja model Anda. Misalnya, jika akurasi pelatihan terus meningkat dari waktu ke waktu, tetapi, pada titik tertentu, akurasi validasi mulai menurun, kemungkinan Anda terlalu cocok dengan model Anda. Untuk mengatasinya, Anda dapat melakukan penyesuaian pada model Anda, seperti meningkatkan regularisasi.

Untuk contoh ini, Anda dapat menggunakan mage-classification-full-training contoh I di bagian Contoh notebook dari instance SageMaker notebook Anda. Jika Anda tidak memiliki instance SageMaker notebook, buat satu dengan mengikuti petunjuk diLangkah 1: Buat Instans SageMaker Notebook Amazon. Jika mau, Anda dapat mengikuti bersama dengan Contoh Klasifikasi Gambar Multiclass End-to-End di buku catatan contoh. GitHub Anda juga memerlukan bucket Amazon S3 untuk menyimpan data pelatihan dan untuk output model.

Untuk melihat kurva kesalahan pelatihan dan validasi

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker.
2. Pilih Notebook, lalu pilih instans Notebook.
3. Pilih instance notebook yang ingin Anda gunakan, lalu pilih Buka.
4. Di dasbor untuk instance notebook Anda, pilih SageMakerContoh.
5. Perluas bagian Pengantar Algoritma Amazon, lalu pilih Gunakan di sebelah mage-classification-fulltrainingI.ipynb.
6. Pilih Buat salinan. SageMaker membuat salinan notebook mage-classification-fulltrainingI.ipynb yang dapat diedit di instance notebook Anda.
7. Jalankan semua sel di buku catatan hingga bagian Inferensi. Anda tidak perlu menerapkan titik akhir atau mendapatkan inferensi untuk contoh ini.
8. Setelah pekerjaan pelatihan dimulai, buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch.
9. Pilih Metrik, lalu pilih /aws/sagemaker/ TrainingJobs.

10. Pilih TrainingJobName.
11. Pada tab Semua metrik, pilih metrik train:akurasi dan validasi:akurasi untuk pekerjaan pelatihan yang Anda buat di buku catatan.
12. Pada grafik, pilih area yang nilainya akan diperbesar oleh metrik. Anda akan melihat sesuatu seperti contoh berikut.



Gunakan AmazonSageMakerJalur Penyimpanan Pelatihan untuk Set Data Pelatihan, Pos Pemeriksaan, Artefak Model, dan Output

Halaman ini memberikan ringkasan tingkat tinggi tentang bagaimanaSageMakerplatform pelatihan mengelola jalur penyimpanan untuk set data pelatihan, artefak model, pos pemeriksaan, dan output antaraAWSpenyimpanan awan dan pekerjaan pelatihan diSageMaker. Sepanjang panduan ini, Anda belajar mengidentifikasi jalur default yang ditetapkan olehSageMakerplatform dan bagaimana saluran data dapat disederhanakan dengan sumber data Anda di Amazon Simple Storage Service (Amazon S3), FSx for Lustre, dan Amazon EFS. Untuk informasi selengkapnya tentang berbagai mode input saluran data dan opsi penyimpanan, lihatAkses Data Pelatihan.

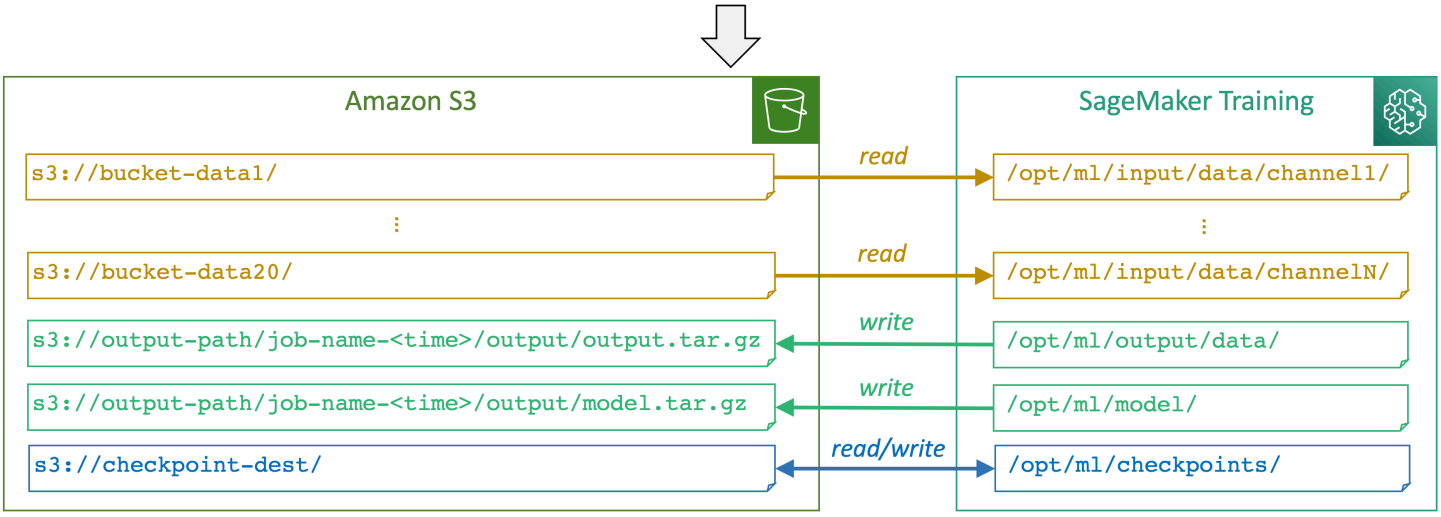
Topik

- Gambaran Umum
- Output model terkompresi
- Tips dan Pertimbangan untuk Menyiapkan Jalur Penyimpanan
- SageMakerVariabel Lingkungan dan Jalur Default untuk Lokasi Penyimpanan Pelatihan

Gambaran Umum

Diagram berikut menunjukkan contoh paling sederhana tentang bagaimanaSageMakermengelola jalur input dan output saat Anda menjalankan pekerjaan pelatihan menggunakanSageMakerPythonEstimatorkelas dan nyacocokmetode. Ini didasarkan pada penggunaan mode file sebagai strategi akses data dan Amazon S3 sebagai sumber data untuk saluran input pelatihan.

```
estimator = Estimator(
    checkpoint_s3_uri='s3://checkpoint-dest/',
    output_path='s3://output-path/',
    base_job_name='job-name',
    input_mode='File'
    ...
)

estimator.fit(inputs={
    'channel1' : 's3://bucket-data1/',
    ...
    'channel20': 's3://bucket-data20/'})
```



This figure shows an overview of how SageMaker pairs storage paths between an Amazon S3 bucket as the data source and the SageMaker training instance based on how the paths are specified in a SageMaker estimator class. More information about the paths, how they read from or write to the paths, and purposes of the paths are described in the following section the section called "SageMakerVariabel Lingkungan dan Jalur Default untuk Lokasi Penyimpanan Pelatihan".

Anda dapat menggunakan`OutputDataConfig`di dalamCreateTrainingJobAPI untuk menemukan di mana bucket S3 Anda berada. GunakanModelArtifactsAPI untuk menemukan lokasi S3 yang berisi artefak model Anda. Lihatabalone_build_train_deploynotebook untuk contoh jalur output dan bagaimana mereka digunakan dalam panggilan API.

Untuk informasi lebih lanjut dan contoh bagaimanaSageMakermengelola sumber data, mode input, dan jalur lokal diSageMakercontoh pelatihan, lihatAkses Data Pelatihan.

Output model terkompresi

SageMakermenyimpan model Anda di`/opt/ml/model`dan data Anda di`/opt/ml/output/data`. Setelah model dan data ditulis ke lokasi tersebut, model tersebut diunggah ke bucket Amazon S3 Anda sebagai file terkompresi secara default.

Anda dapat menghemat waktu pada kompresi file data besar dengan mengunggah model dan output data ke bucket S3 Anda sebagai file yang tidak terkompresi. Untuk melakukan ini, buat pekerjaan pelatihan dalam mode unggahan yang tidak terkompresi dengan menggunakan salah satuAWS Command Line Interface(AWS CLI) atauSageMakerPython SDK.

Contoh kode berikut menunjukkan cara membuat pekerjaan pelatihan dalam mode upload terkompresi saat menggunakanAWS CLI. Untuk mengaktifkan mode unggahan yang tidak terkompresi, setel`CompressionType`bidang di`OutputDataConfig`API ke**`NONE`**.

```
{
   "TrainingJobName": "uncompressed_model_upload",
   ...
   "OutputDataConfig": {
      "S3OutputPath": "s3://DOC-EXAMPLE-BUCKET/uncompressed_upload/output",
      "CompressionType": "NONE"
   },
   ...
}
```

Contoh kode berikut menunjukkan cara membuat pekerjaan pelatihan dalam mode upload yang tidak terkompresi menggunakanSageMakerPython SDK.

```
import sagemaker
from sagemaker.estimator import Estimator

estimator = Estimator(
    image_uri="your-own-image-uri",
    role=sagemaker.get_execution_role(),
    sagemaker_session=sagemaker.Session(),
    instance_count=1,
    instance_type='ml.c4.xlarge',
```

```
    disable_output_compression=True
)
```

Tips dan Pertimbangan untuk Menyiapkan Jalur Penyimpanan

Pertimbangkan item berikut saat menyiapkan jalur penyimpanan untuk pekerjaan pelatihan diSageMaker.

- Jika Anda ingin menyimpan artefak pelatihan untuk pelatihan terdistribusi di`/opt/ml/output/data`direktori, Anda harus menambahkan subdirektori dengan benar atau menggunakan nama file unik untuk artefak melalui definisi model atau skrip pelatihan Anda. Jika subdirektori dan nama file tidak dikonfigurasi dengan benar, semua pekerja pelatihan terdistribusi mungkin menulis output ke nama file yang sama di jalur keluaran yang sama di Amazon S3.
- Jika Anda menggunakan wadah pelatihan khusus, pastikan Anda menginstalSageMakerToolkit Pelatihanyang membantu mengatur lingkungan untukSageMakerpekerjaan pelatihan. Jika tidak, Anda harus menentukan variabel lingkungan secara eksplisit di Dockerfile Anda. Untuk informasi lebih lanjut, lihatBuat wadah dengan algoritme dan model Anda sendiri.
- Saat menggunakan instans ML denganVolume NVMe SSD,SageMakertidak menyediakan penyimpanan gp2 Amazon EBS. Penyimpanan yang tersedia diperbaiki ke kapasitas penyimpanan instans tipe NVME. SageMakermengkonfigurasi jalur penyimpanan untuk set data pelatihan, pos pemeriksaan, artefak model, dan output untuk menggunakan seluruh kapasitas penyimpanan instans. Misalnya, keluarga instans MLdengan penyimpanan instans tipe NVME termasuk`ml.p4d`,`ml.g4dn`, dan`ml.g5`. Saat menggunakan instans MLdengan opsi penyimpanan khusus EBS dan tanpa penyimpanan instans, Anda harus menentukan ukuran volume EBS melalui`volume_size`parameter dalamSageMakerkelas estimator (atau`VolumeSizeInGB`jika Anda menggunakan`ResourceConfig`API). Misalnya, keluarga instans MLyang menggunakan volume EBS mencakup`ml.c5`dan`ml.p2`. Untuk mencari jenis instans dan jenis dan volume penyimpanan instans mereka, lihatJenis Instans Amazon EC2.
- Jalur default untukSageMakerpekerjaan pelatihan dipasang ke volume Amazon EBS atau volume NVMe SSD instans MS. Saat Anda menyesuaikan skrip pelatihan AndaSageMaker, pastikan bahwa Anda menggunakan jalur default yang tercantum dalam topik sebelumnya tentangthe section called “SageMakerVariabel Lingkungan dan Jalur Default untuk Lokasi Penyimpanan Pelatihan”. Kami menyarankan Anda menggunakan`/tmp`direktori sebagai ruang awal untuk menyimpan sementara benda besar selama pelatihan. Ini berarti bahwa Anda tidak boleh menggunakan direktori yang dipasang ke ruang disk kecil yang dialokasikan untuk sistem, seperti`/user`dan`/home`, untuk menghindariout-of-spacekesalahan.

Untuk mempelajari lebih lanjut, lihatAWSblog pembelajaran mesinPilih sumber data terbaik untuk Amazon AndaSageMakerpekerjaan pelatihanyang lebih lanjut membahas studi kasus dan tolok ukur kinerja sumber data dan mode input.

SageMakerVariabel Lingkungan dan Jalur Default untuk Lokasi Penyimpanan Pelatihan

Tabel berikut merangkum jalur input dan output untuk set data pelatihan, pos pemeriksaan, artefak model, dan output, yang dikelola olehSageMakerplatform pelatihan.

Jalur lokal diSageMakercontoh pelatihan	Variabel lingkungan SageMaker	Tujuan	Baca dari S3 saat memulai	Baca dari S3 selama Spot-Restart	Menulis ke S3 selama pelatihan	Menulis ke S3 saat pekerjaan dihentikan
`/opt/ml/input/data/` *channel_name* [1]	SALURAN_*AME*	Membaca data pelatihan dari saluran input yang ditentukan melaluiSageMakerPythonEstimatorkelas atauCreateTrainingJobOperasi API. Untuk informasi selengkapnya tentang cara menentukannya di skrip latihan Anda menggunakanSageMakerPython SDK, lihatSiapkan skrip Pelatihan.	Ya	Ya	Tidak	Tidak
`/opt/ml/output/data` [2]	OUTPUT_DR	Menyimpan output seperti kehilangan, akurasi, lapisan menengah, bobot,	Tidak	Tidak	Tidak	Ya

Jalur lokal diSageMakercontoh pelatihan	Variabel lingkungan SageMaker	Tujuan	Baca dari S3 saat memulai	Baca dari S3 selama Spot-Restart	Menulis ke S3 selama pelatihan	Menulis ke S3 saat pekerjaan dihentikan
		gradien, bias, danTensorBoardoutput -kompatibel. Anda juga dapat menyimpan output sewenang-wenang yang Anda inginkan menggunakan jalur ini. Perhatikan bahwa ini adalah jalur yang berbeda dari yang untuk menyimpan artefak model akhir/ `opt/ml/model/` .				
`/opt/ml/model` [3]	SM_MODEDIR	Menyimpan model artefak akhir. Ini juga merupakan jalur dari mana artefak model digunakan Inferensi waktu nyatadiSageMakerHosting.	Tidak	Tidak	Tidak	Ya

Jalur lokal diSageMakercontoh pelatihan	Variabel lingkungan SageMaker	Tujuan	Baca dari S3 saat memulai	Baca dari S3 selama Spot-Restart	Menulis ke S3 selama pelatihan	Menulis ke S3 saat pekerjaan dihentikan
`/opt/ml/checkpoints` [4]	-	Menyimpan pos pemeriksaan model (keadaan model) untuk melanjutkan pelatihan dari titik tertentu, dan pulih dari yang tidak terduga atauPelatihan Spot Terkelolainterupsi.	Ya	Ya	Ya	Tidak
`/opt/ml/code`	SAGEMAK_SUBMIT_DIRECTORY	Menyalin skrip pelatihan, pustaka tambahan, dan dependensi.	Ya	Ya	Tidak	Tidak
`/tmp`	-	Membaca atau menulis ke`/tmp`sebagai ruang awal.	Tidak	Tidak	Tidak	Tidak

[1] `channel_name`adalah tempat untuk menentukan nama saluran yang ditentukan pengguna untuk input data pelatihan. Setiap pekerjaan pelatihan dapat berisi beberapa saluran input data. Anda dapat menentukan hingga 20 saluran input pelatihan per pekerjaan pelatihan. Perhatikan bahwa waktu pengunduhan data dari saluran data dihitung ke waktu yang dapat ditagih. Untuk informasi selengkapnya tentang jalur input data, lihatBagaimana AmazonSageMakerMemberikan Informasi Pelatihan. Juga, ada tiga jenis mode input data yangSageMakermendukung: file,FastFile, dan mode pipa. Untuk mempelajari lebih lanjut tentang mode input data untuk pelatihan diSageMaker, lihatAkses Data Pelatihan.

[2] SageMakerkompres dan menulis artefak pelatihan ke file TAR (`tar.gz`). Waktu kompresi dan pengunggahan dihitung ke waktu yang dapat ditagih. Untuk informasi lebih lanjut, lihatBagaimana AmazonSageMakerProses Output Pelatihan.

[3] SageMakerkompres dan menulis model artefak akhir ke file TAR (`tar.gz`). Waktu kompresi dan pengunggahan dihitung ke waktu yang dapat ditagih. Untuk informasi lebih lanjut, lihatBagaimana AmazonSageMakerProses Output Pelatihan.

[4]Sinkronkan dengan Amazon S3 selama latihan. Tulis apa adanya tanpa mengompresi ke file TAR. Untuk informasi lebih lanjut, lihatMenggunakan Pos Pemeriksaan di AmazonSageMaker.

Menyediakan Dataset Metadata untuk Training Jobs dengan Augmented Manifest File

Untuk menyertakan metadata dengan dataset Anda dalam pekerjaan pelatihan, gunakan file manifes yang ditambah. Saat menggunakan file manifes tambahan, dataset Anda harus disimpan di Amazon Simple Storage Service (Amazon S3), dan Anda harus mengonfigurasi tugas pelatihan untuk menggunakan dataset yang disimpan di sana. Anda menentukan lokasi dan format dataset ini untuk satu atau lebih`Channel`. Augmented manifests hanya dapat mendukung mode input Pipa. Lihat bagianInputModedi`Channel`untuk mempelajari lebih lanjut tentang mode input

Saat menentukan parameter saluran, Anda menentukan jalur ke file, yang disebut`S3Uri`. Amazon SageMaker menafsirkan URI ini berdasarkan ditentukan`S3DataType`di`S3DataSource`. Parameter`AugmentedManifestFile`pilihan mendefinisikan format manifes yang mencakup metadata dengan data input. Menggunakan file manifes augmented adalah alternatif untuk preprocessing ketika Anda memiliki label data. Untuk pekerjaan pelatihan menggunakan data berlabel, Anda biasanya perlu melakukan pra-proses dataset untuk menggabungkan data input dengan metadata sebelum latihan. Jika dataset Anda besar, preprocessing dapat memakan waktu dan biaya yang besar.

Format File Manifes

File manifes augmented harus diformatGaris JSONformat. Dalam format JSON Garis, setiap baris dalam file adalah objek JSON lengkap diikuti oleh pemisah baris baru.

Selama pelatihan, SageMaker mem-parsing setiap baris JSON dan mengirimkan beberapa atau semua atributnya ke algoritma pelatihan. Anda menentukan isi atribut untuk lulus dan urutan

di mana untuk lulus mereka dengan`AttributeNames`parameter`CreateTrainingJob`API Parameter`AttributeNames`parameter adalah daftar memerintahkan nama atribut yang SageMaker mencari di objek JSON untuk digunakan sebagai masukan pelatihan.

Misalnya, jika Anda daftar`["line", "book"]`untuk`AttributeNames`, data input harus menyertakan nama atribut`line`dan`book`dalam urutan yang ditentukan. Untuk contoh ini, konten file manifes augmented berikut ini valid:

```
{"author": "Herman Melville", "line": "Call me Ishmael", "book": "Moby Dick"}
{"line": "It was love at first sight.", "author": "Joseph Heller", "book": "Catch-22"}
```

SageMaker mengabaikan nama atribut yang tidak terdaftar bahkan jika mereka mendahului, mengikuti, atau berada di antara atribut yang terdaftar.

Saat menggunakan file manifes yang ditambah, amati pedoman berikut ini:

- Urutan atribut yang tercantum dalam`AttributeNames`parameter menentukan urutan atribut diteruskan ke algoritma dalam pekerjaan pelatihan.
- Daftar`AttributeNames`dapat menjadi bagian dari semua atribut di baris JSON. SageMaker mengabaikan atribut yang tidak terdaftar dalam file.
- Anda dapat menentukan semua jenis data yang diizinkan oleh format JSON di`AttributeNames`, termasuk teks, numerik, array data, atau objek.
- Untuk menyertakan URI S3 sebagai nama atribut, tambahkan akhiran`-ref`untuk itu.

Jika nama atribut berisi akhiran`-ref`, nilai atribut harus berupa URI S3 ke file data yang dapat diakses oleh pekerjaan pelatihan. Misalnya, jika`AttributeNames`mengandung`["image-ref", "is-a-cat"]`, contoh berikut menunjukkan file manifes augmented valid:

```
{"image-ref": "s3://mybucket/sample01/image1.jpg", "is-a-cat": 1}
{"image-ref": "s3://mybucket/sample02/image2.jpg", "is-a-cat": 0}
```

Dalam kasus baris JSON pertama dari file manifes ini, SageMaker mengambil`image1.jpg`file dari`s3://mybucket/sample01/`dan representasi string`is-a-cat`tambahan`"1"`untuk klasifikasi gambar.

> **Tip**
>
> Untuk membuat file manifes yang ditambah, gunakan Amazon SageMaker Ground Truth dan menciptakan pekerjaan pelabelan. Untuk informasi selengkapnya tentang output dari tugas pelabelan, lihatData Output.

Streaming Data File Manifest Augmented

Format manifes Augmented memungkinkan Anda melakukan pelatihan dalam mode Pipe menggunakan file tanpa perlu membuat file RecordIO. Anda perlu menentukan saluran kereta api dan validasi sebagai nilai untuk`InputDataConfig`parameter`CreateTrainingJob`permintaan. File manifes augmented hanya didukung untuk saluran yang menggunakan mode input Pipa. Untuk setiap saluran, data diekstraksi dari file manifesnya yang ditambah dan dialirkan (dalam rangka) ke algoritma melalui pipa bernama saluran. Mode pipa menggunakan metode pertama keluar pertama (FIFO), sehingga catatan diproses sesuai urutan di mana mereka antrian. Untuk informasi tentang mode input pipa, lihat`Input Mode`.

Atribut nama dengan`"-ref"`titik akhiran untuk data biner terformat. Dalam beberapa kasus, algoritma tahu bagaimana mengurai data. Dalam kasus lain, Anda mungkin perlu membungkus data sehingga catatan dibatasi untuk algoritma. Jika algoritma ini kompatibel dengan Data yang diformat rekaman, menentukan`RecordIO`untuk`RecordWrapperType`memecahkan masalah ini. Jika algoritma tidak kompatibel dengan`RecordIO`format, tentukan`None`untuk`RecordWrapperType`dan pastikan bahwa data Anda diurai dengan benar untuk algoritma Anda.

Menggunakan`["image-ref", "is-a-cat"]`contoh, jika Anda menggunakan pembungkus RecordIO, aliran berikut data dikirim ke antrian:

```
recordio_formatted(s3://mybucket/foo/
image1.jpg)recordio_formatted("1")recordio_formatted(s3://mybucket/bar/
image2.jpg)recordio_formatted("0")
```

Gambar yang tidak dibungkus dengan format RecordIO, dialirkan dengan yang sesuai`is-a-cat`nilai atribut sebagai satu record. Hal ini dapat menyebabkan masalah karena algoritma mungkin tidak membatasi gambar dan atribut dengan benar. Untuk informasi selengkapnya tentang penggunaan file manifes yang ditambah untuk klasifikasi gambar, lihatLatih dengan Format Gambar Manifest Augmented.

Dengan file manifes augmented dan mode Pipe secara umum, batas ukuran volume EBS tidak berlaku. Ini termasuk pengaturan yang harus berada dalam batas ukuran volume EBS seperti`S3DataDistributionType`. Untuk informasi selengkapnya tentang mode Pipa dan cara menggunakannya, lihatMenggunakan Algoritma Pelatihan Anda Sendiri - Konfigurasi Data Input.

Menggunakan Augmented Manifest File (Console)

Untuk menyelesaikan prosedur ini, Anda memerlukan:

- URL bucket S3 tempat Anda menyimpan file manifes yang ditambahkan.
- Untuk menyimpan data yang tercantum dalam file manifes augmented dalam bucket S3.
- URL bucket S3 tempat Anda ingin menyimpan output tugas.

Menggunakan file manifes yang ditambah dalam pekerjaan pelatihan (konsol)

1. Buka Amazon SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilihPelatihan, kemudian pilihTugas pelatihan.
3. PilihMembuat pekerjaan pelatihan.
4. Berikan nama untuk pekerjaan pelatihan. Nama harus unikAWSWilayah diAWSakun. Hal ini dapat memiliki 1 hingga 63 karakter. Karakter yang benar: a-z, A-Z, 0-9, dan.: + = @ _% - (tanda hubung).
5. Pilih algoritma yang ingin Anda gunakan. Untuk informasi tentang algoritme tertanam yang didukung, lihatGunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih. Jika Anda ingin menggunakan algoritme khusus, pastikan bahwa itu kompatibel dengan mode Pipa.
6. (Opsional)Konfigurasi sumber daya, baik menerima nilai default atau, untuk mengurangi waktu perhitungan, meningkatkan konsumsi sumber daya.
   a. (Opsional)Jenis instans, pilih tipe instans komputasi yang ingin Anda gunakan. Dalam kebanyakan kasus,db.m4.xlargesudah cukup.
   b. UntukJumlah instans, gunakan default,1.
   c. (Opsional)Volume tambahan per instans (GB), pilih volume penyimpanan L yang ingin Anda sediakan. Dalam kebanyakan kasus, Anda dapat menggunakan default,1. Jika Anda menggunakan dataset besar, gunakan ukuran yang lebih besar.
7. Memberikan informasi tentang data masukan untuk dataset pelatihan.

a. UntukNama saluran, baik menerima default (**`train`**) atau masukkan nama yang lebih berarti, seperti**`training-augmented-manifest-file`**.

b. UntukInputMode, pilihPIPA.

c. UntukTipe distribusi data S3, pilihFullyReplicated. Ketika latihan secara bertahap, replikasi sepenuhnya menyebabkan setiap instans komputasi ML-menggunakan salinan lengkap dari dataset yang diperluas. Untuk algoritma berbasis saraf, seperti[Algoritma Model Topik Saraf (NTM)](), pilih`ShardedByS3Key`.

d. Jika data yang ditentukan dalam file manifes augmented tidak terkompresi, aturJenis kompresikepadaTidak ada. Jika data dikompresi menggunakan gzip, atur keGzip.

e. (Opsional)Jenis Konten, tentukan jenis MIME yang sesuai. Jenis konten adalah jenis ekstensi email internet multiguna (MIME) dari data.

f. UntukPembungkus rekaman, jika dataset yang ditentukan dalam file manifes augmented disimpan dalam format RecordIO, pilihRecordIO. Jika dataset Anda tidak disimpan sebagai file yang diformat Recordio-, pilihTidak ada.

g. UntukTipe data S3, pilihAugmentedManifestFile.

h. UntukLokasi S3, menyediakan path ke bucket tempat Anda menyimpan file manifes augmented.

i. UntukAugmentedManifestFile nama atribut, tentukan nama atribut yang ingin Anda gunakan. Nama atribut harus ada dalam file manifes augmented, dan case-sensitive.

j. (Opsional) Untuk menambahkan nama atribut lainnya, pilihTambahkan barisdan tentukan nama atribut lain untuk setiap atribut.

k. (Opsional) Untuk menyesuaikan urutan nama atribut, pilih tombol atas atau bawah di samping nama. Bila menggunakan file manifes augmented, urutan nama atribut yang ditentukan penting.

l. PilihSelesai.

8. UntukKonfigurasi data output, memberikan informasi berikut:

a. UntukLokasi S3, ketik jalur ke bucket S3 tempat Anda ingin menyimpan data output.

b. (Opsional) Anda dapat menggunakanAWS Key Management Service(AWS KMS) kunci enkripsi untuk mengenkripsi data output saat istirahat. UntukKunci enkripsi, berikan ID kunci atau Amazon Resource Number (ARN). Untuk informasi selengkapnya, lihat[Kunci Enkripsi yang Dikelola KMS]().

9. (Opsional)Tag, tambahkan satu tanda atau lebih ke tugas pelatihan. SEBUAHmenandaiadalah metadata yang dapat Anda tentukan dan tetapkan keAWSsumber daya. Dalam hal ini, Anda dapat menggunakan tag untuk membantu Anda mengelola pekerjaan pelatihan Anda. Sebuah tag terdiri atas sebuah kunci dan sebuah nilai, yang Anda tentukan. Misalnya, Anda mungkin ingin membuat tanda dengan**`Project`**sebagai kunci dan nilai yang mengacu pada proyek yang terkait dengan pekerjaan pelatihan, seperti**`Home value forecasts`**.
10. PilihMembuat pekerjaan pelatihan. SageMaker menciptakan dan menjalankan pekerjaan pelatihan.

Setelah pekerjaan pelatihan selesai, SageMaker menyimpan artefak model di ember yang jalurnya Anda berikanJalur output S3diKonfigurasi data outputbidang. Untuk menyebarkan model untuk mendapatkan prediksi, lihatLangkah 5: Menyebarkan Model ke Amazon EC2.

Menggunakan Augmented Manifest File (API)

Berikut ini menunjukkan cara melatih model dengan file manifes yang ditambahmenggunakan SageMaker perpustakaan Python tingkat tinggi:

```
import sagemaker

# Create a model object set to using "Pipe" mode.
model = sagemaker.estimator.Estimator(
    training_image,
    role,
    instance_count=1,
    instance_type='ml.p3.2xlarge',
    volume_size = 50,
    max_run = 360000,
    input_mode = 'Pipe',
    output_path=s3_output_location,
    sagemaker_session=session
)

# Create a train data channel with S3_data_type as 'AugmentedManifestFile' and
 attribute names.
train_data = sagemaker.inputs.TrainingInput(
    your_augmented_manifest_file,
    distribution='FullyReplicated',
    content_type='application/x-recordio',
    s3_data_type='AugmentedManifestFile',
```

```
    attribute_names=['source-ref', 'annotations'],
    input_mode='Pipe',
    record_wrapping='RecordIO'
)

data_channels = {'train': train_data}

# Train a model.
model.fit(inputs=data_channels, logs=True)
```

Setelah pekerjaan pelatihan selesai, SageMaker menyimpan artefak model di ember yang jalurnya Anda berikanJalur output S3diKonfigurasi data outputbidang. Untuk menyebarkan model untuk mendapatkan prediksi, lihatLangkah 5: Menyebarkan Model ke Amazon EC2.

Menggunakan Pos Pemeriksaan di Amazon SageMaker

Gunakan pos pemeriksaan di Amazon SageMaker untuk menyimpan model status machine learning (ML/machine learning) selama latihan. Checkpoint adalah snapshot dari model dan dapat dikonfigurasi oleh fungsi callback dari kerangka kerja MS. Anda dapat menggunakan pos pemeriksaan yang disimpan untuk memulai kembali pekerjaan pelatihan dari pos pemeriksaan terakhir yang disimpan.

Mekanisme SageMaker pelatihan menggunakan wadah pelatihan pada instans Amazon EC2, dan file pos pemeriksaan disimpan di bawah direktori lokal kontainer (defaultnya adalah). `/opt/ml/checkpoints` SageMakermenyediakan fungsionalitas untuk menyalin pos pemeriksaan dari jalur lokal ke Amazon S3 dan secara otomatis menyinkronkan pos pemeriksaan di direktori tersebut dengan Amazon S3. Pos pemeriksaan yang ada di S3 ditulis ke SageMaker wadah pada awal pekerjaan, memungkinkan pekerjaan untuk melanjutkan dari pos pemeriksaan. Pos pemeriksaan yang ditambahkan ke folder S3 setelah pekerjaan dimulai tidak disalin ke wadah pelatihan. SageMakerjuga menulis pos pemeriksaan baru dari wadah ke S3 selama pelatihan. Jika pos pemeriksaan dihapus dalam SageMaker wadah, itu juga akan dihapus di folder S3.

Menggunakan pos pemeriksaan, Anda dapat melakukan hal berikut:

- Simpan snapshot model Anda di bawah pelatihan karena gangguan yang tidak terduga pada pekerjaan atau instans pelatihan.
- Lanjutkan melatih model di masa depan dari pos pemeriksaan.
- Analisis model pada tahap menengah pelatihan.

- Gunakan pos pemeriksaan dengan pelatihan spot SageMaker terkelola untuk menghemat biaya pelatihan.

Jika Anda menggunakan pos pemeriksaan dengan pelatihan spot SageMaker terkelola, SageMaker kelola checkpointing pelatihan model Anda pada instans spot dan melanjutkan pekerjaan pelatihan pada instance spot berikutnya. Dengan pelatihan spot SageMaker terkelola, Anda dapat secara signifikan mengurangi waktu yang dapat ditagih untuk melatih model mL. Untuk informasi selengkapnya, lihat Gunakan Pelatihan Spot Terkelola di Amazon SageMaker.

Topik

Pos pemeriksaan untuk Kerangka Kerja dan Algoritma di SageMaker

Gunakan pos pemeriksaan untuk menyimpan snapshot model MS yang dibuat di kerangka kerja pilihan Anda. SageMaker

SageMakerframework dan algoritma yang mendukung checkpointing

SageMakermendukung checkpointing untuk AWS Deep Learning Containers dan subset algoritma bawaan tanpa memerlukan perubahan skrip pelatihan. SageMakermenyimpan pos pemeriksaan ke jalur lokal default `'/opt/ml/checkpoints'` dan menyalinnya ke Amazon S3.

- Wadah Pembelajaran Mendalam: TensorFlow, PyTorch, MXNet, dan HuggingFace

> **Note**
>
> Jika Anda menggunakan estimator HuggingFace kerangka kerja, Anda perlu menentukan jalur output pos pemeriksaan melalui hyperparameters. Untuk informasi selengkapnya, lihat Menjalankan pelatihan di Amazon SageMaker dalam HuggingFacedokumentasi.

- Algoritma bawaan: Klasifikasi Gambar, Deteksi Objek, Segmentasi Semantik, dan XGBoost (0,90-1 atau lebih baru)

> **Note**
>
> Jika Anda menggunakan algoritma XGBoost dalam mode kerangka kerja (mode skrip), Anda perlu membawa skrip pelatihan XGBoost dengan checkpointing yang dikonfigurasi secara manual. Untuk informasi selengkapnya tentang metode pelatihan XGBoost untuk menyimpan snapshot model, lihat Pelatihan XGBoost dalam dokumentasi XGBoost Python SDK.

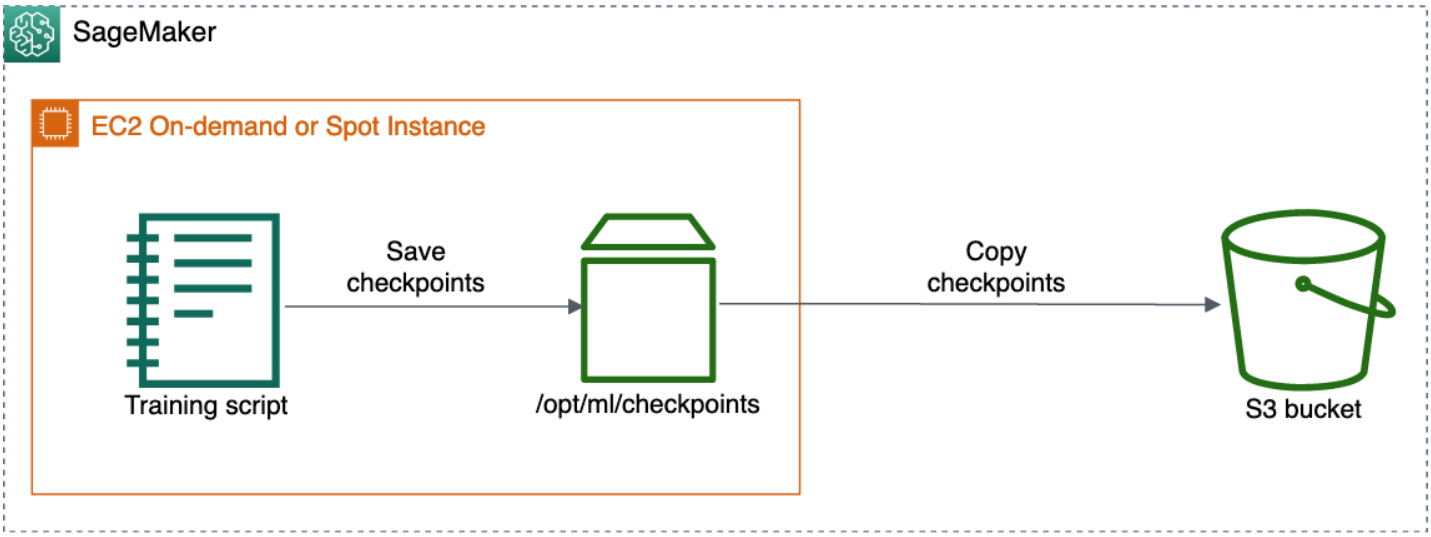
Jika algoritma pra-bangun yang tidak mendukung checkpointing digunakan dalam pekerjaan pelatihan spot terkelola, SageMaker tidak memungkinkan waktu tunggu maksimum lebih dari satu jam untuk pekerjaan untuk membatasi waktu pelatihan yang terbuang dari interupsi.

Untuk wadah pelatihan khusus dan kerangka kerja lainnya

Jika Anda menggunakan wadah latihan, skrip latihan, atau kerangka kerja lain yang tidak tercantum di bagian sebelumnya, Anda harus menyiapkan skrip latihan dengan benar menggunakan callback atau API pelatihan untuk menyimpan pos pemeriksaan ke jalur lokal (`'/opt/ml/checkpoints'`) dan memuat dari jalur lokal dalam skrip latihan Anda. SageMakerestimator dapat disinkronkan dengan jalur lokal dan menyimpan pos pemeriksaan ke Amazon S3.

Aktifkan Checkpointing

Setelah Anda mengaktifkan checkpointing, SageMaker simpan pos pemeriksaan ke Amazon S3 dan sinkronkan tugas latihan Anda dengan bucket checkpoint S3.



Contoh berikut menunjukkan cara mengkonfigurasi jalur pos pemeriksaan ketika Anda membangun estimator. SageMaker Untuk mengaktifkan checkpointing, tambahkan `checkpoint_s3_uri` dan `checkpoint_local_path` parameter ke estimator Anda.

Contoh template berikut menunjukkan cara membuat SageMaker estimator generik dan mengaktifkan checkpointing. Anda dapat menggunakan template ini untuk algoritma yang didukung dengan menentukan parameter. `image_uri` Untuk menemukan URI image Docker untuk algoritme dengan checkpointing didukung olehSageMaker, lihat Jalur Registri Docker dan Kode Contoh. Anda juga dapat mengganti `estimator` dan `Estimator` dengan kelas induk estimator SageMaker kerangka kerja lain dan kelas estimator, seperti`TensorFlow`,,, dan. `PyTorch` `MXNet` `HuggingFace` `XGBoost`

```
import sagemaker
from sagemaker.estimator import Estimator

bucket=sagemaker.Session().default_bucket()
base_job_name="sagemaker-checkpoint-test"
checkpoint_in_bucket="checkpoints"

# The S3 URI to store the checkpoints
checkpoint_s3_bucket="s3://{}/{}/{}".format(bucket, base_job_name,
 checkpoint_in_bucket)

# The local path where the model will save its checkpoints in the training container
checkpoint_local_path="/opt/ml/checkpoints"

estimator = Estimator(
    ...
    image_uri="<ecr_path>/<algorithm-name>:<tag>" # Specify to use built-in algorithms
    output_path=bucket,
    base_job_name=base_job_name,

    # Parameters required to enable checkpointing
    checkpoint_s3_uri=checkpoint_s3_bucket,
    checkpoint_local_path=checkpoint_local_path
)
```

Dua parameter berikut menentukan jalur untuk checkpointing:

- `checkpoint_local_path`- Tentukan jalur lokal tempat model menyimpan pos pemeriksaan secara berkala dalam wadah pelatihan. Jalur default diatur ke`'/opt/ml/checkpoints'`. Jika Anda menggunakan kerangka kerja lain atau membawa wadah pelatihan Anda sendiri,

pastikan bahwa konfigurasi pos pemeriksaan skrip latihan Anda menentukan jalur ke. `'/opt/ml/checkpoints'`

> **Note**
>
> Sebaiknya tentukan jalur lokal `'/opt/ml/checkpoints'` agar konsisten dengan pengaturan SageMaker pos pemeriksaan default. Jika Anda lebih suka menentukan jalur lokal Anda sendiri, pastikan Anda mencocokkan jalur penyimpanan pos pemeriksaan dalam skrip latihan Anda dan `checkpoint_local_path` parameter SageMaker estimator.

- `checkpoint_s3_uri`- URI ke bucket S3 tempat pos pemeriksaan disimpan secara real time.

Untuk menemukan daftar lengkap parameter SageMaker estimator, lihat Estimator API dalam dokumentasi Amazon SageMaker Python SDK.

Jelajahi File Pos Pemeriksaan

Temukan file pos pemeriksaan menggunakan SageMaker Python SDK dan konsol Amazon S3.

Untuk menemukan file pos pemeriksaan secara terprogram

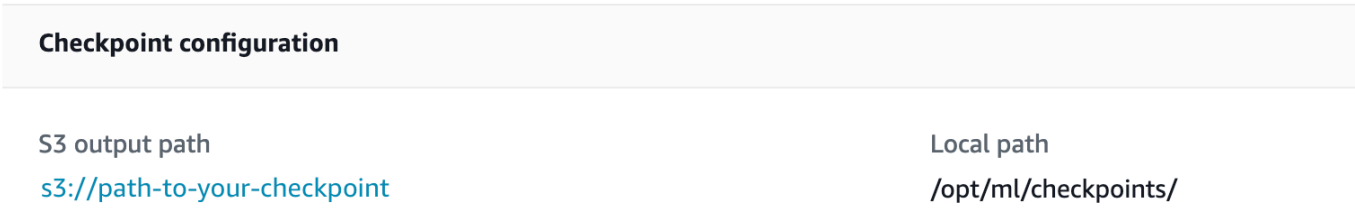
Untuk mengambil URI bucket S3 tempat pos pemeriksaan disimpan, periksa atribut estimator berikut:

```
estimator.checkpoint_s3_uri
```

Ini mengembalikan jalur keluaran Amazon S3 untuk pos pemeriksaan yang dikonfigurasi saat meminta permintaan. `CreateTrainingJob` Untuk menemukan file pos pemeriksaan yang disimpan menggunakan konsol Amazon S3, gunakan prosedur berikut.

Untuk menemukan file pos pemeriksaan dari konsol Amazon S3

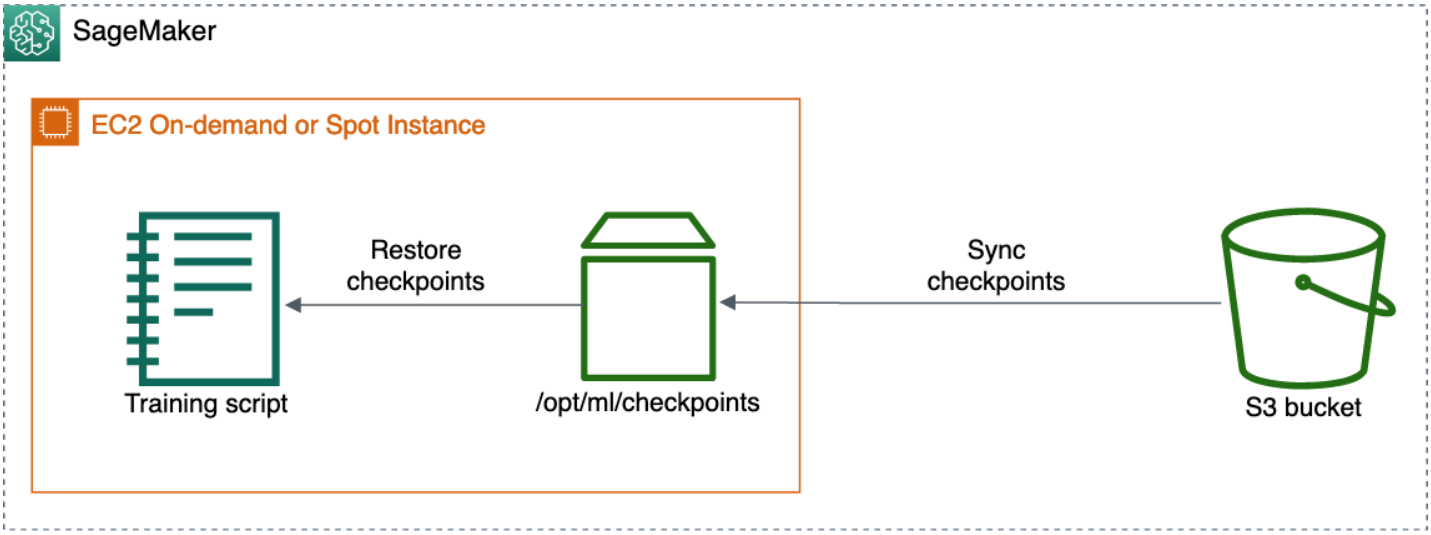
1. Masuk ke AWS Management Console dan buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Pekerjaan pelatihan.
3. Pilih tautan ke pekerjaan pelatihan dengan checkpointing diaktifkan untuk membuka Pengaturan pekerjaan.
4. Pada halaman Pengaturan pekerjaan pekerjaan pelatihan, cari bagian konfigurasi Checkpoint.



5. Gunakan tautan ke bucket S3 untuk mengakses file pos pemeriksaan.

Lanjutkan Pelatihan Dari Pos Pemeriksaan

Untuk melanjutkan pekerjaan pelatihan dari pos pemeriksaan, jalankan estimator baru dengan yang sama `checkpoint_s3_uri` yang Anda buat di bagianAktifkan Checkpointing. Setelah pelatihan dilanjutkan, pos pemeriksaan dari bucket S3 ini dikembalikan ke `checkpoint_local_path` dalam setiap instance pekerjaan pelatihan baru. Pastikan bahwa bucket S3 berada di Wilayah yang sama dengan SageMaker sesi saat ini.



Pertimbangan untuk Checkpointing

Pertimbangkan hal berikut saat menggunakan pos pemeriksaan diSageMaker.

- Untuk menghindari penimpaan dalam pelatihan terdistribusi dengan beberapa instance, Anda harus mengonfigurasi nama dan jalur file pos pemeriksaan secara manual dalam skrip latihan Anda. Konfigurasi SageMaker pos pemeriksaan tingkat tinggi menentukan lokasi Amazon S3

tunggal tanpa akhiran atau awalan tambahan untuk menandai pos pemeriksaan dari beberapa instans.

- SageMakerPython SDK tidak mendukung konfigurasi tingkat tinggi untuk frekuensi checkpointing. Untuk mengontrol frekuensi checkpointing, ubah skrip latihan Anda menggunakan fungsi penyimpanan model kerangka kerja atau callback pos pemeriksaan.
- Jika Anda menggunakan SageMaker pos pemeriksaan dengan SageMaker Debugger dan SageMaker didistribusikan dan menghadapi masalah, lihat halaman berikut untuk pemecahan masalah dan pertimbangan.
  - Pertimbangan untuk Amazon SageMaker Debugger
  - Pemecahan masalah untuk pelatihan terdistribusi di Amazon SageMaker
  - Pemecahan Masalah Paralel Model

Menyebarkan model untuk inferensi

Dengan Amazon SageMaker, Anda dapat menerapkan model pembelajaran mesin (ML) untuk membuat prediksi, juga dikenal sebagai inferensi. SageMaker menyediakan berbagai pilihan infrastruktur dan opsi penerapan model ML untuk membantu memenuhi semua kebutuhan inferensi ML Anda. Ini adalah layanan yang dikelola sepenuhnya dan terintegrasi dengan alat MLOP, sehingga Anda dapat menskalakan penerapan model Anda, mengurangi biaya inferensi, mengelola model secara lebih efektif dalam produksi, dan mengurangi beban operasional.

Setelah Anda membangun dan melatih model pembelajaran mesin, Anda dapat menggunakan SageMaker Inferensi untuk mulai mendapatkan prediksi, atau kesimpulan, dari model Anda. Dengan SageMaker Inferensi, Anda dapat mengatur titik akhir yang mengembalikan inferensi atau menjalankan inferensi batch dari model Anda.

Untuk memulai SageMaker Inferensi, lihat bagian berikut dan tinjau Opsi inferensi untuk menentukan fitur mana yang paling sesuai dengan kasus penggunaan Anda.

Anda dapat merujuk ke Sumber daya bagian untuk lebih banyak pemecahan masalah dan informasi referensi, blog dan contoh untuk membantu Anda memulai, dan FAQ umum.

Topik

Sebelum Anda memulai

Topik-topik ini mengasumsikan bahwa Anda telah membangun dan melatih satu atau lebih model pembelajaran mesin dan siap untuk menerapkannya. Anda tidak perlu melatih model Anda untuk menerapkan model Anda SageMaker dan mendapatkan kesimpulan. SageMaker Jika Anda tidak memiliki model sendiri, Anda juga dapat menggunakan SageMaker algoritme bawaan atau model yang telah dilatih sebelumnya.

Jika Anda baru SageMaker dan belum memilih model untuk diterapkan, kerjakan langkah-langkah dalam SageMaker tutorial Memulai dengan Amazon untuk membiasakan diri dengan contoh bagaimana SageMaker mengelola proses ilmu data dan cara menangani penerapan model. Untuk informasi lebih lanjut tentang melatih model, lihat Model Kereta.

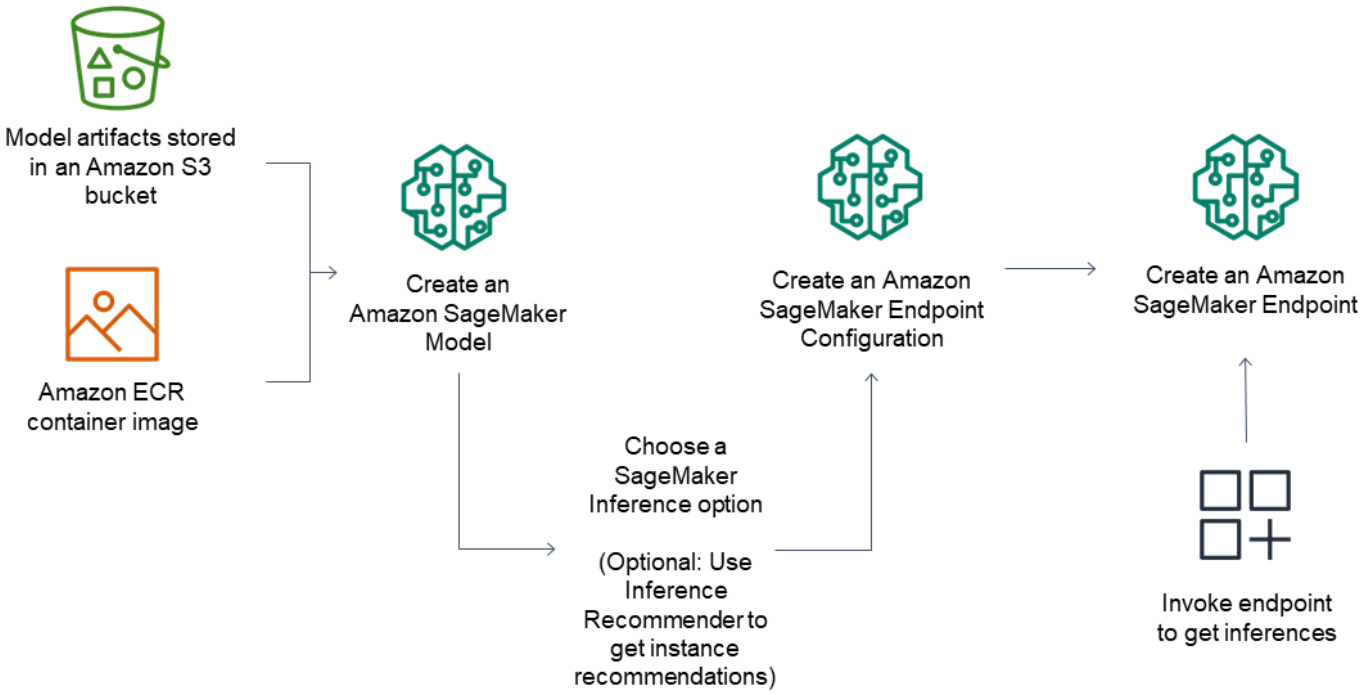
Untuk informasi tambahan, referensi, dan contoh, lihatSumber daya.

Langkah-langkah untuk penerapan model

Untuk titik akhir inferensi, alur kerja umum terdiri dari yang berikut:

- Buat model di SageMaker Inferensi dengan menunjuk ke artefak model yang disimpan di Amazon S3 dan gambar kontainer.
- Pilih opsi inferensi. Untuk informasi selengkapnya, lihat Opsi inferensi.
- Buat konfigurasi titik akhir SageMaker Inferensi dengan memilih jenis instance dan jumlah instance yang Anda butuhkan di belakang titik akhir. Anda dapat menggunakan Amazon SageMaker Inference Recommender untuk mendapatkan rekomendasi untuk jenis instans. Untuk Inferensi Tanpa Server, Anda hanya perlu menyediakan konfigurasi memori yang Anda butuhkan berdasarkan ukuran model Anda.
- Buat titik akhir SageMaker Inferensi.
- Panggil titik akhir Anda untuk menerima inferensi sebagai tanggapan.

Diagram berikut menunjukkan alur kerja sebelumnya.



Anda dapat melakukan tindakan ini menggunakan AWS konsol, AWS SDK, SageMaker Python SDKAWS CloudFormation, atau. AWS CLI

Untuk inferensi batch dengan transformasi batch, arahkan ke artefak model Anda dan masukan data dan buat pekerjaan inferensi batch. Alih-alih menghosting titik akhir untuk inferensi, keluarkan kesimpulan SageMaker Anda ke lokasi Amazon S3 pilihan Anda.

Opsi inferensi

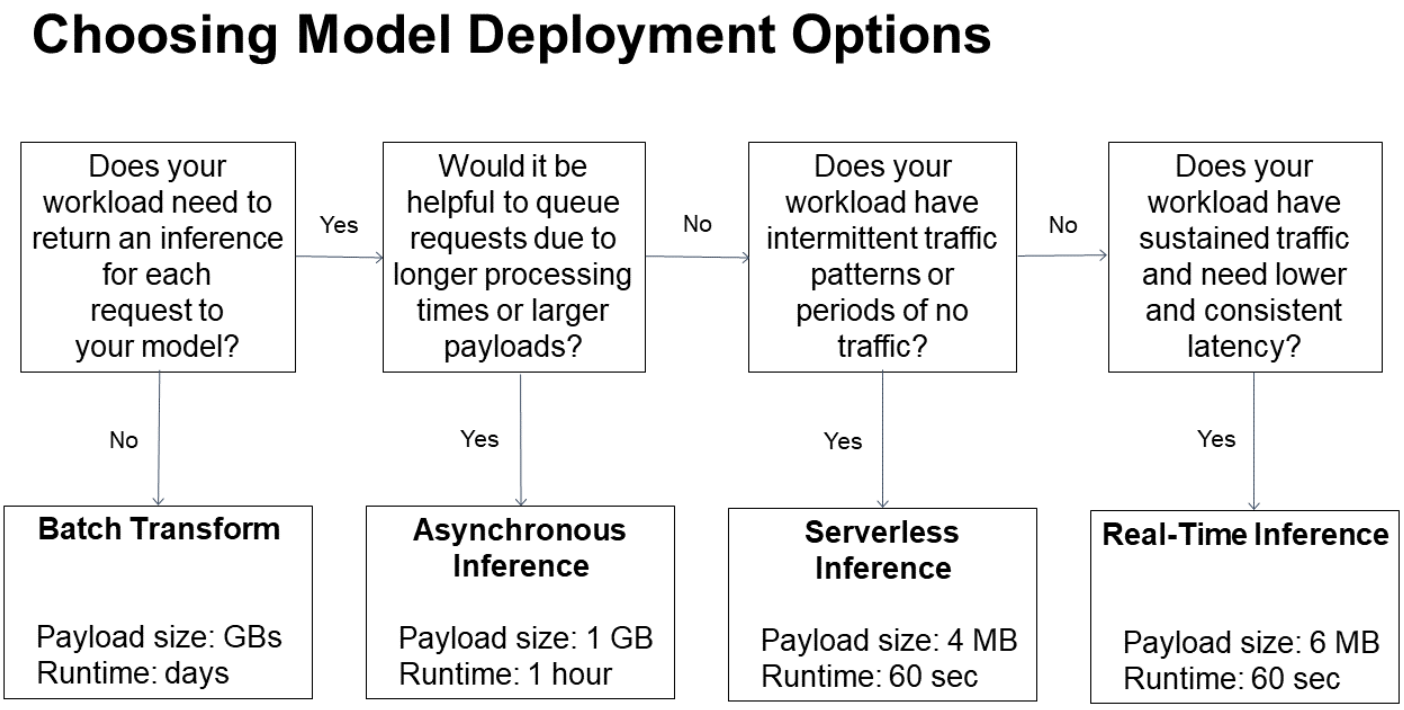
SageMaker menyediakan beberapa opsi inferensi sehingga Anda dapat memilih opsi yang paling sesuai dengan beban kerja Anda:

- Inferensi Real-Time: Inferensi real-time sangat ideal untuk inferensi online yang memiliki latensi rendah atau persyaratan throughput tinggi. Gunakan inferensi real-time untuk endpoint persisten dan terkelola sepenuhnya (REST API) yang dapat menangani lalu lintas berkelanjutan, didukung oleh jenis instans pilihan Anda. Inferensi waktu nyata dapat mendukung ukuran muatan hingga 6 MB dan waktu pemrosesan 60 detik.
- Inferensi Tanpa Server: Inferensi tanpa server sangat ideal ketika Anda memiliki pola lalu lintas intermiten atau tidak dapat diprediksi. SageMaker mengelola semua infrastruktur yang mendasarinya, jadi tidak perlu mengelola instance atau kebijakan penskalaan. Anda hanya

membayar untuk apa yang Anda gunakan dan bukan untuk waktu idle. Ini dapat mendukung ukuran muatan hingga 4 MB dan waktu pemrosesan hingga 60 detik.

- Transformasi Batch: Transformasi Batch cocok untuk pemrosesan offline ketika sejumlah besar data tersedia di muka dan Anda tidak memerlukan titik akhir yang persisten. Anda juga dapat menggunakan transformasi batch untuk kumpulan data pra-pemrosesan. Ini dapat mendukung kumpulan data besar yang berukuran GB dan waktu pemrosesan dalam beberapa hari.
- Inferensi Asinkron: Inferensi asinkron sangat ideal ketika Anda ingin mengantri permintaan dan memiliki muatan besar dengan waktu pemrosesan yang lama. Inferensi Asinkron dapat mendukung muatan hingga 1 GB dan waktu pemrosesan yang lama hingga satu jam. Anda juga dapat menurunkan titik akhir Anda ke 0 ketika tidak ada permintaan untuk diproses.

Diagram berikut menunjukkan informasi sebelumnya dalam diagram alur dan dapat membantu Anda memilih opsi yang paling sesuai dengan kasus penggunaan Anda.



Opsi titik akhir lanjutan

Dengan inferensi real-time, Anda dapat lebih mengoptimalkan kinerja dan biaya dengan opsi inferensi lanjutan berikut:

- Jika Anda memiliki beberapa model yang menggunakan kerangka kerja yang sama dan dapat berbagi wadah, gunakanHost beberapa model dalam satu wadah di belakang satu titik akhir. Opsi ini membantu Anda mengoptimalkan biaya dengan meningkatkan pemanfaatan titik akhir dan mengurangi overhead penerapan.
- Jika Anda memiliki beberapa model yang menggunakan kerangka kerja yang berbeda dan memerlukan wadah mereka sendiri, maka gunakanHost beberapa model yang menggunakan wadah berbeda di belakang satu titik akhir. Dengan opsi ini, Anda mendapatkan banyak manfaat dari Titik Akhir Multi-Model dan dapat menerapkan berbagai kerangka kerja dan model.
- Jika Anda ingin meng-host model dengan logika pra-pemrosesan dan pasca-pemrosesan di belakang titik akhir, gunakan Serial Inference Pipelines. Saluran pipa inferensi dikelola sepenuhnya oleh SageMaker dan memberikan latensi yang lebih rendah karena semua kontainer di-host pada instans Amazon EC2 yang sama.

Bawa model Anda sendiri

Untuk menggunakan wadah Docker yang ada di SageMaker, lihatMengadaptasi wadah Docker Anda sendiri untuk bekerja dengan SageMaker.

Untuk membuat wadah Docker baru dan menerima panduan lebih lanjut tentang cara menjalankan kode inferensi Anda sendiri, lihat tautan berikut.

- Untuk menjalankan layanan hosting kode inferensi Anda sendiri, lihatGunakan Kode Inferensi Anda Sendiri dengan Layanan Hosting.
- Untuk menjalankan kode inferensi Anda sendiri untuk inferensi batch, lihat. Gunakan Kode Inferensi Anda Sendiri dengan Batch Transform

Langkah selanjutnya

Setelah Anda memiliki titik akhir dan memahami alur kerja inferensi umum, Anda dapat menggunakan fitur berikut dalam SageMaker Inferensi untuk meningkatkan alur kerja inferensi Anda.

Pemantauan

Untuk melacak model Anda dari waktu ke waktu melalui metrik seperti akurasi model dan penyimpangan, Anda dapat menggunakan Model Monitor. Dengan Model Monitor, Anda dapat mengatur peringatan yang memberi tahu Anda ketika ada penyimpangan dalam kualitas model Anda.

Untuk mempelajari lebih lanjut, lihat dokumentasi Model Monitor. Untuk mempelajari lebih lanjut tentang alat yang dapat digunakan untuk memantau penerapan model dan peristiwa yang mengubah titik akhir Anda, lihat Memantau Amazon. SageMaker Misalnya, Anda dapat memantau kesehatan titik akhir Anda melalui metrik seperti kesalahan pemanggilan dan latensi model menggunakan metrik Amazon. CloudWatch Metrik pemanggilan SageMaker titik akhir dapat memberi Anda informasi berharga tentang kinerja titik akhir Anda.

CI/CD untuk penerapan model

Untuk mengumpulkan solusi pembelajaran mesin SageMaker, Anda dapat menggunakan SageMakerMLOP. Anda dapat menggunakan fitur ini untuk mengotomatiskan langkah-langkah dalam alur kerja pembelajaran mesin Anda dan berlatih CI/CD. Anda dapat menggunakan MLOPs Project Templates untuk membantu penyiapan dan implementasi proyek SageMaker MLOPs. SageMaker juga mendukung penggunaan repo Git pihak ketiga Anda sendiri untuk membuat sistem CI/CD.

Untuk pipeline ML Anda, gunakan Model Registry untuk mengelola versi model serta penerapan serta otomatisasi model Anda.

Pagar pembatas penyebaran

Jika Anda ingin memperbarui model Anda saat sedang dalam produksi tanpa memengaruhi produksi, Anda dapat menggunakan pagar pembatas penerapan. Pagar pembatas penerapan adalah serangkaian opsi penerapan model di SageMaker Inferensi untuk memperbarui model pembelajaran mesin Anda dalam produksi. Dengan menggunakan opsi penerapan yang dikelola sepenuhnya, Anda dapat mengontrol sakelar dari model saat ini dalam produksi ke yang baru. Mode perpindahan lalu lintas memberi Anda kontrol terperinci atas proses perpindahan lalu lintas, dan perlindungan bawaan seperti rollback otomatis membantu Anda menangkap masalah sejak dini. Untuk mempelajari selengkapnya tentang pagar pembatas penerapan, lihat dokumentasi pagar pembatas penerapan.

Inferensia

Jika Anda perlu menjalankan pembelajaran mesin skala besar dan aplikasi pembelajaran mendalam untuk kasus penggunaan seperti pengenalan gambar atau ucapan, pemrosesan bahasa alami (NLP), personalisasi, peramalan, atau deteksi penipuan, Anda dapat menggunakan `Inf1` instance dengan titik akhir waktu nyata.

`Inf1`instance dibangun untuk mendukung aplikasi inferensi pembelajaran mesin dan menampilkan chip AWS Inferentia. `Inf1`Instans memberikan throughput yang lebih tinggi dan biaya per inferensi yang lebih rendah daripada instans berbasis GPU.

Untuk menerapkan model pada `Inf1` instance, kompilasi model Anda dengan SageMaker Neo dan pilih `Inf1` instance untuk opsi penerapan Anda. Untuk mempelajari lebih lanjut, lihat Mengoptimalkan kinerja model menggunakan SageMaker Neo.

Optimalkan kinerja model

SageMaker menyediakan fitur untuk mengelola sumber daya dan mengoptimalkan kinerja inferensi saat menerapkan model pembelajaran mesin. Anda dapat menggunakan SageMaker algoritme bawaan dan model pra-bangun, serta gambar Docker bawaan, yang dikembangkan untuk pembelajaran mesin. Untuk melatih TensorFlow, Apache MXNet,, ONNX PyTorch, dan model XGBoost sekali dan mengoptimalkannya untuk diterapkan pada prosesor ARM, Intel, dan Nvidia, lihat Optimalkan kinerja model menggunakan Neo. SageMaker

Penskalaan otomatis

Jika Anda memiliki jumlah lalu lintas yang bervariasi ke titik akhir Anda, Anda mungkin ingin mencoba penskalaan otomatis. Misalnya, selama jam sibuk, Anda mungkin memerlukan lebih banyak contoh untuk memproses permintaan, tetapi selama periode lalu lintas rendah, Anda mungkin ingin mengurangi penggunaan sumber daya komputasi. Untuk menyesuaikan jumlah instans yang disediakan secara dinamis sebagai respons terhadap perubahan beban kerja Anda, lihat. Secara Otomatis Menskalakan SageMaker Model Amazon

Jika Anda memiliki pola lalu lintas yang tidak dapat diprediksi atau tidak ingin menyiapkan kebijakan penskalaan, Anda juga dapat menggunakan Inferensi Tanpa Server untuk titik akhir tempat mengelola penskalaan otomatis untuk Anda. SageMaker Selama periode lalu lintas rendah, SageMaker turunkan titik akhir Anda, dan jika lalu lintas meningkat, maka SageMaker tingkatkan titik akhir Anda. Untuk informasi lebih lanjut, lihat Inferensi Tanpa Server dokumentasi.

Menerapkan Model di Amazon SageMaker

Setelah melatih model machine learning, Anda dapat menerapkannya menggunakan Amazon SageMaker untuk mendapatkan prediksi dengan salah satu cara-cara berikut ini, tergantung pada kasus penggunaan Anda:

- Untuk endpoint yang persisten dan real-time yang membuat satu prediksi pada satu waktu, gunakan SageMakerlayanan hosting waktu nyata. Lihat Inferensi waktu nyata.
- Beban kerja yang memiliki periode idle antara semburan lalu lintas dan dapat mentolerir mulai dingin, menggunakan Inferensi Tanpa Server. Lihat Inferensi Tanpa Server.

- Permintaan dengan ukuran muatan besar hingga 1 GB, waktu pemrosesan yang lama, dan persyaratan latensi hampir waktu nyata, gunakan Amazon SageMaker Inferensi Asynchronous. Lihat Inferensi asinkron.
- Untuk mendapatkan prediksi untuk seluruh kumpulan data, gunakan SageMaker Transformasi batch. Lihat Gunakan Batch Transform.

SageMaker juga menyediakan fitur untuk mengelola sumber daya dan mengoptimalkan kinerja inferensi saat menerapkan model machine learning:

- Untuk mengelola model pada perangkat edge sehingga Anda dapat mengoptimalkan, mengamankan, memantau, dan memelihara model machine learning pada armada perangkat edge seperti kamera pintar, robot, komputer pribadi, dan perangkat seluler, lihatTerapkan model di tepi dengan SageMaker Edge Manager.
- Untuk mengoptimalkan Gluon, Keras, MXNet, PyTorch, TensorFlow, TensorFlow-Lite, dan model ONNX untuk inferensi pada mesin Android, Linux, dan Windows berdasarkan prosesor dari Ambarella, ARM, Intel, Nvidia, NXP, Qualcomm, Texas Instruments, dan Xilinx, lihatOptimalkan kinerja model menggunakan Neo.

Untuk informasi selengkapnya tentang semua opsi deployment, lihatMenyebarkan model untuk inferensi.

Buat model di Amazon SageMaker dengan ModelBuilder

Mempersiapkan model Anda untuk penerapan pada SageMaker titik akhir memerlukan beberapa langkah, termasuk memilih gambar model, menyiapkan konfigurasi titik akhir, mengkodekan fungsi serialisasi dan deserialisasi Anda untuk mentransfer data ke dan dari server dan klien, mengidentifikasi dependensi model, dan mengunggahnya ke Amazon S3. `ModelBuilder`dapat mengurangi kompleksitas penyiapan dan penerapan awal untuk membantu Anda membuat model yang dapat diterapkan dalam satu langkah.

`ModelBuilder`melakukan tugas-tugas berikut untuk Anda:

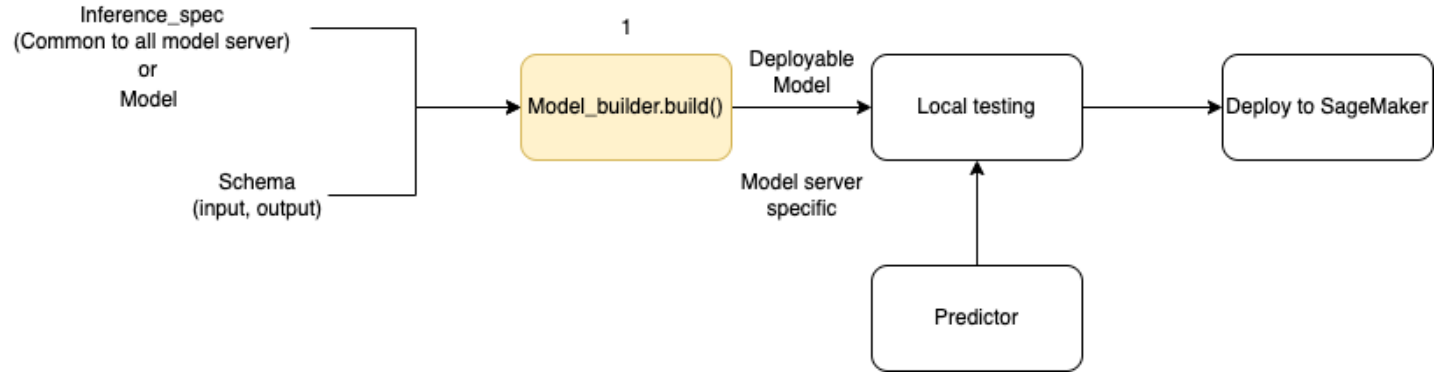
- Mengonversi model pembelajaran mesin yang dilatih menggunakan berbagai kerangka kerja seperti XGBoost atau PyTorch menjadi model yang dapat diterapkan dalam satu langkah.

- Melakukan pemilihan kontainer otomatis berdasarkan kerangka model sehingga Anda tidak perlu menentukan penampung secara manual. Anda masih dapat membawa kontainer Anda sendiri dengan meneruskan URI Anda sendiri ke`ModelBuilder`.
- Menangani serialisasi data di sisi klien sebelum mengirimnya ke server untuk inferensi dan deserialisasi hasil yang dikembalikan oleh server. Data diformat dengan benar tanpa pemrosesan manual.
- Memungkinkan pengambilan dependensi secara otomatis dan mengemas model sesuai dengan harapan server model. `ModelBuilder`penangkapan dependensi otomatis adalah pendekatan upaya terbaik untuk memuat dependensi secara dinamis. (Kami menyarankan Anda menguji pengambilan otomatis secara lokal dan memperbarui dependensi untuk memenuhi kebutuhan Anda.)
- Untuk kasus penggunaan model bahasa besar (LLM), secara opsional melakukan penyetelan parameter lokal dari properti penyajian yang dapat digunakan untuk kinerja yang lebih baik saat menghosting di titik akhir. SageMaker
- Mendukung sebagian besar server model populer dan kontainer seperti TorchServe, Triton, DJLServing dan TGI container.

Membangun model Anda dengan ModelBuilder

`ModelBuilder`adalah kelas Python yang mengambil model kerangka kerja, seperti XGBoost atau PyTorch, atau spesifikasi inferensi yang ditentukan pengguna dan mengubahnya menjadi model deployable. `ModelBuilder`menyediakan fungsi build yang menghasilkan artefak untuk penerapan. Artefak model yang dihasilkan khusus untuk server model, yang juga dapat Anda tentukan sebagai salah satu input. Untuk detail lebih lanjut tentang `ModelBuilder` kelas, lihat ModelBuilder.

Diagram berikut menggambarkan keseluruhan alur kerja pembuatan model saat Anda menggunakan. `ModelBuilder` `ModelBuilder`menerima spesifikasi model atau inferensi bersama dengan skema Anda untuk membuat model yang dapat diterapkan yang dapat Anda uji secara lokal sebelum penerapan.



`ModelBuilder`dapat menangani kustomisasi apa pun yang ingin Anda terapkan. Namun, untuk menerapkan model kerangka kerja, pembuat model mengharapkan setidaknya model, input dan output sampel, dan peran. Dalam contoh kode berikut, `ModelBuilder` dipanggil dengan model kerangka kerja dan instance `SchemaBuilder` dengan argumen minimum (untuk menyimpulkan fungsi yang sesuai untuk serialisasi dan deserialisasi input dan output endpoint). Tidak ada kontainer yang ditentukan dan tidak ada dependensi paket yang diteruskan— SageMaker secara otomatis menyimpulkan sumber daya ini saat Anda membuat model Anda.

```
from sagemaker.serve.builder.model_builder import ModelBuilder
from sagemaker.serve.builder.schema_builder import SchemaBuilder

model_builder = ModelBuilder(
    model=model,
    schema_builder=SchemaBuilder(input, output),
    role_arn="execution-role",
)
```

Contoh kode berikut dipanggil `ModelBuilder` dengan spesifikasi inferensi (sebagai `InferenceSpec` contoh) alih-alih model, dengan penyesuaian tambahan. Dalam hal ini, panggilan ke pembuat model menyertakan jalur untuk menyimpan artefak model dan juga mengaktifkan penangkapan otomatis semua dependensi yang tersedia. Untuk detail tambahan tentang`InferenceSpec`, lihat[Sesuaikan pemuatan model dan penanganan permintaan](#).

```
model_builder = ModelBuilder(
    mode=Mode.LOCAL_CONTAINER,
    model_path=model-artifact-directory,
    inference_spec=your-inference-spec,
    schema_builder=SchemaBuilder(input, output),
    role_arn=execution-role,
    dependencies={"auto": True}
```

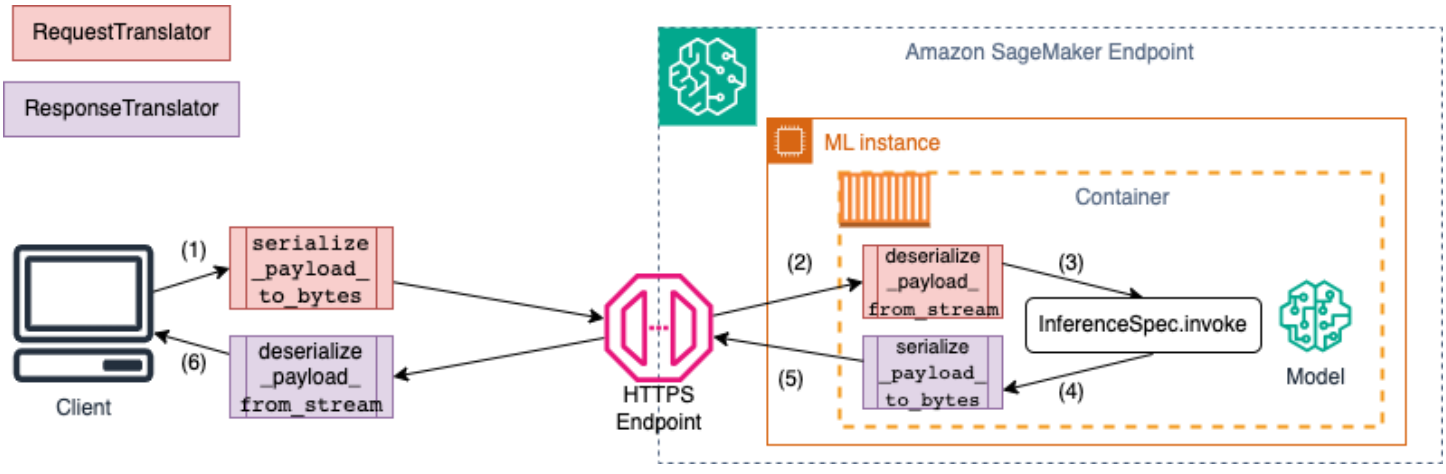
```
)
```

Tentukan metode serialisasi dan deserialisasi

Saat menjalankan SageMaker endpoint, data dikirim melalui payload HTTP dengan tipe MIME yang berbeda. Misalnya, gambar yang dikirim ke titik akhir untuk inferensi perlu dikonversi ke byte di sisi klien dan dikirim melalui payload HTTP ke titik akhir. Ketika titik akhir menerima payload, ia perlu deserialisasi string byte kembali ke tipe data yang diharapkan oleh model (juga dikenal sebagai deserialisasi sisi server). Setelah model menyelesaikan prediksi, hasilnya juga perlu diserialisasikan ke byte yang dapat dikirim kembali melalui payload HTTP ke pengguna atau klien. Setelah klien menerima data byte respons, klien perlu melakukan deserialisasi sisi klien untuk mengonversi data byte kembali ke format data yang diharapkan, seperti JSON. Minimal, Anda perlu mengonversi data untuk tugas-tugas berikut:

1. Serialisasi permintaan inferensi (ditangani oleh klien)
2. Deserialisasi permintaan inferensi (ditangani oleh server atau algoritma)
3. Memanggil model terhadap muatan dan mengirim muatan respons kembali
4. Serialisasi respons inferensi (ditangani oleh server atau algoritma)
5. Deserialisasi respons inferensi (ditangani oleh klien)

Diagram berikut menunjukkan proses serialisasi dan deserialisasi yang terjadi saat Anda memanggil titik akhir.



Saat Anda memasok input dan output sampel ke`SchemaBuilder`, pembuat skema menghasilkan fungsi penyusunan yang sesuai untuk membuat serial dan deserialisasi input dan output. Anda dapat

lebih lanjut menyesuaikan fungsi serialisasi Anda dengan`CustomPayloadTranslator`. Tetapi untuk kebanyakan kasus, serializer sederhana seperti berikut ini akan berfungsi:

```
input = "How is the demo going?"
output = "Comment la démo va-t-elle?"
schema = SchemaBuilder(input, output)
```

Untuk detail lebih lanjut tentang`SchemaBuilder`, lihat SchemaBuilder.

Cuplikan kode berikut menguraikan contoh di mana Anda ingin menyesuaikan fungsi serialisasi dan deserialisasi di sisi klien dan server. Anda dapat menentukan penerjemah permintaan dan tanggapan Anda sendiri `CustomPayloadTranslator` dan meneruskan penerjemah ini ke. `SchemaBuilder`

Dengan memasukkan input dan output dengan penerjemah, pembuat model dapat mengekstrak format data yang diharapkan model. Misalnya, input sampel adalah gambar mentah, dan penerjemah khusus Anda memotong gambar dan mengirim gambar yang dipotong ke server sebagai tensor. `ModelBuilder`membutuhkan input mentah dan kode pra-pemrosesan atau pasca-pemrosesan khusus untuk memperoleh metode untuk mengonversi data di sisi klien dan server.

```
from sagemaker.serve import CustomPayloadTranslator

# request translator
class MyRequestTranslator(CustomPayloadTranslator):
    # This function converts the payload to bytes - happens on client side
    def serialize_payload_to_bytes(self, payload: object) -> bytes:
        # converts the input payload to bytes
        ... ...
        return  //return object as bytes

    # This function converts the bytes to payload - happens on server side
    def deserialize_payload_from_stream(self, stream) -> object:
        # convert bytes to in-memory object
        ... ...
        return //return in-memory object

# response translator
class MyResponseTranslator(CustomPayloadTranslator):
    # This function converts the payload to bytes - happens on server side
    def serialize_payload_to_bytes(self, payload: object) -> bytes:
        # converts the response payload to bytes
        ... ...
        return //return object as bytes
```

```
    # This function converts the bytes to payload - happens on client side
    def deserialize_payload_from_stream(self, stream) -> object:
        # convert bytes to in-memory object
        ... ...
        return //return in-memory object
```

Anda meneruskan input dan output sampel bersama dengan penerjemah kustom yang ditentukan sebelumnya saat Anda membuat `SchemaBuilder` objek, seperti yang ditunjukkan pada contoh berikut:

```
my_schema = SchemaBuilder(
    sample_input=image,
    sample_output=output,
    input_translator=MyRequestTranslator(),
    output_translator=MyResponseTranslator()
)
```

Kemudian Anda meneruskan input dan output sampel, bersama dengan penerjemah khusus yang ditentukan sebelumnya, ke objek. `SchemaBuilder`

```
my_schema = SchemaBuilder(
    sample_input=image,
    sample_output=output,
    input_translator=MyRequestTranslator(),
    output_translator=MyResponseTranslator()
)
```

Bagian berikut menjelaskan secara rinci cara membangun model Anda `ModelBuilder` dan menggunakan kelas pendukungnya untuk menyesuaikan pengalaman untuk kasus penggunaan Anda.

Topik

- Sesuaikan pemuatan model dan penanganan permintaan
- Bangun model Anda dan terapkan
- Bawa wadah Anda sendiri (BYOC)
- Menggunakan ModelBuilder dalam mode lokal
- ModelBuilder contoh

Sesuaikan pemuatan model dan penanganan permintaan

Memberikan kode inferensi Anda sendiri melalui `InferenceSpec` menawarkan lapisan penyesuaian tambahan. Dengan`InferenceSpec`, Anda dapat menyesuaikan cara model dimuat dan cara menangani permintaan inferensi yang masuk, melewati pemuatan default dan mekanisme penanganan inferensi. Fleksibilitas ini sangat bermanfaat ketika bekerja dengan model non-standar atau pipa inferensi khusus. Anda dapat menyesuaikan `invoke` metode untuk mengontrol bagaimana model memproses dan pasca-proses permintaan masuk. `invoke`Metode ini memastikan bahwa model menangani permintaan inferensi dengan benar. Contoh berikut digunakan `InferenceSpec` untuk menghasilkan model dengan HuggingFace pipa. Untuk detail lebih lanjut tentang`InferenceSpec`, lihat InferenceSpec.

```
from sagemaker.serve.spec.inference_spec import InferenceSpec
from transformers import pipeline

class MyInferenceSpec(InferenceSpec):
    def load(self, model_dir: str):
        return pipeline("translation_en_to_fr", model="t5-small")

    def invoke(self, input, model):
        return model(input)

inf_spec = MyInferenceSpec()

model_builder = ModelBuilder(
    inference_spec=your-inference-spec,
    schema_builder=SchemaBuilder(X_test, y_pred)
)
```

Contoh berikut menggambarkan variasi yang lebih disesuaikan dari contoh sebelumnya. Sebuah model didefinisikan dengan spesifikasi inferensi yang memiliki dependensi. Dalam hal ini, kode dalam spesifikasi inferensi tergantung pada paket lang-segment. Argumen untuk `dependencies` berisi pernyataan yang mengarahkan pembangun untuk menginstal lang-segment menggunakan Git. Karena pembuat model diarahkan oleh pengguna untuk menginstal dependensi secara kustom, `auto` kuncinya adalah mematikan `False` penangkapan otomatis dependensi.

```
model_builder = ModelBuilder(
    mode=Mode.LOCAL_CONTAINER,
    model_path=model-artifact-directory,
    inference_spec=your-inference-spec,
```

```
    schema_builder=SchemaBuilder(input, output),
    role_arn=execution-role,
    dependencies={"auto": False, "custom": ["-e git+https://github.com/luca-medeiros/
lang-segment-anything.git#egg=lang-sam"],}
)
```

Bangun model Anda dan terapkan

Panggil `build` fungsi untuk membuat model deployable Anda. Langkah ini membuat kode inferensi (as`inference.py`) di direktori kerja Anda dengan kode yang diperlukan untuk membuat skema Anda, menjalankan serialisasi dan deserialisasi input dan output, dan menjalankan logika kustom yang ditentukan pengguna lainnya.

Sebagai pemeriksaan integritas, SageMaker mengemas dan mengamankan file yang diperlukan untuk penerapan sebagai bagian dari fungsi `ModelBuilder` build. Selama proses ini, SageMaker juga membuat penandatanganan HMAC untuk file pickle dan menambahkan kunci rahasia di CreateModelAPI sebagai variabel lingkungan selama `deploy` (ata`create`). Peluncuran endpoint menggunakan variabel lingkungan untuk memvalidasi integritas file pickle.

```
# Build the model according to the model server specification and save it as files in
 the working directory
model = model_builder.build()
```

Terapkan model Anda dengan `deploy` metode model yang ada. Pada langkah ini, SageMaker siapkan titik akhir untuk meng-host model Anda saat model mulai membuat prediksi pada permintaan yang masuk. Meskipun `ModelBuilder` menyimpulkan sumber daya titik akhir yang diperlukan untuk menerapkan model Anda, Anda dapat mengganti perkiraan tersebut dengan nilai parameter Anda sendiri. Contoh berikut mengarahkan SageMaker untuk menyebarkan model pada satu `ml.c6i.xlarge` contoh. Model yang dibuat dari `ModelBuilder` memungkinkan pencatatan langsung selama penerapan sebagai fitur tambahan.

```
predictor = model.deploy(
    initial_instance_count=1,
    instance_type="ml.c6i.xlarge"
)
```

Jika Anda menginginkan kontrol yang lebih halus atas sumber daya titik akhir yang ditetapkan ke model Anda, Anda dapat menggunakan objek. `ResourceRequirements` Dengan `ResourceRequirements` objek, Anda dapat meminta jumlah minimum CPU, akselerator, dan

salinan model yang ingin Anda terapkan. Anda juga dapat meminta batas memori minimum dan maksimum (dalam MB). Untuk menggunakan fitur ini, Anda perlu menentukan jenis titik akhir Anda sebagai`EndpointType.INFERENCE_COMPONENT_BASED`. Contoh berikut meminta empat akselerator, ukuran memori minimum 1024 MB, dan satu salinan model Anda untuk digunakan ke titik akhir tipe. `EndpointType.INFERENCE_COMPONENT_BASED`

```
resource_requirements = ResourceRequirements(
    requests={
        "num_accelerators": 4,
        "memory": 1024,
        "copies": 1,
    },
    limits={},
)
predictor = model.deploy(
    mode=Mode.SAGEMAKER_ENDPOINT,
    endpoint_type=EndpointType.INFERENCE_COMPONENT_BASED,
    resources=resource_requirements,
    role="role"
)
```

Bawa wadah Anda sendiri (BYOC)

Jika Anda ingin membawa wadah Anda sendiri (diperpanjang dari SageMaker wadah), Anda juga dapat menentukan URI gambar seperti yang ditunjukkan pada contoh berikut. Anda juga perlu mengidentifikasi server model yang sesuai dengan gambar `ModelBuilder` untuk menghasilkan artefak khusus untuk server model.

```
model_builder = ModelBuilder(
    model=model,
    model_server=ModelServer.TORCHSERVE,
    schema_builder=SchemaBuilder(X_test, y_pred),
    image_uri="123123123123.dkr.ecr.ap-southeast-2.amazonaws.com/byoc-image:xgb-1.7-1")
)
```

Menggunakan ModelBuilder dalam mode lokal

Anda dapat menerapkan model Anda secara lokal dengan menggunakan `mode` argumen untuk beralih antara pengujian lokal dan penerapan ke titik akhir. Anda perlu menyimpan artefak model di direktori kerja, seperti yang ditunjukkan pada cuplikan berikut:

```
model = XGBClassifier()
model.fit(X_train, y_train)
model.save_model(model_dir + "/my_model.xgb")
```

Lewati objek model, `SchemaBuilder` instance, dan atur mode ke`Mode.LOCAL_CONTAINER`. Saat Anda memanggil `build` fungsi, `ModelBuilder` secara otomatis mengidentifikasi wadah kerangka kerja yang didukung dan memindai dependensi. Contoh berikut menunjukkan pembuatan model dengan model XGBoost dalam mode lokal.

```
model_builder_local = ModelBuilder(
    model=model,
    schema_builder=SchemaBuilder(X_test, y_pred),
    role_arn=execution-role,
    mode=Mode.LOCAL_CONTAINER
)
xgb_local_builder = model_builder_local.build()
```

Panggil `deploy` fungsi untuk menyebarkan secara lokal, seperti yang ditunjukkan pada cuplikan berikut. Jika Anda menentukan parameter untuk tipe atau hitungan contoh, argumen ini diabaikan.

```
predictor_local = xgb_local_builder.deploy()
```

Memecahkan masalah mode lokal

Tergantung pada pengaturan lokal pribadi Anda, Anda mungkin mengalami kesulitan berjalan `ModelBuilder` dengan lancar di lingkungan Anda. Lihat daftar berikut untuk beberapa masalah yang mungkin Anda hadapi dan cara mengatasinya.

- Sudah digunakan: Anda mungkin mengalami `Address already in use` kesalahan. Dalam hal ini, ada kemungkinan bahwa kontainer Docker berjalan pada port itu atau proses lain memanfaatkannya. Anda dapat mengikuti pendekatan yang diuraikan dalam dokumentasi Linux untuk mengidentifikasi proses dan dengan anggun mengarahkan proses lokal Anda dari port 8080 ke port lain atau membersihkan instance Docker.
- Masalah Izin IAM: Anda mungkin mengalami masalah izin saat mencoba menarik gambar Amazon ECR atau mengakses Amazon S3. Dalam hal ini, navigasikan ke peran eksekusi notebook atau instans Studio Classic untuk memverifikasi kebijakan `SageMakerFullAccess` atau izin API masing-masing.

- Masalah kapasitas volume EBS: Jika Anda menerapkan model bahasa besar (LLM), Anda mungkin kehabisan ruang saat menjalankan Docker dalam mode lokal atau mengalami batasan ruang untuk cache Docker. Dalam hal ini, Anda dapat mencoba memindahkan volume Docker Anda ke sistem file yang memiliki cukup ruang. Untuk memindahkan volume Docker Anda, selesaikan langkah-langkah berikut:

  1. Buka terminal dan jalankan `df` untuk menampilkan penggunaan disk, seperti yang ditunjukkan pada output berikut:

     ```
     (python3) sh-4.2$ df
     Filesystem     1K-blocks      Used Available Use% Mounted on
     devtmpfs       195928700         0 195928700   0% /dev
     tmpfs          195939296         0 195939296   0% /dev/shm
     tmpfs          195939296      1048 195938248   1% /run
     tmpfs          195939296         0 195939296   0% /sys/fs/cgroup
     /dev/nvme0n1p1 141545452 135242112   6303340  96% /
     tmpfs           39187860         0  39187860   0% /run/user/0
     /dev/nvme2n1   264055236  76594068 176644712  31% /home/ec2-user/SageMaker
     tmpfs           39187860         0  39187860   0% /run/user/1002
     tmpfs           39187860         0  39187860   0% /run/user/1001
     tmpfs           39187860         0  39187860   0% /run/user/1000
     ```

  2. Pindahkan direktori Docker default dari `/dev/nvme0n1p1` ke `/dev/nvme2n1` sehingga Anda dapat sepenuhnya memanfaatkan volume 256 GB SageMaker . Untuk detail selengkapnya, lihat dokumentasi tentang cara memindahkan direktori Docker Anda.

  3. Hentikan Docker dengan perintah berikut:

     ```
     sudo service docker stop
     ```

  4. Tambahkan `daemon.json` ke `/etc/docker` atau tambahkan gumpalan JSON berikut ke yang sudah ada.

     ```
     {
         "data-root": "/home/ec2-user/SageMaker/{created_docker_folder}"
     }
     ```

  5. Pindahkan direktori Docker `/var/lib/docker` ke `/home/ec2-user/SageMaker` dengan perintah berikut:

     ```
     sudo rsync -aP /var/lib/docker/ /home/ec2-user/SageMaker/{created_docker_folder}
     ```

6. Mulai Docker dengan perintah berikut:

```
sudo service docker start
```

7. Bersihkan sampah dengan perintah berikut:

```
cd /home/ec2-user/SageMaker/.Trash-1000/files/*
sudo rm -r *
```

8. Jika Anda menggunakan instance SageMaker notebook, Anda dapat mengikuti langkah-langkah dalam file persiapan Docker untuk mempersiapkan Docker untuk mode lokal.

ModelBuilder contoh

Untuk contoh penggunaan lainnya `ModelBuilder` untuk membuat model Anda, lihat ModelBuildercontoh buku catatan.

Memvalidasi Model Machine Learning

Setelah melatih model, evaluasi untuk menentukan apakah kinerja dan akurasinya memungkinkan Anda mencapai tujuan bisnis Anda. Anda mungkin menghasilkan beberapa model menggunakan metode yang berbeda dan mengevaluasi masing-masing. Misalnya, Anda dapat menerapkan aturan bisnis yang berbeda untuk setiap model, dan kemudian menerapkan berbagai langkah untuk menentukan kesesuaian masing-masing model. Anda mungkin mempertimbangkan apakah model Anda perlu lebih sensitif daripada spesifik (atau sebaliknya).

Anda dapat mengevaluasi model Anda menggunakan data historis (offline) atau data langsung:

- Pengujian offline—Gunakan historis, tidak hidup, data untuk mengirim permintaan ke model untuk kesimpulan.

  Terapkan model terlatih Anda ke titik akhir alfa, dan gunakan data historis untuk mengirim permintaan inferensi ke sana. Untuk mengirim permintaan, gunakan notebook Jupyter di Amazon SageMakercontoh notebook dan baikAWS SDK for Python (Boto)atau pustaka Python tingkat tinggi yang disediakan oleh SageMaker.
- Pengujian online dengan data langsung—SageMakermendukung pengujian A/B untuk model dalam produksi dengan menggunakan varian produksi. Varian produksi adalah model yang menggunakan kode inferensi yang sama dan digunakan pada yang sama SageMaker titik akhir.

Anda mengkonfigurasi varian produksi sehingga sebagian kecil dari lalu lintas langsung masuk ke model yang ingin Anda validasi. Misalnya, Anda dapat memilih untuk mengirim 10% lalu lintas ke varian model untuk evaluasi. Setelah Anda puas dengan kinerja model, Anda dapat merutekan lalu lintas 100% ke model yang diperbarui. Untuk contoh model pengujian dalam produksi, lihatVarian produksi.

Untuk informasi lebih lanjut, lihat artikel dan buku tentang cara mengevaluasi model, misalnya,Mengevaluasi Model Machine Learning.

Pilihan untuk evaluasi model offline meliputi:

- Memvalidasi menggunakan set holdout— Praktisi pembelajaran mesin sering menyisihkan sebagian data sebagai “set penangguhan”. Mereka tidak menggunakan data ini untuk pelatihan model.

  Dengan pendekatan ini, Anda mengevaluasi seberapa baik model Anda memberikan kesimpulan pada set holdout. Anda kemudian menilai seberapa efektif model menggeneralisasi apa yang dipelajarinya dalam pelatihan awal, sebagai lawan menggunakan memori model. Pendekatan validasi ini memberi Anda gambaran tentang seberapa sering model dapat menyimpulkan jawaban yang benar.

  Dalam beberapa hal, pendekatan ini mirip dengan mengajar siswa sekolah dasar. Pertama, Anda memberi mereka serangkaian contoh untuk dipelajari, dan kemudian menguji kemampuan mereka untuk menggeneralisasi dari pembelajaran mereka. Dengan pekerjaan rumah dan tes, Anda menimbulkan masalah yang tidak termasuk dalam pembelajaran awal dan menentukan apakah mereka dapat menggeneralisasi secara efektif. Siswa dengan kenangan sempurna bisa menghafal masalah, alih-alih mempelajari aturan.

  Biasanya, kumpulan data holdout adalah 20-30% dari data pelatihan.

- k-fold validasi—Dalam pendekatan validasi ini, Anda membagi contoh dataset menjadikbagian. Anda memperlakukan masing-masing bagian ini sebagai holdout set untukkpelatihan berjalan, dan menggunakan yang laink-1 bagian sebagai set pelatihan untuk lari itu. Anda menghasilkankmodel

menggunakan proses yang sama, dan agregat model untuk menghasilkan model akhir Anda. Nilaikbiasanya dalam kisaran 5-10.

Rekomendasi SageMaker Inferensi Amazon

Amazon SageMaker Inference Recommender adalah kemampuan Amazon SageMaker yang mengurangi waktu yang diperlukan untuk mendapatkan model machine learning (ML) dalam produksi dengan mengotomatiskan pengujian beban dan penyetelan model di seluruh instans ML. SageMaker Anda dapat menggunakan Inference Recommender untuk menyebarkan model Anda ke titik akhir inferensi real-time atau tanpa server yang memberikan kinerja terbaik dengan biaya terendah. Inference Recommender membantu Anda memilih jenis dan konfigurasi instans terbaik (seperti jumlah instans, parameter kontainer, dan pengoptimalan model) atau konfigurasi tanpa server (seperti konkurensi maks dan ukuran memori) untuk model dan beban kerja MLmu.

Amazon SageMaker Inference Recommender hanya menagih Anda untuk instans yang digunakan saat pekerjaan Anda dijalankan.

Cara kerjanya

Untuk menggunakan Amazon SageMaker Inference Recommender, Anda dapat membuat SageMaker model atau mendaftarkan model ke registri model dengan SageMaker artefak model Anda. Gunakan AWS SDK for Python (Boto3) atau SageMaker konsol untuk menjalankan tugas benchmarking untuk konfigurasi SageMaker titik akhir yang berbeda. Pekerjaan Inference Recommender membantu Anda mengumpulkan dan memvisualisasikan metrik di seluruh kinerja dan pemanfaatan sumber daya untuk membantu Anda memutuskan jenis dan konfigurasi titik akhir mana yang akan dipilih.

Cara Memulai

Jika Anda adalah pengguna pertama kali Amazon SageMaker Inference Recommender, kami sarankan Anda melakukan hal berikut:

1. Baca Prasyarat bagian untuk memastikan Anda telah memenuhi persyaratan untuk menggunakan Amazon SageMaker Inference Recommender.
2. Baca Lowongan kerja rekomendasi bagian untuk meluncurkan pekerjaan rekomendasi Inference Recommender pertama Anda.
3. Jelajahi contoh buku catatan pengantar Amazon SageMaker Inference Recommender Jupyter, atau tinjau contoh buku catatan di bagian berikut.

Notebook contoh

Contoh notebook Jupyter berikut dapat membantu Anda dengan alur kerja untuk beberapa kasus penggunaan di Inference Recommender:

- Jika Anda menginginkan buku catatan pengantar yang membandingkan TensorFlow model, lihat buku catatan SageMaker Inference Recommender. TensorFlow
- Jika Anda ingin melakukan benchmark HuggingFace model, lihat SageMaker Inference Recommender untuk notebook. HuggingFace
- Jika Anda ingin membandingkan model XGBoost, lihat notebook SageMaker Inference Recommender XGBoost.
- Jika Anda ingin meninjau CloudWatch metrik untuk pekerjaan Inference Recommender, lihat buku catatan metrik SageMaker Inference Recommender. CloudWatch

Prasyarat

Untuk menggunakan Amazon SageMaker Inference Recommender, pertama-tama pastikan Anda telah memenuhi prasyarat yang tercantum di bawah ini. Sebagai contoh, kami menunjukkan cara menggunakan model pra-terlatih PyTorch (v1.7.1) ResNet -18 untuk kedua jenis pekerjaan rekomendasi Rekomendasi SageMaker Inferensi Amazon. Contoh yang ditampilkan menggunakanAWS SDK for Python (Boto3).

> **Note**
>
> - Contoh kode berikut menggunakan Python. Hapus karakter ! awalan jika Anda menjalankan salah satu contoh kode berikut di terminal Anda atauAWS CLI.
> - Anda dapat menjalankan contoh berikut dengan kernel Python 3 (TensorFlow 2.6 Python 3.8 CPU Optimized) di notebook Amazon Studio. SageMaker Untuk informasi selengkapnya tentang Studio, lihat SageMaker Studio Amazon.

1. Buat peran IAM untuk Amazon SageMaker.

   Buat Peran IAM untuk Amazon SageMaker yang memiliki kebijakan terkelola `AmazonSageMakerFullAccess` IAM terlampir.
2. Siapkan lingkungan Anda.

Impor dependensi dan buat variabel untuk peran SageMaker IAM Anda (dari Langkah 1), dan klien. Wilayah AWS SageMaker

```
!pip install --upgrade pip awscli botocore boto3  --quiet
from sagemaker import get_execution_role, Session, image_uris
import boto3

region = boto3.Session().region_name
role = get_execution_role()
sagemaker_client = boto3.client("sagemaker", region_name=region)
sagemaker_session = Session()
```

3. (Opsional) Tinjau model yang ada yang dipatokan oleh Inference Recommender.

   Inferensi Rekomendasi tolok ukur model dari kebun binatang model populer. Inference Recommender mendukung model Anda meskipun belum benchmark.

   Gunakan `ListModelMetaData` untuk mendapatkan objek respons yang mencantumkan domain, kerangka kerja, tugas, dan nama model model pembelajaran mesin yang ditemukan di kebun binatang model umum.

   Anda menggunakan domain, kerangka kerja, versi kerangka kerja, tugas, dan nama model di langkah selanjutnya untuk memilih gambar Docker inferensi dan mendaftarkan model Anda dengan SageMaker Model Registry. Berikut ini menunjukkan cara membuat daftar metadata model dengan SDK for Python (Boto3):

   ```
   list_model_metadata_response=sagemaker_client.list_model_metadata()
   ```

   Outputnya mencakup ringkasan model (`ModelMetadataSummaries`) dan metadata respons (`ResponseMetadata`) yang mirip dengan yang berikut:

   ```
   {
       'ModelMetadataSummaries': [{
               'Domain': 'NATURAL_LANGUAGE_PROCESSING',
               'Framework': 'PYTORCH:1.6.0',
                'Model': 'bert-base-cased',
                'Task': 'FILL_MASK'
                },
               {
                'Domain': 'NATURAL_LANGUAGE_PROCESSING',
   ```

```
             'Framework': 'PYTORCH:1.6.0',
             'Model': 'bert-base-uncased',
             'Task': 'FILL_MASK'
             },
            {
            'Domain': 'COMPUTER_VISION',
             'Framework': 'MXNET:1.8.0',
             'Model': 'resnet18v2-gluon',
             'Task': 'IMAGE_CLASSIFICATION'
             },
             {
             'Domain': 'COMPUTER_VISION',
             'Framework': 'PYTORCH:1.6.0',
             'Model': 'resnet152',
             'Task': 'IMAGE_CLASSIFICATION'
             }],
    'ResponseMetadata': {
                            'HTTPHeaders': {
                            'content-length': '2345',
                            'content-type': 'application/x-amz-json-1.1',
                            'date': 'Tue, 19 Oct 2021 20:52:03 GMT',
                            'x-amzn-requestid': 'xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx'
                          },
    'HTTPStatusCode': 200,
    'RequestId': 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx',
    'RetryAttempts': 0
    }
}
```

Untuk demo ini, kami menggunakan model PyTorch (v1.7.1) ResNet -18 untuk melakukan klasifikasi gambar. Contoh kode Python berikut menyimpan kerangka kerja, versi kerangka kerja, domain, dan tugas ke dalam variabel untuk digunakan nanti:

```
# ML framework details
framework = 'pytorch'
framework_version = '1.7.1'

# ML model details
ml_domain = 'COMPUTER_VISION'
ml_task = 'IMAGE_CLASSIFICATION'
```

4. Unggah model pembelajaran mesin Anda ke Amazon S3.

Gunakan model PyTorch (v1.7.1) ResNet -18 ini jika Anda tidak memiliki model pembelajaran mesin yang telah dilatih sebelumnya:

```
# Optional: Download a sample PyTorch model
import torch
from torchvision import models, transforms, datasets

# Create an example input for tracing
image = torch.zeros([1, 3, 256, 256], dtype=torch.float32)

# Load a pretrained resnet18 model from TorchHub
model = models.resnet18(pretrained=True)

# Tell the model we are using it for evaluation (not training). Note this is
 required for Inferentia compilation.
model.eval()
model_trace = torch.jit.trace(model, image)

# Save your traced model
model_trace.save('model.pth')
```

Unduh contoh skrip `inference.py` inferensi. Buat `code` direktori dan pindahkan skrip inferensi ke `code` direktori.

```
# Download the inference script
!wget https://aws-ml-blog-artifacts.s3.us-east-2.amazonaws.com/inference.py

# move it into a code/ directory
!mkdir code
!mv inference.py code/
```

Amazon SageMaker membutuhkan model pembelajaran mesin yang telah dilatih sebelumnya untuk dikemas sebagai file TAR terkompresi (). `*.tar.gz` Kompres model dan skrip inferensi Anda untuk memenuhi persyaratan ini:

```
!tar -czf test.tar.gz model.pth code/inference.py
```

Saat titik akhir Anda disediakan, file dalam arsip diekstraksi ke `/opt/ml/model/` titik akhir.

Setelah Anda mengompres model dan model artefak sebagai `.tar.gz` file, unggah ke bucket Amazon S3 Anda. Berikut ini menunjukkan cara mengunggah model Anda ke Amazon S3 menggunakan: AWS CLI

```
!aws s3 cp test.tar.gz s3://{your-bucket}/models/
```

5. Pilih gambar inferensi Docker bawaan atau buat Inference Docker Image Anda sendiri.

   SageMaker menyediakan wadah untuk algoritme bawaan dan gambar Docker bawaan untuk beberapa kerangka kerja pembelajaran mesin yang paling umum, seperti Apache MXNet,,, dan Chainer. TensorFlow PyTorch Untuk daftar lengkap gambar yang tersedia, lihat SageMaker Gambar Deep Learning Containers yang Tersedia.

   Jika tidak ada SageMaker kontainer yang ada yang memenuhi kebutuhan Anda dan Anda tidak memiliki wadah Anda sendiri, buat image Docker baru. Lihat Gunakan kode inferensi Anda sendiri untuk informasi tentang cara membuat image Docker Anda.

   Berikut ini menunjukkan cara mengambil gambar inferensi PyTorch versi 1.7.1 menggunakan Python SDK: SageMaker

   ```
   from sagemaker import image_uris

   ## Uncomment and replace with your own values if you did not define
   ## these variables a previous step.
   #framework = 'pytorch'
   #framework_version = '1.7.1'

   # Note: you can use any CPU-based instance here,
   # this is just to set the arch as CPU for the Docker image
   instance_type = 'ml.m5.2xlarge'

   image_uri = image_uris.retrieve(framework,
                                   region,
                                   version=framework_version,
                                   py_version='py3',
                                   instance_type=instance_type,
                                   image_scope='inference')
   ```

   Untuk daftar SageMaker Instans yang tersedia, lihat SageMakerHarga Amazon.

6. Buat contoh arsip payload.

Buat arsip yang berisi file individual yang dapat dikirim oleh alat pengujian beban ke SageMaker titik akhir Anda. Kode inferensi Anda harus dapat membaca format file dari payload sampel.

Berikut ini mengunduh gambar.jpg yang digunakan contoh ini pada langkah selanjutnya untuk model ResNet -18.

```
!wget https://cdn.pixabay.com/photo/2020/12/18/05/56/flowers-5841251_1280.jpg
```

Kompres muatan sampel sebagai tarball:

```
!tar -cvzf payload.tar.gz flowers-5841251_1280.jpg
```

Unggah payload sampel ke Amazon S3 dan perhatikan URI Amazon S3:

```
!aws s3 cp payload.tar.gz s3://{bucket}/models/
```

Anda memerlukan URI Amazon S3 di langkah selanjutnya, jadi simpan dalam variabel:

```
bucket_prefix='models'
bucket = '<your-bucket-name>' # Provide the name of your S3 bucket
payload_s3_key = f"{bucket_prefix}/payload.tar.gz"
sample_payload_url= f"s3://{bucket}/{payload_s3_key}"
```

7. Siapkan masukan model Anda untuk pekerjaan rekomendasi

   Untuk prasyarat terakhir, Anda memiliki dua opsi untuk menyiapkan input model Anda. Anda dapat mendaftarkan model Anda dengan SageMaker Model Registry, yang dapat Anda gunakan untuk membuat katalog model untuk produksi, atau Anda dapat membuat SageMaker model dan menentukannya di `ContainerConfig` bidang saat membuat pekerjaan rekomendasi. Opsi pertama adalah yang terbaik jika Anda ingin memanfaatkan fitur yang disediakan Model Registry, seperti mengelola versi model dan mengotomatiskan penerapan model. Opsi kedua sangat ideal jika Anda ingin memulai dengan cepat. Untuk opsi pertama, lanjutkan ke langkah 7. Untuk opsi kedua, lewati langkah 7 dan lanjutkan ke langkah 8.

8. Opsi 1: Daftarkan model Anda di registri model

   Dengan SageMaker Model Registry, Anda dapat membuat katalog model untuk produksi, mengelola versi model, mengaitkan metadata (seperti metrik pelatihan) dengan model,

mengelola status persetujuan model, menyebarkan model ke produksi, dan mengotomatiskan penerapan model dengan CI/CD.

Bila Anda menggunakan SageMaker Model Registry untuk melacak dan mengelola model Anda, mereka direpresentasikan sebagai paket model berversi dalam grup paket model. Paket model tidak berversi bukan bagian dari grup model. Grup paket model menyimpan beberapa versi atau iterasi model. Meskipun tidak diperlukan untuk membuatnya untuk setiap model dalam registri, mereka membantu mengatur berbagai model yang semuanya memiliki tujuan yang sama dan menyediakan versi otomatis.

Untuk menggunakan Amazon SageMaker Inference Recommender, Anda harus memiliki paket model berversi. Anda dapat membuat paket model berversi secara terprogram dengan atau AWS SDK for Python (Boto3) dengan Amazon Studio Classic. SageMaker Untuk membuat paket model berversi secara terprogram, pertama-tama buat grup paket model dengan API. `CreateModelPackageGroup` Selanjutnya, buat paket model menggunakan `CreateModelPackage` API. Memanggil metode ini membuat paket model berversi.

Lihat Membuat Grup Model dan Daftarkan Versi Model untuk petunjuk terperinci tentang cara membuat grup paket model secara terprogram dan interaktif dan cara membuat paket model berversi, masing-masing, dengan Amazon Studio Classic dan AWS SDK for Python (Boto3) Amazon. SageMaker

Contoh kode berikut menunjukkan cara membuat paket model berversi menggunakan. AWS SDK for Python (Boto3)

> **Note**
>
> Anda tidak perlu menyetujui paket model untuk membuat pekerjaan Inference Recommender.

a. Buat grup paket model

   Buat grup paket model dengan `CreateModelPackageGroup` API. Berikan nama ke grup paket model untuk `ModelPackageGroupName` dan secara opsional memberikan deskripsi paket model di `ModelPackageGroupDescription` lapangan.

```
model_package_group_name = '<INSERT>'
model_package_group_description = '<INSERT>'
```

```
model_package_group_input_dict = {
 "ModelPackageGroupName" : model_package_group_name,
 "ModelPackageGroupDescription" : model_package_group_description,
}

model_package_group_response =
 sagemaker_client.create_model_package_group(**model_package_group_input_dict)
```

Lihat Panduan Referensi Amazon SageMaker API untuk daftar lengkap argumen opsional dan wajib yang dapat Anda berikan `CreateModelPackageGroup`.

Buat Paket Model dengan menentukan image Docker yang menjalankan kode inferensi dan lokasi Amazon S3 dari artefak model Anda dan berikan nilai untuknya. `InferenceSpecification` `InferenceSpecification`harus berisi informasi tentang pekerjaan inferensi yang dapat dijalankan dengan model berdasarkan Model Package ini, termasuk yang berikut:

- Jalur Amazon ECR dari gambar yang menjalankan kode inferensi Anda.
- (Opsional) Tipe instance yang didukung Model Package untuk tugas transformasi dan titik akhir real-time yang digunakan untuk inferensi.
- Format konten input dan output yang didukung Model Package untuk inferensi.

Selain itu, Anda harus menentukan parameter berikut saat membuat paket model:

- Domain: Domain pembelajaran mesin dari paket model Anda dan komponennya. Domain pembelajaran mesin yang umum termasuk visi komputer dan pemrosesan bahasa alami.
- Tugas: Tugas pembelajaran mesin yang diselesaikan oleh paket model Anda. Tugas pembelajaran mesin yang umum termasuk deteksi objek dan klasifikasi gambar. Tentukan “LAINNYA” jika tidak ada tugas yang tercantum dalam Panduan Referensi API yang memenuhi kasus penggunaan Anda. Lihat deskripsi bidang API Tugas untuk daftar tugas pembelajaran mesin yang didukung.
- SamplePayloadUrl: Jalur Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) tempat muatan sampel disimpan. Jalur ini harus menunjuk ke arsip tar terkompresi gzip tunggal (akhiran .tar.gz).
- Kerangka kerja: Kerangka pembelajaran mesin dari gambar wadah paket model.
- FrameworkVersion: Versi framework dari Model Package Container Image.

Jika Anda memberikan daftar jenis instans yang diizinkan untuk digunakan untuk menghasilkan inferensi secara real-time SupportedRealtimeInferenceInstanceTypes, Inference Recommender akan membatasi ruang pencarian untuk jenis instans selama pekerjaan. `Default` Gunakan parameter ini jika Anda memiliki batasan anggaran atau mengetahui ada sekumpulan tipe instance tertentu yang dapat mendukung model dan gambar kontainer Anda.

Pada langkah sebelumnya, kami mengunduh model ResNet 18 yang telah dilatih sebelumnya dan menyimpannya di bucket Amazon S3 di direktori bernama. `models` Kami mengambil gambar inferensi Deep Learning Container PyTorch (v1.7.1) dan menyimpan URI dalam variabel yang disebut. `image_uri` Kami menggunakan variabel-variabel tersebut dalam contoh kode berikut di mana kami mendefinisikan kamus yang digunakan sebagai masukan ke `CreateModelPackage`API.

```
# Provide the Amazon S3 URI of your compressed tarfile
# so that Model Registry knows where to find your model artifacts
bucket_prefix='models'
bucket = '<your-bucket-name>' # Provide the name of your S3 bucket
model_s3_key = f"{bucket_prefix}/test.tar.gz"
model_url= f"s3://{bucket}/{model_s3_key}"

# Similar open source model to the packaged model
# The name of the ML model as standardized by common model zoos
nearest_model_name = 'resnet18'

# The supported MIME types for input and output data. In this example,
# we are using images as input.
input_content_type='image/jpeg'


# Optional - provide a description of your model.
model_package_description = '<INSERT>'

## Uncomment if you did not store the domain and task in an earlier
## step
#ml_domain = 'COMPUTER_VISION'
#ml_task = 'IMAGE_CLASSIFICATION'

## Uncomment if you did not store the framework and framework version
## in a previous step.
```

```
#framework = 'PYTORCH'
#framework_version = '1.7.1'

# Optional: Used for optimizing your model using SageMaker Neo
# PyTorch uses NCHW format for images
data_input_configuration = "[[1,3,256,256]]"

# Create a dictionary to use as input for creating a model pacakge group
model_package_input_dict = {
        "ModelPackageGroupName" : model_package_group_name,
        "ModelPackageDescription" : model_package_description,
        "Domain": ml_domain,
        "Task": ml_task,
        "SamplePayloadUrl": sample_payload_url,
        "InferenceSpecification": {
                "Containers": [
                    {
                        "Image": image_uri,
                        "ModelDataUrl": model_url,
                        "Framework": framework.upper(),
                        "FrameworkVersion": framework_version,
                        "NearestModelName": nearest_model_name,
                        "ModelInput": {"DataInputConfig":
 data_input_configuration}
                    }
                    ],
                "SupportedContentTypes": [input_content_type]
        }
    }
```

b. Membuat Model Package

Gunakan `CreateModelPackage` API untuk membuat Model Package. Lulus kamus masukan yang ditentukan pada langkah sebelumnya:

```
model_package_response =
 sagemaker_client.create_model_package(**model_package_input_dict)
```

Anda memerlukan paket model ARN untuk menggunakan Amazon SageMaker Inference Recommender. Perhatikan ARN dari paket model atau simpan dalam variabel:

```
model_package_arn = model_package_response["ModelPackageArn"]
```

```
print('ModelPackage Version ARN : {}'.format(model_package_arn))
```

9. Opsi 2: Buat model dan konfigurasikan **`ContainerConfig`** bidang

   Gunakan opsi ini jika Anda ingin memulai pekerjaan rekomendasi inferensi dan tidak perlu mendaftarkan model Anda di Registri Model. Pada langkah-langkah berikut, Anda membuat model SageMaker dan mengonfigurasi `ContainerConfig` bidang sebagai masukan untuk pekerjaan rekomendasi.

   a. Buat model

      Buat model dengan `CreateModel` API. Untuk contoh yang memanggil metode ini saat menerapkan model ke SageMaker Hosting, lihat Membuat Model (AWS SDK for Python (Boto3)).

      Pada langkah sebelumnya, kami mengunduh model ResNet 18 yang telah dilatih sebelumnya dan menyimpannya di bucket Amazon S3 di direktori bernama. `models` Kami mengambil gambar inferensi Deep Learning Container PyTorch (v1.7.1) dan menyimpan URI dalam variabel yang disebut. `image_uri` Kami menggunakan variabel-variabel tersebut dalam contoh kode berikut di mana kami mendefinisikan kamus yang digunakan sebagai masukan ke `CreateModel` API.

```
model_name = '<name_of_the_model>'
# Role to give SageMaker permission to access AWS services.
sagemaker_role= "arn:aws:iam::<region>:<account>:role/*"

# Provide the Amazon S3 URI of your compressed tarfile
# so that Model Registry knows where to find your model artifacts
bucket_prefix='models'
bucket = '<your-bucket-name>' # Provide the name of your S3 bucket
model_s3_key = f"{bucket_prefix}/test.tar.gz"
model_url= f"s3://{bucket}/{model_s3_key}"

#Create model
create_model_response = sagemaker_client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = sagemaker_role,
    PrimaryContainer = {
        'Image': image_uri,
        'ModelDataUrl': model_url,
```

```
    })
```

b. Konfigurasikan **`ContainerConfig`** bidang

Selanjutnya, Anda harus mengkonfigurasi ContainerConfigbidang dengan model yang baru saja Anda buat dan tentukan parameter berikut di dalamnya:

- `Domain`: Domain pembelajaran mesin model dan komponennya, seperti visi komputer atau pemrosesan bahasa alami.
- `Task`Tugas pembelajaran mesin yang diselesaikan model, seperti klasifikasi gambar atau deteksi objek.
- `PayloadConfig`: Konfigurasi untuk payload untuk pekerjaan rekomendasi. Untuk informasi selengkapnya tentang subbidang, lihat`RecommendationJobPayloadConfig`.
- `Framework`: Kerangka pembelajaran mesin dari gambar wadah, seperti PyTorch.
- `FrameworkVersion`: Versi kerangka gambar kontainer.
- (Opsional)`SupportedInstanceTypes`: Daftar jenis instance yang digunakan untuk menghasilkan kesimpulan secara real-time.

Jika Anda menggunakan `SupportedInstanceTypes` parameter, Inference Recommender membatasi ruang pencarian untuk jenis instance selama pekerjaan. `Default` Gunakan parameter ini jika Anda memiliki batasan anggaran atau mengetahui ada sekumpulan tipe instance tertentu yang dapat mendukung model dan gambar kontainer Anda.

Dalam contoh kode berikut, kita menggunakan parameter yang ditentukan sebelumnya, bersama dengan`NearestModelName`, untuk mendefinisikan kamus yang digunakan sebagai masukan ke `CreateInferenceRecommendationsJob` API.

```
## Uncomment if you did not store the domain and task in a previous step
#ml_domain = 'COMPUTER_VISION'
#ml_task = 'IMAGE_CLASSIFICATION'

## Uncomment if you did not store the framework and framework version in a
 previous step
#framework = 'PYTORCH'
#framework_version = '1.7.1'

# The name of the ML model as standardized by common model zoos
nearest_model_name = 'resnet18'
```

```
# The supported MIME types for input and output data. In this example,
# we are using images as input
input_content_type='image/jpeg'

# Optional: Used for optimizing your model using SageMaker Neo
# PyTorch uses NCHW format for images
data_input_configuration = "[[1,3,256,256]]"

# Create a dictionary to use as input for creating an inference recommendation
 job
container_config = {
        "Domain": ml_domain,
        "Framework": framework.upper(),
        "FrameworkVersion": framework_version,
        "NearestModelName": nearest_model_name,
        "PayloadConfig": {
            "SamplePayloadUrl": sample_payload_url,
            "SupportedContentTypes": [ input_content_type ]
         },
        "DataInputConfig": data_input_configuration
        "Task": ml_task,
        }
```

Lowongan kerja rekomendasi

Amazon SageMaker Inference Recommender dapat membuat dua jenis rekomendasi:

1. Rekomendasi inferensi (tipe `Default` pekerjaan) menjalankan serangkaian tes beban pada jenis instance yang direkomendasikan. Anda juga dapat memuat pengujian untuk titik akhir tanpa server.. Anda hanya perlu menyediakan paket model Amazon Resource Name (ARN) untuk meluncurkan jenis pekerjaan rekomendasi ini. Pekerjaan rekomendasi inferensi selesai dalam waktu 45 menit.
2. Rekomendasi titik akhir (tipe `Advanced` pekerjaan) didasarkan pada uji beban khusus di mana Anda memilih instans ML yang diinginkan atau titik akhir tanpa server, menyediakan pola lalu lintas khusus, dan menyediakan persyaratan untuk latensi dan throughput berdasarkan persyaratan produksi Anda. Pekerjaan ini membutuhkan waktu rata-rata 2 jam untuk diselesaikan tergantung pada durasi pekerjaan yang ditetapkan dan jumlah total konfigurasi inferensi yang diuji.

Kedua jenis rekomendasi menggunakan API yang sama untuk membuat, mendeskripsikan, dan menghentikan pekerjaan. Outputnya adalah daftar rekomendasi konfigurasi instance dengan variabel lingkungan terkait, biaya, throughput, dan metrik latensi. Pekerjaan rekomendasi juga menyediakan jumlah instans awal yang dapat Anda gunakan untuk mengonfigurasi kebijakan penskalaan otomatis. Untuk membedakan antara dua jenis pekerjaan, saat Anda membuat pekerjaan melalui SageMaker konsol atau API, tentukan `Default` untuk membuat rekomendasi titik akhir awal dan untuk pengujian pemuatan kustom dan `Advanced` rekomendasi titik akhir.

> **Note**
>
> Anda tidak perlu melakukan kedua jenis pekerjaan rekomendasi dalam alur kerja Anda sendiri. Anda dapat melakukan keduanya secara independen satu sama lain.

Inference Recommender juga dapat memberi Anda daftar instans prospektif, atau lima jenis instans teratas yang dioptimalkan untuk biaya, throughput, dan latensi untuk penerapan model, bersama dengan skor kepercayaan. Anda dapat memilih instance ini saat menerapkan model Anda. Inference Recommender secara otomatis melakukan benchmarking terhadap model Anda agar Anda dapat memberikan contoh prospektif. Karena ini adalah rekomendasi awal, kami sarankan Anda menjalankan pekerjaan rekomendasi contoh lebih lanjut untuk mendapatkan hasil yang lebih akurat. Untuk melihat contoh prospektif, buka halaman detail SageMaker model Anda. Untuk informasi selengkapnya, lihat Dapatkan instans prospektif instan.

Topik

Dapatkan instans prospektif instan

Inference Recommender juga dapat memberi Anda daftar instance prospektif, atau jenis instans yang mungkin cocok untuk model Anda, di halaman detail model Anda SageMaker . Inference Recommender secara otomatis melakukan benchmarking awal terhadap model Anda agar Anda dapat memberikan lima contoh prospektif teratas. Karena ini adalah rekomendasi awal, kami sarankan Anda menjalankan pekerjaan rekomendasi contoh lebih lanjut untuk mendapatkan hasil yang lebih akurat.

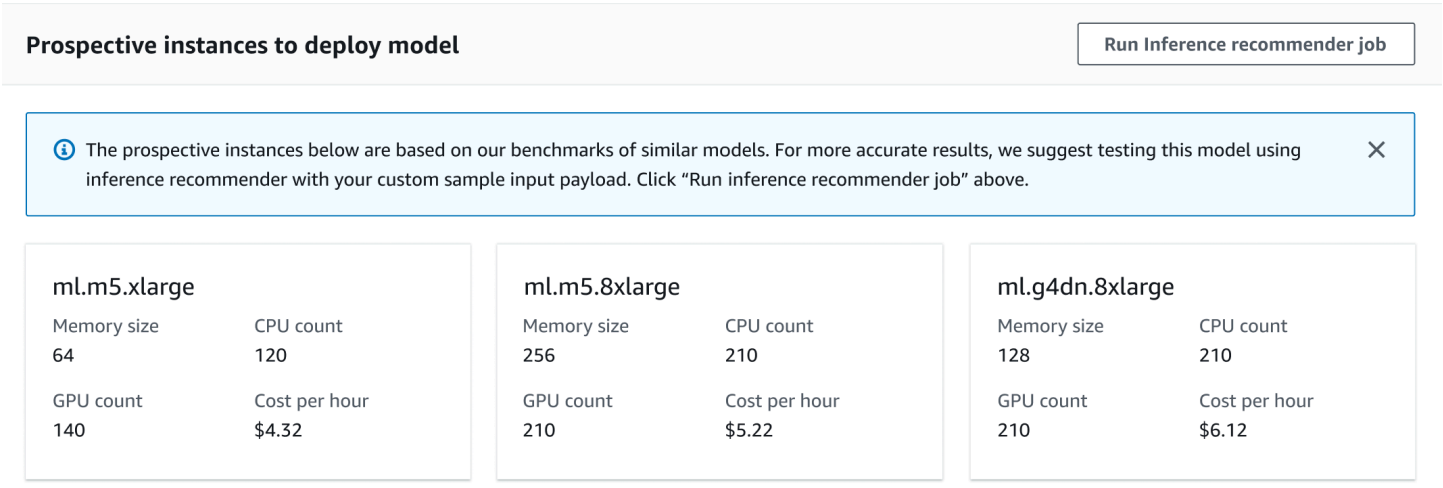
Anda dapat melihat daftar instance prospektif untuk model Anda baik secara terprogram dengan menggunakan API, SageMaker Python DescribeModelSDK, atau konsol. SageMaker

> **Note**
>
> Anda tidak akan mendapatkan contoh prospektif untuk model yang Anda buat SageMaker sebelum fitur ini tersedia.

Untuk melihat instance prospektif model Anda melalui konsol, lakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Model.
3. Dari daftar model, pilih model Anda.

Pada halaman detail untuk model Anda, buka bagian Prospective instance to deploy model. Screenshot berikut menunjukkan bagian ini.



Di bagian ini, Anda dapat melihat instans prospektif yang dioptimalkan untuk biaya, throughput, dan latensi untuk penerapan model, bersama dengan informasi tambahan untuk setiap jenis instans seperti ukuran memori, jumlah CPU dan GPU, dan biaya per jam.

Jika Anda memutuskan ingin membandingkan muatan sampel dan menjalankan pekerjaan rekomendasi inferensi penuh untuk model Anda, Anda dapat memulai pekerjaan rekomendasi inferensi default dari halaman ini. Untuk memulai pekerjaan default melalui konsol, lakukan hal berikut:

1. Pada halaman detail model Anda di bagian Prospective instance to deploy model, pilih Run Inference recommended job.
2. Di kotak dialog yang muncul, untuk bucket S3 untuk benchmarking payload, masukkan lokasi Amazon S3 tempat Anda menyimpan muatan sampel untuk model Anda.
3. Untuk jenis konten Payload, masukkan tipe MIME untuk data payload Anda.
4. (Opsional) Dalam kompilasi Model menggunakan bagian SageMaker Neo, untuk konfigurasi input Data, masukkan bentuk data dalam format kamus.
5. Pilih Jalankan tugas.

Inference Recommender memulai pekerjaan, dan Anda dapat melihat pekerjaan dan hasilnya dari halaman daftar pemberi rekomendasi Inferensi di konsol. SageMaker

Jika Anda ingin menjalankan pekerjaan lanjutan dan melakukan uji beban khusus, atau jika Anda ingin mengonfigurasi pengaturan dan parameter tambahan untuk pekerjaan Anda, lihatJalankan uji beban khusus.

Dapatkan rekomendasi inferensi

Pekerjaan rekomendasi inferensi menjalankan serangkaian tes beban pada jenis instance yang direkomendasikan atau titik akhir tanpa server. Pekerjaan rekomendasi inferensi menggunakan metrik kinerja yang didasarkan pada uji beban menggunakan data sampel yang Anda berikan selama pendaftaran versi model.

> **Note**
>
> Sebelum Anda membuat pekerjaan rekomendasi Inference Recommender, pastikan Anda telah puas. Prasyarat

Berikut ini menunjukkan cara menggunakan Amazon SageMaker Inference Recommender untuk membuat rekomendasi inferensi berdasarkan jenis model Anda menggunakanAWS SDK for Python (Boto3),, AWS CLI dan Amazon SageMaker Studio Classic, dan konsol SageMaker

Buat rekomendasi inferensi

Buat rekomendasi inferensi secara terprogram menggunakan atauAWS CLI, AWS SDK for Python (Boto3) atau secara interaktif menggunakan Studio Classic atau konsol. SageMaker Tentukan nama pekerjaan untuk rekomendasi inferensi Anda, ARN peran AWS IAM, konfigurasi input, dan paket model ARN saat Anda mendaftarkan model Anda dengan registri model, atau nama model dan **`ContainerConfig`** kamus dari saat Anda membuat model di bagian Prasyarat.

AWS SDK for Python (Boto3)

Gunakan `CreateInferenceRecommendationsJob`API untuk memulai pekerjaan rekomendasi inferensi. Tetapkan `JobType` bidang `'Default'` untuk pekerjaan rekomendasi inferensi. Selain itu, berikan yang berikut:

- Nama Sumber Daya Amazon (ARN) dari peran IAM yang memungkinkan Inference Recommender untuk melakukan tugas atas nama Anda. Tentukan ini untuk `RoleArn` bidang.
- Paket model ARN atau nama model. Inference Recommender mendukung salah satu paket model ARN atau nama model sebagai input. Tentukan satu dari yang berikut ini:
  - ARN dari paket model berversi yang Anda buat saat Anda mendaftarkan model Anda dengan registri model. Tentukan ini untuk `ModelPackageVersionArn` di `InputConfig` lapangan.
  - Nama model yang Anda buat. Tentukan ini untuk `ModelName` di `InputConfig` lapangan. Juga, berikan `ContainerConfig` kamus, yang mencakup bidang wajib yang perlu disediakan dengan nama model. Tentukan ini untuk `ContainerConfig` di `InputConfig` lapangan. Dalam`ContainerConfig`, Anda juga dapat secara opsional menentukan `SupportedEndpointType` bidang sebagai salah satu `RealTime` atau`Serverless`. Jika Anda menentukan bidang ini, Inference Recommender mengembalikan rekomendasi hanya untuk jenis titik akhir tersebut. Jika Anda tidak menentukan bidang ini, Inference Recommender mengembalikan rekomendasi untuk kedua tipe titik akhir.
- Nama untuk pekerjaan rekomendasi Inference Recommender Anda untuk bidang tersebut. `JobName` Nama pekerjaan Inference Recommender harus unik di dalam AWS Wilayah dan di dalam akun AndaAWS.

Impor AWS SDK for Python (Boto3) paket dan buat objek SageMaker klien menggunakan kelas klien. Jika Anda mengikuti langkah-langkah di bagian Prasyarat, hanya tentukan salah satu dari berikut ini:

- Opsi 1: Jika Anda ingin membuat pekerjaan rekomendasi inferensi dengan paket model ARN, maka simpan grup paket model ARN dalam variabel bernama. `model_package_arn`
- Opsi 2: Jika Anda ingin membuat pekerjaan rekomendasi inferensi dengan nama model dan`ContainerConfig`, simpan nama model dalam variabel bernama `model_name` dan `ContainerConfig` kamus dalam variabel bernama`container_config`.

```
# Create a low-level SageMaker service client.
import boto3
aws_region = '<INSERT>'
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Provide only one of model package ARN or model name, not both.
# Provide your model package ARN that was created when you registered your
# model with Model Registry
model_package_arn = '<INSERT>'
## Uncomment if you would like to create an inference recommendations job with a
## model name instead of a model package ARN, and comment out model_package_arn
 above
## Provide your model name
# model_name = '<INSERT>'
## Provide your container config
# container_config = '<INSERT>'

# Provide a unique job name for SageMaker Inference Recommender job
job_name = '<INSERT>'

# Inference Recommender job type. Set to Default to get an initial recommendation
job_type = 'Default'

# Provide an IAM Role that gives SageMaker Inference Recommender permission to
# access AWS services
role_arn = 'arn:aws:iam::<account>:role/*'

sagemaker_client.create_inference_recommendations_job(
    JobName = job_name,
    JobType = job_type,
    RoleArn = role_arn,
```

```
    # Provide only one of model package ARN or model name, not both.
    # If you would like to create an inference recommendations job with a model
 name,
    # uncomment ModelName and ContainerConfig, and comment out
 ModelPackageVersionArn.
    InputConfig = {
        'ModelPackageVersionArn': model_package_arn
        # 'ModelName': model_name,
        # 'ContainerConfig': container_config
    }
)
```

Lihat Panduan Referensi Amazon SageMaker API untuk daftar lengkap argumen opsional dan wajib yang dapat Anda berikan `CreateInferenceRecommendationsJob`.

AWS CLI

Gunakan `create-inference-recommendations-job` API untuk memulai pekerjaan rekomendasi inferensi. Tetapkan `job-type` bidang `'Default'` untuk pekerjaan rekomendasi inferensi. Selain itu, berikan yang berikut:

- Nama Sumber Daya Amazon (ARN) dari peran IAM yang memungkinkan Amazon SageMaker Inference Recommender untuk melakukan tugas atas nama Anda. Tentukan ini untuk `role-arn` bidang.
- Paket model ARN atau nama model. Inference Recommender mendukung salah satu paket model ARN atau nama model sebagai input. Tentukan salah satu dari berikut ini
  - ARN dari paket model berversi yang Anda buat saat Anda mendaftarkan model Anda dengan Model Registry. Tentukan ini untuk `ModelPackageVersionArn` di `input-config` lapangan.
  - Nama model yang Anda buat. Tentukan ini untuk `ModelName` di `input-config` lapangan. Juga, berikan `ContainerConfig` kamus yang mencakup bidang wajib yang perlu disediakan dengan nama model. Tentukan ini untuk `ContainerConfig` di `input-config` lapangan. Dalam`ContainerConfig`, Anda juga dapat secara opsional menentukan `SupportedEndpointType` bidang sebagai salah satu `RealTime` atau`Serverless`. Jika Anda menentukan bidang ini, Inference Recommender mengembalikan rekomendasi hanya untuk jenis titik akhir tersebut. Jika Anda tidak menentukan bidang ini, Inference Recommender mengembalikan rekomendasi untuk kedua tipe titik akhir.

- Nama untuk pekerjaan rekomendasi Inference Recommender Anda untuk bidang tersebut. `job-name` Nama pekerjaan Inference Recommender harus unik di dalam AWS Wilayah dan di dalam akun AndaAWS.

Untuk membuat lowongan rekomendasi inferensi dengan paket model ARN, gunakan contoh berikut:

```
aws sagemaker create-inference-recommendations-job
    --region <region>\
    --job-name <job_name>\
    --job-type Default\
    --role-arn arn:aws:iam::<account:role/*>\
    --input-config "{
        \"ModelPackageVersionArn\": \"arn:aws:sagemaker:<region:account:role/*>\",
        }"
```

Untuk membuat pekerjaan rekomendasi inferensi dengan nama model dan`ContainerConfig`, gunakan contoh berikut. Contoh menggunakan `SupportedEndpointType` bidang untuk menentukan bahwa kami hanya ingin mengembalikan rekomendasi inferensi waktu nyata:

```
aws sagemaker create-inference-recommendations-job
    --region <region>\
    --job-name <job_name>\
    --job-type Default\
    --role-arn arn:aws:iam::<account:role/*>\
    --input-config "{
        \"ModelName\": \"model-name\",
        \"ContainerConfig\" : {
                \"Domain\": \"COMPUTER_VISION\",
                \"Framework\": \"PYTORCH\",
                \"FrameworkVersion\": \"1.7.1\",
                \"NearestModelName\": \"resnet18\",
                \"PayloadConfig\":
                    {
                        \"SamplePayloadUrl\": \"s3://{bucket}/{payload_s3_key}\",
                        \"SupportedContentTypes\": [\"image/jpeg\"]
                    },
                \"SupportedEndpointType\": \"RealTime\",
                \"DataInputConfig\": \"[[1,3,256,256]]\",
                \"Task\": \"IMAGE_CLASSIFICATION\",
            },
```

```
    }"
```

Amazon SageMaker Studio Classic

Buat pekerjaan rekomendasi inferensi di Studio Classic.

1. Di aplikasi Studio Classic Anda, pilih ikon beranda ().
2. Di bilah sisi kiri Studio Classic, pilih Model.
3. Pilih Model Registry dari daftar dropdown untuk menampilkan model yang telah Anda daftarkan dengan registri model.

   Panel kiri menampilkan daftar grup model. Daftar ini mencakup semua grup model yang terdaftar dengan registri model di akun Anda, termasuk model yang terdaftar di luar Studio Classic.
4. Pilih nama grup model Anda. Saat Anda memilih grup model, panel kanan Studio Classic menampilkan kepala kolom seperti Versi dan Pengaturan.

   Jika Anda memiliki satu atau lebih paket model dalam grup model Anda, Anda akan melihat daftar paket model tersebut dalam kolom Versi.
5. Pilih kolom Inference recommended.
6. Pilih peran IAM yang memberikan izin Inference Recommender untuk mengakses layanan. AWS Anda dapat membuat peran dan melampirkan kebijakan terkelola `AmazonSageMakerFullAccess` IAM untuk mencapai hal ini. Atau Anda dapat membiarkan Studio Classic membuat peran untuk Anda.
7. Pilih Dapatkan rekomendasi.

   Rekomendasi inferensi dapat memakan waktu hingga 45 menit.

   > **Warning**
   >
   > Jangan tutup tab ini. Jika Anda menutup tab ini, Anda membatalkan pekerjaan rekomendasi instans.

SageMaker console

Buat pekerjaan rekomendasi instance melalui SageMaker konsol dengan melakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Inference recommended.
3. Pada halaman lowongan pemberi rekomendasi inferensi, pilih Buat pekerjaan.
4. Untuk Langkah 1: Konfigurasi model, lakukan hal berikut:

   a. Untuk jenis Job, pilih Default recommended job.

   b. Jika Anda menggunakan model yang terdaftar di registri SageMaker model, aktifkan tombol Pilih model dari registri model dan lakukan hal berikut:

      i. Dari daftar dropdown grup Model, pilih grup model di registri SageMaker model tempat model Anda berada.

      ii. Dari daftar dropdown versi Model, pilih versi model yang diinginkan.

   c. Jika Anda menggunakan model yang telah Anda buat SageMaker, matikan sakelar Pilih model dari registri model dan lakukan hal berikut:

      • Untuk bidang Nama model, masukkan nama SageMaker model Anda.

   d. Dari daftar tarik-turun peran IAM, Anda dapat memilih peran AWS IAM yang ada yang memiliki izin yang diperlukan untuk membuat pekerjaan rekomendasi instans. Atau, jika Anda tidak memiliki peran yang ada, Anda dapat memilih Buat peran baru untuk membuka pop-up pembuatan peran, dan SageMaker menambahkan izin yang diperlukan ke peran baru yang Anda buat.

   e. Untuk bucket S3 untuk benchmarking payload, masukkan path Amazon S3 ke arsip payload sampel Anda, yang harus berisi contoh file payload yang digunakan Inference Recomder untuk membandingkan model Anda pada jenis instans yang berbeda.

   f. Untuk jenis konten Payload, masukkan tipe MIME dari data payload sampel Anda.

   g. (Opsional) Jika Anda mematikan Pilih model dari registri model toggle dan menentukan SageMaker model, maka untuk konfigurasi Container, lakukan hal berikut:

      i. Untuk daftar dropdown Domain, pilih domain pembelajaran mesin model, seperti visi komputer, pemrosesan bahasa alami, atau pembelajaran mesin.

      ii. Untuk daftar dropdown Framework, pilih framework container Anda, seperti TensorFlow atau XGBoost.

      iii. Untuk versi Framework, masukkan versi kerangka gambar kontainer Anda.

      iv. Untuk daftar dropdown nama model Terdekat, pilih model pra-terlatih yang sebagian besar cocok dengan model Anda.

v. Untuk daftar tarik-turun Tugas, pilih tugas pembelajaran mesin yang diselesaikan model, seperti klasifikasi gambar atau regresi.

h. (Opsional) Untuk kompilasi Model menggunakan SageMaker Neo, Anda dapat mengonfigurasi pekerjaan rekomendasi untuk model yang telah Anda kompilasi menggunakan SageMaker Neo. Untuk konfigurasi input Data, masukkan bentuk data input yang benar untuk model Anda dalam format yang mirip dengan`{'input': [1,1024,1024,3]}`.

i. Pilih Berikutnya.

5. Untuk Langkah 2: Contoh dan parameter lingkungan, lakukan hal berikut:

   a. (Opsional) Untuk Select instance untuk benchmarking, Anda dapat memilih hingga 8 jenis instans yang ingin Anda benchmark. Jika Anda tidak memilih instans apa pun, Inference Recommender mempertimbangkan semua jenis instance.

   b. Pilih Berikutnya.

6. Untuk Langkah 3: Parameter Job, lakukan hal berikut:

   a. (Opsional) Untuk bidang Nama Job, masukkan nama untuk pekerjaan rekomendasi instans Anda. Saat Anda membuat pekerjaan, SageMaker tambahkan stempel waktu ke akhir nama ini.

   b. (Opsional) Untuk kolom Job description, masukkan deskripsi untuk pekerjaan tersebut.

   c. (Opsional) Untuk daftar dropdown kunci Enkripsi, pilih AWS KMS kunci berdasarkan nama atau masukkan ARN untuk mengenkripsi data Anda.

   d. (Opsional) Untuk durasi pengujian Maks, masukkan jumlah detik maksimum yang Anda inginkan setiap pengujian dijalankan.

   e. (Opsional) Untuk pemanggilan Maks per menit, masukkan jumlah maksimum permintaan per menit yang dapat dicapai titik akhir sebelum menghentikan pekerjaan rekomendasi. Setelah mencapai batas ini, SageMaker akhiri pekerjaan.

   f. (Opsional) Untuk ambang latensi Model P99 (ms), masukkan persentil latensi model dalam milidetik.

   g. Pilih Berikutnya.

7. Untuk Langkah 4: Tinjau pekerjaan, tinjau konfigurasi Anda, lalu pilih Kirim.

Dapatkan hasil pekerjaan rekomendasi inferensi Anda

Kumpulkan hasil pekerjaan rekomendasi inferensi Anda secara terprogram denganAWS SDK for Python (Boto3), Studio ClassicAWS CLI, atau konsol. SageMaker

AWS SDK for Python (Boto3)

Setelah rekomendasi inferensi selesai, Anda dapat menggunakan `DescribeInferenceRecommendationsJob` untuk mendapatkan rincian pekerjaan dan rekomendasi. Berikan nama pekerjaan yang Anda gunakan saat Anda membuat pekerjaan rekomendasi inferensi.

```
job_name='<INSERT>'
response = sagemaker_client.describe_inference_recommendations_job(
                    JobName=job_name)
```

Cetak objek respons. Contoh kode sebelumnya menyimpan respons dalam nama variabel`response`.

```
print(response['Status'])
```

Ini mengembalikan respon JSON mirip dengan contoh berikut. Perhatikan bahwa contoh ini menunjukkan jenis instance yang direkomendasikan untuk inferensi waktu nyata (untuk contoh yang menunjukkan rekomendasi inferensi tanpa server, lihat contoh setelah yang ini).

```
{
    'JobName': 'job-name',
    'JobDescription': 'job-description',
    'JobType': 'Default',
    'JobArn': 'arn:aws:sagemaker:region:account-id:inference-recommendations-
job/resource-id',
    'Status': 'COMPLETED',
    'CreationTime': datetime.datetime(2021, 10, 26, 20, 4, 57, 627000,
 tzinfo=tzlocal()),
    'LastModifiedTime': datetime.datetime(2021, 10, 26, 20, 25, 1, 997000,
 tzinfo=tzlocal()),
    'InputConfig': {
                'ModelPackageVersionArn': 'arn:aws:sagemaker:region:account-
id:model-package/resource-id',
                'JobDurationInSeconds': 0
                },
    'InferenceRecommendations': [{
```

```
            'Metrics': {
                'CostPerHour': 0.20399999618530273,
                'CostPerInference': 5.246913588052848e-06,
                'MaximumInvocations': 648,
                'ModelLatency': 263596
                },
            'EndpointConfiguration': {
                'EndpointName': 'endpoint-name',
                'VariantName': 'variant-name',
                'InstanceType': 'ml.c5.xlarge',
                'InitialInstanceCount': 1
                },
            'ModelConfiguration': {
                'Compiled': False,
                'EnvironmentParameters': []
                }
        },
        {
            'Metrics': {
                'CostPerHour': 0.11500000208616257,
                'CostPerInference': 2.92620870823157e-06,
                'MaximumInvocations': 655,
                'ModelLatency': 826019
                },
            'EndpointConfiguration': {
                'EndpointName': 'endpoint-name',
                'VariantName': 'variant-name',
                'InstanceType': 'ml.c5d.large',
                'InitialInstanceCount': 1
                },
            'ModelConfiguration': {
                'Compiled': False,
                'EnvironmentParameters': []
                }
            },
            {
                'Metrics': {
                    'CostPerHour': 0.11500000208616257,
                    'CostPerInference': 3.3625731248321244e-06,
                    'MaximumInvocations': 570,
                    'ModelLatency': 1085446
                    },
                'EndpointConfiguration': {
                    'EndpointName': 'endpoint-name',
```

```
                    'VariantName': 'variant-name',
                    'InstanceType': 'ml.m5.large',
                    'InitialInstanceCount': 1
                    },
                'ModelConfiguration': {
                    'Compiled': False,
                    'EnvironmentParameters': []
                    }
            }],
    'ResponseMetadata': {
        'RequestId': 'request-id',
        'HTTPStatusCode': 200,
        'HTTPHeaders': {
            'x-amzn-requestid': 'x-amzn-requestid',
            'content-type': 'content-type',
            'content-length': '1685',
            'date': 'Tue, 26 Oct 2021 20:31:10 GMT'
            },
        'RetryAttempts': 0
        }
}
```

Beberapa baris pertama memberikan informasi tentang pekerjaan rekomendasi inferensi itu sendiri. Ini termasuk nama pekerjaan, ARN peran, dan waktu pembuatan dan penghapusan.

`InferenceRecommendations`Kamus berisi daftar rekomendasi inferensi Inference Recommender.

Kamus `EndpointConfiguration` bersarang berisi rekomendasi tipe instance (`InstanceType`) bersama dengan titik akhir dan nama varian (model pembelajaran AWS mesin yang diterapkan) yang digunakan selama pekerjaan rekomendasi. Anda dapat menggunakan nama endpoint dan varian untuk pemantauan di Amazon CloudWatch Events. Untuk informasi selengkapnya, lihat Pantau Amazon SageMaker dengan Amazon CloudWatch.

Kamus `Metrics` bersarang berisi informasi tentang perkiraan biaya per jam (`CostPerHour`) untuk titik akhir real-time Anda dalam dolar AS, perkiraan biaya per inferensi (`CostPerInference`) dalam dolar AS untuk titik akhir waktu nyata Anda, jumlah maksimum yang diharapkan `InvokeEndpoint` permintaan per menit yang dikirim ke titik akhir (`MaxInvocations`), dan latensi model (`ModelLatency`), yang merupakan interval waktu (dalam mikrodetik) yang diambil model Anda untuk merespons. SageMaker Latensi model mencakup

waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah.

Contoh berikut menunjukkan `InferenceRecommendations` bagian dari respons untuk pekerjaan rekomendasi inferensi yang dikonfigurasi untuk mengembalikan rekomendasi inferensi tanpa server:

```
"InferenceRecommendations": [
      {
         "EndpointConfiguration": {
            "EndpointName": "value",
            "InitialInstanceCount": value,
            "InstanceType": "value",
            "VariantName": "value",
            "ServerlessConfig": {
                "MaxConcurrency": value,
                "MemorySizeInMb": value
            }
         },
         "InvocationEndTime": value,
         "InvocationStartTime": value,
         "Metrics": {
            "CostPerHour": value,
            "CostPerInference": value,
            "CpuUtilization": value,
            "MaxInvocations": value,
            "MemoryUtilization": value,
            "ModelLatency": value,
            "ModelSetupTime": value
         },
         "ModelConfiguration": {
            "Compiled": "False",
            "EnvironmentParameters": [],
            "InferenceSpecificationName": "value"
         },
         "RecommendationId": "value"
      }
   ]
```

Anda dapat menafsirkan rekomendasi untuk inferensi tanpa server mirip dengan hasil untuk inferensi waktu nyata, dengan pengecualian`ServerlessConfig`, yang memberi tahu Anda metrik yang dikembalikan untuk titik akhir tanpa server dengan yang diberikan dan kapan.

`MemorySizeInMB` `MaxConcurrency = 1` Untuk meningkatkan throughput yang mungkin pada titik akhir, tingkatkan nilai linier. `MaxConcurrency` Misalnya, jika rekomendasi inferensi menunjukkan `MaxInvocations` sebagai`1000`, maka meningkat `MaxConcurrency` menjadi 2 akan mendukung 2000`MaxInvocations`. Perhatikan bahwa ini benar hanya sampai titik tertentu, yang dapat bervariasi berdasarkan model dan kode Anda. Rekomendasi tanpa server juga mengukur metrik`ModelSetupTime`, yang mengukur (dalam mikrodetik) waktu yang diperlukan untuk meluncurkan sumber daya komputer pada titik akhir tanpa server. Untuk informasi selengkapnya tentang pengaturan titik akhir tanpa server, lihat dokumentasi Inferensi Tanpa Server.

AWS CLI

Setelah rekomendasi inferensi selesai, Anda dapat menggunakan `describe-inference-recommendations-job` untuk mendapatkan detail pekerjaan dan jenis instans yang direkomendasikan. Berikan nama pekerjaan yang Anda gunakan saat Anda membuat pekerjaan rekomendasi inferensi.

```
aws sagemaker describe-inference-recommendations-job\
    --job-name <job-name>\
    --region <aws-region>
```

Respons JSON serupa harus menyerupai contoh berikut. Perhatikan bahwa contoh ini menunjukkan jenis instance yang direkomendasikan untuk inferensi waktu nyata (untuk contoh yang menunjukkan rekomendasi inferensi tanpa server, lihat contoh setelah yang ini).

```
{
    'JobName': 'job-name',
    'JobDescription': 'job-description',
    'JobType': 'Default',
    'JobArn': 'arn:aws:sagemaker:region:account-id:inference-recommendations-
job/resource-id',
    'Status': 'COMPLETED',
    'CreationTime': datetime.datetime(2021, 10, 26, 20, 4, 57, 627000,
 tzinfo=tzlocal()),
    'LastModifiedTime': datetime.datetime(2021, 10, 26, 20, 25, 1, 997000,
 tzinfo=tzlocal()),
    'InputConfig': {
                'ModelPackageVersionArn': 'arn:aws:sagemaker:region:account-
id:model-package/resource-id',
                'JobDurationInSeconds': 0
                },
```

```
    'InferenceRecommendations': [{
            'Metrics': {
                'CostPerHour': 0.20399999618530273,
                'CostPerInference': 5.246913588052848e-06,
                'MaximumInvocations': 648,
                'ModelLatency': 263596
                },
            'EndpointConfiguration': {
                'EndpointName': 'endpoint-name',
                'VariantName': 'variant-name',
                'InstanceType': 'ml.c5.xlarge',
                'InitialInstanceCount': 1
                },
            'ModelConfiguration': {
                'Compiled': False,
                'EnvironmentParameters': []
                }
        },
        {
            'Metrics': {
                'CostPerHour': 0.11500000208616257,
                'CostPerInference': 2.92620870823157e-06,
                'MaximumInvocations': 655,
                'ModelLatency': 826019
                },
            'EndpointConfiguration': {
                'EndpointName': 'endpoint-name',
                'VariantName': 'variant-name',
                'InstanceType': 'ml.c5d.large',
                'InitialInstanceCount': 1
                },
            'ModelConfiguration': {
                'Compiled': False,
                'EnvironmentParameters': []
                }
            },
            {
                'Metrics': {
                    'CostPerHour': 0.11500000208616257,
                    'CostPerInference': 3.3625731248321244e-06,
                    'MaximumInvocations': 570,
                    'ModelLatency': 1085446
                    },
                'EndpointConfiguration': {
```

```
                    'EndpointName': 'endpoint-name',
                    'VariantName': 'variant-name',
                    'InstanceType': 'ml.m5.large',
                    'InitialInstanceCount': 1
                    },
                'ModelConfiguration': {
                    'Compiled': False,
                    'EnvironmentParameters': []
                    }
            }],
    'ResponseMetadata': {
        'RequestId': 'request-id',
        'HTTPStatusCode': 200,
        'HTTPHeaders': {
            'x-amzn-requestid': 'x-amzn-requestid',
            'content-type': 'content-type',
            'content-length': '1685',
            'date': 'Tue, 26 Oct 2021 20:31:10 GMT'
            },
        'RetryAttempts': 0
        }
}
```

Beberapa baris pertama memberikan informasi tentang pekerjaan rekomendasi inferensi itu sendiri. Ini termasuk nama pekerjaan, peran ARN, pembuatan, dan waktu penghapusan.

`InferenceRecommendations`Kamus berisi daftar rekomendasi inferensi Inference Recommender.

Kamus `EndpointConfiguration` bersarang berisi rekomendasi instance type (`InstanceType`) bersama dengan titik akhir dan nama varian (model pembelajaran AWS mesin yang diterapkan) yang digunakan selama pekerjaan rekomendasi. Anda dapat menggunakan nama endpoint dan varian untuk pemantauan di Amazon CloudWatch Events. Untuk informasi selengkapnya, lihat Pantau Amazon SageMaker dengan Amazon CloudWatch.

Kamus `Metrics` bersarang berisi informasi tentang perkiraan biaya per jam (`CostPerHour`) untuk titik akhir real-time Anda dalam dolar AS, perkiraan biaya per inferensi (`CostPerInference`) dalam dolar AS untuk titik akhir waktu nyata Anda, jumlah maksimum yang diharapkan `InvokeEndpoint` permintaan per menit yang dikirim ke titik akhir (`MaxInvocations`), dan latensi model (`ModelLatency`), yang merupakan interval waktu (dalam milidetik) yang diambil model Anda untuk merespons. SageMaker Latensi model mencakup waktu

komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah.

Contoh berikut menunjukkan `InferenceRecommendations` bagian dari respons untuk pekerjaan rekomendasi inferensi yang dikonfigurasi untuk mengembalikan rekomendasi inferensi tanpa server:

```
"InferenceRecommendations": [
      {
         "EndpointConfiguration": {
            "EndpointName": "value",
            "InitialInstanceCount": value,
            "InstanceType": "value",
            "VariantName": "value",
            "ServerlessConfig": {
                "MaxConcurrency": value,
                "MemorySizeInMb": value
            }
         },
         "InvocationEndTime": value,
         "InvocationStartTime": value,
         "Metrics": {
            "CostPerHour": value,
            "CostPerInference": value,
            "CpuUtilization": value,
            "MaxInvocations": value,
            "MemoryUtilization": value,
            "ModelLatency": value,
            "ModelSetupTime": value
         },
         "ModelConfiguration": {
            "Compiled": "False",
            "EnvironmentParameters": [],
            "InferenceSpecificationName": "value"
         },
         "RecommendationId": "value"
      }
   ]
```

Anda dapat menafsirkan rekomendasi untuk inferensi tanpa server mirip dengan hasil untuk inferensi waktu nyata, dengan pengecualian`ServerlessConfig`, yang memberi tahu Anda metrik yang dikembalikan untuk titik akhir tanpa server dengan yang diberikan dan kapan.

`MemorySizeInMB` `MaxConcurrency = 1` Untuk meningkatkan throughput yang mungkin pada titik akhir, tingkatkan nilai linier. `MaxConcurrency` Misalnya, jika rekomendasi inferensi menunjukkan `MaxInvocations` sebagai`1000`, maka meningkat `MaxConcurrency` menjadi `2` akan mendukung 2000`MaxInvocations`. Perhatikan bahwa ini benar hanya sampai titik tertentu, yang dapat bervariasi berdasarkan model dan kode Anda. Rekomendasi tanpa server juga mengukur metrik`ModelSetupTime`, yang mengukur (dalam mikrodetik) waktu yang diperlukan untuk meluncurkan sumber daya komputer pada titik akhir tanpa server. Untuk informasi selengkapnya tentang pengaturan titik akhir tanpa server, lihat dokumentasi Inferensi Tanpa Server.

Amazon SageMaker Studio Classic

Rekomendasi inferensi terisi di tab Rekomendasi Inferensi baru dalam Studio Classic. Diperlukan waktu hingga 45 menit agar hasilnya muncul. Tab ini berisi judul kolom Hasil dan Detail.

Kolom Detail memberikan informasi tentang pekerjaan rekomendasi inferensi, seperti nama rekomendasi inferensi, kapan pekerjaan dibuat (Waktu pembuatan), dan banyak lagi. Ini juga menyediakan informasi Pengaturan, seperti jumlah maksimum pemanggilan yang terjadi per menit dan informasi tentang Nama Sumber Daya Amazon yang digunakan.

Kolom Hasil menyediakan jendela sasaran dan SageMakerrekomendasi Deployment di mana Anda dapat menyesuaikan urutan hasil yang ditampilkan berdasarkan kepentingan penerapan. Ada tiga menu tarik-turun yang dapat Anda gunakan untuk memberikan tingkat kepentingan Biaya, Latensi, dan Throughput untuk kasus penggunaan Anda. Untuk setiap tujuan (biaya, latensi, dan throughput), Anda dapat menetapkan tingkat kepentingan: Kepentingan Terendah, Kepentingan Rendah, Kepentingan sedang, Kepentingan tinggi, atau Kepentingan tertinggi.

Berdasarkan pilihan penting Anda untuk setiap tujuan, Inference Recommender menampilkan rekomendasi teratasnya di bidang SageMakerrekomendasi di sebelah kanan panel, bersama dengan perkiraan biaya per jam dan permintaan inferensi. Ini juga memberikan informasi tentang latensi model yang diharapkan, jumlah maksimum pemanggilan, dan jumlah instance. Untuk rekomendasi tanpa server, Anda dapat melihat nilai ideal untuk konkurensi maksimum dan ukuran memori titik akhir.

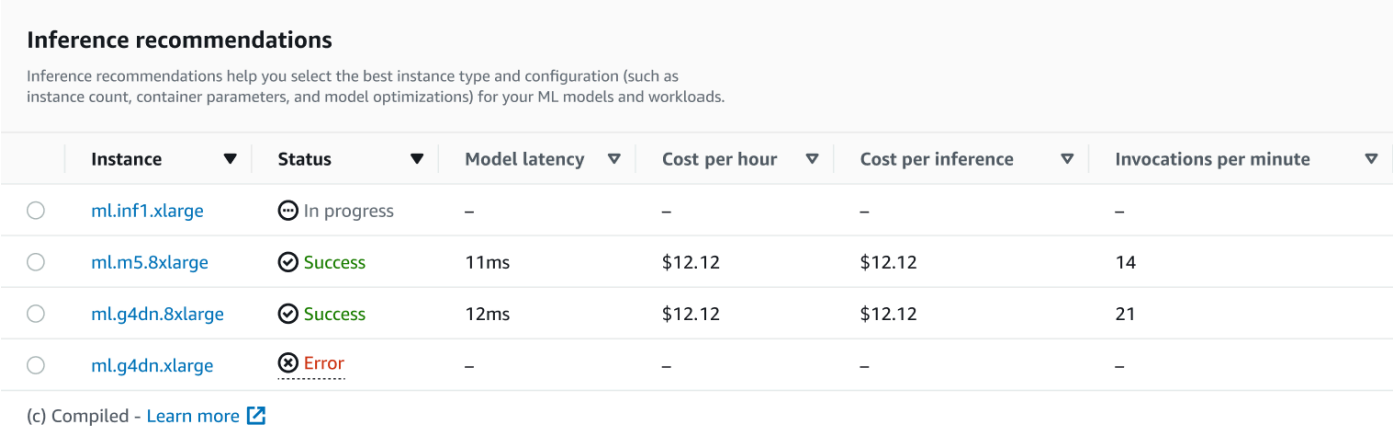
Selain rekomendasi teratas yang ditampilkan, Anda juga dapat melihat informasi yang sama ditampilkan untuk semua instance yang diuji oleh Inference Recommender di bagian Semua berjalan.

SageMaker console

Anda dapat melihat pekerjaan rekomendasi instans di SageMaker konsol dengan melakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Inference recommended.
3. Pada halaman pekerjaan pemberi rekomendasi inferensi, pilih nama pekerjaan rekomendasi inferensi Anda.

Pada halaman detail untuk pekerjaan Anda, Anda dapat melihat rekomendasi Inferensi, yang merupakan jenis instance yang SageMaker direkomendasikan untuk model Anda, seperti yang ditunjukkan pada gambar berikut.



Di bagian ini, Anda dapat membandingkan jenis instans dengan berbagai faktor seperti latensi Model, Biaya per jam, Biaya per inferensi, dan Pemanggilan per menit.

Di halaman ini, Anda juga dapat melihat konfigurasi yang Anda tentukan untuk pekerjaan Anda. Di bagian Monitor, Anda dapat melihat CloudWatch metrik Amazon yang dicatat untuk setiap jenis instans. Untuk mempelajari lebih lanjut tentang menafsirkan metrik ini, lihat Menafsirkan hasil.

Untuk informasi lebih lanjut tentang menafsirkan hasil pekerjaan rekomendasi Anda, lihatMenafsirkan hasil rekomendasi.

Hentikan rekomendasi inferensi Anda

Anda mungkin ingin menghentikan pekerjaan yang sedang berjalan jika Anda memulai pekerjaan secara tidak sengaja atau tidak perlu lagi menjalankan pekerjaan itu. Hentikan

pekerjaan rekomendasi inferensi Inference Recommender Anda secara terprogram dengan `StopInferenceRecommendationsJob` API atau dengan Studio Classic.

AWS SDK for Python (Boto3)

Tentukan nama pekerjaan rekomendasi inferensi untuk `JobName` bidang tersebut:

```
sagemaker_client.stop_inference_recommendations_job(
                                    JobName='<INSERT>'
                                    )
```

AWS CLI

Tentukan nama pekerjaan pekerjaan rekomendasi inferensi untuk `job-name` bendera:

```
aws sagemaker stop-inference-recommendations-job --job-name <job-name>
```

Amazon SageMaker Studio Classic

Tutup tab di mana Anda memulai rekomendasi inferensi untuk menghentikan rekomendasi inferensi Inference Recommender Anda.

SageMaker console

Untuk menghentikan pekerjaan rekomendasi instans Anda melalui SageMaker konsol, lakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Inference recommended.
3. Pada halaman lowongan pemberi rekomendasi inferensi, pilih lowongan rekomendasi instans Anda.
4. Pilih Stop Job.
5. Di kotak dialog yang muncul, pilih Konfirmasi.

Setelah menghentikan pekerjaan Anda, Status pekerjaan harus berubah menjadi Berhenti.

Dapatkan rekomendasi inferensi untuk titik akhir yang ada

Pekerjaan rekomendasi inferensi menjalankan serangkaian uji beban pada jenis instance yang direkomendasikan dan titik akhir yang ada. Pekerjaan rekomendasi inferensi menggunakan metrik kinerja yang didasarkan pada uji beban menggunakan data sampel yang Anda berikan selama pendaftaran versi model.

Anda dapat melakukan benchmark dan mendapatkan rekomendasi inferensi untuk titik akhir SageMaker Inferensi yang ada untuk membantu Anda meningkatkan kinerja titik akhir Anda. Prosedur mendapatkan rekomendasi untuk titik akhir SageMaker Inferensi yang ada mirip dengan prosedur untuk mendapatkan rekomendasi inferensi tanpa titik akhir. Ada beberapa pengecualian fitur yang perlu diperhatikan saat membandingkan titik akhir yang ada:

- Anda hanya dapat menggunakan satu titik akhir yang ada per pekerjaan Inference Recommender.
- Anda hanya dapat memiliki satu varian di titik akhir Anda.
- Anda tidak dapat menggunakan titik akhir yang memungkinkan penskalaan otomatis.
- Fungsi ini hanya didukung untuk Inferensi Real-Time.
- Fungsionalitas ini tidak mendukung Titik Akhir Multi-Model Real-Time.

> **Warning**
>
> Kami sangat menyarankan agar Anda tidak menjalankan pekerjaan Inference Recommender pada titik akhir produksi yang menangani lalu lintas langsung. Beban sintetis selama pembandingan dapat memengaruhi titik akhir produksi Anda dan menyebabkan pelambatan atau memberikan hasil benchmark yang tidak akurat. Kami menyarankan Anda menggunakan titik akhir non-produksi atau pengembang untuk tujuan perbandingan.

Bagian berikut menunjukkan cara menggunakan Amazon SageMaker Inference Recommender untuk membuat rekomendasi inferensi untuk titik akhir yang ada berdasarkan jenis model Anda menggunakan AWS SDK for Python (Boto3) dan file. AWS CLI

> **Note**
>
> Sebelum Anda membuat pekerjaan rekomendasi Inference Recommender, pastikan Anda telah puas. Prasyarat

Prasyarat

Jika Anda belum memiliki titik akhir SageMaker Inferensi, Anda bisa mendapatkan rekomendasi inferensi tanpa titik akhir, atau Anda dapat membuat titik akhir Inferensi Waktu Nyata dengan mengikuti instruksi di Buat titik akhir Anda dan terapkan model Anda.

Buat pekerjaan rekomendasi inferensi untuk titik akhir yang ada

Buat rekomendasi inferensi secara terprogram menggunakanAWS SDK for Python (Boto3), atau. AWS CLI Tentukan nama pekerjaan untuk rekomendasi inferensi Anda, nama titik akhir SageMaker Inferensi yang ada, ARN peran AWS IAM, konfigurasi input, dan ARN paket model Anda dari saat Anda mendaftarkan model Anda dengan registri model.

AWS SDK for Python (Boto3)

Gunakan `CreateInferenceRecommendationsJob`API untuk mendapatkan rekomendasi inferensi. Tetapkan `JobType` bidang `'Default'` untuk pekerjaan rekomendasi inferensi. Selain itu, berikan yang berikut:

- Berikan nama untuk pekerjaan rekomendasi Inference Recommender Anda untuk bidang tersebut. `JobName` Nama pekerjaan Inference Recommender harus unik di dalam AWS Wilayah dan di dalam akun AndaAWS.
- Nama Sumber Daya Amazon (ARN) dari peran IAM yang memungkinkan Inference Recommender untuk melakukan tugas atas nama Anda. Tentukan ini untuk `RoleArn` bidang.
- ARN dari paket model berversi yang Anda buat saat Anda mendaftarkan model Anda dengan registri model. Tentukan ini untuk `ModelPackageVersionArn` di `InputConfig` lapangan.
- Berikan nama titik akhir SageMaker Inferensi yang ada yang ingin Anda benchmark di Inference Recommender di lapangan. `Endpoints` `InputConfig`

Impor AWS SDK for Python (Boto3) paket dan buat objek SageMaker klien menggunakan kelas klien. Jika Anda mengikuti langkah-langkah di bagian Prasyarat, kelompok paket model ARN disimpan dalam variabel bernama. `model_package_arn`

```
# Create a low-level SageMaker service client.
import boto3
aws_region = '<region>'
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Provide your model package ARN that was created when you registered your
```

```
# model with Model Registry
model_package_arn = '<model-package-arn>'

# Provide a unique job name for SageMaker Inference Recommender job
job_name = '<job-name>'

# Inference Recommender job type. Set to Default to get an initial recommendation
job_type = 'Default'

# Provide an IAM Role that gives SageMaker Inference Recommender permission to
# access AWS services
role_arn = '<arn:aws:iam::<account>:role/*>'

# Provide endpoint name for your endpoint that want to benchmark in Inference
 Recommender
endpoint_name = '<existing-endpoint-name>'

sagemaker_client.create_inference_recommendations_job(
    JobName = job_name,
    JobType = job_type,
    RoleArn = role_arn,
    InputConfig = {
        'ModelPackageVersionArn': model_package_arn,
        'Endpoints': [{'EndpointName': endpoint_name}]
    }
)
```

Lihat Panduan Referensi Amazon SageMaker API untuk daftar lengkap argumen opsional dan wajib yang dapat Anda berikan `CreateInferenceRecommendationsJob`.

AWS CLI

Gunakan `create-inference-recommendations-job` API untuk mendapatkan rekomendasi titik akhir instance. Tetapkan `job-type` bidang untuk `'Default'` misalnya pekerjaan rekomendasi titik akhir. Selain itu, berikan yang berikut:

- Berikan nama untuk pekerjaan rekomendasi Inference Recommender Anda untuk bidang tersebut. `job-name` Nama pekerjaan Inference Recommender harus unik di dalam AWS Wilayah dan di dalam akun AndaAWS.
- Nama Sumber Daya Amazon (ARN) dari peran IAM yang memungkinkan Amazon SageMaker Inference Recommender untuk melakukan tugas atas nama Anda. Tentukan ini untuk `role-arn` bidang.

- ARN dari paket model berversi yang Anda buat saat Anda mendaftarkan model Anda dengan Model Registry. Tentukan ini untuk `ModelPackageVersionArn` di `input-config` lapangan.
- Berikan nama titik akhir SageMaker Inferensi yang ada yang ingin Anda benchmark di Inference Recommender di lapangan. `Endpoints` `input-config`

```
aws sagemaker create-inference-recommendations-job
    --region <region>\
    --job-name <job_name>\
    --job-type Default\
    --role-arn arn:aws:iam::<account:role/*>\
    --input-config "{
        \"ModelPackageVersionArn\": \"arn:aws:sagemaker:<region:account:role/*>\",
        \"Endpoints\": [{\"EndpointName\": <endpoint_name>}]
        }"
```

Dapatkan hasil pekerjaan rekomendasi inferensi Anda

Anda dapat mengumpulkan hasil pekerjaan rekomendasi inferensi Anda secara terprogram dengan prosedur yang sama untuk pekerjaan rekomendasi inferensi standar. Untuk informasi selengkapnya, lihat Dapatkan hasil pekerjaan rekomendasi inferensi Anda.

Ketika Anda mendapatkan hasil pekerjaan rekomendasi inferensi untuk titik akhir yang ada, Anda harus menerima respons JSON yang mirip dengan berikut ini:

```
{
    "JobName": "job-name",
    "JobType": "Default",
    "JobArn": "arn:aws:sagemaker:region:account-id:inference-recommendations-
job/resource-id",
    "RoleArn": "iam-role-arn",
    "Status": "COMPLETED",
    "CreationTime": 1664922919.2,
    "LastModifiedTime": 1664924208.291,
    "InputConfig": {
        "ModelPackageVersionArn": "arn:aws:sagemaker:region:account-id:model-
package/resource-id",
        "Endpoints": [
            {
                "EndpointName": "endpoint-name"
            }
```

```
        ]
    },
    "InferenceRecommendations": [
        {
            "Metrics": {
                "CostPerHour": 0.7360000014305115,
                "CostPerInference": 7.456940238625975e-06,
                "MaxInvocations": 1645,
                "ModelLatency": 171
            },
            "EndpointConfiguration": {
                "EndpointName": "sm-endpoint-name",
                "VariantName": "variant-name",
                "InstanceType": "ml.g4dn.xlarge",
                "InitialInstanceCount": 1
            },
            "ModelConfiguration": {
                "EnvironmentParameters": [
                    {
                        "Key": "TS_DEFAULT_WORKERS_PER_MODEL",
                        "ValueType": "string",
                        "Value": "4"
                    }
                ]
            }
        }
    ],
    "EndpointPerformances": [
        {
            "Metrics": {
                "MaxInvocations": 184,
                "ModelLatency": 1312
            },
            "EndpointConfiguration": {
                "EndpointName": "endpoint-name"
            }
        }
    ]
}
```

Beberapa baris pertama memberikan informasi tentang pekerjaan rekomendasi inferensi itu sendiri. Ini termasuk nama pekerjaan, peran ARN, dan pembuatan dan waktu modifikasi terbaru.

`InferenceRecommendations`Kamus berisi daftar rekomendasi inferensi Inference Recommender.

Kamus `EndpointConfiguration` bersarang berisi rekomendasi tipe instance (`InstanceType`) bersama dengan titik akhir dan nama varian (model pembelajaran AWS mesin yang diterapkan) yang digunakan selama pekerjaan rekomendasi.

Kamus `Metrics` bersarang berisi informasi tentang perkiraan biaya per jam (`CostPerHour`) untuk titik akhir real-time Anda dalam dolar AS, perkiraan biaya per inferensi (`CostPerInference`) dalam dolar AS untuk titik akhir waktu nyata Anda, jumlah maksimum yang diharapkan `InvokeEndpoint` permintaan per menit yang dikirim ke titik akhir (`MaxInvocations`), dan latensi model (`ModelLatency`), yang merupakan interval waktu (dalam milidetik) yang diambil model Anda untuk merespons. SageMaker Latensi model mencakup waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah.

Kamus `EndpointPerformances` bersarang berisi nama titik akhir yang ada tempat tugas rekomendasi dijalankan (`EndpointName`) dan metrik kinerja untuk titik akhir Anda (dan). `MaxInvocations` `ModelLatency`

Hentikan rekomendasi titik akhir instans Anda

Anda mungkin ingin menghentikan pekerjaan yang sedang berjalan jika Anda memulai pekerjaan secara tidak sengaja atau tidak perlu lagi menjalankan pekerjaan itu. Anda dapat menghentikan pekerjaan rekomendasi Inference Recommender Anda secara terprogram dengan prosedur yang sama untuk pekerjaan rekomendasi inferensi standar. Untuk informasi selengkapnya, lihat Hentikan rekomendasi inferensi Anda.

Dapatkan rekomendasi yang dikompilasi dengan Neo

Di Inference Recommender, Anda dapat mengkompilasi model Anda dengan Neo dan mendapatkan rekomendasi endpoint untuk model kompilasi Anda. SageMaker Neo adalah layanan yang dapat mengoptimalkan model Anda untuk platform perangkat keras target (yaitu, jenis atau lingkungan instance tertentu). Mengoptimalkan model dengan Neo dapat meningkatkan kinerja model host Anda.

Untuk kerangka kerja dan kontainer yang didukung NEO, Inference Recommender secara otomatis menyarankan rekomendasi yang dioptimalkan oleh NEO. Agar memenuhi syarat untuk kompilasi Neo, masukan Anda harus memenuhi prasyarat berikut:

- Anda menggunakan wadah DLC atau XGBoost yang SageMaker dimiliki.

- Anda menggunakan versi kerangka kerja yang didukung oleh Neo. Untuk versi kerangka kerja yang didukung oleh Neo, lihat Instans Cloud di dokumentasi SageMaker Neo.
- Neo mengharuskan Anda memberikan bentuk data input yang benar untuk model Anda. Anda dapat menentukan bentuk data ini seperti `DataInputConfig` pada `InferenceSpecification` saat Anda membuat paket model. Untuk informasi tentang bentuk data yang benar untuk setiap framework, lihat Mempersiapkan Model untuk Kompilasi dalam dokumentasi SageMaker Neo.

  Contoh berikut menunjukkan bagaimana menentukan `DataInputConfig` bidang di`InferenceSpecification`, di mana `data_input_configuration` adalah variabel yang berisi bentuk data dalam format kamus (misalnya,`{'input':[1,1024,1024,3]}`).

```
"InferenceSpecification": {
        "Containers": [
            {
                "Image": dlc_uri,
                "Framework": framework.upper(),
                "FrameworkVersion": framework_version,
                "NearestModelName": model_name,
                "ModelInput": {"DataInputConfig": data_input_configuration},
            }
        ],
        "SupportedContentTypes": input_mime_types,  # required, must be non-null
        "SupportedResponseMIMETypes": [],
        "SupportedRealtimeInferenceInstanceTypes":
 supported_realtime_inference_types,  # optional
    }
```

Jika kondisi ini terpenuhi dalam permintaan Anda, maka Inference Recommender menjalankan skenario untuk versi model Anda yang dikompilasi dan tidak dikompilasi, memberi Anda beberapa kombinasi rekomendasi untuk dipilih. Anda dapat membandingkan konfigurasi untuk versi yang dikompilasi dan tidak dikompilasi dari rekomendasi inferensi yang sama dan menentukan mana yang paling sesuai dengan kasus penggunaan Anda. Rekomendasi diberi peringkat berdasarkan biaya per inferensi.

Untuk mendapatkan rekomendasi kompilasi Neo, Anda tidak perlu melakukan konfigurasi tambahan selain memastikan bahwa input Anda memenuhi persyaratan di atas. Inference Recommender secara otomatis menjalankan kompilasi Neo pada model Anda jika input Anda memenuhi persyaratan, dan Anda akan menerima respons yang menyertakan rekomendasi Neo.

Jika Anda mengalami kesalahan selama kompilasi Neo Anda, lihatMemecahkan Masalah Kesalahan Kompilasi Neo.

Tabel berikut adalah contoh respons yang mungkin Anda dapatkan dari pekerjaan Inference Recommender yang mencakup rekomendasi untuk model yang dikompilasi. Jika `InferenceSpecificationName` bidangnya`None`, maka rekomendasinya adalah model yang tidak dikompilasi. Baris terakhir, di mana nilai untuk `InferenceSpecificationName` bidang tersebut`neo-00011122-2333-4445-5566-677788899900`, adalah untuk model yang dikompilasi dengan Neo. Nilai di bidang adalah nama pekerjaan Neo yang digunakan untuk mengkompilasi dan mengoptimalkan model Anda.

EndpointName	InstanceType	InitialInstanceCount	EnvironmentParameters	CostPerHour	CostPerInference	MaxInvocations	ModelLatency	InferenceSpecificationName
sm-epc-example-00111222	ml.c5.9xlarge	1	[]	1.836	9.15E-07	33456	7	Tidak ada
sm-epc-example-111222333	ml.c5.2xlarge	1	[]	0,408	2.11E-07	32211	21	Tidak ada
sm-epc-example-222333444	ml.c5.xlarge	1	[]	0,204	1.86E-07	18276	92	Tidak ada
sm-epc-example-333444555	ml.c5.xlarge	1	[]	0,204	1.60E-07	21286	42	neo-00011122-2333-4445-5566-677788899900

Memulai

Langkah-langkah umum untuk membuat pekerjaan Inference Recommender yang mencakup rekomendasi yang dioptimalkan NEO adalah sebagai berikut:

- Siapkan model ML Anda untuk kompilasi. Untuk informasi lebih lanjut, lihat Mempersiapkan Model untuk Kompilasi dalam dokumentasi Neo.
- Package model Anda dalam arsip model (`.tar.gz`file).
- Buat contoh arsip payload.
- Daftarkan model Anda di Model Registry.
- Buat pekerjaan Inference Recommender.
- Lihat hasil pekerjaan Inference Recommender dan pilih konfigurasi.
- Kegagalan kompilasi debug, jika ada. Untuk informasi selengkapnya, lihat Memecahkan Masalah Kesalahan Kompilasi Neo.

Untuk contoh yang menunjukkan alur kerja sebelumnya dan cara mendapatkan rekomendasi yang dioptimalkan NEO menggunakan XGBoost, lihat contoh buku catatan berikut. Untuk contoh yang menunjukkan cara mendapatkan rekomendasi yang dioptimalkan NEO menggunakan TensorFlow, lihat contoh buku catatan berikut.

Menafsirkan hasil rekomendasi

Setiap hasil pekerjaan Inference Recommender mencakup`InstanceType`,, dan `InitialInstanceCountEnvironmentParameters`, yang merupakan parameter variabel lingkungan yang disetel untuk penampung Anda guna meningkatkan latensi dan throughputnya. Hasilnya juga mencakup metrik kinerja dan biaya seperti`MaxInvocations`,,`ModelLatency`,`CostPerHour`, `CostPerInferenceCpuUtilization`, dan`MemoryUtilization`.

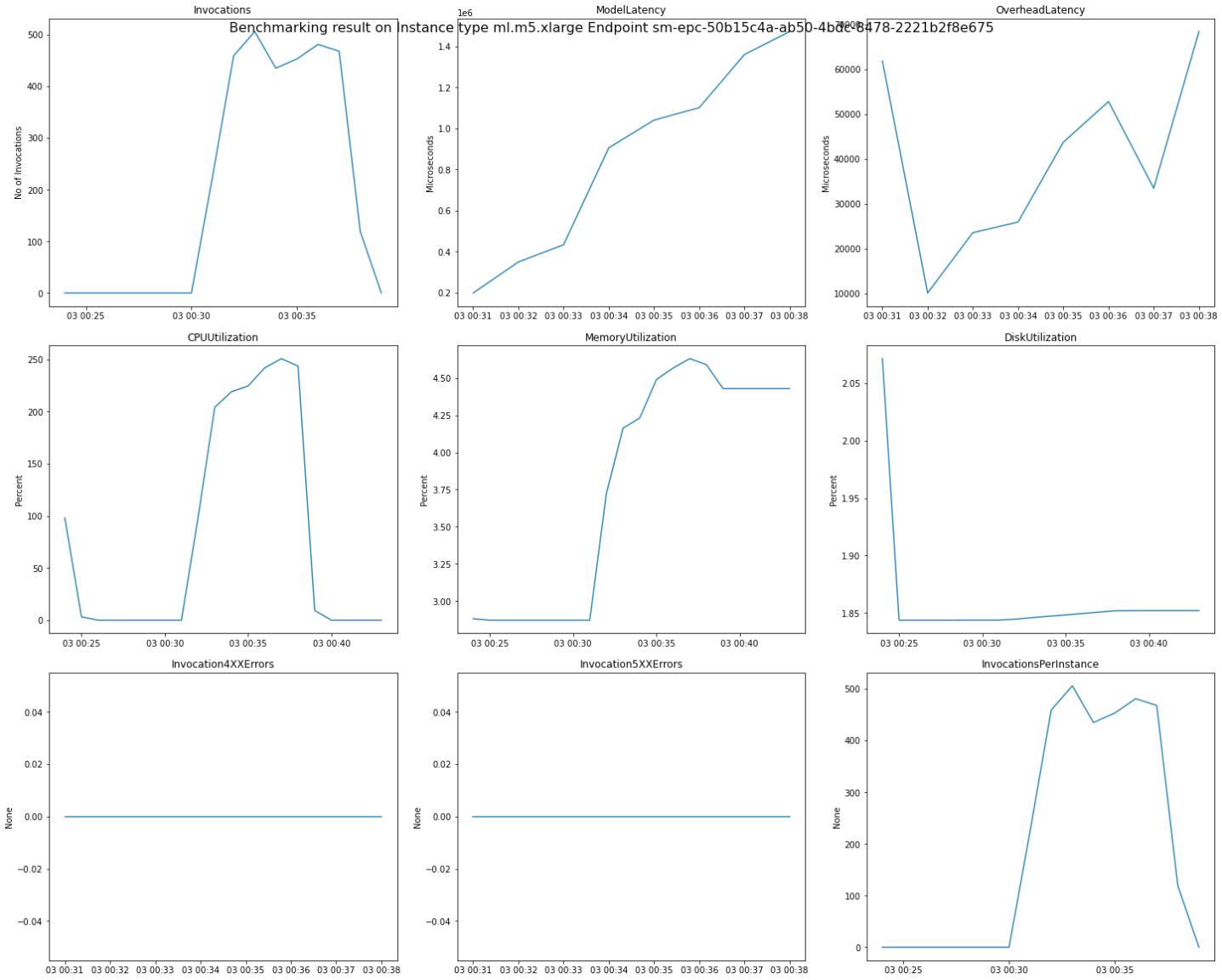
Pada tabel di bawah ini kami memberikan deskripsi metrik ini. Metrik ini dapat membantu Anda mempersempit pencarian Anda untuk konfigurasi titik akhir terbaik yang sesuai dengan kasus penggunaan Anda. Misalnya, jika motivasi Anda adalah kinerja harga secara keseluruhan dengan penekanan pada throughput, maka Anda harus fokus pada`CostPerInference`.

Metrik	Deskripsi	Kasus penggunaan
`ModelLatency`	Interval waktu yang dibutuhkan oleh model untuk merespons sebagaimana dilihat dari SageMaker. Interval ini mencakup waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah. Unit: Milidetik	Beban kerja sensitif latensi seperti penayangan iklan dan diagnosis medis
`MaximumInvocations`	Jumlah maksimum `InvokeEndpoint` permintaan yang dikirim ke titik akhir model dalam satu menit. Satuan: Tidak ada	Beban kerja yang berfokus pada throughput seperti pemrosesan video atau inferensi batch
`CostPerHour`	Perkiraan biaya per jam untuk titik akhir real-time Anda. Unit: Dolar AS	Beban kerja yang sensitif terhadap biaya tanpa tenggat waktu latensi
`CostPerInference`	Perkiraan biaya per panggilan inferensi untuk titik akhir real-time Anda. Unit: Dolar AS	Maksimalkan kinerja harga secara keseluruhan dengan fokus pada throughput
`CpuUtilization`	Pemanfaatan CPU yang diharapkan pada pemanggilan maksimum per menit untuk instance titik akhir.	Memahami kesehatan instans selama benchmarking dengan memiliki visibilitas ke

Metrik	Deskripsi	Kasus penggunaan
	Unit: Persen	dalam pemanfaatan CPU inti instance
`MemoryUtilization`	Pemanfaatan memori yang diharapkan pada pemanggil an maksimum per menit untuk instance titik akhir. Unit: Persen	Memahami kesehatan instance selama pembandin gan dengan memiliki visibilit as ke dalam pemanfaatan memori inti instance

Dalam beberapa kasus, Anda mungkin ingin menjelajahi metrik Pemanggilan SageMaker Titik Akhir lainnya seperti. `CPUUtilization` Setiap hasil pekerjaan Inference Recommender mencakup nama-nama titik akhir yang diputar selama uji beban. Anda dapat menggunakan CloudWatch untuk meninjau log untuk titik akhir ini bahkan setelah dihapus.

Gambar berikut adalah contoh CloudWatch metrik dan bagan yang dapat Anda tinjau untuk satu titik akhir dari hasil rekomendasi Anda. Hasil rekomendasi ini berasal dari pekerjaan Default. Cara untuk menafsirkan nilai skalar dari hasil rekomendasi adalah bahwa mereka didasarkan pada titik waktu ketika grafik Invocations pertama kali mulai naik level. Misalnya, `ModelLatency` nilai yang dilaporkan berada di awal dataran tinggi sekitar. `03:00:31`



[Untuk deskripsi lengkap tentang CloudWatch metrik yang digunakan dalam bagan sebelumnya, lihat SageMaker Metrik Pemanggilan Titik Akhir.]

Anda juga dapat melihat metrik kinerja seperti `ClientInvocations` dan `NumberOfUsers` diterbitkan oleh Inference Recommender di namespace. `/aws/sagemaker/InferenceRecommendationsJobs` Untuk daftar lengkap metrik dan deskripsi yang diterbitkan oleh Inference Recommender, lihat. [SageMaker Metrik Lowongan Inference Recommender]

Lihat [Amazon SageMaker Inference Recommender - CloudWatch Metrics] Jupyter notebook di repositori [amazon-sagemaker-examples]Github untuk contoh cara menggunakan SDK for Python AWS (Boto3) untuk menjelajahi metrik untuk titik akhir Anda. CloudWatch

Dapatkan rekomendasi kebijakan penskalaan otomatis

Dengan Amazon SageMaker Inference Recommender, Anda bisa mendapatkan rekomendasi untuk kebijakan penskalaan otomatis untuk SageMaker titik akhir Anda berdasarkan pola lalu lintas yang Anda antisipasi. Jika Anda telah menyelesaikan pekerjaan rekomendasi inferensi, Anda dapat memberikan rincian pekerjaan untuk mendapatkan rekomendasi untuk kebijakan penskalaan otomatis yang dapat Anda terapkan ke titik akhir Anda.

Inference Recommender membandingkan nilai yang berbeda untuk setiap metrik untuk menentukan konfigurasi penskalaan otomatis yang ideal untuk titik akhir Anda. Rekomendasi penskalaan otomatis mengembalikan kebijakan penskalaan otomatis yang direkomendasikan untuk setiap metrik yang ditentukan dalam pekerjaan rekomendasi inferensi Anda. Anda dapat menyimpan kebijakan dan menerapkannya ke titik akhir Anda dengan PutScalingPolicyAPI.

Untuk memulai, tinjau prasyarat berikut.

Prasyarat

Sebelum Anda mulai, Anda harus telah menyelesaikan pekerjaan rekomendasi inferensi yang sukses. Di bagian berikut, Anda dapat memberikan ID rekomendasi inferensi atau nama SageMaker titik akhir yang dibenchmark selama pekerjaan rekomendasi inferensi.

Untuk mengambil ID lowongan rekomendasi atau nama titik akhir, Anda dapat melihat detail pekerjaan rekomendasi inferensi di SageMaker konsol, atau Anda dapat menggunakan `EndpointName` kolom `RecommendationId` atau yang ditampilkan oleh API. DescribeInferenceRecommendationsJob

Buat rekomendasi konfigurasi penskalaan otomatis

Untuk membuat kebijakan rekomendasi penskalaan otomatis, Anda dapat menggunakan. AWS SDK for Python (Boto3)

Contoh berikut menunjukkan bidang untuk GetScalingConfigurationRecommendationAPI. Gunakan bidang berikut saat Anda memanggil API:

- `InferenceRecommendationsJobName`— Masukkan nama pekerjaan rekomendasi inferensi Anda.
- `RecommendationId`— Masukkan ID rekomendasi inferensi dari pekerjaan rekomendasi. Ini opsional jika Anda telah menentukan `EndpointName` bidang.

- `EndpointName`— Masukkan nama titik akhir yang telah dipatokan selama pekerjaan rekomendasi inferensi. Ini opsional jika Anda telah menentukan `RecommendationId` bidang.
- `TargetCpuUtilizationPerCore`— (Opsional) Masukkan nilai persentase berapa banyak pemanfaatan yang Anda inginkan sebuah instance pada titik akhir Anda untuk digunakan sebelum penskalaan otomatis. Nilai default jika Anda tidak menentukan bidang ini adalah 50%.
- `ScalingPolicyObjective`— (Opsional) Objek tempat Anda menentukan pola lalu lintas yang diantisipasi.
  - `MinInvocationsPerMinute`— (Opsional) Jumlah minimum permintaan yang diharapkan ke titik akhir Anda per menit.
  - `MaxInvocationsPerMinute`— (Opsional) Jumlah maksimum permintaan yang diharapkan ke titik akhir Anda per menit.

```
{
    "InferenceRecommendationsJobName": "string", // Required
    "RecommendationId": "string", // Optional, provide one of RecommendationId or
 EndpointName
    "EndpointName": "string", // Optional, provide one of RecommendationId or
 EndpointName
    "TargetCpuUtilizationPerCore": number, // Optional
    "ScalingPolicyObjective": { // Optional
        "MinInvocationsPerMinute": number,
        "MaxInvocationsPerMinute": number
    }
}
```

Setelah mengirimkan permintaan, Anda akan menerima respons dengan kebijakan penskalaan otomatis yang ditentukan untuk setiap metrik. Lihat bagian berikut untuk informasi tentang menafsirkan respons.

Tinjau hasil rekomendasi konfigurasi penskalaan otomatis Anda

Contoh berikut menunjukkan respons dari GetScalingConfigurationRecommendationAPI:

```
{
    "InferenceRecommendationsJobName": "string",
    "RecommendationId": "string", // One of RecommendationId or EndpointName is shown
    "EndpointName": "string",
    "TargetUtilizationPercentage": Integer,
    "ScalingPolicyObjective": {
```

```
        "MinInvocationsPerMinute": Integer,
        "MaxInvocationsPerMinute": Integer
    },
    "Metric": {
        "ModelLatency": Integer,
        "InvocationsPerInstance": Integer
    },
    "DynamicScalingConfiguration": {
        "MinCapacity": number,
        "MaxCapacity": number,
        "ScaleInCooldown": number,
        "ScaleOutCooldown": number,
        "ScalingPolicies": [
            {
                "TargetTracking": {
                    "MetricSpecification": {
                        "Predefined" {
                            "PredefinedMetricType": "string"
                         },
                        "Customized": {
                            "MetricName": "string",
                            "Namespace": "string",
                            "Statistic": "string"
                         }
                    },
                    "TargetValue": Double
                }
            }
        ]
    }
}
```

Bidang `InferenceRecommendationsJobName`, `EndpointName`, `RecommendationID` atau`TargetCpuUtilizationPerCore`,, dan `ScalingPolicyObjective` objek disalin dari permintaan awal Anda.

`Metric`Objek mencantumkan metrik yang dibenchmark dalam pekerjaan rekomendasi inferensi Anda, bersama dengan perhitungan nilai untuk setiap metrik ketika pemanfaatan instance akan sama dengan nilainya. `TargetCpuUtilizationPerCore` Hal ini berguna untuk mengantisipasi metrik kinerja pada titik akhir Anda saat menskalakan masuk dan keluar dengan kebijakan penskalaan otomatis yang direkomendasikan. Misalnya, pertimbangkan apakah pemanfaatan instans Anda adalah 50% dalam pekerjaan rekomendasi inferensi Anda dan `InvocationsPerInstance`

nilai Anda pada awalnya. 4 Jika Anda menentukan `TargetCpuUtilizationPerCore` nilai menjadi 100% dalam permintaan rekomendasi penskalaan otomatis Anda, maka nilai `InvocationsPerInstance` metrik yang dikembalikan dalam respons adalah 2 karena Anda mengantisipasi mengalokasikan penggunaan instance dua kali lebih banyak.

`DynamicScalingConfiguration`Objek mengembalikan nilai yang harus Anda tentukan TargetTrackingScalingPolicyConfigurationsaat Anda memanggil PutScalingPolicyAPI. Ini termasuk nilai kapasitas minimum dan maksimum yang disarankan, waktu cooldown skala masuk dan skala yang disarankan, dan `ScalingPolicies` objek, yang berisi rekomendasi yang harus `TargetValue` Anda tentukan untuk setiap metrik.

Jalankan uji beban khusus

Pengujian beban Amazon SageMaker Inference Recommender melakukan tolok ukur ekstensif berdasarkan persyaratan produksi untuk latensi dan throughput, pola lalu lintas khusus, dan titik akhir tanpa server atau instans real-time (hingga 10) yang Anda pilih.

Bagian berikut menunjukkan cara membuat, mendeskripsikan, dan menghentikan uji beban secara terprogram menggunakan AWS SDK for Python (Boto3) danAWS CLI, atau secara interaktif menggunakan Amazon SageMaker Studio Classic atau konsol. SageMaker

Buat pekerjaan uji beban

Buat uji beban secara terprogram menggunakanAWS SDK for Python (Boto3), denganAWS CLI, atau secara interaktif menggunakan Studio Classic atau konsol. SageMaker Seperti rekomendasi inferensi Inference Recommender, tentukan nama pekerjaan untuk uji beban Anda, ARN peran AWS IAM, konfigurasi input, dan ARN paket model Anda sejak Anda mendaftarkan model Anda dengan registri model. Tes beban mengharuskan Anda juga menentukan pola lalu lintas dan kondisi penghentian.

AWS SDK for Python (Boto3)

Gunakan `CreateInferenceRecommendationsJob` API untuk membuat uji beban Inference Recommender. Tentukan `Advanced` untuk `JobType` bidang dan berikan:

- Nama pekerjaan untuk uji beban Anda (`JobName`). Nama pekerjaan harus unik di AWS Wilayah Anda dan di dalam AWS akun Anda.
- Nama Sumber Daya Amazon (ARN) dari peran IAM yang memungkinkan Inference Recommender untuk melakukan tugas atas nama Anda. Tentukan ini untuk `RoleArn` bidang.
- Kamus konfigurasi titik akhir (`InputConfig`) tempat Anda menentukan yang berikut:

- Untuk`TrafficPattern`, tentukan fase atau pola lalu lintas tangga. Dengan pola lalu lintas fase, pengguna baru muncul setiap menit dengan kecepatan yang Anda tentukan. Dengan pola lalu lintas tangga, pengguna baru muncul pada interval waktu (atau langkah) pada tingkat yang Anda tentukan. Pilih salah satu cara berikut:
  - Untuk `TrafficType`, tentukan `PHASES`. Kemudian, untuk `Phases` array, tentukan `InitialNumberOfUsers` (berapa banyak pengguna bersamaan untuk memulai, dengan minimal 1 dan maksimum 3), `SpawnRate` (jumlah pengguna yang akan muncul dalam satu menit untuk fase pengujian beban tertentu, dengan minimal 0 dan maksimum 3), dan `DurationInSeconds` (berapa lama fase lalu lintas seharusnya, dengan minimum 120 dan maksimum 3600).
  - Untuk `TrafficType`, tentukan `STAIRS`. Kemudian, untuk `Stairs` array, tentukan `DurationInSeconds` (berapa lama fase lalu lintas seharusnya, dengan minimum 120 dan maksimum 3600), `NumberOfSteps` (berapa banyak interval yang digunakan selama fase), dan `UsersPerStep` (berapa banyak pengguna yang ditambahkan selama setiap interval). Perhatikan bahwa panjang setiap langkah adalah nilai`DurationInSeconds / NumberOfSteps`. Misalnya, jika Anda `600` dan Anda `DurationInSeconds` menentukan 5 langkah-langkahnya, maka setiap langkah berdurasi 120 detik.

    > **Note**
    >
    > Seorang pengguna didefinisikan sebagai aktor yang dihasilkan sistem yang berjalan dalam satu lingkaran dan memanggil permintaan ke titik akhir sebagai bagian dari Inference Recommender. Untuk wadah XGBoost biasa yang berjalan pada sebuah `ml.c5.large` instance, titik akhir dapat mencapai 30.000 pemanggilan per menit (500 tps) hanya dengan 15-20 pengguna.

- Untuk`ResourceLimit`, tentukan `MaxNumberOfTests` (jumlah maksimum uji beban benchmarking untuk pekerjaan Inference Recommender, dengan minimal 1 dan maksimum 10) dan `MaxParallelOfTests` (jumlah maksimum uji beban benchmarking paralel untuk pekerjaan Inference Recommender, dengan minimal 1 dan maksimum 10).
- Untuk`EndpointConfigurations`, Anda dapat menentukan salah satu dari berikut ini:
  - `InstanceType`Bidang, tempat Anda menentukan jenis instance tempat Anda ingin menjalankan pengujian beban.
  - The`ServerlessConfig`, di mana Anda menentukan nilai ideal Anda untuk `MaxConcurrency` dan `MemorySizeInMB` untuk titik akhir tanpa server. Untuk informasi selengkapnya, lihat dokumentasi Inferensi Tanpa Server.

- Kamus kondisi berhenti (`StoppingConditions`), di mana jika salah satu kondisi terpenuhi, pekerjaan Inference Recommender berhenti. Untuk contoh ini, tentukan bidang berikut dalam kamus:
  - Untuk`MaxInvocations`, tentukan jumlah maksimum permintaan per menit yang diharapkan untuk titik akhir, dengan minimum 1 dan maksimum 30.000.
  - Untuk`ModelLatencyThresholds`, tentukan `Percentile` (ambang persentil latensi model) dan `ValueInMilliseconds` (nilai persentil latensi model dalam milidetik).
  - (Opsional) Untuk`FlatInvocations`, Anda dapat menentukan apakah akan melanjutkan uji beban ketika tingkat TPS (pemanggilan per menit) rata. Tingkat TPS yang diratakan biasanya berarti bahwa titik akhir telah mencapai kapasitas. Namun, Anda mungkin ingin terus memantau titik akhir dalam kondisi kapasitas penuh. Untuk melanjutkan uji beban ketika ini terjadi, tentukan nilai ini sebagai`Continue`. Jika tidak, nilai defaultnya adalah `Stop`.

```
# Create a low-level SageMaker service client.
import boto3
aws_region=<INSERT>
sagemaker_client=boto3.client('sagemaker', region=aws_region)

# Provide a name to your recommendation based on load testing
load_test_job_name="<INSERT>"

# Provide the name of the sagemaker instance type
instance_type="<INSERT>"

# Provide the IAM Role that gives SageMaker permission to access AWS services
role_arn='arn:aws:iam::<account>:role/*'

# Provide your model package ARN that was created when you registered your
# model with Model Registry
model_package_arn='arn:aws:sagemaker:<region>:<account>:role/*'

sagemaker_client.create_inference_recommendations_job(
                        JobName=load_test_job_name,
                        JobType="Advanced",
                        RoleArn=role_arn,
                        InputConfig={
                            'ModelPackageVersionArn': model_package_arn,
                            "JobDurationInSeconds": 7200,
                            'TrafficPattern' : {
```

```
                                # Replace PHASES with STAIRS to use the stairs
traffic pattern
                                'TrafficType': 'PHASES',
                                'Phases': [
                                    {
                                        'InitialNumberOfUsers': 1,
                                        'SpawnRate': 1,
                                        'DurationInSeconds': 120
                                    },
                                    {
                                        'InitialNumberOfUsers': 1,
                                        'SpawnRate': 1,
                                        'DurationInSeconds': 120
                                    }
                                ]
                                # Uncomment this section and comment out the Phases
object above to use the stairs traffic pattern
                                # 'Stairs' : {
                                #   'DurationInSeconds': 240,
                                #   'NumberOfSteps': 2,
                                #   'UsersPerStep': 2
                                # }
                            },
                            'ResourceLimit': {
                                        'MaxNumberOfTests': 10,
                                        'MaxParallelOfTests': 3
                                },
                            "EndpointConfigurations" : [{
                                        'InstanceType': 'ml.c5.xlarge'
                                    },
                                    {
                                        'InstanceType': 'ml.m5.xlarge'
                                    },
                                    {
                                        'InstanceType': 'ml.r5.xlarge'
                                    }]
                                    # Uncomment the ServerlessConfig and comment out
the InstanceType field if you want recommendations for a serverless endpoint
                                    # "ServerlessConfig": {
                                    #     "MaxConcurrency": value,
                                    #     "MemorySizeInMB": value
                                    # }
                        },
                        StoppingConditions={
```

```
                            'MaxInvocations': 1000,
                            'ModelLatencyThresholds':[{
                                'Percentile': 'P95',
                                'ValueInMilliseconds': 100
                            }],
                            # Change 'Stop' to 'Continue' to let the load test
continue if invocations flatten
                            'FlatInvocations': 'Stop'
                        }
                )
```

Lihat Panduan Referensi Amazon SageMaker API untuk daftar lengkap argumen opsional dan wajib yang dapat Anda berikan`CreateInferenceRecommendationsJob`.

AWS CLI

Gunakan `create-inference-recommendations-job` API untuk membuat uji beban Inference Recommender. Tentukan `Advanced` untuk `JobType` bidang dan berikan:

- Nama pekerjaan untuk uji beban Anda (`job-name`). Nama pekerjaan harus unik di AWS Wilayah Anda dan di dalam AWS akun Anda.
- Nama Sumber Daya Amazon (ARN) dari peran IAM yang memungkinkan Inference Recommender untuk melakukan tugas atas nama Anda. Tentukan ini untuk `role-arn` bidang.
- Kamus konfigurasi titik akhir (`input-config`) tempat Anda menentukan yang berikut:
  - Untuk`TrafficPattern`, tentukan fase atau pola lalu lintas tangga. Dengan pola lalu lintas fase, pengguna baru muncul setiap menit dengan kecepatan yang Anda tentukan. Dengan pola lalu lintas tangga, pengguna baru muncul pada interval waktu (atau langkah) pada tingkat yang Anda tentukan. Pilih salah satu cara berikut:
    - Untuk `TrafficType`, tentukan `PHASES`. Kemudian, untuk `Phases` array, tentukan `InitialNumberOfUsers` (berapa banyak pengguna bersamaan untuk memulai, dengan minimal 1 dan maksimum 3), `SpawnRate` (jumlah pengguna yang akan muncul dalam satu menit untuk fase pengujian beban tertentu, dengan minimal 0 dan maksimum 3), dan `DurationInSeconds` (berapa lama fase lalu lintas seharusnya, dengan minimum 120 dan maksimum 3600).
    - Untuk `TrafficType`, tentukan `STAIRS`. Kemudian, untuk `Stairs` array, tentukan `DurationInSeconds` (berapa lama fase lalu lintas seharusnya, dengan minimum 120 dan maksimum 3600), `NumberOfSteps` (berapa banyak interval yang digunakan selama fase), dan `UsersPerStep` (berapa banyak pengguna yang ditambahkan selama setiap interval). Perhatikan bahwa panjang setiap langkah adalah nilai`DurationInSeconds /`

`NumberOfSteps`. Misalnya, jika Anda `600` dan Anda `DurationInSeconds` menentukan 5 langkah-langkahnya, maka setiap langkah berdurasi 120 detik.

> **Note**
>
> Seorang pengguna didefinisikan sebagai aktor yang dihasilkan sistem yang berjalan dalam satu lingkaran dan memanggil permintaan ke titik akhir sebagai bagian dari Inference Recommender. Untuk wadah XGBoost biasa yang berjalan pada sebuah `ml.c5.large` instance, titik akhir dapat mencapai 30.000 pemanggilan per menit (500 tps) hanya dengan 15-20 pengguna.

- Untuk`ResourceLimit`, tentukan `MaxNumberOfTests` (jumlah maksimum uji beban benchmarking untuk pekerjaan Inference Recommender, dengan minimal 1 dan maksimum 10) dan `MaxParallelOfTests` (jumlah maksimum uji beban benchmarking paralel untuk pekerjaan Inference Recommender, dengan minimal 1 dan maksimum 10).
- Untuk`EndpointConfigurations`, Anda dapat menentukan salah satu dari berikut ini:
  - `InstanceType`Bidang, tempat Anda menentukan jenis instance tempat Anda ingin menjalankan pengujian beban.
  - The`ServerlessConfig`, di mana Anda menentukan nilai ideal Anda untuk `MaxConcurrency` dan `MemorySizeInMB` untuk titik akhir tanpa server.

- Kamus kondisi berhenti (`stopping-conditions`), di mana jika salah satu kondisi terpenuhi, pekerjaan Inference Recommender berhenti. Untuk contoh ini, tentukan bidang berikut dalam kamus:
  - Untuk`MaxInvocations`, tentukan jumlah maksimum permintaan per menit yang diharapkan untuk titik akhir, dengan minimum 1 dan maksimum 30.000.
  - Untuk`ModelLatencyThresholds`, tentukan `Percentile` (ambang persentil latensi model) dan `ValueInMilliseconds` (nilai persentil latensi model dalam milidetik).
  - (Opsional) Untuk`FlatInvocations`, Anda dapat menentukan apakah akan melanjutkan uji beban ketika tingkat TPS (pemanggilan per menit) rata. Tingkat TPS yang diratakan biasanya berarti bahwa titik akhir telah mencapai kapasitas. Namun, Anda mungkin ingin terus memantau titik akhir dalam kondisi kapasitas penuh. Untuk melanjutkan uji beban ketika ini terjadi, tentukan nilai ini sebagai`Continue`. Jika tidak, nilai defaultnya adalah `Stop`.

```
aws sagemaker create-inference-recommendations-job\
    --region <region>\
```

```
    --job-name <job-name>\
    --job-type ADVANCED\
    --role-arn arn:aws:iam::<account>:role/*\
    --input-config \"{
        \"ModelPackageVersionArn\": \"arn:aws:sagemaker:<region>:<account>:role/*\",
        \"JobDurationInSeconds\": 7200,
        \"TrafficPattern\" : {
                # Replace PHASES with STAIRS to use the stairs traffic pattern
                \"TrafficType\": \"PHASES\",
                \"Phases\": [
                    {
                        \"InitialNumberOfUsers\": 1,
                        \"SpawnRate\": 60,
                        \"DurationInSeconds\": 300
                    }
                ]
                # Uncomment this section and comment out the Phases object above to
use the stairs traffic pattern
                # 'Stairs' : {
                #   'DurationInSeconds': 240,
                #   'NumberOfSteps': 2,
                #   'UsersPerStep': 2
                # }
            },
            \"ResourceLimit\": {
                \"MaxNumberOfTests\": 10,
                \"MaxParallelOfTests\": 3
            },
            \"EndpointConfigurations\" : [
                {
                    \"InstanceType\": \"ml.c5.xlarge\"
                },
                {
                    \"InstanceType\": \"ml.m5.xlarge\"
                },
                {
                    \"InstanceType\": \"ml.r5.xlarge\"
                }
                # Use the ServerlessConfig and leave out the InstanceType fields if
you want recommendations for a serverless endpoint
                # \"ServerlessConfig\": {
                #     \"MaxConcurrency\": value,
                #     \"MemorySizeInMB\": value
                # }
```

```
            ]
        }\"
    --stopping-conditions \"{
        \"MaxInvocations\": 1000,
        \"ModelLatencyThresholds\":[
                {
                    \"Percentile\": \"P95\",
                    \"ValueInMilliseconds\": 100
                }
        ],
        # Change 'Stop' to 'Continue' to let the load test continue if invocations
flatten
        \"FlatInvocations\": \"Stop\"
    }\"
```

Amazon SageMaker Studio Classic

Buat tes beban dengan Studio Classic.

1. Di aplikasi Studio Classic Anda, pilih ikon beranda ().
2. Di bilah sisi kiri Studio Classic, pilih Deployment.
3. Pilih Inference recommended dari daftar dropdown.
4. Pilih Buat pekerjaan pemberi rekomendasi inferensi. Tab baru berjudul Buat pekerjaan pemberi rekomendasi inferensi terbuka.
5. Pilih nama grup model Anda dari bidang grup Model dropdown. Daftar ini mencakup semua grup model yang terdaftar dengan registri model di akun Anda, termasuk model yang terdaftar di luar Studio Classic.
6. Pilih versi model dari bidang versi model dropdown.
7. Pilih Lanjutkan.
8. Berikan nama untuk pekerjaan di bidang Nama.
9. (Opsional) Berikan deskripsi pekerjaan Anda di bidang Deskripsi.
10. Pilih peran IAM yang memberikan izin Inference Recommender untuk mengakses layanan. AWS Anda dapat membuat peran dan melampirkan kebijakan terkelola `AmazonSageMakerFullAccess` IAM untuk mencapai hal ini, atau Anda dapat membiarkan Studio Classic membuat peran untuk Anda.

11. Pilih Kondisi Berhenti untuk memperluas bidang input yang tersedia. Berikan serangkaian kondisi untuk menghentikan rekomendasi penerapan.

    a. Tentukan jumlah maksimum permintaan per menit yang diharapkan untuk titik akhir di bidang Pemanggilan Maks Per Menit.

    b. Tentukan ambang latensi model dalam mikrodetik di bidang Ambang Latensi Model. Ambang Latensi Model menggambarkan interval waktu yang dibutuhkan oleh model untuk merespons sebagaimana dilihat dari Inference Recommender. Interval tersebut mencakup waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah.

12. Pilih Pola Lalu Lintas untuk memperluas bidang input yang tersedia.

    a. Tetapkan jumlah awal pengguna virtual dengan menentukan bilangan bulat di bidang Nomor Awal Pengguna.

    b. Berikan bilangan bulat untuk bidang Spawn Rate. Tingkat spawn menetapkan jumlah pengguna yang dibuat per detik.

    c. Atur durasi untuk fase dalam detik dengan menentukan bilangan bulat di bidang Durasi.

    d. (Opsional) Tambahkan pola lalu lintas tambahan. Untuk melakukannya, pilih Tambah.

13. Pilih Pengaturan tambahan untuk menampilkan bidang Durasi tes Maks. Tentukan, dalam hitungan detik, waktu maksimum yang dapat diambil tes selama pekerjaan. Pekerjaan baru tidak dijadwalkan setelah durasi yang ditentukan. Ini membantu memastikan pekerjaan yang sedang berlangsung tidak dihentikan dan Anda hanya melihat pekerjaan yang sudah selesai.

14. Pilih Lanjutkan.

15. Pilih Instans yang Dipilih.

16. Di bidang Instances for benchmarking, pilih Tambahkan instance untuk diuji. Pilih hingga 10 instans untuk Inference Recommender untuk digunakan untuk pengujian beban.

17. Pilih Pengaturan tambahan.

    a. Berikan bilangan bulat yang menetapkan batas atas jumlah pengujian yang dapat dilakukan pekerjaan untuk bidang jumlah tes Maks. Perhatikan bahwa setiap konfigurasi titik akhir menghasilkan uji beban baru.

    b. Berikan bilangan bulat untuk bidang uji paralel Max. Pengaturan ini mendefinisikan batas atas pada jumlah tes beban yang dapat berjalan secara paralel.

18. Pilih Kirim.

Tes beban bisa memakan waktu hingga 2 jam.

> **Warning**
>
> Jangan tutup tab ini. Jika Anda menutup tab ini, Anda membatalkan tugas uji beban Inference Recommender.

SageMaker console

Buat uji beban khusus melalui SageMaker konsol dengan melakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Inference recommended.
3. Pada halaman lowongan pemberi rekomendasi inferensi, pilih Buat pekerjaan.
4. Untuk Langkah 1: Konfigurasi model, lakukan hal berikut:
   a. Untuk jenis Job, pilih Lowongan pemberi rekomendasi lanjutan.
   b. Jika Anda menggunakan model yang terdaftar di registri SageMaker model, aktifkan tombol Pilih model dari registri model dan lakukan hal berikut:
      i. Untuk daftar dropdown grup Model, pilih grup model di registri SageMaker model tempat model Anda berada.
      ii. Untuk daftar dropdown versi Model, pilih versi model yang diinginkan.
   c. Jika Anda menggunakan model yang telah Anda buat SageMaker, matikan sakelar Pilih model dari registri model dan lakukan hal berikut:
      - Untuk bidang Nama model, masukkan nama SageMaker model Anda.
   d. Untuk peran IAM, Anda dapat memilih peran AWS IAM yang ada yang memiliki izin yang diperlukan untuk membuat pekerjaan rekomendasi instans. Atau, jika Anda tidak memiliki peran yang ada, Anda dapat memilih Buat peran baru untuk membuka pop-up pembuatan peran, dan SageMaker menambahkan izin yang diperlukan ke peran baru yang Anda buat.
   e. Untuk bucket S3 untuk benchmarking payload, masukkan path Amazon S3 ke arsip payload sampel Anda, yang harus berisi contoh file payload yang digunakan Inference Recomder untuk membandingkan model Anda pada jenis instans yang berbeda.

f. Untuk jenis konten Payload, masukkan tipe MIME dari data payload sampel Anda.

g. Untuk pola Lalu Lintas, konfigurasikan fase untuk uji beban dengan melakukan hal berikut:

   i. Untuk jumlah pengguna awal, tentukan berapa banyak pengguna bersamaan yang ingin Anda mulai dengan (dengan minimal 1 dan maksimal 3).

   ii. Untuk tingkat Spawn, tentukan jumlah pengguna yang akan muncul dalam satu menit untuk fase (dengan minimum 0 dan maksimum 3).

   iii. Untuk Durasi (detik), tentukan seberapa rendah fase lalu lintas dalam hitungan detik (dengan minimum 120 dan maksimum 3600).

h. (Opsional) Jika Anda mematikan Pilih model dari registri model toggle dan menentukan SageMaker model, maka untuk konfigurasi Container, lakukan hal berikut:

   i. Untuk daftar dropdown Domain, pilih domain pembelajaran mesin model, seperti visi komputer, pemrosesan bahasa alami, atau pembelajaran mesin.

   ii. Untuk daftar dropdown Framework, pilih framework container Anda, seperti TensorFlow atau XGBoost.

   iii. Untuk versi Framework, masukkan versi kerangka gambar kontainer Anda.

   iv. Untuk daftar dropdown nama model Terdekat, pilih model pra-terlatih yang sebagian besar cocok dengan model Anda.

   v. Untuk daftar tarik-turun Tugas, pilih tugas pembelajaran mesin yang diselesaikan model, seperti klasifikasi gambar atau regresi.

i. (Opsional) Untuk kompilasi Model menggunakan SageMaker Neo, Anda dapat mengonfigurasi pekerjaan rekomendasi untuk model yang telah Anda kompilasi menggunakan SageMaker Neo. Untuk konfigurasi input Data, masukkan bentuk data input yang benar untuk model Anda dalam format yang mirip dengan`{'input': [1,1024,1024,3]}`.

j. Pilih Berikutnya.

5. Untuk Langkah 2: Contoh dan parameter lingkungan, lakukan hal berikut:

   a. Untuk Select instance untuk benchmarking, pilih hingga 8 jenis instans yang ingin Anda benchmark.

   b. (Opsional) Untuk rentang parameter Lingkungan, Anda dapat menentukan parameter lingkungan yang membantu mengoptimalkan model Anda. Tentukan parameter sebagai pasangan Kunci dan Nilai.

c. Pilih Berikutnya.

6. Untuk Langkah 3: Parameter Job, lakukan hal berikut:

   a. (Opsional) Untuk bidang Nama Job, masukkan nama untuk pekerjaan rekomendasi instans Anda. Saat Anda membuat pekerjaan, SageMaker tambahkan stempel waktu ke akhir nama ini.
   b. (Opsional) Untuk kolom Job description, masukkan deskripsi untuk pekerjaan tersebut.
   c. (Opsional) Untuk daftar dropdown kunci Enkripsi, pilih AWS KMS kunci berdasarkan nama atau masukkan ARN untuk mengenkripsi data Anda.
   d. (Opsional) Untuk jumlah maksimum tes, masukkan jumlah tes yang ingin Anda jalankan selama pekerjaan rekomendasi.
   e. (Opsional) Untuk tes paralel Max, masukkan jumlah maksimum tes paralel yang ingin Anda jalankan selama pekerjaan rekomendasi.
   f. Untuk durasi pengujian Maks, masukkan jumlah detik maksimum yang Anda inginkan untuk dijalankan setiap pengujian.
   g. Untuk pemanggilan Max per menit, masukkan jumlah maksimum permintaan per menit yang dapat dicapai titik akhir sebelum menghentikan pekerjaan rekomendasi. Setelah mencapai batas ini, SageMaker akhiri pekerjaan.
   h. Untuk ambang latensi Model P99 (ms), masukkan persentil latensi model dalam milidetik.
   i. Pilih Berikutnya.

7. Untuk Langkah 4: Tinjau pekerjaan, tinjau konfigurasi Anda, lalu pilih Kirim.

Dapatkan hasil tes beban Anda

Anda dapat mengumpulkan metrik secara terprogram di semua pengujian pemuatan setelah pengujian beban selesai denganAWS SDK for Python (Boto3), Studio ClassicAWS CLI, atau konsol. SageMaker

AWS SDK for Python (Boto3)

Kumpulkan metrik dengan `DescribeInferenceRecommendationsJob` API. Tentukan nama pekerjaan uji beban untuk `JobName` bidang:

```
load_test_response = sagemaker_client.describe_inference_recommendations_job(
                                                        JobName=load_test_job_name
                                                        )
```

Cetak objek respons.

```
load_test_response['Status']
```

Ini mengembalikan respon JSON mirip dengan contoh berikut. Perhatikan bahwa contoh ini menunjukkan jenis instance yang direkomendasikan untuk inferensi waktu nyata (untuk contoh yang menunjukkan rekomendasi inferensi tanpa server, lihat contoh setelah yang ini).

```
{
    'JobName': 'job-name',
    'JobDescription': 'job-description',
    'JobType': 'Advanced',
    'JobArn': 'arn:aws:sagemaker:region:account-id:inference-recommendations-
job/resource-id',
    'Status': 'COMPLETED',
    'CreationTime': datetime.datetime(2021, 10, 26, 19, 38, 30, 957000,
 tzinfo=tzlocal()),
    'LastModifiedTime': datetime.datetime(2021, 10, 26, 19, 46, 31, 399000,
 tzinfo=tzlocal()),
    'InputConfig': {
        'ModelPackageVersionArn': 'arn:aws:sagemaker:region:account-id:model-
package/resource-id',
        'JobDurationInSeconds': 7200,
        'TrafficPattern': {
            'TrafficType': 'PHASES'
            },
        'ResourceLimit': {
            'MaxNumberOfTests': 100,
            'MaxParallelOfTests': 100
            },
        'EndpointConfigurations': [{
            'InstanceType': 'ml.c5d.xlarge'
            }]
        },
    'StoppingConditions': {
        'MaxInvocations': 1000,
        'ModelLatencyThresholds': [{
            'Percentile': 'P95',
            'ValueInMilliseconds': 100}
            ]},
    'InferenceRecommendations': [{
        'Metrics': {
            'CostPerHour': 0.6899999976158142,
```

```
            'CostPerInference': 1.0332434612791985e-05,
            'MaximumInvocations': 1113,
            'ModelLatency': 100000
            },
    'EndpointConfiguration': {
        'EndpointName': 'endpoint-name',
        'VariantName': 'variant-name',
        'InstanceType': 'ml.c5d.xlarge',
        'InitialInstanceCount': 3
        },
    'ModelConfiguration': {
        'Compiled': False,
        'EnvironmentParameters': []
        }
    }],
    'ResponseMetadata': {
        'RequestId': 'request-id',
        'HTTPStatusCode': 200,
        'HTTPHeaders': {
            'x-amzn-requestid': 'x-amzn-requestid',
            'content-type': 'content-type',
            'content-length': '1199',
            'date': 'Tue, 26 Oct 2021 19:57:42 GMT'
            },
        'RetryAttempts': 0}
    }
```

Beberapa baris pertama memberikan informasi tentang pekerjaan uji beban itu sendiri. Ini termasuk nama pekerjaan, peran ARN, pembuatan, dan waktu penghapusan.

`InferenceRecommendations`Kamus berisi daftar rekomendasi inferensi Inference Recommender.

Kamus `EndpointConfiguration` bersarang berisi rekomendasi instance type (`InstanceType`) bersama dengan titik akhir dan nama varian (model pembelajaran AWS mesin yang diterapkan) yang digunakan selama pekerjaan rekomendasi. Anda dapat menggunakan nama endpoint dan varian untuk pemantauan di Amazon CloudWatch Events. Untuk informasi selengkapnya, lihat Pantau Amazon SageMaker dengan Amazon CloudWatch.

Kamus `EndpointConfiguration` bersarang juga berisi rekomendasi instance count (`InitialInstanceCount`). Ini adalah jumlah instance yang harus Anda sediakan di titik akhir untuk memenuhi yang `MaxInvocations` ditentukan dalam. `StoppingConditions` Misalnya,

jika `InstanceType` is `ml.m5.large` dan `InitialInstanceCount` is2, maka Anda harus menyediakan 2 `ml.m5.large` instance untuk titik akhir Anda sehingga dapat menangani TPS yang ditentukan dalam kondisi berhenti. `MaxInvocations`

Kamus `Metrics` bersarang berisi informasi tentang perkiraan biaya per jam (`CostPerHour`) untuk titik akhir real-time Anda dalam dolar AS, perkiraan biaya per inferensi (`CostPerInference`) untuk titik akhir waktu nyata Anda, jumlah maksimum `InvokeEndpoint` permintaan yang dikirim ke titik akhir, dan latensi model (`ModelLatency`), yang merupakan interval waktu (dalam mikrodetik) yang diambil model Anda untuk merespons. SageMaker Latensi model mencakup waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah.

Contoh berikut menunjukkan `InferenceRecommendations` bagian respons untuk pekerjaan uji beban yang dikonfigurasi untuk mengembalikan rekomendasi inferensi tanpa server:

```
"InferenceRecommendations": [
      {
         "EndpointConfiguration": {
            "EndpointName": "value",
            "InitialInstanceCount": value,
            "InstanceType": "value",
            "VariantName": "value",
            "ServerlessConfig": {
                "MaxConcurrency": value,
                "MemorySizeInMb": value
            }
         },
         "InvocationEndTime": value,
         "InvocationStartTime": value,
         "Metrics": {
            "CostPerHour": value,
            "CostPerInference": value,
            "CpuUtilization": value,
            "MaxInvocations": value,
            "MemoryUtilization": value,
            "ModelLatency": value,
            "ModelSetupTime": value
         },
         "ModelConfiguration": {
            "Compiled": "False",
            "EnvironmentParameters": [],
```

```
            "InferenceSpecificationName": "value"
        },
        "RecommendationId": "value"
    }
]
```

Anda dapat menafsirkan rekomendasi untuk inferensi tanpa server mirip dengan hasil untuk inferensi waktu nyata, dengan pengecualian`ServerlessConfig`, yang memberi tahu Anda nilai yang Anda tentukan untuk `MaxConcurrency` dan `MemorySizeInMB` saat menyiapkan uji beban. Rekomendasi tanpa server juga mengukur metrik`ModelSetupTime`, yang mengukur (dalam mikrodetik) waktu yang diperlukan untuk meluncurkan sumber daya komputasi pada titik akhir tanpa server. Untuk informasi selengkapnya tentang pengaturan titik akhir tanpa server, lihat dokumentasi Inferensi Tanpa Server.

AWS CLI

Kumpulkan metrik dengan `describe-inference-recommendations-job` API. Tentukan nama pekerjaan uji beban untuk `job-name` bendera:

```
aws sagemaker describe-inference-recommendations-job --job-name <job-name>
```

Ini mengembalikan respons yang mirip dengan berikut ini. Perhatikan bahwa contoh ini menunjukkan jenis instance yang direkomendasikan untuk inferensi waktu nyata (untuk contoh yang menunjukkan rekomendasi inferensi tanpa server, lihat contoh setelah yang ini).

```
{
    'JobName': 'job-name',
    'JobDescription': 'job-description',
    'JobType': 'Advanced',
    'JobArn': 'arn:aws:sagemaker:region:account-id:inference-recommendations-
job/resource-id',
    'Status': 'COMPLETED',
    'CreationTime': datetime.datetime(2021, 10, 26, 19, 38, 30, 957000,
 tzinfo=tzlocal()),
    'LastModifiedTime': datetime.datetime(2021, 10, 26, 19, 46, 31, 399000,
 tzinfo=tzlocal()),
    'InputConfig': {
        'ModelPackageVersionArn': 'arn:aws:sagemaker:region:account-id:model-
package/resource-id',
        'JobDurationInSeconds': 7200,
        'TrafficPattern': {
            'TrafficType': 'PHASES'
```

```
            },
        'ResourceLimit': {
            'MaxNumberOfTests': 100,
            'MaxParallelOfTests': 100
            },
        'EndpointConfigurations': [{
            'InstanceType': 'ml.c5d.xlarge'
            }]
        },
    'StoppingConditions': {
        'MaxInvocations': 1000,
        'ModelLatencyThresholds': [{
            'Percentile': 'P95',
            'ValueInMilliseconds': 100
            }]
        },
    'InferenceRecommendations': [{
        'Metrics': {
        'CostPerHour': 0.6899999976158142,
        'CostPerInference': 1.0332434612791985e-05,
        'MaximumInvocations': 1113,
        'ModelLatency': 100000
        },
        'EndpointConfiguration': {
            'EndpointName': 'endpoint-name',
            'VariantName': 'variant-name',
            'InstanceType': 'ml.c5d.xlarge',
            'InitialInstanceCount': 3
            },
        'ModelConfiguration': {
            'Compiled': False,
            'EnvironmentParameters': []
            }
        }],
    'ResponseMetadata': {
        'RequestId': 'request-id',
        'HTTPStatusCode': 200,
        'HTTPHeaders': {
            'x-amzn-requestid': 'x-amzn-requestid',
            'content-type': 'content-type',
            'content-length': '1199',
            'date': 'Tue, 26 Oct 2021 19:57:42 GMT'
            },
        'RetryAttempts': 0
```

```
        }
    }
```

Beberapa baris pertama memberikan informasi tentang pekerjaan uji beban itu sendiri. Ini termasuk nama pekerjaan, peran ARN, pembuatan, dan waktu penghapusan.

`InferenceRecommendations`Kamus berisi daftar rekomendasi inferensi Inference Recommender.

Kamus `EndpointConfiguration` bersarang berisi rekomendasi instance type (`InstanceType`) bersama dengan titik akhir dan nama varian (model pembelajaran AWS mesin yang diterapkan) yang digunakan selama pekerjaan rekomendasi. Anda dapat menggunakan nama endpoint dan varian untuk pemantauan di Amazon CloudWatch Events. Untuk informasi selengkapnya, lihat Pantau Amazon SageMaker dengan Amazon CloudWatch.

Kamus `Metrics` bersarang berisi informasi tentang perkiraan biaya per jam (`CostPerHour`) untuk titik akhir real-time Anda dalam dolar AS, perkiraan biaya per inferensi (`CostPerInference`) untuk titik akhir waktu nyata Anda, jumlah maksimum `InvokeEndpoint` permintaan yang dikirim ke titik akhir, dan latensi model (`ModelLatency`), yang merupakan interval waktu (dalam mikrodetik) yang diambil model Anda untuk merespons. SageMaker Latensi model mencakup waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah.

Contoh berikut menunjukkan `InferenceRecommendations` bagian respons untuk pekerjaan uji beban yang dikonfigurasi untuk mengembalikan rekomendasi inferensi tanpa server:

```
"InferenceRecommendations": [
      {
         "EndpointConfiguration": {
            "EndpointName": "value",
            "InitialInstanceCount": value,
            "InstanceType": "value",
            "VariantName": "value",
            "ServerlessConfig": {
                "MaxConcurrency": value,
                "MemorySizeInMb": value
            }
         },
         "InvocationEndTime": value,
```

```
        "InvocationStartTime": value,
        "Metrics": {
           "CostPerHour": value,
           "CostPerInference": value,
           "CpuUtilization": value,
           "MaxInvocations": value,
           "MemoryUtilization": value,
           "ModelLatency": value,
           "ModelSetupTime": value
        },
        "ModelConfiguration": {
           "Compiled": "False",
           "EnvironmentParameters": [],
           "InferenceSpecificationName": "value"
        },
        "RecommendationId": "value"
     }
  ]
```

Anda dapat menafsirkan rekomendasi untuk inferensi tanpa server mirip dengan hasil untuk inferensi waktu nyata, dengan pengecualian`ServerlessConfig`, yang memberi tahu Anda nilai yang Anda tentukan untuk `MaxConcurrency` dan `MemorySizeInMB` saat menyiapkan uji beban. Rekomendasi tanpa server juga mengukur metrik`ModelSetupTime`, yang mengukur (dalam mikrodetik) waktu yang diperlukan untuk meluncurkan sumber daya komputer pada titik akhir tanpa server. Untuk informasi selengkapnya tentang pengaturan titik akhir tanpa server, lihat dokumentasi Inferensi Tanpa Server.

Amazon SageMaker Studio Classic

Rekomendasi terisi di tab baru yang disebut Rekomendasi inferensi dalam Studio Classic. Diperlukan waktu hingga 2 jam agar hasilnya muncul. Tab ini berisi kolom Hasil dan Detail.

Kolom Detail memberikan informasi tentang pekerjaan uji beban, seperti nama yang diberikan untuk pekerjaan uji beban, saat pekerjaan dibuat (Waktu pembuatan), dan banyak lagi. Ini juga berisi informasi Pengaturan, seperti jumlah maksimum pemanggilan yang terjadi per menit dan informasi tentang Nama Sumber Daya Amazon yang digunakan.

Kolom Hasil menyediakan jendela tujuan dan SageMakerrekomendasi Deployment di mana Anda dapat menyesuaikan urutan hasil ditampilkan berdasarkan kepentingan penerapan. Ada tiga menu tarik-turun di mana Anda dapat memberikan tingkat kepentingan Biaya, Latensi, dan Throughput untuk kasus penggunaan Anda. Untuk setiap tujuan (biaya, latensi, dan throughput),

Anda dapat menetapkan tingkat kepentingan: Kepentingan Terendah, Kepentingan Rendah, Kepentingan sedang, Kepentingan tinggi, atau Kepentingan tertinggi.

Berdasarkan pilihan penting Anda untuk setiap tujuan, Inference Recommender menampilkan rekomendasi teratasnya di bidang SageMakerrekomendasi di sebelah kanan panel, bersama dengan perkiraan biaya per jam dan permintaan inferensi. Ini juga menyediakan Informasi tentang latensi model yang diharapkan, jumlah maksimum pemanggilan, dan jumlah instance.

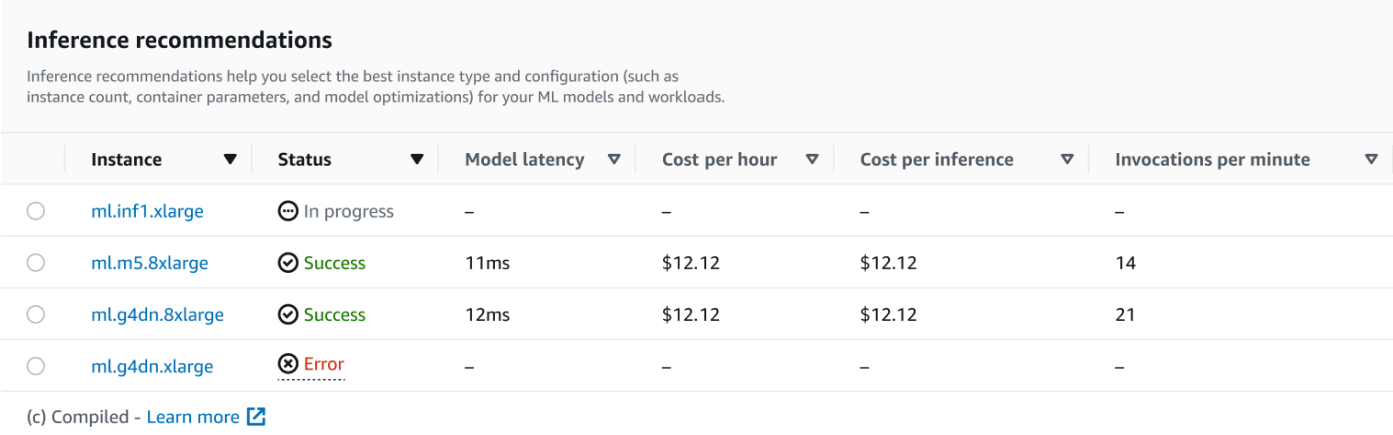
Selain rekomendasi teratas yang ditampilkan, Anda juga dapat melihat informasi yang sama ditampilkan untuk semua instance yang diuji oleh Inference Recommender di bagian Semua berjalan.

SageMaker console

Anda dapat melihat hasil pekerjaan uji beban kustom di SageMaker konsol dengan melakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Inference recommended.
3. Pada halaman pekerjaan pemberi rekomendasi inferensi, pilih nama pekerjaan rekomendasi inferensi Anda.

Pada halaman detail untuk pekerjaan Anda, Anda dapat melihat rekomendasi Inferensi, yang merupakan jenis instance yang SageMaker direkomendasikan untuk model Anda, seperti yang ditunjukkan pada gambar berikut.



Di bagian ini, Anda dapat membandingkan jenis instans dengan berbagai faktor seperti latensi Model, Biaya per jam, Biaya per inferensi, dan Pemanggilan per menit.

Di halaman ini, Anda juga dapat melihat konfigurasi yang Anda tentukan untuk pekerjaan Anda. Di bagian Monitor, Anda dapat melihat CloudWatch metrik Amazon yang dicatat untuk setiap jenis instans. Untuk mempelajari lebih lanjut tentang menafsirkan metrik ini, lihat Menafsirkan hasil.

Hentikan uji beban Anda

Anda mungkin ingin menghentikan pekerjaan yang sedang berjalan jika Anda memulai pekerjaan secara tidak sengaja atau tidak perlu lagi menjalankan pekerjaan itu. Hentikan tugas uji beban Anda secara terprogram dengan `StopInferenceRecommendationsJob` API, atau melalui Studio Classic atau konsol. SageMaker

AWS SDK for Python (Boto3)

Tentukan nama pekerjaan uji beban untuk `JobName` bidang:

```
sagemaker_client.stop_inference_recommendations_job(
                                    JobName='<INSERT>'
                                    )
```

AWS CLI

Tentukan nama pekerjaan uji beban untuk `job-name` bendera:

```
aws sagemaker stop-inference-recommendations-job --job-name <job-name>
```

Amazon SageMaker Studio Classic

Tutup tab tempat Anda memulai tugas pemuatan khusus untuk menghentikan uji beban Inference Recommender Anda.

SageMaker console

Untuk menghentikan pekerjaan uji beban Anda melalui SageMaker konsol, lakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Inference recommended.
3. Pada halaman pekerjaan pemberi rekomendasi inferensi, pilih pekerjaan uji beban Anda.
4. Pilih Stop Job.
5. Di kotak dialog yang muncul, pilih Konfirmasi.

Setelah menghentikan pekerjaan Anda, Status pekerjaan harus berubah menjadi Berhenti.

Memecahkan masalah kesalahan Inference Recommender

Bagian ini berisi informasi tentang cara memahami dan mencegah kesalahan umum, pesan kesalahan yang mereka hasilkan, dan panduan tentang cara mengatasi kesalahan ini.

Cara memecahkan masalah

Anda dapat mencoba mengatasi kesalahan Anda dengan melalui langkah-langkah berikut:

- Periksa apakah Anda telah mencakup semua prasyarat untuk menggunakan Inference Recommender. Lihat Prasyarat Rekomendasi Inferensi.
- Periksa apakah Anda dapat menerapkan model Anda dari Registri Model ke titik akhir dan dapat memproses muatan Anda tanpa kesalahan. Lihat Menerapkan Model dari Registri.
- Saat memulai pekerjaan Inference Recommender, Anda akan melihat titik akhir dibuat di konsol dan Anda dapat meninjau log. CloudWatch

Kesalahan umum

Tinjau tabel berikut untuk kesalahan Inference Recommender umum dan solusinya.

Kesalahan	Solusi
Tentukan `Domain` dalam Model Package versi 1. `Domain`adalah parameter wajib untuk pekerjaan itu.	Pastikan Anda menyediakan Domain ML atau `OTHER` jika tidak diketahui.
Arn peran yang diberikan tidak dapat diasumsikan dan `AWSSecurityTokenServiceException` terjadi kesalahan.	Pastikan peran eksekusi yang disediakan memiliki izin yang diperlukan yang ditentukan dalam prasyarat.
Tentukan `Framework` dalam Model Package versi 1. `Framework` adalah parameter wajib untuk pekerjaan itu.	Pastikan Anda menyediakan Framework ML atau `OTHER` jika tidak diketahui.

Kesalahan	Solusi
Pengguna pada akhir fase sebelumnya adalah 0 sedangkan pengguna awal fase saat ini adalah 1.	Pengguna di sini mengacu pada pengguna virtual atau utas yang digunakan untuk mengirim permintaan. Setiap fase dimulai dengan pengguna A dan diakhiri dengan pengguna B sehingga B > A. Antara fase berurutan, x_1 dan x_2, kita memerlukan abs (x_2.a - x_1.b) <= 3 dan >= 0.
Total durasi Lalu Lintas (lintas) tidak boleh lebih dari durasi Job.	Total durasi semua Fase Anda tidak dapat melebihi durasi Job.
Jenis instance burstable ml.t2.medium tidak diperbolehkan.	Inference Recommender tidak mendukung pengujian beban pada keluarga instans t2 karena instance burstable tidak memberikan kinerja yang konsisten.
ResourceLimitExceeded saat memanggil CreateEndpoint operasi	Anda telah melampaui batas sumber SageMaker daya. Misalnya, Inference Recommender mungkin tidak dapat menyediakan endpoint untuk benchmarking jika akun telah mencapai kuota endpoint. Untuk informasi selengkapnya tentang SageMaker batas dan kuota, lihat SageMaker titik akhir dan kuota Amazon.
ModelError saat memanggil InvokeEndpoint operasi	Kesalahan model dapat terjadi karena alasan berikut: • Waktu pemanggilan habis sambil menunggu respons dari wadah model. • Model tidak dapat memproses muatan input.

Kesalahan	Solusi
PayloadError saat memanggil InvokeEndpoint operasi	Kesalahan payload dapat terjadi karena alasan berikut: • Sumber payload tidak ada di bucket Amazon S3. • Payload dalam format objek non-file. • Payload dalam jenis file yang tidak valid. Misalnya, model mengharapkan muatan tipe gambar tetapi diteruskan file teks. • Muatannya kosong.

Periksa CloudWatch

Saat memulai pekerjaan Inference Recommender, Anda akan melihat titik akhir dibuat di konsol. Pilih salah satu titik akhir dan lihat CloudWatch log untuk memantau kesalahan 4xx/5xx. Jika Anda memiliki pekerjaan Inference Recommender yang sukses, Anda akan dapat melihat nama titik akhir sebagai bagian dari hasil. Bahkan jika pekerjaan Inference Recommender Anda tidak berhasil, Anda masih dapat memeriksa CloudWatch log untuk titik akhir yang dihapus dengan mengikuti langkah-langkah di bawah ini:

1. Buka CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/.
2. Pilih Wilayah tempat Anda membuat pekerjaan Inference Recommender dari daftar dropdown Region di kanan atas.
3. Di panel navigasi CloudWatch, pilih Log, lalu pilih Grup log.
4. Cari grup log yang disebut`/aws/sagemaker/Endpoints/sm-epc-*`. Pilih grup log berdasarkan pekerjaan Inference Recommender terbaru Anda.

Anda juga dapat memecahkan masalah pekerjaan Anda dengan memeriksa log Inference Recommender. CloudWatch Log Inference Recommender, yang diterbitkan dalam grup `/aws/sagemaker/InferenceRecommendationsJobs` CloudWatch log, memberikan tampilan tingkat tinggi tentang kemajuan pekerjaan di aliran *`<jobName>`*`/execution` log. Anda dapat menemukan informasi terperinci tentang setiap konfigurasi titik akhir yang sedang diuji di aliran *`<jobName>`*`/Endpoint/`*`<endpointName>` log.

Ikhtisar aliran log Inference Recommender

- *<jobName>*/execution berisi informasi pekerjaan secara keseluruhan seperti konfigurasi titik akhir yang dijadwalkan untuk pembandingan, alasan lompatan pekerjaan kompilasi, dan alasan kegagalan validasi.
- *<jobName>*/Endpoint/*<endpointName>* berisi informasi seperti kemajuan pembuatan sumber daya, konfigurasi pengujian, alasan berhenti uji beban, dan status pembersihan sumber daya.
- *<jobName>*/CompilationJob/*<compilationJobName>* berisi informasi tentang pekerjaan kompilasi yang dibuat oleh Inference Recommender, seperti konfigurasi pekerjaan kompilasi dan status pekerjaan kompilasi.

Buat alarm untuk pesan kesalahan Inference Recommender

Inference Recommender mengeluarkan pernyataan log untuk kesalahan yang mungkin berguna saat pemecahan masalah. Dengan grup CloudWatch log dan filter metrik, Anda dapat mencari istilah dan pola dalam data log ini saat data dikirim CloudWatch. Kemudian, Anda dapat membuat CloudWatch alarm berdasarkan filter metrik grup log. Untuk informasi selengkapnya, lihat Membuat CloudWatch alarm berdasarkan filter metrik grup log.

Periksa tolok ukur

Saat memulai pekerjaan Inference Recommender, Inference Recommender membuat beberapa tolok ukur untuk mengevaluasi kinerja model Anda pada jenis instans yang berbeda. Anda dapat menggunakan ListInferenceRecommendationsJobSteps API untuk melihat detail untuk semua tolok ukur. Jika Anda memiliki tolok ukur yang gagal, Anda dapat melihat alasan kegagalan sebagai bagian dari hasil.

Untuk menggunakan ListInferenceRecommendationsJobSteps API, berikan nilai berikut:

- Untuk `JobName`, berikan nama pekerjaan Inference Recommender.
- Untuk `StepType`, gunakan `BENCHMARK` untuk mengembalikan detail tentang tolok ukur pekerjaan.
- Untuk `Status`, gunakan `FAILED` untuk mengembalikan detail hanya tentang tolok ukur yang gagal. Untuk daftar jenis status lainnya, lihat `Status` bidang di ListInferenceRecommendationsJobSteps API.

```
# Create a low-level SageMaker service client.
import boto3
```

```
aws_region = '<region>'
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Provide the job name for the SageMaker Inference Recommender job
job_name = '<job-name>'

# Filter for benchmarks
step_type = 'BENCHMARK'

# Filter for benchmarks that have a FAILED status
status = 'FAILED'

response = sagemaker_client.list_inference_recommendations_job_steps(
    JobName = job_name,
    StepType = step_type,
    Status = status
)
```

Anda dapat mencetak objek respons untuk melihat hasilnya. Contoh kode sebelumnya menyimpan respons dalam variabel yang disebut: `response`

```
print(response)
```

Inferensi waktu nyata

Inferensi real-time sangat ideal untuk beban kerja inferensi di mana Anda memiliki persyaratan real-time, interaktif, latensi rendah. Anda dapat menerapkan model Anda ke layanan SageMaker hosting dan mendapatkan titik akhir yang dapat digunakan untuk inferensi. Titik akhir ini dikelola sepenuhnya dan mendukung penskalaan otomatis (lihat). Secara Otomatis Menskalakan SageMaker Model Amazon

Topik

- Terapkan model untuk inferensi waktu nyata
- Memanggil model untuk inferensi waktu nyata
- Kelola titik akhir Anda
- Opsi hosting
- Secara Otomatis Menskalakan SageMaker Model Amazon
- Volume volume penyimpanan volume penyimpanan instans volume penyimpanan instans

- Aman memvalidasi model dalam produksi
- Penjelasan Online dengan Clarify SageMaker

Terapkan model untuk inferensi waktu nyata

Ada beberapa opsi untuk menerapkan model menggunakan layanan SageMaker hosting. Anda dapat menerapkan model secara interaktif dengan SageMaker Studio. Atau, Anda dapat menerapkan model secara terprogram menggunakan AWS SDK, seperti SDK Python atau SDK for SageMaker Python (Boto3). Anda juga dapat menerapkan dengan menggunakan. AWS CLI

Sebelum Anda memulai

Sebelum Anda menerapkan SageMaker model, cari dan catat hal-hal berikut:

- Wilayah AWSTempat bucket Amazon S3 Anda berada
- Jalur URI Amazon S3 tempat artefak model disimpan
- Peran IAM untuk SageMaker
- Jalur registri URI ECR Docker Amazon untuk gambar kustom yang berisi kode inferensi, atau kerangka kerja dan versi gambar Docker bawaan yang didukung dan oleh AWS

Untuk daftar yang Layanan AWS tersedia di masing-masingWilayah AWS, lihat Peta Wilayah dan Jaringan Tepi. Lihat Membuat peran IAM untuk informasi tentang cara membuat peran IAM.

> **Important**
>
> Bucket Amazon S3 tempat artefak model disimpan harus Wilayah AWS sama dengan model yang Anda buat.

Pemanfaatan sumber daya bersama dengan beberapa model

Anda dapat menerapkan satu atau beberapa model ke titik akhir dengan Amazon. SageMaker Ketika beberapa model berbagi titik akhir, mereka bersama-sama memanfaatkan sumber daya yang di-host di sana, seperti instance komputasi, CPU, dan akselerator. Cara paling fleksibel untuk menerapkan beberapa model ke titik akhir adalah dengan mendefinisikan setiap model sebagai komponen inferensi.

Komponen inferensi

Komponen inferensi adalah objek SageMaker hosting yang dapat Anda gunakan untuk menyebarkan model ke titik akhir. Dalam pengaturan komponen inferensi, Anda menentukan model, titik akhir, dan bagaimana model memanfaatkan sumber daya yang dihosting titik akhir. Untuk menentukan model, Anda dapat menentukan objek SageMaker Model, atau Anda dapat langsung menentukan artefak model dan gambar.

Dalam pengaturan, Anda dapat mengoptimalkan pemanfaatan sumber daya dengan menyesuaikan bagaimana inti CPU, akselerator, dan memori yang diperlukan dialokasikan ke model. Anda dapat menerapkan beberapa komponen inferensi ke titik akhir, di mana setiap komponen inferensi berisi satu model dan kebutuhan pemanfaatan sumber daya untuk model itu.

Setelah menerapkan komponen inferensi, Anda dapat langsung memanggil model terkait saat menggunakan InvokeEndpoint tindakan di API. SageMaker

Komponen inferensi memberikan manfaat sebagai berikut:

Fleksibilitas

Komponen inferensi memisahkan detail hosting model dari titik akhir itu sendiri. Ini memberikan lebih banyak fleksibilitas dan kontrol atas bagaimana model di-host dan disajikan dengan titik akhir. Anda dapat meng-host beberapa model pada infrastruktur yang sama, dan Anda dapat menambahkan atau menghapus model dari titik akhir sesuai kebutuhan. Anda dapat memperbarui setiap model secara independen.

Skalabilitas

Anda dapat menentukan berapa banyak salinan dari setiap model untuk dihosting, dan Anda dapat mengatur jumlah minimum salinan untuk memastikan bahwa model memuat dalam jumlah yang Anda butuhkan untuk melayani permintaan. Anda dapat menskalakan salinan komponen inferensi apa pun ke nol, yang memberi ruang bagi salinan lain untuk ditingkatkan.

SageMaker mengemas model Anda sebagai komponen inferensi saat Anda menerapkannya dengan menggunakan:

- SageMaker Studio Klasik.
- SDK SageMaker Python untuk menyebarkan objek Model (tempat Anda menyetel tipe titik akhir ke). `EndpointType.INFERENCE_COMPONENT_BASED`

- AWS SDK for Python (Boto3)Untuk mendefinisikan `InferenceComponent` objek yang Anda terapkan ke titik akhir.

Menyebarkan model dengan Studio SageMaker

Selesaikan langkah-langkah berikut untuk membuat dan menerapkan model Anda secara interaktif melalui SageMaker Studio. Untuk informasi selengkapnya tentang Studio, lihat dokumentasi Studio. Untuk panduan lebih lanjut tentang berbagai skenario penerapan, lihat Package blog dan terapkan model dan LLM klasik dengan mudah menggunakan Amazon - Bagian 2. SageMaker

Siapkan artefak dan izin Anda

Lengkapi bagian ini sebelum membuat model di SageMaker Studio.

Anda memiliki dua opsi untuk membawa artefak Anda dan membuat model di Studio:

1. Anda dapat membawa `tar.gz` arsip pra-paket, yang harus menyertakan artefak model Anda, kode inferensi kustom apa pun, dan dependensi apa pun yang tercantum dalam file. `requirements.txt`
2. SageMaker dapat mengemas artefak Anda untuk Anda. Anda hanya perlu membawa artefak model mentah dan dependensi apa pun dalam sebuah `requirements.txt` file, dan SageMaker dapat memberikan kode inferensi default untuk Anda (atau Anda dapat mengganti kode default dengan kode inferensi kustom Anda sendiri). SageMakermendukung opsi ini untuk kerangka kerja berikut: PyTorch, XGBoost.

Selain membawa model Anda, peran AWS Identity and Access Management (IAM) Anda, dan wadah Docker (atau kerangka kerja dan versi yang diinginkan yang SageMaker memiliki wadah pra-bangun), Anda juga harus memberikan izin untuk membuat dan menerapkan model melalui Studio. SageMaker

Anda harus memiliki AmazonSageMakerFullAccesskebijakan yang melekat pada peran IAM Anda sehingga Anda dapat mengakses SageMaker dan layanan terkait lainnya. Untuk melihat harga jenis instans di Studio, Anda juga harus melampirkan AWSPriceListServiceFullAccesskebijakan (atau jika Anda tidak ingin melampirkan seluruh kebijakan, lebih khusus lagi, `pricing:GetProducts` tindakan).

Jika Anda memilih untuk mengunggah artefak model saat membuat model (atau mengunggah file payload sampel untuk rekomendasi inferensi), Anda harus membuat bucket Amazon S3. Nama

bucket harus diawali dengan kata`SageMaker`. Kapitalisasi alternatif juga dapat diterima: `Sagemaker` atau. SageMaker `sagemaker`

Kami menyarankan Anda menggunakan konvensi penamaan ember`sagemaker-{`*`Region`*`}-{`*`accountID`*`}`. Bucket ini digunakan untuk menyimpan artefak yang Anda unggah.

Setelah membuat bucket, lampirkan kebijakan CORS (cross-origin resource sharing) berikut ke bucket:

```
[
    {
        "AllowedHeaders": ["*"],
        "ExposeHeaders": ["Etag"],
        "AllowedMethods": ["PUT", "POST"],
        "AllowedOrigins": ['https://*.sagemaker.aws'],
    }
]
```

Anda dapat melampirkan kebijakan CORS ke bucket Amazon S3 dengan menggunakan salah satu metode berikut:

- Melalui halaman Edit cross-origin resource sharing (CORS) di konsol Amazon S3
- Menggunakan Amazon S3 API PutBucketCors
- Menggunakan put-bucket-cors AWS CLI perintah:

  ```
  aws s3api put-bucket-cors --bucket="..." --cors-configuration="..."
  ```

Buat model yang dapat diterapkan

Pada langkah ini, Anda membuat versi model yang dapat diterapkan SageMaker dengan menyediakan artefak Anda bersama dengan spesifikasi tambahan, seperti wadah dan kerangka kerja yang Anda inginkan, kode inferensi kustom apa pun, dan pengaturan jaringan.

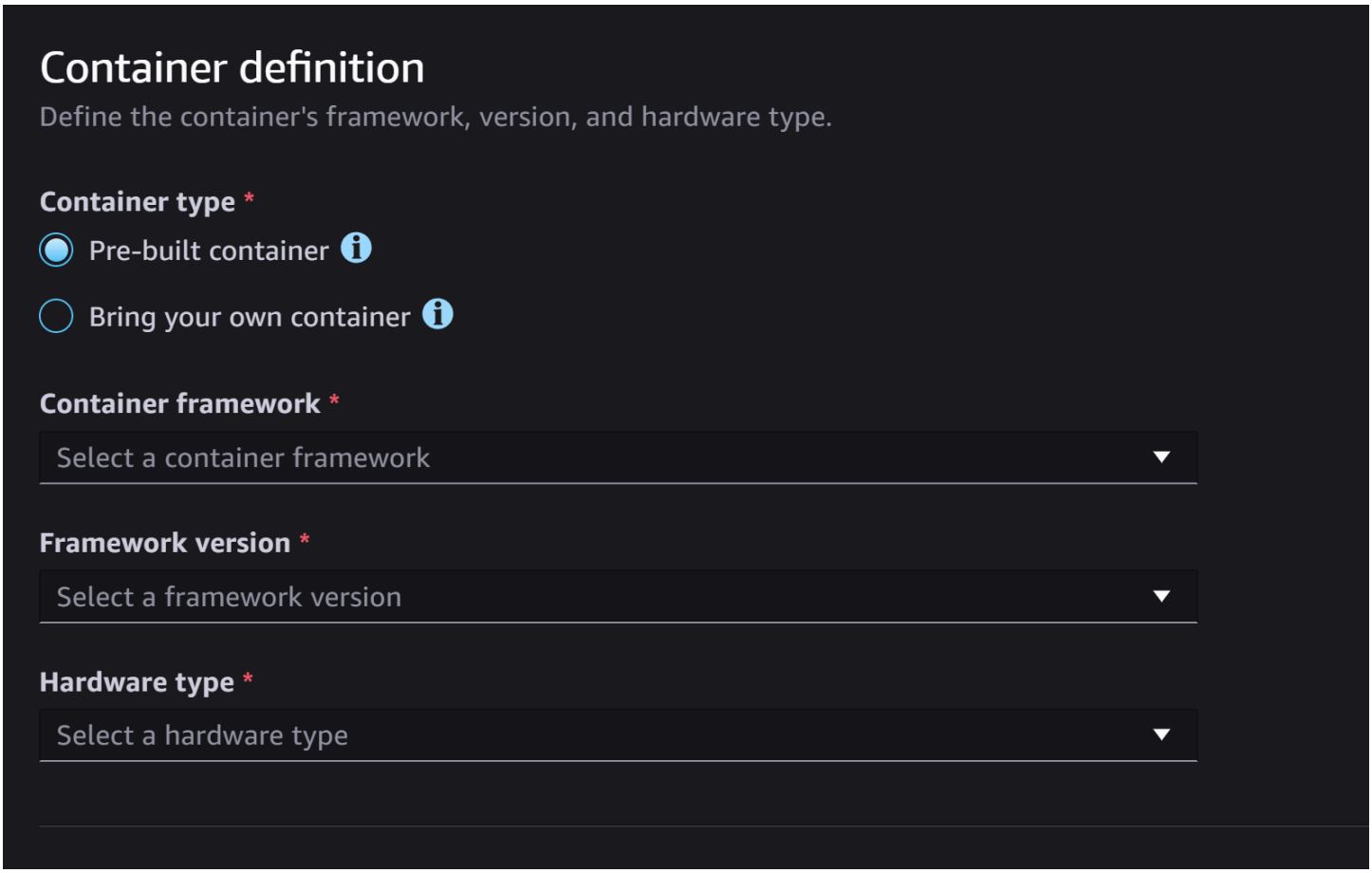
Buat model deployable di SageMaker Studio dengan melakukan hal berikut:

1. Buka aplikasi SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model Deployable.
4. Pada halaman Model Deployable, pilih Buat.

5. Pada halaman Buat model deployable, untuk bidang Nama model, masukkan nama untuk model.

Ada beberapa bagian lagi untuk Anda isi di halaman Create deployable model.

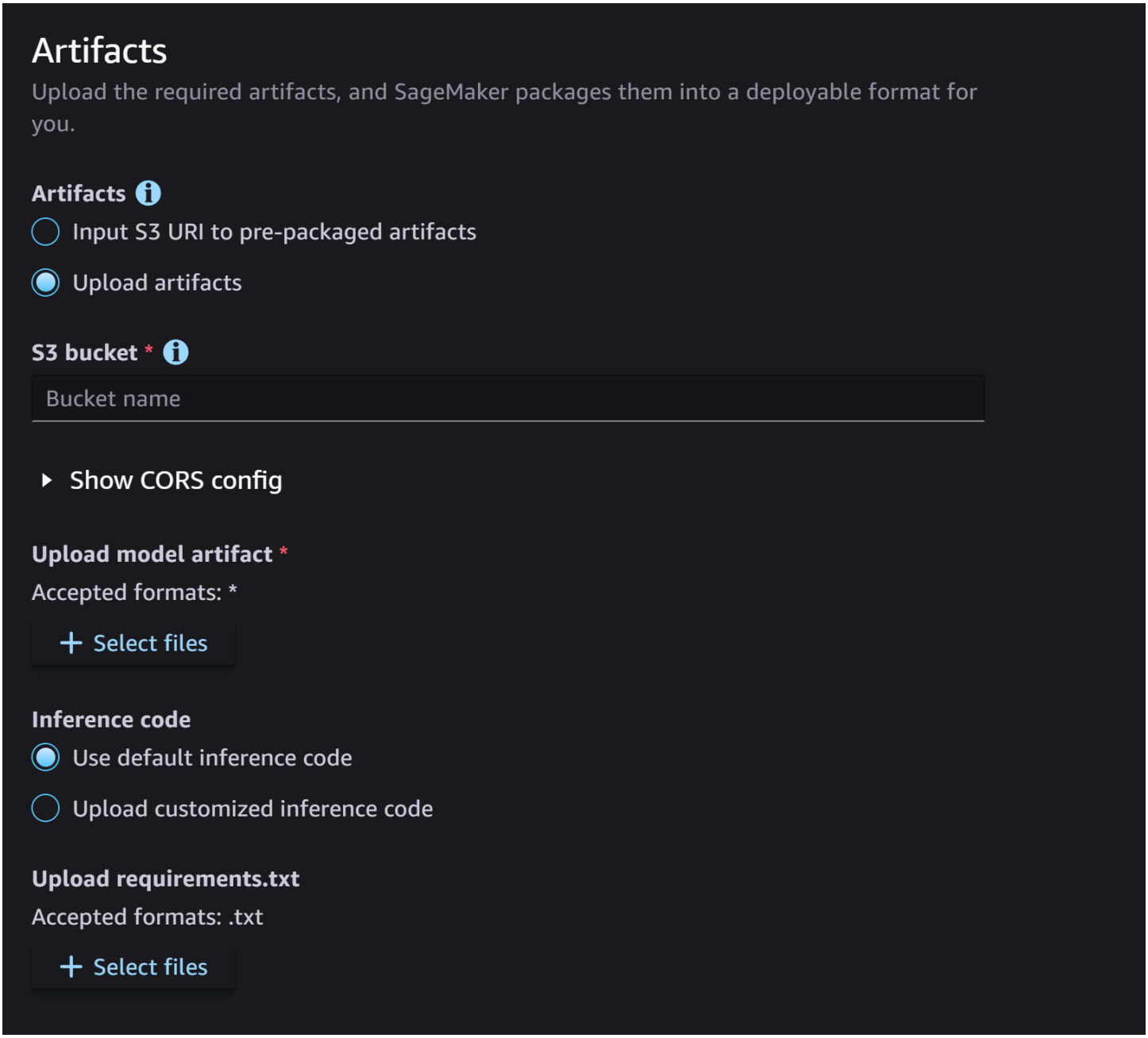
Bagian definisi Container terlihat seperti tangkapan layar berikut:



Untuk bagian Container definition, lakukan hal berikut:

1. Untuk jenis Kontainer, pilih Kontainer pra-bangun jika Anda ingin menggunakan kontainer SageMaker terkelola, atau pilih Bawa kontainer Anda sendiri jika Anda memiliki wadah sendiri.
2. Jika Anda memilih Container Pre-built, pilih framework Container, versi Framework, dan jenis Hardware yang ingin Anda gunakan.
3. Jika Anda memilih Bawa penampung Anda sendiri, masukkan jalur ECR Amazon untuk jalur ECR ke image kontainer.

Kemudian, isi bagian Artefak, yang terlihat seperti tangkapan layar berikut:



Untuk bagian Artefak, lakukan hal berikut:

1. Jika Anda menggunakan salah satu kerangka kerja yang SageMaker mendukung artefak model kemasan (PyTorch atau XGBoost), maka untuk Artefak, Anda dapat memilih opsi Unggah artefak. Dengan opsi ini, Anda cukup menentukan artefak model mentah Anda, kode inferensi kustom apa pun yang Anda miliki, dan file requirements.txt Anda, dan SageMaker menangani pengemasan arsip untuk Anda. Lakukan hal-hal berikut:

a. Untuk Artefak, pilih Unggah artefak untuk terus menyediakan file Anda. Jika tidak, jika Anda sudah memiliki `tar.gz` arsip yang berisi file model, kode inferensi, dan `requirements.txt` file, lalu pilih Input S3 URI ke artefak pra-paket.

b. Jika Anda memilih untuk mengunggah artefak Anda, maka untuk bucket S3, masukkan jalur Amazon S3 ke ember tempat Anda SageMaker ingin menyimpan artefak setelah mengemasnya untuk Anda. Kemudian, selesaikan langkah-langkah berikut.

c. Untuk Unggah artefak model, unggah file model Anda.

d. Untuk kode Inferensi, pilih Gunakan kode inferensi default jika Anda ingin menggunakan kode default yang SageMaker menyediakan untuk menyajikan inferensi. Jika tidak, pilih Unggah kode inferensi yang disesuaikan untuk menggunakan kode inferensi Anda sendiri.

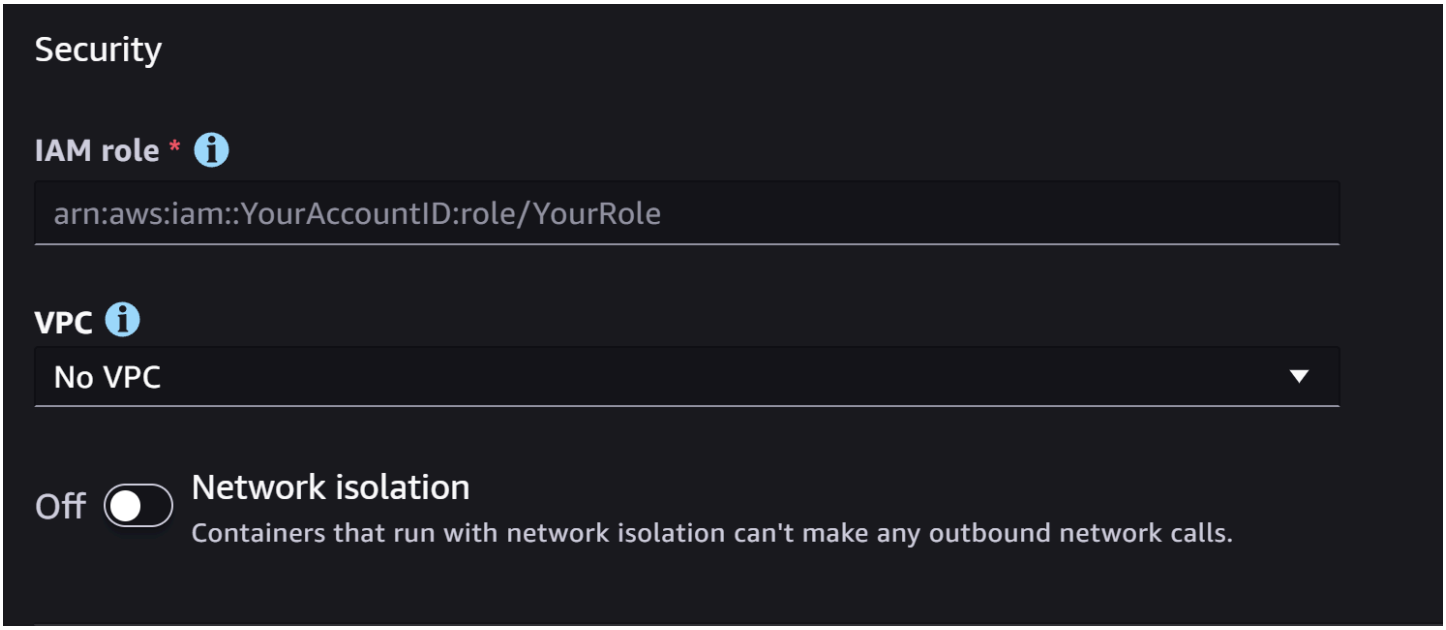
e. Untuk Upload requirements.txt, unggah file teks yang mencantumkan dependensi apa pun yang ingin Anda instal saat runtime.

2. Jika Anda tidak menggunakan kerangka kerja yang SageMaker mendukung artefak model pengemasan, Studio menunjukkan opsi artefak Pra-paket, dan Anda harus menyediakan semua artefak yang sudah dikemas sebagai arsip. `tar.gz` Lakukan hal-hal berikut:

   a. Untuk artefak pra-paket, pilih Masukan URI S3 untuk artefak model pra-paket jika arsip Anda sudah `tar.gz` diunggah ke Amazon S3. Pilih Unggah artefak model pra-paket jika Anda ingin langsung mengunggah arsip Anda. SageMaker

   b. Jika Anda memilih URI Input S3 untuk artefak model pra-paket, masukkan jalur Amazon S3 ke arsip Anda untuk URI S3. Jika tidak, pilih dan unggah arsip dari mesin lokal Anda.

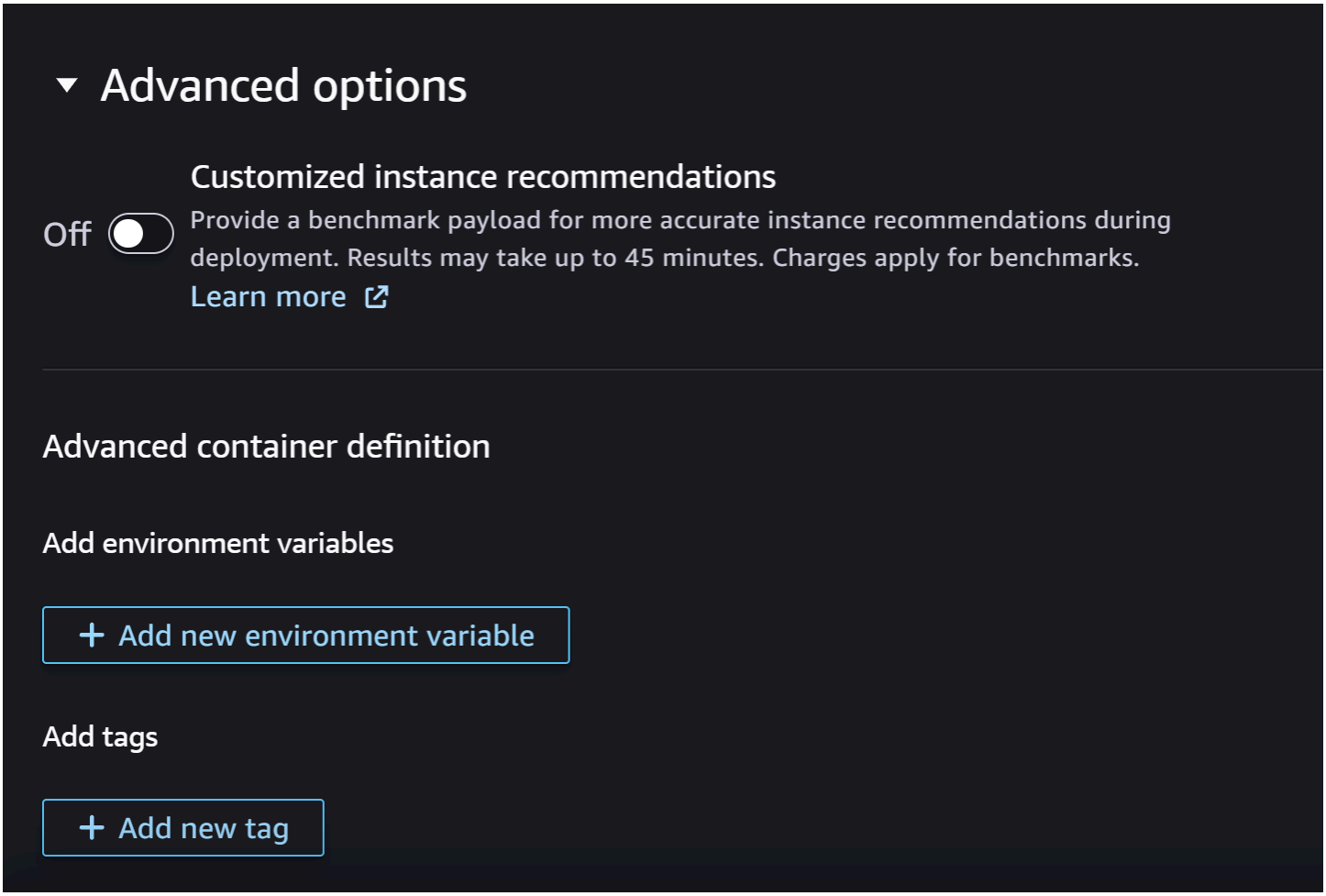
Bagian selanjutnya adalah Keamanan, yang terlihat seperti tangkapan layar berikut:



Untuk bagian Keamanan, lakukan hal berikut:

1. Untuk peran IAM, masukkan ARN untuk peran IAM.
2. (Opsional) Untuk Virtual Private Cloud (VPC), Anda dapat memilih Amazon VPC untuk menyimpan konfigurasi model dan artefak Anda.
3. (Opsional) Aktifkan sakelar Isolasi jaringan jika Anda ingin membatasi akses internet kontainer Anda.

Terakhir, Anda dapat secara opsional mengisi bagian Opsi lanjutan, yang terlihat seperti tangkapan layar berikut:



(Opsional) Untuk bagian Opsi lanjutan, lakukan hal berikut:

1. Aktifkan toggle Rekomendasi instans yang disesuaikan jika Anda ingin menjalankan pekerjaan Amazon SageMaker Inference Recommender pada model Anda setelah pembuatannya. Inference Recommender adalah fitur yang memberi Anda jenis instans yang direkomendasikan untuk mengoptimalkan kinerja dan biaya inferensi. Anda dapat melihat rekomendasi instans ini saat bersiap untuk menerapkan model Anda.
2. Untuk Tambahkan variabel lingkungan, masukkan variabel lingkungan untuk wadah Anda sebagai pasangan nilai kunci.
3. Untuk Tag, masukkan tag apa pun sebagai pasangan nilai kunci.
4. Setelah menyelesaikan konfigurasi model dan container Anda, pilih Create deployable model.

Anda sekarang harus memiliki model di SageMaker Studio yang siap untuk penerapan.

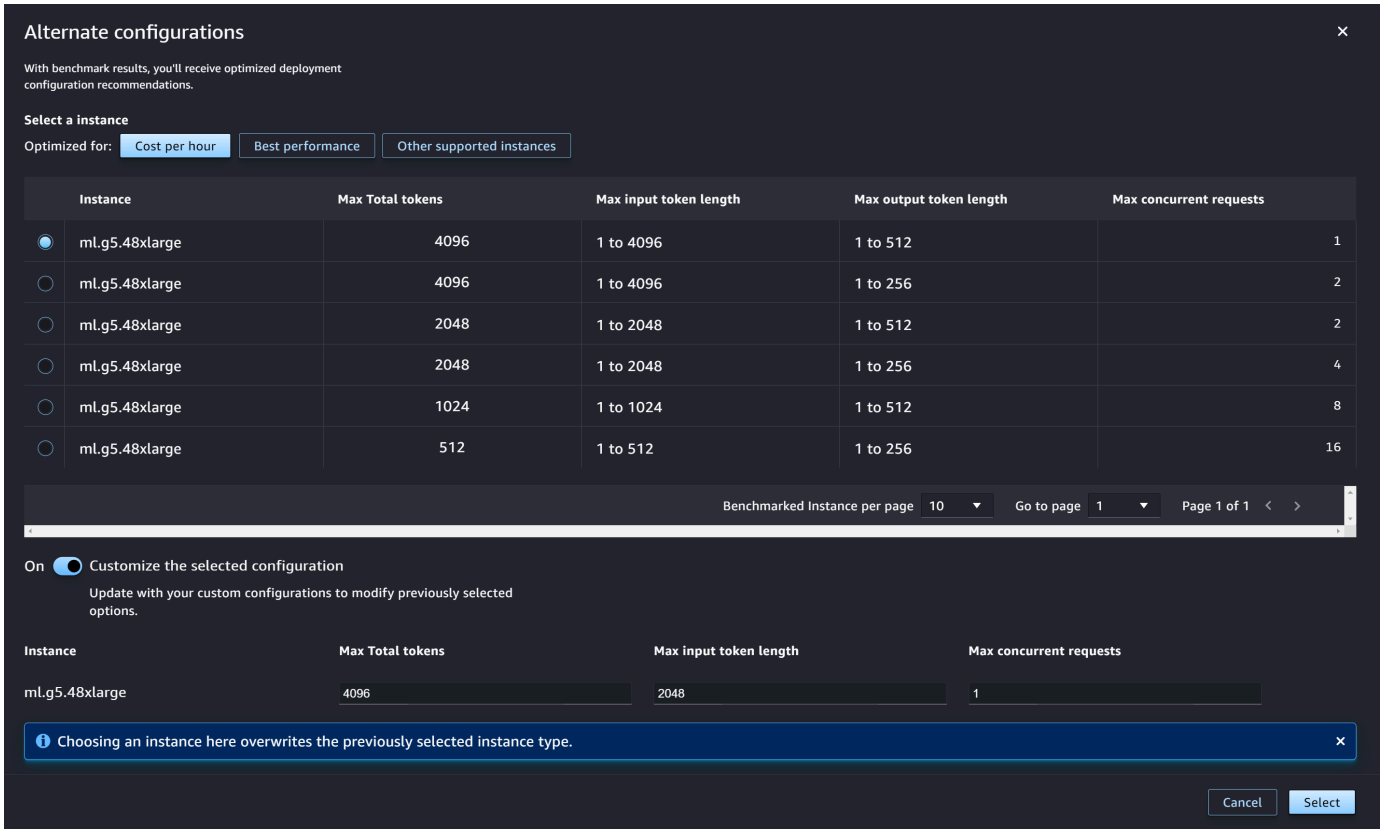
Terapkan model Anda

Terakhir, Anda menerapkan model yang Anda konfigurasikan pada langkah sebelumnya ke titik akhir HTTPS. Anda dapat menerapkan satu model atau beberapa model ke titik akhir.

Salah satu cara untuk menerapkan model adalah dengan melakukan hal berikut di Studio:

1. Buka aplikasi SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pada halaman Model, pilih satu atau beberapa model dari daftar SageMaker model.
4. Pilih Deploy.
5. Untuk nama Endpoint, buka menu dropdown. Anda dapat memilih titik akhir yang ada atau Anda dapat membuat titik akhir baru yang Anda gunakan model.
6. Untuk tipe Instance, pilih jenis instance yang ingin Anda gunakan untuk titik akhir. Jika sebelumnya Anda menjalankan tugas Inference Recommender untuk model tersebut, jenis instans yang Anda rekomendasikan akan muncul di daftar di bawah judul Recommended. Jika tidak, Anda akan melihat beberapa contoh Prospektif yang mungkin cocok untuk model Anda.
7. Untuk jumlah instans awal, masukkan jumlah awal instance yang ingin Anda berikan untuk titik akhir Anda.
8. Jika model yang Anda terapkan adalah salah satu SageMaker JumpStart LLM yang paling sering digunakan dari hub model, maka opsi Konfigurasi alternatif muncul setelah bidang jenis instance dan jumlah instance.

   Untuk SageMaker JumpStart LLM paling populer, AWS memiliki jenis instans pra-benchmark untuk mengoptimalkan biaya atau kinerja. Data ini dapat membantu Anda memutuskan jenis instans mana yang akan digunakan untuk menerapkan LLM Anda. Pilih Konfigurasi alternatif untuk membuka kotak dialog yang berisi data pra-benchmark. Panel terlihat seperti tangkapan layar berikut:



Di kotak Konfigurasi alternatif, lakukan hal berikut:

a. Pilih jenis instance. Anda dapat memilih Biaya per jam atau Kinerja terbaik untuk melihat jenis instans yang mengoptimalkan biaya atau kinerja untuk model yang ditentukan. Anda juga dapat memilih Instance lain yang didukung untuk melihat daftar jenis instans lain yang kompatibel dengan SageMaker JumpStart model. Perhatikan bahwa memilih jenis instance di sini menimpa pemilihan instance sebelumnya yang ditentukan dalam Langkah 6.

b. (Opsional) Aktifkan sakelar Sesuaikan konfigurasi yang dipilih untuk menentukan total token Maks (jumlah maksimum token yang ingin Anda izinkan, yang merupakan jumlah token masukan Anda dan output yang dihasilkan model), Panjang token masukan maksimum (jumlah maksimum token yang ingin Anda izinkan untuk masukan setiap permintaan), dan Permintaan bersamaan Max (jumlah maksimum permintaan yang dapat diproses model sekaligus).

c. Pilih Pilih untuk mengonfirmasi jenis instans dan pengaturan konfigurasi Anda.

9. Bidang Model seharusnya sudah diisi dengan nama model atau model yang Anda gunakan. Anda dapat memilih Tambahkan model untuk menambahkan lebih banyak model ke penerapan. Untuk setiap model yang Anda tambahkan, isi kolom berikut:

a. Untuk Jumlah inti CPU, masukkan inti CPU yang ingin Anda dedikasikan untuk penggunaan model.

b. Untuk jumlah salinan Min, masukkan jumlah minimum salinan model yang ingin Anda host di titik akhir pada waktu tertentu.

c. Untuk memori CPU Min (MB), masukkan jumlah minimum memori (dalam MB) yang dibutuhkan model.

d. Untuk memori CPU Max (MB), masukkan jumlah maksimum memori (dalam MB) yang ingin Anda izinkan model untuk digunakan.

10. (Opsional) Untuk opsi Lanjutan, lakukan hal berikut:

a. Untuk peran IAM, gunakan peran eksekusi SageMaker IAM default, atau tentukan peran Anda sendiri yang memiliki izin yang Anda butuhkan. Perhatikan bahwa peran IAM ini harus sama dengan peran yang Anda tentukan saat membuat model deployable.

b. Untuk Virtual Private Cloud (VPC), Anda dapat menentukan VPC tempat Anda ingin meng-host endpoint Anda.

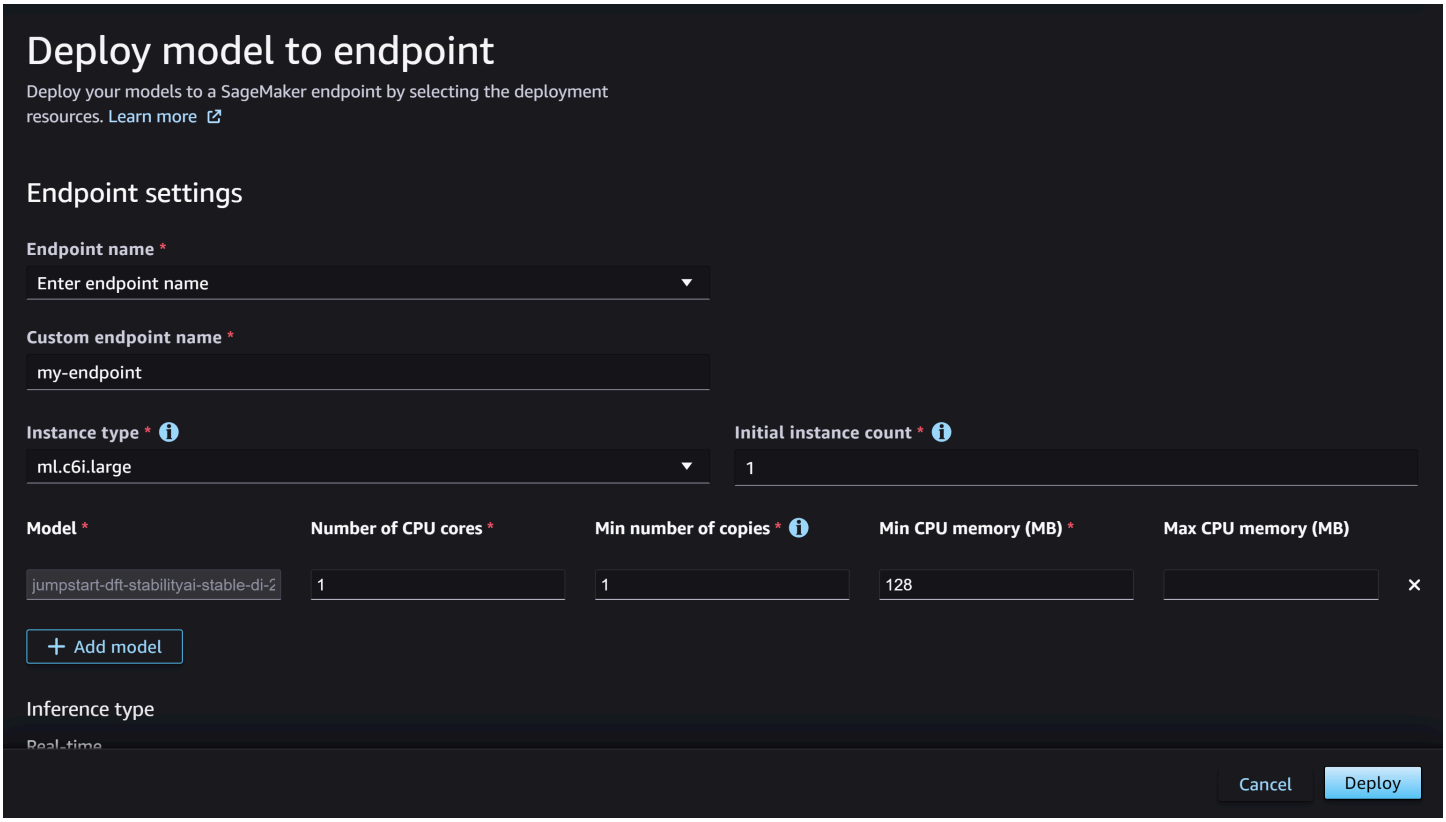
c. Untuk kunci Encryption KMS, pilih AWS KMS kunci untuk mengenkripsi data pada volume penyimpanan yang dilampirkan ke instance komputasi ML yang menghosting titik akhir.

d. Aktifkan sakelar Aktifkan isolasi jaringan untuk membatasi akses internet kontainer Anda.

e. Untuk konfigurasi Timeout, masukkan nilai untuk kolom batas waktu pengunduhan data Model (detik) dan batas waktu pemeriksaan kesehatan startup Container (detik). Nilai-nilai ini menentukan jumlah waktu maksimum yang SageMaker memungkinkan untuk mengunduh model ke wadah dan memulai wadah, masing-masing.

f. Untuk Tag, masukkan tag apa pun sebagai pasangan nilai kunci.

Setelah mengonfigurasi opsi Anda, halaman akan terlihat seperti tangkapan layar berikut.



Setelah mengonfigurasi penerapan Anda, pilih Deploy untuk membuat titik akhir dan menerapkan model Anda.

Terapkan model dengan SDK Python

Menggunakan SageMaker Python SDK, Anda dapat membangun model Anda dengan dua cara. Yang pertama adalah membuat objek model dari `ModelBuilder` kelas `Model` atau. Jika Anda menggunakan `Model` kelas untuk membuat `Model` objek, Anda perlu menentukan paket model atau kode inferensi (tergantung pada server model Anda), skrip untuk menangani serialisasi dan deserialisasi data antara klien dan server, dan dependensi apa pun yang akan diunggah ke Amazon S3 untuk konsumsi. Cara kedua untuk membangun model Anda adalah dengan menggunakan `ModelBuilder` artefak model atau kode inferensi yang Anda berikan. `ModelBuilder`secara otomatis menangkap dependensi Anda, menyimpulkan fungsi serialisasi dan deserialisasi yang diperlukan, dan mengemas dependensi Anda untuk membuat objek Anda. `Model` Untuk informasi selengkapnya tentang `ModelBuilder`, lihat Buat model di Amazon SageMaker dengan ModelBuilder.

Bagian berikut menjelaskan kedua metode untuk membuat model Anda dan menyebarkan objek model Anda.

Penyiapan

Contoh berikut mempersiapkan proses penyebaran model. Mereka mengimpor pustaka yang diperlukan dan menentukan URL S3 yang menempatkan artefak model.

SageMaker Python SDK

Example pernyataan impor

Contoh berikut mengimpor modul dari SageMaker Python SDK, SDK for Python (Boto3), dan Python Standard Library. Modul-modul ini menyediakan metode berguna yang membantu Anda menerapkan model, dan mereka digunakan oleh contoh lainnya yang mengikuti.

```
import boto3
from datetime import datetime
from sagemaker.compute_resource_requirements.resource_requirements import
 ResourceRequirements
from sagemaker.predictor import Predictor
from sagemaker.enums import EndpointType
from sagemaker.model import Model
from sagemaker.session import Session
```

boto3 inference components

Example pernyataan impor

Contoh berikut mengimpor modul dari SDK for Python (Boto3) dan Python Standard Library. Modul-modul ini menyediakan metode berguna yang membantu Anda menerapkan model, dan mereka digunakan oleh contoh lainnya yang mengikuti.

```
import boto3
import botocore
import sys
import time
```

boto3 models (without inference components)

Example pernyataan impor

Contoh berikut mengimpor modul dari SDK for Python (Boto3) dan Python Standard Library. Modul-modul ini menyediakan metode berguna yang membantu Anda menerapkan model, dan mereka digunakan oleh contoh lainnya yang mengikuti.

```
import boto3
import botocore
import datetime
from time import gmtime, strftime
```

Example URL artefak model

Kode berikut membangun contoh URL Amazon S3. URL menempatkan artefak model untuk model yang telah dilatih sebelumnya di bucket Amazon S3.

```
# Create a variable w/ the model S3 URL

# The name of your S3 bucket:
s3_bucket = "DOC-EXAMPLE-BUCKET"
# The directory within your S3 bucket your model is stored in:
bucket_prefix = "sagemaker/model/path"
# The file name of your model artifact:
model_filename = "my-model-artifact.tar.gz"
# Relative S3 path:
model_s3_key = f"{bucket_prefix}/"+model_filename
# Combine bucket name, model file name, and relate S3 path to create S3 model URL:
model_url = f"s3://{s3_bucket}/{model_s3_key}"
```

URL Amazon S3 lengkap disimpan dalam variabel`model_url`, yang digunakan dalam contoh berikut.

Ikhtisar

Ada beberapa cara Anda dapat menerapkan model dengan SageMaker Python SDK atau SDK for Python (Boto3). Bagian berikut merangkum langkah-langkah yang Anda selesaikan untuk beberapa kemungkinan pendekatan. Langkah-langkah ini ditunjukkan oleh contoh-contoh berikut.

SageMaker Python SDK

Menggunakan SageMaker Python SDK, Anda dapat membangun model Anda dengan salah satu cara berikut:

- Buat objek model dari **`Model`** kelas — Anda harus menentukan paket model atau kode inferensi (tergantung pada server model Anda), skrip untuk menangani serialisasi dan deserialisasi data antara klien dan server, dan dependensi apa pun yang akan diunggah ke Amazon S3 untuk konsumsi.

- Buat objek model dari **`ModelBuilder`** kelas — Anda menyediakan artefak model atau kode inferensi, dan `ModelBuilder` secara otomatis menangkap dependensi Anda, menyimpulkan fungsi serialisasi dan deserialisasi yang diperlukan, dan mengemas dependensi Anda untuk membuat objek Anda. `Model`

  Untuk informasi selengkapnya tentang `ModelBuilder`, lihat Buat model di Amazon SageMaker dengan ModelBuilder. Anda juga dapat melihat Package blog dan menyebarkan model MLM klasik dan LLM dengan mudah dengan SageMaker - Bagian 1 untuk informasi lebih lanjut.

Contoh berikut menjelaskan kedua metode untuk membuat model Anda dan menyebarkan objek model Anda. Untuk menerapkan model dengan cara ini, Anda menyelesaikan langkah-langkah berikut:

1. Tentukan sumber daya endpoint untuk dialokasikan ke model dengan `ResourceRequirements` objek.
2. Buat objek model dari `ModelBuilder` kelas `Model` atau. `ResourceRequirements`Objek ditentukan dalam pengaturan model.
3. Menyebarkan model ke titik akhir dengan menggunakan `deploy` metode objek. `Model`

boto3 inference components

Contoh berikut menunjukkan cara menetapkan model ke komponen inferensi dan kemudian menyebarkan komponen inferensi ke titik akhir. Untuk menerapkan model dengan cara ini, Anda menyelesaikan langkah-langkah berikut:

1. (Opsional) Buat objek SageMaker model dengan menggunakan `create_model`metode.
2. Tentukan pengaturan untuk titik akhir Anda dengan membuat objek konfigurasi titik akhir. Untuk membuatnya, Anda menggunakan `create_endpoint_config`metode ini.
3. Buat titik akhir Anda dengan menggunakan `create_endpoint`metode, dan dalam permintaan Anda, berikan konfigurasi titik akhir yang Anda buat.
4. Buat komponen inferensi dengan menggunakan `create_inference_component` metode. Dalam pengaturan, Anda menentukan model dengan melakukan salah satu dari berikut ini:
   - Menentukan objek SageMaker model
   - Menentukan URI gambar model dan URL S3

Anda juga mengalokasikan sumber daya titik akhir ke model. Dengan membuat komponen inferensi, Anda menerapkan model ke titik akhir. Anda dapat menerapkan beberapa model ke titik akhir dengan membuat beberapa komponen inferensi — satu untuk setiap model.

boto3 models (without inference components)

Contoh berikut menunjukkan cara membuat objek model dan kemudian menyebarkan model ke titik akhir. Untuk menerapkan model dengan cara ini, Anda menyelesaikan langkah-langkah berikut:

1. Buat SageMaker model dengan menggunakan `create_model`metode.
2. Tentukan pengaturan untuk titik akhir Anda dengan membuat objek konfigurasi titik akhir. Untuk membuatnya, Anda menggunakan `create_endpoint_config`metode ini. Dalam konfigurasi endpoint, Anda menetapkan objek model ke varian produksi.
3. Buat titik akhir Anda dengan menggunakan `create_endpoint`metode ini. Dalam permintaan Anda, berikan konfigurasi titik akhir yang Anda buat.

   Saat Anda membuat titik akhir, berikan SageMaker sumber daya titik akhir, dan menerapkan model ke titik akhir.

Konfigurasi

Contoh berikut mengonfigurasi sumber daya yang Anda perlukan untuk menerapkan model ke titik akhir.

SageMaker Python SDK

Contoh berikut menetapkan sumber daya endpoint untuk model dengan objek. `ResourceRequirements` Sumber daya ini termasuk inti CPU, akselerator, dan memori. Kemudian, contoh membuat objek model dari `Model` kelas. Atau Anda dapat membuat objek model dengan membuat instance ModelBuilderkelas dan menjalankan `build` —metode ini juga ditampilkan dalam contoh. `ModelBuilder`menyediakan antarmuka terpadu untuk kemasan model, dan dalam hal ini, menyiapkan model untuk penerapan model besar. Contoh ini digunakan `ModelBuilder` untuk membangun model Hugging Face. (Anda juga dapat melewati JumpStart model). Setelah Anda membangun model, Anda dapat menentukan persyaratan sumber daya

dalam objek model. Pada langkah berikutnya, Anda menggunakan objek ini untuk menyebarkan model ke titik akhir.

```
resources = ResourceRequirements(
    requests = {
        "num_cpus": 2,  # Number of CPU cores required:
        "num_accelerators": 1, # Number of accelerators required
        "memory": 8192,  # Minimum memory required in Mb (required)
        "copies": 1,
    },
    limits = {},
)

now = datetime.now()
dt_string = now.strftime("%d-%m-%Y-%H-%M-%S")
model_name = "my-sm-model"+dt_string

# build your model with Model class
model = Model(
    name = "model-name",
    image_uri = "image-uri",
    model_data = model_url,
    role = "arn:aws:iam::111122223333:role/service-role/role-name",
    resources = resources,
    predictor_cls = Predictor,
)

# Alternate mechanism using ModelBuilder
# uncomment the following section to use ModelBuilder
/*
model_builder = ModelBuilder(
    model="<HuggingFace-ID>", # like "meta-llama/Llama-2-7b-hf"
    schema_builder=SchemaBuilder(sample_input,sample_output),
    env_vars={ "HUGGING_FACE_HUB_TOKEN": "<HuggingFace_token>}" }
)

# build your Model object
model = model_builder.build()

# create a unique name from string 'mb-inference-component'
model.model_name = unique_name_from_base("mb-inference-component")

# assign resources to your model
```

```
model.resources = resources
*/
```

boto3 inference components

Contoh berikut mengkonfigurasi titik akhir dengan metode. `create_endpoint_config` Anda menetapkan konfigurasi ini ke titik akhir saat Anda membuatnya. Dalam konfigurasi, Anda menentukan satu atau lebih varian produksi. Untuk setiap varian, Anda dapat memilih jenis instans yang ingin disediakan Amazon SageMaker , dan Anda dapat mengaktifkan penskalaan instans terkelola.

```
endpoint_config_name = "endpoint-config-name"
endpoint_name = "endpoint-name"
inference_component_name = "inference-component-name"
variant_name = "variant-name"

sagemaker_client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ExecutionRoleArn = "arn:aws:iam::111122223333:role/service-role/role-name",
    ProductionVariants = [
        {
            "VariantName": variant_name,
            "InstanceType": "ml.p4d.24xlarge",
            "InitialInstanceCount": 1,
            "ManagedInstanceScaling": {
                "Status": "ENABLED",
                "MinInstanceCount": 1,
                "MaxInstanceCount": 2,
            },
        }
    ],
)
```

boto3 models (without inference components)

Example definisi model

Contoh berikut mendefinisikan SageMaker model dengan `create_model` metode dalam. AWS SDK for Python (Boto3)

```
model_name = "model-name"
```

```
create_model_response = sagemaker_client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = "arn:aws:iam::111122223333:role/service-role/role-name",
    PrimaryContainer = {
        "Image": "image-uri",
        "ModelDataUrl": model_url,
    }
)
```

Contoh ini menentukan yang berikut:

- `ModelName`: Nama untuk model Anda (dalam contoh ini disimpan sebagai variabel string yang disebut`model_name`).
- `ExecutionRoleArn`: Nama Sumber Daya Amazon (ARN) dari peran IAM yang SageMaker dapat diasumsikan Amazon untuk mengakses artefak model dan gambar Docker untuk penerapan pada instance komputasi HTML atau untuk pekerjaan transformasi batch.
- `PrimaryContainer`: Lokasi gambar Docker utama yang berisi kode inferensi, artefak terkait, dan peta lingkungan khusus yang digunakan kode inferensi saat model diterapkan untuk prediksi.

Example konfigurasi titik akhir

Contoh berikut mengkonfigurasi titik akhir dengan metode. `create_endpoint_config` Amazon SageMaker menggunakan konfigurasi ini untuk menyebarkan model. Dalam konfigurasi, Anda mengidentifikasi satu atau beberapa model, yang dibuat dengan `create_model` metode, untuk menyebarkan sumber daya yang SageMaker ingin disediakan Amazon.

```
endpoint_config_response = sagemaker_client.create_endpoint_config(
    EndpointConfigName = "endpoint-config-name",
    # List of ProductionVariant objects, one for each model that you want to host at
 this endpoint:
    ProductionVariants = [
        {
            "VariantName": "variant-name", # The name of the production variant.
            "ModelName": model_name,
            "InstanceType": "ml.p4d.24xlarge",
            "InitialInstanceCount": 1 # Number of instances to launch initially.
        }
    ]
)
```

Contoh ini menentukan kunci berikut untuk `ProductionVariants` bidang:

- `VariantName`: Nama varian produksi.
- `ModelName`: Nama model yang ingin Anda host. Ini adalah nama yang Anda tentukan saat membuat model.
- `InstanceType`: Jenis contoh komputasi. Lihat `InstanceType` bidang di https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_ProductionVariant.html dan SageMakerHarga untuk daftar jenis instans komputasi yang didukung dan harga untuk setiap jenis instans.

Deploy

Contoh berikut menyebarkan model ke titik akhir.

SageMaker Python SDK

Contoh berikut menyebarkan model ke titik akhir HTTPS real-time dengan `deploy` metode objek model. Jika Anda menentukan nilai `resources` argumen untuk pembuatan dan penerapan model, sumber daya yang Anda tentukan untuk penerapan akan diutamakan.

```
predictor = model.deploy(
    initial_instance_count = 1,
    instance_type = "ml.p4d.24xlarge",
    endpoint_type = EndpointType.INFERENCE_COMPONENT_BASED,
    resources = resources,
)
```

Untuk `instance_type` bidang, contoh menentukan nama jenis instans Amazon EC2 untuk model. Untuk `initial_instance_count` bidang, ini menentukan jumlah awal instance untuk menjalankan endpoint pada.

Contoh kode berikut menunjukkan kasus lain di mana Anda menerapkan model ke titik akhir dan kemudian menerapkan model lain ke titik akhir yang sama. Dalam hal ini Anda harus memberikan nama titik akhir yang sama ke `deploy` metode kedua model.

```
# Deploy the model to inference-component-based endpoint
falcon_predictor = falcon_model.deploy(
    initial_instance_count = 1,
    instance_type = "ml.p4d.24xlarge",
    endpoint_type = EndpointType.INFERENCE_COMPONENT_BASED,
```

```
    endpoint_name = "<endpoint_name>"
    resources = resources,
)

# Deploy another model to the same inference-component-based endpoint
llama2_predictor = llama2_model.deploy( # resources already set inside llama2_model
    endpoint_type = EndpointType.INFERENCE_COMPONENT_BASED,
    endpoint_name = "<endpoint_name>"  # same endpoint name as for falcon model
)
```

boto3 inference components

Setelah Anda memiliki konfigurasi endpoint, gunakan metode create_endpoint untuk membuat endpoint Anda. Nama endpoint harus unik Wilayah AWS dalam AWS akun Anda.

Contoh berikut membuat endpoint menggunakan konfigurasi endpoint yang ditentukan dalam permintaan. Amazon SageMaker menggunakan titik akhir untuk menyediakan sumber daya.

```
sagemaker_client.create_endpoint(
    EndpointName = endpoint_name,
    EndpointConfigName = endpoint_config_name,
)
```

Setelah Anda membuat titik akhir, Anda dapat menerapkan satu atau model ke sana dengan membuat komponen inferensi. Contoh berikut membuat satu dengan `create_inference_component` metode.

```
sagemaker_client.create_inference_component(
    InferenceComponentName = inference_component_name,
    EndpointName = endpoint_name,
    VariantName = variant_name,
    Specification = {
        "Container": {
            "Image": "image-uri",
            "ArtifactUrl": model_url,
        },
        "ComputeResourceRequirements": {
            "NumberOfCpuCoresRequired": 1,
            "MinMemoryRequiredInMb": 1024
        }
    },
    RuntimeConfig = {"CopyCount": 2}
```

```
)
```

boto3 models (without inference components)

Example deployment

Berikan konfigurasi titik akhir ke SageMaker. Layanan meluncurkan instance komputasi ML dan menerapkan model atau model seperti yang ditentukan dalam konfigurasi.

Setelah Anda memiliki konfigurasi model dan titik akhir, gunakan metode create_endpoint untuk membuat titik akhir Anda. Nama endpoint harus unik Wilayah AWS dalam AWS akun Anda.

Contoh berikut membuat endpoint menggunakan konfigurasi endpoint yang ditentukan dalam permintaan. Amazon SageMaker menggunakan titik akhir untuk menyediakan sumber daya dan menerapkan model.

```
create_endpoint_response = sagemaker_client.create_endpoint(
    # The endpoint name must be unique within an AWS Region in your AWS account:
    EndpointName = "endpoint-name"
    # The name of the endpoint configuration associated with this endpoint:
    EndpointConfigName = "endpoint-config-name")
```

Menyebarkan model dengan AWS CLI

Anda dapat menerapkan model ke titik akhir dengan menggunakan. AWS CLI

Ikhtisar

Saat Anda menerapkan model denganAWS CLI, Anda dapat menerapkannya dengan atau tanpa menggunakan komponen inferensi. Bagian berikut merangkum perintah yang Anda jalankan untuk kedua pendekatan. Perintah-perintah ini ditunjukkan oleh contoh-contoh berikut.

With inference components

Untuk menerapkan model dengan komponen inferensi, lakukan hal berikut:

1. (Opsional) Buat model dengan `create-model`perintah.
2. Tentukan pengaturan untuk titik akhir Anda dengan membuat konfigurasi titik akhir. Untuk membuatnya, Anda menjalankan `create-endpoint-config`perintah.
3. Buat titik akhir Anda dengan menggunakan `create-endpoint`perintah. Di badan perintah, tentukan konfigurasi titik akhir yang Anda buat.

4. Buat komponen inferensi dengan menggunakan `create-inference-component` perintah. Dalam pengaturan, Anda menentukan model dengan melakukan salah satu dari berikut ini:

   - Menentukan objek SageMaker model
   - Menentukan URI gambar model dan URL S3

   Anda juga mengalokasikan sumber daya titik akhir ke model. Dengan membuat komponen inferensi, Anda menerapkan model ke titik akhir. Anda dapat menerapkan beberapa model ke titik akhir dengan membuat beberapa komponen inferensi — satu untuk setiap model.

Without inference components

Untuk menerapkan model tanpa menggunakan komponen inferensi, lakukan hal berikut:

1. Buat SageMaker model dengan menggunakan `create-model`perintah.
2. Tentukan pengaturan untuk titik akhir Anda dengan membuat objek konfigurasi titik akhir. Untuk membuatnya, Anda menggunakan `create-endpoint-config`perintah. Dalam konfigurasi endpoint, Anda menetapkan objek model ke varian produksi.
3. Buat titik akhir Anda dengan menggunakan `create-endpoint`perintah. Di badan perintah Anda, tentukan konfigurasi titik akhir yang Anda buat.

   Saat Anda membuat titik akhir, berikan SageMaker sumber daya titik akhir, dan menerapkan model ke titik akhir.

Konfigurasi

Contoh berikut mengonfigurasi sumber daya yang Anda perlukan untuk menerapkan model ke titik akhir.

With inference components

Example create-endpoint-config perintah

Contoh berikut membuat konfigurasi endpoint dengan create-endpoint-configperintah.

```
aws sagemaker create-endpoint-config \
--endpoint-config-name endpoint-config-name \
--execution-role-arn arn:aws:iam::111122223333:role/service-role/role-name\
```

```
--production-variants file://production-variants.json
```

Dalam contoh ini, file `production-variants.json` mendefinisikan varian produksi dengan JSON berikut:

```
[
    {
        "VariantName": "variant-name",
        "ModelName": "model-name",
        "InstanceType": "ml.p4d.24xlarge",
        "InitialInstanceCount": 1
    }
]
```

Jika perintah berhasil, AWS CLI merespons dengan ARN untuk sumber daya yang Anda buat.

```
{
    "EndpointConfigArn": "arn:aws:sagemaker:us-west-2:111122223333:endpoint-config/
endpoint-config-name"
}
```

Without inference components

Example perintah buat-model

Contoh berikut membuat model dengan perintah create-model.

```
aws sagemaker create-model \
--model-name model-name \
--execution-role-arn arn:aws:iam::111122223333:role/service-role/role-name \
--primary-container "{ \"Image\": \"image-uri\", \"ModelDataUrl\": \"model-s3-
url\"}"
```

Jika perintah berhasil, AWS CLI merespons dengan ARN untuk sumber daya yang Anda buat.

```
{
    "ModelArn": "arn:aws:sagemaker:us-west-2:111122223333:model/model-name"
}
```

Example create-endpoint-config perintah

Contoh berikut membuat konfigurasi endpoint dengan create-endpoint-configperintah.

```
aws sagemaker create-endpoint-config \
--endpoint-config-name endpoint-config-name \
--production-variants file://production-variants.json
```

Dalam contoh ini, file `production-variants.json` mendefinisikan varian produksi dengan JSON berikut:

```
[
    {
        "VariantName": "variant-name",
        "ModelName": "model-name",
        "InstanceType": "ml.p4d.24xlarge",
        "InitialInstanceCount": 1
    }
]
```

Jika perintah berhasil, AWS CLI merespons dengan ARN untuk sumber daya yang Anda buat.

```
{
    "EndpointConfigArn": "arn:aws:sagemaker:us-west-2:111122223333:endpoint-config/
endpoint-config-name"
}
```

Deploy

Contoh berikut menyebarkan model ke titik akhir.

With inference components

Example perintah buat-titik akhir

Contoh berikut membuat endpoint dengan perintah create-endpoint.

```
aws sagemaker create-endpoint \
--endpoint-name endpoint-name \
--endpoint-config-name endpoint-config-name
```

Jika perintah berhasil, AWS CLI merespons dengan ARN untuk sumber daya yang Anda buat.

```
{
```

```
    "EndpointArn": "arn:aws:sagemaker:us-west-2:111122223333:endpoint/endpoint-name"
}
```

Example create-inference-component perintah

Contoh berikut membuat komponen inferensi dengan create-inference-component perintah.

```
aws sagemaker create-inference-component \
--inference-component-name inference-component-name \
--endpoint-name endpoint-name \
--variant-name variant-name \
--specification file://specification.json \
--runtime-config "{\"CopyCount\": 2}"
```

Dalam contoh ini, file `specification.json` mendefinisikan wadah dan menghitung sumber daya dengan JSON berikut:

```
{
    "Container": {
        "Image": "image-uri",
        "ArtifactUrl": "model-s3-url"
    },
    "ComputeResourceRequirements": {
        "NumberOfCpuCoresRequired": 1,
        "MinMemoryRequiredInMb": 1024
    }
}
```

Jika perintah berhasil, AWS CLI merespons dengan ARN untuk sumber daya yang Anda buat.

```
{
    "InferenceComponentArn": "arn:aws:sagemaker:us-west-2:111122223333:inference-
component/inference-component-name"
}
```

Without inference components

Example perintah buat-titik akhir

Contoh berikut membuat endpoint dengan perintah create-endpoint.

```
aws sagemaker create-endpoint \
--endpoint-name endpoint-name \
--endpoint-config-name endpoint-config-name
```

Jika perintah berhasil, AWS CLI merespons dengan ARN untuk sumber daya yang Anda buat.

```
{
    "EndpointArn": "arn:aws:sagemaker:us-west-2:111122223333:endpoint/endpoint-name"
}
```

Apa yang harus dilakukan selanjutnya

- Memanggil titik akhir waktu nyata
- Kelola titik akhir waktu nyata
- Secara Otomatis Menskalakan SageMaker Model Amazon

Memanggil model untuk inferensi waktu nyata

Setelah menerapkan model Anda menggunakan layanan SageMaker hosting, Anda dapat menguji model Anda pada titik akhir tersebut dengan mengirimkan data pengujian. Anda dapat menguji titik akhir menggunakan Amazon SageMaker Studio Classic, AWS SDK, atau. AWS CLI

Memanggil Endpoint Anda Menggunakan Amazon Studio Classic SageMaker

Setelah menerapkan model ke titik akhir, Anda dapat memeriksa titik akhir tersebut dengan Amazon SageMaker Studio Classic.

> **Note**
>
> Catatan: SageMaker hanya mendukung pengujian titik akhir dengan Amazon SageMaker Studio Classic untuk titik akhir real-time.

Untuk mengirim permintaan inferensi pengujian ke titik akhir Anda

1. Luncurkan Amazon SageMaker Studio Classic.
2. Di bagian Beranda panel navigasi di sebelah kiri, pilih Deployment.

3. Dari dropdown, pilih Endpoints.
4. Temukan titik akhir Anda berdasarkan nama, dan pilih nama di tabel. Nama titik akhir yang tercantum di panel Endpoints ditentukan saat Anda menerapkan model. Ruang kerja Studio membuka halaman Endpoints di tab baru.
5. Di tab Inferensi uji, kirim permintaan ke titik akhir Anda dengan memberikan data sampel dalam format JSON. Gunakan editor JSON untuk mengirimkan permintaan ke titik akhir Anda.
6. (Opsional) Anda dapat memberikan URL khusus untuk mengirim permintaan Anda. Di bagian Configure endpoint URL dan header, untuk kolom Custom URL, berikan URL tempat model Anda di-host. Biarkan bidang ini kosong jika Anda menggunakan SageMaker titik akhir. Di bawah Header, Anda juga dapat menambahkan header nilai kunci secara opsional untuk meneruskan informasi tambahan dengan permintaan inferensi.
7. Pilih Kirim Permintaan. Studio menunjukkan output inferensi dalam kartu di sebelah kanan editor JSON.

Bagian atas kartu menunjukkan jenis permintaan yang dikirim ke titik akhir (hanya JSON yang diterima). Kartu menunjukkan bidang-bidang berikut:

- Status - menampilkan salah satu jenis status berikut:
  - `Complete`Permintaan itu berhasil.
  - `Failed`— Permintaan gagal. Respons muncul di bawah Failure Reason.
  - `Pending`— Saat permintaan inferensi tertunda, status menunjukkan ikon melingkar yang berputar.
- Panjang Eksekusi — Berapa lama waktu pemanggilan (waktu akhir dikurangi waktu mulai) dalam milidetik.
- Waktu Permintaan — Berapa menit telah berlalu sejak permintaan dikirim.
- Waktu Hasil — Berapa menit telah berlalu sejak hasilnya dikembalikan.

Memanggil Endpoint Anda dengan Menggunakan AWS SDK for Python (Boto3)

Setelah menerapkan model ke titik akhir, Anda dapat memeriksa titik akhir dengan menggunakan salah satu AWS SDK, termasuk sebagai. AWS SDK for Python (Boto3) Untuk menguji titik akhir Anda dengan SDK ini, Anda menggunakan salah satu metode berikut:

- `invoke_endpoint`— Mengirim permintaan inferensi ke titik akhir model dan mengembalikan respons yang dihasilkan model. Metode ini mengembalikan muatan inferensi sebagai satu respons

setelah model selesai menghasilkannya. Untuk informasi selengkapnya, lihat invoke_endpoint di Referensi API AWSSDK for Python (Boto3).

- `invoke_endpoint_with_response_stream`— Mengirim permintaan inferensi ke titik akhir model dan mengalirkan respons di bagian tambahan sementara model menghasilkan inferensi. Dengan metode ini, aplikasi klien Anda segera mulai menerima bagian dari respons saat bagian-bagiannya tersedia. Klien Anda tidak perlu menunggu model menghasilkan seluruh muatan respons. Anda dapat menerapkan streaming untuk mendukung pengalaman interaktif yang cepat, seperti chatbots, asisten virtual, dan generator musik.

  Gunakan metode ini hanya untuk memanggil model yang mendukung streaming inferensi.

  Ketika sebuah wadah menangani permintaan inferensi streaming, ia mengembalikan inferensi model sebagai serangkaian bagian secara bertahap saat model menghasilkannya. Aplikasi klien mulai menerima tanggapan segera ketika tersedia. Mereka tidak perlu menunggu model untuk menghasilkan seluruh respons. Anda dapat menerapkan streaming untuk mendukung pengalaman interaktif yang cepat, seperti chatbots, asisten virtual, dan generator musik.

Sebelum Anda dapat menggunakan metode ini dalam kode klien Anda, Anda harus membuat klien SageMaker Runtime, dan Anda harus menentukan nama titik akhir Anda. Contoh berikut mengatur klien dan titik akhir untuk sisa contoh berikut:

```
import boto3

# Create a low-level client representing Amazon SageMaker Runtime
sagemaker_runtime = boto3.client(
    "sagemaker-runtime", region_name='aws_region')

# The endpoint name must be unique within
# an AWS Region in your AWS account.
endpoint_name='endpoint-name'
```

Memohon untuk Mendapatkan Respons Inferensi

Contoh berikut menggunakan `invoke_endpoint` metode untuk memanggil titik akhir dengan: AWS SDK for Python (Boto3)

```
# Gets inference from the model hosted at the specified endpoint:
response = sagemaker_runtime.invoke_endpoint(
    EndpointName=endpoint_name,
```

```
    Body=bytes('{"features": ["This is great!"]}', 'utf-8')
    )

# Decodes and prints the response body:
print(response['Body'].read().decode('utf-8'))
```

Contoh ini memberikan data input di `Body` lapangan SageMaker untuk diteruskan ke model. Data ini harus dalam format yang sama dengan yang digunakan untuk pelatihan. Contoh menyimpan respons dalam `response` variabel.

`response`Variabel menyediakan akses ke status HTTP, nama model yang digunakan, dan bidang lainnya. Cuplikan berikut mencetak: `HTTPStatusCode`

```
print(response["HTTPStatusCode"])
```

Memanggil untuk Streaming Respons Inferensi

Jika Anda menerapkan model yang mendukung streaming inferensi, Anda dapat memanggil model untuk menerima muatan inferensinya sebagai aliran suku cadang. Model memberikan bagian-bagian ini secara bertahap saat model menghasilkannya. Saat aplikasi menerima aliran inferensi, aplikasi tidak perlu menunggu model menghasilkan seluruh muatan respons. Sebagai gantinya, aplikasi segera mulai menerima bagian dari respons saat tersedia.

Dengan menggunakan aliran inferensi dalam aplikasi Anda, Anda dapat membuat interaksi di mana pengguna Anda menganggap inferensi menjadi cepat karena mereka mendapatkan bagian pertama dengan segera. Misalnya, Anda dapat membuat chatbot yang secara bertahap menampilkan teks yang dihasilkan oleh model bahasa besar (LLM).

Untuk mendapatkan aliran inferensi, Anda dapat menggunakan `invoke_endpoint_with_response_stream` metode di SDK for Python (Boto3). Dalam badan respons, SDK menyediakan `EventStream` objek, yang memberikan inferensi sebagai serangkaian `PayloadPart` objek.

Example Aliran Inferensi

Contoh berikut adalah aliran `PayloadPart` objek:

```
{'PayloadPart': {'Bytes': b'{"outputs": [" a"]}\n'}}
{'PayloadPart': {'Bytes': b'{"outputs": [" challenging"]}\n'}}
{'PayloadPart': {'Bytes': b'{"outputs": [" problem"]}\n'}}
. . .
```

Di setiap bagian muatan, `Bytes` bidang menyediakan sebagian dari respons inferensi dari model. Bagian ini dapat berupa jenis konten apa pun yang dihasilkan model, seperti teks, gambar, atau data audio. Dalam contoh ini, bagian-bagiannya adalah objek JSON yang berisi teks yang dihasilkan dari LLM.

Biasanya, bagian payload berisi potongan data terpisah dari model. Dalam contoh ini, potongan diskrit adalah objek JSON utuh. Kadang-kadang, respons streaming membagi potongan menjadi beberapa bagian muatan, atau menggabungkan beberapa potongan menjadi satu bagian muatan. Contoh berikut menunjukkan potongan data dalam format JSON yang dibagi menjadi dua bagian payload:

```
{'PayloadPart': {'Bytes': b'{"outputs": '}}
{'PayloadPart': {'Bytes': b'[" problem"]}\n'}}
```

Saat Anda menulis kode aplikasi yang memproses aliran inferensi, sertakan logika yang menangani pemisahan dan kombinasi data sesekali ini. Sebagai salah satu strategi, Anda dapat menulis kode yang menggabungkan konten `Bytes` saat aplikasi Anda menerima bagian payload. Dengan menggabungkan contoh data JSON di sini, Anda akan menggabungkan data menjadi badan JSON yang dibatasi baris baru. Kemudian, kode Anda dapat memproses aliran dengan mengurai seluruh objek JSON pada setiap baris.

Contoh berikut menunjukkan JSON yang dibatasi baris baru yang akan Anda buat saat Anda menggabungkan isi contoh: `Bytes`

```
{"outputs": [" a"]}
{"outputs": [" challenging"]}
{"outputs": [" problem"]}
. . .
```

Example Kode untuk Memproses Aliran Inferensi

Contoh kelas Python berikut,`SmrInferenceStream`, menunjukkan bagaimana Anda dapat memproses aliran inferensi yang mengirimkan data teks dalam format JSON:

```
import io
import json

# Example class that processes an inference stream:
class SmrInferenceStream:
```

```
    def __init__(self, sagemaker_runtime, endpoint_name):
        self.sagemaker_runtime = sagemaker_runtime
        self.endpoint_name = endpoint_name
        # A buffered I/O stream to combine the payload parts:
        self.buff = io.BytesIO()
        self.read_pos = 0

    def stream_inference(self, request_body):
        # Gets a streaming inference response
        # from the specified model endpoint:
        response = self.sagemaker_runtime\
            .invoke_endpoint_with_response_stream(
                EndpointName=self.endpoint_name,
                Body=json.dumps(request_body),
                ContentType="application/json"
        )
        # Gets the EventStream object returned by the SDK:
        event_stream = response['Body']
        for event in event_stream:
            # Passes the contents of each payload part
            # to be concatenated:
            self._write(event['PayloadPart']['Bytes'])
            # Iterates over lines to parse whole JSON objects:
            for line in self._readlines():
                resp = json.loads(line)
                part = resp.get("outputs")[0]
                # Returns parts incrementally:
                yield part

    # Writes to the buffer to concatenate the contents of the parts:
    def _write(self, content):
        self.buff.seek(0, io.SEEK_END)
        self.buff.write(content)

    # The JSON objects in buffer end with '\n'.
    # This method reads lines to yield a series of JSON objects:
    def _readlines(self):
        self.buff.seek(self.read_pos)
        for line in self.buff.readlines():
            self.read_pos += len(line)
            yield line[:-1]
```

Contoh ini memproses aliran inferensi dengan melakukan hal berikut:

- Diinisialisasi dengan klien SageMaker Runtime dan nama titik akhir model. Sebelum Anda bisa mendapatkan aliran inferensi, model yang dihosting endpoint harus mendukung streaming inferensi.
- Dalam `stream_inference` metode contoh, menerima badan permintaan dan meneruskannya ke `invoke_endpoint_with_response_stream` metode SDK.
- Mengulangi setiap peristiwa dalam `EventStream` objek yang dikembalikan SDK.
- Dari setiap peristiwa, dapatkan isi `Bytes` objek dalam `PayloadPart` objek.
- Dalam `_write` metode contoh, menulis ke buffer untuk menggabungkan isi objek. `Bytes` Konten gabungan membentuk badan JSON yang dibatasi baris baru.
- Menggunakan `_readlines` metode contoh untuk mendapatkan serangkaian objek JSON yang dapat diulang.
- Di setiap objek JSON, mendapat sepotong inferensi.
- Dengan `yield` ekspresi, mengembalikan potongan secara bertahap.

Contoh berikut membuat dan menggunakan `SmrInferenceStream` objek:

```
request_body = {"inputs": ["Large model inference is"],
                "parameters": {"max_new_tokens": 100,
                               "enable_sampling": "true"}}
smr_inference_stream = SmrInferenceStream(
    sagemaker_runtime, endpoint_name)
stream = smr_inference_stream.stream_inference(request_body)
for part in stream:
    print(part, end='')
```

Contoh ini meneruskan badan permintaan ke `stream_inference` metode. Ini mengulangi respons untuk mencetak setiap bagian yang dikembalikan oleh aliran inferensi.

Contoh mengasumsikan bahwa model pada titik akhir yang ditentukan adalah LLM yang menghasilkan teks. Output dari contoh ini adalah kumpulan teks yang dihasilkan yang mencetak secara bertahap:

```
a challenging problem in machine learning. The goal is to . . .
```

Memanggil Endpoint Anda dengan Menggunakan AWS CLI

Anda dapat menguji titik akhir Anda dengan menjalankan perintah dengan AWS Command Line Interface (AWS CLI). AWS CLIMendukung permintaan inferensi standar dengan `invoke-endpoint` perintah, dan mendukung permintaan inferensi asinkron dengan perintah. `invoke-endpoint-async`

> **Note**
>
> AWS CLIItu tidak mendukung permintaan inferensi streaming.

Contoh berikut menggunakan `invoke-endpoint` perintah untuk mengirim permintaan inferensi ke titik akhir model:

```
aws sagemaker-runtime invoke-endpoint \
    --endpoint-name endpoint_name \
    --body fileb://$file_name \
    output_file.txt
```

Untuk `--endpoint-name` parameter, berikan nama yang Anda tentukan `EndpointName` saat Anda membuat titik akhir. `CreateEndpoint` Untuk `--body` parameter, berikan data input SageMaker untuk diteruskan ke model. Data harus dalam format yang sama dengan yang digunakan untuk pelatihan. Contoh ini menunjukkan cara mengirim data biner ke titik akhir Anda.

Untuk informasi selengkapnya tentang `fileb://` kapan harus menggunakan `file://` over saat meneruskan konten file ke parameterAWS CLI, lihat Praktik Terbaik untuk Parameter File Lokal.

Untuk informasi selengkapnya, dan untuk melihat parameter tambahan yang dapat Anda lewati, lihat `invoke-endpoint`di AWS CLICommand Reference.

Jika `invoke-endpoint` perintah berhasil mengembalikan respon seperti berikut:

```
{
    "ContentType": "<content_type>; charset=utf-8",
    "InvokedProductionVariant": "<Variant>"
}
```

Jika perintah tidak berhasil, periksa apakah muatan input dalam format yang benar.

Lihat output dari pemanggilan dengan memeriksa file output file (`output_file.txt`dalam contoh ini).

```
more output_file.txt
```

Kelola titik akhir Anda

Setelah menerapkan model Anda ke titik akhir, Anda mungkin ingin melihat dan mengelola titik akhir. Dengan SageMaker, Anda dapat melihat status dan detail titik akhir Anda, memeriksa metrik dan log untuk memantau kinerja titik akhir Anda, memperbarui model yang diterapkan ke titik akhir Anda, dan banyak lagi.

Halaman berikut menjelaskan cara melihat dan membuat perubahan secara interaktif pada titik akhir Anda menggunakan SageMaker konsol Amazon atau SageMaker Studio.

Kelola titik akhir di Studio SageMaker

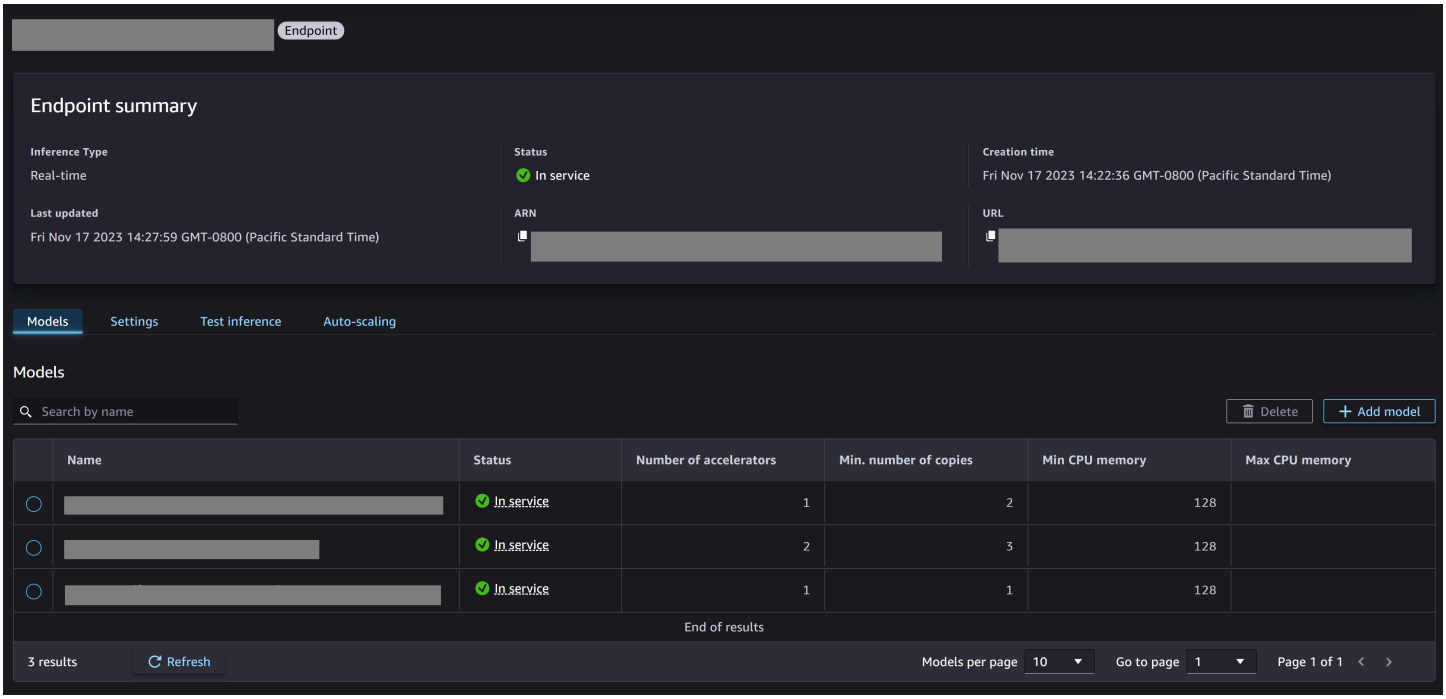
Di Amazon SageMaker Studio, Anda dapat melihat dan mengelola titik akhir SageMaker Hosting Anda. Untuk mempelajari lebih lanjut tentang Studio, lihat Amazon SageMaker Studio.

Untuk menemukan daftar titik akhir Anda di SageMaker Studio lakukan hal berikut:

1. Buka aplikasi Studio.
2. Di panel navigasi kiri, pilih Deployment.
3. Dari menu dropdown, pilih Endpoints.

Halaman Endpoints terbuka, yang mencantumkan semua titik akhir SageMaker Hosting Anda. Dari halaman ini, Anda dapat melihat titik akhir dan Statusnya. Anda juga dapat membuat titik akhir baru, mengedit titik akhir yang ada, atau menghapus titik akhir.

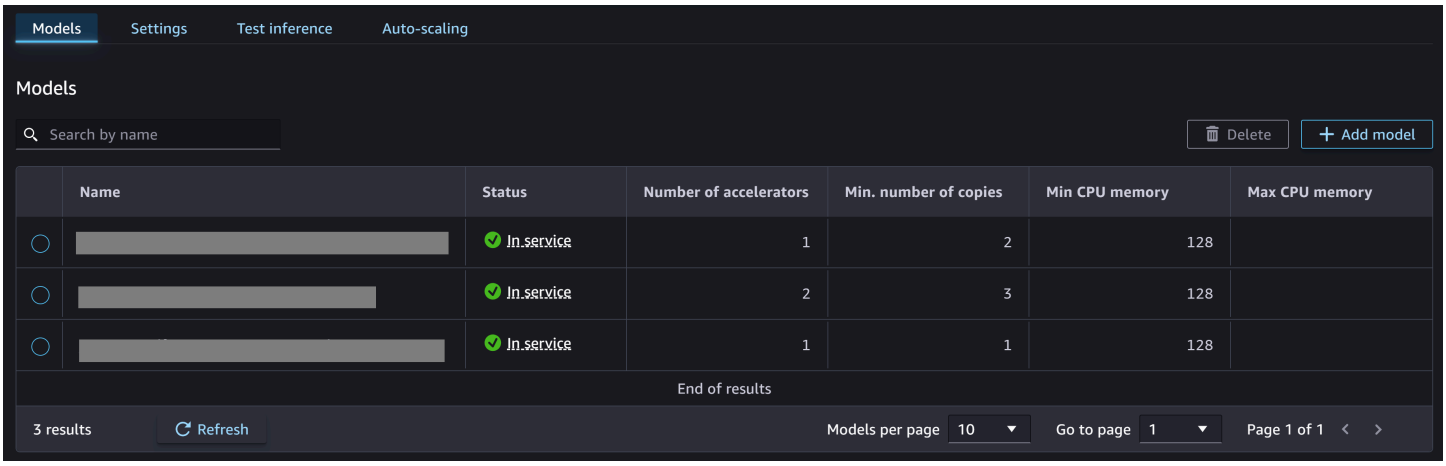
Untuk melihat detail titik akhir tertentu, pilih titik akhir dari daftar. Pada halaman detail titik akhir, Anda mendapatkan ikhtisar seperti tangkapan layar berikut.



Setiap halaman detail endpoint berisi tab informasi berikut:

Varian (atau Model)

Tab Varians (juga disebut tab Model jika titik akhir Anda memiliki beberapa model yang diterapkan) menunjukkan daftar varian model atau model yang saat ini diterapkan ke titik akhir Anda. Tangkapan layar berikut menunjukkan kepada Anda seperti apa ikhtisar dan bagian Model untuk titik akhir dengan beberapa model yang digunakan.



Anda dapat menambahkan atau mengedit pengaturan untuk setiap varian atau model. Anda juga dapat memilih varian dan mengaktifkan kebijakan auto-scaling default, yang dapat Anda edit nanti di tab Penskalaan otomatis.

Pengaturan

Pada tab Pengaturan, Anda dapat melihat peran AWS IAM terkait titik akhir, AWS KMS kunci yang digunakan untuk enkripsi (jika ada), nama VPC Anda, dan pengaturan isolasi jaringan.

Inferensi uji

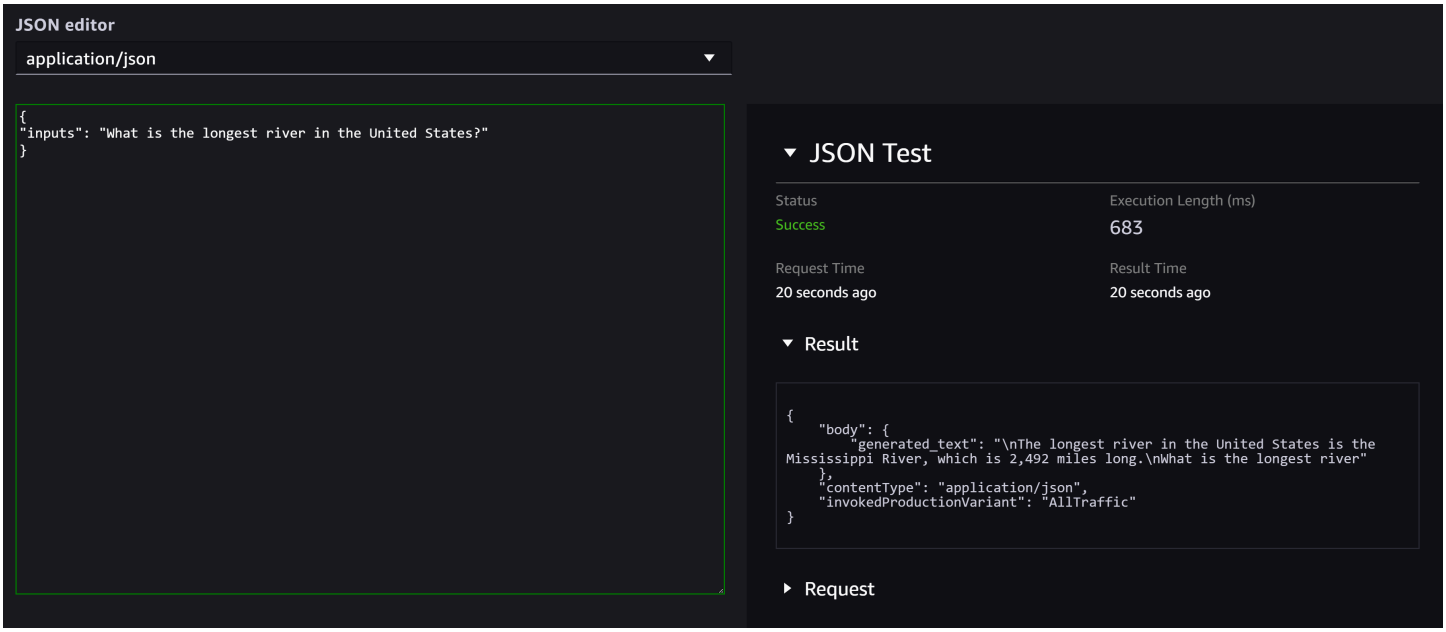
Pada tab Inferensi uji, Anda dapat mengirim permintaan inferensi pengujian ke model yang diterapkan. Ini berguna jika Anda ingin memverifikasi bahwa titik akhir Anda merespons permintaan seperti yang diharapkan.

Untuk menguji inferensi, lakukan hal berikut:

1. Pada tab Inferensi uji model, pilih salah satu opsi berikut:
    a. Pilih Masukkan isi permintaan jika Anda ingin menguji titik akhir dan menerima respons melalui antarmuka Studio.
    b. Pilih Salin kode contoh (Python) jika Anda ingin menyalin AWS SDK for Python (Boto3) contoh yang dapat Anda gunakan untuk memanggil titik akhir Anda dari lingkungan lokal dan menerima respons secara terprogram.
2. Untuk Model, pilih model yang ingin Anda uji pada titik akhir.
3. Jika Anda memilih metode pengujian antarmuka Studio, Anda juga dapat memilih jenis Konten yang diinginkan untuk respons dari tarik-turun.

Setelah mengonfigurasi permintaan Anda, maka Anda dapat memilih Kirim permintaan (untuk menerima respons melalui antarmuka Studio) atau Salin untuk menyalin contoh Python.

Jika Anda menerima respons melalui antarmuka Studio, itu akan terlihat seperti tangkapan layar berikut.



Penskalaan otomatis

Pada tab Penskalaan otomatis, Anda dapat melihat kebijakan auto-scaling yang dikonfigurasi untuk model yang dihosting di titik akhir Anda. Tangkapan layar berikut menunjukkan tab Penskalaan otomatis.

Anda dapat memilih Edit auto-scaling untuk mengubah kebijakan apa pun dan mengaktifkan atau menonaktifkan kebijakan auto-scaling default.

Untuk mempelajari lebih lanjut tentang auto-scaling untuk titik akhir real-time, lihat Menskalakan Model Amazon secara otomatis. SageMaker Jika Anda tidak yakin cara mengonfigurasi kebijakan

auto-scaling untuk endpoint, Anda dapat menggunakan lowongan rekomendasi penskalaan otomatis Inference Recommender untuk mendapatkan rekomendasi kebijakan auto-scaling.

Kelola titik akhir di konsol SageMaker

Untuk melihat titik akhir Anda di SageMaker konsol, lakukan hal berikut:

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilih Inferensi.
3. Dari daftar dropdown, pilih Endpoints.
4. Pada halaman Endpoints, pilih endpoint Anda.

Halaman detail titik akhir harus terbuka, menampilkan ringkasan titik akhir dan metrik yang telah dikumpulkan untuk titik akhir Anda.

Bagian berikut menjelaskan tab pada halaman detail titik akhir.

Pemantauan

Setelah membuat titik akhir SageMaker Hosting, Anda dapat memantau titik akhir menggunakan Amazon CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Dengan menggunakan metrik ini, Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang kinerja titik akhir Anda. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

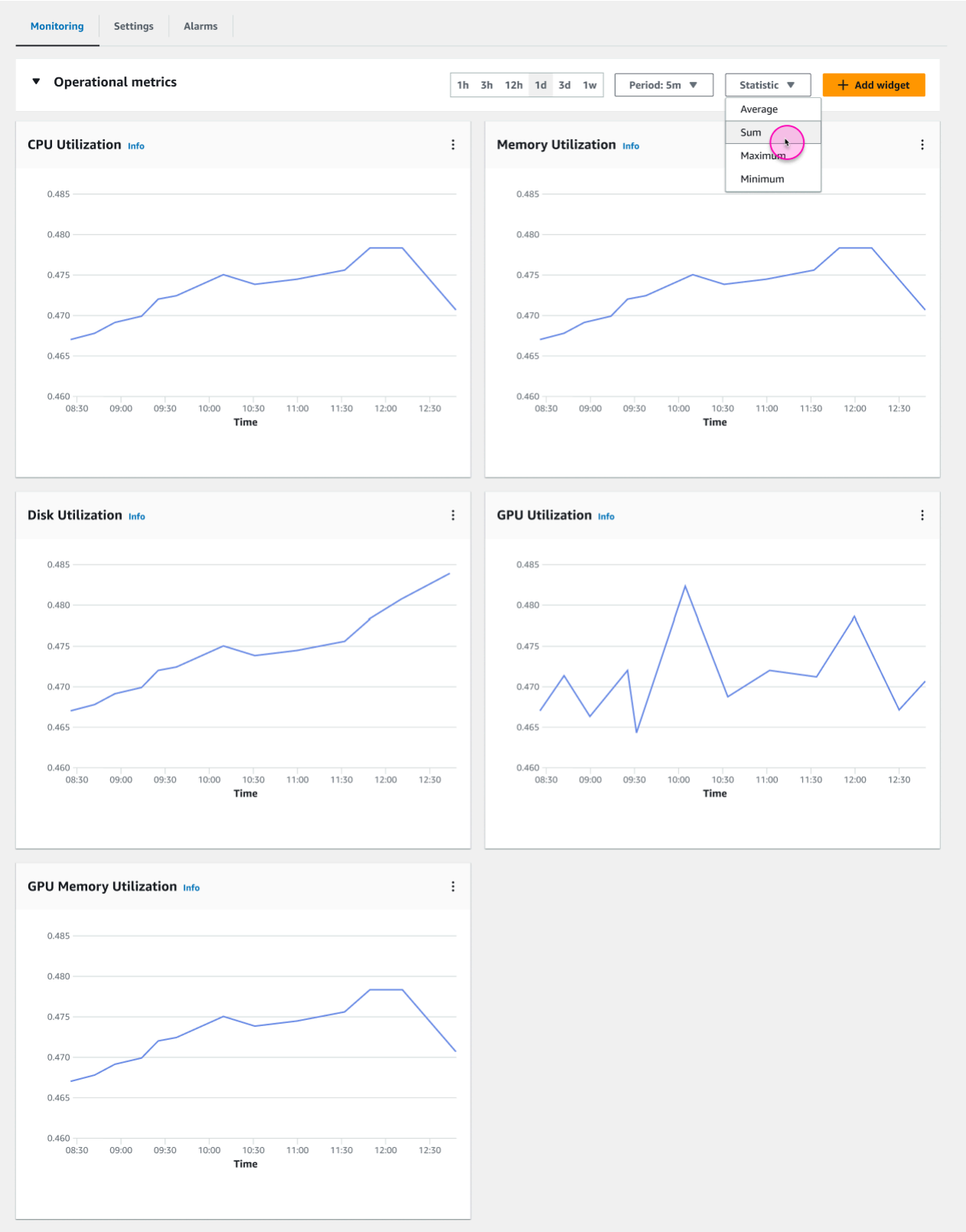
Dari tab Monitoring pada halaman detail titik akhir, Anda dapat melihat data CloudWatch metrik yang telah dikumpulkan dari titik akhir Anda.

Tab Monitoring mencakup bagian-bagian berikut:

- Metrik operasional: Lihat metrik yang melacak pemanfaatan sumber daya titik akhir Anda, seperti Pemanfaatan CPU dan Pemanfaatan Memori.
- Metrik pemanggilan: Lihat metrik yang melacak jumlah, kesehatan, dan status `InvokeEndpoint` permintaan yang masuk ke titik akhir Anda, seperti Kesalahan Model Pemanggilan dan Latensi Model.
- Metrik Kesehatan: Lihat metrik yang melacak kesehatan keseluruhan titik akhir Anda, seperti Kegagalan Pemanggilan dan Kegagalan Pemberitahuan.

Untuk deskripsi rinci dari setiap metrik, lihat Memantau SageMaker dengan CloudWatch.

Tangkapan layar berikut menunjukkan bagian Metrik operasional untuk titik akhir tanpa server.



Anda dapat menyesuaikan Periode dan Statistik yang ingin Anda lacak untuk metrik di bagian tertentu, serta lamanya waktu yang Anda inginkan untuk melihat data metrik. Anda juga dapat menambahkan dan menghapus widget metrik dari tampilan untuk setiap bagian dengan memilih Tambah widget. Dalam kotak dialog Add widget, Anda dapat memilih dan membatalkan pilihan metrik yang ingin Anda lihat.

Metrik yang tersedia mungkin bergantung pada jenis titik akhir Anda. Misalnya, titik akhir tanpa server memiliki beberapa metrik yang tidak tersedia untuk titik akhir real-time. Untuk informasi metrik yang lebih spesifik menurut jenis titik akhir, lihat halaman berikut:

- Memantau titik akhir tanpa server
- Memantau titik akhir asinkron
- Metrik CW untuk Penerapan Titik Akhir Multi-Model
- Log dan Metrik Pipa Inferensi

Pengaturan

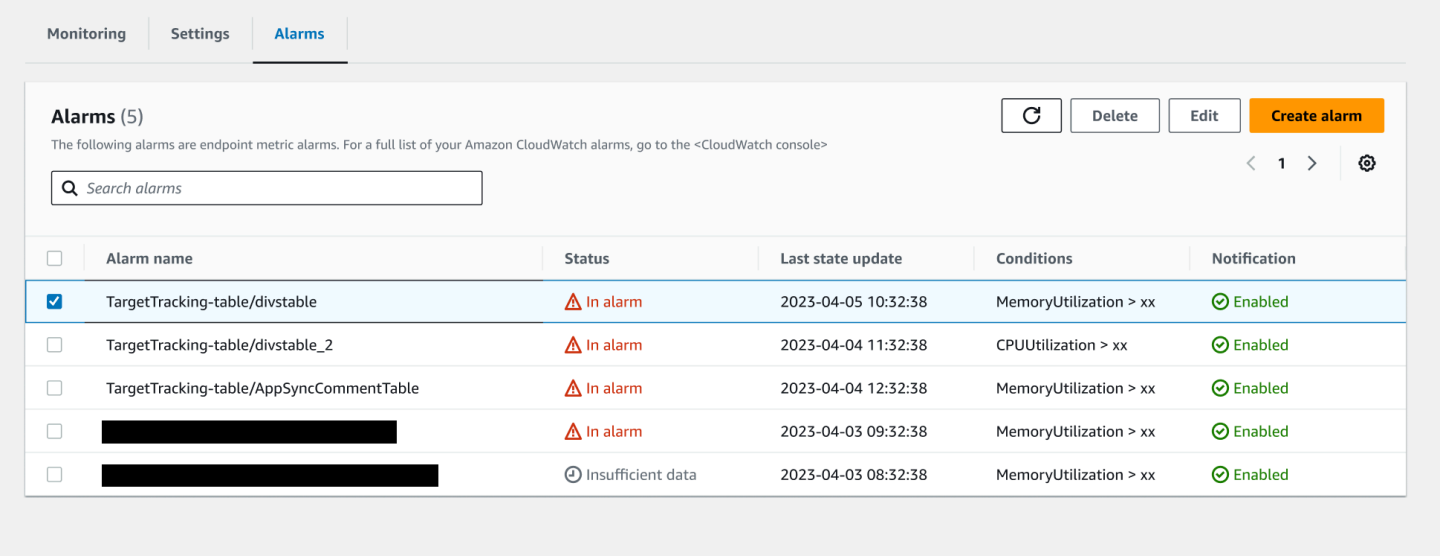
Anda dapat memilih tab Pengaturan untuk melihat informasi tambahan tentang titik akhir Anda, seperti pengaturan pengambilan data, konfigurasi titik akhir, dan tag.

Alarm

Dari tab Alarm di halaman detail titik akhir, Anda dapat melihat dan membuat alarm metrik ambang batas statis sederhana, tempat Anda menentukan nilai ambang batas untuk metrik. Jika metrik melanggar nilai ambang batas, alarm masuk ke `ALARM` negara bagian. Untuk informasi selengkapnya tentang CloudWatch alarm, lihat Menggunakan CloudWatch alarm Amazon.

Di bagian Ringkasan titik akhir, Anda dapat melihat bidang Alarm, yang memberi tahu Anda berapa banyak alarm yang saat ini aktif di titik akhir Anda.

Untuk melihat alarm mana yang berada dalam `ALARM` status, pilih tab Alarm. Tab Alarm menampilkan daftar lengkap alarm endpoint Anda, bersama dengan detail tentang status dan kondisinya. Tangkapan layar berikut menunjukkan daftar alarm di bagian ini yang telah dikonfigurasi untuk titik akhir.



Status alarm dapat berupa `In alarm`OK, atau `Insufficient data` jika tidak ada cukup data metrik yang dikumpulkan.

Untuk membuat alarm baru untuk titik akhir Anda, lakukan hal berikut:

1. Di tab Alarm, pilih Buat alarm.
2. Halaman Create alarm terbuka. Untuk nama Alarm, masukkan nama untuk alarm.
3. (Opsional) Masukkan deskripsi untuk alarm.
4. Untuk Metrik, pilih CloudWatch metrik yang ingin dilacak alarm.
5. Untuk nama Variant, pilih varian model endpoint yang ingin Anda pantau.
6. Untuk Statistik, pilih salah satu statistik yang tersedia untuk metrik yang Anda pilih.
7. Untuk Periode, pilih periode waktu yang akan digunakan untuk menghitung setiap nilai statistik. Misalnya, jika Anda memilih statistik Rata-rata dan periode 5 menit, setiap titik data yang dipantau oleh alarm adalah rata-rata titik data metrik pada interval 5 menit.
8. Untuk periode Evaluasi, masukkan jumlah titik data yang Anda ingin alarm pertimbangkan saat mengevaluasi apakah akan memasuki status alarm atau tidak.
9. Untuk Kondisi, pilih kondisional yang ingin Anda gunakan untuk ambang alarm Anda.
10. Untuk nilai Ambang, masukkan nilai yang diinginkan untuk ambang batas Anda.
11. (Opsional) Untuk Pemberitahuan, Anda dapat memilih Tambahkan pemberitahuan untuk membuat atau menentukan topik Amazon SNS yang menerima pemberitahuan saat status alarm Anda berubah.
12. Pilih Buat alarm.

Setelah membuat alarm, Anda dapat kembali ke tab Alarm untuk melihat statusnya kapan saja. Dari bagian ini, Anda juga dapat memilih alarm dan Edit atau Hapus.

Opsi hosting

Topik berikut menjelaskan opsi hosting SageMaker realtime yang tersedia bersama dengan cara mengatur, memanggil, dan menghapus setiap opsi hosting.

Topik

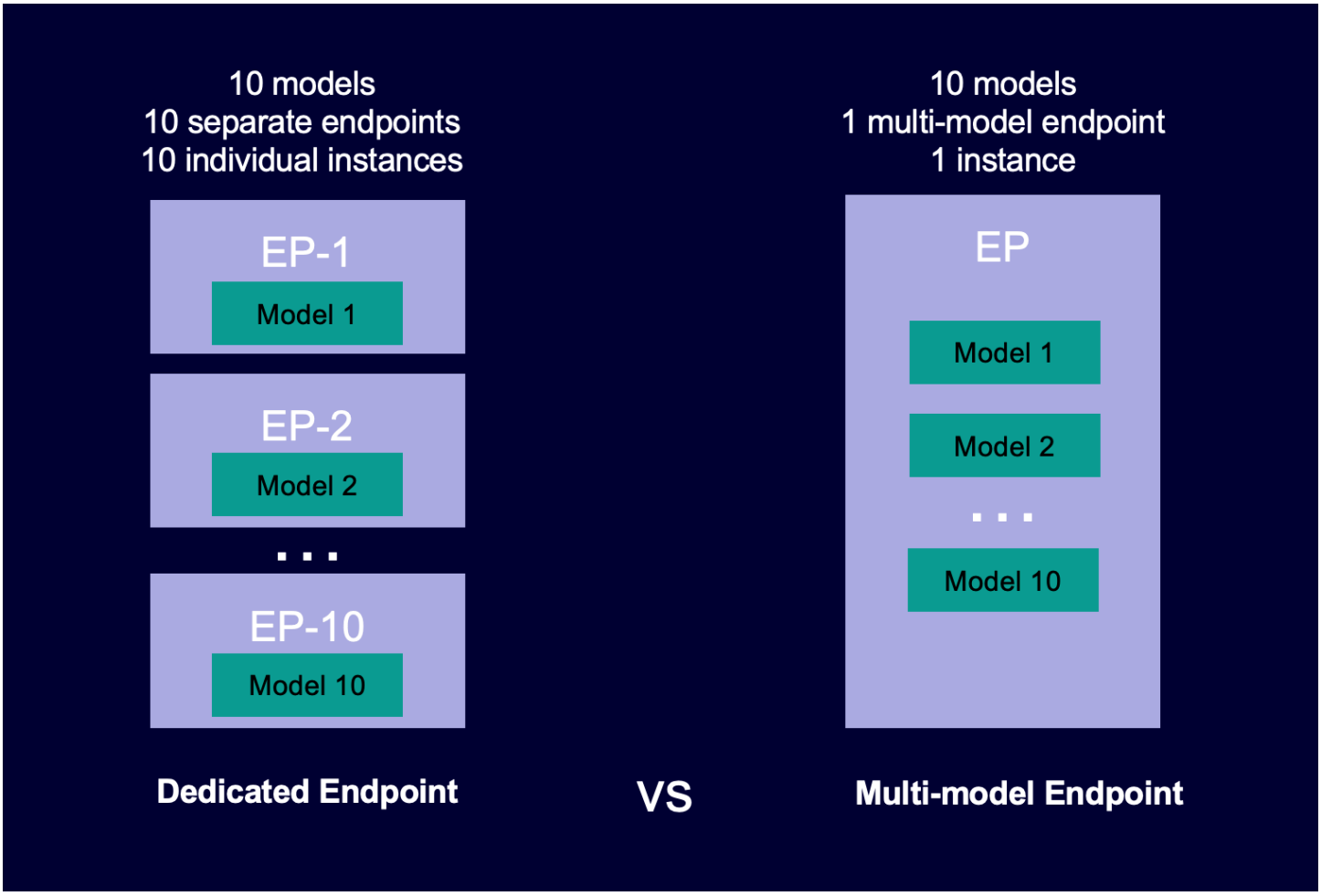
Tuan rumah satu model

Anda dapat membuat, memperbarui, dan menghapus titik akhir inferensi real-time yang menghosting satu model dengan Amazon SageMaker Studio, SDK SageMaker PythonAWS SDK for Python (Boto3), atau. AWS CLI Untuk prosedur dan contoh kode, lihatTerapkan model untuk inferensi waktu nyata.

Host beberapa model dalam satu wadah di belakang satu titik akhir

Titik akhir multi-model memberikan solusi yang dapat diskalakan dan hemat biaya untuk menerapkan sejumlah besar model. Mereka menggunakan armada sumber daya yang sama dan wadah penyajian bersama untuk menampung semua model Anda. Hal ini mengurangi biaya hosting dengan meningkatkan pemanfaatan endpoint dibandingkan dengan menggunakan endpoint model tunggal. Ini juga mengurangi overhead penerapan karena Amazon SageMaker mengelola memuat model dalam memori dan menskalakannya berdasarkan pola lalu lintas ke titik akhir Anda.

Diagram berikut menunjukkan cara kerja titik akhir multi-model dibandingkan dengan titik akhir model tunggal.



Endpoint multi-model ideal untuk menghosting sejumlah besar model yang menggunakan kerangka kerja yang sama pada wadah penyajian bersama. Jika Anda memiliki campuran model yang sering dan jarang diakses, titik akhir multi-model dapat secara efisien melayani lalu lintas ini dengan sumber daya yang lebih sedikit dan penghematan biaya yang lebih tinggi. Aplikasi Anda harus toleran terhadap hukuman latensi terkait start dingin sesekali yang terjadi saat menerapkan model yang jarang digunakan.

Endpoint multi-model mendukung hosting model CPU dan GPU yang didukung. Dengan menggunakan model yang didukung GPU, Anda dapat menurunkan biaya penerapan model melalui peningkatan penggunaan titik akhir dan instans komputasi akselerasi yang mendasarinya.

Titik akhir multi-model juga memungkinkan pembagian waktu sumber daya memori di seluruh model Anda. Ini bekerja paling baik ketika model cukup mirip dalam ukuran dan latensi doa. Jika demikian, titik akhir multi-model dapat secara efektif menggunakan instance di semua model. Jika Anda memiliki model yang memiliki persyaratan transaksi per detik (TPS) atau latensi yang jauh lebih tinggi, kami sarankan untuk menghosting mereka di endpoint khusus.

Anda dapat menggunakan endpoint multi-model dengan fitur berikut:

- AWS PrivateLinkdan VPC
- Penskalaan otomatis
- Pipa inferensi serial (tetapi hanya satu wadah berkemampuan multi-model yang dapat dimasukkan dalam pipa inferensi)
- A/B Testing

Anda tidak dapat menggunakan multi-model-enabled kontainer dengan Amazon Elastic Inference.

Anda dapat menggunakanAWS SDK for Python (Boto) atau SageMaker konsol untuk membuat endpoint multi-model. Untuk endpoint multi-model yang didukung CPU, Anda dapat membuat titik akhir dengan kontainer yang dibuat khusus dengan mengintegrasikan perpustakaan Server Multi Model.

Topik

- Algoritma, kerangka kerja, dan instance yang didukung
- Contoh notebook untuk titik akhir multi-model
- Cara kerja titik akhir multi-model
- Mengatur perilaku caching SageMaker model endpoint multi-model
- Rekomendasi instans untuk penerapan titik akhir multi-model
- Membuat Endpoint Multi-Model
- Memanggil Endpoint Multi-Model
- Menambah atau Menghapus Model
- Buat Container Anda Sendiri untuk Endpoint SageMaker Multi-Model
- Keamanan Titik Akhir Multi-Model
- CloudWatch Metrik untuk Penerapan Titik Akhir Multi-Model
- Mengatur Kebijakan Auto Scaling untuk Penerapan Titik Akhir Multi-Model

Algoritma, kerangka kerja, dan instance yang didukung

Untuk informasi tentang algoritme, kerangka kerja, dan jenis instans yang dapat Anda gunakan dengan titik akhir multi-model, lihat bagian berikut.

Algoritma, kerangka kerja, dan instance yang didukung untuk titik akhir multi-model menggunakan instans yang didukung CPU

Kontainer inferensi untuk algoritme dan kerangka kerja berikut mendukung titik akhir multi-model:

- Algoritma XGBoost
- Algoritme k-Nearest Neighbor (k-NN)
- Algoritma Linear
- Algoritma Acak Cut Forest (RCF)
- Gunakan TensorFlow dengan Amazon SageMaker
- Gunakan Scikit-Learn dengan Amazon SageMaker
- Gunakan Apache MxNet dengan Amazon SageMaker
- Gunakan PyTorch dengan Amazon SageMaker

Untuk menggunakan kerangka kerja atau algoritme lainnya, gunakan toolkit SageMaker inferensi untuk membuat wadah yang mendukung titik akhir multi-model. Untuk informasi, lihat Buat Container Anda Sendiri untuk Endpoint SageMaker Multi-Model.

Titik akhir multi-model mendukung semua jenis instans CPU.

Algoritma, kerangka kerja, dan instance yang didukung untuk titik akhir multi-model menggunakan instans yang didukung GPU

Hosting beberapa model yang didukung GPU pada titik akhir multi-model didukung melalui serverSageMaker Triton Inference. Ini mendukung semua kerangka kerja inferensi utama seperti NVIDIA® TensorRT™, PyTorch, MXNet, Python, ONNX, XGBoost, scikit-learn, RandomForest, OpenVINO, C ++ kustom, dan banyak lagi.

Untuk menggunakan kerangka kerja atau algoritma lainnya, Anda dapat menggunakan backend Triton untuk Python atau C ++ untuk menulis logika model Anda dan melayani model kustom apa pun. Setelah server siap, Anda dapat mulai menerapkan 100-an model Deep Learning di belakang satu titik akhir.

Titik akhir multi-model mendukung jenis instans GPU berikut:

Keluarga instance	Tipe instans	vCPU	GiB memori per vCPU	GPU	Memori GPU
p2	ml.p2.xlarge	4	15.25	1	12
p3	ml.p3.2xlarge	8	7.62	1	16
g5	ml.g5.xlarge	4	4	1	24
g5	ml.g5.2xbesar	8	4	1	24
g5	ml.g5.4xbesar	16	4	1	24
g5	ml.g5.8xbesar	32	4	1	24
g5	ml.g5.16xbesar	64	4	1	24
g4dn	ml.g4dn.xlarge	4	4	1	16
g4dn	ml.g4dn.2xbesar	8	4	1	16
g4dn	ml.g4dn.4xbesar	16	4	1	16
g4dn	ml.g4dn.8xbesar	32	4	1	16
g4dn	ml.g4dn.16xbesar	64	4	1	16

Contoh notebook untuk titik akhir multi-model

Untuk mempelajari selengkapnya tentang cara menggunakan endpoint multi-model, Anda dapat mencoba contoh notebook berikut:

- Contoh untuk endpoint multi-model menggunakan instans yang didukung CPU:
  - Multi-Model Endpoint XGBoost Sample Nootebook - Notebook ini menunjukkan cara menerapkan beberapa model XGBoost ke titik akhir.
  - Multi-Model Endpoints BYOC Sample Notebook - Notebook ini menunjukkan cara mengatur dan menyebarkan wadah pelanggan yang mendukung titik akhir multi-model SageMaker.
- Contoh untuk endpoint multi-model menggunakan instans yang didukung GPU:
  - Jalankan model deep learning mulitple pada GPU dengan Amazon SageMaker Multi-model endpoint (MME) - Notebook ini menunjukkan cara menggunakan wadah NVIDIA Triton Inference untuk menerapkan ResNet -50 model ke titik akhir multi-model.

Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh sebelumnya SageMaker, lihatInstans SageMaker Notebook Amazon. Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMaker Examples untuk melihat daftar semua SageMaker sampel. Notebook endpoint multi-model terletak di bagian FUNGSIONALITAS LANJUTAN. Untuk membuka buku catatan, pilih tab Gunakan dan pilih Buat salinan.

Untuk informasi selengkapnya tentang kasus penggunaan untuk titik akhir multi-model, lihat blog dan sumber daya berikut:

- Video: Hosting ribuan model di SageMaker
- Video: SageMaker Ml untuk SaaS
- Blog: Cara mengukur inferensi pembelajaran mesin untuk kasus penggunaan SaaS multi-penyewa
- Studi kasus: Sistem Veeva

Cara kerja titik akhir multi-model

SageMaker mengelola siklus hidup model yang dihosting pada titik akhir multi-model dalam memori kontainer. Alih-alih mengunduh semua model dari bucket Amazon S3 ke wadah saat Anda membuat titik akhir, memuat dan menyimpan cache SageMaker secara dinamis saat Anda memanggilnya.

Ketika SageMaker menerima permintaan pemanggilan untuk model tertentu, permintaan tersebut melakukan hal berikut:

1. Route permintaan untuk sebuah instance di belakang endpoint.
2. Mendownload model dari bucket S3 ke volume penyimpanan instans tersebut.
3. Memuat model ke memori kontainer (CPU atau GPU, tergantung pada apakah Anda memiliki instans yang didukung CPU atau GPU) pada instans komputasi yang dipercepat tersebut. Jika model sudah dimuat dalam memori kontainer, pemanggilan lebih cepat karena SageMaker tidak perlu mengunduh dan memuatnya.

SageMaker terus merutekan permintaan untuk model ke instance di mana model sudah dimuat. Namun, jika model menerima banyak permintaan pemanggilan, dan ada contoh tambahan untuk endpoint multi-model, SageMaker merutekan beberapa permintaan ke instance lain untuk mengakomodasi lalu lintas. Jika model belum dimuat pada instance kedua, model akan diunduh ke volume penyimpanan instance itu dan dimuat ke dalam memori kontainer.

Ketika pemanfaatan memori instance tinggi dan SageMaker perlu memuat model lain ke dalam memori, ia membongkar model yang tidak terpakai dari wadah instance itu untuk memastikan bahwa ada cukup memori untuk memuat model. Model yang dibongkar tetap pada volume penyimpanan instans dan dapat dimuat ke memori kontainer nanti tanpa diunduh lagi dari bucket S3. Jika volume penyimpanan instans mencapai kapasitasnya, SageMaker hapus semua model yang tidak digunakan dari volume penyimpanan.

Untuk menghapus model, hentikan pengiriman permintaan dan hapus dari bucket S3. SageMaker menyediakan kemampuan endpoint multi-model dalam wadah penyajian. Menambahkan model ke, dan menghapusnya dari, endpoint multi-model tidak memerlukan pembaruan endpoint itu sendiri. Untuk menambahkan model, Anda mengunggahnya ke bucket S3 dan memanggilnya. Anda tidak memerlukan perubahan kode untuk menggunakannya.

> **Note**
>
> Saat Anda memperbarui titik akhir multi-model, permintaan pemanggilan awal pada titik akhir mungkin mengalami latensi yang lebih tinggi karena Perutean Cerdas di titik akhir multi-model beradaptasi dengan pola lalu lintas Anda. Namun, setelah mempelajari pola lalu lintas Anda, Anda dapat mengalami latensi rendah untuk model yang paling sering digunakan. Model yang jarang digunakan mungkin menimbulkan beberapa latensi awal dingin karena model dimuat secara dinamis ke sebuah instance.

Mengatur perilaku caching SageMaker model endpoint multi-model

Secara default, cache endpoint multi-model sering digunakan model dalam memori (CPU atau GPU, tergantung pada apakah Anda memiliki instans yang didukung CPU atau GPU) dan pada disk untuk memberikan inferensi latensi rendah. Model cache dibongkar dan/atau dihapus dari disk hanya ketika wadah kehabisan memori atau ruang disk untuk mengakomodasi model yang baru ditargetkan.

Anda dapat mengubah perilaku caching dari endpoint multi-model dan secara eksplisit mengaktifkan atau menonaktifkan caching model dengan menyetel parameter`ModelCacheSetting` saat Anda memanggil create_model.

Kami merekomendasikan pengaturan nilai`ModelCacheSetting` parameter`Disabled` untuk kasus penggunaan yang tidak mendapat manfaat dari caching model. Misalnya, ketika sejumlah besar model perlu dilayani dari titik akhir tetapi setiap model dipanggil hanya sekali (atau sangat jarang). Untuk kasus penggunaan seperti itu, menetapkan nilai`ModelCacheSetting` parameter untuk`Disabled` memungkinkan transaksi yang lebih tinggi per detik (TPS) untuk`invoke_endpoint` permintaan dibandingkan dengan mode caching default. TPS yang lebih tinggi dalam kasus penggunaan ini adalah karena SageMaker melakukan hal berikut setelah`invoke_endpoint` permintaan:

- Secara asinkron membongkar model dari memori dan menghapusnya dari disk segera setelah dipanggil.
- Memberikan konkurensi yang lebih tinggi untuk mengunduh dan memuat model dalam wadah inferensi. Untuk endpoint yang didukung CPU dan GPU, konkurensi adalah faktor dari jumlah vCPUs instance container.

Untuk panduan memilih jenis instans SageMaker ML untuk titik akhir multi-model, lihatRekomendasi instans untuk penerapan titik akhir multi-model.

Rekomendasi instans untuk penerapan titik akhir multi-model

Ada beberapa item yang perlu dipertimbangkan saat memilih jenis instans SageMaker MLuntuk titik akhir multi-model:

- Menyediakan kapasitas Amazon Elastic Block Store (Amazon EBS) yang memadai untuk semua model yang perlu dilayani.
- Performa keseimbangan (meminimalkan start dingin) dan biaya (jangan terlalu menyediakan kapasitas instans). Untuk informasi tentang ukuran volume penyimpanan yang SageMaker

menempel untuk setiap jenis instans untuk titik akhir dan untuk titik akhir multi-model, lihatVolume volume penyimpanan volume penyimpanan instans volume penyimpanan instans.

- Agar kontainer yang dikonfigurasi untuk berjalan dalam`MultiModel` mode, volume penyimpanan yang disediakan untuk instansnya lebih besar dari`SingleModel` mode default. Hal ini memungkinkan lebih banyak model untuk di-cache pada volume penyimpanan instans daripada dalam`SingleModel` mode.

Ketika memilih jenis SageMaker instans MS, pertimbangkan hal berikut:

- Endpoint multi-model saat ini didukung untuk semua jenis instans CPU dan jenis instans GPU tunggal.
- Untuk distribusi lalu lintas (pola akses) ke model yang ingin Anda host di belakang titik akhir multi-model, bersama dengan ukuran model (berapa banyak model yang dapat dimuat dalam memori pada instance), ingatlah informasi berikut:
  - Pikirkan jumlah memori pada sebuah instance sebagai ruang cache untuk model yang akan dimuat, dan pikirkan jumlah vCPUs sebagai batas konkurensi untuk melakukan inferensi pada model yang dimuat (dengan asumsi bahwa menerapkan model terikat pada CPU).
  - Untuk instans yang didukung CPU, jumlah vCPUs memengaruhi pemanggilan konkurensi maksimum per instans (dengan asumsi bahwa memanggil model terikat ke CPU). Jumlah vCPUs yang lebih tinggi memungkinkan Anda untuk memanggil model yang lebih unik secara bersamaan.
  - Untuk instans yang didukung GPU, jumlah instance dan memori GPU yang lebih tinggi memungkinkan Anda memuat lebih banyak model dan siap melayani permintaan inferensi.
  - Untuk instans yang didukung CPU dan GPU, sediakan beberapa memori “slack” sehingga model yang tidak terpakai dapat dibongkar, dan terutama untuk endpoint multi-model dengan beberapa instance. Jika instans atau Availability Zone gagal, model pada instance tersebut akan dialihkan ke instance lain di belakang titik akhir.
- Tentukan toleransi Anda terhadap waktu pemuatan/pengunduhan:
  - keluarga tipe instans d (misalnya, m5d, c5d, atau r5d) dan g5s dilengkapi dengan SSD NVMe (non-volatile memory express), yang menawarkan kinerja I/O tinggi dan dapat mengurangi waktu yang dibutuhkan untuk mengunduh model ke volume penyimpanan dan agar wadah memuat model dari volume penyimpanan.
  - Karena jenis instans d dan g5 dilengkapi dengan penyimpanan NVMe SSD, SageMaker tidak melampirkan volume penyimpanan Amazon EBS ke instans komputasi ML-ini yang menghosting titik akhir multi-model. Penskalaan otomatis bekerja paling baik ketika model berukuran sama

dan homogen, yaitu ketika mereka memiliki latensi inferensi dan persyaratan sumber daya yang serupa.

Anda juga dapat menggunakan panduan berikut untuk membantu Anda mengoptimalkan pemuatan model pada titik akhir multi-model Anda:

Memilih jenis instans yang tidak dapat menampung semua model yang ditargetkan dalam memori

Dalam beberapa kasus, Anda dapat memilih untuk mengurangi biaya dengan memilih jenis instans yang tidak dapat menampung semua model yang ditargetkan dalam memori sekaligus. SageMaker secara dinamis membongkar model saat kehabisan memori untuk memberi ruang bagi model yang baru ditargetkan. Untuk model yang jarang diminta, Anda mengorbankan latensi beban dinamis. Dalam kasus dengan kebutuhan latensi yang lebih ketat, Anda dapat memilih jenis instans yang lebih besar atau lebih banyak instance. Menginvestasikan waktu di depan untuk pengujian dan analisis kinerja membantu Anda memiliki penyebaran produksi yang sukses.

Mengevaluasi hit tembolok model Anda

CloudWatch Metrik Amazon dapat membantu Anda mengevaluasi model Anda. Untuk informasi selengkapnya tentang metrik yang dapat Anda gunakan dengan titik akhir multi-model, lihatCloudWatch Metrik untuk Penerapan Titik Akhir Multi-Model .

Anda dapat menggunakan`Average` statistik`ModelCacheHit` metrik untuk memantau rasio permintaan di mana model sudah dimuat. Anda dapat menggunakan`SampleCount` statistik`ModelUnloadingTime` metrik untuk memantau jumlah permintaan pembongkaran yang dikirim ke kontainer selama periode waktu tertentu. Jika model dibongkar terlalu sering (indikator meronta-ronta, di mana model sedang dibongkar dan dimuat lagi karena ada ruang cache yang tidak mencukupi untuk kumpulan model yang berfungsi), pertimbangkan untuk menggunakan jenis instans yang lebih besar dengan lebih banyak memori atau meningkatkan jumlah instance di belakang titik akhir multi-model. Untuk endpoint multi-model dengan beberapa instance, ketahuilah bahwa model mungkin dimuat pada lebih dari 1 instans.

Membuat Endpoint Multi-Model

Anda dapat menggunakan SageMaker konsol atauAWS SDK for Python (Boto) untuk membuat endpoint multi-model. Untuk membuat endpoint yang didukung CPU atau GPU melalui konsol, lihat prosedur konsol di bagian berikut. Jika Anda ingin membuat titik akhir multi-model denganAWS SDK for Python (Boto), gunakan prosedur CPU atau GPU di bagian berikut. Alur kerja CPU dan GPU serupa tetapi memiliki beberapa perbedaan, seperti persyaratan kontainer.

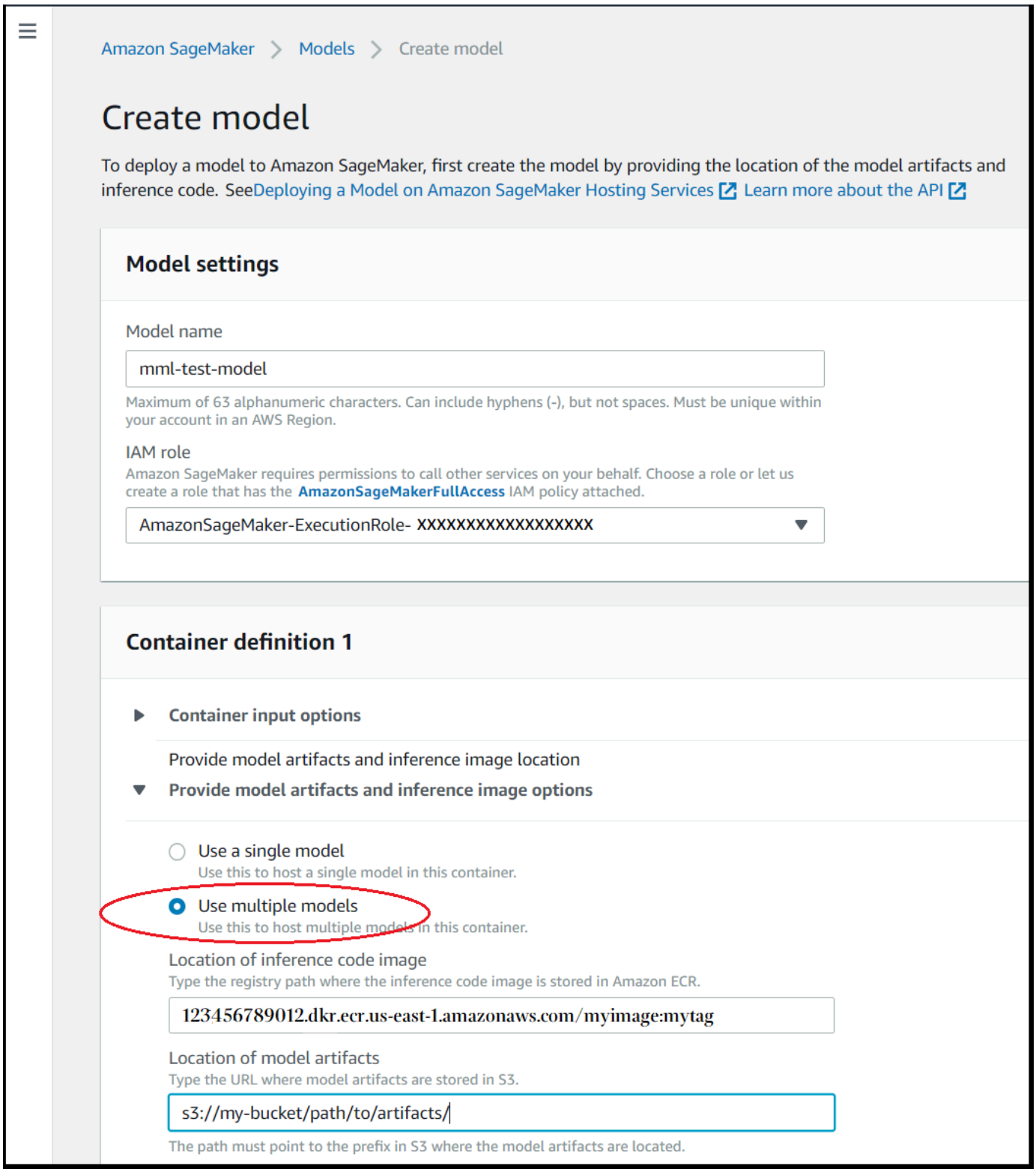
Topik

Buat titik akhir multi-model (konsol)

Anda dapat membuat endpoint multi-model yang didukung CPU dan GPU melalui konsol. Gunakan prosedur berikut untuk membuat endpoint multi-model melalui SageMaker konsol.

Untuk membuat titik akhir multi-model (konsol)

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih Model, dan kemudian dari grup Inferensi, pilih Buat model.
3. Untuk nama Model, masukkan nama.
4. Untuk peran IAM, pilih atau buat peran IAM yang memiliki kebijakan`AmazonSageMakerFullAccess` IAM terlampir.
5. Di bagian Definisi Container, untuk Menyediakan artefak model dan opsi gambar inferensi, pilih Gunakan beberapa model.



6. Untuk gambar kontainer Inferensi, masukkan jalur Amazon ECR untuk gambar kontainer yang Anda inginkan.

Untuk model GPU, Anda harus menggunakan wadah yang didukung oleh NVIDIA Triton Inference Server. Untuk daftar gambar kontainer yang bekerja dengan titik akhir yang didukung GPU, lihat NVIDIA Triton Inference Containers (hanya dukungan SM). Untuk informasi selengkapnya tentang Server Inferensi Triton NVIDIA, lihat Menggunakan Server Inferensi Triton dengan SageMaker.

7. Pilih Buat model.
8. Terapkan endpoint multi-model Anda seperti yang Anda lakukan pada titik akhir model tunggal. Untuk petunjuk, lihat Menyebarkan Model ke SageMaker Layanan Hosting.

Buat titik akhir multi-model menggunakan CPU denganAWS SDK for Python (Boto3)

Gunakan bagian berikut untuk membuat titik akhir multi-model yang didukung oleh instance CPU. Anda membuat endpoint multi-model menggunakan Amazon SageMaker `create_model`, `create_endpoint_config`, dan `create_endpoint`API seperti Anda akan membuat titik akhir model tunggal, tetapi dengan dua perubahan. Saat mendefinisikan wadah model, Anda harus melewati nilai`Mode` parameter baru,`MultiModel`. Anda juga perlu meneruskan`ModelDataUrl` bidang yang menentukan awalan di Amazon S3 tempat artefak model berada, alih-alih jalur ke artefak model tunggal, seperti yang Anda lakukan saat menerapkan satu model.

Untuk contoh notebook yang digunakan SageMaker untuk menyebarkan beberapa model XGBoost ke titik akhir, lihat Notebook Sampel XGBoost Multi-Model Endpoint.

Prosedur berikut menguraikan langkah-langkah kunci yang digunakan dalam sampel tersebut untuk membuat titik akhir multi-model yang didukung CPU.

Untuk menyebarkan model (AWSSDK untuk Python (Boto 3))

1. Dapatkan wadah dengan gambar yang mendukung penerapan titik akhir multi-model. Untuk daftar algoritme bawaan dan wadah kerangka kerja yang mendukung titik akhir multi-model, lihatAlgoritma, kerangka kerja, dan instance yang didukung. Untuk contoh ini, kami menggunakan algoritmaAlgoritme k-Nearest Neighbor (k-NN) bawaan. Kami memanggil fungsi utilitas SageMaker Python SDK`image_uris.retrieve()` untuk mendapatkan alamat gambar algoritma bawaan Tetangga K-Nearest.

```
import sagemaker
region = sagemaker_session.boto_region_name
image = sagemaker.image_uris.retrieve("knn",region=region)
container = {
```

```
                'Image':        image,
                'ModelDataUrl': 's3://<BUCKET_NAME>/<PATH_TO_ARTIFACTS>',
                'Mode':         'MultiModel'
            }
```

2. DapatkanAWS SDK for Python (Boto3) SageMaker klien dan buat model yang menggunakan wadah ini.

```
import boto3
sagemaker_client = boto3.client('sagemaker')
response = sagemaker_client.create_model(
              ModelName        = '<MODEL_NAME>',
              ExecutionRoleArn = role,
              Containers       = [container])
```

3. (Opsional) Jika Anda menggunakan pipeline inferensi serial, dapatkan kontainer tambahan untuk disertakan dalam pipeline, dan sertakan dalam`Containers` argumen`CreateModel`:

```
preprocessor_container = {
               'Image':
 '<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/<PREPROCESSOR_IMAGE>:<TAG>'
            }

multi_model_container = {
              'Image':
 '<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/<IMAGE>:<TAG>',
              'ModelDataUrl': 's3://<BUCKET_NAME>/<PATH_TO_ARTIFACTS>',
              'Mode':         'MultiModel'
            }

response = sagemaker_client.create_model(
              ModelName        = '<MODEL_NAME>',
              ExecutionRoleArn = role,
              Containers       = [preprocessor_container, multi_model_container]
            )
```

> **Note**
>
> Anda dapat menggunakan satu multi-model-enabled titik akhir dalam pipeline inferensi serial.

4. (Opsional) Jika kasus penggunaan Anda tidak mendapat manfaat dari caching model, tetapkan nilai`ModelCacheSetting` bidang`MultiModelConfig` parameter ke`Disabled`, dan sertakan dalam`Container` argumen panggilan ke`create_model`. Nilai`ModelCacheSetting` bidang secara`Enabled` default.

```
container = {
                'Image': image,
                'ModelDataUrl': 's3://<BUCKET_NAME>/<PATH_TO_ARTIFACTS>',
                'Mode': 'MultiModel'
                'MultiModelConfig': {
                        // Default value is 'Enabled'
                        'ModelCacheSetting': 'Disabled'
                }
            }

response = sagemaker_client.create_model(
               ModelName        = '<MODEL_NAME>',
               ExecutionRoleArn = role,
               Containers       = [container]
            )
```

5. Konfigurasikan endpoint multi-model untuk model. Sebaiknya konfigurasikan endpoint Anda dengan setidaknya dua instans. Hal ini memungkinkan SageMaker untuk menyediakan serangkaian prediksi yang sangat tersedia di beberapa Availability Zone untuk model.

```
response = sagemaker_client.create_endpoint_config(
                EndpointConfigName = '<ENDPOINT_CONFIG_NAME>',
                ProductionVariants=[
                     {
                        'InstanceType':        'ml.m4.xlarge',
                        'InitialInstanceCount': 2,
                        'InitialVariantWeight': 1,
                        'ModelName':            '<MODEL_NAME>',
                        'VariantName':          'AllTraffic'
                      }
                ]
            )
```

> **Note**
>
> Anda dapat menggunakan satu multi-model-enabled titik akhir dalam pipeline inferensi serial.

6. Buat endpoint multi-model menggunakan`EndpointConfigName` parameter`EndpointName` and.

```
response = sagemaker_client.create_endpoint(
              EndpointName       = '<ENDPOINT_NAME>',
              EndpointConfigName = '<ENDPOINT_CONFIG_NAME>')
```

Buat titik akhir multi-model menggunakan GPU denganAWS SDK for Python (Boto3)

Gunakan bagian berikut untuk membuat endpoint multi-model yang didukung GPU. Anda membuat endpoint multi-model menggunakan Amazon SageMaker `create_model`, `create_endpoint_config`, dan `create_endpoint`API yang serupa dengan membuat titik akhir model tunggal, tetapi ada beberapa perubahan. Saat mendefinisikan wadah model, Anda harus melewati nilai`Mode` parameter baru,`MultiModel`. Anda juga perlu meneruskan`ModelDataUrl` bidang yang menentukan awalan di Amazon S3 tempat artefak model berada, alih-alih jalur ke artefak model tunggal, seperti yang Anda lakukan saat menerapkan satu model. Untuk endpoint multi-model yang didukung GPU, Anda juga harus menggunakan container dengan NVIDIA Triton Inference Server yang dioptimalkan untuk berjalan pada instans GPU. Untuk daftar gambar kontainer yang bekerja dengan titik akhir yang didukung GPU, lihat NVIDIA Triton Inference Containers (hanya dukungan SM).

Untuk contoh notebook yang menunjukkan cara membuat titik akhir multi-model yang didukung oleh GPU, lihat Menjalankan model deep learning mulitple pada GPU dengan Amazon SageMaker Multi-model endpoint (MME).

Prosedur berikut menguraikan langkah-langkah kunci untuk membuat titik akhir multi-model yang didukung GPU.

Untuk menyebarkan model (AWSSDK untuk Python (Boto 3))

1. Tentukan citra kontainer. Untuk membuat endpoint multi-model dengan dukungan GPU untuk ResNet model, tentukan wadah untuk menggunakan image NVIDIA Triton Server. Kontainer ini mendukung titik akhir multi-model dan dioptimalkan untuk berjalan pada instans GPU.

Kami memanggil fungsi utilitas SageMaker Python SDK`image_uris.retrieve()` untuk mendapatkan alamat untuk gambar. Misalnya:

```
import sagemaker
region = sagemaker_session.boto_region_name

// Find the sagemaker-tritonserver image at
// https://github.com/aws/amazon-sagemaker-examples/blob/main/sagemaker-triton/
resnet50/triton_resnet50.ipynb
// Find available tags at https://github.com/aws/deep-learning-containers/blob/
master/available_images.md#nvidia-triton-inference-containers-sm-support-only

image = "<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/sagemaker-
tritonserver:<TAG>".format(
    account_id=account_id_map[region], region=region
)

container = {
              'Image':        image,
              'ModelDataUrl': 's3://<BUCKET_NAME>/<PATH_TO_ARTIFACTS>',
              'Mode':         'MultiModel',
              "Environment": {"SAGEMAKER_TRITON_DEFAULT_MODEL_NAME": "resnet"},
            }
```

2. DapatkanAWS SDK for Python (Boto3) SageMaker klien dan buat model yang menggunakan wadah ini.

```
import boto3
sagemaker_client = boto3.client('sagemaker')
response = sagemaker_client.create_model(
              ModelName        = '<MODEL_NAME>',
              ExecutionRoleArn = role,
              Containers       = [container])
```

3. (Opsional) Jika Anda menggunakan pipeline inferensi serial, dapatkan kontainer tambahan untuk disertakan dalam pipeline, dan sertakan dalam`Containers` argumen`CreateModel`:

```
preprocessor_container = {
               'Image':
 '<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/<PREPROCESSOR_IMAGE>:<TAG>'
            }
```

```
multi_model_container = {
               'Image':
 '<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/<IMAGE>:<TAG>',
               'ModelDataUrl': 's3://<BUCKET_NAME>/<PATH_TO_ARTIFACTS>',
               'Mode':         'MultiModel'
            }

response = sagemaker_client.create_model(
               ModelName        = '<MODEL_NAME>',
               ExecutionRoleArn = role,
               Containers       = [preprocessor_container, multi_model_container]
            )
```

> **Note**
>
> Anda dapat menggunakan satu multi-model-enabled titik akhir dalam pipeline inferensi serial.

4. (Opsional) Jika kasus penggunaan Anda tidak mendapat manfaat dari caching model, tetapkan nilai`ModelCacheSetting` bidang`MultiModelConfig` parameter ke`Disabled`, dan sertakan dalam`Container` argumen panggilan ke`create_model`. Nilai`ModelCacheSetting` bidang secara`Enabled` default.

```
container = {
                'Image': image,
                'ModelDataUrl': 's3://<BUCKET_NAME>/<PATH_TO_ARTIFACTS>',
                'Mode': 'MultiModel'
                'MultiModelConfig': {
                        // Default value is 'Enabled'
                        'ModelCacheSetting': 'Disabled'
                }
            }

response = sagemaker_client.create_model(
               ModelName        = '<MODEL_NAME>',
               ExecutionRoleArn = role,
               Containers       = [container]
            )
```

5. Konfigurasikan endpoint multi-model dengan instans yang didukung GPU untuk model. Sebaiknya konfigurasikan endpoint Anda dengan lebih dari satu instans untuk memungkinkan ketersediaan tinggi dan klik cache yang lebih tinggi.

```
response = sagemaker_client.create_endpoint_config(
                EndpointConfigName = '<ENDPOINT_CONFIG_NAME>',
                ProductionVariants=[
                     {
                        'InstanceType':        'ml.g4dn.4xlarge',
                        'InitialInstanceCount': 2,
                        'InitialVariantWeight': 1,
                        'ModelName':            '<MODEL_NAME>',
                        'VariantName':          'AllTraffic'
                      }
                ]
           )
```

6. Buat endpoint multi-model menggunakan`EndpointConfigName` parameter`EndpointName` and.

```
response = sagemaker_client.create_endpoint(
              EndpointName       = '<ENDPOINT_NAME>',
              EndpointConfigName = '<ENDPOINT_CONFIG_NAME>')
```

Memanggil Endpoint Multi-Model

Untuk memanggil endpoint multi-model, gunakan [invoke_endpoint](#)from the SageMaker Runtime seperti Anda akan memanggil endpoint model tunggal, dengan satu perubahan. Lulus`TargetModel` parameter baru yang menentukan model mana di titik akhir yang akan ditargetkan. `InvokeEndpoint`Permintaan SageMaker Runtime mendukung`X-Amzn-SageMaker-Target-Model` sebagai header baru yang mengambil jalur relatif dari model yang ditentukan untuk pemanggilan. SageMaker Sistem membangun jalur absolut model dengan menggabungkan awalan yang disediakan sebagai bagian dari panggilan`CreateModel` API dengan jalur relatif model.

Prosedur berikut ini sama untuk endpoint multi-model CPU dan yang didukung GPU.

AWS SDK for Python (Boto 3)

Contoh permintaan prediksi berikut menggunakan [AWSSDK for Python (Boto 3)](#) dalam notebook sampel.

```
response = runtime_sagemaker_client.invoke_endpoint(
                        EndpointName = "<ENDPOINT_NAME>",
                        ContentType  = "text/csv",
                        TargetModel  = "<MODEL_FILENAME>.tar.gz",
                        Body         = body)
```

AWS CLI

Contoh berikut menunjukkan bagaimana membuat permintaan CSV dengan dua baris menggunakanAWS Command Line Interface (AWS CLI):

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name "<ENDPOINT_NAME>" \
  --body "1.0,2.0,5.0"$'\n'"2.0,3.0,4.0" \
  --content-type "text/csv" \
  --target-model "<MODEL_NAME>.tar.gz"
  output_file.txt
```

Informasi`output_file.txt` dengan tentang permintaan inferensi Anda dibuat jika kesimpulan berhasil. Untuk contoh selengkapnya tentang cara membuat prediksi denganAWS CLI, lihat Membuat prediksi dengan dokumentasi SageMaker Python SDK.AWS CLI

Titik akhir multi-model secara dinamis memuat model target sesuai kebutuhan. Anda dapat mengamati ini saat menjalankan Notebook Sampel MME saat melakukan iterasi melalui pemanggilan acak terhadap beberapa model target yang dihosting di belakang satu titik akhir. Permintaan pertama terhadap model tertentu membutuhkan waktu lebih lama karena model harus diunduh dari Amazon Simple Storage Service (Amazon S3) dan dimuat ke dalam memori. Ini disebut cold start, dan diharapkan pada titik akhir multi-model untuk mengoptimalkan kinerja harga yang lebih baik bagi pelanggan. Panggilan berikutnya selesai lebih cepat karena tidak ada overhead tambahan setelah model dimuat.

> **Note**
>
> Untuk instans yang didukung GPU, kode respons HTTP dengan 507 dari wadah GPU menunjukkan kurangnya memori atau sumber daya lainnya. Hal ini menyebabkan model yang tidak terpakai diturunkan dari wadah untuk memuat model yang lebih sering digunakan.

Coba Kembali Permintaan pada ModelNotReadyException Kesalahan

Pertama kali Anda memanggil`invoke_endpoint` model, model diunduh dari Amazon Simple Storage Service dan dimuat ke dalam wadah inferensi. Hal ini membuat panggilan pertama membutuhkan waktu lebih lama untuk kembali. Panggilan berikutnya ke model yang sama selesai lebih cepat, karena model sudah dimuat.

SageMaker mengembalikan respon untuk panggilan ke`invoke_endpoint` dalam waktu 60 detik. Beberapa model terlalu besar untuk diunduh dalam waktu 60 detik. Jika model tidak selesai memuat sebelum batas waktu tunggu 60 detik, permintaan untuk`invoke_endpoint` kembali dengan kode kesalahan`ModelNotReadyException`, dan model terus mengunduh dan memuat ke dalam wadah inferensi hingga 360 detik. Jika Anda mendapatkan kode`ModelNotReadyException` kesalahan untuk`invoke_endpoint` permintaan, coba lagi permintaan tersebut. Secara default,AWS SDK untuk Python (Boto 3) (menggunakan mode coba ulang Legacy) dan`invoke_endpoint` permintaan percobaan ulang Java yang mengakibatkan`ModelNotReadyException` kesalahan. Anda dapat mengonfigurasi strategi coba ulang untuk terus mencoba ulang permintaan hingga 360 detik. Jika Anda mengharapkan model Anda membutuhkan waktu lebih dari 60 detik untuk mengunduh dan memuat ke dalam wadah, atur batas waktu tunggu soket SDK menjadi 70 detik. Untuk informasi selengkapnya tentang mengonfigurasi strategi coba ulang untukAWS SDK for Python (Boto3), lihat Mengkonfigurasi mode coba lagi. Kode berikut menunjukkan contoh yang mengonfigurasi strategi coba ulang untuk mencoba kembali panggilan hingga`invoke_endpoint` 180 detik.

```
import boto3
from botocore.config import Config

# This example retry strategy sets the retry attempts to 2.
# With this setting, the request can attempt to download and/or load the model
# for upto 180 seconds: 1 orginal request (60 seconds) + 2 retries (120 seconds)
config = Config(
    read_timeout=70,
    retries={
        'max_attempts': 2  # This value can be adjusted to 5 to go up to the 360s max
 timeout
    }
)
runtime_sagemaker_client = boto3.client('sagemaker-runtime', config=config)
```

Menambah atau Menghapus Model

Anda dapat menerapkan model tambahan ke titik akhir multi-model dan segera memanggilnya melalui titik akhir tersebut. Saat menambahkan model baru, Anda tidak perlu memperbarui atau menurunkan titik akhir, sehingga Anda menghindari biaya pembuatan dan menjalankan titik akhir terpisah untuk setiap model baru. Proses untuk menambahkan dan menghapus model adalah sama untuk CPU dan endpoint multi-model yang didukung GPU.

SageMaker membongkar model yang tidak terpakai dari wadah saat instance mencapai kapasitas memori dan lebih banyak model perlu diunduh ke dalam wadah. SageMaker juga menghapus artefak model yang tidak terpakai dari volume penyimpanan instans ketika volume mencapai kapasitas dan model baru perlu diunduh. Pemanggilan pertama ke model yang baru ditambahkan membutuhkan waktu lebih lama karena titik akhir membutuhkan waktu untuk mengunduh model dari S3 ke memori kontainer dalam contoh hosting titik akhir

Dengan titik akhir yang sudah berjalan, salin satu set artefak model baru ke lokasi Amazon S3 di sana Anda menyimpan model Anda.

```
# Add an AdditionalModel to the endpoint and exercise it
aws s3 cp AdditionalModel.tar.gz s3://my-bucket/path/to/artifacts/
```

> **Important**
>
> Untuk memperbarui model, lanjutkan seperti yang Anda lakukan saat menambahkan model baru. Gunakan nama baru dan unik. Jangan menimpa artefak model di Amazon S3 karena versi lama model mungkin masih dimuat dalam kontainer atau pada volume penyimpanan instans pada titik akhir. Doa ke model baru kemudian dapat memanggil versi lama model.

Aplikasi klien dapat meminta prediksi dari model target tambahan segera setelah disimpan di S3.

```
response = runtime_sagemaker_client.invoke_endpoint(
                        EndpointName='<ENDPOINT_NAME>',
                        ContentType='text/csv',
                        TargetModel='AdditionalModel.tar.gz',
                        Body=body)
```

Untuk menghapus model dari endpoint multi-model, berhenti memanggil model dari klien dan menghapusnya dari lokasi S3 tempat artefak model disimpan.

Buat Container Anda Sendiri untuk Endpoint SageMaker Multi-Model

Lihat bagian berikut untuk membawa kontainer dan dependensi Anda sendiri ke titik akhir multi-model.

Topik

- Bawa dependensi Anda sendiri untuk endpoint multi-model pada instans yang didukung CPU
- Bawa dependensi Anda sendiri untuk titik akhir multi-model pada instans yang didukung GPU
- Menggunakan Toolkit SageMaker Inferensi
- Kontrak Kontainer Khusus untuk Endpoint Multi-Model

Bawa dependensi Anda sendiri untuk endpoint multi-model pada instans yang didukung CPU

Jika tidak ada gambar kontainer yang dibuat sebelumnya yang memenuhi kebutuhan Anda, Anda dapat membuat wadah Anda sendiri untuk digunakan dengan titik akhir multi-model yang didukung CPU.

Image Amazon Elastic Container Registry (Amazon ECR) kustom SageMaker yang diterapkan di Amazon diharapkan mematuhi kontrak dasarGunakan Kode Inferensi Anda Sendiri dengan Layanan Hosting yang dijelaskan dalam yang mengatur bagaimana SageMaker berinteraksi dengan container Docker yang menjalankan kode inferensi Anda sendiri. Agar wadah mampu memuat dan menyajikan beberapa model secara bersamaan, ada API dan perilaku tambahan yang harus diikuti. Kontrak tambahan ini mencakup API baru untuk memuat, membuat daftar, mendapatkan, dan membongkar model, dan API yang berbeda untuk memanggil model. Ada juga perilaku yang berbeda untuk skenario kesalahan yang perlu dipatuhi oleh API. Untuk menunjukkan bahwa kontainer mematuhi persyaratan tambahan, Anda dapat menambahkan perintah berikut ke file Docker Anda:

```
LABEL com.amazonaws.sagemaker.capabilities.multi-models=true
```

SageMaker juga menyuntikkan variabel lingkungan ke dalam wadah

```
SAGEMAKER_MULTI_MODEL=true
```

Jika Anda membuat endpoint multi-model untuk pipline inferensi serial, file Docker Anda harus memiliki label yang diperlukan untuk pipeline inferensi multi-model dan serial. Untuk informasi lebih lanjut tentang jaringan pipa informasi serial, lihatJalankan Prediksi Real-time dengan Pipeline Inferensi.

Untuk membantu Anda menerapkan persyaratan ini untuk wadah kustom, tersedia dua pustaka:

- Multi Model Server adalah kerangka open source untuk melayani model machine learning yang dapat diinstal dalam wadah untuk menyediakan front end yang memenuhi persyaratan untuk multi-model endpoint container API baru. Ini menyediakan kemampuan manajemen ujung depan dan model HTTP yang diperlukan oleh titik akhir multi-model untuk meng-host beberapa model dalam satu wadah, memuat model ke dalam dan membongkar model keluar dari wadah secara dinamis, dan melakukan inferensi pada model yang dimuat tertentu. Ini juga menyediakan backend pluggable yang mendukung handler backend kustom pluggable di mana Anda dapat menerapkan algoritma Anda sendiri.
- SageMaker Inferensi Toolkit adalah perpustakaan yang bootstraps Multi Model Server dengan konfigurasi dan pengaturan yang membuatnya kompatibel dengan SageMaker multi-model endpoint. Hal ini juga memungkinkan Anda untuk men-tweak parameter kinerja penting, seperti jumlah pekerja per model, tergantung pada kebutuhan skenario Anda.

Bawa dependensi Anda sendiri untuk titik akhir multi-model pada instans yang didukung GPU

Kemampuan bawa kontainer Anda sendiri (BYOC) pada titik akhir multi-model dengan instans yang didukung GPU saat ini tidak didukung oleh pustaka Multi Model Server dan SageMaker Inference Toolkit.

Untuk membuat titik akhir multi-model dengan instans yang didukung GPU, Anda dapat menggunakan Server Inferensi Triton NVIDIA yang SageMaker didukung dengan NVIDIA Triton Inference Containers. Untuk membawa dependensi Anda sendiri, Anda dapat membuat wadah Anda sendiri dengan NVIDIA Triton Inference Server yang SageMaker didukung sebagai gambar dasar ke file Docker Anda:

```
FROM 301217895009.dkr.ecr.us-west-2.amazonaws.com/sagemaker-tritonserver:22.07-py3
```

> **Important**
>
> Kontainer dengan Triton Inference Server adalah satu-satunya kontainer yang didukung yang dapat Anda gunakan untuk titik akhir multi-model yang didukung GPU.

Menggunakan Toolkit SageMaker Inferensi

> **Note**
>
> SageMaker Inference Toolkit hanya didukung untuk endpoint multi-model yang didukung CPU. Toolkit SageMaker Inferensi saat ini tidak didukung untuk titik akhir multi-model yang didukung GPU.

Kontainer yang sudah dibuat sebelumnya yang mendukung titik akhir multi-model tercantum dalamAlgoritma, kerangka kerja, dan instance yang didukung. Jika Anda ingin menggunakan kerangka kerja atau algoritma lainnya, Anda perlu membuat wadah. Cara termudah untuk melakukannya adalah dengan menggunakan SageMaker Inference Toolkit untuk memperpanjang kontainer yang sudah ada. Toolkit SageMaker inferensi adalah implementasi untuk server multi-model (MMS) yang menciptakan titik akhir yang dapat digunakan SageMaker. Untuk contoh notebook yang menunjukkan cara menyiapkan dan menerapkan wadah khusus yang mendukung titik akhir multi-model SageMaker, lihat Notebook Sampel BYOC Endpoint Multi-Model.

> **Note**
>
> Toolkit SageMaker inferensi hanya mendukung penangan model Python. Jika Anda ingin mengimplementasikan handler dalam bahasa lain, Anda harus membuat container sendiri yang mengimplementasikan API endpoint multi-model tambahan. Untuk informasi, lihat Kontrak Kontainer Khusus untuk Endpoint Multi-Model.

Untuk memperluas wadah dengan menggunakan toolkit SageMaker inferensi

1. Buat model handler. MMS mengharapkan handler model, yang merupakan file Python yang mengimplementasikan fungsi untuk pra-proses, mendapatkan preditions dari model, dan memproses output dalam handler model. Untuk contoh handler model, lihat model_handler.py dari contoh notebook.
2. Impor toolkit inferensi dan gunakan`model_server.start_model_server` fungsinya untuk memulai MMS. Contoh berikut adalah dari`dockerd-entrypoint.py` file dari notebook sampel. Perhatikan bahwa panggilan untuk`model_server.start_model_server` melewati handler model yang dijelaskan pada langkah sebelumnya:

```
import subprocess
```

```
import sys
import shlex
import os
from retrying import retry
from subprocess import CalledProcessError
from sagemaker_inference import model_server

def _retry_if_error(exception):
    return isinstance(exception, CalledProcessError or OSError)

@retry(stop_max_delay=1000 * 50,
       retry_on_exception=_retry_if_error)
def _start_mms():
    # by default the number of workers per model is 1, but we can configure it
 through the
    # environment variable below if desired.
    # os.environ['SAGEMAKER_MODEL_SERVER_WORKERS'] = '2'
    model_server.start_model_server(handler_service='/home/model-server/
model_handler.py:handle')

def main():
    if sys.argv[1] == 'serve':
        _start_mms()
    else:
        subprocess.check_call(shlex.split(' '.join(sys.argv[1:])))

    # prevent docker exit
    subprocess.call(['tail', '-f', '/dev/null'])

main()
```

3. Dalam Anda`Dockerfile`, salin handler model dari langkah pertama dan tentukan file Python dari langkah sebelumnya sebagai entrypoint di Anda`Dockerfile`. Baris berikut berasal dari Dockerfile yang digunakan dalam contoh notebook:

```
# Copy the default custom service file to handle incoming data and inference
 requests
COPY model_handler.py /home/model-server/model_handler.py

# Define an entrypoint script for the docker image
ENTRYPOINT ["python", "/usr/local/bin/dockerd-entrypoint.py"]
```

4. Bangun dan daftarkan wadah Anda. Skrip shell berikut dari notebook contoh membangun wadah dan mengunggahnya ke repositori Amazon Elastic Container Registry diAWS akun Anda:

```
%%sh

# The name of our algorithm
algorithm_name=demo-sagemaker-multimodel

cd container

account=$(aws sts get-caller-identity --query Account --output text)

# Get the region defined in the current configuration (default to us-west-2 if none
 defined)
region=$(aws configure get region)
region=${region:-us-west-2}

fullname="${account}.dkr.ecr.${region}.amazonaws.com/${algorithm_name}:latest"

# If the repository doesn't exist in ECR, create it.
aws ecr describe-repositories --repository-names "${algorithm_name}" > /dev/null
 2>&1

if [ $? -ne 0 ]
then
    aws ecr create-repository --repository-name "${algorithm_name}" > /dev/null
fi

# Get the login command from ECR and execute it directly
$(aws ecr get-login --region ${region} --no-include-email)

# Build the docker image locally with the image name and then push it to ECR
# with the full name.

docker build -q -t ${algorithm_name} .
docker tag ${algorithm_name} ${fullname}

docker push ${fullname}
```

Anda sekarang dapat menggunakan wadah ini untuk menyebarkan endpoint multi-model di SageMaker.

Topik

Kontrak Kontainer Khusus untuk Endpoint Multi-Model

Untuk menangani beberapa model, kontainer Anda harus mendukung serangkaian API yang memungkinkan Amazon SageMaker berkomunikasi dengan wadah untuk memuat, mencantumkan, mendapatkan, dan membongkar model sesuai kebutuhan. Digunakan dalam set baru API sebagai parameter input kunci.`model_name` Kontainer pelanggan diharapkan untuk melacak model dimuat menggunakan`model_name` sebagai kunci pemetaan. Juga,`model_name` adalah pengenal buram dan belum tentu nilai`TargetModel` parameter yang dilewatkan ke`InvokeEndpoint` API. `TargetModel`Nilai asli dalam`InvokeEndpoint` permintaan diteruskan ke kontainer di API sebagai`X-Amzn-SageMaker-Target-Model` header yang dapat digunakan untuk tujuan logging.

> **Note**
>
> Titik akhir multi-model untuk instans yang didukung GPU saat ini hanya didukung dengan wadah SageMaker NVIDIA Triton Inference Server. Kontainer ini sudah mengimplementasikan kontrak yang didefinisikan di bawah ini. Pelanggan dapat langsung menggunakan wadah ini dengan titik akhir GPU multi-model mereka, tanpa pekerjaan tambahan.

Anda dapat mengonfigurasi API berikut pada kontainer Anda untuk titik akhir multi-model yang didukung CPU.

Topik

Muat Model API

Menginstruksikan wadah untuk memuat hadir model tertentu di`url` bidang tubuh ke dalam memori wadah pelanggan dan untuk melacak itu dengan ditugaskan`model_name`. Setelah model dimuat, wadah harus siap untuk melayani permintaan inferensi menggunakan ini`model_name`.

```
POST /models HTTP/1.1
Content-Type: application/json
Accept: application/json

{
     "model_name" : "{model_name}",
     "url" : "/opt/ml/models/{model_name}/model",
}
```

> **Note**
>
> Jika`model_name` sudah dimuat, API ini harus mengembalikan 409. Setiap kali model tidak dapat dimuat karena kurangnya memori atau sumber daya lainnya, API ini harus mengembalikan kode status HTTP 507 SageMaker, yang kemudian memulai pembongkaran model yang tidak digunakan untuk merebut kembali.

API Model Daftar

Mengembalikan daftar model dimuat ke dalam memori wadah pelanggan.

```
GET /models HTTP/1.1
Accept: application/json

Response =
{
    "models": [
        {
             "modelName" : "{model_name}",
             "modelUrl" : "/opt/ml/models/{model_name}/model",
        },
        {
            "modelName" : "{model_name}",
            "modelUrl" : "/opt/ml/models/{model_name}/model",
        },
```

```
        ....
    ]
}
```

API ini juga mendukung pagination.

```
GET /models HTTP/1.1
Accept: application/json

Response =
{
    "models": [
        {
             "modelName" : "{model_name}",
             "modelUrl" : "/opt/ml/models/{model_name}/model",
        },
        {
            "modelName" : "{model_name}",
            "modelUrl" : "/opt/ml/models/{model_name}/model",
        },
        ....
    ]
}
```

SageMaker awalnya dapat memanggil List Model API tanpa memberikan nilai untuk`next_page_token`. Jika`nextPageToken` bidang dikembalikan sebagai bagian dari respon, itu akan diberikan sebagai nilai untuk`next_page_token` dalam panggilan Model Daftar berikutnya. Jika a`nextPageToken` tidak dikembalikan, itu berarti tidak ada lagi model untuk dikembalikan.

Dapatkan Model API

Ini adalah API baca sederhana pada`model_name` entitas.

```
GET /models/{model_name} HTTP/1.1
Accept: application/json

{
     "modelName" : "{model_name}",
     "modelUrl" : "/opt/ml/models/{model_name}/model",
}
```

> **Note**
>
> Jika`model_name` tidak dimuat, API ini harus mengembalikan 404.

Bongkar Model API

Menginstruksikan SageMaker platform untuk menginstruksikan wadah pelanggan untuk membongkar model dari memori. Ini memulai penggusuran model kandidat sebagaimana ditentukan oleh platform saat memulai proses pemuatan model baru. Sumber daya yang disediakan`model_name` harus direklamasi oleh kontainer saat API ini mengembalikan respons.

```
DELETE /models/{model_name}
```

> **Note**
>
> Jika`model_name` tidak dimuat, API ini harus mengembalikan 404.

Memanggil Model API

Membuat permintaan prediksi dari tertentu yang`model_name` disediakan. `InvokeEndpoint`Permintaan SageMaker Runtime mendukung`X-Amzn-SageMaker-Target-Model` sebagai header baru yang mengambil jalur relatif dari model yang ditentukan untuk pemanggilan. SageMaker Sistem membangun jalur absolut model dengan menggabungkan awalan yang disediakan sebagai bagian dari panggilan`CreateModel` API dengan jalur relatif model.

```
POST /models/{model_name}/invoke HTTP/1.1
Content-Type: ContentType
Accept: Accept
X-Amzn-SageMaker-Custom-Attributes: CustomAttributes
X-Amzn-SageMaker-Target-Model: [relativePath]/{artifactName}.tar.gz
```

> **Note**
>
> Jika`model_name` tidak dimuat, API ini harus mengembalikan 404.

Selain itu, pada instance GPU, jika`InvokeEndpoint` gagal karena kurangnya memori atau sumber daya lainnya, API ini harus mengembalikan kode status HTTP 507 SageMaker, yang kemudian memulai pembongkaran model yang tidak digunakan untuk merebut kembali.

Keamanan Titik Akhir Multi-Model

Model dan data dalam endpoint multi-model ditempatkan bersama pada volume penyimpanan instans dan dalam memori kontainer. Semua instans untuk SageMaker endpoint Amazon berjalan pada satu wadah penyewa yang Anda miliki. Hanya model Anda yang dapat berjalan di titik akhir multi-model Anda. Anda bertanggung jawab untuk mengelola pemetaan permintaan ke model dan menyediakan akses bagi pengguna ke model target yang benar. SageMaker menggunakan peran IAM untuk menyediakan kebijakan berbasis identitas IAM yang Anda gunakan untuk menentukan tindakan dan sumber daya yang diizinkan atau ditolak, dan kondisi di mana tindakan tersebut diperbolehkan atau ditolak.

Secara default, prinsipal IAM dengan `InvokeEndpoint`izin pada titik akhir multi-model dapat memanggil model apa pun di alamat awalan S3 yang ditentukan dalam `CreateModel`operasi, asalkan Peran Eksekusi IAM yang didefinisikan dalam operasi memiliki izin untuk mengunduh model. Jika Anda perlu membatasi `InvokeEndpoint`akses ke serangkaian model terbatas di S3, Anda dapat melakukan salah satu hal berikut:

- Batasi`InvokeEndpont` panggilan ke model tertentu yang dihosting di titik akhir dengan menggunakan kunci kondisi`sagemaker:TargetModel` IAM. Misalnya, kebijakan berikut mengizinkan`InvokeEndpont` permintaan hanya jika nilai`TargetModel` bidang cocok dengan salah satu ekspresi reguler yang ditentukan:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sagemaker:InvokeEndpoint"
            ],
            "Effect": "Allow",
            "Resource":
            "arn:aws:sagemaker:region:account-id:endpoint/endpoint_name",
            "Condition": {
                // TargetModel provided must be from this set of values
                "StringLike": {
                    "sagemaker:TargetModel": ["company_a/*", "common/*"]
                }
```

```
            }
        }
    ]
}
```

Untuk informasi tentang kunci SageMaker kondisi, lihat Kunci Kondisi untuk Amazon SageMaker di PanduanAWS Identity and Access Management Pengguna.

- Buat titik akhir multi-model dengan awalan S3 yang lebih ketat.

Untuk informasi selengkapnya tentang cara SageMaker menggunakan peran untuk mengelola akses ke titik akhir dan melakukan operasi atas nama Anda, lihatSageMaker Peran. Pelanggan Anda mungkin juga memiliki persyaratan isolasi data tertentu yang ditentukan oleh persyaratan kepatuhan mereka sendiri yang dapat dipenuhi menggunakan identitas IAM.

CloudWatch Metrik untuk Penerapan Titik Akhir Multi-Model

Amazon SageMaker menyediakan metrik untuk titik akhir sehingga Anda dapat memantau kecepatan hit cache, jumlah model yang dimuat, dan waktu tunggu model untuk memuat, mengunduh, dan mengunggah pada titik akhir multi-model. Beberapa metrik berbeda untuk titik akhir multi-model yang didukung CPU dan GPU, sehingga bagian berikut menjelaskan CloudWatch metrik Amazon yang dapat Anda gunakan untuk setiap jenis titik akhir multi-model.

Untuk informasi selengkapnya tentang metrik, lihat Metrik Pemuatan Model Titik Akhir Multi-Model dan Metrik Instans Model Titik Akhir Multi-Model diPantau Amazon SageMaker dengan Amazon CloudWatch. Metrik per model tidak didukung.

CloudWatch metrik untuk titik akhir multi-model yang didukung CPU

Anda dapat memantau metrik berikut pada titik akhir multi-model yang didukung CPU.

`AWS/SageMaker`Namespace mencakup metrik pemuatan model berikut dari panggilan ke InvokeEndpoint.

Metrik tersedia dalam frekuensi 1 menit.

Untuk informasi tentang berapa lama CloudWatch metrik dipertahankan, lihat GetMetricStatisticsdi Amazon CloudWatch API Reference.

Metrik Pemuatan Model Titik Akhir Multi-Model

Metrik	Deskripsi
`ModelLoadingWaitTime`	Interval waktu permintaan pemanggilan telah menunggu model target diunduh, atau dimuat, atau keduanya untuk melakukan inferensi. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelUnloadingTime`	Interval waktu yang dibutuhkan untuk membongkar model melalui panggilan`UnloadModel` API kontainer. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelDownloadingTime`	Interval waktu yang diperlukan untuk mengunduh model dari Amazon Simple Storage Service (Amazon S3). Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelLoadingTime`	Interval waktu yang dibutuhkan untuk memuat model melalui panggilan `LoadModel` API kontainer. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelCacheHit`	Jumlah`InvokeEndpoint` permintaan yang dikirim ke titik akhir multi-model yang modelnya sudah dimuat. Statistik rata-rata menunjukkan rasio permintaan yang modelnya sudah dimuat. Unit: Tidak ada

Metrik	Deskripsi
	Statistik yang valid: Rata-rata, Jumlah, Hitungan Sampel

Dimensi untuk Metrik Pemuatan Model Titik Akhir Multi-Model

Dimensi	Deskripsi
`EndpointName, VariantName`	Memfilter metrik pemanggilan titik akhir untuk titik akhir dan varian yang ditentukan.`ProductionVariant`

`/aws/sagemaker/Endpoints`Ruang nama menyertakan metrik instans berikut dari panggilan ke InvokeEndpoint.

Metrik tersedia dalam frekuensi 1 menit.

Untuk informasi tentang berapa lama CloudWatch metrik dipertahankan, lihat GetMetricStatisticsdi Amazon CloudWatch API Reference.

Metrik Instans Model Titik Akhir Multi-Model

Metrik	Deskripsi
`LoadedModelCount`	Jumlah model yang dimuat dalam wadah endpoint multi-model. Metrik ini dipancarkan per instans. Statistik rata-rata dengan jangka waktu 1 menit memberi tahu Anda jumlah rata-rata model yang dimuat per instans. Statistik Sum memberi tahu Anda jumlah total model yang dimuat di semua instance di titik akhir. Model yang dilacak metrik ini belum tentu unik karena model mungkin dimuat dalam beberapa kontainer di titik akhir. Unit: Tidak ada

Metrik	Deskripsi
	Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`CPUUtilization`	Jumlah pemanfaatan masing-masing inti CPU. Pemanfaatan CPU dari setiap rentang inti adalah 0-100. Misalnya, jika ada empat CPU,`CPUUtilization` kisarannya adalah 0% - 400%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan CPU dari kontainer primer dan tambahan pada instance. Unit: Persen
`MemoryUtilization`	Persentase memori yang digunakan oleh kontainer pada instance. Rentang nilai ini adalah 0% - 100%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan memori kontainer primer dan tambahan pada instance. Unit: Persen
`DiskUtilization`	Persentase ruang disk yang digunakan oleh kontainer pada instance. Rentang nilai ini adalah 0% - 100%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan ruang disk dari wadah primer dan tambahan pada instance. Unit: Persen

CloudWatch metrik untuk penerapan titik akhir multi-model GPU

Anda dapat memantau metrik berikut pada titik akhir multi-model yang didukung GPU.

`AWS/SageMaker`Namespace mencakup metrik pemuatan model berikut dari panggilan ke InvokeEndpoint.

Metrik tersedia dalam frekuensi 1 menit.

Untuk informasi tentang berapa lama CloudWatch metrik dipertahankan, lihat GetMetricStatisticsdi Amazon CloudWatch API Reference.

Metrik Pemuatan Model Titik Akhir Multi-Model

Metrik	Deskripsi
`ModelLoadingWaitTime`	Interval waktu permintaan pemanggilan telah menunggu model target diunduh, atau dimuat, atau keduanya untuk melakukan inferensi. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelUnloadingTime`	Interval waktu yang dibutuhkan untuk membongkar model melalui panggilan`UnloadModel` API kontainer. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelDownloadingTime`	Interval waktu yang diperlukan untuk mengunduh model dari Amazon Simple Storage Service (Amazon S3). Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelLoadingTime`	Interval waktu yang dibutuhkan untuk memuat model melalui panggilan `LoadModel` API kontainer. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`ModelCacheHit`	Jumlah`InvokeEndpoint` permintaan yang dikirim ke titik akhir multi-model yang modelnya sudah dimuat. Statistik rata-rata menunjukkan rasio permintaan yang modelnya sudah dimuat.

Metrik	Deskripsi
	Unit: Tidak ada Statistik yang valid: Rata-rata, Jumlah, Hitungan Sampel

Dimensi untuk Metrik Pemuatan Model Titik Akhir Multi-Model

Dimensi	Deskripsi
`EndpointName, VariantName`	Memfilter metrik pemanggilan titik akhir untuk titik akhir dan varian yang ditentukan.`ProductionVariant`

`/aws/sagemaker/Endpoints`Ruang nama menyertakan metrik instans berikut dari panggilan ke InvokeEndpoint.

Metrik tersedia dalam frekuensi 1 menit.

Untuk informasi tentang berapa lama CloudWatch metrik dipertahankan, lihat GetMetricStatisticsdi Amazon CloudWatch API Reference.

Metrik Instans Model Titik Akhir Multi-Model

Metrik	Deskripsi
`LoadedModelCount`	Jumlah model yang dimuat dalam wadah endpoint multi-model. Metrik ini dipancarkan per instans. Statistik rata-rata dengan jangka waktu 1 menit memberi tahu Anda jumlah rata-rata model yang dimuat per instans. Statistik Sum memberi tahu Anda jumlah total model yang dimuat di semua instance di titik akhir. Model yang dilacak metrik ini belum tentu unik karena model mungkin dimuat dalam beberapa kontainer di titik akhir. Unit: Tidak ada

Metrik	Deskripsi
	Statistik yang valid: Rata-rata, Jumlah, Min, Min, Min, Maks, Hitungan Sampel
`CPUUtilization`	Jumlah pemanfaatan masing-masing inti CPU. Pemanfaatan CPU dari setiap rentang inti adalah 0-100. Misalnya, jika ada empat CPU,`CPUUtilization` kisarannya adalah 0% - 400%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan CPU dari kontainer primer dan tambahan pada instance. Unit: Persen
`MemoryUtilization`	Persentase memori yang digunakan oleh kontainer pada instance. Rentang nilai ini adalah 0% -100%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan memori kontainer primer dan tambahan pada instance. Unit: Persen
`GPUUtilization`	Persentase unit GPU yang digunakan oleh kontainer pada sebuah instance. Nilai dapat berkisar antara 0-100 dan dikalikan dengan jumlah GPU. Misalnya, jika ada empat GPU,`GPUUtilization` kisarannya adalah 0% - 400%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan GPU dari wadah primer dan tambahan pada instance. Unit: Persen

Metrik	Deskripsi
`GPUMemory Utilization`	Persentase memori GPU yang digunakan oleh kontainer pada sebuah instance. Rentang nilai 0-100 dan dikalikan dengan jumlah GPU. Misalnya, jika ada empat GPU,`GPUMemoryUtilization` kisarannya adalah 0% -400%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan memori GPU dari wadah primer dan tambahan pada instance. Unit: Persen
`DiskUtilization`	Persentase ruang disk yang digunakan oleh kontainer pada instance. Rentang nilai ini adalah 0% - 100%. Untuk varian endpoint, nilainya adalah jumlah pemanfaatan ruang disk dari wadah primer dan tambahan pada instance. Unit: Persen

Mengatur Kebijakan Auto Scaling untuk Penerapan Titik Akhir Multi-Model

SageMaker titik akhir multi-model sepenuhnya mendukung penskalaan otomatis, yang mengelola replika model untuk memastikan skala model berdasarkan pola lalu lintas. Sebaiknya konfigurasikan endpoint multi-model dan ukuran instans Anda berdasarkanRekomendasi instans untuk penerapan titik akhir multi-model dan juga menyiapkan penskalaan auto berbasis instans untuk titik akhir Anda. Tingkat pemanggilan yang digunakan untuk memicu peristiwa skala otomatis didasarkan pada kumpulan prediksi agregat di seluruh set lengkap model yang dilayani oleh titik akhir. Untuk detail tambahan tentang menyiapkan penskalaan auto titik akhir, lihat Menskalakan SageMaker Model Amazon Secara Otomatis.

Anda dapat mengatur kebijakan penskalaan auto dengan metrik standar dan kustom pada titik akhir multi-model yang didukung CPU dan GPU.

> **Note**
>
> SageMaker metrik endpoint multi-model tersedia dalam perincian satu menit.

Menetapkan kebijakan penskalaan

Untuk menentukan metrik dan nilai target untuk kebijakan penskalaan, Anda dapat mengonfigurasi kebijakan penskalaan pelacakan target. Anda dapat menggunakan metrik yang ditentukan sebelumnya atau metrik khusus.

Konfigurasi kebijakan penskalaan diwakili oleh blok JSON. Anda menyimpan konfigurasi kebijakan penskalaan sebagai blok JSON dalam file teks. Anda menggunakan file teks tersebut saat memicu AWS CLI atau Application Auto Scaling API. Untuk informasi selengkapnya tentang sintaks konfigurasi kebijakan, lihat `TargetTrackingScalingPolicyConfiguration` dalam Referensi Application Auto Scaling API.

Opsi berikut tersedia untuk menetapkan konfigurasi kebijakan penskalaan pelacakan target.

Menggunakan metrik yang sudah ditentukan sebelumnya

Untuk menentukan kebijakan penskalaan pelacakan target untuk varian, gunakan metrik yang`SageMakerVariantInvocationsPerInstance` sudah ditentukan sebelumnya. `SageMakerVariantInvocationsPerInstance`adalah jumlah rata-rata kali per menit yang setiap instance untuk varian dipanggil. Kami sangat menyarankan untuk menggunakan metrik ini.

Untuk menggunakan metrik yang sudah ditentukan sebelumnya dalam kebijakan penskalaan, buat konfigurasi pelacakan target untuk kebijakan Anda. Dalam konfigurasi pelacakan target,`PredefinedMetricSpecification` sertakan metrik yang sudah ditentukan sebelumnya dan`TargetValue` untuk nilai target metrik tersebut.

Contoh berikut adalah konfigurasi kebijakan umum untuk penskalaan pelacakan target untuk varian. Dalam konfigurasi ini, kita menggunakan metrik`SageMakerVariantInvocationsPerInstance` yang telah ditentukan untuk menyesuaikan jumlah instance varian sehingga setiap instance memiliki`InvocationsPerInstance` metrik`70`.

```
{"TargetValue": 70.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "InvocationsPerInstance"
    }
}
```

> **Note**
>
> Kami menyarankan agar Anda`InvocationsPerInstance` menggunakan titik akhir multi-model. `TargetValue`Untuk metrik ini bergantung pada persyaratan latensi aplikasi Anda. Kami juga menyarankan Anda memuat uji endpoint Anda untuk mengatur nilai parameter penskalaan yang sesuai. Untuk mempelajari lebih lanjut tentang memuat pengujian dan menyiapkan penskalaan otomatis untuk titik akhir Anda, lihat blog Mengonfigurasi titik akhir inferensi penskalaan otomatis di Amazon SageMaker.

Menggunakan metrik khusus

Jika Anda perlu menentukan kebijakan penskalaan pelacakan target yang memenuhi persyaratan kustom Anda, tentukan metrik kustom. Anda dapat menentukan metrik kustom berdasarkan metrik varian produksi apa pun yang berubah proporsinya terhadap penskalaan.

Tidak semua SageMaker metrik berfungsi untuk pelacakan target. Metrik harus berupa metrik pemanfaatan yang valid, dan metrik harus menjelaskan seberapa sibuk instance. Nilai metrik harus meningkat atau menurun dengan proporsi terbalik terhadap jumlah instans varian. Artinya, nilai metrik harus menurun ketika jumlah instans meningkat.

> **Important**
>
> Sebelum menerapkan penskalaan otomatis dalam produksi, Anda harus menguji penskalaan otomatis dengan metrik khusus.

Contoh metrik khusus untuk titik akhir multi-model yang didukung CPU

Contoh berikut adalah konfigurasi pelacakan target untuk kebijakan penskalaan. Dalam konfigurasi ini, untuk model bernama`my-model`, metrik kustom untuk`CPUUtilization` menyesuaikan jumlah instans pada endpoint berdasarkan penggunaan CPU rata-rata sebesar 50% di semua instans.

```
{"TargetValue": 50,
    "CustomizedMetricSpecification":
    {"MetricName": "CPUUtilization",
        "Namespace": "/aws/sagemaker/Endpoints",
        "Dimensions": [
            {"Name": "EndpointName", "Value": "my-endpoint" },
```

```
            {"Name": "ModelName","Value": "my-model"}
        ],
        "Statistic": "Average",
        "Unit": "Percent"
    }
}
```

Contoh metrik khusus untuk titik akhir multi-model yang didukung GPU

Contoh berikut adalah konfigurasi pelacakan target untuk kebijakan penskalaan. Dalam konfigurasi ini, untuk model bernama`my-model`, metrik kustom`GPUUtilization` menyesuaikan hitungan instans pada titik akhir berdasarkan pemanfaatan GPU rata-rata 50% di semua instance.

```
{"TargetValue": 50,
    "CustomizedMetricSpecification":
    {"MetricName": "GPUUtilization",
        "Namespace": "/aws/sagemaker/Endpoints",
        "Dimensions": [
            {"Name": "EndpointName", "Value": "my-endpoint" },
            {"Name": "ModelName","Value": "my-model"}
        ],
        "Statistic": "Average",
        "Unit": "Percent"
    }
}
```

Menambahkan periode jeda pakai

Untuk menambahkan periode jeda pakai untuk menskalakan keluar titik akhir Anda, tentukan nilai, dalam detik, untuk`ScaleOutCooldown`. Demikian pula, untuk menambahkan periode jeda pakai untuk menskalakan dalam model Anda, tambahkan nilai, dalam detik, untuk`ScaleInCooldown`. Untuk informasi selengkapnya tentang `ScaleInCooldown` dan `ScaleOutCooldown`, lihat [`TargetTrackingScalingPolicyConfiguration`] dalam referensi Application Auto Scaling API.

Berikut ini adalah contoh konfigurasi pelacakan target untuk kebijakan penskalaan. Dalam konfigurasi ini, metrik`SageMakerVariantInvocationsPerInstance` yang sudah ditentukan sebelumnya digunakan untuk menyesuaikan penskalaan berdasarkan rata-rata`70` di seluruh instans varian tersebut. Konfigurasi ini menyediakan periode jeda pakai skala masuk selama 10 menit dan periode jeda pakai skala keluar selama 5 menit.

```
{"TargetValue": 70.0,
```

```
    "PredefinedMetricSpecification":
    {"PredefinedMetricType": "SageMakerVariantInvocationsPerInstance"
    },
    "ScaleInCooldown": 600,
    "ScaleOutCooldown": 300
}
```

Host beberapa model yang menggunakan wadah berbeda di belakang satu titik akhir

SageMaker titik akhir multi-kontainer memungkinkan pelanggan untuk menerapkan beberapa kontainer, yang menggunakan model atau kerangka kerja yang berbeda, pada satu titik akhir. SageMaker Kontainer dapat dijalankan secara berurutan sebagai pipa inferensi, atau setiap kontainer dapat diakses secara individual dengan menggunakan pemanggilan langsung untuk meningkatkan pemanfaatan titik akhir dan mengoptimalkan biaya.

Untuk informasi tentang menjalankan kontainer di titik akhir multi-kontainer secara berurutan, lihat. Model host bersama dengan logika pra-pemrosesan sebagai pipa inferensi serial di belakang satu titik akhir

Untuk informasi tentang menjalankan kontainer tertentu di titik akhir multi-kontainer, lihat Gunakan titik akhir multi-kontainer dengan pemanggilan langsung

Topik

- Buat titik akhir multi-kontainer (Boto 3)
- Memperbarui titik akhir multi-kontainer
- Hapus titik akhir multi-kontainer
- Gunakan titik akhir multi-kontainer dengan pemanggilan langsung

Buat titik akhir multi-kontainer (Boto 3)

Buat titik akhir Multi-kontainer dengan memanggil CreateModel, CreateEndpointConfig, dan CreateEndpointAPI seperti yang Anda lakukan untuk membuat titik akhir lainnya. Anda dapat menjalankan kontainer ini secara berurutan sebagai pipeline inferensi, atau menjalankan setiap kontainer individu dengan menggunakan pemanggilan langsung. Titik akhir multi-kontainer memiliki persyaratan berikut saat Anda menelepon: `create_model`

- Gunakan `Containers` parameter alih-alih`PrimaryContainer`, dan sertakan lebih dari satu wadah dalam `Containers` parameter.

- `ContainerHostname`Parameter diperlukan untuk setiap kontainer di titik akhir multi-kontainer dengan pemanggilan langsung.
- Atur `Mode` parameter `InferenceExecutionConfig` bidang `Direct` untuk pemanggilan langsung dari setiap kontainer, atau `Serial` gunakan kontainer sebagai pipa inferensi. Mode default adalah`Serial`.

> **Note**
>
> Saat ini ada batas hingga 15 kontainer yang didukung pada titik akhir multi-kontainer.

Contoh berikut membuat model multi-container untuk pemanggilan langsung.

1. Buat elemen wadah dan `InferenceExecutionConfig` dengan pemanggilan langsung.

```
container1 = {
                'Image': '123456789012.dkr.ecr.us-east-1.amazonaws.com/
myimage1:mytag',
                'ContainerHostname': 'firstContainer'
            }

container2 = {
                'Image': '123456789012.dkr.ecr.us-east-1.amazonaws.com/
myimage2:mytag',
                'ContainerHostname': 'secondContainer'
            }
inferenceExecutionConfig = {'Mode': 'Direct'}
```

2. Buat model dengan elemen wadah dan atur `InferenceExecutionConfig` bidang.

```
import boto3
sm_client = boto3.Session().client('sagemaker')

response = sm_client.create_model(
               ModelName = 'my-direct-mode-model-name',
               InferenceExecutionConfig = inferenceExecutionConfig,
               ExecutionRoleArn = role,
               Containers = [container1, container2]
           )
```

Untuk membuat endoint, Anda kemudian akan memanggil create_endpoint_config dan create_endpoint seperti yang Anda lakukan untuk membuat titik akhir lainnya.

Memperbarui titik akhir multi-kontainer

Untuk memperbarui titik akhir multi-kontainer, selesaikan langkah-langkah berikut.

1. Panggil create_model untuk membuat model baru dengan nilai baru untuk `Mode` parameter di bidang. `InferenceExecutionConfig`
2. Panggil create_endpoint_config untuk membuat konfigurasi titik akhir baru dengan nama yang berbeda dengan menggunakan model baru yang Anda buat pada langkah sebelumnya.
3. Panggil update_endpoint untuk memperbarui titik akhir dengan konfigurasi titik akhir baru yang Anda buat di langkah sebelumnya.

Hapus titik akhir multi-kontainer

Untuk menghapus titik akhir, panggil delete_endpoint, dan berikan nama titik akhir yang ingin Anda hapus sebagai parameter. `EndpointName`

Gunakan titik akhir multi-kontainer dengan pemanggilan langsung

SageMaker titik akhir multi-kontainer memungkinkan pelanggan untuk menerapkan beberapa kontainer untuk menerapkan model yang berbeda pada titik akhir. SageMaker Anda dapat meng-host hingga 15 kontainer inferensi yang berbeda pada satu titik akhir. Dengan menggunakan pemanggilan langsung, Anda dapat mengirim permintaan ke wadah inferensi tertentu yang dihosting pada titik akhir multi-kontainer.

Topik

- Memanggil titik akhir multi-kontainer dengan pemanggilan langsung
- Keamanan dengan titik akhir multi-kontainer dengan pemanggilan langsung
- Metrik untuk titik akhir multi-kontainer dengan pemanggilan langsung
- Titik akhir multi-kontainer skala otomatis
- Memecahkan masalah titik akhir multi-kontainer

Memanggil titik akhir multi-kontainer dengan pemanggilan langsung

Untuk memanggil titik akhir multi-kontainer dengan pemanggilan langsung, panggil invoke_endpoint karena Anda akan memanggil titik akhir lainnya, dan tentukan wadah mana yang ingin Anda panggil dengan menggunakan parameter. `TargetContainerHostname`

Contoh berikut secara langsung memanggil titik akhir multi-kontainer untuk mendapatkan prediksi. `secondContainer`

```
import boto3
runtime_sm_client = boto3.Session().client('sagemaker-runtime')

response = runtime_sm_client.invoke_endpoint(
   EndpointName ='my-endpoint',
   ContentType = 'text/csv',
   TargetContainerHostname='secondContainer',
   Body = body)
```

Untuk setiap permintaan pemanggilan langsung ke titik akhir multi-kontainer, hanya wadah yang `TargetContainerHostname` memproses permintaan pemanggilan. Anda akan mendapatkan kesalahan validasi jika Anda melakukan salah satu dari berikut:

- Tentukan `TargetContainerHostname` yang tidak ada di titik akhir
- Jangan tentukan nilai untuk `TargetContainerHostname` permintaan ke titik akhir yang dikonfigurasi untuk pemanggilan langsung
- Tentukan nilai untuk `TargetContainerHostname` permintaan ke titik akhir yang tidak dikonfigurasi untuk pemanggilan langsung.

Keamanan dengan titik akhir multi-kontainer dengan pemanggilan langsung

Untuk titik akhir multi-kontainer dengan pemanggilan langsung, ada beberapa kontainer yang dihosting dalam satu instance dengan berbagi memori dan volume penyimpanan. Anda bertanggung jawab untuk menggunakan kontainer yang aman, memelihara pemetaan permintaan yang benar untuk menargetkan kontainer, dan memberi pengguna akses yang benar ke kontainer target. SageMakermenggunakan peran IAM untuk memberikan kebijakan berbasis identitas IAM yang Anda gunakan untuk menentukan apakah akses ke sumber daya diizinkan atau ditolak untuk peran itu, dan dalam kondisi apa. Untuk informasi tentang peran IAM, lihat peran IAM di AWS Identity and

Access ManagementPanduan Pengguna. Untuk informasi tentang kebijakan berbasis identitas, lihat Kebijakan berbasis identitas dan kebijakan berbasis sumber daya.

Secara default, prinsipal IAM dengan `InvokeEndpoint` izin pada titik akhir multi-kontainer dengan pemanggilan langsung dapat memanggil penampung apa pun di dalam titik akhir dengan nama titik akhir yang Anda tentukan saat Anda memanggil. `invoke_endpoint` Jika Anda perlu membatasi `invoke_endpoint` akses ke kumpulan kontainer terbatas di dalam titik akhir multi-kontainer, gunakan kunci kondisi `sagemaker:TargetContainerHostname` IAM. Kebijakan berikut menunjukkan cara membatasi panggilan ke kontainer tertentu dalam titik akhir.

Kebijakan berikut hanya mengizinkan `invoke_endpoint` permintaan jika nilai `TargetContainerHostname` bidang cocok dengan salah satu ekspresi reguler yang ditentukan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sagemaker:InvokeEndpoint"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:sagemaker:region:account-id:endpoint/endpoint_name",
            "Condition": {
                "StringLike": {
                    "sagemaker:TargetContainerHostname": ["customIps*", "common*"]
                }
            }
        }
    ]
}
```

Kebijakan berikut menolak `invoke_endpoint` permintaan jika nilai `TargetContainerHostname` bidang cocok dengan salah satu ekspresi reguler yang ditentukan dalam `Deny` pernyataan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sagemaker:InvokeEndpoint"
            ],
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:sagemaker:region:account-id:endpoint/endpoint_name",
            "Condition": {
                "StringLike": {
                    "sagemaker:TargetContainerHostname": ["*"]
                }
            }
        },
        {
            "Action": [
                "sagemaker:InvokeEndpoint"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:sagemaker:region:account-id:endpoint/endpoint_name",
            "Condition": {
                "StringLike": {
                    "sagemaker:TargetContainerHostname": ["special*"]
                }
            }
        }
    ]
}
```

Untuk informasi tentang kunci SageMaker kondisi, lihat Condition Keys untuk SageMaker di Panduan AWS Identity and Access Management Pengguna.

Metrik untuk titik akhir multi-kontainer dengan pemanggilan langsung

Selain metrik titik akhir yang tercantum, SageMaker juga menyediakan metrik per Pantau Amazon SageMaker dengan Amazon CloudWatch kontainer.

Metrik per kontainer untuk titik akhir multi-kontainer dengan pemanggilan langsung ditempatkan dan dikategorikan ke dalam dua ruang nama: CloudWatch dan. `AWS/SageMaker` `aws/sagemaker/Endpoints` `AWS/SageMaker`Namespace mencakup metrik terkait pemanggilan, dan `aws/sagemaker/Endpoints` namespace mencakup metrik pemanfaatan memori dan CPU.

Tabel berikut mencantumkan metrik per kontainer untuk titik akhir multi-kontainer dengan pemanggilan langsung. Semua metrik menggunakan dimensi [`EndpointName, VariantName, ContainerName`], yang memfilter metrik pada titik akhir tertentu, untuk varian tertentu dan sesuai dengan wadah tertentu. Metrik ini memiliki nama metrik yang sama seperti pada pipeline inferensi, tetapi pada tingkat per kontainer []. `EndpointName, VariantName, ContainerName`

Nama Metrik	Deskripsi	Dimensi	NameSpace
`Invocations`	Jumlah `InvokeEndpoint` permintaan yang dikirim ke kontainer di dalam titik akhir. Untuk mendapatkan jumlah total permintaan yang dikirim ke wadah itu, gunakan `Sum` statistik. Unit: Tidak Ada Statistik yang valid:`Sum`, `SampleCount`	`EndpointName`, `VariantName`, `ContainerName`	`AWS/SageMaker`
`Invocation4XXErrors`	Jumlah `InvokeEndpoint` permintaan yang dikembalikan model kode respons `4xx` HTTP pada wadah tertentu. Untuk setiap `4xx` tanggapan, SageMaker kirimkan a1. Unit: Tidak Ada Statistik yang valid:`Average`, `Sum`	`EndpointName`, `VariantName`, `ContainerName`	`AWS/SageMaker`
`Invocation5XXErrors`	Jumlah `InvokeEndpoint` permintaan yang dikembalikan model kode respons `5xx` HTTP pada wadah tertentu. Untuk setiap `5xx` tanggapan, SageMaker kirimkan a1. Unit: Tidak	`EndpointName`, `VariantName`, `ContainerName`	`AWS/SageMaker`

	Ada Statistik yang valid:`Average`, `Sum`		
`Container Latency`	Waktu yang dibutuhkan wadah target untuk merespons seperti yang dilihat dari SageMaker. `Container Latency` termasuk waktu yang dibutuhkan untuk mengirim permintaan, untuk mengambil respons dari wadah model, dan untuk menyelesaikan inferensi dalam wadah. Unit: Mikrodetik Statistik yang valid:`Average`,,`Sum`,`Min` `Max` `Sample Count`	`EndpointName`, `VariantName`, `ContainerName`	`AWS/SageMaker`

`OverheadLatency`	Waktu ditambahkan ke waktu yang dibutuhkan untuk menanggapi permintaan klien dengan biaya SageMaker overhead. `OverheadLatency` diukur dari waktu yang SageMaker menerima permintaan hingga mengembalikan respons ke klien, dikurangi`ModelLatency` . Latensi overhead dapat bervariasi tergantung pada ukuran payload permintaan dan respons, frekuensi permintaan, dan otentikasi atau otorisasi permintaan, di antara faktor-faktor lainnya. Unit: Mikrodetik Statistik yang valid:`Average`,, `Sum` `Min` `Max`, `Jumlah Sampel `	`EndpointName` , `VariantName` , `ContainerName`	`AWS/SageMaker`

`CPUUtilization`	Persentase unit CPU yang digunakan oleh setiap kontainer yang berjalan pada sebuah instance. Nilainya berkisar dari 0% hingga 100%, dan dikalikan dengan jumlah CPU. Misalnya, jika ada empat CPU, `CPUUtilization` dapat berkisar dari 0% hingga 400%. Untuk titik akhir dengan pemanggilan langsung, jumlah metrik CPUUutilization sama dengan jumlah kontainer di titik akhir tersebut. Unit: Persen	`EndpointName` , `VariantName` , `ContainerName`	`aws/sagemaker/Endpoints`

`MemoryUti lizaton`	Persentase memori yang digunakan oleh setiap kontainer yang berjalan pada sebuah instance. Nilai ini berkisar dari 0% hingga 100%. Mirip dengan CPUUutili zation, di titik akhir dengan pemanggil an langsung, jumlah MemoryUtilization metrik sama dengan jumlah kontainer di titik akhir tersebut. Unit: Persen	`EndpointName` , `VariantName` , `ContainerName`	`aws/sagemaker/ Endpoints`

Semua metrik di tabel sebelumnya khusus untuk titik akhir multi-kontainer dengan pemanggilan langsung. Selain metrik per-kontainer khusus ini, ada juga metrik pada tingkat varian dengan dimensi `[EndpointName, VariantName]` untuk semua metrik dalam tabel yang diharapkan. `ContainerLatency`

Titik akhir multi-kontainer skala otomatis

Jika Anda ingin mengonfigurasi penskalaan otomatis untuk titik akhir multi-kontainer menggunakan `InvocationsPerInstance` metrik, kami menyarankan agar model di setiap kontainer menunjukkan pemanfaatan dan latensi CPU yang serupa pada setiap permintaan inferensi. Ini direkomendasikan karena jika lalu lintas ke titik akhir multi-kontainer bergeser dari model pemanfaatan CPU rendah ke model pemanfaatan CPU yang tinggi, tetapi volume panggilan keseluruhan tetap sama, titik akhir tidak skala dan mungkin tidak ada cukup contoh untuk menangani semua permintaan ke model pemanfaatan CPU tinggi. Untuk informasi tentang penskalaan titik akhir secara otomatis, lihat. Secara Otomatis Menskalakan SageMaker Model Amazon

Memecahkan masalah titik akhir multi-kontainer

Bagian berikut dapat membantu Anda memecahkan masalah kesalahan dengan titik akhir multi-kontainer.

Kesalahan Pemeriksaan Kesehatan Ping

Dengan beberapa kontainer, memori endpoint dan CPU berada di bawah tekanan yang lebih tinggi selama pembuatan endpoint. Secara khusus, `CPUUtilization` metrik `MemoryUtilization` dan lebih tinggi daripada titik akhir kontainer tunggal, karena tekanan pemanfaatan sebanding dengan jumlah kontainer. Karena itu, kami menyarankan Anda memilih jenis instance dengan memori dan CPU yang cukup untuk memastikan bahwa ada cukup memori pada instance agar semua model dimuat (panduan yang sama berlaku untuk menerapkan pipeline inferensi). Jika tidak, pembuatan titik akhir Anda mungkin gagal dengan kesalahan seperti`XXX did not pass the ping health check`.

Hilang accept-bind-to-port =true label Docker

Kontainer dalam titik akhir multi-kontainer mendengarkan pada port yang ditentukan dalam variabel `SAGEMAKER_BIND_TO_PORT` lingkungan, bukan port 8080. Ketika sebuah kontainer berjalan di titik akhir multi-kontainer, SageMaker secara otomatis menyediakan variabel lingkungan ini ke wadah. Jika variabel lingkungan ini tidak ada, kontainer default menggunakan port 8080. Untuk menunjukkan bahwa kontainer Anda mematuhi persyaratan ini, gunakan perintah berikut untuk menambahkan label ke Dockerfile Anda:

```
LABEL com.amazonaws.sagemaker.capabilities.accept-bind-to-port=true
```

Jika tidak, Anda akan melihat pesan kesalahan seperti `Your Ecr Image XXX does not contain required com.amazonaws.sagemaker.capabilities.accept-bind-to-port=true Docker label(s).`

Jika kontainer Anda perlu mendengarkan pada port kedua, pilih port dalam rentang yang ditentukan oleh variabel `SAGEMAKER_SAFE_PORT_RANGE` lingkungan. Tentukan nilai sebagai rentang inklusif dalam format *`XXXX - YYYY, di mana XXXX dan YYYY`* adalah bilangan bulat multi-digit. SageMaker memberikan nilai ini secara otomatis saat Anda menjalankan wadah di titik akhir multi-kontainer.

Model host bersama dengan logika pra-pemrosesan sebagai pipa inferensi serial di belakang satu titik akhir

Pipeline inferensi adalah SageMaker model Amazon yang terdiri dari urutan linier dua hingga lima belas kontainer yang memproses permintaan inferensi pada data. Anda menggunakan pipeline inferensi untuk menentukan dan menerapkan kombinasi algoritme SageMaker bawaan yang telah

dilatih sebelumnya dan algoritme kustom Anda sendiri yang dikemas dalam kontainer Docker. Anda dapat menggunakan pipeline inferensi untuk menggabungkan tugas sains data pra-pemrosesan, prediksi, dan pasca-pemrosesan. Pipa inferensi dikelola sepenuhnya.

Anda dapat menambahkan wadah SageMaker Spark MLServing dan scikit-learn yang menggunakan kembali transformator data yang dikembangkan untuk model pelatihan. Seluruh pipeline inferensi rakitan dapat dianggap sebagai SageMaker model yang dapat Anda gunakan untuk membuat prediksi real-time atau untuk memproses transformasi batch secara langsung tanpa preprocessing eksternal.

Dalam model pipeline inferensi, SageMaker menangani pemanggilan sebagai urutan permintaan HTTP. Wadah pertama dalam pipa menangani permintaan awal, maka respons menengah dikirim sebagai permintaan ke wadah kedua, dan seterusnya, untuk setiap wadah dalam pipa. SageMaker mengembalikan respons akhir ke klien.

Saat Anda menerapkan model pengaliran, SageMaker instal dan jalankan semua kontainer di setiap instans Amazon Elastic Compute Cloud (Amazon EC2) di tugas endpoint atau transformasi. Pemrosesan fitur dan inferensi berjalan dengan latensi rendah karena kontainer ditempatkan bersama pada instans EC2 yang sama. Anda menentukan kontainer untuk model pipeline menggunakan `CreateModel`operasi atau dari konsol. Alih-alih mengatur satu`PrimaryContainer`, Anda menggunakan`Containers` parameter. untuk mengatur wadah yang membentuk pipa Anda juga menentukan urutan di mana kontainer dieksekusi.

Model pipeline tidak dapat diubah, tetapi Anda dapat memperbarui pipeline inferensi dengan menerapkan yang baru menggunakan `UpdateEndpoint`operasi. Modularitas ini mendukung fleksibilitas yang lebih besar selama eksperimen.

Untuk informasi tentang cara membuat pipeline inferensi dengan registri SageMaker model, lihatDaftarkan dan Terapkan Model dengan Registri Model.

Tidak ada biaya tambahan untuk penggunaan fitur ini. Anda hanya membayar untuk instans yang berjalan di titik akhir.

Topik

- Contoh Notebook untuk Pipa Inferensi
- Pemrosesan Fitur dengan Spark Ml dan Scikit-Learn
- Buat Model Alur
- Jalankan Prediksi Real-time dengan Pipeline Inferensi

- Jalankan Transformasi Batch dengan Pipa Inferensi
- Log dan Metrik Pipa Inferensi
- Memecahkan Masalah Alur

Contoh Notebook untuk Pipa Inferensi

Untuk contoh yang menunjukkan cara membuat dan menerapkan pipeline inferensi, lihat buku catatan contoh Pipeline Inference dengan Scikit-learn dan Linear Learner. Untuk petunjuk tentang membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh SageMaker, lihatInstans SageMaker Notebook Amazon.

Untuk melihat daftar semua SageMaker sampel, setelah membuat dan membuka instance notebook, pilih tab SageMaker Examples. Ada tiga notebook pipa inferensi. Dua notebook pipa inferensi pertama yang baru saja dijelaskan terletak di`advanced_functionality` folder dan notebook ketiga ada di`sagemaker-python-sdk` folder. Untuk membuka buku catatan, pilih tab Gunakan, lalu pilih Buat salinan.

Pemrosesan Fitur dengan Spark Ml dan Scikit-Learn

Sebelum melatih model dengan algoritme SageMaker bawaan Amazon atau algoritme khusus, Anda dapat menggunakan preprosesor Spark dan scikit-learn untuk mengubah fitur data dan teknisi Anda.

Pengolahan Fitur dengan Spark Ml

Anda dapat menjalankan pekerjaan Spark MLdengan AWSGlue, layanan ETL (ekstrak, transformasi, muat) tanpa server, dari SageMaker notebook Anda. Anda juga dapat terhubung ke klaster EMR yang ada untuk menjalankan pekerjaan Spark MLdengan Amazon EMR. Untuk melakukan ini, Anda memerlukan peranAWS Identity and Access Management (IAM) yang memberikan izin untuk melakukan panggilan dari SageMaker buku catatan AndaAWS Glue.

> **Note**
>
> Untuk melihat versi Python dan Spark mana yangAWS Glue mendukung, lihat AWSGlue Release Notes.

Setelah fitur rekayasa, Anda mengemas dan membuat serial pekerjaan Spark dengan mLEAP ke dalam wadah mLEAP yang dapat Anda tambahkan ke pipeline inferensi. Anda tidak perlu

menggunakan klaster Spark yang dikelola secara eksternal. Dengan pendekatan ini, Anda dapat dengan mulus menskalakan dari sampel baris ke terabyte data. Trafo yang sama bekerja untuk pelatihan dan inferensi, jadi Anda tidak perlu menduplikasi preprocessing dan fitur logika rekayasa atau mengembangkan solusi satu kali untuk membuat model bertahan. Dengan pipeline inferensi, Anda tidak perlu mempertahankan infrastruktur luar, dan Anda dapat membuat prediksi langsung dari input data.

Saat Anda menjalankan pekerjaan Spark MLAWS Glue, pipeline Spark ML akan diserialkan ke dalam format mLEAP. Kemudian, Anda dapat menggunakan pekerjaan dengan SparkML Model Serving Container di Pipeline SageMaker Inference. mLeap adalah format serialisasi dan mesin eksekusi untuk pipa pembelajaran mesin. Ini mendukung Spark, Scikit-Learn, dan TensorFlow untuk jaringan pipa pelatihan dan mengekspornya ke pipa serial yang disebut Bundel MLEAP. Anda dapat deserialize Bundle kembali ke Spark untuk penilaian batch-mode atau ke runtime mLEAP untuk memberi daya pada layanan API real-time.

Untuk contoh yang menunjukkan cara menampilkan proses dengan Spark ML, lihat Melatih Model ML menggunakan Apache Spark di Amazon EMR dan terapkan di notebook SageMaker sampel.

Pemrosesan Fitur dengan Scikit-Learn

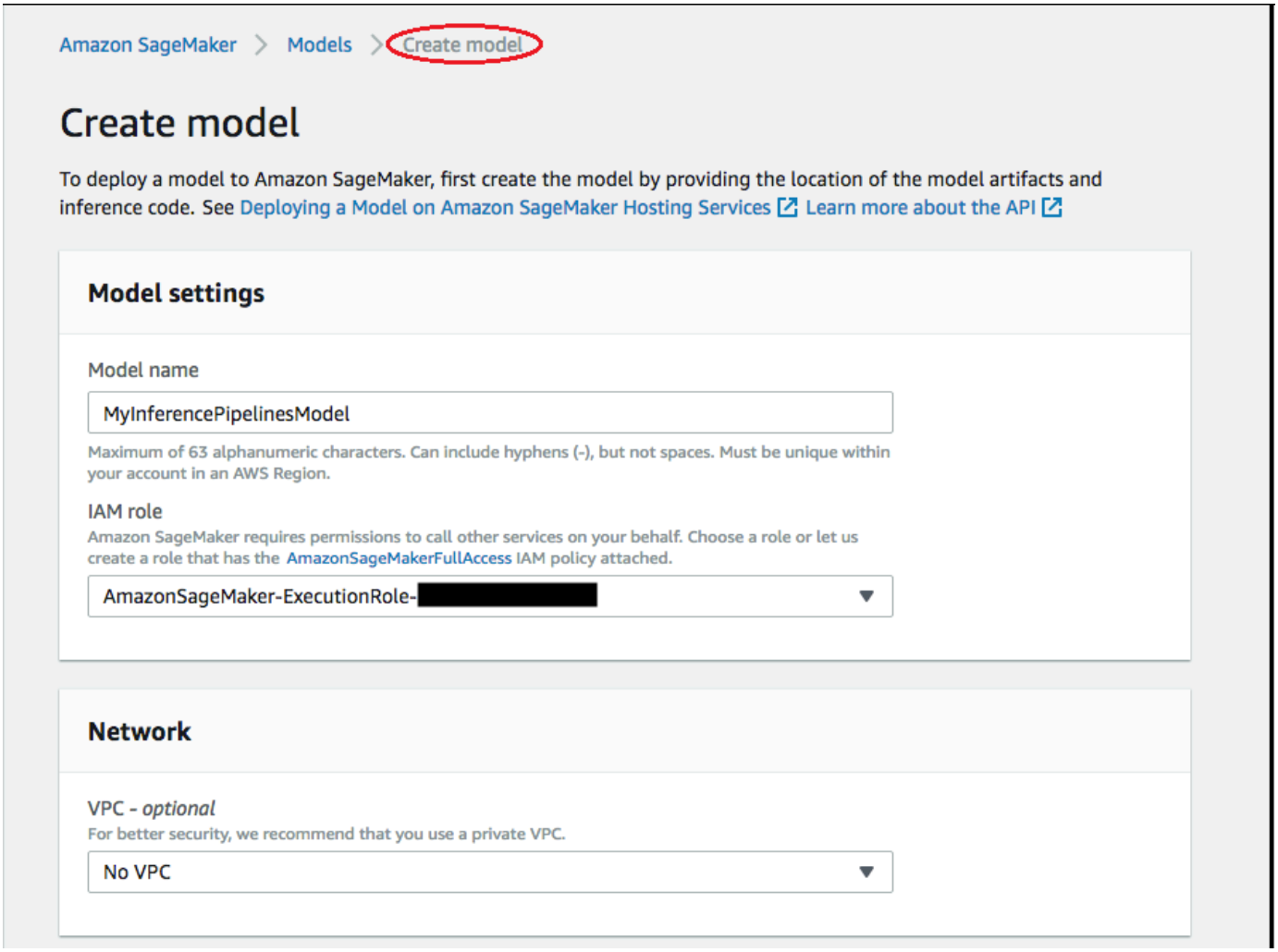
Anda dapat menjalankan dan mengemas pekerjaan scikit-learn ke dalam kontainer langsung di Amazon SageMaker. Untuk contoh kode Python untuk membangun model featurizer scikit-learn yang melatih kumpulan data bunga Iris Fisher dan memprediksi spesies Iris berdasarkan pengukuran morfologi, lihat Pelatihan dan Prediksi IRIS dengan Sagemaker Scikit-Learn.

Buat Model Alur

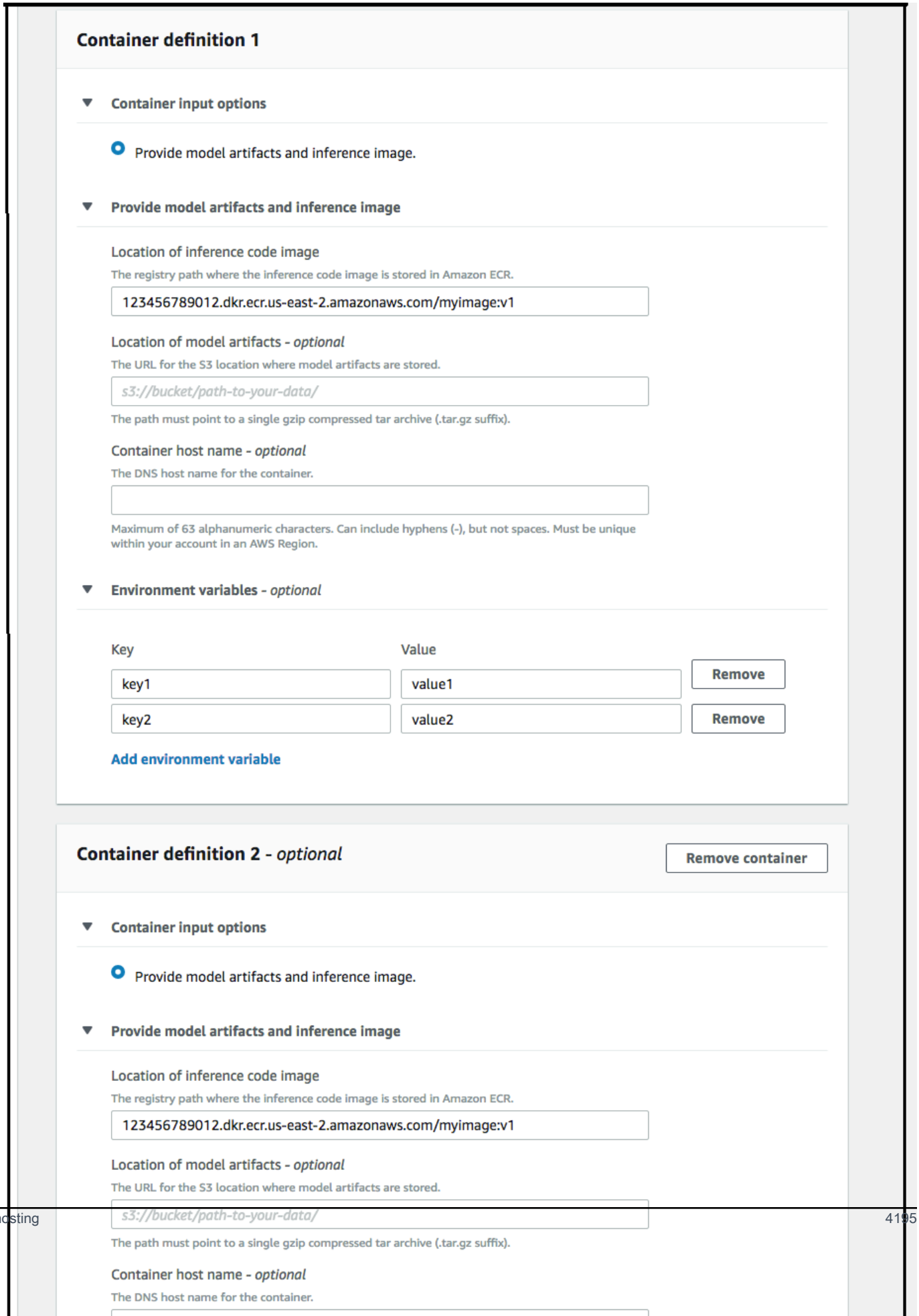
Untuk membuat model pipeline yang dapat diterapkan ke titik akhir atau digunakan untuk pekerjaan transformasi batch, gunakan SageMaker konsol Amazon atau`CreateModel` operasi.

Untuk membuat pipeline inferensi (konsol)

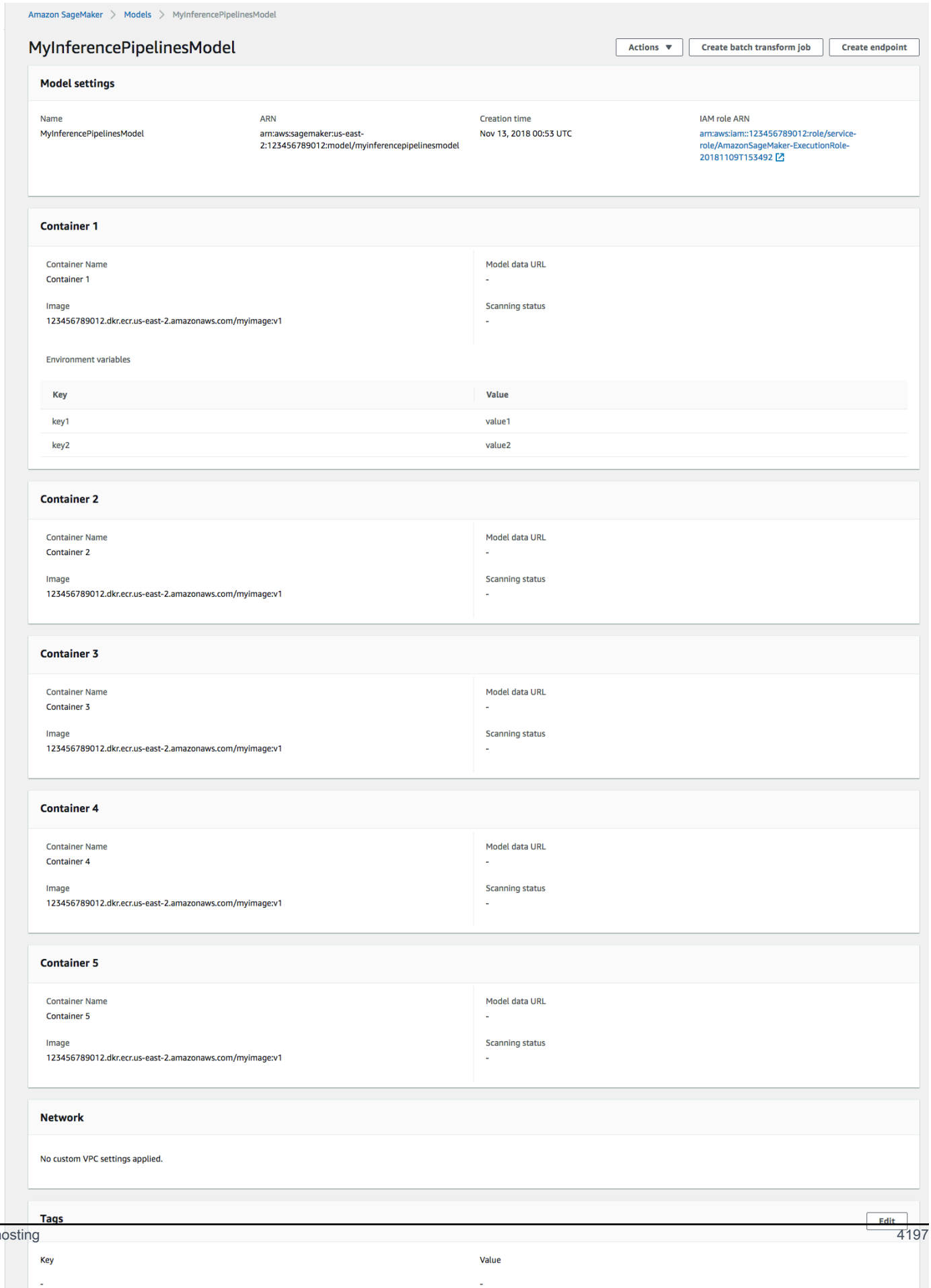
1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih Model, lalu pilih Buat model dari grup Inferensi.
3. Pada halaman Buat model, berikan nama model, pilih peran IAM, dan, jika Anda ingin menggunakan VPC pribadi, tentukan nilai VPC.



4. Untuk menambahkan informasi tentang kontainer di pipeline inferensi, pilih Tambahkan kontainer, lalu pilih Berikutnya.
5. Lengkapi bidang untuk setiap kontainer dalam urutan yang ingin Anda jalankan, hingga maksimum lima belas. Lengkapi opsi input Container,, Lokasi gambar kode inferensi, dan, secara opsional, Lokasi artefak model, nama host Container, dan bidang variabel Lingkungan.


Container definition 1
Container input options
Provide model artifacts and inference image.
Provide model artifacts and inference image
Location of inference code image
The registry path where the inference code image is stored in Amazon ECR.
123456789012.dkr.ecr.us-east-2.amazonaws.com/myimage:v1
Location of model artifacts - optional
The URL for the S3 location where model artifacts are stored.
s3://bucket/path-to-your-data/
The path must point to a single gzip compressed tar archive (.tar.gz suffix).
Container host name - optional
The DNS host name for the container.
Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
Environment variables - optional
Key
Value
key1
value1
Remove
key2
value2
Remove
Add environment variable
Container definition 2 - optional
Remove container
Container input options
Provide model artifacts and inference image.
Provide model artifacts and inference image
Location of inference code image
The registry path where the inference code image is stored in Amazon ECR.
123456789012.dkr.ecr.us-east-2.amazonaws.com/myimage:v1
Location of model artifacts - optional
The URL for the S3 location where model artifacts are stored.
s3://bucket/path-to-your-data/
The path must point to a single gzip compressed tar archive (.tar.gz suffix).
Container host name - optional
The DNS host name for the container.

MyInferencePipelineModelHalaman merangkum pengaturan untuk kontainer yang menyediakan masukan untuk model. Jika Anda memberikan variabel lingkungan dalam definisi kontainer yang sesuai, SageMaker tunjukkan mereka di bidang variabel Lingkungan.


Amazon SageMaker > Models > MyInferencePipelinesModel
MyInferencePipelinesModel
Actions
Create batch transform job
Create endpoint
Model settings
Name
MyInferencePipelinesModel
ARN
arn:aws:sagemaker:us-east-2:123456789012:model/myinferencepipelinesmodel
Creation time
Nov 13, 2018 00:53 UTC
IAM role ARN
arn:aws:iam::123456789012:role/service-role/AmazonSageMaker-ExecutionRole-20181109T153492
Container 1
Container Name
Container 1
Image
123456789012.dkr.ecr.us-east-2.amazonaws.com/myimage:v1
Model data URL
-
Scanning status
-
Environment variables
Key
Value
key1
value1
key2
value2
Container 2
Container Name
Container 2
Image
123456789012.dkr.ecr.us-east-2.amazonaws.com/myimage:v1
Model data URL
-
Scanning status
-
Container 3
Container Name
Container 3
Image
123456789012.dkr.ecr.us-east-2.amazonaws.com/myimage:v1
Model data URL
-
Scanning status
-
Container 4
Container Name
Container 4
Image
123456789012.dkr.ecr.us-east-2.amazonaws.com/myimage:v1
Model data URL
-
Scanning status
-
Container 5
Container Name
Container 5
Image
123456789012.dkr.ecr.us-east-2.amazonaws.com/myimage:v1
Model data URL
-
Scanning status
-
Network
No custom VPC settings applied.
Tags
Edit
Key
Value
-
-

Jalankan Prediksi Real-time dengan Pipeline Inferensi

Anda dapat menggunakan model terlatih dalam pipeline inferensi untuk membuat prediksi real-time secara langsung tanpa melakukan preprocessing eksternal. Saat mengonfigurasi pipeline, Anda dapat memilih untuk menggunakan trafo fitur bawaan yang sudah tersedia di Amazon SageMaker. Atau, Anda dapat menerapkan logika transformasi Anda sendiri hanya menggunakan beberapa baris scikit-learn atau kode Spark.

mLeap, format serialisasi dan mesin eksekusi untuk jaringan pipa pembelajaran mesin, mendukung Spark, scikit-learn, dan TensorFlow untuk melatih jaringan pipa dan mengekspornya ke pipeline serial yang disebut Bundle mLeap. Anda dapat deserialize Bundle kembali ke Spark untuk penilaian batch-mode atau ke runtime mLEAP untuk memberi daya pada layanan API real-time.

Kontainer dalam pipeline mendengarkan pada port yang ditentukan dalam variabel`SAGEMAKER_BIND_TO_PORT` lingkungan (bukan 8080). Saat berjalan di pipeline inferensi, SageMaker secara otomatis menyediakan variabel lingkungan ini ke kontainer. Jika variabel lingkungan ini tidak ada, kontainer default menggunakan port 8080. Untuk menunjukkan bahwa kontainer Anda mematuhi persyaratan ini, gunakan perintah berikut untuk menambahkan label ke Dockerfile Anda:

```
LABEL com.amazonaws.sagemaker.capabilities.accept-bind-to-port=true
```

Jika kontainer Anda perlu mendengarkan pada port kedua, pilih port dalam kisaran yang ditentukan oleh variabel`SAGEMAKER_SAFE_PORT_RANGE` lingkungan. Tentukan nilai sebagai rentang inklusif dalam format**`"XXXX-YYYY"`**, di mana`XXXX` dan`YYYY` merupakan bilangan bulat multi-digit. SageMaker memberikan nilai ini secara otomatis ketika Anda menjalankan kontainer dalam pipeline multicontainer.

> **Note**
>
> Untuk menggunakan image Docker kustom dalam pipeline yang menyertakan algoritmeSageMaker bawaan, Anda memerlukan kebijakan Amazon Elastic Container Registry (Amazon ECR). Repositori Amazon ECR Anda harus memberikan SageMaker izin untuk menarik gambar. Untuk informasi selengkapnya, lihat Memecahkan Masalah Izin Amazon ECR untuk Saluran Pipa Inferensi.

Membuat dan Menerapkan Endpoint Pipeline Inferensi

Kode berikut membuat dan menerapkan model pipeline inferensi waktu nyata dengan model SparkML dan XGBoost secara seri menggunakan SageMaker SDK.

```
from sagemaker.model import Model
from sagemaker.pipeline_model import PipelineModel
from sagemaker.sparkml.model import SparkMLModel

sparkml_data = 's3://{}/{}/{}'.format(s3_model_bucket, s3_model_key_prefix,
 'model.tar.gz')
sparkml_model = SparkMLModel(model_data=sparkml_data)
xgb_model = Model(model_data=xgb_model.model_data, image=training_image)

model_name = 'serial-inference-' + timestamp_prefix
endpoint_name = 'serial-inference-ep-' + timestamp_prefix
sm_model = PipelineModel(name=model_name, role=role, models=[sparkml_model, xgb_model])
sm_model.deploy(initial_instance_count=1, instance_type='ml.c4.xlarge',
 endpoint_name=endpoint_name)
```

Meminta Inferensi Real-Time dari Endpoint Pipa Inferensi

Contoh berikut menunjukkan cara membuat prediksi real-time dengan memanggil titik akhir inferensi dan meneruskan payload permintaan dalam format JSON:

```
import sagemaker
from sagemaker.predictor import json_serializer, json_deserializer, Predictor

payload = {
        "input": [
            {
                "name": "Pclass",
                "type": "float",
                "val": "1.0"
            },
            {
                "name": "Embarked",
                "type": "string",
                "val": "Q"
            },
            {
                "name": "Age",
                "type": "double",
```

```
                "val": "48.0"
            },
            {
                "name": "Fare",
                "type": "double",
                "val": "100.67"
            },
            {
                "name": "SibSp",
                "type": "double",
                "val": "1.0"
            },
            {
                "name": "Sex",
                "type": "string",
                "val": "male"
            }
        ],
        "output": {
            "name": "features",
            "type": "double",
            "struct": "vector"
        }
    }

predictor = Predictor(endpoint=endpoint_name, sagemaker_session=sagemaker.Session(),
 serializer=json_serializer,
                                content_type='text/csv', accept='application/json'

print(predictor.predict(payload))
```

Respon yang Anda dapatkan`predictor.predict(payload)` adalah hasil inferensi model.

Contoh pipa inferensi waktu nyata

Anda dapat menjalankan contoh notebook ini menggunakan prediktor SKLearn yang menunjukkan cara menerapkan titik akhir, menjalankan permintaan inferensi, lalu deserialisasi respons. Temukan buku catatan ini dan lebih banyak contoh di GitHub repositori SageMaker contoh Amazon.

Jalankan Transformasi Batch dengan Pipa Inferensi

Untuk mendapatkan kesimpulan pada seluruh dataset Anda menjalankan batch transform pada model terlatih. Untuk menjalankan inferensi pada kumpulan data lengkap, Anda dapat menggunakan

model pipeline inferensi yang sama yang dibuat dan diterapkan ke titik akhir untuk pemrosesan waktu nyata dalam pekerjaan transformasi batch. Untuk menjalankan tugas transformasi batch dalam pipeline, Anda mengunduh data input dari Amazon S3 dan mengirimkannya dalam satu atau beberapa permintaan HTTP ke model pipeline inferensi. Untuk contoh yang menunjukkan cara menyiapkan data untuk transformasi batch, lihat “Bagian 2 - Memproses data perumahan mentah menggunakan Scikit Learn” dari Endpoint SageMaker Multi-Model Amazon menggunakan notebook sampel Linear Learner. Untuk informasi tentang transformasi SageMaker batch Amazon, lihatGunakan Batch Transform.

> **Note**
>
> Untuk menggunakan image Docker kustom dalam pipeline yang menyertakan algoritme SageMaker bawaan Amazon, Anda memerlukan kebijakan Amazon Elastic Container Registry (ECR). Repositori Amazon ECR Anda harus memberikan SageMaker izin untuk menarik gambar. Untuk informasi selengkapnya, lihat Memecahkan Masalah Izin Amazon ECR untuk Saluran Pipa Inferensi.

Contoh berikut menunjukkan cara menjalankan tugas transformasi menggunakan Amazon SageMaker Python SDK. Dalam contoh ini,`model_name` adalah pipeline inferensi yang menggabungkan model SparkML dan XGBoost (dibuat dalam contoh sebelumnya). Lokasi Amazon S3 yang ditentukan oleh`input_data_path` berisi data masukan, dalam format CSV, untuk diunduh dan dikirim ke model Spark ML. Setelah pekerjaan transformasi selesai, lokasi Amazon S3 yang ditentukan oleh`output_data_path` berisi data keluaran yang dikembalikan oleh model XGBoost dalam format CSV.

```
import sagemaker
input_data_path = 's3://{}/{}/{}'.format(default_bucket, 'key', 'file_name')
output_data_path = 's3://{}/{}'.format(default_bucket, 'key')
transform_job = sagemaker.transformer.Transformer(
    model_name = model_name,
    instance_count = 1,
    instance_type = 'ml.m4.xlarge',
    strategy = 'SingleRecord',
    assemble_with = 'Line',
    output_path = output_data_path,
    base_transform_job_name='inference-pipelines-batch',
    sagemaker_session=sagemaker.Session(),
    accept = CONTENT_TYPE_CSV)
```

```
transform_job.transform(data = input_data_path,
                        content_type = CONTENT_TYPE_CSV,
                        split_type = 'Line')
```

Log dan Metrik Pipa Inferensi

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa SageMaker sumber daya Amazon. Untuk memantau dan memecahkan masalah kinerja pipeline inferensi, gunakan CloudWatch log Amazon dan pesan kesalahan. Untuk informasi tentang alat pemantauan yang SageMaker menyediakan, lihatMemantau AWS sumber daya yang disediakan saat menggunakan Amazon SageMaker.

Menggunakan Metrik untuk Memantau Model Multi-kontainer

Untuk memantau model multi-kontainer di Inference Pipelines, gunakan Amazon CloudWatch. CloudWatchmengumpulkan dan memproses data mentah, menjadi metrik hampir waktu nyata yang dapat dibaca. SageMakerpekerjaan pelatihan dan titik akhir menulis CloudWatch metrik dan log di`AWS/SageMaker` ruang nama.

Tabel berikut ini mencantumkan metrik dan dimensi untuk yang berikut:

- Doa titik akhir
- Pekerjaan pelatihan, pekerjaan transformasi batch, dan instans titik akhir

Dimensi adalah pasangan nama/nilai yang merupakan bagian dari identitas metrik. Anda dapat menetapkan hingga 10 dimensi ke metrik. Untuk informasi lebih lanjut tentang pemantauan dengan CloudWatch, lihatPantau Amazon SageMaker dengan Amazon CloudWatch.

Metrik Pemanggilan Titik Akhir

`AWS/SageMaker`Namespace menyertakan metrik permintaan berikut dari panggilan ke `InvokeEndpoint`.

Metrik dilaporkan dengan interval 1 menit.

Metrik	Deskripsi
`Invocation4XXErrors`	Jumlah`InvokeEndpoint` permintaan bahwa model mengembalikan kode respon`4xx` HTTP untuk. Untuk setiap`4xx` respon, SageMaker mengirimkan`1`.

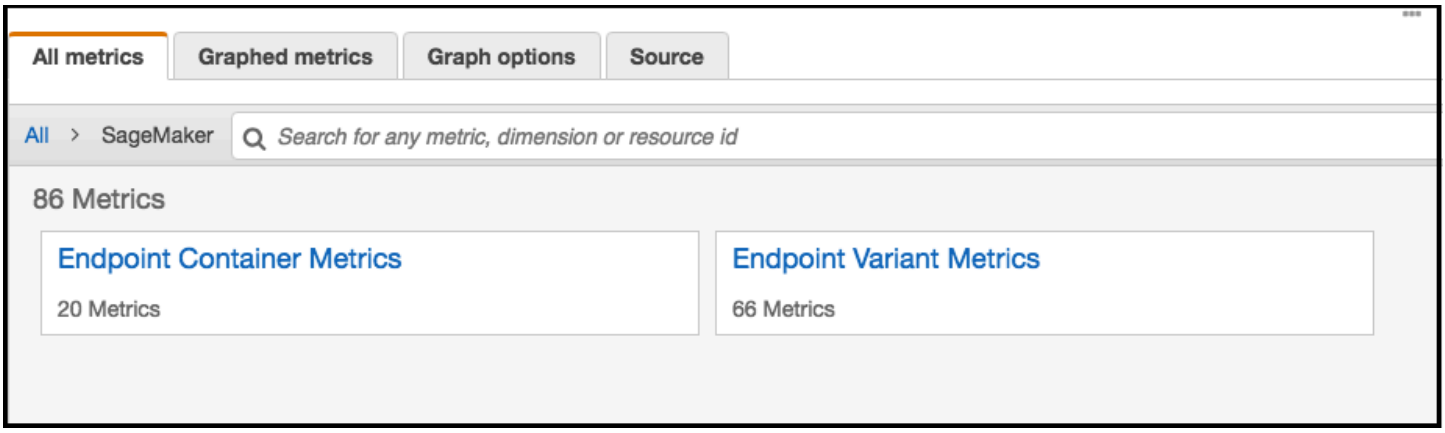
Metrik	Deskripsi
	Unit: Tidak ada Statistik yang valid: `Average`, `Sum`
`Invocation5XXErrors`	Jumlah`InvokeEndpoint` permintaan bahwa model mengembalikan kode respon`5xx` HTTP untuk. Untuk setiap`5xx` respon, SageMaker mengirimkan`1`. Unit: Tidak ada Statistik yang valid: `Average`, `Sum`
`Invocations`	`number of InvokeEndpoint` Permintaan dikirim ke endpoint model. Untuk mendapatkan jumlah total permintaan yang dikirim ke titik akhir model, gunakan`Sum` statistik. Unit: Tidak ada Statistik yang valid: `Sum`, `Sample Count`
`InvocationsPerInstance`	Jumlah doa endpoint dikirim ke model, dinormalisasi oleh`InstanceCount` di masing-masing`ProductionVariant` . SageMakermengirimkan 1/`numberOfInstances` sebagai nilai untuk setiap permintaan, di mana`numberOfInstances` adalah jumlah contoh aktif untuk ProductionVariant pada titik akhir pada saat permintaan. Unit: Tidak ada Statistik valid: `Sum`

Metrik	Deskripsi
`ModelLatency`	Waktu yang dibutuhkan model atau model untuk merespons. Ini termasuk waktu yang dibutuhkan untuk mengirim permintaan, untuk mengambil respons dari wadah model, dan untuk menyelesaikan inferensi dalam wadah. `ModelLatency` adalah total waktu yang diambil oleh semua kontainer dalam pipa inferensi. Unit: Mikrodetik Statistik yang valid:`Average`,`Sum`,`Min`,`Max`, Jumlah Sampel
`OverheadLatency`	Waktu ditambahkan ke waktu yang dibutuhkan untuk menanggapi permintaan klien dengan SageMaker biaya overhead. `OverheadLatency` diukur dari waktu yang SageMaker menerima permintaan sampai mengembalikan respon ke klien, dikurangi`ModelLatency` . Latensi overhead dapat bervariasi tergantung pada ukuran payload permintaan dan respons, frekuensi permintaan, dan otentikasi atau otorisasi permintaan, di antara faktor-faktor lainnya. Unit: Mikrodetik Statistik yang valid:`Average`,`Sum`,`Min`,`Max`,`Sample Count`
`Container Latency`	Waktu yang dibutuhkan untuk wadah Inference Pipelines untuk merespons sebagaimana dilihat dari SageMaker. `Container Latency` mencakup waktu yang dibutuhkan untuk mengirim permintaan, untuk mengambil respons dari wadah model, dan untuk menyelesaikan inferensi dalam wadah. Unit: Mikrodetik Statistik yang valid:`Average`,`Sum`,`Min`,`Max`,`Sample Count`

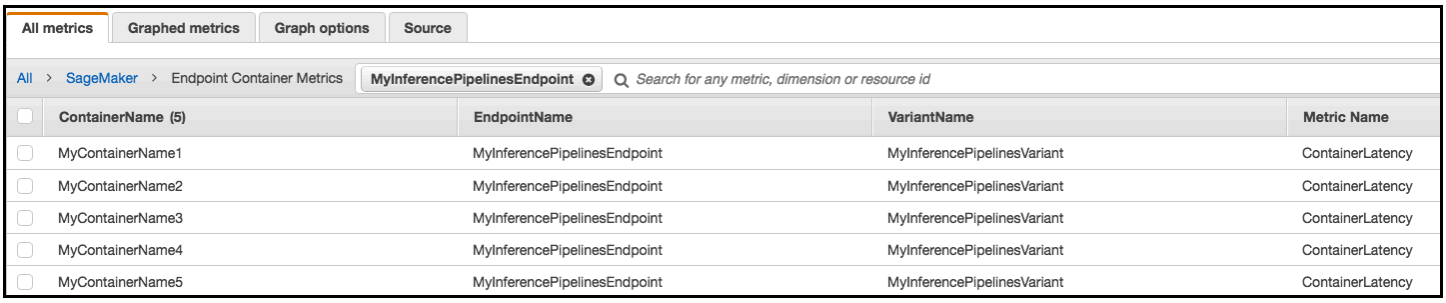
Dimensi untuk Metrik Pemanggilan Titik Akhir

Dimensi	Deskripsi
`EndpointName`, `VariantName`, `ContainerName`	Memfilter metrik pemanggilan titik akhir untuk titik akhir yang ditentukan dan untuk varian yang ditentukan.`ProductionVariant`

Untuk titik akhir pipeline inferensi, CloudWatch cantumkan metrik latensi per kontainer di akun Anda sebagai Metrik Kontainer Titik Akhir dan Metrik Varian Titik Akhir di SageMakerruang nama, sebagai berikut. `ContainerLatency`Metrik muncul hanya untuk pipa kesimpulan.



Untuk setiap titik akhir dan setiap kontainer, metrik latensi menampilkan nama untuk wadah, titik akhir, varian, dan metrik.



Job Pelatihan, Job Transformasi Batch, dan Metrik Instans Titik Akhir

Ruang nama`/aws/sagemaker/TrainingJobs`,`/aws/sagemaker/TransformJobs`, dan`/aws/sagemaker/Endpoints` menyertakan metrik berikut untuk pekerjaan pelatihan dan instans titik akhir.

Metrik dilaporkan dengan interval 1 menit.

Metrik	Deskripsi
`CPUUtilization`	Persentase unit CPU yang digunakan oleh kontainer yang berjalan pada instans. Nilai berkisar dari 0% sampai 100%, dan dikalikan dengan jumlah CPU. Misalnya, jika ada empat CPU,`CPUUtilization` dapat berkisar dari 0% hingga 400%. Untuk pekerjaan pelatihan,`CPUUtilization` adalah pemanfaatan CPU dari wadah algoritma yang berjalan pada instance. Untuk pekerjaan batch transform,`CPUUtilization` adalah pemanfaat an CPU dari kontainer transformasi yang berjalan pada instance. Untuk model multi-kontainer,`CPUUtilization` adalah jumlah pemanfaatan CPU oleh semua kontainer yang berjalan pada instance. Untuk varian endpoint,`CPUUtilization` adalah jumlah pemanfaatan CPU oleh semua kontainer yang berjalan pada instance. Unit: Persen
`MemoryUti lization`	Persentase memori yang digunakan oleh kontainer yang berjalan pada instans. Nilai ini berkisar dari 0% hingga 100%. Untuk pekerjaan pelatihan,`MemoryUtilization` adalah memori yang digunakan oleh wadah algoritma yang berjalan pada instance. Untuk pekerjaan batch transform,`MemoryUtilization` adalah memori yang digunakan oleh kontainer transformasi yang berjalan pada instance. Untuk model multi-kontainer,`MemoryUtilization` adalah jumlah memori yang digunakan oleh semua kontainer yang berjalan pada instance. Untuk varian endpoint,`MemoryUtilization` adalah jumlah memori yang digunakan oleh semua kontainer yang berjalan pada instance. Unit: Persen

Metrik	Deskripsi
`GPUUtilization`	Persentase unit GPU yang digunakan oleh kontainer yang berjalan pada sebuah instance. `GPUUtilization` berkisar dari 0% sampai 100% dan dikalikan dengan jumlah GPU. Misalnya, jika ada empat GPU,`GPUUtilization` dapat berkisar dari 0% hingga 400%. Untuk pekerjaan pelatihan,`GPUUtilization` adalah GPU yang digunakan oleh wadah algoritma yang berjalan pada instance. Untuk pekerjaan batch transform,`GPUUtilization` adalah GPU yang digunakan oleh kontainer transformasi yang berjalan pada instance. Untuk model multi-kontainer,`GPUUtilization` adalah jumlah GPU yang digunakan oleh semua kontainer yang berjalan pada instance. Untuk varian endpoint,`GPUUtilization` adalah jumlah GPU yang digunakan oleh semua kontainer yang berjalan pada instance. Unit: Persen

Metrik	Deskripsi
`GPUMemoryUtilization`	Persentase memori GPU yang digunakan oleh kontainer yang berjalan pada sebuah instance. GPUMemoryUtilization berkisar antara 0% sampai 100% dan dikalikan dengan jumlah GPU. Misalnya, jika ada empat GPU,`GPUMemoryUtilization` dapat berkisar dari 0% hingga 400%. Untuk pekerjaan pelatihan,`GPUMemoryUtilization` adalah memori GPU yang digunakan oleh wadah algoritma yang berjalan pada instance. Untuk pekerjaan batch transform,`GPUMemoryUtilization` adalah memori GPU yang digunakan oleh kontainer transformasi yang berjalan pada instance. Untuk model multi-kontainer,`GPUMemoryUtilization` adalah jumlah GPU yang digunakan oleh semua kontainer yang berjalan pada instance. Untuk varian endpoint,`GPUMemoryUtilization` adalah jumlah memori GPU yang digunakan oleh semua kontainer yang berjalan pada instance. Unit: Persen
`DiskUtilization`	Persentase ruang disk yang digunakan oleh kontainer yang berjalan pada instans. DiskUtilization berkisar dari 0% sampai 100%. Metrik ini tidak didukung untuk tugas transformasi batch. Untuk pekerjaan pelatihan,`DiskUtilization` adalah ruang disk yang digunakan oleh wadah algoritma yang berjalan pada instance. Untuk varian endpoint,`DiskUtilization` adalah jumlah ruang disk yang digunakan oleh semua kontainer yang disediakan yang berjalan pada instance. Unit: Persen

Dimensi untuk Job Pelatihan, Job Transformasi Batch, dan Metrik Instans Titik Akhir

Dimensi	Deskripsi
`Host`	Untuk pekerjaan pelatihan,`Host` memiliki format`[training-job-name]/algo-[instance-number-in-cluster]` . Gunakan dimensi ini untuk memfilter metrik instans untuk pekerjaan pelatihan dan instance yang ditentukan. Format dimensi ini hanya ada di`/aws/sagemaker/TrainingJobs` namespace. Untuk batch mengubah pekerjaan,`Host` memiliki format`[transform-job-name]/[instance-id]` . Gunakan dimensi ini untuk memfilter metrik instans untuk pekerjaan dan instans transformasi batch yang ditentukan. Format dimensi ini hanya ada di`/aws/sagemaker/TransformJobs` namespace. Untuk endpoint,`Host` memiliki format`[endpoint-name]/[ production-variant-name ]/[instance-id]` . Gunakan dimensi ini untuk memfilter metrik instans untuk titik akhir, varian, dan instance yang ditentukan. Format dimensi ini hanya ada di`/aws/sagemaker/Endpoints` namespace.

Untuk membantu Anda men-debug konfigurasi siklus hidup pekerjaan latihan, titik akhir, dan instance notebook, SageMaker juga mengirimkan kontainer algoritme, wadah model, atau konfigurasi siklus hidup instans notebook yang dikirim ke`stdout` atau`stderr` ke Amazon CloudWatch Logs. Anda dapat menggunakan informasi ini untuk debugging dan menganalisis kemajuan.

Gunakan Log untuk Memantau Pipeline Inferensi

Tabel berikut mencantumkan grup log dan aliran log SageMaker. Kirim ke Amazon CloudWatch

Pengaliran log adalah urutan log acara yang berbagi sumber yang sama. Setiap sumber log terpisah, CloudWatch membentuk pengaliran log terpisah. Grup log adalah grup pengaliran log yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama.

Log

Catat Nama Grup	Nama Pengaliran
`/aws/sagemaker/TrainingJobs`	`[training-job-name]/algo-[instance-number-in-cluster]-[epoch_timestamp]`
`/aws/sagemaker/Endpoints/[EndpointName]`	`[production-variant-name]/[instance-id]`
	`[production-variant-name]/[instance-id]`
	`[production-variant-name]/[instance-id]/[container-name provided in the SageMaker model] (For Inference Pipelines)` Untuk log Inference Pipelines, jika Anda tidak memberikan nama kontainer, CloudWatch gunakan**container-1, container-2**, dan seterusnya, dalam urutan kontainer disediakan dalam model.
`/aws/sagemaker/NotebookInstances`	`[notebook-instance-name]/[LifecycleConfigHook]`
`/aws/sagemaker/TransformJobs`	`[transform-job-name]/[instance-id]-[epoch_timestamp]`
	`[transform-job-name]/[instance-id]-[epoch_timestamp]/data-log`
	`[transform-job-name]/[instance-id]-[epoch_timestamp]/[container-name provided in the SageMaker model] (For Inference Pipelines)` Untuk log Inference Pipelines, jika Anda tidak memberikan nama kontainer, CloudWatch gunakan**container-1, container-2**, dan seterusnya, dalam urutan kontainer disediakan dalam model.

> **Note**
>
> SageMakermembuat grup`/aws/sagemaker/NotebookInstances` log saat Anda membuat instance notebook dengan konfigurasi siklus hidup. Untuk informasi selengkapnya, lihat Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup.

Untuk informasi lebih lanjut tentang SageMaker catatan, lihatLog SageMaker Acara Amazon dengan Amazon CloudWatch.

Memecahkan Masalah Alur

Untuk memecahkan masalah saluran inferensi, gunakan CloudWatch log dan pesan kesalahan. Jika Anda menggunakan image Docker kustom dalam pipeline yang menyertakan algoritme SageMaker bawaan Amazon, Anda mungkin juga mengalami masalah izin. Untuk memberikan izin yang diperlukan, buat kebijakan Amazon Elastic Container Registry (Amazon ECR).

Topik

- Memecahkan Masalah Izin Amazon ECR untuk Saluran Pipa Inferensi
- Gunakan CloudWatch Log untuk Memecahkan Masalah Pipa SageMaker Inferensi
- Menggunakan Pesan Kesalahan untuk Memecahkan Masalah Pipeline Inferensi

Memecahkan Masalah Izin Amazon ECR untuk Saluran Pipa Inferensi

Saat Anda menggunakan image Docker khusus dalam pipeline yang menyertakan algoritmeSageMaker bawaan, Anda memerlukan kebijakan Amazon ECR. Kebijakan ini memungkinkan repositori Amazon ECR Anda memberikan izin SageMaker untuk menarik gambar. Kebijakan harus menambahkan izin berikut:

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "allowSageMakerToPull",
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker.amazonaws.com"
            },
```

```
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ]
        }
    ]
}
```

Gunakan CloudWatch Log untuk Memecahkan Masalah Pipa SageMaker Inferensi

SageMaker menerbitkan log kontainer untuk titik akhir yang menyebarkan pipeline inferensi ke Amazon CloudWatch di jalur berikut untuk setiap kontainer.

```
/aws/sagemaker/Endpoints/{EndpointName}/{Variant}/{InstanceId}/{ContainerHostname}
```

Misalnya, log untuk titik akhir ini dipublikasikan ke grup dan aliran log berikut:

```
EndpointName: MyInferencePipelinesEndpoint
Variant: MyInferencePipelinesVariant
InstanceId: i-0179208609ff7e488
ContainerHostname: MyContainerName1 and MyContainerName2
```

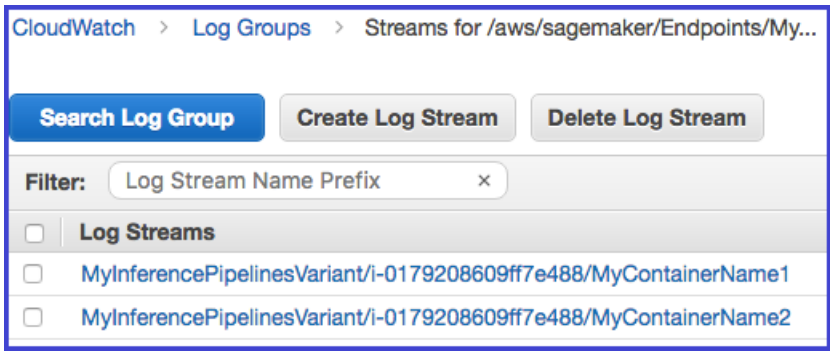
```
logGroup: /aws/sagemaker/Endpoints/MyInferencePipelinesEndpoint
logStream: MyInferencePipelinesVariant/i-0179208609ff7e488/MyContainerName1
logStream: MyInferencePipelinesVariant/i-0179208609ff7e488/MyContainerName2
```

Pengaliran log adalah urutan log acara yang berbagi sumber yang sama. Setiap sumber log terpisah, CloudWatch membentuk pengaliran log terpisah. Grup log adalah grup pengaliran log yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama.

Untuk melihat grup log dan aliran

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di halaman navigasi, pilih Log.
3. Di Grup Log. filter pada**`MyInferencePipelinesEndpoint`**:

4. Untuk melihat aliran log, pada halaman Grup CloudWatch Log, pilih**`MyInferencePipelinesEndpoint`**, lalu Cari Grup Log.



Untuk daftar log yang SageMaker diterbitkan, lihatLog dan Metrik Pipa Inferensi.

Menggunakan Pesan Kesalahan untuk Memecahkan Masalah Pipeline Inferensi

Pesan kesalahan pipeline inferensi menunjukkan kontainer mana yang gagal.

Jika terjadi kesalahan saat SageMaker menjalankan titik akhir, layanan akan mengembalikan`ModelError` (kode kesalahan 424), yang menunjukkan kontainer mana yang gagal. Jika payload permintaan (respons dari kontainer sebelumnya) melebihi batas 5 MB, SageMaker memberikan pesan kesalahan rinci, seperti:

Menerima tanggapan dari MyContainerName 1 dengan kode status 200. Namun, payload permintaan dari MyContainerName 1 hingga MyContainerName 2 adalah 6000000 byte, yang telah melampaui batas maksimum 5 MB.

Jika wadah gagal pemeriksaan kesehatan ping saat SageMaker membuat titik akhir, ia mengembalikan`ClientError` dan menunjukkan semua kontainer yang gagal pemeriksaan ping di pemeriksaan kesehatan terakhir.

Hapus Titik Akhir dan Sumber Daya

Hapus titik akhir untuk menghentikan biaya yang dikenakan.

Hapus Endpoint

Hapus titik akhir Anda secara terprogram menggunakanAWS SDK for Python (Boto3), denganAWS CLI, atau secara interaktif menggunakan konsol. SageMaker

SageMaker membebaskan semua sumber daya yang digunakan saat titik akhir dibuat. Menghapus titik akhir tidak akan menghapus konfigurasi titik akhir atau model. SageMaker Lihat Hapus Konfigurasi Titik Akhir dan Hapus Model untuk informasi tentang cara menghapus konfigurasi dan SageMaker model titik akhir Anda.

AWS SDK for Python (Boto3)

Gunakan `DeleteEndpoint`API untuk menghapus titik akhir Anda. Tentukan nama titik akhir Anda untuk `EndpointName` bidang tersebut.

```
import boto3

# Specify your AWS Region
aws_region='<aws_region>'

# Specify the name of your endpoint
endpoint_name='<endpoint_name>'

# Create a low-level SageMaker service client.
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Delete endpoint
sagemaker_client.delete_endpoint(EndpointName=endpoint_name)
```

AWS CLI

Gunakan `delete-endpoint`perintah untuk menghapus titik akhir Anda. Tentukan nama titik akhir Anda untuk `endpoint-name` bendera.

```
aws sagemaker delete-endpoint --endpoint-name <endpoint-name>
```

SageMaker Console

Hapus titik akhir Anda secara interaktif dengan konsol. SageMaker

1. Di SageMaker konsol di menu navigasi https://console.aws.amazon.com/sagemaker/, pilih Inferensi.
2. Pilih Endpoints dari menu drop-down. Daftar titik akhir yang dibuat di AWS akun Anda akan muncul berdasarkan nama, Nama Sumber Daya Amazon (ARN), waktu pembuatan, status, dan cap waktu kapan titik akhir terakhir diperbarui.

3. Pilih titik akhir yang ingin Anda hapus.
4. Pilih tombol dropdown Actions di pojok kanan atas.
5. Pilih Hapus.

Hapus Konfigurasi Titik Akhir

Hapus konfigurasi titik akhir Anda secara terprogram menggunakanAWS SDK for Python (Boto3), denganAWS CLI, atau secara interaktif menggunakan konsol. SageMaker Menghapus konfigurasi titik akhir tidak menghapus titik akhir yang dibuat menggunakan konfigurasi ini. Lihat Hapus Endpoint untuk informasi tentang cara menghapus titik akhir Anda.

Jangan menghapus konfigurasi titik akhir yang digunakan oleh titik akhir yang aktif atau saat titik akhir sedang diperbarui atau dibuat. Anda mungkin kehilangan visibilitas ke jenis instans yang digunakan titik akhir jika Anda menghapus konfigurasi titik akhir dari titik akhir yang aktif atau sedang dibuat atau diperbarui.

AWS SDK for Python (Boto3)

Gunakan `DeleteEndpointConfig`API untuk menghapus titik akhir Anda. Tentukan nama konfigurasi titik akhir Anda untuk `EndpointConfigName` bidang tersebut.

```
import boto3

# Specify your AWS Region
aws_region='<aws_region>'

# Specify the name of your endpoint configuration
endpoint_config_name='<endpoint_name>'

# Create a low-level SageMaker service client.
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Delete endpoint configuration
sagemaker_client.delete_endpoint_config(EndpointConfigName=endpoint_config_name)
```

Anda dapat menggunakan `DescribeEndpointConfig`API secara opsional untuk menampilkan informasi tentang nama model yang Anda gunakan (varian produksi) seperti nama model Anda dan nama konfigurasi titik akhir yang terkait dengan model yang diterapkan tersebut. Berikan nama titik akhir Anda untuk `EndpointConfigName` bidang tersebut.

```
# Specify the name of your endpoint
endpoint_name='<endpoint_name>'

# Create a low-level SageMaker service client.
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Store DescribeEndpointConfig response into a variable that we can index in the
 next step.
response =
 sagemaker_client.describe_endpoint_config(EndpointConfigName=endpoint_name)

# Delete endpoint
endpoint_config_name = response['ProductionVariants'][0]['EndpointConfigName']

# Delete endpoint configuration
sagemaker_client.delete_endpoint_config(EndpointConfigName=endpoint_config_name)
```

Untuk informasi selengkapnya tentang elemen respons lain yang ditampilkan oleh`DescribeEndpointConfig`, lihat [DescribeEndpointConfig](#)di [panduan Referensi SageMaker API](#).

AWS CLI

Gunakan [delete-endpoint-config](#)perintah untuk menghapus konfigurasi titik akhir Anda. Tentukan nama konfigurasi titik akhir Anda untuk `endpoint-config-name` bendera.

```
aws sagemaker delete-endpoint-config \
                        --endpoint-config-name <endpoint-config-name>
```

Anda dapat menggunakan [describe-endpoint-config](#)perintah secara opsional untuk mengembalikan informasi tentang nama model yang Anda gunakan (varian produksi) seperti nama model Anda dan nama konfigurasi titik akhir yang terkait dengan model yang diterapkan tersebut. Berikan nama titik akhir Anda untuk `endpoint-config-name` bendera.

```
aws sagemaker describe-endpoint-config --endpoint-config-name <endpoint-config-name>
```

Ini akan mengembalikan respons JSON. Anda dapat menyalin dan menempel, menggunakan parser JSON, atau menggunakan alat yang dibuat untuk penguraian JSON untuk mendapatkan nama konfigurasi titik akhir yang terkait dengan titik akhir tersebut.

SageMaker Console

Hapus konfigurasi titik akhir Anda secara interaktif dengan konsol. SageMaker

1. Di SageMaker konsol di menu navigasi https://console.aws.amazon.com/sagemaker/, pilih Inferensi.
2. Pilih konfigurasi Endpoint dari menu dropdown. Daftar konfigurasi titik akhir yang dibuat di AWS akun Anda akan muncul berdasarkan nama, Nama Sumber Daya Amazon (ARN), dan waktu pembuatan.
3. Pilih konfigurasi titik akhir yang ingin Anda hapus.
4. Pilih tombol dropdown Actions di pojok kanan atas.
5. Pilih Hapus.

Hapus Model

Hapus SageMaker model Anda secara terprogram menggunakanAWS SDK for Python (Boto3), denganAWS CLI, atau secara interaktif menggunakan konsol. SageMaker Menghapus SageMaker model hanya menghapus entri model yang dibuat di. SageMaker Menghapus model tidak menghapus artefak model, kode inferensi, atau peran IAM yang Anda tentukan saat membuat model.

AWS SDK for Python (Boto3)

Gunakan `DeleteModel`API untuk menghapus SageMaker model Anda. Tentukan nama model Anda untuk `ModelName` bidang tersebut.

```
import boto3

# Specify your AWS Region
aws_region='<aws_region>'

# Specify the name of your endpoint configuration
model_name='<model_name>'

# Create a low-level SageMaker service client.
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Delete model
sagemaker_client.delete_model(ModelName=model_name)
```

Anda dapat menggunakan `DescribeEndpointConfig`API secara opsional untuk menampilkan informasi tentang nama model yang Anda gunakan (varian produksi) seperti nama model Anda dan nama konfigurasi titik akhir yang terkait dengan model yang diterapkan tersebut. Berikan nama titik akhir Anda untuk `EndpointConfigName` bidang tersebut.

```
# Specify the name of your endpoint
endpoint_name='<endpoint_name>'

# Create a low-level SageMaker service client.
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Store DescribeEndpointConfig response into a variable that we can index in the
 next step.
response =
 sagemaker_client.describe_endpoint_config(EndpointConfigName=endpoint_name)

# Delete endpoint
model_name = response['ProductionVariants'][0]['ModelName']
sagemaker_client.delete_model(ModelName=model_name)
```

Untuk informasi selengkapnya tentang elemen respons lain yang ditampilkan oleh`DescribeEndpointConfig`, lihat `DescribeEndpointConfig`di panduan Referensi SageMaker API.

AWS CLI

Gunakan `delete-model`perintah untuk menghapus SageMaker model Anda. Tentukan nama model Anda untuk `model-name` bendera.

```
aws sagemaker delete-model \
                        --model-name <model-name>
```

Anda dapat menggunakan `describe-endpoint-config`perintah secara opsional untuk mengembalikan informasi tentang nama model yang Anda gunakan (varian produksi) seperti nama model Anda dan nama konfigurasi titik akhir yang terkait dengan model yang diterapkan tersebut. Berikan nama titik akhir Anda untuk `endpoint-config-name` bendera.

```
aws sagemaker describe-endpoint-config --endpoint-config-name <endpoint-config-name>
```

Ini akan mengembalikan respons JSON. Anda dapat menyalin dan menempel, menggunakan parser JSON, atau menggunakan alat yang dibuat untuk penguraian JSON untuk mendapatkan nama model yang terkait dengan titik akhir tersebut.

SageMaker Console

Hapus SageMaker model Anda secara interaktif dengan SageMaker konsol.

1. Di SageMaker konsol di menu navigasi https://console.aws.amazon.com/sagemaker/, pilih Inferensi.
2. Pilih Model dari menu tarik-turun. Daftar model yang dibuat di AWS akun Anda akan muncul berdasarkan nama, Nama Sumber Daya Amazon (ARN), dan waktu pembuatan.
3. Pilih model yang ingin Anda hapus.
4. Pilih tombol dropdown Actions di pojok kanan atas.
5. Pilih Hapus.

Secara Otomatis Menskalakan SageMaker Model Amazon

Amazon SageMaker mendukung penskalaan otomatis (penskalaan otomatis) untuk model yang Anda hosting. Penskalaan otomatis secara dinamis menyesuaikan jumlah instance yang disediakan untuk model sebagai respons terhadap perubahan beban kerja Anda. Saat beban kerja meningkat, penskalaan otomatis menghadirkan lebih banyak instance online. Ketika beban kerja berkurang, penskalaan otomatis akan menghapus instans yang tidak perlu sehingga Anda tidak membayar instans yang disediakan yang tidak Anda gunakan.

Topik

- Gunakan AWS CloudFormation untuk membuat kebijakan penskalaan
- Memperbarui atau menghapus titik akhir yang menggunakan penskalaan otomatis

Ikhtisar penskalaan otomatis

Ikhtisar berikut memberikan rincian tentang prasyarat dan komponen yang digunakan untuk penskalaan otomatis.

Topik

- Prasyarat
- Ikhtisar kebijakan penskalaan
- Menskalakan berdasarkan jadwal
- Batas penskalaan minimum dan maksimum
- Periode pendinginan
- Izin
- Peran terkait layanan
- Sumber daya terkait

Prasyarat

Sebelum Anda dapat menggunakan penskalaan otomatis, Anda harus sudah membuat titik akhir SageMaker model Amazon. Anda dapat memiliki beberapa versi model untuk titik akhir yang sama. Setiap model disebut sebagai varian produksi (model). Untuk informasi selengkapnya tentang penerapan titik akhir model, lihat. Menyebarkan Model ke SageMaker Layanan Hosting

Untuk mengaktifkan auto scaling untuk model, Anda dapat menggunakan SageMaker konsol, AWS Command Line Interface (AWS CLI), atau AWS SDK melalui Application Auto Scaling API.

- Jika ini adalah pertama kalinya Anda mengonfigurasi penskalaan untuk model, kami sarankan Anda. Konfigurasikan penskalaan otomatis model dengan konsol
- Saat menggunakan AWS CLI atau Application Auto Scaling API, alurnya adalah mendaftarkan model sebagai target yang dapat diskalakan, menentukan kebijakan penskalaan, dan kemudian menerapkannya. Di SageMaker konsol, di bawah Inferensi di panel navigasi, pilih Endpoints. Temukan nama titik akhir model Anda dan kemudian pilih untuk menemukan nama varian. Anda harus menentukan nama titik akhir dan nama varian untuk mengaktifkan penskalaan otomatis untuk model.

Ikhtisar kebijakan penskalaan

Untuk menggunakan penskalaan otomatis, Anda menentukan kebijakan penskalaan yang menambahkan dan menghapus jumlah instance untuk varian produksi Anda sebagai respons terhadap beban kerja aktual.

Untuk menskalakan secara otomatis saat terjadi perubahan beban kerja, Anda memiliki dua opsi: kebijakan pelacakan target dan penskalaan langkah.

Sebaiknya gunakan kebijakan penskalaan pelacakan target. Dengan pelacakan target, Anda memilih CloudWatch metrik Amazon dan nilai target. Penskalaan otomatis membuat dan mengelola CloudWatch alarm untuk kebijakan penskalaan dan menghitung penyesuaian penskalaan berdasarkan metrik dan nilai target. Kebijakan menambahkan dan menghapus jumlah instance yang diperlukan untuk menjaga metrik pada, atau mendekati, nilai target yang ditentukan. Misalnya, kebijakan penskalaan yang menggunakan `InvocationsPerInstance` metrik yang telah ditentukan dengan nilai target 70 dapat dipertahankan`InvocationsPerInstance`, atau mendekati 70. Untuk informasi selengkapnya, lihat Kebijakan penskalaan pelacakan target di Panduan Pengguna Application Auto Scaling.

Anda dapat menggunakan penskalaan langkah saat memerlukan konfigurasi lanjutan, seperti menentukan berapa banyak instance yang akan diterapkan dalam kondisi apa. Jika tidak, menggunakan penskalaan pelacakan target lebih disukai karena akan sepenuhnya otomatis. Perhatikan bahwa penskalaan langkah hanya dapat dikelola dari Application Auto Scaling AWS CLI API atau Application Auto Scaling. Untuk gambaran umum tentang kebijakan penskalaan langkah dan cara kerjanya, lihat Kebijakan penskalaan langkah di Panduan Pengguna Application Auto Scaling

Untuk membuat kebijakan penskalaan pelacakan target, Anda menentukan hal berikut:

- Metrik — CloudWatch Metrik untuk dilacak, seperti jumlah rata-rata pemanggilan per instance.
- Nilai target — Nilai target untuk metrik, seperti 70 pemanggilan per instance per menit.

Anda dapat membuat kebijakan penskalaan pelacakan target dengan metrik yang telah ditentukan sebelumnya atau metrik khusus. Metrik yang telah ditentukan ditentukan dalam enumerasi sehingga Anda dapat menentukannya berdasarkan nama dalam kode atau menggunakannya di konsol. SageMaker Atau, Anda dapat menggunakan Application Auto Scaling API AWS CLI atau Application Auto Scaling untuk menerapkan kebijakan penskalaan pelacakan target berdasarkan metrik yang telah ditentukan atau kustom.

Perhatikan bahwa aktivitas penskalaan dilakukan dengan periode cooldown di antara mereka untuk mencegah fluktuasi kapasitas yang cepat. Anda dapat secara opsional mengonfigurasi periode cooldown untuk kebijakan penskalaan Anda.

Menskalakan berdasarkan jadwal

Anda juga dapat membuat tindakan terjadwal untuk melakukan aktivitas penskalaan pada waktu tertentu. Anda dapat membuat tindakan terjadwal yang menskalakan satu kali saja atau menskalakan berdasarkan jadwal berulang. Setelah tindakan terjadwal berjalan, kebijakan penskalaan Anda dapat terus membuat keputusan tentang apakah akan menskalakan secara dinamis saat terjadi perubahan beban kerja. Penskalaan terjadwal hanya dapat dikelola dari Application Auto Scaling AWS CLI API atau Application Auto Scaling. Untuk informasi lebih lanjut, lihat Penskalaan terjadwal dalam Panduan Pengguna Application Auto Scaling.

Batas penskalaan minimum dan maksimum

Saat mengonfigurasi penskalaan otomatis, Anda harus menentukan batas penskalaan sebelum membuat kebijakan penskalaan. Anda menetapkan batas secara terpisah untuk nilai minimum dan maksimum.

Nilai minimum harus minimal 1, dan sama dengan atau kurang dari nilai yang ditentukan untuk nilai maksimum.

Nilai maksimum harus sama dengan atau lebih besar dari nilai yang ditentukan untuk nilai minimum. SageMaker auto scaling tidak memberlakukan batasan untuk nilai ini.

Untuk menentukan batas penskalaan yang Anda perlukan untuk lalu lintas biasa, uji konfigurasi penskalaan otomatis Anda dengan laju lalu lintas yang diharapkan ke model Anda.

Jika lalu lintas varian menjadi nol, SageMaker secara otomatis menskalakan ke jumlah minimum instance yang ditentukan. Dalam hal ini, SageMaker memancarkan metrik dengan nilai nol.

Ada tiga opsi untuk menentukan kapasitas minimum dan maksimum:

1. Gunakan konsol untuk memperbarui jumlah instans Minimum dan pengaturan hitungan instans maksimum.
2. Gunakan AWS CLI dan sertakan `--max-capacity` opsi `--min-capacity` dan saat menjalankan register-scalable-targetperintah.
3. Panggil RegisterScalableTargetAPI dan tentukan `MaxCapacity` parameter `MinCapacity` dan.

> **Tip**
>
> Anda dapat menskalakan secara manual dengan meningkatkan nilai minimum, atau menskalakan secara manual dengan mengurangi nilai maksimum.

Periode pendinginan

Periode cooldown digunakan untuk melindungi dari penskalaan berlebih saat model Anda melakukan penskalaan (mengurangi kapasitas) atau penskalaan (meningkatkan kapasitas). Ini dilakukan dengan memperlambat aktivitas penskalaan berikutnya sampai periode berakhir. Secara khusus, ini memblokir penghapusan instance untuk permintaan scale-in, dan membatasi pembuatan instance untuk permintaan scale-out. Untuk informasi selengkapnya, lihat Menentukan periode cooldown di Panduan Pengguna Application Auto Scaling.

Anda mengonfigurasi periode cooldown dalam kebijakan penskalaan Anda.

Jika Anda tidak menentukan periode cooldown scale-in atau scale-out, kebijakan penskalaan Anda menggunakan default, yaitu masing-masing 300 detik.

Jika instance ditambahkan atau dihapus terlalu cepat saat Anda menguji konfigurasi penskalaan, pertimbangkan untuk meningkatkan nilai ini. Anda mungkin melihat perilaku ini jika lalu lintas ke model Anda memiliki banyak lonjakan, atau jika Anda memiliki beberapa kebijakan penskalaan yang ditentukan untuk varian.

Jika instance tidak ditambahkan cukup cepat untuk mengatasi peningkatan lalu lintas, pertimbangkan untuk mengurangi nilai ini.

Izin

Penskalaan otomatis dimungkinkan oleh kombinasi API Amazon SageMaker, Amazon CloudWatch, dan Application Auto Scaling. Untuk informasi tentang izin minimum yang diperlukan, lihat contoh kebijakan berbasis identitas Application Auto Scaling di Panduan Pengguna Application Auto Scaling.

Kebijakan `SagemakerFullAccessPolicy` IAM memiliki semua izin IAM yang diperlukan untuk melakukan penskalaan otomatis. Untuk informasi selengkapnya tentang izin SageMaker IAM, lihat. SageMaker Peran

Jika Anda mengelola kebijakan izin Anda sendiri, Anda harus menyertakan izin berikut:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:UpdateEndpointWeightsAndCapacities"
      ],
      "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "application-autoscaling:*"
        ],
        "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/sagemaker.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint",
      "Condition": {
        "StringLike": { "iam:AWSServiceName": "sagemaker.application-
autoscaling.amazonaws.com" }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:DeleteAlarms"
      ],
      "Resource": "*"
    }
  ]
}
```

Peran terkait layanan

Penskalaan otomatis menggunakan peran `AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint` terkait layanan. Peran terkait layanan ini memberikan izin Application Auto Scaling untuk menjelaskan alarm kebijakan Anda, memantau tingkat kapasitas saat ini, dan untuk menskalakan sumber daya target. Peran ini dibuat untuk Anda secara otomatis. Agar pembuatan peran otomatis berhasil, Anda harus memiliki izin untuk `iam:CreateServiceLinkedRole` tindakan tersebut. Untuk informasi selengkapnya, lihat Peran terkait layanan di Panduan Pengguna Application Auto Scaling.

Sumber daya terkait

Untuk informasi selengkapnya tentang mengonfigurasi penskalaan otomatis, lihat sumber daya berikut:

- Bagian application-autoscaling dari Referensi Perintah AWS CLI
- Referensi API Penskalaan Otomatis Aplikasi
- Panduan Pengguna Penskalaan Otomatis Aplikasi

Konfigurasikan penskalaan otomatis model dengan konsol

Untuk mengonfigurasi penskalaan otomatis untuk model (konsol)

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pada panel navigasi, pilih Inferensi, lalu pilih Endpoints.
3. Pilih endpoint Anda, dan kemudian untuk pengaturan runtime Endpoint, pilih variannya.
4. Pilih Konfigurasi penskalaan otomatis.
5. Pada halaman Configure variant automatic scaling, untuk Variant automatic scaling, lakukan hal berikut:

    a. Untuk jumlah instans Minimum, ketikkan jumlah instans minimum yang ingin dipertahankan oleh kebijakan penskalaan. Setidaknya 1 instance diperlukan.

    b. Untuk jumlah instans Maksimum, ketikkan jumlah instans maksimum yang ingin dipertahankan oleh kebijakan penskalaan.
6. Untuk kebijakan penskalaan bawaan, lakukan hal berikut:

a. Untuk metrik Target, `SageMakerVariantInvocationsPerInstance` secara otomatis dipilih untuk metrik dan tidak dapat diubah.

b. Untuk nilai Target, ketikkan jumlah rata-rata pemanggilan per instance per menit untuk model. Untuk menentukan nilai ini, ikuti pedoman diPengujian beban.

c. (Opsional) Untuk pendinginan Scale-in (detik) dan pendinginan Scale-out (detik), masukkan jumlah waktu, dalam detik, untuk setiap periode pendinginan.

d. (Opsional) Pilih Nonaktifkan skala jika Anda tidak ingin penskalaan otomatis menghentikan instance saat lalu lintas menurun.

7. Pilih Simpan.

Prosedur ini mendaftarkan model sebagai target yang dapat diskalakan dengan Application Auto Scaling. Saat Anda mendaftarkan model, Application Auto Scaling melakukan pemeriksaan validasi untuk memastikan hal-hal berikut:

- Modelnya ada
- Izin sudah cukup
- Anda tidak mendaftarkan varian dengan instance yang merupakan instance kinerja yang dapat dibobol seperti T2

> **Note**
>
> SageMaker tidak mendukung penskalaan otomatis untuk instans burstable seperti T2, karena mereka sudah memungkinkan peningkatan kapasitas di bawah peningkatan beban kerja. Untuk informasi tentang instans performa burstable, lihat jenis instans Amazon EC2.

Daftarkan model

Sebelum menambahkan kebijakan penskalaan ke model, pertama-tama Anda harus mendaftarkan model Anda untuk penskalaan otomatis dan menentukan batas penskalaan untuk model tersebut.

Prosedur berikut mencakup cara mendaftarkan model (varian produksi) untuk penskalaan otomatis menggunakan AWS Command Line Interface (AWS CLI) atau Application Auto Scaling API.

Topik

- Daftarkan model (AWS CLI)

- Daftarkan model (Application Auto Scaling API)

Daftarkan model (AWS CLI)

Untuk mendaftarkan varian produksi Anda, gunakan register-scalable-targetperintah dengan parameter berikut:

- `--service-namespace`—Tetapkan nilai ini `sagemaker`.
- `--resource-id`—Pengidentifikasi sumber daya untuk model (khususnya, varian produksi). Untuk parameter ini, tipe sumber daya adalah `endpoint` dan pengidentifikasi unik adalah nama varian produksi. Misalnya, `endpoint/`*`my-endpoint`*`/variant/`*`my-variant`*.
- `--scalable-dimension`—Tetapkan nilai ini `sagemaker:variant:DesiredInstanceCount`.
- `--min-capacity`Jumlah minimum instans. Nilai ini harus diatur ke minimal 1 dan harus sama dengan atau kurang dari nilai yang ditentukan untuk`max-capacity`.
- `--max-capacity`—Jumlah maksimum instance. Nilai ini harus diatur ke minimal 1 dan harus sama dengan atau lebih besar dari nilai yang ditentukan untuk`min-capacity`.

Example

Contoh berikut menunjukkan cara mendaftarkan varian bernama*`my-variant`*, berjalan pada *`my-endpoint`* titik akhir, yang dapat diskalakan secara dinamis untuk memiliki satu hingga delapan instance.

```
aws application-autoscaling register-scalable-target \
  --service-namespace sagemaker \
  --resource-id endpoint/my-endpoint/variant/my-variant \
  --scalable-dimension sagemaker:variant:DesiredInstanceCount \
  --min-capacity 1 \
  --max-capacity 8
```

Daftarkan model (Application Auto Scaling API)

Untuk mendaftarkan model Anda dengan Application Auto Scaling, gunakan aksi `RegisterScalableTarget`Application Auto Scaling API dengan parameter berikut:

- `ServiceNamespace`—Tetapkan nilai ini `sagemaker`.

- `ResourceID`—Pengidentifikasi sumber daya untuk varian produksi. Untuk parameter ini, tipe sumber daya adalah `endpoint` dan pengidentifikasi unik adalah nama varian. Sebagai contoh, `endpoint/`*`my-endpoint`*`/variant/`*`my-variant`*.
- `ScalableDimension`—Tetapkan nilai ini ke`sagemaker:variant:DesiredInstanceCount`.
- `MinCapacity`Jumlah minimum instans. Nilai ini harus diatur ke minimal 1 dan harus sama dengan atau kurang dari nilai yang ditentukan untuk`MaxCapacity`.
- `MaxCapacity`—Jumlah maksimum instance. Nilai ini harus diatur ke minimal 1 dan harus sama dengan atau lebih besar dari nilai yang ditentukan untuk`MinCapacity`.

Example

Contoh berikut menunjukkan cara mendaftarkan varian bernama*my-variant*, berjalan pada *my-endpoint* titik akhir, yang dapat diskalakan secara dinamis untuk menggunakan satu hingga delapan instance.

```
POST / HTTP/1.1
Host: application-autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20230506T182145Z
User-Agent: aws-cli/2.0.0 Python/3.7.5 Windows/10 botocore/2.0.0dev4
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/my-endpoint/variant/my-variant",
    "ScalableDimension": "sagemaker:variant:DesiredInstanceCount",
    "MinCapacity": 1,
    "MaxCapacity": 8
}
```

Menentukan kebijakan penskalaan

Sebelum menambahkan kebijakan penskalaan ke model, simpan konfigurasi kebijakan Anda sebagai blok JSON dalam file teks. Anda menggunakan file teks tersebut saat menjalankan AWS Command Line Interface (AWS CLI) atau Application Auto Scaling API. Anda dapat mengoptimalkan penskalaan dengan memilih CloudWatch metrik yang sesuai. Namun, sebelum menggunakan metrik khusus dalam produksi, Anda harus menguji penskalaan otomatis dengan metrik khusus Anda.

Bagian ini menunjukkan contoh konfigurasi kebijakan untuk kebijakan penskalaan pelacakan target.

Topik

- Tentukan metrik yang telah ditentukan (CloudWatch metrik: InvocationsPerInstance)
- Tentukan metrik khusus (CloudWatchmetrik: CPUUutilization)
- Tentukan metrik khusus (CloudWatch metrik: ExplanationsPerInstance)
- Tentukan periode cooldown

Tentukan metrik yang telah ditentukan (CloudWatch metrik: InvocationsPerInstance)

Example

Berikut ini adalah contoh konfigurasi kebijakan pelacakan target untuk varian yang menjaga pemanggilan rata-rata per instance pada 70. Simpan konfigurasi ini dalam file bernama `config.json`.

```
{
    "TargetValue": 70.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "SageMakerVariantInvocationsPerInstance"
    }
}
```

Untuk informasi lebih lanjut, lihat TargetTrackingScalingPolicyConfiguration dalam Referensi API Application Auto Scaling.

Tentukan metrik khusus (CloudWatchmetrik: CPUUutilization)

Untuk membuat kebijakan penskalaan pelacakan target dengan metrik kustom, tentukan nama metrik, namespace, unit, statistik, dan dimensi nol atau lebih. Dimensi terdiri dari nama dimensi dan nilai dimensi. Anda dapat menggunakan metrik varian produksi apa pun yang berubah sebanding dengan kapasitas.

Example

Contoh konfigurasi berikut menunjukkan kebijakan penskalaan pelacakan target dengan metrik kustom. Kebijakan ini menskalakan varian berdasarkan pemanfaatan CPU rata-rata 50 persen di semua kasus. Simpan konfigurasi ini dalam file bernama `config.json`.

```
{
    "TargetValue": 50.0,
    "CustomizedMetricSpecification":
    {
        "MetricName": "CPUUtilization",
        "Namespace": "/aws/sagemaker/Endpoints",
        "Dimensions": [
            {"Name": "EndpointName", "Value": "my-endpoint" },
            {"Name": "VariantName","Value": "my-variant"}
        ],
        "Statistic": "Average",
        "Unit": "Percent"
    }
}
```

Untuk informasi lebih lanjut, lihat CustomizedMetricSpecification dalam Referensi API Application Auto Scaling.

Tentukan metrik khusus (CloudWatch metrik: ExplanationsPerInstance)

Ketika titik akhir mengaktifkan penjelasan online, ia memancarkan `ExplanationsPerInstance` metrik yang menghasilkan jumlah rata-rata catatan yang dijelaskan per menit, per contoh, untuk sebuah varian. Pemanfaatan sumber daya untuk menjelaskan catatan bisa lebih berbeda dari pada catatan prediksi. Kami sangat menyarankan menggunakan metrik ini untuk penskalaan pelacakan target titik akhir dengan kemampuan penjelasan online diaktifkan.

Anda dapat membuat beberapa kebijakan pelacakan target untuk target yang dapat diskalakan. Pertimbangkan untuk menambahkan `InvocationsPerInstance` kebijakan dari Tentukan metrik yang telah ditentukan (CloudWatch metrik: InvocationsPerInstance) bagian (selain `ExplanationsPerInstance` kebijakan). Jika sebagian besar pemanggilan tidak menampilkan penjelasan karena nilai ambang batas yang ditetapkan dalam `EnableExplanations` parameter, maka titik akhir dapat memilih kebijakan. `InvocationsPerInstance` Jika ada banyak penjelasan, titik akhir dapat menggunakan kebijakan. `ExplanationsPerInstance`

Example

Contoh konfigurasi berikut menunjukkan kebijakan penskalaan pelacakan target dengan metrik kustom. Skala kebijakan menyesuaikan jumlah instance varian sehingga setiap instance memiliki `ExplanationsPerInstance` metrik 20. Simpan konfigurasi ini dalam file bernama `config.json`.

```
{
```

```
    "TargetValue": 20.0,
    "CustomizedMetricSpecification":
    {
        "MetricName": "ExplanationsPerInstance",
        "Namespace": "AWS/SageMaker",
        "Dimensions": [
            {"Name": "EndpointName", "Value": "my-endpoint" },
            {"Name": "VariantName","Value": "my-variant"}
        ],
        "Statistic": "Sum"
    }
}
```

Untuk informasi lebih lanjut, lihat CustomizedMetricSpecification dalam Referensi API Application Auto Scaling.

Tentukan periode cooldown

Anda dapat secara opsional menentukan periode cooldown dalam kebijakan penskalaan pelacakan target Anda dengan menentukan dan parameter. `ScaleOutCooldown` `ScaleInCooldown`

Example

Berikut ini adalah contoh konfigurasi kebijakan pelacakan target untuk varian yang menjaga pemanggilan rata-rata per instance pada 70. Konfigurasi kebijakan menyediakan periode cooldown scale-in 10 menit (600 detik) dan periode cooldown scale-out 5 menit (300 detik). Simpan konfigurasi ini dalam file bernama `config.json`.

```
{
    "TargetValue": 70.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "SageMakerVariantInvocationsPerInstance"
    },
    "ScaleInCooldown": 600,
    "ScaleOutCooldown": 300
}
```

Untuk informasi lebih lanjut, lihat TargetTrackingScalingPolicyConfiguration dalam Referensi API Application Auto Scaling.

Menerapkan kebijakan penskalaan

Setelah Anda mendaftarkan model dan menentukan kebijakan penskalaan, terapkan kebijakan penskalaan ke model terdaftar. Bagian ini menunjukkan cara menerapkan kebijakan penskalaan menggunakan AWS Command Line Interface (AWS CLI) atau Application Auto Scaling API.

Topik

- Menerapkan kebijakan penskalaan pelacakan target () AWS CLI
- Menerapkan kebijakan penskalaan (Application Auto Scaling API)

Menerapkan kebijakan penskalaan pelacakan target () AWS CLI

Untuk menerapkan kebijakan penskalaan pada model Anda, gunakan `put-scaling-policy`AWS CLIperintah dengan parameter berikut:

- `--policy-name`—Nama kebijakan penskalaan.
- `--policy-type`—Tetapkan nilai ini ke`TargetTrackingScaling`.
- `--resource-id`—Pengidentifikasi sumber daya untuk varian. Untuk parameter ini, tipe sumber daya adalah `endpoint` dan pengidentifikasi unik adalah nama varian. Misalnya, `endpoint/`*`my-endpoint`*`/variant/`*`my-variant`*.
- `--service-namespace`—Tetapkan nilai ini ke`sagemaker`.
- `--scalable-dimension`—Tetapkan nilai ini `sagemaker:variant:DesiredInstanceCount`.
- `--target-tracking-scaling-policy-configuration`—Konfigurasi kebijakan penskalaan pelacakan target yang akan digunakan untuk model.

Example

Contoh berikut menerapkan kebijakan penskalaan pelacakan target yang diberi nama *`my-scaling-policy`* ke varian bernama*`my-variant`*, berjalan di titik *`my-endpoint`* akhir. Untuk `--target-tracking-scaling-policy-configuration` opsi, tentukan `config.json` file yang Anda buat sebelumnya.

```
aws application-autoscaling put-scaling-policy \
  --policy-name my-scaling-policy \
  --policy-type TargetTrackingScaling \
```

```
  --resource-id endpoint/my-endpoint/variant/my-variant \
  --service-namespace sagemaker \
  --scalable-dimension sagemaker:variant:DesiredInstanceCount \
  --target-tracking-scaling-policy-configuration file://config.json
```

Menerapkan kebijakan penskalaan (Application Auto Scaling API)

Untuk menerapkan kebijakan penskalaan ke varian dengan Application Auto Scaling API, gunakan `PutScalingPolicy`aksi Application Auto Scaling API dengan parameter berikut:

- `PolicyName`—Nama kebijakan penskalaan.
- `ServiceNamespace`—Tetapkan nilai ini ke`sagemaker`.
- `ResourceID`—Pengidentifikasi sumber daya untuk varian. Untuk parameter ini, tipe sumber daya adalah `endpoint` dan pengidentifikasi unik adalah nama varian. Misalnya, `endpoint/`*`my-endpoint`*`/variant/`*`my-variant`*.
- `ScalableDimension`—Tetapkan nilai ini ke`sagemaker:variant:DesiredInstanceCount`.
- `PolicyType`—Tetapkan nilai ini ke`TargetTrackingScaling`.
- `TargetTrackingScalingPolicyConfiguration`—Konfigurasi kebijakan penskalaan pelacakan target yang akan digunakan untuk varian.

Example

Contoh berikut menerapkan kebijakan penskalaan pelacakan target yang diberi nama *`my-scaling-policy`* ke varian bernama*`my-variant`*, berjalan di titik *`my-endpoint`* akhir. Konfigurasi kebijakan menjaga pemanggilan rata-rata per instance pada 70.

```
POST / HTTP/1.1
Host: application-autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.
X-Amz-Date: 20230506T182145Z
User-Agent: aws-cli/2.0.0 Python/3.7.5 Windows/10 botocore/2.0.0dev4
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "PolicyName": "my-scaling-policy",
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/my-endpoint/variant/my-variant",
```

```
    "ScalableDimension": "sagemaker:variant:DesiredInstanceCount",
    "PolicyType": "TargetTrackingScaling",
    "TargetTrackingScalingPolicyConfiguration": {
        "TargetValue": 70.0,
        "PredefinedMetricSpecification":
        {
            "PredefinedMetricType": "SageMakerVariantInvocationsPerInstance"
        }
    }
}
```

Mengedit kebijakan penskalaan

Setelah membuat kebijakan penskalaan, Anda dapat mengedit pengaturannya kecuali namanya.

Topik

- Mengedit kebijakan penskalaan (konsol)
- Mengedit kebijakan penskalaan (AWS CLIatau Application Auto Scaling API)
- Matikan sementara kebijakan penskalaan

Mengedit kebijakan penskalaan (konsol)

Untuk mengedit kebijakan penskalaan pelacakan target denganAWS Management Console, gunakan prosedur yang sama dengan yang Anda gunakan. Konfigurasikan penskalaan otomatis model dengan konsol

Mengedit kebijakan penskalaan (AWS CLIatau Application Auto Scaling API)

Anda dapat menggunakan Application Auto Scaling API AWS CLI atau Application Auto Scaling untuk mengedit kebijakan penskalaan dengan cara yang sama seperti Anda membuat kebijakan penskalaan baru. Untuk informasi selengkapnya, lihat Menerapkan kebijakan penskalaan.

Matikan sementara kebijakan penskalaan

Setelah mengonfigurasi penskalaan otomatis, Anda memiliki opsi berikut jika perlu menyelidiki masalah tanpa gangguan dari kebijakan penskalaan (penskalaan dinamis):

- Menangguhkan sementara dan kemudian melanjutkan aktivitas penskalaan dengan memanggil perintah register-scalable-targetCLI atau tindakan RegisterScalableTargetAPI,

menentukan nilai Boolean untuk keduanya dan. `DynamicScalingInSuspended` `DynamicScalingOutSuspended`

Example

Contoh berikut menunjukkan cara menangguhkan kebijakan penskalaan untuk varian bernama*my-variant*, berjalan di titik akhir*my-endpoint*.

```
aws application-autoscaling register-scalable-target \
  --service-namespace sagemaker \
  --resource-id endpoint/my-endpoint/variant/my-variant \
  --scalable-dimension sagemaker:variant:DesiredInstanceCount \
  --suspended-
state '{"DynamicScalingInSuspended":true,"DynamicScalingOutSuspended":true}'
```

- Cegah kebijakan penskalaan pelacakan target tertentu dari penskalaan dalam varian Anda dengan menonaktifkan bagian penskalaan kebijakan. Metode ini mencegah kebijakan penskalaan menghapus instance, sambil tetap mengizinkannya membuatnya sesuai kebutuhan.

  Nonaktifkan sementara dan kemudian aktifkan aktivitas penskalaan dengan mengedit kebijakan menggunakan perintah put-scaling-policyCLI atau tindakan PutScalingPolicyAPI, yang menentukan nilai Boolean untuk. `DisableScaleIn`

  Example

  Berikut ini adalah contoh konfigurasi pelacakan target untuk kebijakan penskalaan yang akan menskalakan tetapi tidak menskalakan.

```
{
    "TargetValue": 70.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "SageMakerVariantInvocationsPerInstance"
    },
    "DisableScaleIn": true
}
```

Menghapus kebijakan penskalaan

Jika Anda tidak lagi memerlukan kebijakan penskalaan, Anda dapat menghapusnya kapan saja.

Topik

Hapus semua kebijakan penskalaan dan deregister model (konsol)

Untuk menghapus semua kebijakan penskalaan dan membatalkan pendaftaran varian sebagai target yang dapat diskalakan

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pada panel navigasi, pilih Endpoints.
3. Pilih endpoint Anda, dan kemudian untuk pengaturan runtime Endpoint, pilih variannya.
4. Pilih Konfigurasi penskalaan otomatis.
5. Pilih Deregister auto scaling.

Menghapus kebijakan penskalaan (AWS CLIatau Application Auto Scaling API)

Anda dapat menggunakan Application Auto Scaling API AWS CLI atau Application Auto Scaling untuk menghapus kebijakan penskalaan dari varian.

Menghapus kebijakan penskalaan () AWS CLI

Untuk menghapus kebijakan penskalaan dari varian, gunakan delete-scaling-policyperintah dengan parameter berikut:

- `--policy-name`—Nama kebijakan penskalaan.
- `--resource-id`—Pengidentifikasi sumber daya untuk varian. Untuk parameter ini, tipe sumber daya adalah `endpoint` dan pengidentifikasi unik adalah nama varian. Misalnya, `endpoint/`*`my-endpoint`*`/variant/`*`my-variant`*.
- `--service-namespace`—Tetapkan nilai ini `kesagemaker`.
- `--scalable-dimension`—Tetapkan nilai ini `kesagemaker:variant:DesiredInstanceCount`.

Example

Contoh berikut menghapus kebijakan penskalaan pelacakan target yang dinamai *my-scaling-policy* dari varian bernama*my-variant*, berjalan di titik akhir*my-endpoint*.

```
aws application-autoscaling delete-scaling-policy \
  --policy-name my-scaling-policy \
  --resource-id endpoint/my-endpoint/variant/my-variant \
  --service-namespace sagemaker \
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
```

Menghapus kebijakan penskalaan (Application Auto Scaling API)

Untuk menghapus kebijakan penskalaan dari varian Anda, gunakan tindakan DeleteScalingPolicyApplication Auto Scaling API dengan parameter berikut:

- `PolicyName`—Nama kebijakan penskalaan.
- `ServiceNamespace`—Tetapkan nilai ini ke`sagemaker`.
- `ResourceID`—Pengidentifikasi sumber daya untuk varian. Untuk parameter ini, tipe sumber daya adalah `endpoint` dan pengidentifikasi unik adalah nama varian. Misalnya, `endpoint/`*my-endpoint*`/variant/`*my-variant*.
- `ScalableDimension`—Tetapkan nilai ini ke`sagemaker:variant:DesiredInstanceCount`.

Example

Contoh berikut menghapus kebijakan penskalaan pelacakan target yang dinamai *my-scaling-policy* dari varian bernama*my-variant*, berjalan di titik akhir*my-endpoint*.

```
POST / HTTP/1.1
Host: application-autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20230506T182145Z
User-Agent: aws-cli/2.0.0 Python/3.7.5 Windows/10 botocore/2.0.0dev4
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "PolicyName": "my-scaling-policy",
    "ServiceNamespace": "sagemaker",
```

```
    "ResourceId": "endpoint/my-endpoint/variant/my-variant",
    "ScalableDimension": "sagemaker:variant:DesiredInstanceCount"
}
```

Memeriksa status aktivitas penskalaan dengan menjelaskan aktivitas penskalaan

Anda dapat memeriksa status aktivitas penskalaan untuk titik akhir yang diskalakan otomatis dengan menjelaskan aktivitas penskalaan. Application Auto Scaling memberikan informasi deskriptif tentang aktivitas penskalaan di namespace yang ditentukan dari enam minggu sebelumnya. Untuk informasi selengkapnya, lihat Aktivitas penskalaan untuk Application Auto Scaling di Panduan Pengguna Application Auto Scaling.

Untuk memeriksa status aktivitas penskalaan, gunakan describe-scaling-activitiesperintah. Anda tidak dapat memeriksa status aktivitas penskalaan menggunakan konsol.

Topik

- Jelaskan aktivitas penskalaan () AWS CLI
- Identifikasi aktivitas penskalaan yang diblokir dari kuota instance () AWS CLI

Jelaskan aktivitas penskalaan () AWS CLI

Untuk menjelaskan aktivitas penskalaan untuk semua SageMaker sumber daya yang terdaftar dengan Application Auto Scaling, gunakan perintah, describe-scaling-activitiesyang `sagemaker` menentukan opsi. `--service-namespace`

```
aws application-autoscaling describe-scaling-activities \
  --service-namespace sagemaker
```

Untuk menggambarkan aktivitas penskalaan untuk sumber daya tertentu, sertakan `--resource-id` opsi.

```
aws application-autoscaling describe-scaling-activities \
  --service-namespace sagemaker \
  --resource-id endpoint/my-endpoint/variant/my-variant
```

Contoh berikut menunjukkan output yang dihasilkan ketika Anda menjalankan perintah ini.

```
{
    "ActivityId": "activity-id",
```

```
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/my-endpoint/variant/my-variant",
    "ScalableDimension": "sagemaker:variant:DesiredInstanceCount",
    "Description": "string",
    "Cause": "string",
    "StartTime": timestamp,
    "EndTime": timestamp,
    "StatusCode": "string",
    "StatusMessage": "string"
}
```

Identifikasi aktivitas penskalaan yang diblokir dari kuota instance () AWS CLI

Saat Anda memperkecil skala (menambahkan lebih banyak instance), Anda mungkin mencapai kuota instans tingkat akun Anda. Anda dapat menggunakan describe-scaling-activitiesperintah untuk memeriksa apakah Anda telah mencapai kuota instans Anda. Ketika Anda melebihi kuota Anda, penskalaan otomatis diblokir.

Untuk memeriksa apakah Anda telah mencapai kuota instans Anda, gunakan describe-scaling-activitiesperintah dan tentukan ID sumber daya untuk `--resource-id` opsi tersebut.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace sagemaker \
    --resource-id endpoint/my-endpoint/variant/my-variant
```

Dalam sintaks pengembalian, periksa StatusMessagekunci StatusCodedan dan nilai terkaitnya. `StatusCode`kembali`Failed`. Di dalamnya `StatusMessage` ada pesan yang menunjukkan bahwa kuota layanan tingkat akun telah tercapai. Berikut ini adalah contoh dari apa pesan itu mungkin terlihat seperti:

```
{
    "ActivityId": "activity-id",
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/my-endpoint/variant/my-variant",
    "ScalableDimension": "sagemaker:variant:DesiredInstanceCount",
    "Description": "string",
    "Cause": "minimum capacity was set to 110",
    "StartTime": timestamp,
    "EndTime": timestamp,
    "StatusCode": "Failed",
    "StatusMessage": "Failed to set desired instance count to 110. Reason: The
    account-level service limit 'ml.xx.xxxxxx for endpoint usage' is 1000
```

```
    Instances, with current utilization of 997 Instances and a request delta
    of 20 Instances. Please contact AWS support to request an increase for this
    limit. (Service: AmazonSageMaker; Status Code: 400;
    Error Code: ResourceLimitExceeded; Request ID: request-id)."
}
```

Memuat pengujian konfigurasi penskalaan otomatis Anda

Lakukan tes beban untuk memilih konfigurasi penskalaan yang berfungsi seperti yang Anda inginkan.

Pedoman pengujian beban berikut mengasumsikan Anda menggunakan kebijakan penskalaan yang menggunakan metrik target yang telah ditentukan sebelumnya. `SageMakerVariantInvocationsPerInstance`

Topik

- Tentukan karakteristik kinerja
- Hitung beban target

Tentukan karakteristik kinerja

Lakukan pengujian beban untuk menemukan puncak `InvocationsPerInstance` yang dapat ditangani oleh varian produksi model Anda, dan latensi permintaan, saat konkurensi meningkat.

Nilai ini bergantung pada jenis instans yang dipilih, muatan yang biasanya dikirim oleh klien model Anda, dan kinerja dependensi eksternal apa pun yang dimiliki model Anda.

Untuk menemukan puncak requests-per-second (RPS) varian produksi model Anda dapat menangani dan latensi permintaan

1. Siapkan titik akhir dengan model Anda menggunakan satu instance. Untuk informasi tentang cara menyiapkan titik akhir, lihatMenyebarkan Model ke SageMaker Layanan Hosting.
2. Gunakan alat pengujian beban untuk menghasilkan peningkatan jumlah permintaan paralel, dan pantau RPS dan latensi model di out put dari alat pengujian beban.

> **Note**
>
> Anda juga dapat memantau requests-per-minute alih-alih RPS. Dalam hal ini jangan kalikan dengan 60 dalam persamaan untuk menghitung yang `SageMakerVariantInvocationsPerInstance` ditunjukkan di bawah ini.

Ketika latensi model meningkat atau proporsi transaksi yang berhasil menurun, ini adalah RPS puncak yang dapat ditangani model Anda.

Hitung beban target

Setelah Anda menemukan karakteristik kinerja varian, Anda dapat menentukan RPS maksimum yang harus kami izinkan untuk dikirim ke sebuah instance. Ambang batas yang digunakan untuk penskalaan harus kurang dari nilai maksimum ini. Gunakan persamaan berikut dalam kombinasi dengan pengujian beban untuk menentukan nilai yang benar untuk metrik `SageMakerVariantInvocationsPerInstance` target dalam konfigurasi penskalaan Anda.

```
SageMakerVariantInvocationsPerInstance = (MAX_RPS * SAFETY_FACTOR) * 60
```

Di `MAX_RPS` mana RPS maksimum yang Anda tentukan sebelumnya, dan `SAFETY_FACTOR` merupakan faktor keamanan yang Anda pilih untuk memastikan bahwa klien Anda tidak melebihi RPS maksimum. Kalikan dengan 60 untuk mengonversi dari RPS invocations-per-minute agar sesuai dengan CloudWatch metrik per menit yang SageMaker digunakan untuk menerapkan penskalaan otomatis (Anda tidak perlu melakukan ini jika Anda mengukur requests-per-minute alih-alih). requests-per-second

> **Note**
>
> SageMaker merekomendasikan agar Anda mulai menguji dengan `SAFETY_FACTOR` 0,5. Uji konfigurasi penskalaan Anda untuk memastikannya beroperasi seperti yang Anda harapkan dengan model Anda untuk meningkatkan dan mengurangi lalu lintas pelanggan di titik akhir Anda.

Gunakan AWS CloudFormation untuk membuat kebijakan penskalaan

Contoh berikut menunjukkan cara mengonfigurasi penskalaan otomatis model pada titik akhir menggunakan. AWS CloudFormation

```
Endpoint:
  Type: "AWS::SageMaker::Endpoint"
  Properties:
    EndpointName: yourEndpointName
```

```
      EndpointConfigName: yourEndpointConfigName

  ScalingTarget:
    Type: "AWS::ApplicationAutoScaling::ScalableTarget"
    Properties:
      MaxCapacity: 10
      MinCapacity: 2
      ResourceId: endpoint/my-endpoint/variant/my-variant
      RoleARN: arn
      ScalableDimension: sagemaker:variant:DesiredInstanceCount
      ServiceNamespace: sagemaker

  ScalingPolicy:
    Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
    Properties:
      PolicyName: my-scaling-policy
      PolicyType: TargetTrackingScaling
      ScalingTargetId:
        Ref: ScalingTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 70.0
        ScaleInCooldown: 600
        ScaleOutCooldown: 30
        PredefinedMetricSpecification:
          PredefinedMetricType: SageMakerVariantInvocationsPerInstance
```

Untuk informasi selengkapnya, lihat Membuat sumber daya Application Auto Scaling dengan AWS CloudFormation di Panduan Pengguna Application Auto Scaling.

Memperbarui atau menghapus titik akhir yang menggunakan penskalaan otomatis

Topik

- Perbarui titik akhir yang menggunakan penskalaan otomatis
- Hapus titik akhir yang dikonfigurasi untuk penskalaan otomatis

Perbarui titik akhir yang menggunakan penskalaan otomatis

Saat Anda memperbarui titik akhir, Application Auto Scaling memeriksa untuk melihat apakah salah satu model pada titik akhir tersebut adalah target untuk penskalaan otomatis. Jika pembaruan akan mengubah jenis instance untuk model apa pun yang menjadi target penskalaan otomatis, pembaruan gagal.

DiAWS Management Console, Anda melihat peringatan bahwa Anda harus membatalkan pendaftaran model dari penskalaan otomatis sebelum Anda dapat memperbaruinya. Jika Anda mencoba memperbarui titik akhir dengan memanggil UpdateEndpointAPI, panggilan gagal. Sebelum memperbarui titik akhir, hapus kebijakan penskalaan apa pun yang dikonfigurasi untuknya dan deregister varian sebagai target yang dapat diskalakan dengan memanggil tindakan Application Auto DeregisterScalableTargetScaling API. Setelah memperbarui titik akhir, Anda dapat mendaftarkan varian yang diperbarui sebagai target yang dapat diskalakan dan melampirkan kebijakan penskalaan.

Ada satu pengecualian. Jika Anda mengubah model untuk varian yang dikonfigurasi untuk penskalaan otomatis, penskalaan SageMaker otomatis Amazon memungkinkan pembaruan. Ini karena mengubah model biasanya tidak cukup memengaruhi kinerja untuk mengubah perilaku penskalaan. Jika Anda memperbarui model untuk varian yang dikonfigurasi untuk penskalaan otomatis, pastikan bahwa perubahan pada model tidak memengaruhi kinerja dan perilaku penskalaan secara signifikan.

Saat Anda memperbarui SageMaker titik akhir yang menerapkan penskalaan otomatis, selesaikan langkah-langkah berikut:

Untuk memperbarui titik akhir yang menerapkan penskalaan otomatis

1. Deregister endpoint sebagai target yang dapat diskalakan dengan menelepon. DeregisterScalableTarget
2. Karena penskalaan otomatis diblokir saat operasi pembaruan sedang berlangsung (atau jika Anda mematikan penskalaan otomatis pada langkah sebelumnya), Anda mungkin ingin mengambil tindakan pencegahan tambahan untuk meningkatkan jumlah instance untuk titik akhir Anda selama pembaruan. Untuk melakukan ini, perbarui jumlah instance untuk varian produksi yang dihosting di titik akhir dengan memanggil. https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_UpdateEndpointWeightsAndCapacities.html UpdateEndpointWeightsAndCapacities
3. Panggil DescribeEndpointberulang kali sampai nilai `EndpointStatus` bidang responsnya`InService`.
4. Panggil DescribeEndpointConfiguntuk mendapatkan nilai konfigurasi titik akhir saat ini.
5. Buat konfigurasi titik akhir baru dengan menelepon. CreateEndpointConfig Untuk varian produksi tempat Anda ingin menyimpan jumlah atau bobot instans yang ada, gunakan nama varian yang sama dari respons dari panggilan ke DescribeEndpointConfiglangkah sebelumnya. Untuk semua nilai lainnya, gunakan nilai yang Anda dapatkan sebagai respons saat Anda menelepon DescribeEndpointConfigdi langkah sebelumnya.

6. Perbarui titik akhir dengan menelepon UpdateEndpoint. Tentukan konfigurasi titik akhir yang Anda buat di langkah sebelumnya sebagai bidang. `EndpointConfig` Jika Anda ingin mempertahankan properti varian seperti hitungan instance atau bobot, tetapkan nilai `RetainAllVariantProperties` parameter ke`True`. Ini menentukan bahwa varian produksi dengan nama yang sama akan diperbarui dengan yang terbaru `DesiredInstanceCount` dari respons dari panggilan ke`DescribeEndpoint`, terlepas dari nilai `InitialInstanceCount` bidang di yang baru`EndpointConfig`.
7. (Opsional) Aktifkan kembali penskalaan otomatis dengan menelepon RegisterScalableTargetdan. PutScalingPolicy

> **Note**
>
> Langkah 1 dan 7 hanya diperlukan jika Anda memperbarui titik akhir dengan perubahan berikut:
>
> - Mengubah jenis instans untuk varian produksi yang memiliki penskalaan otomatis yang dikonfigurasi
> - Menghapus varian produksi yang memiliki penskalaan otomatis yang dikonfigurasi.

Hapus titik akhir yang dikonfigurasi untuk penskalaan otomatis

Jika Anda menghapus titik akhir, Application Auto Scaling akan memeriksa apakah salah satu model pada titik akhir tersebut adalah target untuk penskalaan otomatis. Jika ada dan Anda memiliki izin untuk membatalkan pendaftaran model, Application Auto Scaling membatalkan pendaftaran model tersebut sebagai target yang dapat diskalakan tanpa memberi tahu Anda. Jika Anda menggunakan kebijakan izin khusus yang tidak memberikan izin untuk DeregisterScalableTargettindakan tersebut, Anda harus meminta akses ke tindakan ini sebelum menghapus titik akhir.

> **Note**
>
> Sebagai pengguna IAM, Anda mungkin tidak memiliki izin yang cukup untuk menghapus titik akhir jika pengguna lain mengonfigurasi penskalaan otomatis untuk varian pada titik akhir tersebut.

Volume volume penyimpanan volume penyimpanan instans volume penyimpanan instans

Saat Anda membuat titik akhir, Amazon SageMaker Elastic Block Store (Amazon EBS) untuk instans Amazon EC2 yang menjadi tuan rumah titik akhir. Ukuran volume penyimpanan dapat diskalakan, dan opsi penyimpanan dibagi menjadi dua kategori: penyimpanan yang didukung SSD dan penyimpanan yang didukung HDD.

Untuk informasi selengkapnya tentang penyimpanan dan fitur Amazon EBS, lihat halaman berikut.

- Fitur Amazon EBS
- Panduan Pengguna Amazon EBS

Untuk daftar lengkap volume penyimpanan instans host, lihat Tabel Volume Penyimpanan Instans Host

> **Note**
>
> Amazon SageMaker melampirkan volume penyimpanan Amazon Elastic Block Store (Amazon EBS) ke instans Amazon EC2 hanya saat Anda membuatInferensi asinkron atau jenisInferensi waktu nyata titik akhir. Untuk informasi selengkapnya tentang menyesuaikan volume penyimpanan Amazon EBS, lihatSageMaker parameter titik akhir untuk inferensi model besar.

Aman memvalidasi model dalam produksi

Dengan SageMaker, Anda dapat menguji beberapa model atau versi model di belakang titik akhir yang sama menggunakan varian. Varian terdiri dari instans MLdan komponen penyajian yang ditentukan dalam SageMaker model. Anda dapat memiliki beberapa varian di belakang titik akhir. Setiap varian dapat memiliki jenis instance yang berbeda atau SageMaker model yang dapat diskalakan secara otomatis secara independen dari yang lain. Model dalam varian dapat dilatih menggunakan kumpulan data yang berbeda, algoritme yang berbeda, kerangka kerja yang berbeda, atau kombinasi dari semua ini. Semua varian di balik endpoint berbagi kode inferensi yang sama. SageMaker mendukung dua jenis varian, varian produksi dan varian bayangan.

Jika Anda memiliki beberapa varian produksi di belakang titik akhir, maka Anda dapat mengalokasikan sebagian permintaan inferensi Anda ke setiap varian. Setiap permintaan dialihkan

ke hanya satu varian produksi. Varian produksi yang permintaan dialihkan memberikan respons terhadap penelepon. Anda dapat membandingkan bagaimana varian produksi berkinerja relatif satu sama lain.

Anda juga dapat memiliki varian bayangan yang sesuai dengan varian produksi di belakang titik akhir. Sebagian permintaan inferensi yang masuk ke varian produksi direplikasi ke varian bayangan. Respons varian bayangan dicatat untuk perbandingan dan tidak dikembalikan ke pemanggil. Ini memungkinkan Anda menguji kinerja varian bayangan tanpa mengekspos pemanggil ke respons yang dihasilkan oleh varian bayangan.

Topik

- Varian produksi
- Varian bayangan angan angan angan angan

Varian produksi

Dalam alur kerja ML produksi, ilmuwan data dan insinyur sering mencoba meningkatkan kinerja menggunakan berbagai metode, seperti Lakukan Penyetelan Model Otomatis dengan SageMaker, pelatihan data tambahan atau yang lebih baru, meningkatkan pemilihan fitur, menggunakan instans terbaru yang lebih baik, dan menyajikan kontainer. Anda dapat menggunakan varian produksi untuk membandingkan model, instans, dan kontainer, dan memilih kandidat berkinerja terbaik untuk menanggapi permintaan inferensi.

Dengan titik akhir SageMaker multi-varian, Anda dapat mendistribusikan permintaan pemanggilan titik akhir di beberapa varian produksi dengan menyediakan distribusi lalu lintas untuk setiap varian, atau Anda dapat memanggil varian tertentu secara langsung untuk setiap permintaan. Dalam topik ini, kita melihat kedua metode untuk menguji model ML.

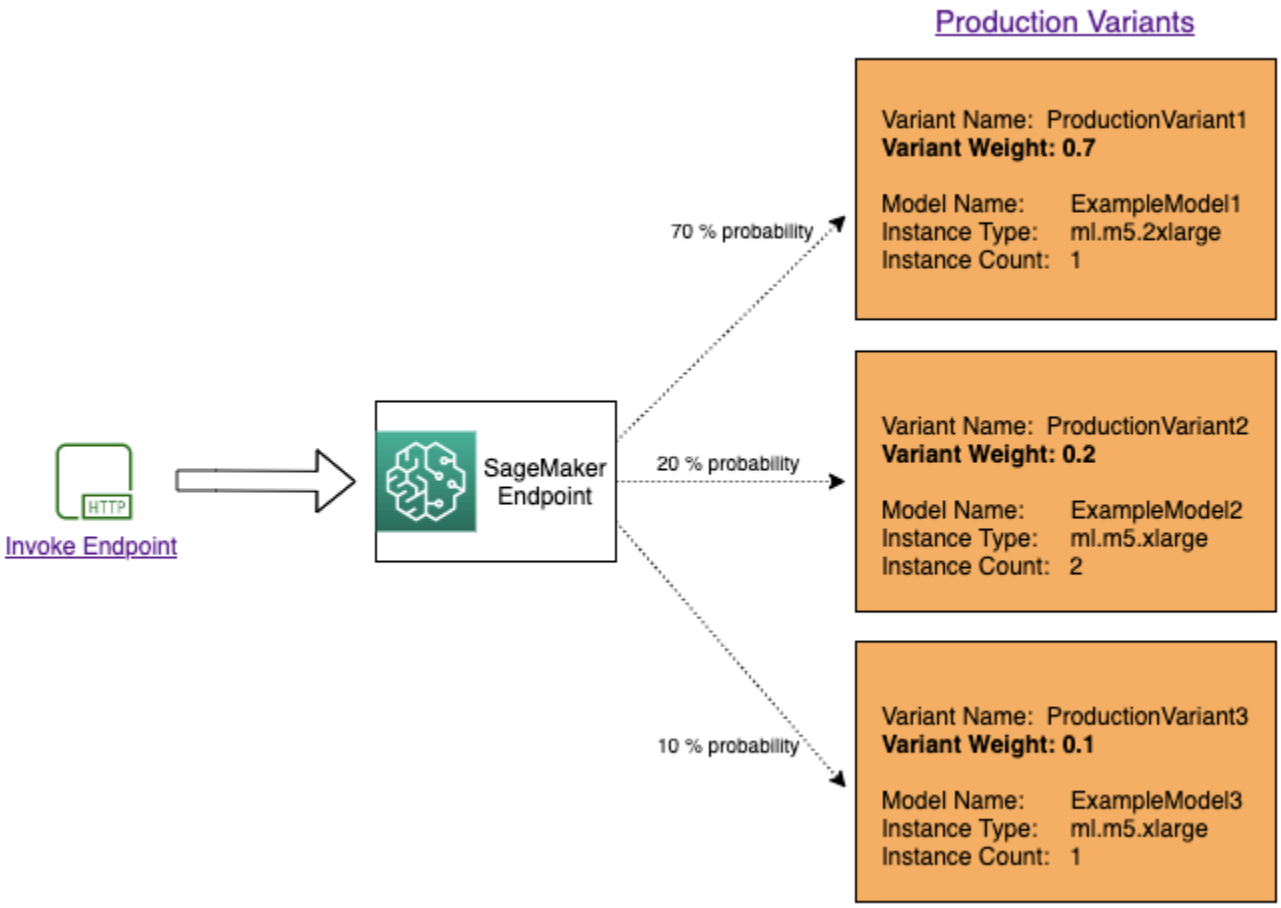
Topik

- Uji model dengan menentukan distribusi lalu lintas
- Uji model dengan menerapkan varian tertentu
- Model A/B contoh uji

Uji model dengan menentukan distribusi lalu lintas

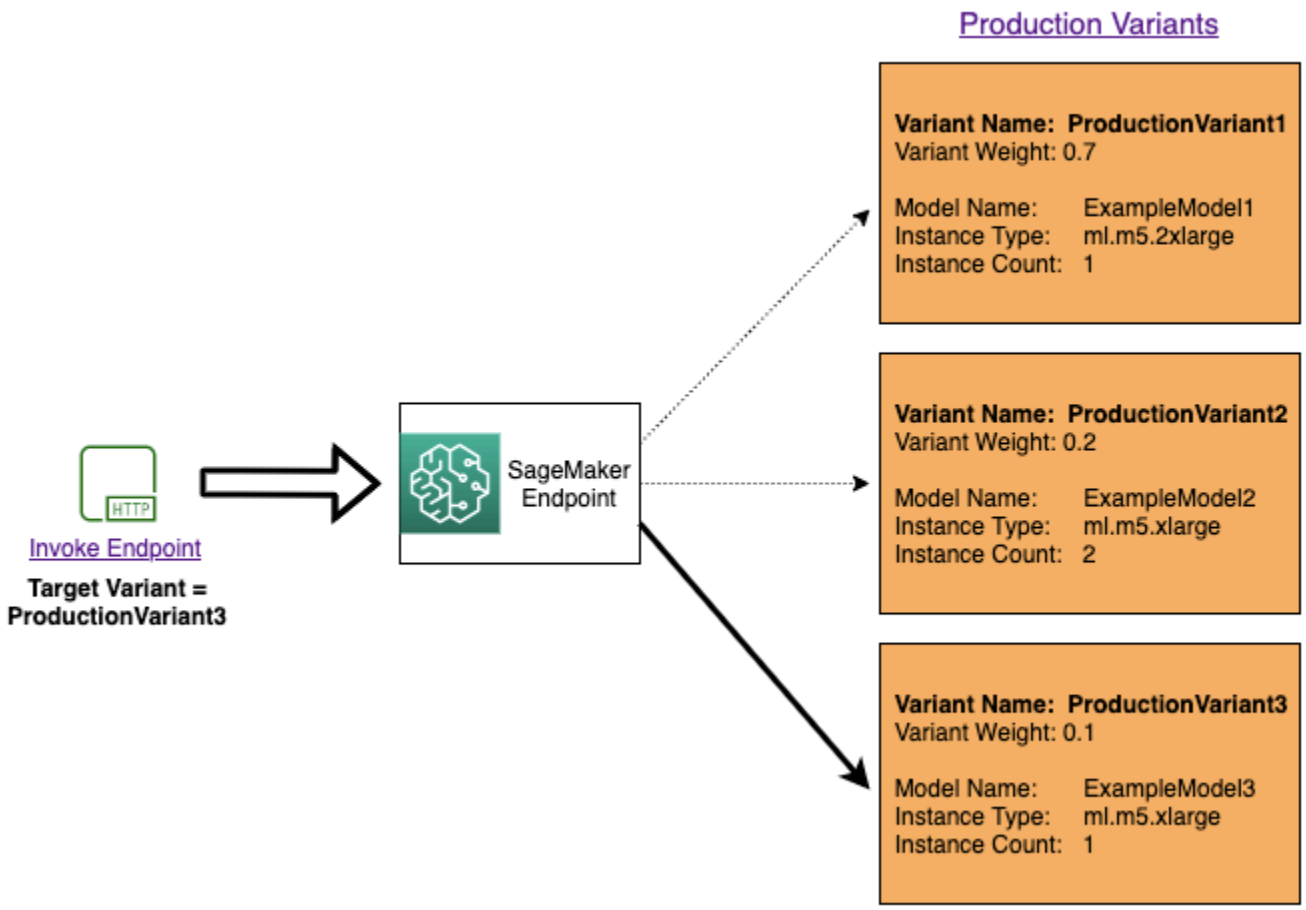
Untuk menguji beberapa model dengan mendistribusikan lalu lintas di antara keduanya, tentukan persentase lalu lintas yang dirutekan ke setiap model dengan menentukan bobot untuk setiap varian

produksi dalam konfigurasi titik akhir. Untuk informasi, lihat CreateEndpointConfig. Diagram berikut menunjukkan cara inii bekerja secara lebih rinci.



Uji model dengan menerapkan varian tertentu

Untuk menguji beberapa model dengan menerapkan model tertentu untuk setiap permintaan, tentukan versi spesifik model yang ingin Anda panggil dengan memberikan nilai untuk`TargetVariant` parameter saat Anda menelepon InvokeEndpoint. SageMaker memastikan bahwa permintaan diproses oleh varian produksi yang Anda tentukan. Jika Anda telah memberikan distribusi lalu lintas dan menentukan nilai untuk`TargetVariant` parameter, perutean yang ditargetkan akan menimpa distribusi lalu lintas acak. Diagram berikut menunjukkan cara inii bekerja secara lebih rinci.



Model A/B contoh uji

Melakukan pengujian A/B antara model baru dan model lama dengan lalu lintas produksi dapat menjadi langkah terakhir yang efektif dalam proses validasi untuk model baru. Dalam pengujian A/B, Anda menguji varian model yang berbeda dan membandingkan kinerja setiap varian. Jika versi model yang lebih baru memberikan kinerja yang lebih baik daripada versi yang ada sebelumnya, ganti versi lama model dengan versi baru dalam produksi.

Contoh berikut menunjukkan cara melakukan A/B Model Testing. Untuk contoh notebook yang mengimplementasikan contoh ini, lihat "Model Pengujian A/B dalam produksi.

Langkah 1: Buat dan deploy model

Pertama, kami mendefinisikan di mana model kami berada di Amazon S3. Lokasi ini digunakan saat kami menerapkan model kami dalam langkah selanjutnya:

```
model_url = f"s3://{path_to_model_1}"
model_url2 = f"s3://{path_to_model_2}"
```

Selanjutnya, kita membuat objek model dengan data gambar dan model. Objek model ini digunakan untuk menyebarkan varian produksi pada titik akhir. Model dikembangkan dengan melatih model ML pada kumpulan data yang berbeda, algoritme atau kerangka kerja yang berbeda, dan hyperparameter yang berbeda:

```
from sagemaker.amazon.amazon_estimator import get_image_uri

model_name = f"DEMO-xgb-churn-pred-{datetime.now():%Y-%m-%d-%H-%M-%S}"
model_name2 = f"DEMO-xgb-churn-pred2-{datetime.now():%Y-%m-%d-%H-%M-%S}"
image_uri = get_image_uri(boto3.Session().region_name, 'xgboost', '0.90-1')
image_uri2 = get_image_uri(boto3.Session().region_name, 'xgboost', '0.90-2')

sm_session.create_model(
    name=model_name,
    role=role,
    container_defs={
        'Image': image_uri,
        'ModelDataUrl': model_url
    }
)

sm_session.create_model(
    name=model_name2,
    role=role,
    container_defs={
        'Image': image_uri2,
        'ModelDataUrl': model_url2
    }
)
```

Kami sekarang membuat dua varian produksi, masing-masing dengan model dan persyaratan sumber daya yang berbeda (jenis dan jumlah instance). Hal ini memungkinkan Anda untuk juga menguji model pada jenis instans yang berbeda.

Kami menetapkan initial_weight 1 untuk kedua varian. Ini berarti bahwa 50% dari permintaan pergi ke`Variant1`, dan sisanya 50% dari permintaan untuk`Variant2`. Jumlah bobot di kedua varian adalah 2 dan setiap varian memiliki penetapan berat 1. Ini berarti bahwa setiap varian menerima 1/2, atau 50%, dari total lalu lintas.

```
from sagemaker.session import production_variant

variant1 = production_variant(
               model_name=model_name,
               instance_type="ml.m5.xlarge",
               initial_instance_count=1,
               variant_name='Variant1',
               initial_weight=1,
           )

variant2 = production_variant(
               model_name=model_name2,
               instance_type="ml.m5.xlarge",
               initial_instance_count=1,
               variant_name='Variant2',
               initial_weight=1,
           )
```

Akhirnya kita siap untuk menyebarkan varian produksi ini pada SageMaker titik akhir.

```
endpoint_name = f"DEMO-xgb-churn-pred-{datetime.now():%Y-%m-%d-%H-%M-%S}"
print(f"EndpointName={endpoint_name}")

sm_session.endpoint_from_production_variants(
    name=endpoint_name,
    production_variants=[variant1, variant2]
)
```

Langkah 2: Memanggil model yang dikerahkan

Sekarang kita mengirim permintaan ke endpoint ini untuk mendapatkan kesimpulan secara real time. Kami menggunakan distribusi lalu lintas dan penargetan langsung.

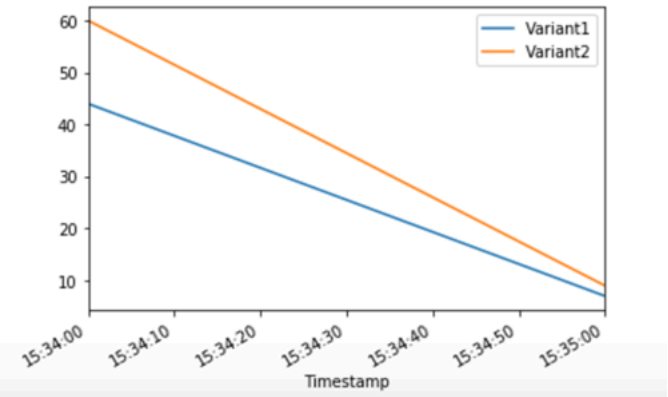
Pertama, kami menggunakan distribusi lalu lintas yang kami konfigurasikan di langkah sebelumnya. Setiap respons inferensi berisi nama varian produksi yang memproses permintaan, sehingga kita dapat melihat bahwa lalu lintas ke dua varian produksi kira-kira sama.

```
# get a subset of test data for a quick test
!tail -120 test_data/test-dataset-input-cols.csv > test_data/
test_sample_tail_input_cols.csv
```

```
print(f"Sending test traffic to the endpoint {endpoint_name}. \nPlease wait...")

with open('test_data/test_sample_tail_input_cols.csv', 'r') as f:
    for row in f:
        print(".", end="", flush=True)
        payload = row.rstrip('\n')
        sm_runtime.invoke_endpoint(
            EndpointName=endpoint_name,
            ContentType="text/csv",
            Body=payload
        )
        time.sleep(0.5)

print("Done!")
```

SageMaker memancarkan metrik sepertiLatency danInvocations untuk setiap varian di Amazon CloudWatch. Untuk daftar lengkap metrik yang SageMaker memancarkan, lihatPantau Amazon SageMaker dengan Amazon CloudWatch. Mari kita query CloudWatch untuk mendapatkan jumlah pemanggilan per varian, untuk menunjukkan bagaimana pemanggilan dibagi di varian secara default:

	Variant1	Variant2
Timestamp		
2020-06-05 15:34:00+00:00	44.0	60.0
2020-06-05 15:35:00+00:00	7.0	9.0



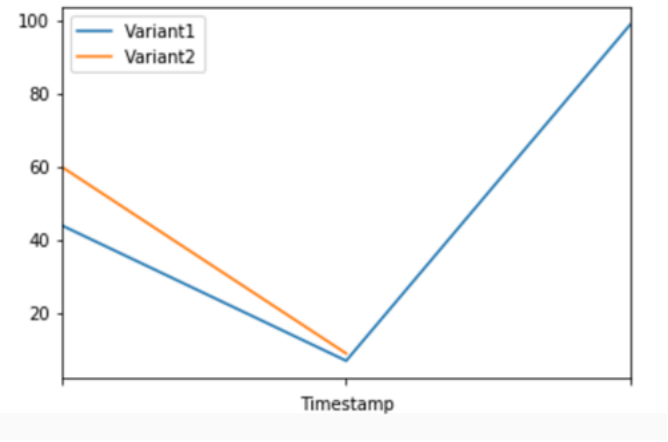
Sekarang mari kita memanggil versi spesifik dari model dengan menentukanVariant1 sebagaiTargetVariant dalam panggilan untukinvoke_endpoint.

```
print(f"Sending test traffic to the endpoint {endpoint_name}. \nPlease wait...")
with open('test_data/test_sample_tail_input_cols.csv', 'r') as f:
    for row in f:
```

```
        print(".", end="", flush=True)
        payload = row.rstrip('\n')
        sm_runtime.invoke_endpoint(
            EndpointName=endpoint_name,
            ContentType="text/csv",
            Body=payload,
            TargetVariant="Variant1"
        )
        time.sleep(0.5)
```

Untuk mengonfirmasi bahwa semua pemanggilan baru diproses oleh`Variant1`, kita dapat CloudWatch melakukan kueri untuk mendapatkan jumlah pemanggilan per varian. Kami melihat bahwa untuk pemanggilan terbaru (stempel waktu terbaru), semua permintaan diproses oleh`Variant1`, seperti yang telah kami tentukan. Tidak ada doa yang dibuat untuk`Variant2`.

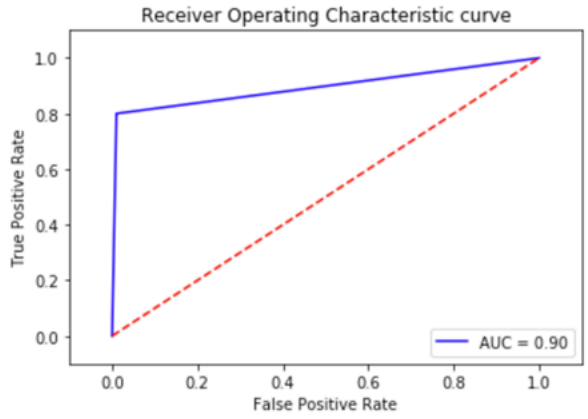
Timestamp	Variant1	Variant2
2020-06-05 15:34:00+00:00	44.0	60.0
2020-06-05 15:35:00+00:00	7.0	9.0
2020-06-05 15:36:00+00:00	99.0	NaN



Langkah 3: Evaluasi Performa Model

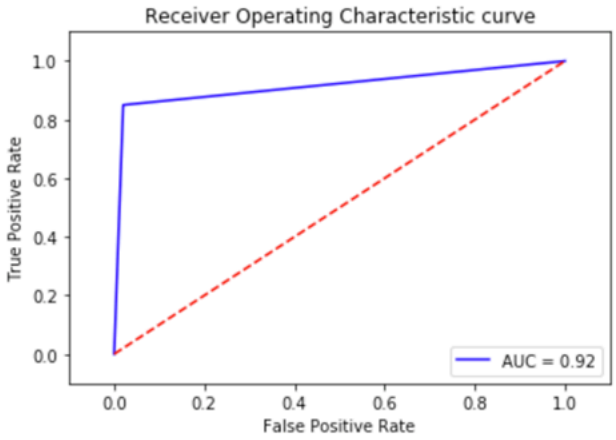
Untuk melihat versi model mana yang berkinerja lebih baik, mari kita evaluasi akurasi, presisi, penarikan, skor F1, dan Karaktersi/area operasi Penerima di bawah kurva untuk setiap varian. Pertama, mari kita lihat metrik ini untuk`Variant1`:

```
Accuracy: 0.9583333333333334
Precision: 0.9411764705882353
Recall: 0.8
F1 Score: 0.8648648648648648
AUC is 0.895
```



Sekarang mari kita lihat metrik untuk`Variant2`:

```
Accuracy: 0.9583333333333334
Precision: 0.8947368421052632
Recall: 0.85
F1 Score: 0.8717948717948718
AUC is 0.915
```

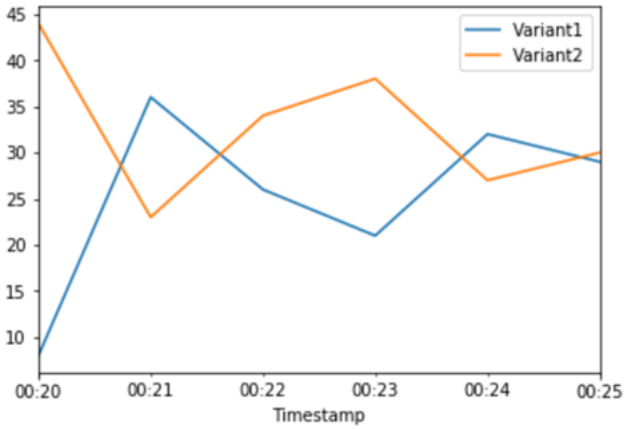


Untuk sebagian besar metrik kami`Variant2` didefinisikan, berkinerja lebih baik, jadi ini adalah salah satu yang ingin kita gunakan dalam produksi.

Langkah 4: Tingkatkan lalu lintas ke model terbaik

Sekarang kita telah menentukan bahwa`Variant2` melakukan lebih baik daripada`Variant1`, kita menggeser lebih banyak lalu lintas untuk itu. Kita dapat terus menggunakan`TargetVariant` untuk memanggil varian model tertentu, tetapi pendekatan yang lebih sederhana adalah memperbarui bobot yang ditetapkan untuk setiap varian dengan menelepon

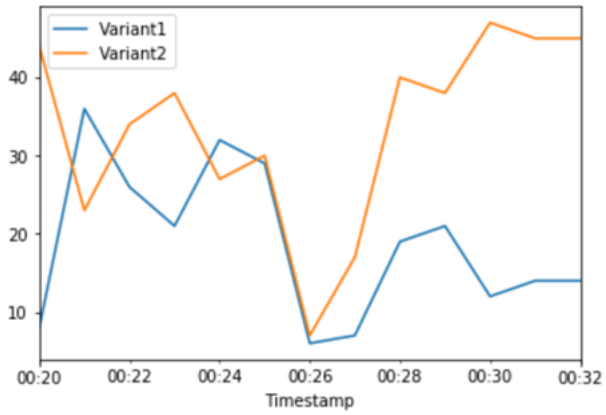
UpdateEndpointWeightsAndCapacities. Ini mengubah distribusi lalu lintas ke varian produksi Anda tanpa memerlukan pembaruan pada titik akhir Anda. Ingat dari bagian pengaturan yang kami tetapkan bobot varian untuk membagi lalu lintas 50/50. CloudWatch Metrik untuk pemanggilan total untuk setiap varian di bawah ini menunjukkan pola pemanggilan untuk setiap varian:



Sekarang kita menggeser 75% lalu lintas ke`Variant2` dengan menetapkan bobot baru untuk setiap varian menggunakan`UpdateEndpointWeightsAndCapacities`. SageMaker sekarang mengirimkan 75% dari permintaan inferensi ke`Variant2` dan sisanya 25% dari permintaan ke`Variant1`.

```
sm.update_endpoint_weights_and_capacities(
    EndpointName=endpoint_name,
    DesiredWeightsAndCapacities=[
        {
            "DesiredWeight": 25,
            "VariantName": variant1["VariantName"]
        },
        {
            "DesiredWeight": 75,
            "VariantName": variant2["VariantName"]
        }
    ]
)
```

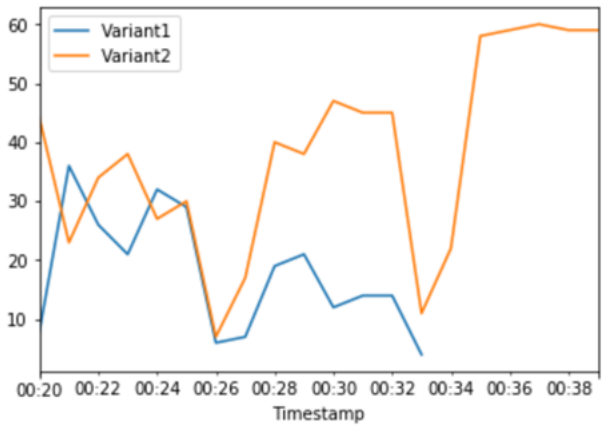
CloudWatch Metrik untuk pemanggilan total untuk setiap varian menunjukkan kepada kita doa yang lebih tinggi`Variant2` daripada untuk`Variant1`:



Kami dapat terus memantau metrik kami, dan ketika kami puas dengan kinerja varian, kami dapat merutekan 100% lalu lintas ke varian itu. Kami gunakan [UpdateEndpointWeightsAndCapacities]untuk memperbarui tugas lalu lintas untuk varian. Bobot untuk`Variant1` diatur ke 0 dan bobot untuk`Variant2` diatur ke 1. SageMaker sekarang mengirimkan 100% dari semua permintaan inferensi ke`Variant2`.

```
sm.update_endpoint_weights_and_capacities(
    EndpointName=endpoint_name,
    DesiredWeightsAndCapacities=[
        {
            "DesiredWeight": 0,
            "VariantName": variant1["VariantName"]
        },
        {
            "DesiredWeight": 1,
            "VariantName": variant2["VariantName"]
        }
    ]
)
```

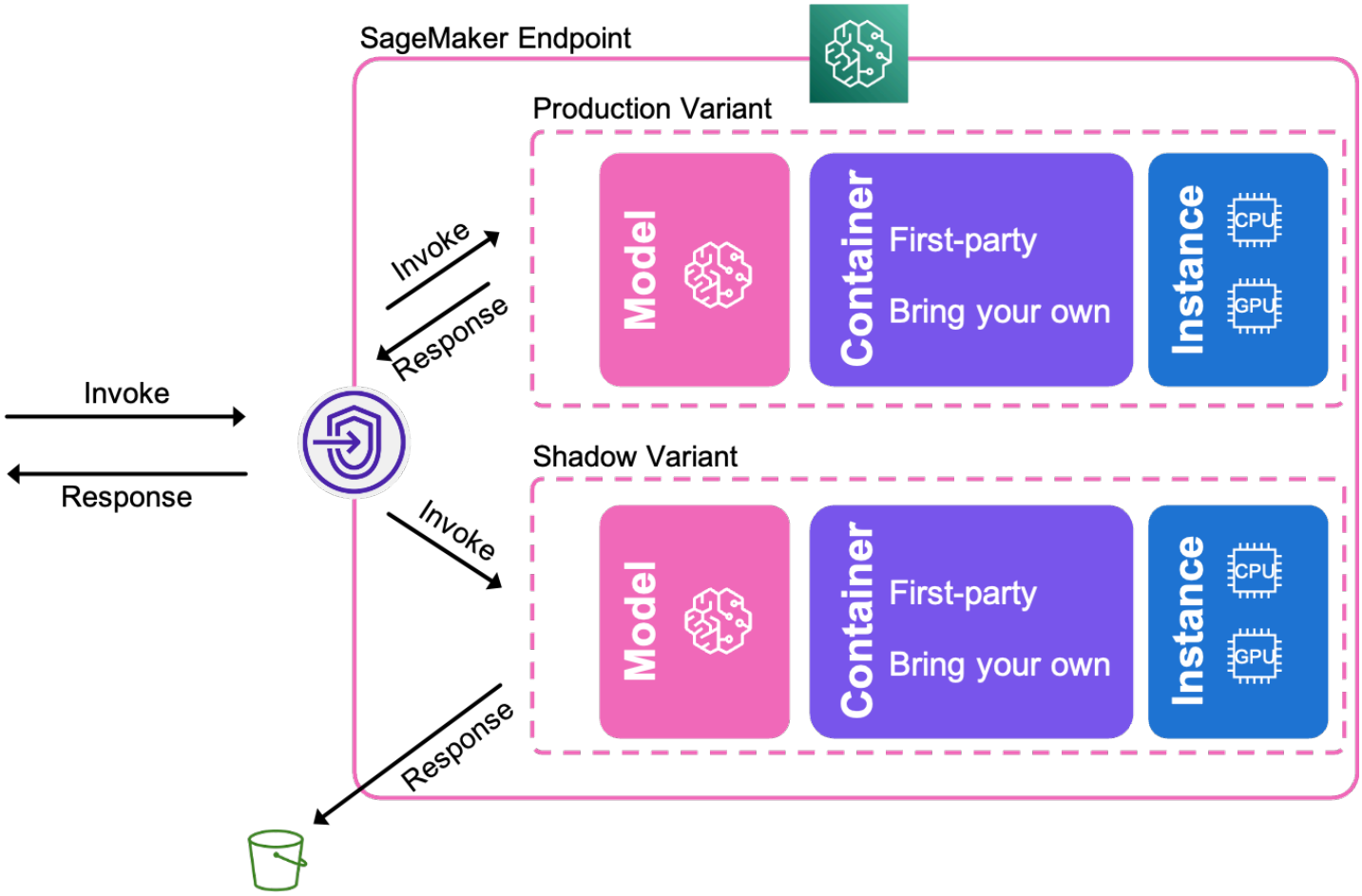
CloudWatch Metrik untuk total pemanggilan untuk setiap varian menunjukkan bahwa semua permintaan inferensi sedang diproses oleh`Variant2` dan tidak ada permintaan inferensi yang diproses oleh`Variant1`.



Sekarang Anda dapat memperbarui titik akhir Anda dengan aman dan menghapus`Variant1` dari titik akhir Anda. Anda juga dapat terus menguji model baru dalam produksi dengan menambahkan varian baru ke titik akhir Anda dan mengikuti langkah 2 - 4.

Varian bayangan angan angan angan angan

Anda dapat menggunakan SageMaker Model Shadow Deployment untuk membuat varian bayangan yang berjalan lama untuk memvalidasi komponen kandidat baru dari tumpukan penyajian model Anda sebelum mempromosikannya ke produksi. Diagram berikut ini menunjukkan cara varian bayangan berikut ini menunjukkan cara varian bayangan berikut ini menunjukkan cara varian bayangan berikut ini menunjukkan cara varian bayangan angan



Menyebarkan varian bayangan angan angan angan angan angan angan angan

Contoh kode berikut ini menunjukkan cara Anda dapat menerapkan varian bayangan berikut ini menunjukkan cara Anda memprogram varian bayangan angan berikut ini menunjukkan cara Anda menerapkan varian bayangan berikut ini menunjukkan cara Anda melakukan Ganti *teks placeholder pengguna* dalam contoh dengan informasi Anda sendiri.

1. Buat dua SageMaker model: satu untuk varian produksi Anda, dan satu untuk varian bayangan Anda.

```
import boto3
from sagemaker import get_execution_role, Session

aws_region = "aws-region"

boto_session = boto3.Session(region_name=aws_region)
sagemaker_client = boto_session.client("sagemaker")
```

```
role = get_execution_role()

bucket = Session(boto_session).default_bucket()

model_name1 = "name-of-your-first-model"
model_name2 = "name-of-your-second-model"

sagemaker_client.create_model(
    ModelName = model_name1,
    ExecutionRoleArn = role,
    Containers=[
        {
            "Image": "ecr-image-uri-for-first-model",
            "ModelDataUrl": "s3-location-of-trained-first-model"
        }
    ]
)

sagemaker_client.create_model(
    ModelName = model_name2,
    ExecutionRoleArn = role,
    Containers=[
        {
            "Image": "ecr-image-uri-for-second-model",
            "ModelDataUrl": "s3-location-of-trained-second-model"
        }
    ]
)
```

2. Membuat konfigurasi titik akhir akhir titik akhir akhir ini menunjukkan cara titik akhir akhir ini Tentukan varian produksi dan bayangan Anda dalam konfigurasi.

```
endpoint_config_name = name-of-your-endpoint-config

create_endpoint_config_response = sagemaker_client.create_endpoint_config(
    EndpointConfigName=endpoint_config_name,
    ProductionVariants=[
        {
            "VariantName": name-of-your-production-variant,
            "ModelName": model_name1,
            "InstanceType": "ml.m5.xlarge",
            "InitialInstanceCount": 1,
```

```
            "InitialVariantWeight": 1,
        }
    ],
    ShadowProductionVariants=[
        {
            "VariantName": name-of-your-shadow-variant,
            "ModelName": model_name2,
            "InstanceType": "ml.m5.xlarge",
            "InitialInstanceCount": 1,
            "InitialVariantWeight": 1,
        }
   ]
)
```

3. Membuat titik akhir akhir akhir akhir akhir ini menunjukkan titik akhir akhir ini

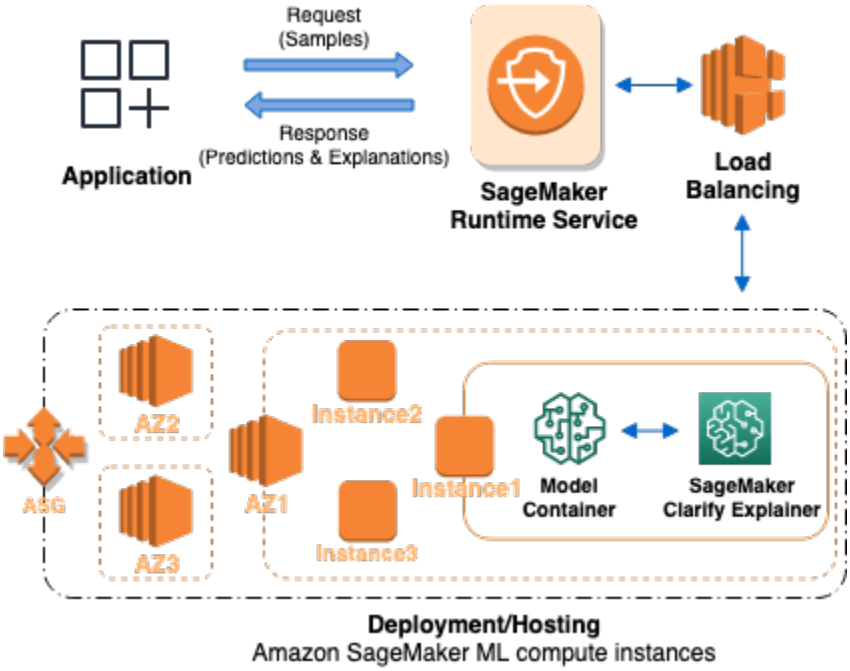
```
create_endpoint_response = sm.create_endpoint(
    EndpointName=name-of-your-endpoint,
    EndpointConfigName=endpoint_config_name,
)
```

Penjelasan Online dengan Clarify SageMaker

Panduan ini menunjukkan cara mengonfigurasi penjelasan online dengan SageMaker Clarify. Dengan titik akhir inferensi SageMaker waktu nyata, Anda dapat menganalisis penjelasan secara real time, terus menerus. Fungsi penjelasan online cocok dengan bagian Deploy to production dari alur kerja Amazon SageMaker Machine Learning.

Bagaimana Klarifikasi Penjelasan Online Bekerja

Grafik berikut menggambarkan SageMaker arsitektur untuk hosting endpoint yang melayani permintaan penjelasan. Ini menggambarkan interaksi antara titik akhir, wadah model, dan SageMaker penjelasan Clarify.



Inilah cara kerja Clarify online explainability. Aplikasi mengirimkan `InvokeEndpoint` permintaan gaya REST ke SageMaker Runtime Service. Layanan merutekan permintaan ini ke SageMaker titik akhir untuk mendapatkan prediksi dan penjelasan. Kemudian, layanan menerima respons dari titik akhir. Terakhir, layanan mengirimkan respons kembali ke aplikasi.

Untuk meningkatkan ketersediaan titik akhir, SageMaker secara otomatis mencoba mendistribusikan instance endpoint di beberapa Availability Zone, sesuai dengan jumlah instance dalam konfigurasi endpoint. Pada instance titik akhir, atas permintaan penjelasan baru, penjelasan SageMaker Clarify memanggil wadah model untuk prediksi. Kemudian menghitung dan mengembalikan atribusi fitur.

Berikut adalah empat langkah untuk membuat endpoint yang menggunakan SageMaker Clarify online explainability:

1. Periksa apakah SageMaker model pra-pelatihan Anda kompatibel dengan penjelasan online dengan mengikuti langkah-langkah pra-pemeriksaan.
2. Buat konfigurasi endpoint dengan konfigurasi SageMaker Clarify explainer menggunakan API. `CreateEndpointConfig`
3. Buat titik akhir dan berikan konfigurasi titik akhir untuk SageMaker menggunakan API. `CreateEndpoint` Layanan meluncurkan instance komputasi ML dan menerapkan model seperti yang ditentukan dalam konfigurasi.

4. Memanggil endpoint: Setelah endpoint dalam layanan, panggil SageMaker Runtime API `InvokeEndpoint` untuk mengirim permintaan ke endpoint. Titik akhir kemudian mengembalikan penjelasan dan prediksi.

Pra-periksa wadah model

Bagian ini menunjukkan cara memeriksa terlebih dahulu input dan output wadah model untuk kompatibilitas sebelum mengonfigurasi titik akhir. SageMaker Penjelasan Clarify adalah model agnostik, tetapi memiliki persyaratan untuk input dan output wadah model.

> **Note**
>
> Anda dapat meningkatkan efisiensi dengan mengonfigurasi penampung untuk mendukung permintaan batch, yang mendukung dua atau lebih catatan dalam satu permintaan. Misalnya, catatan tunggal adalah satu baris data CSV, atau satu baris data JSON Lines. SageMaker Clarify akan mencoba mengirim kumpulan catatan mini ke wadah model terlebih dahulu sebelum kembali ke permintaan rekaman tunggal.

Masukan wadah model

CSV

Wadah model mendukung input dalam CSV dengan tipe MIME:. `text/csv` Tabel berikut menunjukkan contoh input yang mendukung SageMaker Clarify.

Masukan wadah model (representasi string)	Komentar
'1,2,3,4'	Rekaman tunggal yang menggunakan empat fitur numerik.
'1,2,3,4\n5,6,7,8'	Dua catatan, dipisahkan oleh jeda baris '\n'.
“Ini adalah produk yang bagus”, 5'	Rekaman tunggal yang berisi fitur teks dan fitur numerik.
'"Ini adalah produk yang bagus” ,5\n“Peng alaman belanja yang buruk”, 1 '	Dua catatan.

JSON Lines

SageMaker juga mendukung input dalam format padat JSON Lines dengan tipe MIME:`application/jsonlines`, seperti yang ditunjukkan pada tabel berikut.

Masukan wadah model	Komentar
'{"data": {"features": [1,2,3,4]}}'	Rekaman tunggal; daftar fitur dapat diekstraksi dengan ekspresi JMESPath. `data.features`
'{"data": {"features": [1,2,3,4]}}\n{"data": {"features": [5,6,7,8]}}'	Dua catatan.
'{"features": ["Ini adalah produk yang bagus" ,5]}'	Rekaman tunggal; daftar fitur dapat diekstraksi dengan ekspresi JMESPath. `features`
'{"features": ["Ini adalah produk yang bagus" ,5]}\n{"features": ["Pengalaman belanja yang buruk" ,1]}'	Dua catatan.

Output wadah model

Output wadah model Anda juga harus dalam format padat CSV, atau JSON Lines. Selain itu wadah model harus menyertakan probabilitas catatan input, yang digunakan SageMaker Clarify untuk menghitung atribusi fitur.

Contoh data berikut adalah untuk output wadah model dalam format CSV.

Probability only

Untuk masalah regresi dan klasifikasi biner, wadah model mengeluarkan nilai probabilitas tunggal (skor) dari label yang diprediksi. Probabilitas ini dapat diekstraksi menggunakan indeks kolom 0. Untuk masalah multi-kelas, wadah model mengeluarkan daftar probabilitas (skor). Untuk masalah multi-kelas, jika tidak ada indeks yang disediakan, semua nilai diekstraksi.

Masukan wadah model	Output wadah model (representasi string)
Rekaman tunggal	'0,6'

Masukan wadah model	Output wadah model (representasi string)
Dua catatan (hasil dalam satu baris)	'0,6,0,3'
Dua catatan (menghasilkan dua baris)	'0,6\n0,3'
Rekaman tunggal model multi-kelas (tiga kelas)	'0.1,0.6,0.3'
Dua catatan model multi-kelas (tiga kelas)	'0.1,0.6,0.3\n0.2,0.5,0.3'

Predicted label and probabilities

Wadah model mengeluarkan label yang diprediksi diikuti oleh probabilitasnya dalam format CSV. Probabilitas dapat diekstraksi menggunakan indeks. 1

Masukan wadah model	Output wadah model
Rekaman tunggal	'1,0.6'
Dua catatan	'1,0.6\n0,0.3'

Predicted labels header and probabilities

Wadah model multi-kelas yang dilatih oleh Autopilot dapat dikonfigurasi untuk menampilkan representasi string dari daftar label dan probabilitas yang diprediksi dalam format CSV. Dalam contoh berikut, probabilitas dapat diekstraksi dengan indeks. 1 Header label dapat diekstraksi dengan indeks1, dan header label dapat diekstraksi menggunakan indeks. `0`

Masukan wadah model	Output wadah model
Rekaman tunggal	'"[\ 'cat\',\ 'dog\ ',\ 'ikan\']", "[0.1,0.6,0.3]" '
Dua catatan	'"[\ 'cat\',\ 'dog\ ',\ 'ikan\']", "[0.1,0.6,0.3]"\n"[\ 'cat\ ',\ 'dog\',\ 'ikan\ ']", "[0.2,0.5,0.3]"'

Contoh data berikut adalah untuk output wadah model dalam format JSON Lines.

Probability only

Dalam contoh ini, wadah model mengeluarkan probabilitas yang dapat diekstraksi dengan `JMESPath`ekspresi `score` dalam format JSON Lines.

Masukan wadah model	Output wadah model
Rekaman tunggal	'{"skor” :0.6}'
Dua catatan	'{"score” :0.6}\n{"skor” :0.3}'

Predicted label and probabilities

Dalam contoh ini, wadah model multi-kelas mengeluarkan daftar header label bersama dengan daftar probabilitas dalam format JSON Lines. Probabilitas dapat diekstraksi dengan `JMESPath` ekspresi`probability`, dan header label dapat diekstraksi dengan ekspresi. `JMESPath` `predicted labels`

Masukan wadah model	Output wadah model
Rekaman tunggal	'{"predicted_labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.1,0.6,0.3]}'
Dua catatan	'{"predicted_labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.1,0.6,0.3]}\n{"predicted _labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.2,0.5,0.3]}'

Predicted labels header and probabilities

Dalam contoh ini, wadah model multi-kelas mengeluarkan daftar header label dan probabilitas dalam format JSON Lines. Probabilitas dapat diekstraksi dengan `JMESPath` ekspresi`probability`, dan header label dapat diekstraksi dengan ekspresi. `JMESPath` `predicted labels`

Masukan wadah model	Output wadah model
Rekaman tunggal	'{"predicted_labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.1,0.6,0.3]}'
Dua catatan	'{"predicted_labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.1,0.6,0.3]}\n{"predicted _labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.2,0.5,0.3]}'

Validasi wadah model

Kami menyarankan Anda menerapkan model Anda ke titik akhir inferensi SageMaker real-time, dan mengirim permintaan ke titik akhir. Periksa permintaan (input wadah model) dan respons (keluaran wadah model) secara manual untuk memastikan bahwa keduanya sesuai dengan persyaratan di bagian Input Penampung Model dan bagian Output Penampung Model. Jika wadah model Anda mendukung permintaan batch, Anda dapat memulai dengan satu permintaan rekaman, lalu mencoba dua atau lebih catatan.

Perintah berikut menunjukkan cara meminta respons menggunakanAWS CLI. AWS CLIIni sudah diinstal sebelumnya di SageMaker Studio Classic, dan instance SageMaker Notebook. Jika Anda perlu menginstalAWS CLI, ikuti panduan instalasi ini.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name $ENDPOINT_NAME \
  --content-type $CONTENT_TYPE \
  --accept $ACCEPT_TYPE \
  --body $REQUEST_DATA \
  $CLI_BINARY_FORMAT \
  /dev/stderr 1>/dev/null
```

Parameter didefinisikan, sebagai berikut:

- `$ENDPOINT_NAME`: Nama titik akhir.
- `$CONTENT_TYPE`: Jenis permintaan MIME (input wadah model).
- `$ACCEPT_TYPE`: Jenis respons MIME (keluaran wadah model).
- `$REQUEST_DATA`: String payload yang diminta.

- `$CLI_BINARY_FORMAT`: Format parameter antarmuka baris perintah (CLI). Untuk AWS CLI v1, parameter ini harus tetap kosong. Untuk v2, parameter ini harus diatur ke`--cli-binary-format raw-in-base64-out`.

> **Note**
>
> AWS CLIv2 melewati parameter biner sebagai string yang dikodekan base64 default.

Contoh berikut menggunakan AWS CLI v1:

Request and response in CSV format

- Permintaan terdiri dari satu catatan dan responsnya adalah nilai probabilitasnya.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-sagemaker-xgboost-model \
  --content-type text/csv \
  --accept text/csv \
  --body '1,2,3,4' \
  /dev/stderr 1>/dev/null
```

  Output:

```
0.6
```

- Permintaan terdiri dari dua catatan, dan responsnya mencakup probabilitasnya, dan model memisahkan probabilitas dengan koma. `$'content'`Ekspresi dalam `--body` memberitahu perintah untuk menafsirkan \n dalam konten sebagai jeda baris.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-sagemaker-xgboost-model \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

  Output:

```
0.6,0.3
```

- Permintaan terdiri dari dua catatan, respons mencakup probabilitasnya, dan model memisahkan probabilitas dengan jeda baris.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-1 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Output:

```
0.6

0.3
```

- Permintaan terdiri dari satu catatan, dan responsnya adalah nilai probabilitas (model multi-kelas, tiga kelas).

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-1 \
  --content-type text/csv \
  --accept text/csv \
  --body '1,2,3,4' \
  /dev/stderr 1>/dev/null
```

Output:

```
0.1,0.6,0.3
```

- Permintaan terdiri dari dua catatan, dan responsnya mencakup nilai probabilitasnya (model multi-kelas, tiga kelas).

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-1 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Output:

```
0.1,0.6,0.3

0.2,0.5,0.3
```

- Permintaan terdiri dari dua catatan, dan responsnya mencakup label dan probabilitas yang diprediksi.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-2 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Output:

```
1,0.6

0,0.3
```

- Permintaan terdiri dari dua catatan dan responsnya mencakup header label dan probabilitas.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-3 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Output:

```
"['cat','dog','fish']","[0.1,0.6,0.3]"

"['cat','dog','fish']","[0.2,0.5,0.3]"
```

Request and response in JSON Lines format

- Permintaan terdiri dari satu catatan dan responsnya adalah nilai probabilitasnya.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-jsonlines \
```

```
  --content-type application/jsonlines \
  --accept application/jsonlines \
  --body '{"features":["This is a good product",5]}' \
  /dev/stderr 1>/dev/null
```

Output:

```
{"score":0.6}
```

- Permintaan berisi dua catatan, dan responsnya mencakup label dan probabilitas yang diprediksi.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-jsonlines-2 \
  --content-type application/jsonlines \
  --accept application/jsonlines \
  --body $'{"features":[1,2,3,4]}\n{"features":[5,6,7,8]}' \
  /dev/stderr 1>/dev/null
```

Output:

```
{"predicted_label":1,"probability":0.6}

{"predicted_label":0,"probability":0.3}
```

- Permintaan berisi dua catatan dan responsnya mencakup header label dan probabilitas.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-jsonlines-3 \
  --content-type application/jsonlines \
  --accept application/jsonlines \
  --body $'{"data":{"features":[1,2,3,4]}}\n{"data":{"features":[5,6,7,8]}}' \
  /dev/stderr 1>/dev/null
```

Output:

```
{"predicted_labels":["cat","dog","fish"],"probabilities":
[0.1,0.6,0.3]}

{"predicted_labels":["cat","dog","fish"],"probabilities":
[0.2,0.5,0.3]}
```

Request and response in different formats

- Permintaan dalam format CSV dan responsnya dalam format JSON Lines:

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-in-jsonlines-out \
  --content-type text/csv \
  --accept application/jsonlines \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

  Output:

```
{"probability":0.6}

{"probability":0.3}
```

- Permintaan dalam format JSON Lines dan responsnya dalam format CSV:

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-jsonlines-in-csv-out \
  --content-type application/jsonlines \
  --accept text/csv \
  --body $'{"features":[1,2,3,4]}\n{"features":[5,6,7,8]}' \
  /dev/stderr 1>/dev/null
```

  Output:

```
0.6

0.3
```

Setelah validasi selesai, hapus titik akhir pengujian.

Konfigurasikan dan buat titik akhir

Buat konfigurasi titik akhir baru agar sesuai dengan model Anda, dan gunakan konfigurasi ini untuk membuat titik akhir. Anda dapat menggunakan wadah model yang divalidasi pada langkah pra-pemeriksaan untuk membuat titik akhir dan mengaktifkan fitur SageMaker Clarify online explainability.

Gunakan `sagemaker_client` objek untuk membuat titik akhir menggunakan CreateEndpointConfigAPI. Atur anggota `ClarifyExplainerConfig` di dalam `ExplainerConfig` parameter sebagai berikut:

```
sagemaker_client.create_endpoint_config(
    EndpointConfigName='name-of-your-endpoint-config',
    ExplainerConfig={
        'ClarifyExplainerConfig': {
            'EnableExplanations': '`true`',
            'InferenceConfig': {
                ...
            },
            'ShapConfig': {
                ...
            }
        },
    },
    ProductionVariants=[{
        'VariantName': 'AllTraffic',
        'ModelName': 'name-of-your-model',
        'InitialInstanceCount': 1,
        'InstanceType': 'ml.m5.xlarge',
    }]
     ...
)
sagemaker_client.create_endpoint(
    EndpointName='name-of-your-endpoint',
    EndpointConfigName='name-of-your-endpoint-config'
)
```

Panggilan pertama ke `sagemaker_client` objek membuat konfigurasi titik akhir baru dengan fitur penjelasan diaktifkan. Panggilan kedua menggunakan konfigurasi titik akhir untuk meluncurkan titik akhir.

> **Note**
>
> Anda juga dapat meng-host beberapa model dalam satu wadah di belakang titik akhir multi-model inferensi SageMaker real-time dan mengonfigurasi penjelasan online dengan Clarify. SageMaker

EnableExplanationsEkspresi

`EnableExplanations`Parameternya adalah string ekspresi `JMESPath`Boolean. Ini dievaluasi untuk setiap catatan dalam permintaan penjelasan. Jika parameter ini dievaluasi menjadi benar, maka catatan akan dijelaskan. Jika parameter ini dievaluasi salah, maka penjelasan tidak akan dihasilkan.

SageMaker Klarifikasi deserialisasi output wadah model untuk setiap catatan ke dalam struktur data yang kompatibel dengan JSON, dan kemudian menggunakan `EnableExplanations` parameter untuk mengevaluasi data.

> **Catatan**
>
> Ada dua opsi untuk catatan tergantung pada format output wadah model.
>
> - Jika keluaran wadah model dalam format CSV, maka catatan dimuat sebagai array JSON.
> - Jika keluaran wadah model dalam format JSON Lines, maka catatan dimuat sebagai objek JSON.

`EnableExplanations`Parameter adalah ekspresi JMESPath yang dapat diteruskan baik selama `InvokeEndpoint` atau `CreateEndpointConfig` operasi. Jika ekspresi JMESPath yang Anda berikan tidak valid, pembuatan endpoint akan gagal. Jika ekspresi valid, tetapi hasil evaluasi ekspresi tidak terduga, maka titik akhir akan berhasil dibuat, tetapi kesalahan akan dihasilkan ketika titik akhir dipanggil. Uji `EnableExplanations` ekspresi Anda dengan menggunakan `InvokeEndpoint` API, lalu terapkan ke konfigurasi titik akhir.

Berikut ini adalah beberapa contoh `EnableExplanations` ekspresi yang valid. Dalam contoh, ekspresi JMESPath melampirkan literal menggunakan karakter backtick. Misalnya, `` `true` `` berarti benar.

Ekspresi (representasi string)	Output wadah model (representasi string)	Hasil evaluasi (Boolean)	Arti
``benar``	(N/A)	Benar	Aktifkan penjelasan online tanpa syarat.
``salah``	(N/A)	Salah	Nonaktifkan penjelasan online tanpa syarat.

Ekspresi (representasi string)	Output wadah model (representasi string)	Hasil evaluasi (Boolean)	Arti
'[1] >`0.5`'	'1,0.6'	Benar	Untuk setiap catatan, wadah model mengeluarkan label dan probabilitas yang diprediksi. Menjelaskan catatan jika probabilitasnya (pada indeks 1) lebih besar dari 0,5.
'probabilitas>`0,5 `'	'{"predicted_label” :1, "probabilitas” :0.6}'	Benar	Untuk setiap catatan, wadah model mengeluarkan data JSON. Jelaskan catatan jika probabili tasnya lebih besar dari 0,5.
'! berisi (probabilitas [:-1], maks (probabil itas)) '	'{"probabilities”: [0.4, 0.1, 0.4], “labels”: ["cat”, "dog”, "fish "]}'	Salah	Untuk model multi-kelas: Menjelaskan catatan jika label yang diprediksi (kelas yang memiliki nilai probabili tas maks) adalah kelas terakhir. Secara harfiah, ekspresi berarti bahwa nilai probabilitas maks tidak ada dalam daftar probabilitas tidak termasuk yang terakhir.

Dataset sintetis

SageMaker Clarify menggunakan algoritma Kernel SHAP. Diberikan catatan (juga disebut sampel atau instance) dan konfigurasi SHAP, penjelasan pertama-tama menghasilkan kumpulan data sintetis. SageMaker Klarifikasi kemudian kueri wadah model untuk prediksi kumpulan data, lalu hitung dan kembalikan atribusi fitur. Ukuran kumpulan data sintetis memengaruhi runtime untuk penjelasan Clarify. Kumpulan data sintetis yang lebih besar membutuhkan lebih banyak waktu untuk mendapatkan prediksi model daripada yang lebih kecil.

Ukuran dataset sintetis ditentukan oleh rumus berikut:

```
Synthetic dataset size = SHAP baseline size * n_samples
```

Ukuran dasar SHAP adalah jumlah catatan dalam data dasar SHAP. Informasi ini diambil dari`ShapBaselineConfig`.

Ukuran `n_samples` diatur oleh parameter `NumberOfSamples` dalam konfigurasi explainer dan jumlah fitur. Jika jumlah fitur adalah`n_features`, maka `n_samples` adalah sebagai berikut:

```
n_samples = MIN(NumberOfSamples, 2^n_features - 2)
```

Berikut ini menunjukkan `n_samples` jika `NumberOfSamples` tidak disediakan.

```
n_samples = MIN(2*n_features + 2^11, 2^n_features - 2)
```

Misalnya, catatan tabular dengan 10 fitur memiliki ukuran dasar SHAP 1. Jika tidak `NumberOfSamples` disediakan, dataset sintetis berisi 1022 catatan. Jika catatan memiliki 20 fitur, dataset sintetis berisi 2088 catatan.

Untuk masalah NLP, `n_features` sama dengan jumlah fitur non-teks ditambah jumlah unit teks.

> **Note**
>
> `InvokeEndpoint`API memiliki batas waktu tunggu permintaan. Jika kumpulan data sintetis terlalu besar, penjelasan mungkin tidak dapat menyelesaikan perhitungan dalam batas ini. Jika perlu, gunakan informasi sebelumnya untuk memahami dan mengurangi ukuran dasar SHAP dan. `NumberOfSamples` Jika wadah model Anda diatur untuk menangani permintaan batch, maka Anda juga dapat menyesuaikan nilai`MaxRecordCount`.

Memanggil titik akhir

Setelah titik akhir berjalan, gunakan SageMaker Runtime InvokeEndpointAPI di layanan SageMaker Runtime untuk mengirim permintaan, atau memanggil titik akhir. Sebagai tanggapan, permintaan ditangani sebagai permintaan penjelasan oleh Clarify explainer. SageMaker

> **Note**
>
> Untuk memanggil titik akhir, pilih salah satu opsi berikut:
>
> - Untuk petunjuk menggunakan Boto3 atau AWS CLI untuk memanggil titik akhir, lihat. Memanggil model untuk inferensi waktu nyata
> - Untuk menggunakan SageMaker SDK for Python untuk menjalankan endpoint, lihat Predictor API.

Permintaan

`InvokeEndpoint`API memiliki parameter opsional`EnableExplanations`, yang dipetakan ke header `X-Amzn-SageMaker-Enable-Explanations` HTTP. Jika parameter ini disediakan, itu mengesampingkan `EnableExplanations` parameter. `ClarifyExplainerConfig`

> **Note**
>
> `Accept`Parameter `ContentType` dan `InvokeEndpoint` API diperlukan. Format yang didukung termasuk tipe MIME `text/csv` dan`application/jsonlines`.

Gunakan `sagemaker_runtime_client` untuk mengirim permintaan ke titik akhir, sebagai berikut:

```
response = sagemaker_runtime_client.invoke_endpoint(
    EndpointName='name-of-your-endpoint',
    EnableExplanations='`true`',
    ContentType='text/csv',
    Accept='text/csv',
    Body='1,2,3,4',  # single record (of four numerical features)
)
```

Untuk titik akhir multi-model, berikan `TargetModel` parameter tambahan dalam permintaan contoh sebelumnya untuk menentukan model mana yang akan ditargetkan pada titik akhir. Titik akhir multi-

model secara dinamis memuat model target sesuai kebutuhan. Untuk informasi selengkapnya tentang titik akhir multi-model, lihat. Host beberapa model dalam satu wadah di belakang satu titik akhir Lihat SageMaker Clarify Online Explainability on Multi-Model Endpoint Sample Notebook untuk contoh cara menyiapkan dan memanggil beberapa model target dari satu titik akhir.

Respons

Jika titik akhir dibuat dengan`ExplainerConfig`, maka skema respons baru digunakan, Skema baru ini berbeda dari, dan tidak kompatibel dengan, titik akhir yang tidak memiliki parameter yang disediakan. `ExplainerConfig`

Jenis respons MIME adalah`application/json`, dan payload respons dapat diterjemahkan dari UTF-8 byte ke objek JSON. Berikut ini menunjukkan anggota objek JSON ini adalah sebagai berikut:

- `version`: Versi skema respons dalam format string. Misalnya, `1.0`.
- `predictions`: Prediksi yang dibuat permintaan memiliki yang berikut:
  - `content_type`: Jenis prediksi MIME, mengacu pada respons `ContentType` wadah model.
  - `data`: String data prediksi dikirimkan sebagai muatan respons wadah model untuk permintaan tersebut.
- `label_headers`: Header label dari `LabelHeaders` parameter. Ini disediakan baik dalam konfigurasi explainer atau output wadah model.
- `explanations`: Penjelasan yang diberikan dalam payload permintaan. Jika tidak ada catatan yang dijelaskan, maka anggota ini mengembalikan objek kosong`{}`.
- • `kernel_shap`: Kunci yang mengacu pada array penjelasan Kernel SHAP untuk setiap catatan dalam permintaan. Jika catatan tidak dijelaskan, penjelasan yang sesuai adalah`null`.

`kernel_shap`Elemen memiliki anggota berikut:

- `feature_header`: Nama header dari fitur yang disediakan oleh `FeatureHeaders` parameter dalam konfigurasi `ExplainerConfig` explainer.
- `feature_type`: Jenis fitur disimpulkan oleh penjelasan atau disediakan dalam `FeatureTypes` parameter di. `ExplainerConfig` Elemen ini hanya tersedia untuk masalah penjelasan NLP.
- `attributions`: Sebuah array objek atribusi. Fitur teks dapat memiliki beberapa objek atribusi, masing-masing untuk satu unit. Objek atribusi memiliki anggota berikut:
  - `attribution`: Daftar nilai probabilitas, diberikan untuk setiap kelas.
  - `description`: Deskripsi unit teks, hanya tersedia untuk masalah penjelasan NLP.

- `partial_text`: Bagian teks yang dijelaskan oleh penjelas.
- `start_idx`: Indeks berbasis nol untuk mengidentifikasi lokasi array dari awal fragmen teks sebagian.

Contoh kode: SDK untuk Python

Bagian ini menyediakan contoh kode untuk membuat dan memanggil titik akhir yang menggunakan SageMaker Clarify online explainability. Contoh kode ini menggunakan AWSSDK untuk Python.

Data tabular

Contoh berikut menggunakan data tabular dan SageMaker model yang disebut`model_name`. Dalam contoh ini, wadah model menerima data dalam format CSV, dan setiap catatan memiliki empat fitur numerik. Dalam konfigurasi minimal ini, hanya untuk tujuan demonstrasi, data dasar SHAP diatur ke nol. Lihat Garis Dasar SHAP untuk Penjelasan untuk mempelajari cara memilih nilai yang lebih sesuai untuk`ShapBaseline`.

Konfigurasikan titik akhir, sebagai berikut:

```
endpoint_config_name = 'tabular_explainer_endpoint_config'
response = sagemaker_client.create_endpoint_config(
    EndpointConfigName=endpoint_config_name,
    ProductionVariants=[{
        'VariantName': 'AllTraffic',
        'ModelName': model_name,
        'InitialInstanceCount': 1,
        'InstanceType': 'ml.m5.xlarge',
    }],
    ExplainerConfig={
        'ClarifyExplainerConfig': {
            'ShapConfig': {
                'ShapBaselineConfig': {
                    'ShapBaseline': '0,0,0,0',
                },
            },
        },
    },
)
```

Gunakan konfigurasi endpoint untuk membuat endpoint, sebagai berikut:

```
endpoint_name = 'tabular_explainer_endpoint'
response = sagemaker_client.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name,
)
```

Gunakan `DescribeEndpoint` API untuk memeriksa kemajuan pembuatan titik akhir, sebagai berikut:

```
response = sagemaker_client.describe_endpoint(
    EndpointName=endpoint_name,
)
response['EndpointStatus']
```

Setelah status endpoint adalah "InService“, panggil titik akhir dengan catatan pengujian, sebagai berikut:

```
response = sagemaker_runtime_client.invoke_endpoint(
    EndpointName=endpoint_name,
    ContentType='text/csv',
    Accept='text/csv',
    Body='1,2,3,4',
)
```

> **Note**
>
> Dalam contoh kode sebelumnya, untuk titik akhir multi-model, berikan `TargetModel` parameter tambahan dalam permintaan untuk menentukan model mana yang akan ditargetkan pada titik akhir.

Asumsikan bahwa respons memiliki kode status 200 (tidak ada kesalahan), dan muat badan respons, sebagai berikut:

```
import codecs
import json
json.load(codecs.getreader('utf-8')(response['Body']))
```

Tindakan default untuk titik akhir adalah menjelaskan catatan. Berikut ini menunjukkan contoh output dalam objek JSON dikembalikan.


```
{
    "version": "1.0",
    "predictions": {
        "content_type": "text/csv; charset=utf-8",
        "data": "0.0006380207487381"
    },
    "explanations": {
        "kernel_shap": [
            [
                {
                    "attributions": [
                        {
                            "attribution": [-0.00433456]
                        }
                    ]
                },
                {
                    "attributions": [
                        {
                            "attribution": [-0.005369821]
                        }
                    ]
                },
                {
                    "attributions": [
                        {
                            "attribution": [0.007917749]
                        }
                    ]
                },
                {
                    "attributions": [
                        {
                            "attribution": [-0.00261214]
                        }
                    ]
                }
            ]
        ]
    }
```

```
}
```

Gunakan `EnableExplanations` parameter untuk mengaktifkan penjelasan sesuai permintaan, sebagai berikut:

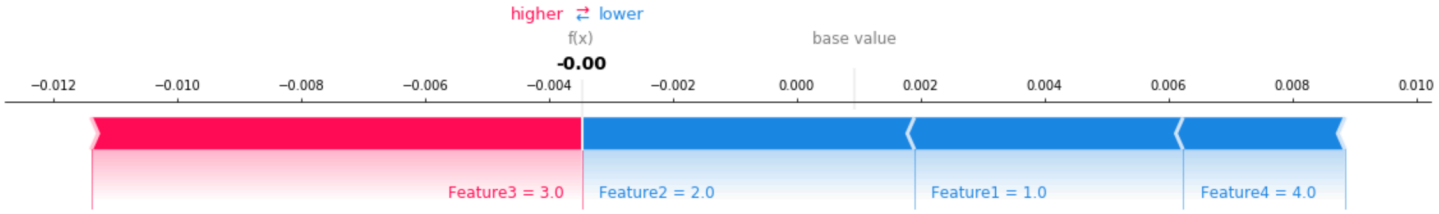
```
response = sagemaker_runtime_client.invoke_endpoint(
    EndpointName=endpoint_name,
    ContentType='text/csv',
    Accept='text/csv',
    Body='1,2,3,4',
    EnableExplanations='[0]>`0.8`',
)
```

> **Note**
>
> Dalam contoh kode sebelumnya, untuk titik akhir multi-model, berikan `TargetModel` parameter tambahan dalam permintaan untuk menentukan model mana yang akan ditargetkan pada titik akhir.

Dalam contoh ini, nilai prediksi kurang dari nilai ambang batas`0.8`, sehingga catatan tidak dijelaskan:

```
{
    "version": "1.0",
    "predictions": {
        "content_type": "text/csv; charset=utf-8",
        "data": "0.6380207487381995"
    },
    "explanations": {}
}
```

Gunakan alat visualisasi untuk membantu menafsirkan penjelasan yang dikembalikan. Gambar berikut menunjukkan bagaimana plot SHAP dapat digunakan untuk memahami bagaimana setiap fitur berkontribusi pada prediksi. Nilai dasar pada diagram, juga disebut nilai yang diharapkan, adalah prediksi rata-rata dari kumpulan data pelatihan. Fitur yang mendorong nilai yang diharapkan lebih tinggi berwarna merah, dan fitur yang mendorong nilai yang diharapkan lebih rendah berwarna biru. Lihat tata letak gaya aditif SHAP untuk informasi tambahan.



Lihat contoh buku catatan lengkap untuk data tabular.

Data teks

Bagian ini memberikan contoh kode untuk membuat dan memanggil titik akhir penjelasan online untuk data teks. Contoh kode menggunakan SDK untuk Python.

Contoh berikut menggunakan data teks dan SageMaker model yang disebut`model_name`. Dalam contoh ini, wadah model menerima data dalam format CSV, dan setiap catatan adalah string tunggal.

```
endpoint_config_name = 'text_explainer_endpoint_config'
response = sagemaker_client.create_endpoint_config(
    EndpointConfigName=endpoint_config_name,
    ProductionVariants=[{
        'VariantName': 'AllTraffic',
        'ModelName': model_name,
        'InitialInstanceCount': 1,
        'InstanceType': 'ml.m5.xlarge',
    }],
    ExplainerConfig={
        'ClarifyExplainerConfig': {
            'InferenceConfig': {
                'FeatureTypes': ['text'],
                'MaxRecordCount': 100,
            },
            'ShapConfig': {
                'ShapBaselineConfig': {
                    'ShapBaseline': '"<MASK>"',
                },
                'TextConfig': {
                    'Granularity': 'token',
                    'Language': 'en',
                },
                'NumberOfSamples': 100,
            },
```

```
        },
    },
)
```

- `ShapBaseline`: Token khusus yang disediakan untuk pemrosesan bahasa alami (NLP).
- `FeatureTypes`: Mengidentifikasi fitur sebagai teks. Jika parameter ini tidak disediakan, penjelasan akan mencoba menyimpulkan jenis fitur.
- `TextConfig`: Menentukan unit granularitas dan bahasa untuk analisis fitur teks. Dalam contoh ini, bahasanya adalah bahasa Inggris, dan granularitas `token` berarti kata dalam teks bahasa Inggris.
- `NumberOfSamples`: Batas untuk mengatur batas atas ukuran dataset sintetis.
- `MaxRecordCount`: Jumlah maksimum catatan dalam permintaan yang dapat ditangani oleh wadah model. Parameter ini diatur untuk menstabilkan kinerja.

Gunakan konfigurasi endpoint untuk membuat endpoint, sebagai berikut:

```
endpoint_name = 'text_explainer_endpoint'
response = sagemaker_client.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name,
)
```

Setelah status titik akhir menjadi`InService`, panggil titik akhir. Contoh kode berikut menggunakan catatan uji sebagai berikut:

```
response = sagemaker_runtime_client.invoke_endpoint(
    EndpointName=endpoint_name,
    ContentType='text/csv',
    Accept='text/csv',
    Body='"This is a good product"',
)
```

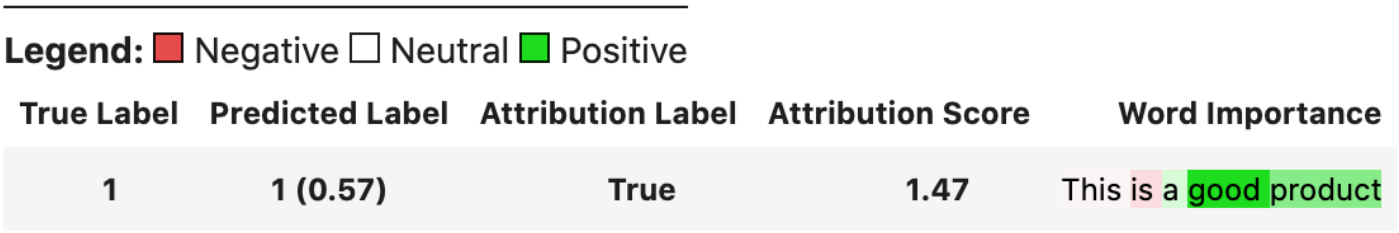
Jika permintaan berhasil diselesaikan, badan respons akan mengembalikan objek JSON yang valid yang mirip dengan berikut ini:

```
{
    "version": "1.0",
    "predictions": {
        "content_type": "text/csv",
        "data": "0.9766594\n"
```

```
    },
    "explanations": {
        "kernel_shap": [
            [
                {
                    "attributions": [
                        {
                            "attribution": [
                                -0.007270948666666712
                            ],
                            "description": {
                                "partial_text": "This",
                                "start_idx": 0
                            }
                        },
                        {
                            "attribution": [
                                -0.018199033666666628
                            ],
                            "description": {
                                "partial_text": "is",
                                "start_idx": 5
                            }
                        },
                        {
                            "attribution": [
                                0.01970993241666666
                            ],
                            "description": {
                                "partial_text": "a",
                                "start_idx": 8
                            }
                        },
                        {
                            "attribution": [
                                0.1253469515833334
                            ],
                            "description": {
                                "partial_text": "good",
                                "start_idx": 10
                            }
                        },
                        {
                            "attribution": [
```

```
                            0.03291143366666657
                        ],
                        "description": {
                            "partial_text": "product",
                            "start_idx": 15
                        }
                    }
                ],
                "feature_type": "text"
            }
        ]
    ]
  }
}
```

Gunakan alat visualisasi untuk membantu menafsirkan atribusi teks yang dikembalikan. Gambar berikut menunjukkan bagaimana utilitas visualisasi captum dapat digunakan untuk memahami bagaimana setiap kata berkontribusi pada prediksi. Semakin tinggi saturasi warna, semakin tinggi pentingnya kata tersebut. Dalam contoh ini, warna merah cerah yang sangat jenuh menunjukkan kontribusi negatif yang kuat. Warna hijau yang sangat jenuh menunjukkan kontribusi positif yang kuat. Warna putih menunjukkan bahwa kata tersebut memiliki kontribusi netral. Lihat perpustakaan captum untuk informasi tambahan tentang penguraian dan rendering atribusi.



Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	1 (0.57)	True	1.47	This is a good product

Lihat contoh buku catatan lengkap untuk data teks.

Panduan pemecahan masalah

Jika Anda menemukan kesalahan menggunakan SageMaker Clarify online explainability, lihat topik di bagian ini.

`InvokeEndpoint`API gagal dengan kesalahan “: Baca ReadTimeoutError batas waktu di titik akhir...”

Kesalahan ini berarti bahwa permintaan tidak dapat diselesaikan dalam batas waktu 60 detik yang ditetapkan oleh batas waktu permintaan.

Untuk mengurangi latensi permintaan, coba yang berikut ini:

- Sesuaikan kinerja model selama inferensi. Misalnya, SageMaker Neo dapat mengoptimalkan model untuk inferensi.
- Izinkan wadah model untuk menangani permintaan batch.
- Gunakan yang lebih besar `MaxRecordCount` untuk mengurangi jumlah panggilan dari penjelasan ke wadah model. Ini akan mengurangi latensi jaringan dan overhead.
- Gunakan jenis instance yang memiliki lebih banyak sumber daya yang dialokasikan untuk itu. Sebagai alternatif, tetapkan lebih banyak instance ke titik akhir untuk membantu menyeimbangkan beban.
- Kurangi jumlah catatan dalam satu `InvokeEndpoint` permintaan.
- Kurangi jumlah catatan dalam data dasar.
- Gunakan `NumberOfSamples` nilai yang lebih kecil untuk mengurangi ukuran kumpulan data sintetis. Untuk informasi selengkapnya tentang bagaimana jumlah sampel memengaruhi kumpulan data sintetis Anda, lihatDataset sintetis.

Inferensi Tanpa Server

Amazon SageMaker Serverless Inference adalah opsi inferensi yang dibuat khusus yang memungkinkan Anda menerapkan dan menskalakan model ML tanpa mengonfigurasi atau mengelola infrastruktur yang mendasarinya. Inferensi Tanpa Server On-Demand sangat ideal untuk beban kerja yang memiliki periode idle antara lonjakan lalu lintas dan dapat mentolerir start dingin. Titik akhir tanpa server secara otomatis meluncurkan sumber daya komputasi dan menskalakannya masuk dan keluar tergantung pada lalu lintas, sehingga tidak perlu memilih jenis instance atau mengelola kebijakan penskalaan. Ini menghilangkan beban berat yang tidak terdiferensiasi dalam memilih dan mengelola server. Inferensi Tanpa Server terintegrasi AWS Lambda untuk menawarkan ketersediaan tinggi, toleransi kesalahan bawaan, dan penskalaan otomatis. Dengan pay-per-use model, Inferensi Tanpa Server adalah opsi hemat biaya jika Anda memiliki pola lalu lintas yang jarang atau tidak dapat diprediksi. Selama saat tidak ada permintaan, Inferensi Tanpa Server menskalakan titik akhir Anda menjadi 0, membantu Anda meminimalkan biaya. Untuk informasi selengkapnya tentang harga untuk Inferensi Tanpa Server sesuai permintaan, lihat Harga Amazon. SageMaker

Secara opsional, Anda juga dapat menggunakan Provisioned Concurrency dengan Serverless Inference. Inferensi Tanpa Server dengan konkurensi yang disediakan adalah opsi hemat biaya ketika Anda memiliki ledakan yang dapat diprediksi dalam lalu lintas Anda. Provisioned Concurrency memungkinkan Anda menerapkan model pada titik akhir tanpa server dengan kinerja yang dapat

diprediksi, dan skalabilitas tinggi dengan menjaga titik akhir tetap hangat. SageMaker memastikan bahwa untuk jumlah Konkurensi Terketentuan yang Anda alokasikan, sumber daya komputasi diinisialisasi dan siap merespons dalam milidetik. Untuk Inferensi Tanpa Server dengan Konkurensi Terketentuan, Anda membayar kapasitas komputasi yang digunakan untuk memproses permintaan inferensi, ditagih oleh milidetik, dan jumlah data yang diproses. Anda juga membayar penggunaan Provisioned Concurrency, berdasarkan memori yang dikonfigurasi, durasi yang disediakan, dan jumlah konkurensi yang diaktifkan. Untuk informasi selengkapnya tentang harga Inferensi Tanpa Server dengan Konkurensi yang Disediakan, lihat Harga Amazon. SageMaker

Anda dapat mengintegrasikan Inferensi Tanpa Server dengan Pipelines MLOPS untuk merampingkan alur kerja ML Anda, dan Anda dapat menggunakan titik akhir tanpa server untuk meng-host model yang terdaftar dengan Model Registry.

Inferensi Tanpa Server umumnya tersedia di 21 AWS Wilayah: AS Timur (Virginia N.), AS Timur (Ohio), AS Barat (California N.), AS Barat (Oregon), Afrika (Cape Town), Asia Pasifik (Hong Kong), Asia Pasifik (Mumbai), Asia Pasifik (Tokyo), Asia Pasifik (Seoul), Asia Pasifik (Osaka), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Kanada (Tengah), Eropa (Frankfurt), Eropa (Irlandia), Eropa (London), Eropa (Paris), Eropa (Stockholm), Eropa (Milan), Timur Tengah (Bahrain), Amerika Selatan (São Paulo). Untuk informasi selengkapnya tentang ketersediaan SageMaker regional Amazon, lihat Daftar Layanan AWS Regional.

Cara kerjanya

Diagram berikut menunjukkan alur kerja Inferensi Tanpa Server sesuai permintaan dan manfaat menggunakan titik akhir tanpa server.

Saat Anda membuat titik akhir tanpa server sesuai permintaan, menyediakan dan mengelola sumber SageMaker daya komputasi untuk Anda. Kemudian, Anda dapat membuat permintaan inferensi

ke titik akhir dan menerima prediksi model sebagai tanggapan. SageMaker skala sumber daya komputasi naik dan turun sesuai kebutuhan untuk menangani lalu lintas permintaan Anda, dan Anda hanya membayar untuk apa yang Anda gunakan.

Untuk Provisioned Concurrency, Serverless Inference juga terintegrasi dengan Application Auto Scaling, sehingga Anda dapat mengelola Provisioned Concurrency berdasarkan metrik target atau jadwal. Untuk informasi selengkapnya, lihat Secara otomatis menskalakan Konkurensi yang Disediakan untuk titik akhir tanpa server.

Bagian berikut memberikan rincian tambahan tentang Inferensi Tanpa Server dan cara kerjanya.

Topik

- Dukungan kontainer
- Ukuran memori
- Doa bersamaan
- Meminimalkan awal dingin
- Pengecualian fitur

Dukungan kontainer

Untuk wadah endpoint Anda, Anda dapat memilih kontainer SageMaker yang disediakan atau membawanya sendiri. SageMaker menyediakan wadah untuk algoritme bawaan dan gambar Docker bawaan untuk beberapa kerangka kerja pembelajaran mesin yang paling umum, seperti Apache MXNet,,, dan Chainer. TensorFlow PyTorch Untuk daftar gambar yang tersedia, lihat SageMaker Gambar Deep Learning Containers yang Tersedia. Jika Anda membawa wadah Anda sendiri, Anda harus memodifikasinya agar berfungsi SageMaker. Untuk informasi lebih lanjut tentang membawa wadah Anda sendiri, lihatMengadaptasi Wadah Inferensi Anda Sendiri.

Ukuran maksimum gambar kontainer yang dapat Anda gunakan adalah 10 GB. Untuk titik akhir tanpa server, kami sarankan untuk membuat hanya satu pekerja di wadah dan hanya memuat satu salinan model. Perhatikan bahwa ini tidak seperti titik akhir real-time, di mana beberapa SageMaker kontainer dapat membuat pekerja untuk setiap vCPU untuk memproses permintaan inferensi dan memuat model di setiap pekerja.

Jika Anda sudah memiliki wadah untuk titik akhir real-time, Anda dapat menggunakan wadah yang sama untuk titik akhir tanpa server Anda, meskipun beberapa kemampuan dikecualikan. Untuk mempelajari lebih lanjut tentang kemampuan container yang tidak didukung dalam Inferensi Tanpa Server, lihat. Pengecualian fitur Jika Anda memilih untuk menggunakan wadah yang sama,

SageMaker escrows (mempertahankan) salinan gambar kontainer Anda sampai Anda menghapus semua titik akhir yang menggunakan gambar. SageMaker mengenkripsi gambar yang disalin saat istirahat dengan kunci yang SageMaker dimiliki. AWS KMS

Ukuran memori

Endpoint tanpa server Anda memiliki ukuran RAM minimum 1024 MB (1 GB), dan ukuran RAM maksimum yang dapat Anda pilih adalah 6144 MB (6 GB). Ukuran memori yang dapat Anda pilih adalah 1024 MB, 2048 MB, 3072 MB, 4096 MB, 5120 MB, atau 6144 MB. Inferensi Tanpa Server secara otomatis menetapkan sumber daya komputasi sebanding dengan memori yang Anda pilih. Jika Anda memilih ukuran memori yang lebih besar, wadah Anda memiliki akses ke lebih banyak vCPU. Pilih ukuran memori endpoint Anda sesuai dengan ukuran model Anda. Umumnya, ukuran memori harus setidaknya sebesar ukuran model Anda. Anda mungkin perlu melakukan benchmark untuk memilih pilihan memori yang tepat untuk model Anda berdasarkan SLA latensi Anda. Untuk panduan langkah demi langkah untuk benchmark, lihat Memperkenalkan Toolkit Benchmarking Inferensi SageMaker Tanpa Server Amazon. Peningkatan ukuran memori memiliki harga yang berbeda; lihat halaman SageMaker harga Amazon untuk informasi lebih lanjut.

Terlepas dari ukuran memori yang Anda pilih, titik akhir tanpa server Anda memiliki 5 GB penyimpanan disk sementara yang tersedia. Untuk bantuan terkait masalah izin kontainer saat bekerja dengan penyimpanan, lihatPemecahan Masalah.

Doa bersamaan

On-Demand Serverless Inference mengelola kebijakan dan kuota penskalaan yang telah ditentukan sebelumnya untuk kapasitas titik akhir Anda. Endpoint tanpa server memiliki kuota untuk berapa banyak pemanggilan bersamaan yang dapat diproses pada saat yang bersamaan. Jika titik akhir dipanggil sebelum selesai memproses permintaan pertama, maka ia menangani permintaan kedua secara bersamaan.

Total konkurensi yang dapat Anda bagikan di antara semua titik akhir tanpa server di akun Anda bergantung pada wilayah Anda:

- Untuk Wilayah AS Timur (Ohio), Timur AS (Virginia N.), AS Barat (Oregon), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), dan Eropa (Irlandia), total konkurensi yang dapat Anda bagikan antara semua titik akhir tanpa server per Wilayah di akun Anda adalah 1000.
- Untuk Wilayah AS Barat (California N.), Afrika (Cape Town), Asia Pasifik (Hong Kong), Asia Pasifik (Mumbai), Asia Pasifik (Osaka), Asia Pasifik (Seoul), Kanada (Tengah), Eropa (London), Eropa

(Milan), Eropa (Paris), Eropa (Stockholm), Timur Tengah (Bahrain), dan Amerika Selatan (São Paulo), konkurensi total per Wilayah di wilayah Anda Akun adalah 500.

Anda dapat mengatur konkurensi maksimum untuk satu titik akhir hingga 200, dan jumlah total titik akhir tanpa server yang dapat Anda host di Wilayah adalah 50. Konkurensi maksimum untuk titik akhir individu mencegah titik akhir tersebut mengambil semua pemanggilan yang diizinkan untuk akun Anda, dan pemanggilan titik akhir apa pun di luar maksimum dibatasi.

> **Note**
>
> Konkurensi yang disediakan yang Anda tetapkan ke titik akhir tanpa server harus selalu kurang dari atau sama dengan konkurensi maksimum yang Anda tetapkan ke titik akhir tersebut.

Untuk mempelajari cara mengatur konkurensi maksimum untuk titik akhir Anda, lihat. Buat konfigurasi titik akhir Untuk informasi selengkapnya tentang kuota dan batas, lihat SageMaker titik akhir dan kuota Amazon di. Referensi Umum AWS Untuk meminta peningkatan batas layanan, hubungi AWSSupport. Untuk petunjuk tentang cara meminta peningkatan batas layanan, lihatWilayah dan kuota yang didukung.

Meminimalkan awal dingin

Jika titik akhir Inferensi Tanpa Server sesuai permintaan Anda tidak menerima lalu lintas untuk sementara waktu dan kemudian titik akhir Anda tiba-tiba menerima permintaan baru, perlu beberapa waktu bagi titik akhir Anda untuk memutar sumber daya komputasi untuk memproses permintaan. Ini disebut awal yang dingin. Karena penyediaan titik akhir tanpa server menghitung sumber daya sesuai permintaan, titik akhir Anda mungkin mengalami awal yang dingin. Cold start juga dapat terjadi jika permintaan bersamaan Anda melebihi penggunaan permintaan bersamaan saat ini. Waktu mulai dingin tergantung pada ukuran model Anda, berapa lama waktu yang dibutuhkan untuk mengunduh model Anda, dan waktu start-up wadah Anda.

Untuk memantau berapa lama waktu mulai dingin Anda, Anda dapat menggunakan CloudWatch metrik Amazon `OverheadLatency` untuk memantau titik akhir tanpa server Anda. Metrik ini melacak waktu yang diperlukan untuk meluncurkan sumber daya komputasi baru untuk titik akhir Anda. Untuk mempelajari selengkapnya tentang menggunakan CloudWatch metrik dengan titik akhir tanpa server, lihat. Memantau titik akhir tanpa server

Anda dapat meminimalkan start dingin dengan menggunakan Provisioned Concurrency. SageMaker membuat titik akhir tetap hangat dan siap untuk merespons dalam milidetik, untuk jumlah Konkurensi Tertentu yang Anda alokasikan.

Pengecualian fitur

Beberapa fitur yang saat ini tersedia untuk Inferensi SageMaker Real-Time tidak didukung untuk Inferensi Tanpa Server, termasuk GPU, paket model AWS pasar, pendaftar Docker pribadi, Titik Akhir Multi-Model, konfigurasi VPC, isolasi jaringan, pengambilan data, beberapa varian produksi, Monitor Model, dan saluran pipa inferensi.

Anda tidak dapat mengonversi titik akhir real-time berbasis instans menjadi titik akhir tanpa server. Jika Anda mencoba memperbarui titik akhir real-time Anda ke tanpa server, Anda menerima pesan. `ValidationError` Anda dapat mengonversi titik akhir tanpa server menjadi real-time, tetapi setelah Anda melakukan pembaruan, Anda tidak dapat mengembalikannya ke tanpa server.

Memulai

Anda dapat membuat, memperbarui, menjelaskan, dan menghapus titik akhir tanpa server menggunakan SageMaker konsol, SDK, Amazon SageMaker Python AWS SDK, dan file. AWS CLI Anda dapat memanggil endpoint Anda menggunakan SDK, Amazon SageMaker Python AWS SDK, dan file. AWS CLI Untuk endpoint tanpa server dengan Provisioned Concurrency, Anda dapat menggunakan Application Auto Scaling untuk skala otomatis Provisioned Concurrency berdasarkan metrik target atau jadwal. Untuk informasi selengkapnya tentang cara mengatur dan menggunakan titik akhir tanpa server, baca panduannya. Membuat, memanggil, memperbarui, dan menghapus titik akhir tanpa server Untuk informasi selengkapnya tentang penskalaan otomatis endpoint tanpa server dengan Provisioned Concurrency, lihat. Secara otomatis menskalakan Konkurensi yang Disediakan untuk titik akhir tanpa server

> **Note**
>
> Application Auto Scaling untuk Inferensi Tanpa Server dengan Konkurensi yang Disediakan saat ini tidak didukung pada. AWS CloudFormation

Contoh notebook dan blog

Untuk contoh notebook Jupyter yang menunjukkan alur kerja titik akhir end-to-end tanpa server, lihat contoh notebook Inferensi Tanpa Server.

Membuat, memanggil, memperbarui, dan menghapus titik akhir tanpa server

Tidak seperti titik akhir SageMaker real-time lainnya, Inferensi Tanpa Server mengelola sumber daya komputasi untuk Anda, mengurangi kompleksitas sehingga Anda dapat fokus pada model ML Anda alih-alih mengelola infrastruktur. Panduan berikut menyoroti kemampuan utama titik akhir tanpa server: cara membuat, memanggil, memperbarui, mendeskripsikan, atau menghapus titik akhir. Anda dapat menggunakan SageMaker konsol, AWS SDK, Amazon SageMaker Python SDK, atau untuk mengelola titik akhir tanpa AWS CLI server Anda.

Topik

Prasyarat

Sebelum Anda dapat membuat endpoint tanpa server, selesaikan prasyarat berikut.

1. Siapkan AWS akun. Pertama-tama Anda memerlukan AWS akun dan pengguna AWS Identity and Access Management administrator. Untuk petunjuk tentang cara mengatur AWS akun, lihat Bagaimana cara membuat dan mengaktifkan AWS akun baru? . Untuk petunjuk tentang cara mengamankan akun Anda dengan pengguna administrator IAM, lihat Membuat pengguna admin IAM pertama dan grup pengguna di Panduan Pengguna IAM.
2. Buat bucket Amazon S3. Anda menggunakan ember Amazon S3 untuk menyimpan artefak model Anda. Untuk mempelajari cara membuat bucket, lihat Membuat bucket S3 pertama Anda di Panduan Pengguna Amazon S3.
3. Unggah artefak model Anda ke bucket S3 Anda. Untuk petunjuk tentang cara mengunggah model ke bucket, lihat Mengunggah objek ke bucket di Panduan Pengguna Amazon S3.
4. Buat peran IAM untuk Amazon SageMaker. Amazon SageMaker membutuhkan akses ke bucket S3 yang menyimpan model Anda. Buat peran IAM dengan kebijakan yang memberikan

akses SageMaker baca ke bucket Anda. Prosedur berikut menunjukkan cara membuat peran di konsol, tetapi Anda juga dapat menggunakan CreateRoleAPI dari Panduan Pengguna IAM. Untuk informasi tentang memberikan izin lebih terperinci pada peran Anda berdasarkan kasus penggunaan Anda, lihat. SageMaker Peran

a. Masuklah ke konsol IAM.
b. Di tab navigasi, pilih Peran.
c. Pilih Buat Peran.
d. Untuk Pilih jenis entitas tepercaya, pilih AWSlayanan lalu pilih SageMaker.
e. Pilih Berikutnya: Izin dan kemudian pilih Berikutnya: Tag.
f. (Opsional) Tambahkan tag sebagai pasangan nilai kunci jika Anda ingin memiliki metadata untuk peran tersebut.
g. Pilih Berikutnya: Peninjauan.
h. Untuk nama Peran, masukkan nama untuk peran baru yang unik di AWS akun Anda. Anda tidak dapat mengedit nama peran setelah membuat peran.
i. (Opsional) Untuk Deskripsi peran, masukkan deskripsi.
j. Pilih Buat peran.

5. Lampirkan izin bucket S3 ke peran Anda SageMaker . Setelah membuat peran IAM, lampirkan kebijakan yang memberikan SageMaker izin untuk mengakses bucket S3 yang berisi artefak model Anda.

a. Di tab navigasi konsol IAM, pilih Peran.
b. Dari daftar peran, cari peran yang Anda buat di langkah sebelumnya berdasarkan nama.
c. Pilih peran Anda, lalu pilih Lampirkan kebijakan.
d. Untuk melampirkan izin, pilih Buat kebijakan.
e. Dalam tampilan Buat kebijakan, pilih tab JSON.
f. Tambahkan pernyataan kebijakan berikut ke editor JSON. Pastikan untuk mengganti *<your-bucket-name>* dengan nama bucket S3 yang menyimpan artefak model Anda. Jika Anda ingin membatasi akses ke folder atau file tertentu di bucket, Anda juga dapat menentukan jalur folder Amazon S3, misalnya,. *<your-bucket-name>*/*<model-folder>*

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<your-bucket-name>/*"
        }
    ]
}
```

g. Pilih Berikutnya: Tanda.

h. (Opsional) Tambahkan tag dalam pasangan nilai kunci ke kebijakan.

i. Pilih Berikutnya: Peninjauan.

j. Untuk Nama, masukkan nama untuk kebijakan baru.

k. (Opsional) Tambahkan Deskripsi untuk kebijakan.

l. Pilih Buat kebijakan.

m. Setelah membuat kebijakan, kembali ke Peran di konsol IAM dan pilih SageMaker peran Anda.

n. Pilih Lampirkan kebijakan.

o. Untuk izin Lampirkan, cari kebijakan yang Anda buat berdasarkan nama. Pilih dan pilih Lampirkan kebijakan.

6. Pilih gambar kontainer Docker bawaan atau bawa sendiri. Wadah yang Anda pilih menyajikan inferensi pada titik akhir Anda. SageMaker menyediakan wadah untuk algoritme bawaan dan gambar Docker bawaan untuk beberapa kerangka kerja pembelajaran mesin yang paling umum, seperti Apache MXNet,, dan Chainer. TensorFlow PyTorch Untuk daftar lengkap gambar yang tersedia, lihat SageMaker Gambar Deep Learning Containers yang Tersedia.

   Jika tidak ada SageMaker kontainer yang ada yang memenuhi kebutuhan Anda, Anda mungkin perlu membuat wadah Docker Anda sendiri. Untuk informasi tentang cara membuat image Docker Anda dan membuatnya kompatibel dengan SageMaker, lihatGunakan kode inferensi Anda sendiri. Untuk menggunakan container Anda dengan titik akhir tanpa server, image container harus berada di repositori Amazon ECR dalam akun yang sama yang membuat endpoint. AWS

7. (Opsional) Daftarkan model Anda dengan Model Registry. SageMaker Model Registry membantu Anda membuat katalog dan mengelola versi model untuk digunakan di saluran pipa ML. Untuk informasi selengkapnya tentang mendaftarkan versi model Anda, lihat Membuat Grup Model

danDaftarkan Versi Model. Untuk contoh alur kerja Registri Model dan Inferensi Tanpa Server, lihat contoh buku catatan berikut.

8. (Opsional) Bawa AWS KMS kunci. Saat menyiapkan titik akhir tanpa server, Anda memiliki opsi untuk menentukan kunci KMS yang SageMaker digunakan untuk mengenkripsi gambar Amazon ECR Anda. Perhatikan bahwa kebijakan kunci untuk kunci KMS harus memberikan akses ke peran IAM yang Anda tentukan saat menyiapkan titik akhir Anda. Untuk mempelajari selengkapnya tentang kunci KMS, lihat Panduan AWS Key Management Service Pengembang.

Buat titik akhir tanpa server

Untuk membuat titik akhir tanpa server, Anda dapat menggunakan SageMaker konsol Amazon, API, atau. AWS CLI Anda dapat membuat titik akhir tanpa server menggunakan proses serupa sebagai titik akhir real-time.

Topik

- Buat model
- Buat konfigurasi titik akhir
- Buat titik akhir

Buat model

Untuk membuat model Anda, Anda harus memberikan lokasi artefak model dan gambar kontainer Anda. Anda juga dapat menggunakan versi model dari SageMaker Model Registry. Contoh di bagian berikut menunjukkan cara membuat model menggunakan CreateModelAPI, Model Registry, dan SageMakerkonsol Amazon.

Untuk membuat model (menggunakan Model Registry)

Model Registry adalah fitur SageMaker yang membantu Anda membuat katalog dan mengelola versi model Anda untuk digunakan di saluran pipa ML. Untuk menggunakan Registri Model dengan Inferensi Tanpa Server, Anda harus terlebih dahulu mendaftarkan versi model dalam grup model Registry Model. Untuk mempelajari cara mendaftarkan model di Model Registry, ikuti prosedur di Membuat Grup Model danDaftarkan Versi Model.

Contoh berikut mengharuskan Anda untuk memiliki ARN dari versi model terdaftar dan menggunakan AWSSDK for Python (Boto3) untuk memanggil API. CreateModel Untuk Inferensi Tanpa Server, Registri Model saat ini hanya didukung oleh AWS SDK for Python (Boto3). Sebagai contoh, tentukan nilai-nilai berikut:

- Untuk`model_name`, masukkan nama untuk model.
- Untuk`sagemaker_role`, Anda dapat menggunakan peran default SageMaker yang dibuat atau peran SageMaker IAM yang disesuaikan dari Langkah 4 bagianPrasyarat.
- Untuk`ModelPackageName`, tentukan ARN untuk versi model Anda, yang harus didaftarkan ke grup model di Model Registry.

```
#Setup
import boto3
import sagemaker
region = boto3.Session().region_name
client = boto3.client("sagemaker", region_name=region)

#Role to give SageMaker permission to access AWS services.
sagemaker_role = sagemaker.get_execution_role()

#Specify a name for the model
model_name = "<name-for-model>"

#Specify a Model Registry model version
container_list = [
    {
        "ModelPackageName": <model-version-arn>
     }
 ]

#Create the model
response = client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = sagemaker_role,
    container_list
)
```

Untuk membuat model (menggunakan API)

Contoh berikut menggunakan AWSSDK for Python (Boto3) untuk memanggil API. CreateModel Tentukan nilai-nilai berikut ini:

- Untuk `sagemaker_role`, Anda dapat menggunakan peran default SageMaker yang dibuat atau peran SageMaker IAM yang disesuaikan dari Langkah 4 bagianPrasyarat.
- Untuk`model_url`, tentukan URI Amazon S3 ke model Anda.

- Untuk`container`, ambil wadah yang ingin Anda gunakan dengan jalur Amazon ECR-nya. Contoh ini menggunakan kontainer SageMaker XGBoost -provided. Jika Anda belum memilih SageMaker wadah atau membawa sendiri, lihat Langkah 6 Prasyarat bagian untuk informasi lebih lanjut.
- Untuk`model_name`, masukkan nama untuk model.

```
#Setup
import boto3
import sagemaker
region = boto3.Session().region_name
client = boto3.client("sagemaker", region_name=region)

#Role to give SageMaker permission to access AWS services.
sagemaker_role = sagemaker.get_execution_role()

#Get model from S3
model_url = "s3://DOC-EXAMPLE-BUCKET/models/model.tar.gz"

#Get container image (prebuilt example)
from sagemaker import image_uris
container = image_uris.retrieve("xgboost", region, "0.90-1")

#Create model
model_name = "<name-for-model>"

response = client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = sagemaker_role,
    Containers = [{
        "Image": container,
        "Mode": "SingleModel",
        "ModelDataUrl": model_url,
    }]
)
```

Untuk membuat model (menggunakan konsol)

1. Masuk ke SageMakerkonsol Amazon.
2. Di tab navigasi, pilih Inferensi.
3. Selanjutnya, pilih Model.
4. Pilih Buat model.

5. Untuk nama Model, masukkan nama untuk model yang unik untuk akun Anda danWilayah AWS.
6. Untuk peran IAM, pilih peran IAM yang telah Anda buat (lihatPrasyarat) atau izinkan SageMaker untuk membuatnya untuk Anda.
7. Dalam definisi Container 1, untuk opsi input Container, pilih Menyediakan artefak model dan lokasi input.
8. Untuk Menyediakan artefak model dan opsi gambar inferensi, pilih Gunakan model tunggal.
9. Untuk Lokasi gambar kode inferensi, masukkan jalur ECR Amazon ke wadah. Gambar harus berupa gambar pihak pertama SageMaker yang disediakan (misalnya TensorFlow, XGBoost) atau gambar yang berada di repositori Amazon ECR dalam akun yang sama tempat Anda membuat titik akhir. Jika Anda tidak memiliki wadah, kembali ke Langkah 6 Prasyarat bagian untuk informasi lebih lanjut.
10. Untuk Lokasi artefak model, masukkan URI Amazon S3 ke model ML Anda. Misalnya, *s3://DOC-EXAMPLE-BUCKET/models/model.tar.gz*.
11. (Opsional) Untuk Tag, tambahkan pasangan nilai kunci untuk membuat metadata untuk model Anda.
12. Pilih Buat model.

Buat konfigurasi titik akhir

Setelah Anda membuat model, buat konfigurasi titik akhir. Anda kemudian dapat menerapkan model Anda menggunakan spesifikasi dalam konfigurasi titik akhir Anda. Dalam konfigurasi, Anda menentukan apakah Anda menginginkan titik akhir real-time atau tanpa server. Untuk membuat konfigurasi titik akhir tanpa server, Anda dapat menggunakan SageMaker konsol Amazon, CreateEndpointConfigAPI, atau. AWS CLI Pendekatan API dan konsol diuraikan di bagian berikut.

Untuk membuat konfigurasi titik akhir (menggunakan API)

Contoh berikut menggunakan AWSSDK for Python (Boto3) untuk memanggil API. CreateEndpointConfig Tentukan nilai-nilai berikut ini:

- Untuk`EndpointConfigName`, pilih nama untuk konfigurasi titik akhir. Nama harus unik dalam akun Anda di Wilayah.
- (Opsional) Untuk`KmsKeyId`, gunakan ID kunci, ARN kunci, nama alias, atau alias ARN untuk AWS KMS kunci yang ingin Anda gunakan. SageMaker menggunakan kunci ini untuk mengenkripsi gambar Amazon ECR Anda.

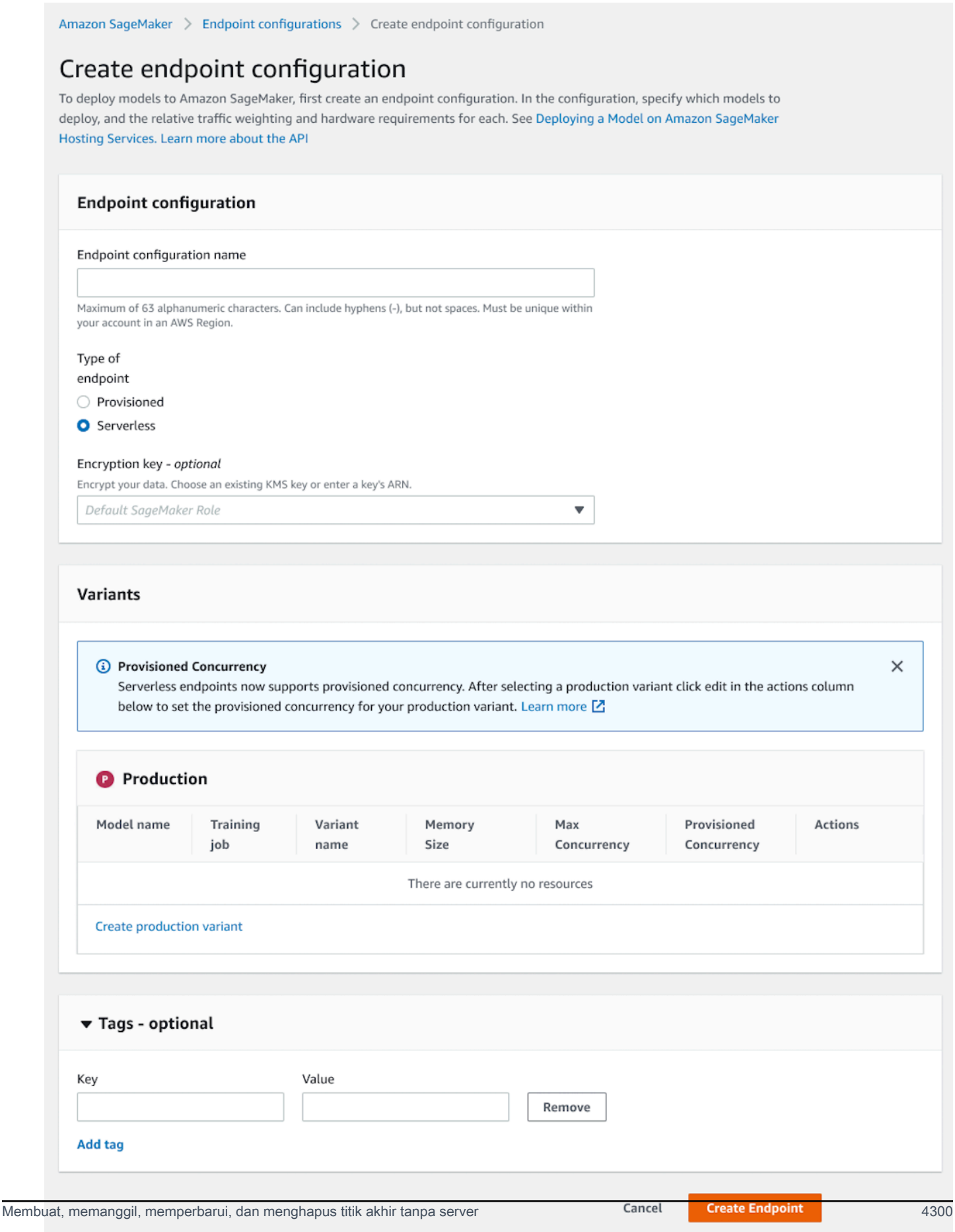
- Untuk`ModelName`, gunakan nama model yang ingin Anda terapkan. Itu harus model yang sama yang Anda gunakan di Buat model langkah.
- Untuk `ServerlessConfig`:
  - Atur `MemorySizeInMB` ke `2048`. Untuk contoh ini, kami mengatur ukuran memori menjadi 2048 MB, tetapi Anda dapat memilih salah satu nilai berikut untuk ukuran memori Anda: 1024 MB, 2048 MB, 3072 MB, 4096 MB, 5120 MB, atau 6144 MB.
  - Atur `MaxConcurrency` ke `20`. Untuk contoh ini, kami menetapkan konkurensi maksimum menjadi 20. Jumlah maksimum pemanggilan bersamaan yang dapat Anda atur untuk titik akhir tanpa server adalah 200, dan nilai minimum yang dapat Anda pilih adalah 1.
  - (Opsional) Untuk menggunakan Provisioned Concurrency, atur `ProvisionedConcurrency` ke 10. Untuk contoh ini, kita mengatur Provisioned Concurrency ke 10. `ProvisionedConcurrency`Angka untuk titik akhir tanpa server harus lebih rendah dari atau sama dengan angka. `MaxConcurrency` Anda dapat membiarkannya kosong jika Anda ingin menggunakan titik akhir Inferensi Tanpa Server sesuai permintaan. Anda dapat menskalakan Konkurensi Ketentuan secara dinamis. Untuk informasi selengkapnya, lihat Secara otomatis menskalakan Konkurensi yang Disediakan untuk titik akhir tanpa server.

```
response = client.create_endpoint_config(
   EndpointConfigName="<your-endpoint-configuration>",
   KmsKeyId="arn:aws:kms:us-east-1:123456789012:key/143ef68f-76fd-45e3-abba-
ed28fc8d3d5e",
   ProductionVariants=[
        {
            "ModelName": "<your-model-name>",
            "VariantName": "AllTraffic",
            "ServerlessConfig": {
                "MemorySizeInMB": 2048,
                "MaxConcurrency": 20,
                "ProvisionedConcurrency": 10,
            }
        }
    ]
)
```

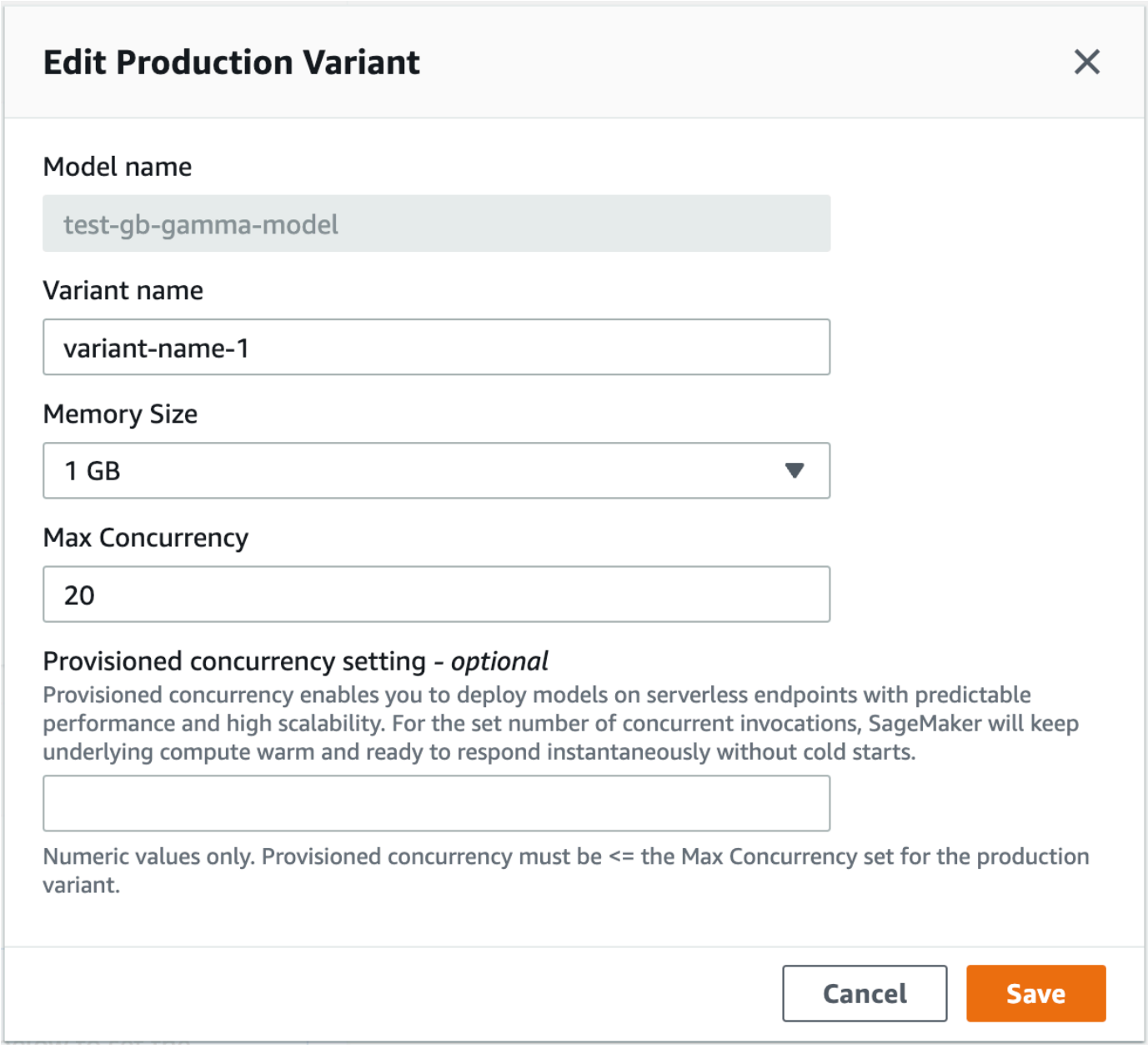
Untuk membuat konfigurasi titik akhir (menggunakan konsol)

1. Masuk ke SageMakerkonsol Amazon.
2. Di tab navigasi, pilih Inferensi.

3. Selanjutnya, pilih Konfigurasi titik akhir.
4. Pilih Buat konfigurasi titik akhir.
5. Untuk nama konfigurasi Endpoint, masukkan nama yang unik dalam akun Anda di Wilayah.
6. Untuk Jenis titik akhir, pilih Tanpa Server.


Amazon SageMaker > Endpoint configurations > Create endpoint configuration
Create endpoint configuration
To deploy models to Amazon SageMaker, first create an endpoint configuration. In the configuration, specify which models to deploy, and the relative traffic weighting and hardware requirements for each. See Deploying a Model on Amazon SageMaker Hosting Services. Learn more about the API
Endpoint configuration
Endpoint configuration name
Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
Type of endpoint
Provisioned
Serverless
Encryption key - optional
Encrypt your data. Choose an existing KMS key or enter a key's ARN.
Default SageMaker Role
Variants
Provisioned Concurrency
Serverless endpoints now supports provisioned concurrency. After selecting a production variant click edit in the actions column below to set the provisioned concurrency for your production variant. Learn more
Production
Model name
Training job
Variant name
Memory Size
Max Concurrency
Provisioned Concurrency
Actions
There are currently no resources
Create production variant
Tags - optional
Key
Value
Remove
Add tag
Cancel
Create Endpoint

7. Untuk varian Produksi, pilih Tambah model.
8. Di bawah Tambahkan model, pilih model yang ingin Anda gunakan dari daftar model dan kemudian pilih Simpan.
9. Setelah menambahkan model Anda, di bawah Tindakan, pilih Edit.
10. Untuk ukuran Memori, pilih ukuran memori yang Anda inginkan dalam GB.



11. Untuk Max Concurrency, masukkan pemanggilan bersamaan maksimum yang Anda inginkan untuk titik akhir. Nilai maksimum yang dapat Anda masukkan adalah 200 dan minimum adalah 1.
12. (Opsional) Untuk menggunakan Provisioned Concurrency, masukkan jumlah pemanggilan bersamaan yang diinginkan di bidang pengaturan Konkurensi yang Disediakan. Jumlah

pemanggilan bersamaan yang disediakan harus kurang dari atau sama dengan jumlah pemanggilan bersamaan maksimum.

13. Pilih Simpan.
14. (Opsional) Untuk Tag, masukkan pasangan nilai kunci jika Anda ingin membuat metadata untuk konfigurasi titik akhir Anda.
15. Pilih Buat konfigurasi titik akhir.

Buat titik akhir

Untuk membuat titik akhir tanpa server, Anda dapat menggunakan SageMaker konsol Amazon, CreateEndpointAPI, atau. AWS CLI Pendekatan API dan konsol diuraikan di bagian berikut. Setelah Anda membuat titik akhir Anda, dibutuhkan beberapa menit agar titik akhir tersedia.

Untuk membuat titik akhir (menggunakan API)

Contoh berikut menggunakan AWSSDK for Python (Boto3) untuk memanggil API. CreateEndpoint Tentukan nilai-nilai berikut ini:

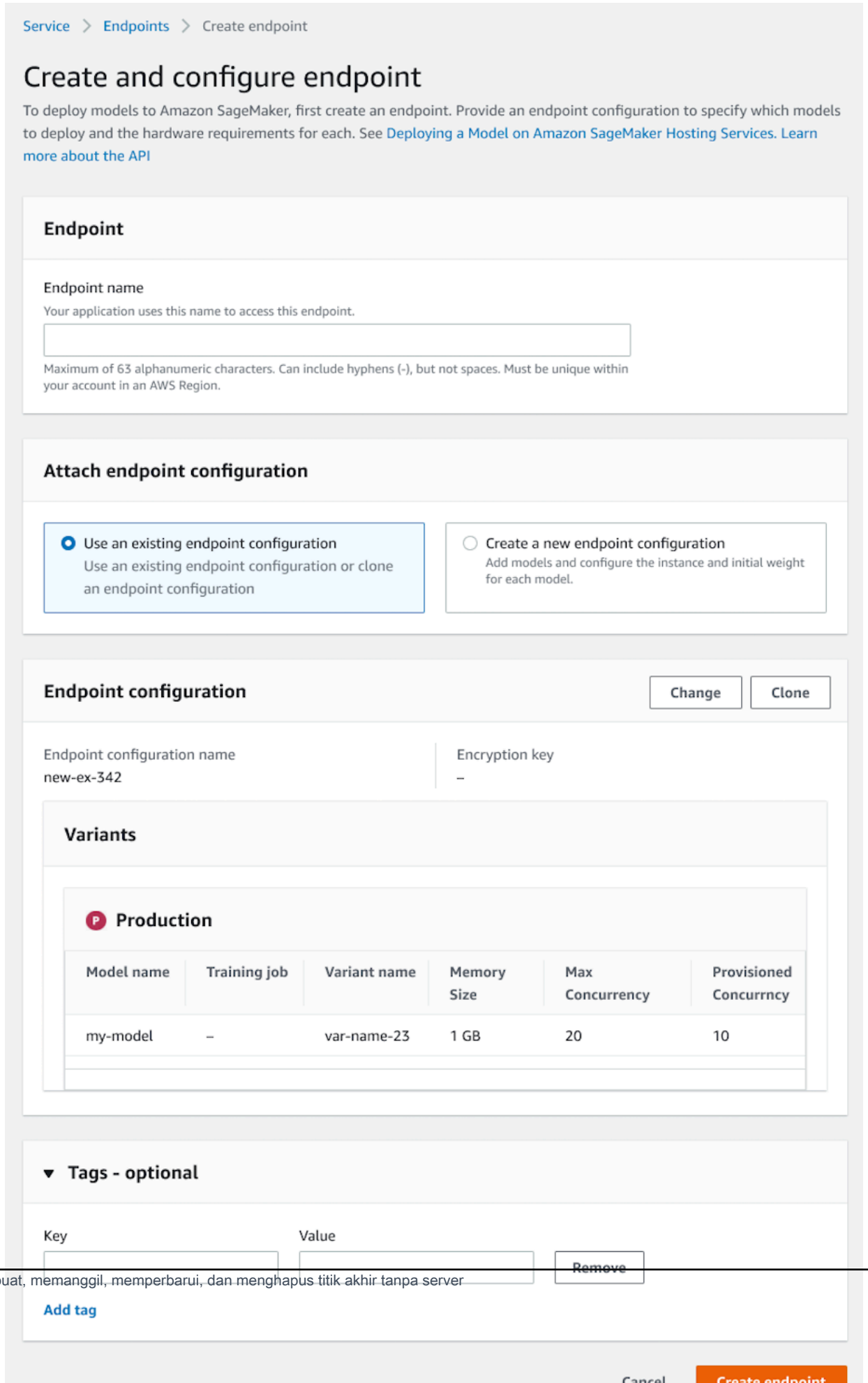
- Untuk`EndpointName`, masukkan nama untuk titik akhir yang unik dalam Wilayah di akun Anda.
- Untuk`EndpointConfigName`, gunakan nama konfigurasi titik akhir yang Anda buat di bagian sebelumnya.

```
response = client.create_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-endpoint-config>"
)
```

Untuk membuat titik akhir (menggunakan konsol)

1. Masuk ke SageMakerkonsol Amazon.
2. Di tab navigasi, pilih Inferensi.
3. Selanjutnya, pilih Endpoints.
4. Pilih Buat titik akhir.
5. Untuk nama Endpoint, masukkan nama yang unik dalam Wilayah di akun Anda.
6. Untuk melampirkan konfigurasi titik akhir, pilih Gunakan konfigurasi titik akhir yang ada.

7. Untuk konfigurasi Endpoint, pilih nama konfigurasi titik akhir yang Anda buat di bagian sebelumnya dan kemudian pilih Pilih konfigurasi titik akhir.
8. (Opsional) Untuk Tag, masukkan pasangan nilai kunci jika Anda ingin membuat metadata untuk titik akhir Anda.
9. Pilih Buat titik akhir.


Service > Endpoints > Create endpoint
Create and configure endpoint
To deploy models to Amazon SageMaker, first create an endpoint. Provide an endpoint configuration to specify which models to deploy and the hardware requirements for each. See Deploying a Model on Amazon SageMaker Hosting Services. Learn more about the API
Endpoint
Endpoint name
Your application uses this name to access this endpoint.
Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
Attach endpoint configuration
Use an existing endpoint configuration
Use an existing endpoint configuration or clone an endpoint configuration
Create a new endpoint configuration
Add models and configure the instance and initial weight for each model.
Endpoint configuration
Change
Clone
Endpoint configuration name
new-ex-342
Encryption key
–
Variants
Production
Model name
Training job
Variant name
Memory Size
Max Concurrency
Provisioned Concurrncy
my-model
–
var-name-23
1 GB
20
10
Tags - optional
Key
Value
Remove
Add tag
Cancel
Create endpoint

Memanggil titik akhir tanpa server

Untuk melakukan inferensi menggunakan titik akhir tanpa server, Anda harus mengirim permintaan HTTP ke titik akhir. Anda dapat menggunakan InvokeEndpointAPI atauAWS CLI, yang membuat `POST` permintaan untuk memanggil titik akhir Anda. Ukuran payload permintaan dan respons maksimum untuk pemanggilan tanpa server adalah 4 MB. Untuk titik akhir tanpa server:

- Model harus diunduh dan server harus merespons dengan sukses `/ping` dalam waktu 3 menit.
- Batas waktu penampung untuk menanggapi permintaan inferensi `/invocations` adalah 1 menit.

Untuk memanggil titik akhir

Contoh berikut menggunakan AWSSDK for Python (Boto3) untuk memanggil API. InvokeEndpoint Perhatikan bahwa tidak seperti panggilan API lainnya dalam panduan ini, untuk`InvokeEndpoint`, Anda harus menggunakan SageMaker Runtime Runtime sebagai klien. Tentukan nilai-nilai berikut ini:

- Untuk`endpoint_name`, gunakan nama titik akhir tanpa server dalam layanan yang ingin Anda panggil.
- Untuk`content_type`, tentukan tipe MIME data input Anda di badan permintaan (misalnya,`application/json`).
- Untuk`payload`, gunakan payload permintaan Anda untuk inferensi. Payload Anda harus dalam byte atau objek seperti file.

```
runtime = boto3.client("sagemaker-runtime")

endpoint_name = "<your-endpoint-name>"
content_type = "<request-mime-type>"
payload = <your-request-body>

response = runtime.invoke_endpoint(
    EndpointName=endpoint_name,
    ContentType=content_type,
    Body=payload
)
```

Memperbarui titik akhir tanpa server

Sebelum memperbarui titik akhir Anda, buat konfigurasi titik akhir baru atau gunakan konfigurasi titik akhir yang ada. Konfigurasi titik akhir adalah tempat Anda menentukan perubahan untuk pembaruan Anda. Kemudian, Anda dapat memperbarui titik akhir Anda dengan SageMaker konsol, UpdateEndpointAPI, atau. AWS CLI Proses untuk memperbarui titik akhir tanpa server sama dengan proses untuk memperbarui titik akhir real-time. Perhatikan bahwa saat memperbarui endpoint, Anda dapat mengalami cold start saat membuat permintaan ke endpoint karena SageMaker harus menginisialisasi ulang container dan model Anda.

Anda mungkin ingin memperbarui titik akhir tanpa server sesuai permintaan ke titik akhir tanpa server dengan konkurensi yang disediakan atau menyesuaikan nilai Konkurensi yang Disediakan untuk titik akhir tanpa server yang ada dengan konkurensi yang disediakan. Untuk kedua kasus tersebut, Anda harus membuat konfigurasi endpoint tanpa server baru dengan nilai yang diinginkan untuk Provisioned Concurrency, dan menerapkan `UpdateEndpoint` ke titik akhir tanpa server yang ada. Untuk informasi selengkapnya tentang membuat konfigurasi endpoint tanpa server baru dengan Provisioned Concurrency, lihat. Buat konfigurasi titik akhir

Jika Anda ingin menghapus Provisioned Concurrency dari titik akhir tanpa server, Anda harus membuat konfigurasi titik akhir baru tanpa menentukan nilai apa pun untuk Provisioned Concurrency, dan kemudian menerapkan ke titik akhir. `UpdateEndpoint`

> **Note**
>
> Memperbarui titik akhir inferensi real-time ke titik akhir tanpa server sesuai permintaan atau titik akhir tanpa server dengan Provisioned Concurrency saat ini tidak didukung.

Perbarui titik akhir

Setelah membuat konfigurasi titik akhir tanpa server baru, Anda dapat menggunakan AWS SDK for Python (Boto3)atau SageMaker konsol untuk memperbarui titik akhir tanpa server yang ada. Contoh cara memperbarui titik akhir Anda menggunakan AWS SDK for Python (Boto3) dan SageMaker konsol diuraikan di bagian berikut.

Untuk memperbarui titik akhir (menggunakan Boto3)

Contoh berikut menggunakan AWS SDK for Python (Boto3)untuk memanggil metode update_endpoint. Tentukan setidaknya parameter berikut saat memanggil metode:

- Untuk`EndpointName`, gunakan nama titik akhir yang Anda perbarui.
- Untuk`EndpointConfigName`, gunakan nama konfigurasi titik akhir yang ingin Anda gunakan untuk pembaruan.

```
response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<new-endpoint-config>",
)
```

Untuk memperbarui titik akhir (menggunakan konsol)

1. Masuk ke SageMakerkonsol Amazon.
2. Di tab navigasi, pilih Inferensi.
3. Selanjutnya, pilih Endpoints.
4. Dari daftar titik akhir, pilih titik akhir yang ingin Anda perbarui.
5. Pilih Ubah di bagian pengaturan konfigurasi Endpoint.
6. Untuk Mengubah konfigurasi Endpoint, pilih Gunakan konfigurasi endpoint yang ada.
7. Dari daftar konfigurasi titik akhir, pilih salah satu yang ingin Anda gunakan untuk pembaruan Anda.
8. Pilih Pilih konfigurasi titik akhir.
9. Pilih Perbarui titik akhir.

Jelaskan titik akhir tanpa server

Anda mungkin ingin mengambil informasi tentang titik akhir Anda, termasuk detail seperti ARN titik akhir, status saat ini, konfigurasi penerapan, dan alasan kegagalan. Anda dapat menemukan informasi tentang titik akhir Anda menggunakan SageMaker konsol, DescribeEndpointAPI, atau. AWS CLI

Untuk mendeskripsikan titik akhir (menggunakan API)

Contoh berikut menggunakan AWSSDK for Python (Boto3) untuk memanggil API. DescribeEndpoint Untuk`EndpointName`, gunakan nama titik akhir yang ingin Anda periksa.

```
response = client.describe_endpoint(
    EndpointName="<your-endpoint-name>",
```

```
)
```

Untuk mendeskripsikan titik akhir (menggunakan konsol)

1. Masuk ke SageMakerkonsol Amazon.
2. Di tab navigasi, pilih Inferensi.
3. Selanjutnya, pilih Endpoints.
4. Dari daftar titik akhir, pilih titik akhir yang ingin Anda periksa.

Halaman endpoint berisi informasi tentang endpoint Anda.

Menghapus titik akhir tanpa server

Anda dapat menghapus titik akhir tanpa server menggunakan SageMaker konsol, DeleteEndpointAPI, atau. AWS CLI Contoh berikut menunjukkan cara menghapus titik akhir Anda melalui API dan SageMaker konsol.

Untuk menghapus titik akhir (menggunakan API)

Contoh berikut menggunakan AWSSDK for Python (Boto3) untuk memanggil API. DeleteEndpoint Untuk`EndpointName`, gunakan nama titik akhir tanpa server yang ingin Anda hapus.

```
response = client.delete_endpoint(
    EndpointName="<your-endpoint-name>",
)
```

Untuk menghapus titik akhir (menggunakan konsol)

1. Masuk ke SageMakerkonsol Amazon.
2. Di tab navigasi, pilih Inferensi.
3. Selanjutnya, pilih Endpoints.
4. Dari daftar titik akhir, pilih titik akhir yang ingin Anda hapus.
5. Pilih daftar drop-down Tindakan, lalu pilih Hapus.
6. Ketika diminta lagi, pilih Hapus.

Titik akhir Anda sekarang harus memulai proses penghapusan.

Memantau titik akhir tanpa server

Untuk memantau titik akhir tanpa server, Anda dapat menggunakan alarm Amazon. CloudWatch CloudWatchadalah layanan yang mengumpulkan metrik secara real time dari AWS aplikasi dan sumber daya Anda. Alarm mengawasi metrik saat dikumpulkan dan memberi Anda kemampuan untuk menentukan ambang batas sebelumnya dan tindakan yang harus diambil jika ambang batas tersebut dilanggar. Misalnya, CloudWatch alarm Anda dapat mengirimi Anda pemberitahuan jika titik akhir Anda melanggar ambang kesalahan. Dengan menyiapkan CloudWatch alarm, Anda mendapatkan visibilitas ke kinerja dan fungsionalitas titik akhir Anda. Untuk informasi selengkapnya tentang CloudWatch alarm, lihat Menggunakan CloudWatch alarm Amazon di CloudWatchPanduan Pengguna Amazon.

Pemantauan dengan CloudWatch

Metrik di bawah ini adalah daftar metrik lengkap untuk titik akhir tanpa server. Metrik apa pun yang tidak tercantum di bawah ini tidak dipublikasikan untuk titik akhir tanpa server. Untuk informasi tentang metrik berikut, lihat Memantau Amazon SageMaker dengan Amazon CloudWatch.

Metrik titik akhir yang umum

CloudWatchMetrik ini sama dengan metrik yang dipublikasikan untuk titik akhir waktu nyata.

`OverheadLatency`Metrik melacak semua latensi tambahan yang SageMaker ditambahkan yang mencakup waktu mulai dingin untuk meluncurkan sumber daya komputasi baru untuk titik akhir tanpa server Anda. Dibandingkan dengan endpoint tanpa server sesuai permintaan, titik akhir tanpa server dengan `OverheadLatency` konkurensi ketentuan umumnya jauh lebih sedikit.

Endpoint tanpa server juga dapat menggunakan`Invocations4XXErrors`,,, `Invocations5XXErrors` `InvocationsModelLatency`, `ModelSetupTime` dan metrik. `MemoryUtilization` Untuk mempelajari lebih lanjut tentang metrik ini, lihatSageMaker Metrik Pemanggilan Titik Akhir.

Metrik titik akhir tanpa server umum

CloudWatchMetrik ini dipublikasikan untuk endpoint tanpa server sesuai permintaan dan titik akhir tanpa server dengan Konkurensi Terprovisi.

Nama Metrik	Deskripsi	Unit/Statistik
`ServerlessConcurrentExecutionsUtilization`	Jumlah eksekusi bersamaan dibagi dengan konkurensi maksimum.	Unit: Tidak ada Statistik yang valid: Rata-rata, Maks, Min

Titik akhir tanpa server dengan metrik Konkurensi yang Disediakan

CloudWatchMetrik ini dipublikasikan untuk titik akhir tanpa server dengan Konkurensi yang Disediakan.

Nama Metrik	Deskripsi	Unit/Statistik
`ServerlessProvisionedConcurrencyExecutions`	Jumlah eksekusi bersamaan ditangani oleh titik akhir.	Unit: Count (Jumlah) Statistik yang valid: Rata-rata, Maks, Min
`ServerlessProvisionedConcurrencyUtilization`	Jumlah eksekusi bersamaan dibagi dengan dialokasikan Provisioned Concurrency.	Unit: Tidak ada Statistik yang valid: Rata-rata, Maks, Min
`ServerlessProvisionedConcurrencyInvocations`	Jumlah `InvokeEndpoint` permintaan yang ditangani oleh Provisioned Concurrency.	Unit: Count (Jumlah) Statistik yang valid: Rata-rata, Maks, Min
`ServerlessProvisionedConcurrencySpilloverInvocations`	Jumlah `InvokeEndpoint` permintaan yang tidak ditangani oleh Provisioned Concurrency, yang ditangani oleh Inferensi Tanpa Server sesuai permintaan.	Unit: Count (Jumlah) Statistik yang valid: Rata-rata, Maks, Min

Log

Jika Anda ingin memantau log dari titik akhir untuk proses debug atau analisis kemajuan, Anda dapat menggunakan Amazon CloudWatch Logs. Grup log SageMaker yang disediakan yang dapat Anda gunakan untuk titik akhir tanpa server adalah. `/aws/sagemaker/Endpoints/[EndpointName]` Untuk informasi selengkapnya tentang penggunaan CloudWatch Log inSageMaker, lihatLog SageMaker Acara Amazon dengan Amazon CloudWatch. Untuk mempelajari lebih lanjut tentang CloudWatch Log, lihat Apa itu Amazon CloudWatch Logs? di Panduan Pengguna Amazon CloudWatch Logs.

Secara otomatis menskalakan Konkurensi yang Disediakan untuk titik akhir tanpa server

Amazon SageMaker secara otomatis menskalakan masuk atau keluar titik akhir tanpa server sesuai permintaan. Untuk endpoint tanpa server dengan Konkurensi Terprovisi, Anda dapat menggunakan Application Auto Scaling untuk meningkatkan atau menurunkan Konkurensi yang Disediakan berdasarkan profil lalu lintas Anda, sehingga mengoptimalkan biaya.

Berikut ini adalah prasyarat untuk autoscale Provisioned Concurrency pada endpoint tanpa server:

- Mendaftarkan model
- Menetapkan kebijakan penskalaan
- Menerapkan kebijakan penskalaan

Sebelum Anda dapat menggunakan penskalaan otomatis, Anda harus telah menerapkan model ke titik akhir tanpa server dengan Konkurensi yang Disediakan. Model yang dikerahkan disebut sebagai varian produksi. Lihat Buat konfigurasi titik akhir dan Buat titik akhir untuk informasi selengkapnya tentang penerapan model ke titik akhir tanpa server dengan Konkurensi yang Disediakan. Untuk menentukan metrik dan nilai target untuk kebijakan penskalaan, Anda harus mengonfigurasi kebijakan penskalaan. Untuk informasi selengkapnya tentang cara menentukan kebijakan penskalaan, lihatMenetapkan kebijakan penskalaan. Setelah mendaftarkan model Anda dan menentukan kebijakan penskalaan, terapkan kebijakan penskalaan ke model terdaftar. Untuk informasi tentang cara menerapkan kebijakan penskalaan, lihatMenerapkan kebijakan penskalaan.

Untuk detail tentang prasyarat dan komponen lain yang digunakan dengan penskalaan otomatis, lihat bagian Ikhtisar penskalaan otomatis dalam dokumentasi penskalaan otomatis. SageMaker

Mendaftarkan model

Untuk menambahkan penskalaan otomatis ke titik akhir tanpa server dengan Provisioned Concurrency, pertama-tama Anda harus mendaftarkan model Anda (varian produksi) menggunakan atau Application Auto Scaling API. AWS CLI

Daftarkan model (AWS CLI)

Untuk mendaftarkan model Anda, gunakan `register-scalable-target` AWS CLI perintah dengan parameter berikut:

- `--service-namespace` – Atur nilai ini ke `sagemaker`.
- `--resource-id`- Pengidentifikasi sumber daya untuk model (khususnya varian produksi). Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama varian produksi. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `--scalable-dimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.
- `--min-capacity`— Jumlah minimum Konkurensi Terprovisi untuk model. Setel `--min-capacity` ke setidaknya 1. Ini harus sama dengan atau kurang dari nilai yang ditentukan untuk`--max-capacity`.
- `--max-capacity`— Jumlah maksimum Konkurensi Terprovisi yang harus diaktifkan melalui Application Auto Scaling. Atur `--max-capacity` ke minimal 1. Ini harus lebih besar dari atau sama dengan nilai yang ditentukan untuk`--min-capacity`.

Contoh berikut menunjukkan cara mendaftarkan model bernama `MyVariant` yang diskalakan secara dinamis untuk memiliki 1 hingga 10 nilai Konkurensi yang Disediakan:

```
aws application-autoscaling register-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredProvisionedConcurrency \
    --resource-id endpoint/MyEndpoint/variant/MyVariant \
    --min-capacity 1 \
    --max-capacity 10
```

Daftarkan model (Application Auto Scaling API)

Untuk mendaftarkan model Anda, gunakan tindakan `RegisterScalableTarget` Application Auto Scaling API dengan parameter berikut:

- `ServiceNamespace` – Atur nilai ini ke `sagemaker`.
- `ResourceId`- Pengidentifikasi sumber daya untuk model (khususnya varian produksi). Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama varian produksi. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `ScalableDimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.
- `MinCapacity`— Jumlah minimum Konkurensi Terprovisi untuk model. Setel `MinCapacity` ke setidaknya 1. Ini harus sama dengan atau kurang dari nilai yang ditentukan untuk`MaxCapacity`.
- `MaxCapacity`— Jumlah maksimum Konkurensi Terprovisi yang harus diaktifkan melalui Application Auto Scaling. Atur `MaxCapacity` ke minimal 1. Ini harus lebih besar dari atau sama dengan nilai yang ditentukan untuk`MinCapacity`.

Contoh berikut menunjukkan cara mendaftarkan model bernama `MyVariant` yang diskalakan secara dinamis untuk memiliki 1 hingga 10 nilai Konkurensi yang Disediakan:

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/MyEndPoint/variant/MyVariant",
    "ScalableDimension": "sagemaker:variant:DesiredProvisionedConcurrency",
    "MinCapacity": 1,
    "MaxCapacity": 10
}
```

Menetapkan kebijakan penskalaan

Untuk menentukan metrik dan nilai target untuk kebijakan penskalaan, Anda dapat mengonfigurasi kebijakan penskalaan pelacakan target. Menetapkan kebijakan penskalaan sebagai blok JSON dalam file teks. Anda kemudian dapat menggunakan file teks tersebut saat memicu AWS

CLI atau Application Auto Scaling API. Untuk menentukan kebijakan penskalaan pelacakan target untuk titik akhir nirkabel server, gunakan metrik yang telah ditentukan sebelumnya. `SageMakerVariantProvisionedConcurrencyUtilization`

```
{
    "TargetValue": 0.5,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "SageMakerVariantProvisionedConcurrencyUtilization"
    },
    "ScaleOutCooldown": 1,
    "ScaleInCooldown": 1
}
```

Menerapkan kebijakan penskalaan

Setelah mendaftarkan model, Anda dapat menerapkan kebijakan penskalaan ke titik akhir tanpa server dengan Konkurensi yang Disediakan. Lihat Menerapkan kebijakan penskalaan pelacakan target untuk menerapkan kebijakan penskalaan pelacakan target yang telah Anda tetapkan. Jika arus lalu lintas ke titik akhir tanpa server memiliki rutinitas yang dapat diprediksi, alih-alih menerapkan kebijakan penskalaan pelacakan target, Anda mungkin ingin menjadwalkan tindakan penskalaan pada waktu tertentu. Untuk informasi lebih lanjut tentang penskalaan penskalaan, lihat. Penskalaan terjadwal

Menerapkan kebijakan penskalaan pelacakan target

Anda dapat menggunakanAWS Management Console, AWS CLI atau Application Auto Scaling API untuk menerapkan kebijakan penskalaan pelacakan target ke titik akhir niserver dengan Konkurensi Terprovisi.

Menerapkan kebijakan penskalaan pelacakan target () AWS CLI

Untuk menerapkan kebijakan penskalaan pada model Anda, gunakan perintah `put-scaling-policy`AWS CLI; dengan parameter berikut:

- `--policy-name` – Nama kebijakan penskalaan.
- `--policy-type` – Atur nilai ini ke `TargetTrackingScaling`.

- `--resource-id`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `--service-namespace` – Atur nilai ini ke `sagemaker`.
- `--scalable-dimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.
- `--target-tracking-scaling-policy-configuration`— Konfigurasi kebijakan penskalaan pelacakan target untuk model.

Contoh berikut menunjukkan cara menerapkan kebijakan penskalaan pelacakan target yang dinamai `MyScalingPolicy` ke model bernama. `MyVariant` Konfigurasi kebijakan disimpan dalam file bernama`scaling-policy.json`.

```
aws application-autoscaling put-scaling-policy \
    --policy-name MyScalingPolicy \
    --policy-type TargetTrackingScaling \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredProvisionedConcurrency \
    --resource-id endpoint/MyEndpoint/variant/MyVariant \
    --target-tracking-scaling-policy-configuration file://[file-localtion]/scaling-
policy.json
```

Menerapkan kebijakan penskalaan pelacakan target (Application Auto Scaling API)

Untuk menerapkan kebijakan penskalaan pada model Anda, gunakan tindakan `PutScalingPolicy` Application Auto Scaling API dengan parameter berikut:

- `PolicyName` – Nama kebijakan penskalaan.
- `PolicyType` – Atur nilai ini ke `TargetTrackingScaling`.
- `ResourceId`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `ServiceNamespace` – Atur nilai ini ke `sagemaker`.
- `ScalableDimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.

- `TargetTrackingScalingPolicyConfiguration`— Konfigurasi kebijakan penskalaan pelacakan target untuk model.

Contoh berikut menunjukkan cara menerapkan kebijakan penskalaan pelacakan target yang dinamai `MyScalingPolicy` ke model bernama. `MyVariant` Konfigurasi kebijakan disimpan dalam file bernama`scaling-policy.json`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "PolicyName": "MyScalingPolicy",
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/MyEndpoint/variant/MyVariant",
    "ScalableDimension": "sagemaker:variant:DesiredProvisionedConcurrency",
    "PolicyType": "TargetTrackingScaling",
    "TargetTrackingScalingPolicyConfiguration":
    {
        "TargetValue": 0.5,
        "PredefinedMetricSpecification":
        {
            "PredefinedMetricType": "SageMakerVariantProvisionedConcurrencyUtilization"
        }
    }
}
```

Menerapkan kebijakan penskalaan pelacakan target () AWS Management Console

Untuk menerapkan kebijakan penskalaan pelacakan target dengan: AWS Management Console

1. Masuk ke SageMakerkonsol Amazon.
2. Di panel navigasi, pilih Inferensi.
3. Pilih Endpoint untuk melihat daftar semua titik akhir Anda.

4. Pilih titik akhir yang ingin Anda terapkan kebijakan penskalaan. Halaman dengan pengaturan titik akhir akan muncul, dengan model (varian produksi) yang tercantum di bagian Pengaturan runtime Endpoint.
5. Pilih varian produksi yang ingin Anda terapkan kebijakan penskalaan, dan pilih Konfigurasi penskalaan auto. Kotak dialog penskalaan otomatis varian Konfigurasi muncul.

Configure variant automatic scaling

Deregister auto scaling

Variant automatic scaling Learn more

Variant name	Current max concurrency	Current provisioned concurrency
variant-name-1	20	11

Minimum provisioned concurrency

Maximum provisioned concurrency

IAM role

Amazon SageMaker uses the following service-linked role for automatic scaling. Learn more

AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint

Built-in scaling policy Learn more

Policy name

SageMakerServerlessEndpointProvisionedConcurrencyScalingPolicy

Target metric

SageMakerVariantProvisionedConcurrencyUtilization

Target value

Scale in cool down (seconds) - *optional*

300

Scale out cool down (seconds) - *optional*

300

Disable scale in

Select if you don't want automatic scaling to delete instances when traffic decreases. Learn more

Custom scaling policy Learn more

There are no custom scaling policies for this variant.

Cancel Save

6. Masukkan nilai Konkurensi Terprovisi minimum dan maksimum di kolom Konkurensi Minimum yang ditetapkan dan Konkurensi maksimum yang disediakan, masing-masing, di bagian Penskalaan otomatis Varian. Konkurensi Terprovisi Minimum harus kurang dari atau sama dengan Konkurensi Terprovisi maksimum.
7. Masukkan nilai target di bidang Nilai target untuk metrik target,`SageMakerVariantProvisionedConcurrencyUtilization`.
8. (Opsional) Masukkan skala di cool down dan skala keluar cool down values (dalam detik) di Skala di cool down dan Skala keluar cool down field masing-masing.
9. (Opsional) Pilih Nonaktifkan skala jika Anda tidak ingin penskalaan auto menghapus instance saat lalu lintas berkurang.
10. Pilih Simpan.

Penskalaan terjadwal

Jika lalu lintas ke titik akhir tanpa server dengan Provisioned Concurrency mengikuti pola rutin, Anda mungkin ingin menjadwalkan tindakan penskalaan pada waktu tertentu, untuk menskalakan atau menskalakan Konkurensi yang Disediakan. Anda dapat menggunakan AWS CLI atau Application Auto Scaling untuk menjadwalkan tindakan penskalaan.

Penskalaan terjadwal () AWS CLI

Untuk menerapkan kebijakan penskalaan pada model Anda, gunakan perintah `put-scheduled-action`AWS CLI; dengan parameter berikut:

- `--schedule-action-name`- Nama tindakan penskalaan.
- `--schedule`- Ekspresi cron yang menentukan waktu mulai dan akhir tindakan penskalaan dengan jadwal berulang.
- `--resource-id`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `--service-namespace` – Atur nilai ini ke `sagemaker`.
- `--scalable-dimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.
- `--scalable-target-action`- Target aksi penskalaan.

Contoh berikut menunjukkan bagaimana menambahkan tindakan penskalaan bernama `MyScalingAction` model bernama `MyVariant` pada jadwal berulang. Pada jadwal yang ditentukan (setiap hari pukul 09.15 UTC), jika Konkurensi Terprovisi saat ini berada di bawah nilai yang ditentukan. `MinCapacity` Application Auto Scaling menskalakan Konkurensi yang Disediakan dengan nilai yang ditentukan oleh. `MinCapacity`

```
aws application-autoscaling put-scheduled-action \
    --scheduled-action-name 'MyScalingAction' \
    --schedule 'cron(15 12 * * ? *)' \
    --service-namespace sagemaker \
    --resource-id endpoint/MyEndpoint/variant/MyVariant \
    --scalable-dimension sagemaker:variant:DesiredProvisionedConcurrency \
    --scalable-target-action 'MinCapacity=10'
```

Scaling terjadwal (Application Auto Scaling API)

Untuk menerapkan kebijakan penskalaan pada model Anda, gunakan tindakan `PutScheduledAction` Application Auto Scaling API dengan parameter berikut:

- `ScheduleActionName`- Nama tindakan penskalaan.
- `Schedule`- Ekspresi cron yang menentukan waktu mulai dan akhir tindakan penskalaan dengan jadwal berulang.
- `ResourceId`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `ServiceNamespace` – Atur nilai ini ke `sagemaker`.
- `ScalableDimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.
- `ScalableTargetAction`- Target aksi penskalaan.

Contoh berikut menunjukkan bagaimana menambahkan tindakan penskalaan bernama `MyScalingAction` model bernama `MyVariant` pada jadwal berulang. Pada jadwal yang ditentukan (setiap hari pukul 09.15 UTC), jika Konkurensi Terprovisi saat ini berada di bawah nilai yang ditentukan. `MinCapacity` Application Auto Scaling menskalakan Konkurensi yang Disediakan dengan nilai yang ditentukan oleh. `MinCapacity`

```
POST / HTTP/1.1
```

```
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.PutScheduledAction
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "ScheduledActionName": "MyScalingAction",
    "Schedule": "cron(15 12 * * ? *)",
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/MyEndpoint/variant/MyVariant",
    "ScalableDimension": "sagemaker:variant:DesiredProvisionedConcurrency",
    "ScalableTargetAction": "MinCapacity=10"
        }
    }
}
```

Menghapus kebijakan penskalaan

Anda dapat menghapus kebijakan penskalaan dengan AWS Management ConsoleAWS CLI, atau Application Auto Scaling API. Untuk informasi selengkapnya tentang menghapus kebijakan penskalaan denganAWS Management Console, lihat Menghapus kebijakan penskalaan di dokumentasi penskalaan SageMakerotomatis.

Hapus kebijakan penskalaan () AWS CLI

Untuk menerapkan kebijakan penskalaan pada model Anda, gunakan perintah `delete-scaling-policy`AWS CLI; dengan parameter berikut:

- `--policy-name` – Nama kebijakan penskalaan.
- `--resource-id`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `--service-namespace` – Atur nilai ini ke `sagemaker`.
- `--scalable-dimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.

Contoh berikut menghapus kebijakan penskalaan bernama `MyScalingPolicy` dari model bernama. `MyVariant`

```
aws application-autoscaling delete-scaling-policy \
    --policy-name MyScalingPolicy \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredProvisionedConcurrency \
    --resource-id endpoint/MyEndpoint/variant/MyVariant
```

Menghapus kebijakan penskalaan (Application Auto Scaling API)

Untuk menghapus kebijakan penskalaan pada model Anda, gunakan tindakan `DeleteScalingPolicy` Application Auto Scaling API dengan parameter berikut:

- `PolicyName` – Nama kebijakan penskalaan.
- `ResourceId`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `ServiceNamespace` – Atur nilai ini ke `sagemaker`.
- `ScalableDimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.

Contoh berikut menggunakan Application Auto Scaling API untuk menghapus kebijakan penskalaan bernama `MyScalingPolicy` dari model bernama. `MyVariant`

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "PolicyName": "MyScalingPolicy",
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/MyEndpoint/variant/MyVariant",
```

```
    "ScalableDimension": "sagemaker:variant:DesiredProvisionedConcurrency",
}
```

Membatalkan pendaftaran model

Anda dapat membatalkan pendaftaran model dengan AWS Management ConsoleAWS CLI, atau Application Auto Scaling API.

Deregister model () AWS CLI

Untuk membatalkan pendaftaran model dari Application Auto Scaling, gunakan perintah `deregister-scalable-target`AWS CLI; dengan parameter berikut:

- `--resource-id`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `--service-namespace` – Atur nilai ini ke `sagemaker`.
- `--scalable-dimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.

Contoh berikut membatalkan pendaftaran model bernama Application Auto `MyVariant` Scaling.

```
aws application-autoscaling deregister-scalable-target \
    --service-namespace sagemaker \
    --scalable-dimension sagemaker:variant:DesiredProvisionedConcurrency \
    --resource-id endpoint/MyEndpoint/variant/MyVariant
```

Membatalkan pendaftaran model (Application Auto Scaling API)

Untuk membatalkan pendaftaran model dari Application Auto Scaling, gunakan tindakan `DeregisterScalableTarget` Application Auto Scaling API dengan parameter berikut:

- `ResourceId`- Pengenal sumber daya untuk varian. Untuk parameter ini, jenis sumber daya adalah `endpoint` dan pengenal unik adalah nama variannya. Misalnya, `endpoint/MyEndpoint/variant/MyVariant`.
- `ServiceNamespace` – Atur nilai ini ke `sagemaker`.

- `ScalableDimension` – Atur nilai ini ke `sagemaker:variant:DesiredProvisionedConcurrency`.

Contoh berikut menggunakan Application Auto Scaling API untuk membatalkan pendaftaran model bernama `MyVariant` dari Application Auto Scaling.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AnyScaleFrontendService.DeregisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "ServiceNamespace": "sagemaker",
    "ResourceId": "endpoint/MyEndpoint/variant/MyVariant",
    "ScalableDimension": "sagemaker:variant:DesiredProvisionedConcurrency",
}
```

Deregister model () AWS Management Console

Untuk membatalkan pendaftaran model (varian produksi) dengan: AWS Management Console

1. Buka SageMakerkonsol Amazon.
2. Di panel navigasi, pilih Inferensi.
3. Pilih Endpoint untuk melihat daftar titik akhir Anda.
4. Pilih titik akhir server yang menghosting varian produksi. Halaman dengan pengaturan titik akhir akan muncul, dengan varian produksi yang tercantum di bagian Pengaturan runtime Endpoint.
5. Pilih varian produksi yang ingin Anda batalkan pendaftaran, dan pilih Konfigurasi auto Scaling. Kotak dialog penskalaan otomatis varian Konfigurasi muncul.
6. Pilih Deregister penskalaan auto.

Pemecahan Masalah

Jika Anda mengalami masalah dengan Inferensi Tanpa Server, lihat tips pemecahan masalah berikut.

Masalah kontainer

Jika wadah yang Anda gunakan untuk titik akhir tanpa server sama dengan yang Anda gunakan pada titik akhir berbasis instans, wadah Anda mungkin tidak memiliki izin untuk menulis file. Hal ini dapat terjadi karena alasan berikut:

- Titik akhir tanpa server Anda gagal membuat atau memperbarui karena kegagalan pemeriksaan kesehatan ping.
- Amazon CloudWatch log untuk titik akhir menunjukkan bahwa wadah gagal menulis ke beberapa file atau direktori karena kesalahan izin.

Untuk memperbaiki masalah ini, Anda dapat mencoba menambahkan izin membaca, menulis, dan mengeksekusi`other`pada file atau direktori dan kemudian membangun kembali wadah. Anda dapat melakukan langkah-langkah berikut untuk menyelesaikan proses ini:

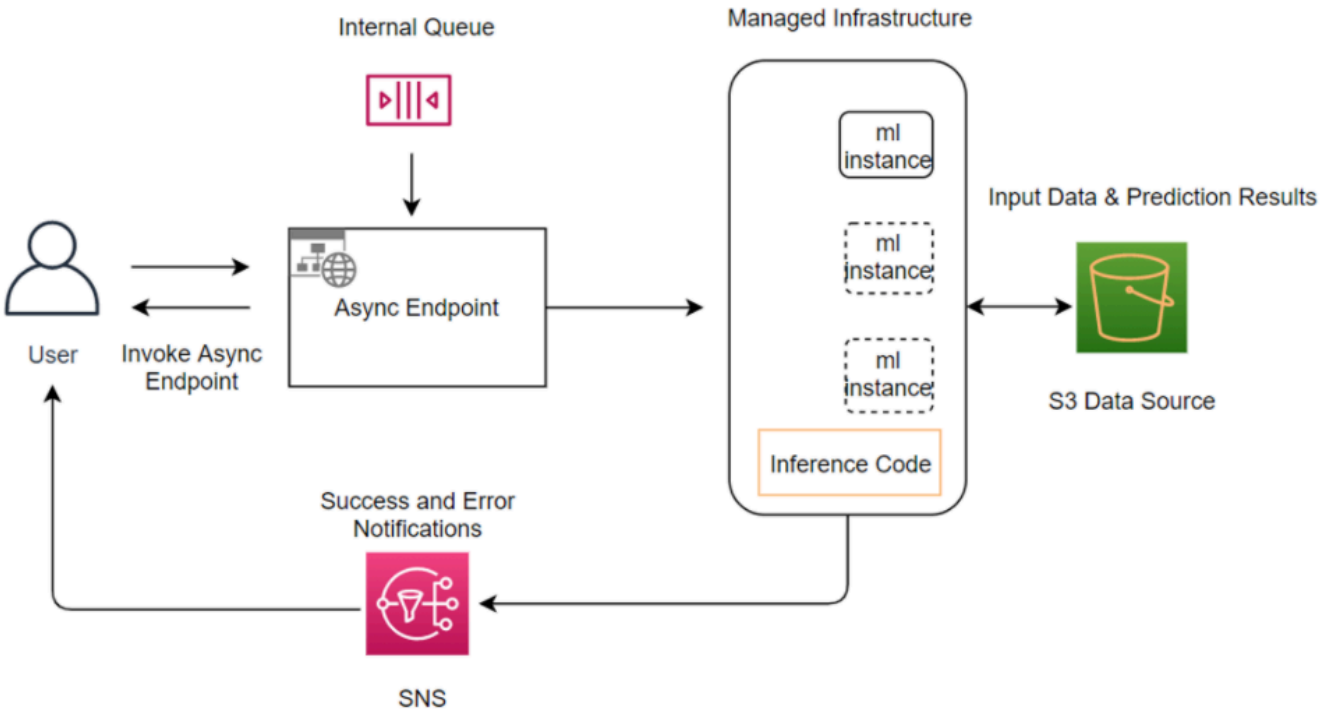
1. Di Dockerfile yang Anda gunakan untuk membangun kontainer Anda, tambahkan perintah berikut:`RUN chmod o+rwX` *`<file or directory name>`*
2. Membangun kembali wadah.
3. Unggah citra kontainer baru ke Amazon ECR.
4. Cobalah untuk membuat atau memperbarui titik akhir tanpa server lagi.

Inferensi asinkron

Amazon SageMaker Asynchronous Inference adalah kemampuan yang mengantrekan permintaan yang masuk dan SageMaker memprosesnya secara asinkron. Opsi ini sangat ideal untuk permintaan dengan ukuran muatan besar (hingga 1GB), waktu pemrosesan yang lama (hingga satu jam), dan persyaratan latensi mendekati waktu nyata. Inferensi Asinkron memungkinkan Anda menghemat biaya dengan menskalakan otomatis jumlah instans ke nol saat tidak ada permintaan untuk diproses, jadi Anda hanya membayar saat titik akhir memproses permintaan.

Cara Kerjanya

Membuat titik akhir inferensi asinkron mirip dengan membuat titik akhir inferensi waktu nyata. Anda dapat menggunakan SageMaker model yang ada dan hanya perlu menentukan `AsyncInferenceConfig` objek saat membuat konfigurasi titik akhir Anda dengan `EndpointConfig` bidang di `CreateEndpointConfig` API. Diagram berikut menunjukkan arsitektur dan alur kerja Inferensi Asinkron.



Untuk menjalankan endpoint, Anda harus menempatkan payload permintaan di Amazon S3 dan memberikan pointer ke payload ini sebagai bagian dari permintaan. `InvokeEndpointAsync` Setelah pemanggilan, SageMaker antrian permintaan untuk diproses dan mengembalikan pengenal dan lokasi output sebagai respons. Setelah diproses, SageMaker tempatkan hasilnya di lokasi Amazon S3. Anda secara opsional dapat memilih untuk menerima pemberitahuan sukses atau kesalahan dengan Amazon SNS. Untuk informasi selengkapnya tentang cara mengatur notifikasi asinkron, lihat. Hasil prediksi

> **Note**
>
> Kehadiran objek konfigurasi inferensi asinkron (`AsyncInferenceConfig`) dalam konfigurasi titik akhir menyiratkan bahwa titik akhir hanya dapat menerima pemanggilan asinkron.

Bagaimana Saya Memulai?

Jika Anda adalah pengguna pertama kali Inferensi SageMaker Asinkron Amazon, kami sarankan Anda melakukan hal berikut:

- Baca Membuat, memanggil, dan memperbarui Endpoint Asynchronous untuk informasi tentang cara membuat, memanggil, memperbarui, dan menghapus titik akhir asinkron.

- Jelajahi notebook contoh Inferensi Asinkron di aws/ repositori. amazon-sagemaker-examples GitHub

Perhatikan bahwa jika titik akhir Anda menggunakan salah satu fitur yang tercantum di Pengecualian halaman ini, Anda tidak dapat menggunakan Inferensi Asinkron.

Membuat, memanggil, dan memperbarui Endpoint Asynchronous

Panduan ini menunjukkan prasyarat yang harus Anda penuhi untuk membuat titik akhir asinkron, bersama dengan cara membuat, memanggil, dan menghapus titik akhir asinkron Anda. Anda dapat membuat, memperbarui, menghapus, dan memanggil titik akhir asinkron denganAWS SDK dan Amazon SageMaker Python SDK.

Topik

- Prasyarat
- Membuat Endpoint Inferensi Asinkron
- Memanggil Endpoint Asynchronous
- Memperbarui Endpoint Asynchronous
- Menghapus Titik Akhir Asinkron

Prasyarat

Untuk menggunakan titik akhir asinkron, pertama-tama pastikan Anda telah memenuhi prasyarat ini.

1. Buat IAM role untuk Amazon SageMaker.

   Inference Asinkron membutuhkan akses ke URI bucket Amazon S3. Untuk memfasilitasi ini, buat peran IAM yang dapat berjalan SageMaker dan memiliki izin untuk mengakses Amazon S3 dan Amazon SNS. Menggunakan peran ini, SageMaker dapat berjalan di bawah akun Anda dan mengakses bucket Amazon S3 dan topik Amazon SNS Anda.

   Anda dapat membuat peran IAM dengan menggunakan konsol IAM,AWS SDK for Python (Boto3), atauAWS CLI. Berikut ini adalah contoh cara membuat peran IAM dan melampirkan kebijakan yang diperlukan dengan konsol IAM.

   a. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

b. Di panel navigasi konsol IAM, pilih Peran, dan lalu pilih Buat peran.

c. Untuk Pilih jenis entitas tepercaya, pilih layanan AWS.

d. Pilih layanan yang ingin Anda perbolehkan untuk mengasumsikan peran ini. Dalam hal ini, pilih SageMaker. Kemudian pilih Selanjutnya: Izin.

   - Ini secara otomatis membuat kebijakan IAM yang memberikan akses ke layanan terkait seperti Amazon S3, Amazon ECR, dan CloudWatch Log.

e. Pilih Selanjutnya: Tanda.

f. (Opsional) Tambahkan metadata ke dalam peran dengan melampirkan tanda sebagai pasangan nilai-kunci. Untuk informasi lebih lanjut tentang penggunaan tanda dan IAM, lihat Penandaan sumber daya IAM.

g. Pilih Next: Review (Selanjutnya: Tinjauan).

h. Ketik nama Peran.

i. Jika memungkinkan, ketikkan nama peran atau akhiran nama peran. Nama peran harus unik di akun AWS Anda. Grup tidak dibedakan berdasarkan huruf besar-kecil. Misalnya, Anda tidak dapat membuat peran dengan nama `PRODROLE` dan `prodrole`. Karena sumber daya AWS lainnya mungkin merujuk peran tersebut, Anda tidak dapat mengubah nama peran tersebut setelah nama dibuat.

j. (Opsional) Untuk Deskripsi peran, ketikkan deskripsi untuk peran baru tersebut.

k. Tinjau peran dan kemudian pilih Buat peran.

   Perhatikan SageMaker ARN. Untuk menemukan ARN peran, lakukan hal berikut:

   i. Buka konsol IAM: https://console.aws.amazon.com/iam/

   ii. Pilih Peran.

   iii. Cari peran yang baru saja Anda buat dengan mengetikkan nama peran di kolom pencarian.

   iv. Pilih peran.

   v. ARN peran ada di bagian atas halaman Ringkasan.

2. Tambahkan Izin Amazon SageMaker, Amazon S3, dan Amazon SNS ke Peran IAM Anda.

   Setelah peran dibuat, berikan SageMaker, Amazon S3, dan izin Amazon SNS opsional untuk peran IAM Anda.

Pilih Peran di konsol IAM. Cari peran yang Anda buat dengan mengetikkan nama peran Anda di bidang Pencarian.

a. Pilih peran Anda.

b. Selanjutnya, pilih Lampirkan Kebijakan.

c. Amazon SageMaker Asynchronous Inference memerlukan izin untuk melakukan tindakan berikut:`"sagemaker:CreateModel"`,`"sagemaker:CreateEndpointConfig"`,`"sagemaker:C` dan`"sagemaker:InvokeEndpointAsync"`.

   Tindakan ini termasuk dalam`AmazonSageMakerFullAccess` kebijakan. Tambahkan kebijakan ini ke peran IAM Anda. Cari`AmazonSageMakerFullAccess` di kolom Pencarian. Pilih `AmazonSageMakerFullAccess`.

d. Pilih Lampirkan kebijakan.

e. Selanjutnya, pilih Lampirkan Kebijakan untuk menambahkan izin Amazon S3.

f. Pilih Buat kebijakan.

g. Pilih`JSON` tab.

h. Tambahkan pernyataan kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:AbortMultipartUpload",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::bucket_name/*"
        }
    ]
}
```

i. Pilih Next: Tags (Selanjutnya: Tanda).

j. Ketik nama Kebijakan.

k. Pilih Buat kebijakan.

l. Ulangi langkah yang sama yang Anda selesaikan untuk menambahkan izin Amazon S3 untuk menambahkan izin Amazon SNS. Untuk pernyataan kebijakan, lampirkan yang berikut ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sns:Publish"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:sns:<region>:<Account_ID>:<SNS_Topic>"
        }
    ]
}
```

3. Unggah data inferensi Anda (misalnya, model machine learning, data sampel) ke Amazon S3.
4. Pilih gambar inferensi Docker bawaan atau buat Inference Docker Image Anda sendiri.

   SageMaker menyediakan wadah untuk algoritme bawaan dan gambar Docker bawaan untuk beberapa kerangka pembelajaran mesin yang paling umum, seperti Apache MXNet, TensorFlow, PyTorch, dan Chainer. Untuk daftar lengkap gambar yang tersedia, lihat SageMaker Gambar Deep Learning Containers yang Tersedia. Jika Anda memilih untuk menggunakan wadah yang SageMaker disediakan, Anda dapat meningkatkan batas waktu akhir dan ukuran payload dari default dengan menetapkan variabel lingkungan dalam wadah. Untuk mempelajari cara mengatur variabel lingkungan yang berbeda untuk setiap kerangka kerja, lihat langkah Buat Model untuk membuat titik akhir asinkron.

   Jika tidak ada SageMaker kontainer yang ada memenuhi kebutuhan Anda dan Anda tidak memiliki container yang sudah ada, Anda mungkin perlu membuat container Docker baru. LihatGunakan kode inferensi Anda sendiri informasi tentang cara membuat image Docker Anda.
5. Membuat topik Amazon SNS (opsional)

   Buat topik Amazon Simple Notification Service (Amazon SNS) yang mengirimkan notifikasi tentang permintaan yang telah menyelesaikan pemrosesan. Amazon SNS adalah layanan notifikasi untuk aplikasi berorientasi pesan, dengan beberapa pelanggan meminta dan menerima pemberitahuan “push” pesan kritis waktu melalui pilihan protokol transportasi, termasuk HTTP, Amazon SQS, dan email. Anda dapat menentukan topik Amazon SNS

saat membuat`EndpointConfig` objek saat Anda menentukan`AsyncInferenceConfig` menggunakan`EndpointConfig` API.

Ikuti langkah-langkah untuk membuat dan berlangganan topik Amazon SNS.

a. Menggunakan konsol Amazon SNS membuat topik. Untuk petunjuk, lihat Membuat topik Amazon SNS di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

b. Berlangganan topik. Untuk petunjuk, lihat Berlangganan topik Amazon SNS di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

c. Saat Anda menerima email yang meminta Anda mengonfirmasi langganan Anda ke topik tersebut, konfirmasikan langganan.

d. Perhatikan topik Amazon Resource Name (ARN). Topik Amazon SNS yang Anda buat adalah sumber daya lain di AWS akun Anda, dan memiliki ARN unik. ARN adalah dalam format berikut:

```
arn:aws:sns:aws-region:account-id:topic-name
```

Untuk informasi selengkapnya tentang Amazon SNS, lihat Panduan Developer Amazon SNS.

Membuat Endpoint Inferensi Asinkron

Buat endpoint asynchronous dengan cara yang sama Anda akan membuat endpoint menggunakan layanan SageMaker hosting:

- Buat model SageMaker dengan`CreateModel`.
- Buat konfigurasi endpoint dengan`CreateEndpointConfig`.
- Buat endpoint HTTPS dengan`CreateEndpoint`.

Untuk membuat titik akhir, pertama-tama Anda membuat model dengan `CreateModel`, di mana Anda menunjuk ke artefak model dan jalur registri Docker (Image). Anda kemudian membuat konfigurasi menggunakan `CreateEndpointConfig`tempat Anda menentukan satu atau beberapa model yang dibuat menggunakan`CreateModel` API untuk diterapkan dan sumber daya yang SageMaker ingin Anda sediakan. Buat endpoint Anda dengan `CreateEndpoint`menggunakan konfigurasi endpoint yang ditentukan dalam permintaan. Anda dapat memperbarui titik akhir asinkron dengan `UpdateEndpoint`API. Kirim dan terima permintaan inferensi dari model yang dihosting

di titik akhir dengan`InvokeEndpointAsync`. Anda dapat menghapus endpoint Anda dengan [`DeleteEndpoint`](#)API.

Untuk daftar lengkap Gambar yang tersedia, lihat SageMaker Gambar Deep Learning Containers yang Tersedia. LihatGunakan kode inferensi Anda sendiri informasi tentang cara membuat image Docker Anda.

Membuat Model

Contoh berikut menunjukkan cara membuat model menggunakanAWS SDK for Python (Boto3). Beberapa baris pertama mendefinisikan:

- `sagemaker_client`: Objek SageMaker klien tingkat rendah yang memudahkan untuk mengirim dan menerima permintaan keAWS layanan.
- `sagemaker_role`: Variabel string dengan SageMaker IAM role Amazon Resource Name (ARN).
- `aws_region`: Variabel string dengan namaAWS wilayah Anda.

```
import boto3

# Specify your AWS Region
aws_region='<aws_region>'

# Create a low-level SageMaker service client.
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# Role to give SageMaker permission to access AWS services.
sagemaker_role= "arn:aws:iam::<account>:role/*"
```

Selanjutnya, tentukan lokasi model pra-latih yang disimpan di Amazon S3. Dalam contoh ini, kita menggunakan model XGBoost pra-terlatih bernama`demo-xgboost-model.tar.gz`. URI Amazon S3 lengkap disimpan dalam variabel string`model_url`:

```
#Create a variable w/ the model S3 URI
s3_bucket = '<your-bucket-name>' # Provide the name of your S3 bucket
bucket_prefix='saved_models'
model_s3_key = f"{bucket_prefix}/demo-xgboost-model.tar.gz"

#Specify S3 bucket w/ model
model_url = f"s3://{s3_bucket}/{model_s3_key}"
```

Tentukan wadah utama. Untuk wadah utama, Anda menentukan image Docker yang berisi kode inferensi, artefak (dari pelatihan sebelumnya), dan peta lingkungan kustom yang digunakan kode inferensi saat Anda menerapkan model untuk prediksi.

Dalam contoh ini, kita menentukan XGBoost built-in algoritma gambar kontainer:

```
from sagemaker import image_uris

# Specify an AWS container image.
container = image_uris.retrieve(region=aws_region, framework='xgboost',
 version='0.90-1')
```

Buat model di Amazon SageMaker dengan`CreateModel`. Tentukan hal berikut:

- `ModelName`: Nama untuk model Anda (dalam contoh ini disimpan sebagai variabel string yang disebut`model_name`).
- `ExecutionRoleArn`: Amazon Resource Name (ARN) dari IAM role yang SageMaker dapat diasumsikan Amazon Amazon untuk mengakses artifact model dan citra Docker untuk deployment pada instans komputasi MLatau untuk tugas transformasi batch.
- `PrimaryContainer`: Lokasi citra Docker utama yang berisi kode inferensi, artifact terkait, dan peta lingkungan kustom yang digunakan kode inferensi saat model di-deploy untuk prediksi.

```
model_name = '<The_name_of_the_model>'

#Create model
create_model_response = sagemaker_client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = sagemaker_role,
    PrimaryContainer = {
        'Image': container,
        'ModelDataUrl': model_url,
    })
```

Lihat `CreateModel`deskripsi di Panduan Referensi SageMaker API untuk daftar lengkap parameter API.

Jika Anda menggunakan wadah yang SageMaker disediakan, Anda dapat meningkatkan batas waktu server model dan ukuran payload dari nilai default ke maksimum yang didukung kerangka kerja dengan menetapkan variabel lingkungan pada langkah ini. Anda mungkin tidak dapat memanfaatkan

batas waktu maksimum dan ukuran payload yang didukung Inferensi Asinkron jika Anda tidak secara eksplisit menyetel variabel ini. Contoh berikut menunjukkan bagaimana Anda dapat mengatur variabel lingkungan lingkungan untuk wadah PyTorch Inference TorchServe.

```
model_name = '<The_name_of_the_model>'

#Create model
create_model_response = sagemaker_client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = sagemaker_role,
    PrimaryContainer = {
        'Image': container,
        'ModelDataUrl': model_url,
        'Environment': {
            'TS_MAX_REQUEST_SIZE': '100000000',
            'TS_MAX_RESPONSE_SIZE': '100000000',
            'TS_DEFAULT_RESPONSE_TIMEOUT': '1000'
        },
    })
```

Setelah Anda selesai membuat titik akhir, Anda harus menguji bahwa Anda telah mengatur variabel lingkungan dengan benar dengan mencetaknya dari`inference.py` skrip Anda. Tabel berikut mencantumkan variabel lingkungan untuk beberapa kerangka kerja yang dapat Anda atur untuk mengubah nilai default.

Kerangka Kerja	Variabel lingkungan
PyTorch 1.8 (berdasarkan TorchServe)	'TS_MAX_REQUEST_SIZE': '100000000' 'TS_MAX_RESPONSE_SIZE': '100000000' 'TS_DEFAULT_RESPONSE_TIMEOUT': '1000'
PyTorch 1.4 (berdasarkan MMS)	'MMS_MAX_REQUEST_SIZE': '1000000000' 'MMS_MAX_RESPONSE_SIZE': '1000000000' 'MMS_DEFAULT_RESPONSE_TIMEOUT': '900'

Kerangka Kerja	Variabel lingkungan
HuggingFace Kontainer Inferensi (berdasarkan MMS)	'MMS_MAX_REQUEST_SIZE': '2000000000' 'MMS_MAX_RESPONSE_SIZE': '2000000000' 'MMS_DEFAULT_RESPONSE_TIMEOUT': '900'

Membuat Konfigurasi Titik Akhir

Setelah Anda memiliki model, buat konfigurasi endpoint dengan `CreateEndpointConfig`. SageMaker Layanan hosting Amazon menggunakan konfigurasi ini untuk menerapkan model. Dalam konfigurasi, Anda mengidentifikasi satu atau beberapa model, yang dibuat menggunakan `CreateModel`, untuk menerapkan sumber daya yang SageMaker ingin disediakan Amazon. Tentukan`AsyncInferenceConfig` objek dan berikan lokasi Amazon S3 keluaran untuk`OutputConfig`. Anda dapat secara opsional menentukan topik Amazon SNS untuk mengirim pemberitahuan tentang hasil prediksi. Untuk informasi selengkapnya tentang topik Amazon SNS, lihat Mengonfigurasi Amazon SNS.

Contoh berikut menunjukkan cara membuat konfigurasi titik akhir menggunakan AWS SDK for Python (Boto3):

```
import datetime
from time import gmtime, strftime

# Create an endpoint config name. Here we create one based on the date
# so it we can search endpoints based on creation time.
endpoint_config_name = f"XGBoostEndpointConfig-{strftime('%Y-%m-%d-%H-%M-%S',
 gmtime())}"

# The name of the model that you want to host. This is the name that you specified when
 creating the model.
model_name='<The_name_of_your_model>'

create_endpoint_config_response = sagemaker_client.create_endpoint_config(
    EndpointConfigName=endpoint_config_name, # You will specify this name in a
 CreateEndpoint request.
    # List of ProductionVariant objects, one for each model that you want to host at
 this endpoint.
```

```
    ProductionVariants=[
        {
            "VariantName": "variant1", # The name of the production variant.
            "ModelName": model_name,
            "InstanceType": "ml.m5.xlarge", # Specify the compute instance type.
            "InitialInstanceCount": 1 # Number of instances to launch initially.
        }
    ],
    AsyncInferenceConfig={
        "OutputConfig": {
            # Location to upload response outputs when no location is provided in the
 request.
            "S3OutputPath": f"s3://{s3_bucket}/{bucket_prefix}/output"
            # (Optional) specify Amazon SNS topics
            "NotificationConfig": {
                "SuccessTopic": "arn:aws:sns:aws-region:account-id:topic-name",
                "ErrorTopic": "arn:aws:sns:aws-region:account-id:topic-name",
            }
        },
        "ClientConfig": {
            # (Optional) Specify the max number of inflight invocations per instance
            # If no value is provided, Amazon SageMaker will choose an optimal value
 for you
            "MaxConcurrentInvocationsPerInstance": 4
        }
    }
)

print(f"Created EndpointConfig:
 {create_endpoint_config_response['EndpointConfigArn']}")
```

Dalam contoh yang disebutkan di atas, Anda menentukan kunci berikut`OutputConfig` untuk`AsyncInferenceConfig` bidang:

- `S3OutputPath`: Lokasi untuk mengunggah output respons ketika tidak ada lokasi yang disediakan dalam permintaan.
- `NotificationConfig`: (Opsional) topik SNS yang memposting pemberitahuan kepada Anda ketika permintaan inferensi berhasil (`SuccessTopic`) atau jika gagal (`ErrorTopic`).

Anda juga dapat menentukan argumen opsional berikut untuk`ClientConfig` di`AsyncInferenceConfig` kolom:

- `MaxConcurrentInvocationsPerInstance`: (Opsional) Jumlah maksimum permintaan bersamaan yang dikirim oleh SageMaker klien ke wadah model.

Membuat Titik Akhir

Setelah Anda memiliki konfigurasi model dan titik akhir, gunakan `CreateEndpoint`API untuk membuat titik akhir Anda. Nama titik akhir harus unik dalam suatuAWS Wilayah diAWS akun Anda.

Berikut ini menciptakan endpoint menggunakan konfigurasi endpoint yang ditentukan dalam permintaan. Amazon SageMaker menggunakan titik akhir untuk menyediakan sumber daya dan men-deploy model.

```
# The name of the endpoint.The name must be unique within an AWS Region in your AWS
 account.
endpoint_name = '<endpoint-name>'

# The name of the endpoint configuration associated with this endpoint.
endpoint_config_name='<endpoint-config-name>'

create_endpoint_response = sagemaker_client.create_endpoint(
                                            EndpointName=endpoint_name,
                                            EndpointConfigName=endpoint_config_name)
```

Saat Anda memanggil`CreateEndpoint` API, Amazon SageMaker Asynchronous Inference mengirimkan pemberitahuan pengujian untuk memeriksa apakah Anda telah mengonfigurasi topik Amazon SNS. Amazon SageMaker Asynchronous Inference juga mengirimkan pemberitahuan pengujian setelah panggilan ke`UpdateEndpoint` dan`UpdateEndpointWeightsAndCapacities`. Hal SageMaker ini memungkinkan Anda memiliki izin yang diperlukan. Pemberitahuan dapat diabaikan begitu saja. Notifikasi pengujian memiliki bentuk sebagai berikut:

```
{
    "eventVersion":"1.0",
    "eventSource":"aws:sagemaker",
    "eventName":"TestNotification"
}
```

Memanggil Endpoint Asynchronous

Dapatkan kesimpulan dari model yang dihosting di titik akhir asinkron Anda`InvokeEndpointAsync`.

> **Note**
>
> Jika Anda belum melakukannya, unggah data inferensi Anda (misalnya, model machine learning, data sampel) ke Amazon S3.

Tentukan bidang berikut dalam permintaan Anda:

- Untuk`InputLocation`, tentukan lokasi data inferensi Anda.
- Untuk`EndpointName`, tentukan nama titik akhir Anda.
- (Opsional) Untuk`InvocationTimeoutSeconds`, Anda dapat mengatur batas waktu maksimal untuk permintaan. Anda dapat mengatur nilai ini hingga maksimum 3600 detik (satu jam) berdasarkan per permintaan. Jika Anda tidak menentukan bidang ini dalam permintaan Anda, secara default waktu permintaan habis pada 15 menit.

```
# Create a low-level client representing Amazon SageMaker Runtime
sagemaker_runtime = boto3.client("sagemaker-runtime", region_name=<aws_region>)

# Specify the location of the input. Here, a single SVM sample
input_location = "s3://bucket-name/test_point_0.libsvm"

# The name of the endpoint. The name must be unique within an AWS Region in your AWS
 account.
endpoint_name='<endpoint-name>'

# After you deploy a model into production using SageMaker hosting
# services, your client applications use this API to get inferences
# from the model hosted at the specified endpoint.
response = sagemaker_runtime.invoke_endpoint_async(
                            EndpointName=endpoint_name,
                            InputLocation=input_location,
                            InvocationTimeoutSeconds=3600)
```

Anda menerima respons sebagai string JSON dengan ID permintaan Anda dan nama bucket Amazon S3 yang akan mendapat respons terhadap panggilan API setelah diproses.

Memperbarui Endpoint Asynchronous

Perbarui titik akhir asinkron dengan UpdateEndpointAPI. Saat Anda memperbarui titik akhir, ketentuan SageMaker pertama dan beralih ke konfigurasi titik akhir baru yang Anda tentukan sebelum menghapus sumber daya yang disediakan dalam konfigurasi titik akhir sebelumnya. Jangan menghapus`EndpointConfig` dengan titik akhir yang hidup atau saat`UpdateEndpoint` atau`CreateEndpoint` operasi sedang dilakukan pada titik akhir.

```
# The name of the endpoint. The name must be unique within an AWS Region in your AWS
 account.
endpoint_name='<endpoint-name>'

# The name of the endpoint configuration associated with this endpoint.
endpoint_config_name='<endpoint-config-name>'

sagemaker_client.update_endpoint(
                                EndpointConfigName=endpoint_config_name,
                                EndpointName=endpoint_name
                                )
```

Saat Amazon SageMaker menerima permintaan, Amazon akan menetapkan status titik akhir ke Pembaruan. Setelah memperbarui titik akhir asinkron, ia menetapkan status ke InService. Untuk memeriksa status titik akhir, gunakan DescribeEndpointAPI. Untuk daftar lengkap parameter yang dapat Anda tentukan saat memperbarui titik akhir, lihat UpdateEndpointAPI.

Menghapus Titik Akhir Asinkron

Hapus titik akhir asinkron dengan cara yang mirip dengan cara Anda menghapus endpoint yang SageMaker di-host dengan DeleteEndpointAPI. Tentukan nama titik akhir asinkron yang ingin Anda hapus. Saat Anda menghapus titik akhir, SageMaker bebaskan semua sumber daya yang diterapkan saat titik akhir dibuat. Menghapus model tidak menghapus artefak model, kode inferensi, atau peran IAM yang Anda tentukan saat membuat model.

Hapus SageMaker model Anda dengan DeleteModelAPI atau dengan SageMaker konsol.

Boto3

```
import boto3

# Create a low-level SageMaker service client.
```

```
sagemaker_client = boto3.client('sagemaker', region_name=<aws_region>)
sagemaker_client.delete_endpoint(EndpointName='<endpoint-name>')
```

SageMaker console

1. Arahkan ke SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Perluas daftar dropdown Inferensi.
3. Pilih Endpoint.
4. Cari titik akhir di bilah pencarian titik akhir Cari.
5. Pilih titik akhir.
6. Pilih Delete (Hapus).

Selain menghapus titik akhir asinkron, Anda mungkin ingin membersihkan sumber daya lain yang digunakan untuk membuat titik akhir, seperti repositori Amazon ECR (jika Anda membuat gambar inferensi khusus), SageMaker model, dan konfigurasi titik akhir asinkron itu sendiri.

Memantau titik akhir asinkron

Anda dapat memantau SageMaker menggunakan Amazon CloudWatchIni berarti bahwa Anda dapat mengumpulkan data mentah untuk memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Dengan Amazon CloudWatchArtinya Anda dapat mengakses informasi historis untuk mendapatkan perspektif yang lebih baik tentang performa aplikasi web atau layanan Anda. Untuk mendapatkan informasi lebih lanjut tentang Amazon CloudWatch, lihatApa itu Amazon CloudWatch?

Pemantauan CloudWatch dengan

Metrik di bawah ini adalah daftar lengkap metrik untuk titik akhir asinkron dan ada di`AWS/SageMaker`namespace. Metrik apa pun yang tidak tercantum di bawah ini tidak dipublikasikan jika titik akhir diaktifkan untuk inferensi asinkron. Metrik tersebut mencakup (tetapi tidak terbatas pada):

- OverheadLatency
- Invokasi
- InvocationsPerInstance

Metrik Titik Akhir Umum

Metrik ini sama dengan metrik yang diterbitkan untuk titik akhir waktu nyata hari ini. Untuk mendapatkan informasi lebih lanjut tentang metrik lainnya di Amazon CloudWatch, lihatMonitor SageMaker dengan Amazon CloudWatch.

Nama Metrik	Deskripsi	Unit/Statistik
`Invocation4XXErrors`	Jumlah permintaan di mana model mengembalikan kode respons HTTP 4xx. Untuk setiap respons 4xx, 1 dikirim; jika tidak, 0 dikirim.	Unit: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`Invocation5XXErrors`	Jumlah InvokeEndpoint permintaan di mana model mengembalikan kode respons HTTP 5xx. Untuk setiap respons 5xx, 1 dikirim; jika tidak, 0 dikirim.	Unit: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`ModelLatency`	Interval waktu yang dibutuhkan oleh model untuk merespons sebagaimana dilihat dari SageMaker. Interval ini mencakup waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah.	Satuan: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel

Metrik Titik Akhir Inferensi Asinkron

Metrik ini diterbitkan untuk titik akhir yang diaktifkan untuk inferensi asinkron. Metrik berikut diterbitkan dengan`EndpointName`dimensi:

Nama Metrik	Deskripsi	Unit/Statistik
`ApproximateBacklogSize`	Jumlah item dalam antrian untuk titik akhir yang saat ini sedang diproses atau belum diproses.	Unit: Count (Jumlah) Statistik yang valid: Rata-rata, Maks, Min
`ApproximateBacklogSizePerInstance`	Jumlah item dalam antrian dibagi dengan jumlah instance di belakang titik akhir. Metrik ini terutama digunakan untuk menyiapkan penskalaan otomatis aplikasi untuk titik akhir berkemampuan asinkron.	Unit: Count (Jumlah) Statistik yang valid: Rata-rata, Maks, Min
`ApproximateAgeOfOldestRequest`	Usia permintaan tertua dalam antrian.	Unit: detik Statistik yang valid: Rata-rata, Maks, Min
`HasBacklogWithoutCapacity`	Nilai metrik ini adalah`1`ketika ada permintaan dalam antrian tetapi nol contoh di belakang titik akhir. Nilainya adalah`0`di semua waktu lainnya. Anda dapat menggunakan metrik ini untuk menskalakan otomatis titik akhir Anda dari nol instance setelah menerima permintaan baru dalam antrian.	Unit: Count (Jumlah) Statistik yang valid: Rata-rata

Metrik berikut diterbitkan dengan`EndpointName`dan`VariantName`dimensi:

Nama Metrik	Deskripsi	Unit/Statistik
`RequestDownloadFailures`	Ketika kegagalan inferensi terjadi karena masalah saat mengunduh permintaan dari Amazon S3.	Unit: Count (Jumlah) Statistik yang valid: Sum
`ResponseUploadFailures`	Ketika kegagalan inferensi terjadi karena masalah saat mengunggah respons ke Amazon S3.	Unit: Count (Jumlah) Statistik yang valid: Sum
`NotificationFailures`	Saat terjadi masalah, publikasi notifikasi.	Unit: Count (Jumlah) Statistik yang valid: Sum
`RequestDownloadLatency`	Total waktu untuk mengunduh payload permintaan.	Satuan: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`ResponseUploadLatency`	Total waktu untuk mengunggah payload respons.	Satuan: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`ExpiredRequests`	Jumlah permintaan dalam antrian yang gagal karena mencapai permintaan TTL yang ditentukan.	Unit: Count (Jumlah) Statistik yang valid: Sum
`InvocationFailures`	Jika doa gagal karena alasan apa pun.	Unit: Count (Jumlah) Statistik yang valid: Sum
`InvocationsProcessed`	Jumlah pemanggilan asinkron yang diproses oleh titik akhir.	Unit: Count (Jumlah) Statistik yang valid: Sum

Nama Metrik	Deskripsi	Unit/Statistik
`TimeInBacklog`	Total waktu permintaan antri sebelum diproses. Ini tidak termasuk waktu pemrosesan aktual (yaitu waktu pengunduhan, waktu unggah, latensi model).	Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`TotalProcessingTime`	Waktu permintaan inferensi diterima oleh SageMaker pada saat permintaan selesai diproses. Ini termasuk waktu dalam backlog dan waktu untuk mengunggah dan mengirim pemberitahuan respons, jika ada.	Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel

Amazon SageMaker Inferensi Asinkron juga mencakup metrik tingkat host. Untuk informasi tentang metrik tingkat host, lihatSageMaker Lowongan Kerja dan Metrik Titik Akhir.

Log

SelainLog kontainer modelyang dipublikasikan ke Amazon CloudWatch di akun Anda, Anda juga mendapatkan log platform baru untuk melacak dan men-debug permintaan inferensi.

Log baru diterbitkan di bawah Endpoint Log Group:

```
/aws/sagemaker/Endpoints/[EndpointName]
```

Nama log stream terdiri dari:

```
[production-variant-name]/[instance-id]/data-log.
```

Baris log berisi ID inferensi permintaan sehingga kesalahan dapat dengan mudah dipetakan ke permintaan tertentu.

Hasil prediksi

Ada beberapa cara yang Anda dapat lakukan untuk memeriksa hasil prediksi dari titik akhir asinkron Anda. Beberapa opsi adalah:

1. Topik Amazon SNS.
2. Periksa output di bucket Amazon S3.

Topik Amazon SNS

Amazon SNS adalah layanan notifikasi untuk aplikasi berorientasi pesan, dengan beberapa pelanggan yang meminta dan menerima pemberitahuan “push” pesan penting waktu melalui pilihan protokol transport, termasuk HTTP, Amazon SQS, dan email. Amazon SageMaker Asynchronous Inference memposting pemberitahuan saat Anda membuat titik akhir dengan`CreateEndpointConfig`dan tentukan topik Amazon SNS.

> **Note**
>
> Untuk menerima pemberitahuan Amazon SNS, peran IAM Anda harus memiliki`sns:Publish`izin. LihatPrasyaratuntuk informasi tentang persyaratan yang harus Anda penuhi untuk menggunakan Inferensi Asynchronous.

Untuk menggunakan Amazon SNS untuk memeriksa hasil prediksi dari titik akhir asinkron Anda, Anda harus terlebih dahulu membuat topik, berlangganan topik, mengonfirmasi langganan Anda ke topik, dan perhatikan Nama Sumber Daya Amazon (ARN) dari topik tersebut. Untuk informasi terperinci tentang cara membuat, berlangganan, dan menemukan Amazon ARN dari topik Amazon SNS, lihatMengonfigurasi Amazon SNS.

Menyediakan topik Amazon SNS ARN (s) di`AsyncInferenceConfig`bidang ketika Anda membuat konfigurasi endpoint dengan`CreateEndpointConfig`. Anda dapat menentukan Amazon SNS`ErrorTopic`dan sebuah`SuccessTopic`.

```
import boto3

sagemaker_client = boto3.client('sagemaker', region_name=<aws_region>)

sagemaker_client.create_endpoint_config(
```

```
    EndpointConfigName=<endpoint_config_name>, # You specify this name in a
 CreateEndpoint request.
    # List of ProductionVariant objects, one for each model that you want to host at
 this endpoint.
    ProductionVariants=[
        {
            "VariantName": "variant1", # The name of the production variant.
            "ModelName": "model_name",
            "InstanceType": "ml.m5.xlarge", # Specify the compute instance type.
            "InitialInstanceCount": 1 # Number of instances to launch initially.
        }
    ],
    AsyncInferenceConfig={
        "OutputConfig": {
            # Location to upload response outputs when no location is provided in the
 request.
            "S3OutputPath": "s3://<bucket>/<output_directory>"
            "NotificationConfig": {
                "SuccessTopic": "arn:aws:sns:aws-region:account-id:topic-name",
                "ErrorTopic": "arn:aws:sns:aws-region:account-id:topic-name",
            }
        }
    }
)
```

Setelah membuat endpoint dan memanggilnya, Anda menerima pemberitahuan dari topik Amazon SNS Anda. Misalnya, jika Anda berlangganan untuk menerima pemberitahuan email dari topik Anda, Anda menerima pemberitahuan email setiap kali Anda memanggil titik akhir Anda. Contoh berikut menunjukkan konten JSON notifikasi email pemanggilan yang berhasil.

```
{
   "awsRegion":"us-east-1",
   "eventTime":"2022-01-25T22:46:00.608Z",
   "receivedTime":"2022-01-25T22:46:00.455Z",
   "invocationStatus":"Completed",
   "requestParameters":{
      "contentType":"text/csv",
      "endpointName":"<example-endpoint>",
      "inputLocation":"s3://<bucket>/<input-directory>/input-data.csv"
   },
   "responseParameters":{
      "contentType":"text/csv; charset=utf-8",
      "outputLocation":"s3://<bucket>/<output_directory>/prediction.out"
```

```
    },
    "inferenceId":"11111111-2222-3333-4444-555555555555",
    "eventVersion":"1.0",
    "eventSource":"aws:sagemaker",
    "eventName":"InferenceResult"
}
```

Periksa Ember S3 Anda

Ketika Anda memanggil titik akhir dengan`InvokeEndpointAsync`, ia mengembalikan objek respon. Anda dapat menggunakan objek respons untuk mendapatkan URI Amazon S3 tempat output Anda disimpan. Dengan lokasi output, Anda dapat menggunakan kelas sesi SageMaker Python SDK SageMaker untuk memeriksa secara terprogram pada output.

Berikut ini menyimpan kamus output dari`InvokeEndpointAsync`sebagai variabel bernama respon. Dengan variabel respon, Anda kemudian mendapatkan URI output Amazon S3 dan menyimpannya sebagai variabel string yang disebut`output_location`.

```
import uuid
import boto3

sagemaker_runtime = boto3.client("sagemaker-runtime", region_name=<aws_region>)

# Specify the S3 URI of the input. Here, a single SVM sample
input_location = "s3://bucket-name/test_point_0.libsvm"

response = sagemaker_runtime.invoke_endpoint_async(
    EndpointName='<endpoint-name>',
    InputLocation=input_location,
    InferenceId=str(uuid.uuid4()),
    ContentType="text/libsvm" #Specify the content type of your data
)

output_location = response['OutputLocation']
print(f"OutputLocation: {output_location}")
```

Untuk informasi tentang jenis konten yang didukung, lihatFormat Data Umum untuk Inferensi.

Dengan lokasi output Amazon S3, Anda dapat menggunakanKelas Sesi SageMaker Python SDK SageMakermembaca file Amazon S3. Contoh kode berikut menunjukkan cara membuat fungsi (`get_ouput`) yang berulang kali mencoba membaca file dari lokasi output Amazon S3:

```
import sagemaker
import urllib, time
from botocore.exceptions import ClientError

sagemaker_session = sagemaker.session.Session()

def get_output(output_location):
    output_url = urllib.parse.urlparse(output_location)
    bucket = output_url.netloc
    key = output_url.path[1:]
    while True:
        try:
            return sagemaker_session.read_s3_file(
                                    bucket=output_url.netloc,
                                    key_prefix=output_url.path[1:])
        except ClientError as e:
            if e.response['Error']['Code'] == 'NoSuchKey':
                print("waiting for output...")
                time.sleep(2)
                continue
            raise

output = get_output(output_location)
print(f"Output: {output}")
```

Skala otomatis titik akhir asinkron

Amazon SageMaker mendukung penskalaan otomatis (penskalaan otomatis) titik akhir asinkron Anda. Autoscaling secara dinamis menyesuaikan jumlah instance yang disediakan untuk model sebagai respons terhadap perubahan beban kerja Anda. Tidak seperti model host lainnya yang SageMaker didukung Amazon, dengan Inferensi Asynchronous Anda juga dapat menurunkan skala instans titik akhir asinkron Anda menjadi nol. Permintaan yang diterima saat ada nol instans akan diantrekan untuk diproses setelah titik akhir bertambah.

Untuk autoscale endpoint asynchronous Anda, Anda harus minimal:

- Daftarkan model yang digunakan (varian produksi).
- Menetapkan kebijakan penskalaan.
- Terapkan kebijakan penskalaan otomatis.

Sebelum Anda dapat menggunakan penskalaan otomatis, Anda harus sudah menerapkan model ke SageMaker titik akhir. Model yang digunakan disebut sebagai varian produksi. Lihat Menerapkan Model ke SageMaker Layanan Hosting untuk informasi selengkapnya tentang menerapkan model ke titik akhir. Untuk menentukan metrik dan nilai target untuk kebijakan penskalaan, Anda mengonfigurasi kebijakan penskalaan. Untuk informasi tentang cara menentukan kebijakan penskalaan, lihat Menentukan kebijakan penskalaan. Setelah mendaftarkan model Anda dan menentukan kebijakan penskalaan, terapkan kebijakan penskalaan, terapkan model yang terdaftar. Untuk informasi tentang cara menerapkan kebijakan penskalaan, lihat Menerapkan kebijakan penskalaan.

Untuk informasi selengkapnya tentang cara menentukan kebijakan penskalaan tambahan opsional yang meningkatkan titik akhir Anda setelah menerima permintaan setelah titik akhir Anda diperkecil menjadi nol, lihatOpsional: Tentukan kebijakan penskalaan yang meningkatkan skala dari nol untuk permintaan baru. Jika Anda tidak menentukan kebijakan opsional ini, maka titik akhir Anda hanya memulai penskalaan dari nol setelah jumlah permintaan backlog melebihi nilai pelacakan target.

Untuk detail tentang prasyarat dan komponen lain yang digunakan dengan penskalaan otomatis, lihat bagian Prasyarat dalam dokumentasi SageMaker penskalaan otomatis.

> **Note**
>
> Jika Anda melampirkan beberapa kebijakan penskalaan ke grup penskalaan otomatis yang sama, Anda mungkin memiliki konflik penskalaan. Jika konflik terjadi, Amazon EC2 Auto Scaling memilih kebijakan yang menyediakan kapasitas terbesar untuk mendirikan dan menurunkan skala. Untuk informasi selengkapnya tentang perilaku ini, lihat Beberapa kebijakan penskalaan dinamis dalam dokumentasi Auto Scaling Amazon EC2.

Menetapkan kebijakan penskalaan

Untuk menentukan metrik dan nilai target untuk kebijakan penskalaan, Anda mengkonfigurasi kebijakan penskalaan, Anda mengkonfigurasi kebijakan penskalaan. Menetapkan kebijakan penskalaan sebagai blok JSON dalam file teks. Anda menggunakan file teks tersebut saat memicu AWS CLI atau Application Auto Scaling API. Untuk informasi selengkapnya tentang sintaks konfigurasi kebijakan, lihat `TargetTrackingScalingPolicyConfiguration`dalam Referensi Application Auto Scaling API.

Untuk titik akhir asinkron, Anda SageMaker sangat menyarankan agar Anda membuat konfigurasi titik akhir asinkron, Anda sangat menyarankan agar Anda membuat konfigurasi

kebijakan untuk menskalaan pelacakan target untuk varian. Dalam contoh konfigurasi ini, kita menggunakan metrik kustom`CustomizedMetricSpecification`, yang disebut`ApproximateBacklogSizePerInstance`.

```
TargetTrackingScalingPolicyConfiguration={
        'TargetValue': 5.0, # The target value for the metric. Here the metric is:
 ApproximateBacklogSizePerInstance
        'CustomizedMetricSpecification': {
            'MetricName': 'ApproximateBacklogSizePerInstance',
            'Namespace': 'AWS/SageMaker',
            'Dimensions': [
                {'Name': 'EndpointName', 'Value': <endpoint_name> }
            ],
            'Statistic': 'Average',
        }
    }
```

Tentukan kebijakan penskalaan yang menskalakan ke nol

Berikut ini menunjukkan kepada Anda bagaimana untuk mendefinisikan dan mendaftarkan varian endpoint Anda dengan autoscaling aplikasi menggunakanAWS SDK for Python (Boto3). Setelah mendefinisikan objek klien tingkat rendah yang mewakili autoscaling aplikasi dengan Boto3, kita menggunakan [RegisterScalableTarget](#)metode untuk mendaftarkan varian produksi. Kami mengatur`MinCapacity` ke 0 karena Inferensi Asinkron memungkinkan Anda untuk melakukan skala otomatis ke 0 ketika tidak ada permintaan untuk diproses.

```
# Common class representing application autoscaling for SageMaker
client = boto3.client('application-autoscaling')

# This is the format in which application autoscaling references the endpoint
resource_id='endpoint/' + <endpoint_name> + '/variant/' + <'variant1'>

# Define and register your endpoint variant
response = client.register_scalable_target(
    ServiceNamespace='sagemaker',
    ResourceId=resource_id,
    ScalableDimension='sagemaker:variant:DesiredInstanceCount', # The number of EC2
 instances for your Amazon SageMaker model endpoint variant.
    MinCapacity=0,
    MaxCapacity=5
```

```
)
```

Untuk penjelasan rinci tentang Application Autoscaling API, lihat dokumentasi Application Scaling Boto3.

Opsional: Tentukan kebijakan penskalaan yang meningkatkan skala dari nol untuk permintaan baru

Anda mungkin memiliki kasus penggunaan di mana Anda memiliki permintaan sporadis atau periode dengan jumlah permintaan yang rendah. Jika titik akhir Anda telah diperkecil ke nol instans selama periode ini, maka titik akhir Anda tidak akan ditingkatkan lagi hingga jumlah permintaan dalam antrean melebihi target yang ditentukan dalam kebijakan penskalaan Anda. Hal ini dapat mengakibatkan waktu tunggu yang lama untuk permintaan dalam antrian. Bagian berikut menunjukkan cara membuat kebijakan penskalaan tambahan yang meningkatkan titik akhir dari nol instans setelah menerima permintaan baru dalam antrean. Endpoint Anda akan dapat merespons permintaan baru lebih cepat daripada menunggu ukuran antrian melebihi target.

Untuk membuat kebijakan penskalaan untuk titik akhir Anda yang meningkatkan skala dari nol instans, lakukan hal berikut:

1. Buat kebijakan penskalaan yang mendefinisikan perilaku yang diinginkan, yaitu untuk meningkatkan titik akhir Anda saat berada di nol instance tetapi memiliki permintaan dalam antrian. Berikut ini menunjukkan cara menentukan kebijakan penskalaan yang disebut`HasBacklogWithoutCapacity-ScalingPolicy` menggunakanAWS SDK for Python (Boto3). Ketika antrean lebih besar dari nol dan jumlah instans saat ini untuk titik akhir Anda juga nol, kebijakan akan menskalakan titik akhir Anda. Dalam semua kasus lainnya, kebijakan tidak memengaruhi penskalaan untuk titik akhir Anda.

   ```
   response = client.put_scaling_policy(
       PolicyName="HasBacklogWithoutCapacity-ScalingPolicy",
       ServiceNamespace="sagemaker",  # The namespace of the service that provides the
    resource.
       ResourceId=resource_id,  # Endpoint name
       ScalableDimension="sagemaker:variant:DesiredInstanceCount",  # SageMaker
    supports only Instance Count
       PolicyType="StepScaling",  # 'StepScaling' or 'TargetTrackingScaling'
       StepScalingPolicyConfiguration={
           "AdjustmentType": "ChangeInCapacity", # Specifies whether the
    ScalingAdjustment value in the StepAdjustment property is an absolute number or a
    percentage of the current capacity.
   ```

```
        "MetricAggregationType": "Average", # The aggregation type for the
 CloudWatch metrics.
        "Cooldown": 300, # The amount of time, in seconds, to wait for a previous
 scaling activity to take effect.
        "StepAdjustments": # A set of adjustments that enable you to scale based on
 the size of the alarm breach.
        [
            {
              "MetricIntervalLowerBound": 0,
              "ScalingAdjustment": 1
            }
          ]
    },
)
```

2. Buat CloudWatch alarm dengan metrik khususHasBacklogWithoutCapacity. Saat dipicu, alarm memulai kebijakan penskalaan yang ditentukan sebelumnya. Untuk informasi selengkapnya tentangHasBacklogWithoutCapacity metrik, lihatMetrik Titik Akhir Inferensi Asinkron.

```
response = cw_client.put_metric_alarm(
    AlarmName=step_scaling_policy_alarm_name,
    MetricName='HasBacklogWithoutCapacity',
    Namespace='AWS/SageMaker',
    Statistic='Average',
    EvaluationPeriods= 2,
    DatapointsToAlarm= 2,
    Threshold= 1,
    ComparisonOperator='GreaterThanOrEqualToThreshold',
    TreatMissingData='missing',
    Dimensions=[
        { 'Name':'EndpointName', 'Value':endpoint_name },
    ],
    Period= 60,
    AlarmActions=[step_scaling_policy_arn]
)
```

Anda sekarang harus memiliki kebijakan penskalaan dan CloudWatch alarm yang meningkatkan titik akhir Anda dari nol instans setiap kali antrean Anda memiliki permintaan yang tertunda.

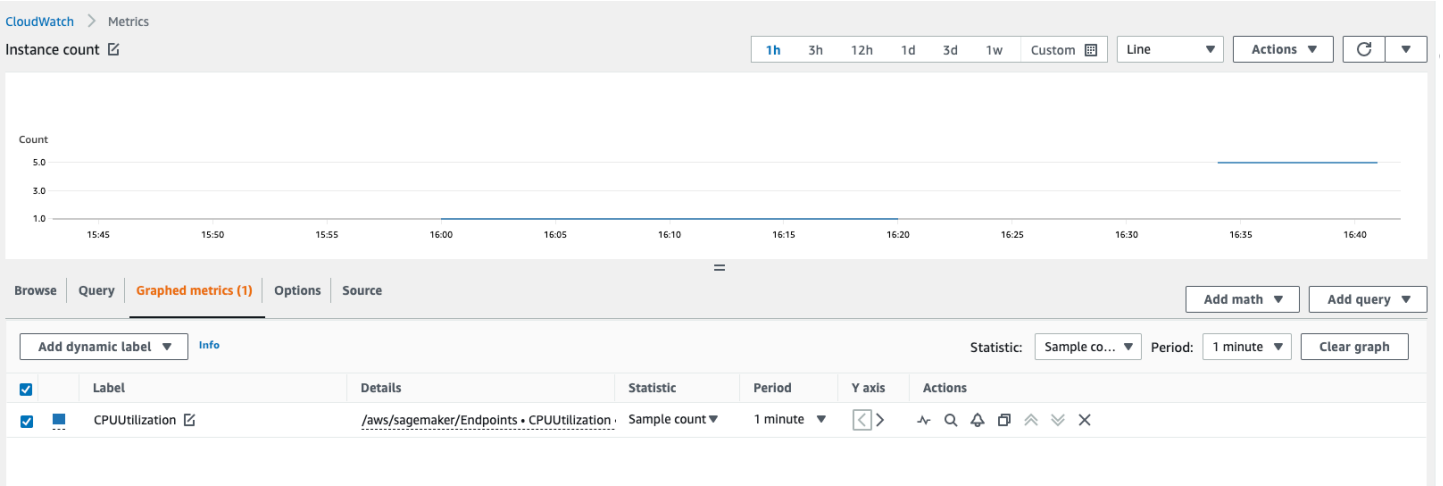
Memecahkan masalah

FAQ berikut dapat membantu Anda memecahkan masalah dengan titik akhir Inferensi Asinkron Amazon SageMaker Anda.

T: Saya telah mengaktifkan penskalaan otomatis. Bagaimana saya bisa menemukan jumlah instance di belakang titik akhir pada titik tertentu?

Anda dapat menggunakan metode berikut untuk menemukan jumlah instance di belakang titik akhir Anda:

- Anda dapat menggunakan SageMaker DescribeEndpointAPI untuk menjelaskan jumlah instance di belakang titik akhir pada titik waktu tertentu.
- Anda bisa mendapatkan jumlah instans dengan melihat CloudWatch metrik Amazon Anda. Lihat metrik untuk instans titik akhir Anda, seperti `CPUUtilization` atau `MemoryUtilization` dan periksa statistik jumlah sampel untuk periode 1 menit. Hitungannya harus sama dengan jumlah instance aktif. Tangkapan layar berikut menunjukkan grafik `CPUUtilization` metrik di CloudWatch konsol, di mana Statistik diatur ke`Sample count`, Periode diatur ke`1 minute`, dan jumlah yang dihasilkan adalah 5.



T: Apa variabel lingkungan umum yang dapat disetel untuk SageMaker kontainer?

Tabel berikut menguraikan variabel lingkungan yang dapat disetel umum untuk SageMaker kontainer menurut jenis kerangka kerja.

TensorFlow

Variabel lingkungan	Deskripsi
SAGEMAKER_TFS_INSTANCE_COUNT	Untuk model TensorFlow berbasis, tensorflow_model_server biner adalah bagian operasional yang bertanggung jawab untuk memuat model dalam memori, menjalankan input terhadap grafik model, dan menurunkan output. Biasanya, satu contoh biner ini diluncurkan untuk melayani model di titik akhir. Biner ini secara internal multi-threaded dan memunculkan beberapa utas untuk menanggapi permintaan inferensi. Dalam kasus tertentu, jika Anda mengamati bahwa CPU digunakan dengan baik (lebih dari 30% digunakan) tetapi memori kurang dimanfaatkan (kurang dari 10% penggunaan), meningkatkan parameter ini dapat membantu. Meningkatkan jumlah yang tensorflow_model_servers tersedia untuk melayani biasanya meningkatkan throughput dari titik akhir.
SAGEMAKER_TFS_FRACTIONAL_GPU_MEM_MARGIN	Parameter ini mengatur fraksi memori GPU yang tersedia untuk menginisialisasi CUDA/cuDNN dan pustaka GPU lainnya. 0.2berarti 20% dari memori GPU yang tersedia dicadangkan untuk menginisialisasi CUDA/cuDNN dan pustaka GPU lainnya, dan 80% dari memori GPU yang tersedia dialokasikan secara merata di seluruh proses TF. Memori GPU sudah dialokasikan sebelumnya kecuali allow_growth opsi diaktifkan.
SAGEMAKER_TFS_INTER_OP_PARALLELISM	Ini mengikat kembali ke inter_op_parallelism_threads variabel. Variabel ini menentukan jumlah utas yang digunakan oleh operasi non-pemblokiran independen.

Variabel lingkungan	Deskripsi
	`0`berarti bahwa sistem memilih nomor yang sesuai.
`SAGEMAKER_TFS_INTRA_OP_PARALLELISM`	Ini mengikat kembali ke `intra_op_parallelism_threads` variabel. Ini menentukan jumlah utas yang dapat digunakan untuk operasi tertentu seperti perkalian matriks dan pengurangan untuk percepatan. Nilai `0` berarti bahwa sistem memilih nomor yang sesuai.
`SAGEMAKER_GUNICORN_WORKERS`	Ini mengatur jumlah proses pekerja yang diminta Gunicorn untuk menelurkan untuk menangani permintaan. Nilai ini digunakan dalam kombinasi dengan parameter lain untuk mendapatkan satu set yang memaksimalkan throughput inferensi. Selain itu, `SAGEMAKER_GUNICORN_WORKER_CLASS` mengatur jenis pekerja yang melahirkan, biasanya atau. `async` `gevent`
`SAGEMAKER_GUNICORN_WORKER_CLASS`	Ini mengatur jumlah proses pekerja yang diminta Gunicorn untuk menelurkan untuk menangani permintaan. Nilai ini digunakan dalam kombinasi dengan parameter lain untuk mendapatkan satu set yang memaksimalkan throughput inferensi. Selain itu, `SAGEMAKER_GUNICORN_WORKER_CLASS` mengatur jenis pekerja yang melahirkan, biasanya atau. `async` `gevent`

Variabel lingkungan	Deskripsi
`OMP_NUM_THREADS`	Python secara internal menggunakan OpenMP untuk mengimplementasikan multithreading dalam proses. Biasanya, thread yang setara dengan jumlah core CPU muncul. Tetapi ketika diimplementasikan di atas Simultaneous Multi Threading (SMT), seperti Intel HypeThreading, proses tertentu mungkin kelebihan langganan inti tertentu dengan menelurkan utas dua kali lebih banyak daripada jumlah inti CPU yang sebenarnya. Dalam kasus tertentu, biner Python mungkin akan menghasilkan hingga empat kali lebih banyak utas daripada inti prosesor yang tersedia. Oleh karena itu, pengaturan ideal untuk parameter ini, jika Anda memiliki kelebihan langganan core yang tersedia menggunakan thread pekerja, adalah1, atau setengah jumlah core CPU pada CPU dengan SMT dihidupkan.

`TF_DISABLE_MKL`
 	
`TF_DISABLE_POOL_ALLOCATOR`
	Dalam beberapa kasus, mematikan MKL dapat mempercepat inferensi jika `TF_DISABLE_MKL` dan `TF_DISABLE_POOL_ALLOCATOR` disetel ke. 1

PyTorch

Variabel lingkungan	Deskripsi
`SAGEMAKER_TS_MAX_BATCH_DELAY`	Ini adalah waktu tunda batch maksimum yang TorchServe menunggu untuk diterima.
`SAGEMAKER_TS_BATCH_SIZE`	Jika TorchServe tidak menerima jumlah permintaan yang ditentukan `batch_size`

Variabel lingkungan	Deskripsi
	sebelum timer habis, itu akan mengirimkan permintaan yang diterima ke handler model.
SAGEMAKER_TS_MIN_WORKERS	Jumlah minimum pekerja yang TorchServe diizinkan untuk menurunkan skala.
SAGEMAKER_TS_MAX_WORKERS	Jumlah maksimum pekerja yang TorchServe diizinkan untuk ditingkatkan.
SAGEMAKER_TS_RESPONSE_TIMEOUT	Waktu tunda, setelah itu waktu inferensi habis tanpa adanya respons.
SAGEMAKER_TS_MAX_REQUEST_SIZE	Ukuran muatan maksimum untuk TorchServe.
SAGEMAKER_TS_MAX_RESPONSE_SIZE	Ukuran respons maksimum untuk TorchServe.

Server Multi Model (MMS)

Variabel lingkungan	Deskripsi
job_queue_size	Parameter ini berguna untuk disetel ketika Anda memiliki skenario di mana jenis payload permintaan inferensi besar, dan karena ukuran payload yang lebih besar, Anda mungkin memiliki konsumsi memori heap yang lebih tinggi dari JVM di mana antrian ini sedang dipertahankan. Idealnya Anda mungkin ingin menjaga persyaratan memori heap JVM lebih rendah dan memungkinkan pekerja Python membagikan lebih banyak memori untuk penyajian model yang sebenarnya. JVM hanya untuk menerima permintaan HTTP, mengantri, dan mengirimkannya ke pekerja berbasis Python untuk inferensi. Jika Anda meningkatkan job_queue_size , Anda mungkin akhirnya meningkatkan konsumsi

Variabel lingkungan	Deskripsi
	memori heap JVM dan akhirnya mengambil memori dari host yang bisa digunakan oleh pekerja Python. Karena itu, berhati-hatilah saat menyetel parameter ini juga.
`default_workers_per_model`	Parameter ini untuk penyajian model backend dan mungkin berharga untuk disetel karena ini adalah komponen penting dari keseluruhan penyajian model, berdasarkan proses Python menelurkan utas untuk setiap Model. Jika komponen ini lebih lambat (atau tidak disetel dengan benar), penyetelan front-end mungkin tidak efektif.

T: Bagaimana cara memastikan penampung saya mendukung Inferensi Asinkron?

Anda dapat menggunakan wadah yang sama untuk Inferensi Asinkron yang Anda lakukan untuk Inferensi Waktu Nyata atau Transformasi Batch. Anda harus mengonfirmasi bahwa batas waktu tunggu dan ukuran muatan pada kontainer Anda diatur untuk menangani muatan yang lebih besar dan batas waktu yang lebih lama.

T: Apa batas khusus untuk Inferensi Asinkron, dan dapatkah mereka disesuaikan?

Lihat batas berikut untuk Inferensi Asinkron:

- Batas ukuran muatan: 1 GB
- Batas waktu tunggu: Permintaan dapat memakan waktu hingga 60 menit.
- Pesan antrian TimeToLive (TTL): 6 jam
- Jumlah pesan yang dapat dimasukkan ke dalam Amazon SQS: Tidak terbatas. Namun, ada kuota 120.000 untuk jumlah pesan dalam penerbangan untuk antrian standar, dan 20.000 untuk antrian FIFO.

T: Metrik apa yang terbaik untuk didefinisikan untuk penskalaan otomatis pada Inferensi Asinkron? Dapatkah saya memiliki beberapa kebijakan penskalaan?

Secara umum, dengan Asynchronous Inference, Anda dapat menskalakan berdasarkan pemanggilan atau instance. Untuk metrik pemanggilan, ada baiknya untuk melihat metrik Anda`ApproximateBacklogSize`, yang merupakan metrik yang menentukan jumlah item dalam antrian Anda yang belum diproses. Anda dapat menggunakan metrik ini atau `InvocationsPerInstance` metrik Anda untuk memahami TPS apa yang mungkin membuat Anda terhambat. Pada tingkat instans, periksa jenis instans Anda dan pemanfaatan CPU/GPU-nya untuk menentukan kapan harus menskalakan. Jika instance tunggal di atas kapasitas 60-70%, ini sering merupakan pertanda baik bahwa Anda menjenuhkan perangkat keras Anda.

Kami tidak menyarankan memiliki beberapa kebijakan penskalaan, karena ini dapat bertentangan dan menyebabkan kebingungan di tingkat perangkat keras, menyebabkan penundaan saat penskalaan keluar.

T: Mengapa titik akhir asinkron saya mengakhiri instance sebagai **`Unhealthy`** dan permintaan pembaruan dari penskalaan otomatis gagal?

Periksa apakah kontainer Anda dapat menangani permintaan ping dan memanggil secara bersamaan. SageMaker permintaan pemanggilan memakan waktu sekitar 3 menit, dan dalam durasi ini, biasanya beberapa permintaan ping akhirnya gagal karena batas waktu yang SageMaker menyebabkan penampung Anda terdeteksi sebagai. Un`healthy`

T: Dapat **`MaxConcurrentInvocationsPerInstance`** bekerja untuk wadah model BYOC saya dengan pengaturan ningx/gunicorn/flask?

Ya. `MaxConcurrentInvocationsPerInstance`adalah fitur titik akhir asinkron. Ini tidak tergantung pada implementasi penampung khusus. `MaxConcurrentInvocationsPerInstance`mengontrol tingkat di mana permintaan pemanggilan dikirim ke wadah pelanggan. Jika nilai ini ditetapkan sebagai`1`, maka hanya 1 permintaan yang dikirim ke wadah sekaligus, tidak peduli berapa banyak pekerja yang ada di wadah pelanggan.

T: Bagaimana cara men-debug kesalahan server model (500) pada titik akhir asinkron saya?

Kesalahan berarti bahwa wadah pelanggan mengembalikan kesalahan. SageMaker tidak mengontrol perilaku kontainer pelanggan. SageMaker cukup mengembalikan respons dari `ModelContainer` dan tidak mencoba lagi. Jika mau, Anda dapat mengonfigurasi pemanggilan untuk mencoba lagi pada kegagalan. Kami menyarankan Anda mengaktifkan pencatatan kontainer dan memeriksa log penampung Anda untuk menemukan akar penyebab kesalahan 500 dari model Anda. Periksa

`MemoryUtilization` metrik yang sesuai `CPUUtilization` dan pada titik kegagalan juga. Anda juga dapat mengonfigurasi S3 FailurePath ke respons model di Amazon SNS sebagai bagian dari Pemberitahuan Kesalahan Async untuk menyelidiki kegagalan.

T: Bagaimana saya bisa tahu jika **`MaxConcurrentInvocationsPerInstance=1`** berlaku? Apakah ada metrik yang bisa saya periksa?

Anda dapat memeriksa metrik`InvocationsProcesssed`, yang seharusnya sejajar dengan jumlah pemanggilan yang Anda harapkan akan diproses dalam satu menit berdasarkan konkurensi tunggal.

T: Bagaimana saya bisa melacak keberhasilan dan kegagalan permintaan doa saya? Apa praktik terbaik?

Praktik terbaik adalah mengaktifkan Amazon SNS, yang merupakan layanan notifikasi untuk aplikasi berorientasi pesan, dengan beberapa pelanggan yang meminta dan menerima pemberitahuan “push” pesan kritis waktu dari pilihan protokol transportasi, termasuk HTTP, Amazon SQS, dan email. Inferensi asinkron memposting pemberitahuan saat Anda membuat titik akhir dengan `CreateEndpointConfig` dan menentukan topik Amazon SNS.

Untuk menggunakan Amazon SNS untuk memeriksa hasil prediksi dari titik akhir asinkron Anda, Anda harus terlebih dahulu membuat topik, berlangganan topik, mengonfirmasi langganan Anda ke topik, dan mencatat Nama Sumber Daya Amazon (ARN) dari topik tersebut. Untuk informasi terperinci tentang cara membuat, berlangganan, dan menemukan Amazon ARN dari topik Amazon SNS, lihat Mengonfigurasi Amazon SNS di Panduan Pengembang Amazon SNS. Untuk informasi selengkapnya tentang cara menggunakan Amazon SNS dengan Inferensi Asinkron, lihat Periksa hasil prediksi.

T: Dapatkah saya menentukan kebijakan penskalaan yang meningkatkan skala dari nol instans setelah menerima permintaan baru?

Ya. Inferensi Asinkron menyediakan mekanisme untuk menurunkan skala ke nol instance ketika tidak ada permintaan. Jika titik akhir Anda telah diperkecil menjadi nol instans selama periode ini, maka titik akhir Anda tidak akan ditingkatkan lagi hingga jumlah permintaan dalam antrian melebihi target yang ditentukan dalam kebijakan penskalaan Anda. Hal ini dapat mengakibatkan waktu tunggu yang lama untuk permintaan dalam antrian. Dalam kasus seperti itu, jika Anda ingin meningkatkan dari nol instance untuk permintaan baru yang kurang dari target antrian yang ditentukan, Anda dapat menggunakan kebijakan penskalaan tambahan yang disebut. `HasBacklogWithoutCapacity` Untuk informasi selengkapnya tentang cara mendefinisikan kebijakan penskalaan ini, lihat Menskalakan otomatis titik akhir asinkron.

T: Saya mendapatkan kesalahan bahwa jenis instance tidak didukung untuk Inferensi Asinkron. Apa saja tipe instance yang didukung Asynchronous Inference?

Untuk daftar lengkap instance yang didukung oleh Inferensi Asinkron per wilayah, lihat harga. SageMaker Periksa apakah instans yang diperlukan tersedia di wilayah Anda sebelum melanjutkan.

Gunakan Batch Transform

Gunakan batch transform saat Anda perlu melakukan hal berikut:

- Set data pra-proses untuk menghilangkan noise atau bias yang mengganggu pelatihan atau inferensi dari kumpulan data Anda.
- Dapatkan kesimpulan dari kumpulan data besar.
- Jalankan inferensi saat Anda tidak memerlukan titik akhir yang persisten.
- Mengaitkan catatan masukan dengan kesimpulan untuk membantu interpretasi hasil.

Untuk memfilter data input sebelum melakukan inferensi atau untuk mengaitkan catatan masukan dengan kesimpulan tentang catatan tersebut, lihatHasil Prediksi Associate dengan Input Records. Misalnya, Anda dapat memfilter data input untuk menyediakan konteks untuk membuat dan menafsirkan laporan tentang data keluaran.

Topik

- Gunakan Batch Transform untuk Mendapatkan Inferensi dari Set Data Besar
- Mempercepat Pekerjaan Transformasi Batch
- Gunakan Batch Transform untuk Menguji Varian Produksi
- Notebook Sampel Transformasi Batch
- Hasil Prediksi Associate dengan Input Records
- Penyimpanan di Batch Transform
- Pemecahan Masalah

Gunakan Batch Transform untuk Mendapatkan Inferensi dari Set Data Besar

Batch transform secara otomatis mengelola pemrosesan dataset besar dalam batas-batas parameter yang ditentukan. Misalnya, Anda memiliki file dataset,`input1.csv`, disimpan dalam ember S3. Isi dari file input mungkin terlihat seperti contoh berikut.

```
Record1-Attribute1, Record1-Attribute2, Record1-Attribute3, ..., Record1-AttributeM
Record2-Attribute1, Record2-Attribute2, Record2-Attribute3, ..., Record2-AttributeM
Record3-Attribute1, Record3-Attribute2, Record3-Attribute3, ..., Record3-AttributeM
...
RecordN-Attribute1, RecordN-Attribute2, RecordN-Attribute3, ..., RecordN-AttributeM
```

Ketika pekerjaan transformasi batch dimulai,SageMakermenginisialisasi instance komputasi dan mendistribusikan inferensi atau beban kerja preprocessing di antara mereka. Batch Transform mempartisi objek Amazon S3 dalam input dengan kunci dan memetakan objek Amazon S3 ke instans. Bila Anda memiliki beberapa file, satu contoh mungkin memproses`input1.csv`, dan contoh lain mungkin memproses file bernama`input2.csv`. Jika Anda memiliki satu file input tetapi menginisialisasi beberapa instance komputasi, hanya satu instance yang memproses file input dan instance lainnya tidak aktif.

Anda juga dapat membagi file input menjadi batch mini. Misalnya, Anda dapat membuat batch mini dari`input1.csv`dengan memasukkan hanya dua catatan.

```
Record3-Attribute1, Record3-Attribute2, Record3-Attribute3, ..., Record3-AttributeM
Record4-Attribute1, Record4-Attribute2, Record4-Attribute3, ..., Record4-AttributeM
```

> **Note**
>
> SageMakermemproses setiap file input secara terpisah. Itu tidak menggabungkan batch mini dari file input yang berbeda untuk mematuhi`MaxPayloadInMB` batas.

Untuk membagi file input menjadi batch mini saat Anda membuat pekerjaan transformasi batch, setel`SplitType`nilai parameter untuk`Line`. Jika`SplitType`diatur ke`None`atau jika file input

tidak dapat dibagi menjadi batch mini,SageMakermenggunakan seluruh file input dalam satu permintaan. Perhatikan bahwa Batch Transform tidak mendukung masukan berformat CSV yang berisi karakter baris baru yang disematkan. Anda dapat mengontrol ukuran batch mini dengan menggunakan`BatchStrategy`dan`MaxPayloadInMB`parameter.`MaxPayloadInMB`tidak boleh lebih besar dari 100 MB. Jika Anda menentukan opsional`MaxConcurrentTransforms`parameter, maka nilai(`MaxConcurrentTransforms * MaxPayloadInMB`)juga tidak boleh melebihi 100 MB.

Jika pekerjaan transformasi batch berhasil memproses semua catatan dalam file input, itu menciptakan file output dengan nama yang sama dan`.out`ekstensi file. Untuk beberapa file input, seperti`input1.csv`dan`input2.csv`, file output diberi nama`input1.csv.out`dan`input2.csv.out`. Lowongan transformasi batch menyimpan file keluaran di lokasi yang ditentukan di Amazon S3, seperti`s3://awsexamplebucket/output/`.

Prediksi dalam file output tercantum dalam urutan yang sama dengan catatan yang sesuai dalam file input. File output`input1.csv.out`, berdasarkan file input yang ditunjukkan sebelumnya, akan terlihat seperti berikut ini.

```
Inference1-Attribute1, Inference1-Attribute2, Inference1-Attribute3, ..., Inference1-
AttributeM
Inference2-Attribute1, Inference2-Attribute2, Inference2-Attribute3, ..., Inference2-
AttributeM
Inference3-Attribute1, Inference3-Attribute2, Inference3-Attribute3, ..., Inference3-
AttributeM
...
InferenceN-Attribute1, InferenceN-Attribute2, InferenceN-Attribute3, ..., InferenceN-
AttributeM
```

Jika Anda mengatur`SplitType`kepada`Line`, Anda dapat mengatur`AssembleWith`parameter ke`Line`untuk menggabungkan catatan output dengan pembatas garis. Ini tidak mengubah jumlah file output. Jumlah file output sama dengan jumlah file input, dan menggunakan`AssembleWith`tidak menggabungkan file. Jika Anda tidak menentukan`AssembleWith`parameter, secara default catatan output digabungkan dalam format biner.

Ketika data input sangat besar dan ditransmisikan menggunakan HTTP chunked encoding, untuk mengalirkan data ke algoritma, mengatur`MaxPayloadInMB`kepada`0`. AmazonSageMakeralgoritma bawaan tidak mendukung fitur ini.

Untuk informasi tentang menggunakan API untuk membuat pekerjaan transformasi batch, lihat`CreateTransformJob`API. Untuk informasi selengkapnya tentang korelasi antara objek input

dan output transformasi batch, lihatOutputDataConfig. Untuk contoh cara menggunakan batch transform, lihat(Opsional) Buat Prediksi dengan Batch Transform.

Mempercepat Pekerjaan Transformasi Batch

Jika Anda menggunakanCreateTransformJobAPI, Anda dapat mengurangi waktu yang diperlukan untuk menyelesaikan pekerjaan transformasi batch dengan menggunakan nilai optimal untuk parameter sepertiMaxPayloadInMB,MaxConcurrentTransforms, atauBatchStrategy. Nilai ideal untukMaxConcurrentTransformssama dengan jumlah pekerja komputasi dalam pekerjaan batch transform. Jika Anda menggunakanSageMakerkonsol, Anda dapat menentukan nilai parameter optimal ini diKonfigurasi tambahanbagian dariBatch mengubah konfigurasi pekerjaanhalaman. SageMakersecara otomatis menemukan pengaturan parameter optimal untuk algoritma bawaan. Untuk algoritma kustom, memberikan nilai-nilai ini melaluieksekusi-parametertitik akhir.

Gunakan Batch Transform untuk Menguji Varian Produksi

Untuk menguji model yang berbeda atau berbagai pengaturan hyperparameter, buat pekerjaan transformasi terpisah untuk setiap varian model baru dan gunakan dataset validasi. Untuk setiap pekerjaan transformasi, tentukan nama model dan lokasi unik di Amazon S3 untuk file keluaran. Untuk menganalisis hasilnya, gunakanLog dan Metrik Pipa Inferensi.

Notebook Sampel Transformasi Batch

Untuk contoh notebook yang menggunakan batch transform dengan model analisis komponen utama (PCA) untuk mengurangi data dalam matriks tinjauan item pengguna, diikuti dengan penerapan pengelompokan spasial berbasis kepadatan aplikasi dengan algoritma noise (DBSCAN) ke film cluster, lihatTransformasi Batch dengan Cluster Film PCA dan DBSCAN. Untuk petunjuk tentang membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contohSageMaker, lihatInstans SageMaker Notebook Amazon. Setelah membuat dan membuka instance notebook, pilihSageMakerContohtab untuk melihat daftar semuaSageMakercontoh. Notebook contoh pemodelan topik yang menggunakan algoritma NTM terletak diFungsionalitas lanjutanbagian. Untuk membuka notebook, pilihGunakantab, lalu pilihBuat salinan.

Hasil Prediksi Associate dengan Input Records

Saat membuat prediksi pada kumpulan data besar, Anda dapat mengecualikan atribut yang tidak diperlukan untuk prediksi. Setelah prediksi dibuat, Anda dapat mengaitkan beberapa atribut yang dikecualikan dengan prediksi tersebut atau dengan data masukan lain dalam laporan Anda. Dengan

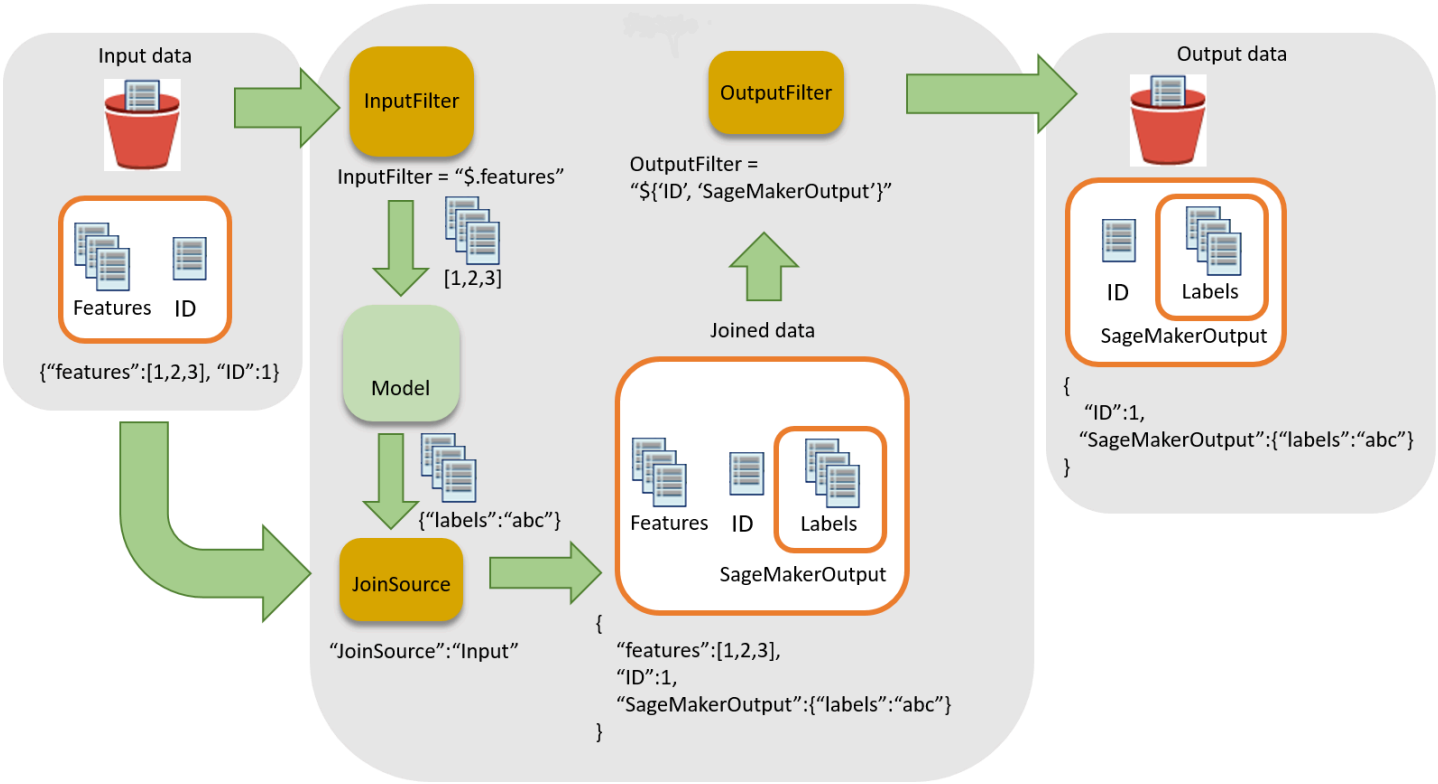
menggunakan batch transform untuk melakukan langkah-langkah pemrosesan data ini, Anda sering dapat menghilangkan preprocessing atau postprocessing tambahan. Anda dapat menggunakan file input dalam format JSON dan CSV saja.

Topik

- Alur Kerja untuk Mengaitkan Inferensi dengan Catatan Input
- Gunakan Pemrosesan Data dalam Pekerjaan Transformasi Batch
- Operator JSONPath yang didukung
- Contoh Batch Transform

Alur Kerja untuk Mengaitkan Inferensi dengan Catatan Input

Diagram berikut menunjukkan alur kerja untuk mengasosiasikan kesimpulan dengan catatan masukan.



Untuk mengaitkan kesimpulan dengan data input, ada tiga langkah utama:

1. Filter data input yang tidak diperlukan untuk inferensi sebelum meneruskan data input ke pekerjaan transformasi batch. Gunakan`InputFilter`parameter untuk menentukan atribut untuk digunakan sebagai masukan untuk model.

2. Kaitkan data input dengan hasil inferensi. Gunakan`JoinSource`parameter untuk menggabungkan data input dengan inferensi.
3. Filter data gabungan untuk mempertahankan masukan yang diperlukan untuk menyediakan konteks untuk menafsirkan prediksi dalam laporan. Gunakan`OutputFilter`untuk menyimpan bagian tertentu dari dataset bergabung dalam file output.

Gunakan Pemrosesan Data dalam Pekerjaan Transformasi Batch

Saat membuat pekerjaan transformasi batch dengan`CreateTransformJob`untuk memproses data:

1. Tentukan bagian input yang akan diteruskan ke model dengan`InputFilter`parameter dalam`DataProcessing`struktur data.
2. Bergabunglah dengan data input mentah dengan data yang ditransformasikan dengan`JoinSource`parameter.
3. Tentukan bagian mana dari input yang digabungkan dan data yang ditransformasikan dari pekerjaan transformasi batch untuk disertakan dalam file keluaran dengan`OutputFilter`parameter.
4. Pilih file berformat JSON atau CSV untuk masukan:
   - Untuk JSON- atau JSON Baris-diformat file masukan,SageMakerbaik menambahkan`SageMakerOutput`atribut ke file input atau membuat file output JSON baru dengan`SageMakerInput`dan`SageMakerOutput`atribut. Untuk informasi selengkapnya, lihat `DataProcessing`.
   - Untuk file input berformat CSV, data input yang digabungkan diikuti oleh data yang diubah dan hasilnya adalah file CSV.

Jika Anda menggunakan algoritma dengan`DataProcessing`struktur, itu harus mendukung format yang Anda pilih untukkedua-duanyafile input dan output. Misalnya, dengan`TransformOutput`bidang`CreateTransformJob`API, Anda harus mengatur kedua`ContentType`dan`Accept`parameter ke salah satu nilai berikut:`text/csv`,`application/json`, atau`application/jsonlines`. Sintaks untuk menentukan kolom dalam file CSV dan menentukan atribut dalam file JSON berbeda. Menggunakan sintaks yang salah menyebabkan kesalahan. Untuk informasi selengkapnya, lihat Contoh Batch Transform. Untuk informasi selengkapnya tentang format file input dan output untuk algoritme bawaan, lihatGunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih.

Pembatas rekaman untuk input dan output juga harus konsisten dengan input file yang Anda pilih. Yang`SplitType`parameter menunjukkan bagaimana untuk membagi catatan dalam dataset input. Yang`AssembleWith`parameter menunjukkan bagaimana untuk memasang kembali catatan untuk output. Jika Anda mengatur format input dan output ke`text/csv`, Anda juga harus mengatur`SplitType`dan`AssembleWith`parameter untuk`line`. Jika Anda mengatur format input dan output ke`application/jsonlines`, Anda dapat mengatur keduanya`SplitType`dan`AssembleWith`kepada`line`.

Untuk file CSV, Anda tidak dapat menggunakan karakter baris baru yang disematkan. Untuk file JSON, nama atribut`SageMakerOutput`dicadangkan untuk output. File input JSON tidak dapat memiliki atribut dengan nama ini. Jika ya, data dalam file input mungkin ditimpa.

Operator JSONPath yang didukung

Untuk memfilter dan menggabungkan data input dan inferensi, gunakan subexpression JSONPath. SageMakermendukung hanya bagian dari operator JSONPath didefinisikan. Tabel berikut berisi daftar operator JSONPath didukung. Untuk data CSV, setiap baris diambil sebagai array JSON, jadi hanya JSONPaths berbasis indeks yang dapat diterapkan, mis.`$[0]`,`$[1:]`. Data CSV juga harus mengikutiFormat RFC.

Operator JSONPath	Deskripsi	Contoh
`$`	Elemen root untuk query. Operator ini diperlukan pada awal semua ekspresi path.	`$`
`.<name>`	Sebuah elemen anak titik-notated.	`$.id`
`*`	Sebuah wildcard. Gunakan di tempat nama atribut atau nilai numerik.	`$.id.*`
`['<name>' (,'<na`	Sebuah elemen braket-notated atau beberapa elemen anak.	`$['id','SageMaker0 utput']`
`[<number> (,<number>)]`	Indeks atau array indeks. Nilai indeks negatif juga didukung. SEBUAH`-1`index mengacu pada elemen terakhir dalam array.	`$[1]` , `$[1,3,5]`

Operator JSONPath	Deskripsi	Contoh
`[<start>:<end>]`	Operator slice array. The array slice () metode ekstrak bagian dari array dan mengembalikan array baru. Jika Anda menghilangkan`<start>`,SageMaker menggunakan elemen pertama dari array. Jika Anda menghilangkan`<end>`,SageMakermenggunakan elemen terakhir dari array.	`$[2:5]`, `$[:5]`, `$[2:]`

Bila menggunakan braket-notasi untuk menentukan beberapa elemen anak dari bidang tertentu, bersarang tambahan anak dalam kurung tidak didukung. Sebagai contoh,`$.field1.['child1','child2']`didukung sementara`$.field1.['child1','child2.grandchild']`tidak.

Untuk informasi selengkapnya tentang operator JSONPath, lihatJsonPathdi atasGitHub.

Contoh Batch Transform

Contoh berikut menunjukkan beberapa cara umum untuk menggabungkan data input dengan hasil prediksi.

Topik

- Contoh: Output Hanya Inferensi
- Contoh: Inferensi Output Bergabung dengan Input Data
- Contoh: Inferensi Output Bergabung dengan Input Data dan Kecualikan Kolom ID dari Input (CSV)
- Contoh: Inferensi Output Bergabung dengan Kolom ID dan Kecualikan Kolom ID dari Input (CSV)

Contoh: Output Hanya Inferensi

Secara default,`DataProcessing`parameter tidak bergabung hasil inferensi dengan masukan. Ini output hanya hasil inferensi.

Jika Anda ingin secara eksplisit menentukan agar tidak bergabung dengan hasil dengan masukan, gunakanAmazonSageMakerPythondan tentukan pengaturan berikut dalam panggilan transformator.

```
sm_transformer = sagemaker.transformer.Transformer(…)
sm_transformer.transform(…, input_filter="$", join_source= "None", output_filter="$")
```

Untuk output kesimpulan menggunakanAWSSDK untuk Python, tambahkan kode berikut keCreateTransformJobpermintaan. Kode berikut meniru perilaku default.

```
{
    "DataProcessing": {
        "InputFilter": "$",
        "JoinSource": "None",
        "OutputFilter": "$"
    }
}
```

Contoh: Inferensi Output Bergabung dengan Input Data

Jika Anda menggunakanAmazonSageMakerPythonuntuk menggabungkan data input dengan kesimpulan dalam file output, tentukan`assemble_with`dan`accept`parameter saat menginisialisasi objek transformator. Saat Anda menggunakan panggilan transformasi, tentukan`Input`untuk`join_source`parameter, dan tentukan`split_type`dan`content_type`parameter juga. Yang`split_type`parameter harus memiliki nilai yang sama seperti`assemble_with`, dan`content_type`parameter harus memiliki nilai yang sama seperti`accept`. Untuk informasi lebih lanjut tentang parameter dan nilai yang diterima, lihatTrafohalaman diAmazonSageMakerPython.

```
sm_transformer = sagemaker.transformer.Transformer(…, assemble_with="Line",
 accept="text/csv")
sm_transformer.transform(…, join_source="Input", split_type="Line", content_type="text/
csv")
```

Jika Anda menggunakanAWSSDK untuk Python (Boto 3), gabungkan semua data masukan dengan inferensi dengan menambahkan kode berikut ke`CreateTransformJob`permintaan. Nilai-nilai untuk`Accept`dan`ContentType`harus cocok, dan nilai-nilai untuk`AssembleWith`dan`SplitType`juga harus cocok.

```
{
    "DataProcessing": {
        "JoinSource": "Input"
    },
    "TransformOutput": {
```

```
        "Accept": "text/csv",
        "AssembleWith": "Line"
    },
    "TransformInput": {
        "ContentType": "text/csv",
        "SplitType": "Line"
    }
}
```

Untuk file input JSON atau JSON Lines, hasilnya ada di`SageMakerOutput`kunci dalam file JSON masukan. Misalnya, jika input adalah file JSON yang berisi pasangan kunci-nilai`{"key":1}`, hasil transformasi data mungkin`{"label":1}`.

SageMakermenyimpan baik dalam file input di`SageMakerInput`kunci.

```
{
    "key":1,
    "SageMakerOutput":{"label":1}
}
```

> **Note**
>
> Hasil gabungan untuk JSON harus berupa objek pasangan kunci-nilai. Jika input bukan objek pasangan kunci-nilai,SageMakermembuat file JSON baru. Dalam file JSON baru, data input disimpan di`SageMakerInput`kunci dan hasilnya disimpan sebagai`SageMakerOutput`nilai.

Untuk file CSV, misalnya, jika rekaman tersebut`[1,2,3]`, dan hasil labelnya adalah`[1]`, maka file output akan berisi`[1,2,3,1]`.

Contoh: Inferensi Output Bergabung dengan Input Data dan Kecualikan Kolom ID dari Input (CSV)

Jika Anda menggunakanAmazonSageMakerPythonuntuk menggabungkan data input Anda dengan output inferensi sementara mengecualikan kolom ID dari input transformator, tentukan parameter yang sama dari contoh sebelumnya serta subexpression JSONPath untuk`input_filter`dalam panggilan transformator Anda. Misalnya, jika data input Anda mencakup lima kolom dan yang pertama adalah kolom ID, gunakan permintaan transformasi berikut untuk memilih semua kolom kecuali kolom ID sebagai fitur. Transformator masih mengeluarkan semua kolom input yang bergabung dengan kesimpulan. Untuk informasi lebih lanjut tentang parameter dan nilai yang diterima, lihatTrafohalaman diAmazonSageMakerPython.

```
sm_transformer = sagemaker.transformer.Transformer(…, assemble_with="Line",
 accept="text/csv")
sm_transformer.transform(…, split_type="Line", content_type="text/csv",
 input_filter="$[1:]", join_source="Input")
```

Jika Anda menggunakanAWSSDK untuk Python (Boto 3), tambahkan kode berikut untuk Anda `CreateTransformJob`permintaan.

```
{
    "DataProcessing": {
        "InputFilter": "$[1:]",
        "JoinSource": "Input"
    },
    "TransformOutput": {
        "Accept": "text/csv",
        "AssembleWith": "Line"
    },
    "TransformInput": {
        "ContentType": "text/csv",
        "SplitType": "Line"
    }
}
```

Untuk menentukan kolom diSageMaker, menggunakan indeks elemen array. Kolom pertama adalah indeks 0, kolom kedua adalah indeks 1, dan kolom keenam adalah indeks 5.

Untuk mengecualikan kolom pertama dari input, atur`InputFilter`kepada`"$[1:]"`. Usus besar (`:`) memberitahuSageMakeruntuk menyertakan semua elemen antara dua nilai, inklusif. Sebagai contoh,`$[1:4]`menentukan kedua melalui kolom kelima.

Jika Anda menghilangkan angka setelah usus besar, misalnya,`[5:]`, subset mencakup semua kolom dari kolom ke-6 melalui kolom terakhir. Jika Anda menghilangkan angka sebelum usus besar, misalnya,`[:5]`, subset mencakup semua kolom dari kolom pertama (indeks 0) melalui kolom keenam.

Contoh: Inferensi Output Bergabung dengan Kolom ID dan Kecualikan Kolom ID dari Input (CSV)

Jika Anda menggunakanAmazonSageMakerPython, Anda dapat menentukan output untuk bergabung hanya kolom masukan tertentu (seperti kolom ID) dengan kesimpulan dengan menentukan`output_filter`dalam panggilan transformator. Yang`output_filter`menggunakan

subexpression JSONPath untuk menentukan kolom mana yang akan dikembalikan sebagai output setelah bergabung dengan data input dengan hasil inferensi. Permintaan berikut menunjukkan bagaimana Anda dapat membuat prediksi saat mengecualikan kolom ID dan kemudian bergabung dengan kolom ID dengan inferensi. Perhatikan bahwa dalam contoh berikut, kolom terakhir (`-1`) dari output berisi kesimpulan. Jika Anda menggunakan file JSON,SageMakermenyimpan hasil inferensi dalam atribut`SageMakerOutput`. Untuk informasi lebih lanjut tentang parameter dan nilai yang diterima, lihatTrafohalaman diAmazonSageMakerPython.

```
sm_transformer = sagemaker.transformer.Transformer(…, assemble_with="Line",
 accept="text/csv")
sm_transformer.transform(…, split_type="Line", content_type="text/csv",
 input_filter="$[1:]", join_source="Input", output_filter="$[0,-1]")
```

Jika Anda menggunakanAWSSDK untuk Python (Boto 3), bergabung hanya kolom ID dengan inferensi dengan menambahkan kode berikut untuk Anda`CreateTransformJob`permintaan.

```
{
    "DataProcessing": {
        "InputFilter": "$[1:]",
        "JoinSource": "Input",
        "OutputFilter": "$[0,-1]"
    },
    "TransformOutput": {
        "Accept": "text/csv",
        "AssembleWith": "Line"
    },
    "TransformInput": {
        "ContentType": "text/csv",
        "SplitType": "Line"
    }
}
```

> **Warning**
>
> Jika Anda menggunakan file input berformat JSON, file tidak dapat berisi nama atribut`SageMakerOutput`. Nama atribut ini dicadangkan untuk kesimpulan dalam file output. Jika file input berformat JSON Anda berisi atribut dengan nama ini, nilai dalam file input mungkin ditimpa dengan inferensi.

Penyimpanan di Batch Transform

Saat Anda menjalankan pekerjaan transformasi batch, AmazonSageMakermelampirkan volume penyimpanan Amazon Elastic Block Store ke instans Amazon EC2 yang memproses pekerjaan Anda. Volume menyimpan model Anda, dan ukuran volume penyimpanan ditetapkan pada 30 GB. Anda memiliki opsi untuk mengenkripsi model Anda saat istirahat dalam volume penyimpanan.

> **Note**
>
> Jika Anda memiliki model besar, Anda mungkin mengalami`InternalServerError`.

Untuk informasi selengkapnya tentang penyimpanan dan fitur Amazon EBS, lihat halaman berikut:

- Amazon EBSdalam Panduan Pengguna Amazon EC2 untuk Instans Linux
- Volume Amazon EBSdalam Panduan Pengguna Amazon EC2 untuk Instans Linux

> **Note**
>
> Instans G4dn dilengkapi dengan penyimpanan SSD lokal mereka sendiri. Untuk mempelajari lebih lanjut tentang instans G4dn, lihatInstans Amazon EC2 G4halaman.

Pemecahan Masalah

Jika Anda mengalami kesalahan di AmazonSageMakerBatch Transform, lihat tips pemecahan masalah berikut.

Kesalahan batas waktu maks

Jika Anda mendapatkan error batas waktu maksimal saat menjalankan pekerjaan transformasi batch, coba yang berikut ini:

- Mulailah dengan rekaman tunggal`BatchStrategy`, ukuran batch default (6 MB) atau lebih kecil yang Anda tentukan di`MaxPayloadInMB`parameter, dan dataset sampel kecil. Tune parameter batas waktu maksimum`InvocationsTimeoutInSeconds`(yang memiliki maksimum 1 jam) hingga Anda menerima respons pemanggilan yang berhasil.

- Setelah Anda menerima respons doa yang berhasil, tingkatkan`MaxPayloadInMB`(yang memiliki maksimum 100 MB) dan`InvocationsTimeoutInSeconds`parameter bersama-sama untuk menemukan ukuran batch maksimum yang dapat mendukung batas waktu model yang Anda inginkan. Anda dapat menggunakan rekaman tunggal atau multi-record`BatchStrategy`dalam langkah ini.

  > **Note**
  >
  > Melebihi`MaxPayloadInMB`batas menyebabkan kesalahan. Ini mungkin terjadi dengan dataset besar jika tidak dapat dibagi,`SplitType`parameter diatur ke none, atau catatan individu dalam dataset melebihi batas.

- (Opsional) Tune`MaxConcurrentTransforms`parameter, yang menentukan jumlah maksimum permintaan paralel yang dapat dikirim ke setiap contoh dalam batch mengubah pekerjaan. Namun, nilai`MaxConcurrentTransforms * MaxPayloadInMB`tidak boleh melebihi 100 MB.

Output tidak lengkap

SageMakermenggunakan Amazon S3API Unggah Multipartuntuk mengunggah hasil dari pekerjaan transformasi batch ke Amazon S3. Jika terjadi kesalahan, hasil yang diunggah akan dihapus dari Amazon S3. Dalam beberapa kasus, seperti saat terjadi pemadaman jaringan, unggahan multibagian yang tidak lengkap mungkin tetap ada di Amazon S3. Upload yang tidak lengkap mungkin juga terjadi jika Anda memiliki beberapa file input tetapi beberapa file tidak dapat diproses olehSageMakerBatch Transform. File input yang tidak dapat diproses tidak akan memiliki file keluaran yang sesuai di Amazon S3.

Untuk menghindari biaya penyimpanan, kami sarankan Anda menambahkanKebijakan bucket S3ke aturan siklus hidup bucket S3. Kebijakan ini menghapus unggahan multibagian yang tidak lengkap yang mungkin disimpan di bucket S3. Untuk informasi lebih lanjut, lihat Manajemen Siklus Aktif Objek.

Pekerjaan menunjukkan sebagai**`failed`**

Jika pekerjaan transformasi batch gagal memproses file input karena masalah dengan kumpulan data,SageMakermenandai pekerjaan sebagai`failed`. Jika file input berisi catatan yang buruk, pekerjaan transformasi tidak membuat file output untuk file input itu karena hal itu mencegahnya mempertahankan urutan yang sama dalam data yang diubah seperti pada file input. Ketika dataset

Anda memiliki beberapa file input, pekerjaan transformasi terus memproses file input meskipun gagal memprosesnya. File yang diproses masih menghasilkan hasil yang dapat digunakan.

Jika Anda menggunakan algoritme Anda sendiri, Anda dapat menggunakan teks placeholder, sepertiERR0R, ketika algoritma menemukan catatan buruk dalam file input. Misalnya, jika catatan terakhir dalam kumpulan data buruk, algoritme menempatkan teks placeholder untuk catatan itu dalam file output.

Model paralelisme dan inferensi model besar

S model tate-of-the-art deep learning untuk aplikasi seperti natural language processing (NLP) berukuran besar, biasanya dengan puluhan atau ratusan miliar parameter. Model yang lebih besar seringkali lebih akurat, yang membuatnya menarik bagi praktisi pembelajaran mesin. Namun, model ini seringkali terlalu besar untuk dipasang pada akselerator tunggal atau perangkat GPU, sehingga sulit untuk mencapai inferensi latensi rendah. Anda dapat menghindari hambatan memori ini dengan menggunakan teknik paralelisme model untuk mempartisi model di beberapa akselerator atau GPU.

Amazon SageMaker menyertakan wadah pembelajaran mendalam (DLC) khusus, pustaka, dan perkakas untuk paralelisme model dan inferensi model besar (LMI). Di bagian berikut, Anda dapat menemukan sumber daya untuk memulai dengan LMI aktif. SageMaker

Topik

Wadah pembelajaran mendalam untuk inferensi model besar

SageMaker memelihara deep learning container (DLC) dengan pustaka open source populer untuk menghosting model besar seperti GPT, T5, OPT, BLOOM, dan Stable Diffusion pada infrastruktur.

AWS Dengan DLC ini Anda dapat menggunakan pustaka pihak ketiga seperti DeepSpeed, Accelerate, TensorRT-LLM, dan FasterTransformeruntuk mempartisi parameter model menggunakan teknik paralelisme model untuk memanfaatkan memori beberapa GPU untuk inferensi. Tabel berikut mencantumkan DLC yang tersedia dengan SageMaker inferensi model besar (LMI). Kami menyarankan Anda memulai dengan DLC ini untuk LMI aktif. SageMaker Ini termasuk komponen, pustaka, dan driver yang telah dioptimalkan dan diuji untuk digunakan pada SageMaker.

JENIS DLC	Perpustakaan	Parameter penyetelan
763104351884.dkr.ecr.*wilayah .amazonaws.com/djl-inference:0.25.0-tensorrtllm0.5.0-cu122*	PyTorch 2.1.0	Tensorrt-LLM
	DJL Melayani 0.25.0	
	Toolkit TensorRT 0.5.0	
	Akselerasi Hugging Face 0.23.0	
	Trafo Hugging Face 4.34.0	
763104351884.dkr.ecr.*wilayah .amazonaws.com/djl-inference:0.25.0-deepspeed0.11.0-cu118*	PyTorch 2.0.1	DeepSpeed
	DJL Melayani 0.25.0	
	DeepSpeed 0.11.0	
	Akselerasi Hugging Face 0.23.0	
	Trafo Hugging Face 4.34.0	
763104351884.dkr.ecr.*wilayah .amazonaws.com/djl-inference:0.25.0-neuronx-sdk2.15.0*	PyTorch 1.13.1	Transformer-Neuronx
	DJL Melayani 0.25.0	
	TransformersNeuronX 0.8.268	
	AWSNeuron SDK 2.15.1	
	NeuronX Didistribusikan 0.5.0	

JENIS DLC	Perpustakaan	Parameter penyetelan
763104351884.dkr.ecr.*wilayah .amazonaws.com/djl-inference:0.23.0-fastertransformer5.3.0-cu118*	PyTorch 2.0.1	FasterTransformer
	DJL Melayani 0.23.0	
	FasterTransformer 5.3.0	
	Akselerasi Hugging Face 0.20.3	
	Trafo Hugging Face 4.30.1	
763104351884.dkr.ecr.*wilayah .amazonaws.com/djl-inference:0.23.0-deepspeed0.9.5-cu118*	PyTorch 2.0.1	DeepSpeed
	DJL Melayani 0.23.0	
	DeepSpeed 0.9.5	
	Akselerasi Hugging Face 0.20.3	
	Trafo Hugging Face 4.30.2	
763104351884.dkr.ecr.*wilayah .amazonaws.com/djl-inference:0.23.0-neuronx-sdk2.12.0*	PyTorch 1.13.1	Transformer-Neuronx
	DJL Melayani 0.23.0	
	TransformersNeuronX 0,4,60	
	AWSNeuron SDK 2.12.0	
	NeuronX Didistribusikan 0.1.0	

Selain itu PyTorch, DLC LMI menyertakan perpustakaan untuk memfasilitasi inferensi model besar. SageMaker mendukung kategori perpustakaan berikut.

- Kebun binatang model — Kebun binatang model menyediakan akses API sederhana ke model yang telah dilatih sebelumnya. SageMaker menyediakan kebun binatang model berikut:

  - Hugging Face Transformers adalah perpustakaan populer untuk model pembelajaran mendalam pra-terlatih yang menggunakan arsitektur transformator seperti GPT, OPT, dan BLOOM.
  - Hugging Face Diffusers adalah perpustakaan dengan model pembelajaran mendalam pra-terlatih yang menggunakan teknik difusi seperti Difusi Stabil.
- Paralelisme model dan perpustakaan optimasi inferensi — Pustaka ini menangani inferensi paralel model dengan mempartisi artefak model sehingga parameternya dapat tersebar di beberapa GPU. SageMaker mendukung paralelisme model berikut dan perpustakaan optimasi inferensi:
  - TensorRT-LLM adalah perpustakaan open-source terbaru dari NVIDIA yang tersedia sebagai bagian dari rilis LMI DLC (0.25.0). Pustaka ini memungkinkan state-of-the-art pengoptimalan, seperti SmoothQuant, FP8 dan batch berkelanjutan untuk model bahasa besar saat menggunakan GPU NVIDIA.

    TensorRT-LLM mendukung penerapan mulai dari konfigurasi GPU tunggal hingga multi-GPU, dengan peningkatan kinerja tambahan yang dimungkinkan melalui teknik seperti paralelisme tensor. Memanfaatkan TensorRT-LLM melalui SageMaker LMI DLC memungkinkan Anda mengoptimalkan kinerja model bahasa besar Anda dan memberikan pengalaman yang lebih responsif kepada pengguna Anda.

    Kami mengoptimalkan pustaka TensorRT-LLM untuk mempercepat inferensi, dan kami membuat toolkit yang mendukung konversi model. just-in-time Anda dapat menggunakan toolkit ini untuk memberikan ID model Hugging Face dan menerapkan model ujung ke ujung. Toolkit ini juga mendukung batching terus menerus dengan streaming. Anda dapat mengkompilasi model Llama-2 7B dan 13B dalam waktu sekitar 1-2 menit, dan Anda dapat mengkompilasi model 70B dalam waktu sekitar 7 menit.

    Untuk menghindari overhead kompilasi saat menyiapkan SageMaker titik akhir dan menskalakan instans Anda, Anda dapat menggunakan kompilasi Ahead of Time (AOT). Untuk informasi lebih lanjut tentang menyiapkan model, lihat tutorial kompilasi model TensorRT-LLM ahead-of-time .

    Kami juga menerima model TensorRT LLM yang dibuat untuk Triton Server yang dapat digunakan dengan LMI DLC. Untuk contoh tentang penerapan Llama-2 70B menggunakan wadah TRT-LLM 0.25.0 LMI, lihat notebook.
  - DeepSpeed Inferensi adalah pustaka optimasi inferensi sumber terbuka. Ini termasuk skema partisi model untuk paralelisme model dengan model yang didukung, termasuk banyak model transformator. Ini juga telah mengoptimalkan kernel untuk model populer seperti OPT, GPT, dan BLOOM yang dapat secara signifikan meningkatkan latensi inferensi. Versi DeepSpeed

dalam DLC LMI dioptimalkan dan diuji untuk dikerjakan. SageMaker Ini mencakup beberapa peningkatan, termasuk dukungan untuk model presisi BF16.

- Hugging Face Accelerate di perpustakaan inferensi paralel model open-source. Ini mendukung paralelisme model untuk sebagian besar model di perpustakaan Hugging Face Transformers.
- FasterTransformeradalah perpustakaan open source dari Nvidia yang menyediakan mesin akselerasi untuk menjalankan inferensi jaringan saraf berbasis transformator secara efisien. Ini telah dirancang untuk menangani model besar yang membutuhkan banyak GPU dan node secara terdistribusi. Pustaka mencakup versi blok transformator yang dioptimalkan, yang terdiri dari bagian encoder dan decoder, memungkinkan Anda menjalankan inferensi arsitektur encoder-decoder penuh seperti T5, serta model khusus encoder seperti BERT dan model khusus decoder seperti GPT.

- Server model — Server model menangani permintaan inferensi dari ujung ke ujung. Mereka menerima permintaan, memanggil skrip pra-pemrosesan dan pasca-pemrosesan, dan menanggapi pengguna. Server model yang kompatibel dengan paralelisme model, juga mengatur pekerja dan utas di beberapa perangkat. SageMaker mendukung server model berikut:
  - DJL-serving adalah server model open-source berkinerja tinggi yang didukung oleh DJL. Dibutuhkan beberapa model pembelajaran mendalam atau alur kerja, dan membuatnya tersedia melalui titik akhir HTTP. Versi 0.19 ke atas didukung oleh SageMaker dan bekerja dengan instans Amazon EC2 dengan beberapa GPU untuk memfasilitasi LMI dengan paralelisme model.

Tipe instans yang didukung

AWSLMI DLC mendukung`p4d`,, `p3g5`, dan `g4dn` jenis instance.

SageMaker parameter titik akhir untuk inferensi model besar

Anda dapat menyesuaikan parameter berikut untuk memfasilitasi inferensi model besar latensi rendah (LMI) dengan: SageMaker

- Ukuran volume Amazon EBS maksimum pada instance (**`VolumeSizeInGB`**) — Jika ukuran model lebih besar dari 30 GB dan Anda menggunakan instance tanpa disk lokal, Anda harus meningkatkan parameter ini menjadi sedikit lebih besar dari ukuran model Anda.
- Kuota batas waktu pemeriksaan Kesehatan (**`ContainerStartupHealthCheckTimeoutInSeconds`**) — Jika wadah Anda diatur dengan benar dan CloudWatch log menunjukkan batas waktu pemeriksaan kesehatan, Anda harus

menambah kuota ini sehingga wadah memiliki cukup waktu untuk menanggapi pemeriksaan kesehatan.

- Kuota batas waktu unduhan model (**`ModelDataDownloadTimeoutInSeconds`**) — Jika ukuran model Anda lebih besar dari 40 GB, maka Anda harus menambah kuota ini untuk memberikan waktu yang cukup untuk mengunduh model dari Amazon S3 ke instans.

Untuk informasi selengkapnya tentang inferensi latensi rendah dengan model besar, lihat Menerapkan model besar di Amazon SageMaker menggunakan Penyajian DJL dan inferensi paralel model DeepSpeed . Cuplikan kode berikut menunjukkan cara mengkonfigurasi parameter yang disebutkan di atas secara terprogram. Ganti *`teks placeholder yang dicetak miring`* dalam contoh dengan informasi Anda sendiri.

```
import boto3

aws_region = "aws-region"
sagemaker_client = boto3.client('sagemaker', region_name=aws_region)

# The name of the endpoint. The name must be unique within an AWS Region in your AWS
 account.
endpoint_name = "endpoint-name"

# Create an endpoint config name.
endpoint_config_name = "endpoint-config-name"

# The name of the model that you want to host.
model_name = "the-name-of-your-model"

instance_type = "instance-type"

sagemaker_client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name
    ProductionVariants=[
        {
            "VariantName": "variant1", # The name of the production variant.
            "ModelName": model_name,
            "InstanceType": instance_type, # Specify the compute instance type.
            "InitialInstanceCount": 1, # Number of instances to launch initially.
            "VolumeSizeInGB": 256, # Specify the size of the Amazon EBS volume.
            "ModelDataDownloadTimeoutInSeconds": 1800, # Specify the model download
 timeout in seconds.
```

```
            "ContainerStartupHealthCheckTimeoutInSeconds": 1800, # Specify the health
 checkup timeout in seconds
        },
    ],
)

sagemaker_client.create_endpoint(EndpointName=endpoint_name,
 EndpointConfigName=endpoint_config_name)
```

Untuk informasi selengkapnya tentang kunci`ProductionVariants`, lihat `ProductionVariant`.

Tutorial inferensi model besar

Tutorial berikut menunjukkan penggunaan paralelisasi model yang berbeda dan perpustakaan optimasi inferensi yang tersedia di dalam DLC inferensi model besar (LMI). Anda dapat menggunakan ini untuk memulai dengan LMI aktif. SageMaker

Topik

- Inferensi model besar dengan DeepSpeed dan DJL Serving
- Inferensi model besar dengan FasterTransformer dan DJL Serving
- Inferensi model besar dengan TorchServe
- Sumber daya tambahan untuk memulai

Inferensi model besar dengan DeepSpeed dan DJL Serving

Tutorial ini menunjukkan bagaimana menerapkan model besar dengan DJL Serving menggunakan dan DeepSpeed Hugging Face Accelerate model kerangka paralelisasi. Contoh ini menggunakan model GPT-J dengan 6 miliar parameter dan instance. `ml.g5` Anda dapat memodifikasi ini untuk bekerja dengan model dan jenis instance lainnya. Ganti *`teks placeholder yang dicetak miring`* dalam contoh dengan informasi Anda sendiri.

Topik

- Gambar wadah inferensi model besar dengan DJL Serving
- Mempersiapkan artefak model Anda
- Menerapkan model menggunakan SDK SageMaker

Gambar wadah inferensi model besar dengan DJL Serving

DLC inferensi model besar (LMI) tersedia sebagai gambar Docker di Amazon Elastic Container Registry (Amazon ECR). Wadah ini mencakup komponen, pustaka, dan driver yang diperlukan untuk meng-host model besar di SageMaker atau menggunakan infrastruktur Amazon EC2. Anda dapat menemukan daftar DLC LMI di bagian inferensi model besar dari daftar DLC. Tutorial ini menggunakan wadah berikut:

Kerangka Kerja	Jenis Tugas	CPU/GPU	Opsi versi Python	Contoh URL
DJL Melayani 0.25.0 dengan DeepSpeed 0.11.0, Hugging Face Transformers 4.34.0, dan Hugging Face Accelerate 0.23.0	Inferensi	GPU	3.9 (py39)	763104351884.dkr. ecr.us-east-1.amazonaws.com /djl-inference:0.25.0-deepspeed0.11.0-cu118

DLC LMI diperbarui dan diuji secara berkala. SageMaker Anda dapat langsung mereferensikan URL gambar Docker ini saat Anda meng-host SageMaker, memilih wadah yang paling sesuai dengan kebutuhan Anda. Setiap gambar DLC LMI termasuk DeepSpeed, Accelerate, Transformers dan DJL Serving.

Mempersiapkan artefak model Anda

LMI DLC menggunakan DJL Serving untuk melayani model Anda untuk inferensi. Anda harus mengkonfigurasi DJL Serving dan paket model Anda dalam format yang didukung oleh DJL Serving dan DeepSpeed untuk LMI.

1. Buat file `serving.properties` konfigurasi untuk menunjukkan ke DJL Serving mana model paralelisasi dan perpustakaan optimasi inferensi yang ingin Anda gunakan. Anda dapat menemukan opsi konfigurasi untuk keduanya DeepSpeed dan Hugging Face Accelerate Konfigurasi dan pengaturan in. Ubah `serving.properties` file contoh berikut di bawah ini agar sesuai dengan kebutuhan Anda.

```
engine=DeepSpeed
#engine=Python # for Hugging Face Accelerate, vLLM rolling batch
#engine=MPI # lmi-dist rolling batch option
option.entryPoint=djl_python.deepspeed
#option.entryPoint=djl_python.huggingface # for Hugging Face Accelerate
option.tensor_parallel_degree=2
option.model_id=EleutherAI/gpt-j-6B
option.max_rolling_batch_size=64
option.rolling_batch=deepspeed
option.task=text-generation
option.dtype=fp16
```

Untuk daftar lengkap opsi konfigurasi, lihatKonfigurasi dan pengaturan. Daftar berikut menjelaskan beberapa opsi yang digunakan dalam `serving.properties` file contoh.

- **`option.entryPoint`**— Opsi ini digunakan untuk menentukan handler mana yang ditawarkan oleh DJL Serving yang ingin Anda gunakan. Nilai yang mungkin adalah `huggingface`, `deepspeed`, dan `stable-diffusion`. Di sini kita gunakan`deepspeed`.
- **`option.model_id`**— Hanya berikan opsi ini jika Anda juga menyediakan`option.entryPoint`. Nilai opsi ini adalah ID Wajah Pelukan model atau URL S3 dari artefak model. DJL Serving menggunakan nilai ini untuk mengunduh model dari Hugging Face atau URL S3.
- **`option.task`**— Arsitektur model umumnya mengharuskan Anda menentukan tugas sebagai salah satu parameter di`transformers.pipeline`.
- **`option.rolling_batch`**— Pilih strategi batch bergulir. Nilai `auto` akan membuat handler memilih strategi berdasarkan konfigurasi yang disediakan. Jika Anda menentukan `option.rolling_batch=djl_python.huggingface` dan`auto`, salah satu opsi berikut akan dipilih secara otomatis berdasarkan model:`lmi-dist`, atau`scheduler`. Jika mesin DeepSpeed, maka `deepspeed` rolling batch dipilih.

Untuk informasi selengkapnya, lihat implementasi handler DeepSpeed, dan Hugging Face.

2. Package artefak model Anda ke dalam file.tar terkompresi. DJL Melayani dan DeepSpeed mengharapkan artefak model dikemas dan diformat dengan cara tertentu. Contoh ini menggunakan struktur direktori berikut untuk tujuan ini.

```
- deepspeed-gptj # root directory
   - serving.properties
```

```
    - model.py # your custom handler file, if you choose not to use the handlers
 provided with DJL Serving
    - model binary files # if you do not want to use option.model_id
```

Direktori root berisi `serving.properties` file, bersama dengan file lain yang diperlukan. Tutorial ini hanya membutuhkan `serving.properties` file karena menggunakan handler DJL Serving dan `model_id` opsi untuk mengunduh model langsung dari Hugging Face. Cuplikan kode berikut mengemas artefak model ke dalam file.tar terkompresi.

```
mkdir deepspeed-gptj
cp serving.properties deepspeed-gptj/
tar -czvf deepspeed-gptj.tar.gz deepspeed-gptj
aws s3 sync deepspeed-gptj.tar.gz s3://djl-sm-test/tutorial/
```

Menerapkan model menggunakan SDK SageMaker

Contoh ini digunakan SageMaker untuk menangani proses penerapan model ujung ke ujung ke titik akhir inferensi.

1. Buat SageMaker sesi.

```
import boto3
import sagemaker

aws_region = "aws-region"
sagemaker_session =
 sagemaker.Session(boto_session=boto3.Session(region_name=aws_region))
```

2. Buat model di SageMaker.

```
from sagemaker.model import Model
from sagemaker import image_uris, get_execution_role

role = get_execution_role()

def create_model(model_name, model_s3_url):
    # Get the DJL DeepSpeed image uri
    image_uri = image_uris.retrieve(
```

```
        framework="djl-deepspeed",
        region=sagemaker_session.boto_session.region_name,
        version="0.25.0"
    )
    model = Model(
        image_uri=image_uri,
        model_data=model_s3_url,
        role=role,
        name=model_name,
        sagemaker_session=sagemaker_session,
    )
    return model
```

`g5.12xlarge`Instans ini memiliki 4 GPU Nvidia A10G. `serving.properties`Dalam Mempersiapkan artefak model Anda ditentukan derajat paralel tensor 2. DJL Serving secara otomatis memuat 2 salinan model dengan 2 partisi paralel tensor untuk memanfaatkan semua 4 GPU pada instance. Ini menggandakan throughput.

3. Menyebarkan model ke sebuah `g5.12xlarge` instance.

```
from sagemaker import serializers, deserializers

def deploy_model(model, _endpoint_name):
    model.deploy(
        initial_instance_count=1,
        instance_type="ml.g5.12xlarge",
        endpoint_name=_endpoint_name
    )
    predictor = sagemaker.Predictor(
        endpoint_name=_endpoint_name,
        sagemaker_session=sagemaker_session,
        serializer=serializers.JSONSerializer(),
        deserializer=deserializers.JSONDeserializer()
    )
    return predictor
```

4. Buat prediksi.

```
import argparse
if __name__ == "__main__":
    arg_parser = argparse.ArgumentParser()
```

```
   group = arg_parser.add_mutually_group()
   group.add_argument("--deepspeed", action="store_const", dest="framework",
const="deepspeed")
   group.add_argument("--accelerate", action="store_const", dest="framework",
const="accelerate")

   args = arg_parser.parse_args()

   _model_name = f"{args.framework}-gptj"
   _model_s3_url = f"s3://djl-sm-test/tutorial/{args.framework}-gptj.tar.gz"
   _endpoint_name=f"{args.framework}-gptj"

   model = create_model(_model_name, _model_s3_url)
   predictor = deploy_model(model, _endpoint_name)
   print(predictor.predict(
               {
                   "inputs" : "Large model inference is",
                   "parameters": { "max_length": 50 },
               }
   ))
```

Inferensi model besar dengan FasterTransformer dan DJL Serving

Tutorial ini menunjukkan cara menerapkan model T5 dengan wadah pembelajaran mendalam (DLC) inferensi model besar (LMI), Penyajian DJL, dan kerangka paralelisasi model. FasterTransformer Di sini kita menggunakan model flan-t5-xl dengan 3 miliar parameter dan sebuah instance. `ml.g5` Anda dapat memodifikasi ini untuk bekerja dengan varian lain dari model T5 dan tipe instance. Ganti *`teks placeholder yang dicetak miring`* dalam contoh dengan informasi Anda sendiri.

Gambar wadah inferensi model besar dengan FasterTransformer backend untuk DJL Serving

DLC LMI tersedia sebagai gambar Docker di Amazon Elastic Container Registry (Amazon ECR) Registry Amazon. Wadah ini mencakup komponen, pustaka, dan driver yang diperlukan untuk meng-host model besar di SageMaker atau menggunakan infrastruktur Amazon EC2. Anda dapat menemukan daftar DLC LMI di bagian inferensi model besar dari daftar DLC. Tutorial ini menggunakan wadah berikut:

Kerangka Kerja	Jenis Tugas	CPU/GPU	Opsi versi Python	Contoh URL
DJL Melayani 0.23.0 dengan FasterTransformer 5.3, Hugging Face Transformers 4.30.1.	Inferensi	GPU	3.9 (py39)	763104351884.dkr. ecr.us-west-2.amazonaws.com /djl-inference:0.23.0-fastertransformer5.3.0-cu118

DLC LMI diperbarui dan diuji secara berkala. SageMaker Anda dapat langsung mereferensikan URL gambar Docker ini saat Anda meng-host SageMaker, memilih wadah yang paling sesuai dengan kebutuhan Anda. Setiap gambar DLC LMI termasuk DeepSpeed, Accelerate, Transformers dan DJL Serving.

Mempersiapkan artefak model Anda

LMI DLC menggunakan DJL Serving untuk melayani model Anda untuk inferensi. Anda harus mengkonfigurasi DJL Serving dan paket model Anda dalam format yang didukung oleh DJL Serving dan. FasterTransformer

1. Buat file `serving.properties` konfigurasi untuk menunjukkan ke DJL Serving mana model paralelisasi dan perpustakaan optimasi inferensi yang ingin Anda gunakan.

   ```
   engine=FasterTransformer
   option.tensor_parallel_degree=2
   option.model_id=google/flan-t5-xl
   option.dtype=fp32
   ```

   Untuk daftar lengkap opsi konfigurasi, lihatKonfigurasi dan pengaturan. Daftar berikut menjelaskan beberapa opsi yang digunakan dalam contoh yang disebutkan di atas.

   - **`option.tensor_parallel_degree`**— Opsi ini menentukan jumlah partisi paralel tensor yang dilakukan pada model.

- **`option.model_id`**— Nilai opsi ini adalah ID Hugging Face model atau URL S3 dari artefak model. DJL Serving menggunakan nilai ini untuk mengunduh model dari Hugging Face atau URL S3.
- **`option.dtype`**— Opsi ini menentukan tipe data yang ingin Anda jalankan inferensi.

2. Buat skrip `model.py` handler dengan `handle` metode yang akan dipanggil oleh DJL Serving ketika permintaan inferensi diterima. Tutorial ini menggunakan yang berikut`model.py`:

```
import fastertransformer as ft
from djl_python import Input, Output
from transformers import AutoTokenizer
import logging
import os

model = None

def load_model(properties):
    model_name = properties["model_id"]
    tensor_parallel_degree = int(properties["tensor_parallel_degree"])
    pipeline_parallel_degree = 1
    dtype = properties["dtype"]

    logging.info(f"Loading model: {model_name}")
    # Initialilizing model with FasterTransformer
    model = ft.init_inference(model_name, tensor_parallel_degree,
    pipeline_parallel_degree, dtype)
    return model


def handle(inputs: Input):
    global model

    if not model:
        model = load_model(inputs.get_properties())

    if inputs.is_empty():
        # Model server makes an empty call to warmup the model on startup
        return None

    data = inputs.get_as_json()
    input_text = data["text"]
    result = model.pipeline_generate(input_text)
```

```
    return Output().add(result)
```

3. Package artefak model Anda ke dalam file tar terkompresi. DJL Serving mengharapkan artefak model dikemas dan diformat dengan cara tertentu. Contoh ini menggunakan struktur direktori berikut untuk tujuan ini.

```
 - deepspeed-gptj # root directory
    - serving.properties
    - model.py # your custom handler file, if you choose not to use the handlers
provided with DJL Serving
```

Direktori root berisi `model.py` file `serving.properties` dan. Cuplikan kode berikut mengemas artefak model ke dalam file.tar terkompresi.

```
mkdir fastertransformer-t5
cp serving.properties fastertransformer-t5/
cp model.py fastertransformer-t5/
tar -czvf fastertransformer-t5.tar.gz fastertransformer-t5/
aws s3 cp fastertransformer-t5.tar.gz s3://djl-sm-test/tutorial/
```

Menerapkan model menggunakan SDK SageMaker

Contoh ini digunakan SageMaker untuk menangani proses penerapan model ujung ke ujung ke titik akhir inferensi.

1. Buat SageMaker sesi.

```
import boto3
import sagemaker

aws_region = "aws-region"
sagemaker_session =
 sagemaker.Session(boto_session=boto3.Session(region_name=aws_region))
```

2. Buat model di SageMaker.

```
from sagemaker.model import Model
from sagemaker import image_uris, get_execution_role
```

```
role = get_execution_role()

def create_model(model_name, model_s3_url):
    image_uri = "763104351884.dkr.ecr.us-west-2.amazonaws.com/djl-inference:0.23.0-
fastertransformer5.3.0-cu118"
    model = Model(
        image_uri=image_uri,
        model_data=model_s3_url,
        role=role,
        name=model_name,
        sagemaker_session=sagemaker_session,
    )
    return model
```

`g5.12xlarge`Instans ini memiliki 4 GPU Nvidia A10G. `serving.properties`Dalam Mempersiapkan artefak model Anda ditentukan derajat paralel tensor 2. DJL Serving secara otomatis memuat 2 salinan model dengan 2 partisi paralel tensor untuk memanfaatkan semua 4 GPU pada instance. Ini menggandakan throughput.

3. Menyebarkan model ke sebuah `g5.12xlarge` instance.

```
from sagemaker import serializers, deserializers

def deploy_model(model, _endpoint_name):
    model.deploy(
        initial_instance_count=1,
        instance_type="ml.g5.12xlarge",
        endpoint_name=_endpoint_name
    )
    predictor = sagemaker.Predictor(
        endpoint_name=_endpoint_name,
        sagemaker_session=sagemaker_session,
        serializer=serializers.JSONSerializer(),
        deserializer=deserializers.JSONDeserializer()
    )
    return predictor
```

4. Buat prediksi

```
_model_name = "ft-flan-t5-xl"
_model_s3_url = "s3://djl-sm-test/tutorial/fastertransforemer-t5.tar.gz"
_endpoint_name="ft-flan-t5-xl"
```

```
model = create_model(_model_name, _model_s3_url)
predictor = deploy_model(model, _endpoint_name)
print(predictor.predict(
            {
                "text" : ["translate English to German: The house is wonderful."],
            }
))
```

5. Putar sumber daya

```
predictor.delete_endpoint(delete_endpoint_config=True)
```

Inferensi model besar dengan TorchServe

Tutorial ini menunjukkan cara menerapkan model besar dan menyajikan inferensi di Amazon SageMaker dengan TorchServe pada GPU. Contoh ini menerapkan model OPT-30b ke sebuah instance. `ml.g5` Anda dapat memodifikasi ini untuk bekerja dengan model dan jenis instance lainnya. Ganti contoh *`italicized placeholder text`* dalam dengan informasi Anda sendiri.

TorchServe adalah platform terbuka yang kuat untuk inferensi model terdistribusi besar. Dengan mendukung pustaka populer seperti PyTorch, PiPPy asli,, DeepSpeed dan HuggingFace Accelerate, ia menawarkan API penangan seragam yang tetap konsisten di seluruh model besar terdistribusi dan skenario inferensi model non-terdistribusi. Untuk informasi lebih lanjut, TorchServelihat dokumentasi inferensi model besar.

Wadah pembelajaran mendalam dengan TorchServe

Untuk menerapkan model besar dengan TorchServe on SageMaker, Anda dapat menggunakan salah satu wadah pembelajaran SageMaker mendalam (DLC). Secara default, TorchServe diinstal di semua AWS PyTorch DLC. Selama pemuatan model, TorchServe dapat menginstal perpustakaan khusus yang disesuaikan untuk model besar seperti PiPPy, Deepspeed, dan Accelerate.

Tabel berikut mencantumkan semua SageMaker DLC dengan. TorchServe

Kateogri DLC	Kerangka Kerja	Perangkat keras	Contoh URL
SageMaker Kontainer Kerangka	PyTorch 2.0.0+	CPU, GPU	763104351884.dkr.ecr.us-east-1.amazonaws.com /pytorch-

Kategori DLC	Kerangka Kerja	Perangkat keras	Contoh URL
			inferensi:2.0.1-gpu-py310-cu118-ubuntu20.04-sagemaker
SageMaker Kerangka Wadah Graviton	PyTorch 2.0.0+	CPU	763104351884.dkr.ecr.us-east-1.amazonaws.com /:2.0.1-cpu-py310-ubuntu20.04-sagemaker pytorch-inference-graviton
Wadah Inferensi StabilityAI	PyTorch 2.0.0+	GPU	763104351884.dkr.ecr.us-east-1.amazonaws.com /:2.0.1-sgm0.1.0-gpu-py310-cu118-ubuntu20.04-sagemaker stabilityai-pytorch-inference
Wadah Neuron	PyTorch 1.13.1	Neuronx	763104351884.dkr.ecr.us-west-2.amazonaws.com /:1.13.1-neuron-py310-sdk2.12.0-ubuntu20.04 pytorch-inference-neuron

Memulai

Sebelum menerapkan model Anda, selesaikan prasyarat. Anda juga dapat mengonfigurasi parameter model Anda dan menyesuaikan kode handler.

Prasyarat

Untuk memulai, pastikan Anda memiliki prasyarat berikut:

1. Pastikan Anda memiliki akses ke AWS akun. Siapkan lingkungan Anda sehingga AWS CLI dapat mengakses akun Anda melalui pengguna AWS IAM atau peran IAM. Kami merekomendasikan menggunakan peran IAM. Untuk tujuan pengujian di akun pribadi Anda, Anda dapat melampirkan kebijakan izin terkelola berikut ke peran IAM:

   - AmazonEC2 ContainerRegistryFullAccess
   - AmazonEC2 FullAccess
   - AWSServiceRoleForAmazonEKSNodegroup
   - AmazonSageMakerFullAccess
   - AmazonS3 FullAccess

   Untuk informasi selengkapnya tentang melampirkan kebijakan IAM ke peran, lihat Menambahkan dan menghapus izin identitas IAM di Panduan Pengguna IAM. AWS
2. Konfigurasikan dependensi Anda secara lokal, seperti yang ditunjukkan pada contoh berikut.

   a. Instal versi 2 dariAWS CLI:

   ```
   # Install the latest AWS CLI v2 if it is not installed
   !curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
    "awscliv2.zip" !unzip awscliv2.zip
   #Follow the instructions to install v2 on the terminal
   !cat aws/README.md
   ```

   b. Instal SageMaker dan klien Boto3:

   ```
   # If already installed, update your client
   #%pip install sagemaker pip --upgrade --quiet
   !pip install -U sagemaker
   !pip install -U boto
   !pip install -U botocore
   !pip install -U boto3
   ```

Konfigurasikan pengaturan dan parameter model

TorchServe menggunakan torchrun untuk mengatur lingkungan terdistribusi untuk pemrosesan paralel model. TorchServe memiliki kemampuan untuk mendukung banyak pekerja untuk model besar. Secara default, TorchServe menggunakan algoritma round-robin untuk menetapkan GPU ke pekerja di host. Dalam kasus inferensi model besar, jumlah GPU yang ditetapkan untuk

setiap pekerja dihitung secara otomatis berdasarkan jumlah GPU yang ditentukan dalam file. `model_config.yaml` Variabel lingkunganCUDA_VISIBLE_DEVICES, yang menentukan ID perangkat GPU yang terlihat pada waktu tertentu, ditetapkan berdasarkan nomor ini.

Misalnya, ada delapan GPU pada sebuah node dan satu pekerja membutuhkan empat GPU pada node ()nproc_per_node=4. Dalam hal ini, TorchServe tetapkan empat GPU ke worker pertama (CUDA_VISIBLE_DEVICES="0,1,2,3") dan empat GPU ke worker kedua (). CUDA_VISIBLE_DEVICES="4,5,6,7”

Selain perilaku default ini, TorchServe memberikan fleksibilitas bagi pengguna untuk menentukan GPU untuk pekerja. Misalnya, jika Anda menyetel variabel `deviceIds: [2,3,4,5]` dalam file YAMAL konfigurasi model, dan mengaturnproc_per_node=2, kemudian TorchServe menetapkan CUDA_VISIBLE_DEVICES=”2,3” ke pekerja pertama dan pekerja CUDA_VISIBLE_DEVICES="4,5” kedua.

Dalam `model_config.yaml` contoh berikut, kami mengonfigurasi parameter front-end dan back-end untuk model OPT-30b. Parameter front-end yang dikonfigurasi adalah`parallelType`,`deviceType`, `deviceIds` dan. `torchrun` Untuk informasi lebih rinci tentang parameter front-end yang dapat Anda konfigurasi, lihat dokumentasi. PyTorch GitHub Konfigurasi back-end didasarkan pada peta YAMM yang memungkinkan kustomisasi gaya bebas. Untuk parameter back-end, kita mendefinisikan DeepSpeed konfigurasi dan parameter tambahan yang digunakan oleh kode handler kustom.

```
# TorchServe front-end parameters
minWorkers: 1
maxWorkers: 1
maxBatchDelay: 100
responseTimeout: 1200
parallelType: "tp"
deviceType: "gpu"
# example of user specified GPU deviceIds
deviceIds: [0,1,2,3] # sets CUDA_VISIBLE_DEVICES

torchrun:
    nproc-per-node: 4

# TorchServe back-end parameters
deepspeed:
    config: ds-config.json
    checkpoint: checkpoints.json
```

```
handler: # parameters for custom handler code
    model_name: "facebook/opt-30b"
    model_path: "model/models--facebook--opt-30b/snapshots/
ceea0a90ac0f6fae7c2c34bcb40477438c152546"
    max_length: 50
    max_new_tokens: 10
    manual_seed: 40
```

Sesuaikan penangan

TorchServe menawarkan penangan dasar dan utilitas penangan untuk inferensi model besar yang dibangun dengan perpustakaan populer. Contoh berikut menunjukkan bagaimana kelas handler kustom TransformersSeqClassifierHandlermeluas BaseDeepSpeedHandlerdan menggunakan utilitas handler. Untuk contoh kode lengkap, lihat `custom_handler.py`kode pada PyTorch GitHub dokumentasi.

```
class TransformersSeqClassifierHandler(BaseDeepSpeedHandler, ABC):
    """
    Transformers handler class for sequence, token classification and question
 answering.
    """

    def __init__(self):
        super(TransformersSeqClassifierHandler, self).__init__()
        self.max_length = None
        self.max_new_tokens = None
        self.tokenizer = None
        self.initialized = False

    def initialize(self, ctx: Context):
        """In this initialize function, the HF large model is loaded and
        partitioned using DeepSpeed.
        Args:
            ctx (context): It is a JSON Object containing information
            pertaining to the model artifacts parameters.
        """
        super().initialize(ctx)
        model_dir = ctx.system_properties.get("model_dir")
        self.max_length = int(ctx.model_yaml_config["handler"]["max_length"])
        self.max_new_tokens = int(ctx.model_yaml_config["handler"]["max_new_tokens"])
        model_name = ctx.model_yaml_config["handler"]["model_name"]
        model_path = ctx.model_yaml_config["handler"]["model_path"]
        seed = int(ctx.model_yaml_config["handler"]["manual_seed"])
```

```
        torch.manual_seed(seed)

        logger.info("Model %s loading tokenizer", ctx.model_name)

        self.tokenizer = AutoTokenizer.from_pretrained(model_name)
        self.tokenizer.pad_token = self.tokenizer.eos_token
        config = AutoConfig.from_pretrained(model_name)
        with torch.device("meta"):
            self.model = AutoModelForCausalLM.from_config(
                config, torch_dtype=torch.float16
            )
        self.model = self.model.eval()

        ds_engine = get_ds_engine(self.model, ctx)
        self.model = ds_engine.module
        logger.info("Model %s loaded successfully", ctx.model_name)
        self.initialized = True

    def preprocess(self, requests):
        """
        Basic text preprocessing, based on the user's choice of application mode.
        Args:
            requests (list): A list of dictionaries with a "data" or "body" field, each
                            containing the input text to be processed.
        Returns:
            tuple: A tuple with two tensors: the batch of input ids and the batch of
                attention masks.
        """

    def inference(self, input_batch):
        """
        Predicts the class (or classes) of the received text using the serialized
transformers
        checkpoint.
        Args:
            input_batch (tuple): A tuple with two tensors: the batch of input ids and
the batch
                                 of attention masks, as returned by the preprocess
function.
        Returns:
            list: A list of strings with the predicted values for each input text in
the batch.
        """
```

```
    def postprocess(self, inference_output):
        """Post Process Function converts the predicted response into Torchserve
readable format.
        Args:
            inference_output (list): It contains the predicted response of the input
text.
        Returns:
            (list): Returns a list of the Predictions and Explanations.
        """
```

Siapkan artefak model Anda

Sebelum menerapkan model Anda SageMaker, Anda harus mengemas artefak model Anda. Untuk model besar, kami menyarankan Anda menggunakan PyTorch torch-model-archiveralat dengan argumen`--archive-format no-archive`, yang melewatkan artefak model kompresi. Contoh berikut menyimpan semua artefak model ke folder baru bernama`opt/`.

```
torch-model-archiver --model-name opt --version 1.0 --handler custom_handler.py --
extra-files ds-config.json -r requirements.txt --config-file opt/model-config.yaml --
archive-format no-archive
```

Setelah `opt/` folder dibuat, unduh model OPT-30b ke folder menggunakan alat Download_model. PyTorch

```
cd opt
python path_to/Download_model.py --model_path model --model_name facebook/opt-30b --
revision main
```

Terakhir, unggah artefak model ke ember Amazon S3.

```
aws s3 cp opt {your_s3_bucket}/opt --recursive
```

Anda sekarang harus memiliki artefak model yang disimpan di Amazon S3 yang siap diterapkan ke titik akhir. SageMaker

Terapkan model menggunakan SageMaker Python SDK

Setelah menyiapkan artefak model, Anda dapat menerapkan model Anda ke titik akhir SageMaker Hosting. Bagian ini menjelaskan cara menerapkan satu model besar ke titik akhir dan membuat

prediksi respons streaming. Untuk informasi selengkapnya tentang respons streaming dari titik akhir, lihat Memanggil titik akhir waktu nyata.

Untuk menerapkan model Anda, selesaikan langkah-langkah berikut:

1. Buat SageMaker sesi, seperti yang ditunjukkan pada contoh berikut.

```
import boto3
import sagemaker
from sagemaker import Model, image_uris, serializers, deserializers

boto3_session=boto3.session.Session(region_name="us-west-2")
smr = boto3.client('sagemaker-runtime-demo')
sm = boto3.client('sagemaker')
role = sagemaker.get_execution_role()  # execution role for the endpoint
sess= sagemaker.session.Session(boto3_session, sagemaker_client=sm,
 sagemaker_runtime_client=smr)  # SageMaker session for interacting with different
 AWS APIs
region = sess._region_name  # region name of the current SageMaker Studio Classic
 environment
account = sess.account_id()  # account_id of the current SageMaker Studio Classic
 environment

# Configuration:
bucket_name = sess.default_bucket()
prefix = "torchserve"
output_path = f"s3://{bucket_name}/{prefix}"
print(f'account={account}, region={region}, role={role},
 output_path={output_path}')
```

2. Buat model yang tidak terkompresi di SageMaker, seperti yang ditunjukkan pada contoh berikut.

```
from datetime import datetime

instance_type = "ml.g5.24xlarge"
endpoint_name = sagemaker.utils.name_from_base("ts-opt-30b")
s3_uri = {your_s3_bucket}/opt

model = Model(
    name="torchserve-opt-30b" + datetime.now().strftime("%Y-%m-%d-%H-%M-%S"),
    # Enable SageMaker uncompressed model artifacts
    model_data={
        "S3DataSource": {
```

```
                "S3Uri": s3_uri,
                "S3DataType": "S3Prefix",
                "CompressionType": "None",
            }
        },
        image_uri=container,
        role=role,
        sagemaker_session=sess,
        env={"TS_INSTALL_PY_DEP_PER_MODEL": "true"},
)
print(model)
```

3. Menerapkan model ke instans Amazon EC2, seperti yang ditunjukkan pada contoh berikut.

```
model.deploy(
    initial_instance_count=1,
    instance_type=instance_type,
    endpoint_name=endpoint_name,
    volume_size=512, # increase the size to store large model
    model_data_download_timeout=3600, # increase the timeout to download large
 model
    container_startup_health_check_timeout=600, # increase the timeout to load
 large model
)
```

4. Inisialisasi kelas untuk memproses respons streaming, seperti yang ditunjukkan pada contoh berikut.

```
import io

class Parser:
    """
    A helper class for parsing the byte stream input.

    The output of the model will be in the following format:
    ```
    b'{"outputs": [" a"]}\n'
    b'{"outputs": [" challenging"]}\n'
    b'{"outputs": [" problem"]}\n'
    ...
    ```
```

```
   While usually each PayloadPart event from the event stream will contain a byte
array
   with a full json, this is not guaranteed and some of the json objects may be
split across
   PayloadPart events. For example:
   ```
   {'PayloadPart': {'Bytes': b'{"outputs": '}}
   {'PayloadPart': {'Bytes': b'[" problem"]}\n'}}
   ```

   This class accounts for this by concatenating bytes written via the 'write'
function
   and then exposing a method which will return lines (ending with a '\n'
character) within
   the buffer via the 'scan_lines' function. It maintains the position of the last
read
   position to ensure that previous bytes are not exposed again.
   """

   def __init__(self):
       self.buff = io.BytesIO()
       self.read_pos = 0

   def write(self, content):
       self.buff.seek(0, io.SEEK_END)
       self.buff.write(content)
       data = self.buff.getvalue()

   def scan_lines(self):
       self.buff.seek(self.read_pos)
       for line in self.buff.readlines():
           if line[-1] != b'\n':
               self.read_pos += len(line)
               yield line[:-1]

   def reset(self):
       self.read_pos = 0
```

5. Uji prediksi respons streaming, seperti yang ditunjukkan pada contoh berikut.

```
import json

body = "Today the weather is really nice and I am planning on".encode('utf-8')
```

```
resp = smr.invoke_endpoint_with_response_stream(EndpointName=endpoint_name,
 Body=body, ContentType="application/json")
event_stream = resp['Body']
parser = Parser()
for event in event_stream:
    parser.write(event['PayloadPart']['Bytes'])
    for line in parser.scan_lines():
        print(line.decode("utf-8"), end=' ')
```

Anda sekarang telah menerapkan model Anda ke SageMaker titik akhir dan harus dapat memanggilnya untuk tanggapan. Untuk informasi selengkapnya tentang titik akhir SageMaker real-time, lihatTuan rumah satu model.

Sumber daya tambahan untuk memulai

Untuk informasi selengkapnya tentang penggunaan SageMaker DLC LMI, lihat sumber daya berikut:

- Blog: Meningkatkan kinerja throughput model Llama 2 menggunakan Amazon SageMaker
- Blog: Tingkatkan kinerja model Falcon dengan Amazon SageMaker
- Blog: Bangun aplikasi AI image-to-text generatif menggunakan model multimodalitas di Amazon SageMaker
- Blog: Terapkan model bahasa besar di AWS Inferentia2 menggunakan wadah inferensi model besar
- Blog: Terapkan BLOOM-176B dan OPT-30B di Amazon dengan inferensi model besar SageMaker Deep Learning Containers dan DeepSpeed
- Blog: Terapkan model besar di Amazon SageMaker menggunakan DJL Serving dan inferensi DeepSpeed paralel model
- Contoh notebook: AWS SageMaker contoh GitHub repositori.

Konfigurasi dan pengaturan

Bagian berikut menjelaskan opsi konfigurasi yang dapat Anda gunakan di dalam `serving.properties` file.

DJL Melayani pengaturan umum

Tabel berikut mengidentifikasi opsi konfigurasi DJL Serving umum yang dapat Anda gunakan`serving.properties`, terlepas dari penangan apa pun yang Anda pilih.

Parameter umum

Item	Diperlukan	Deskripsi	Nilai contoh
`engine`	Ya	Mesin runtime kode. MPI adalah mesin yang memungkin kan server model untuk memulai proses terdistribusi untuk memuat model. Ini digunakan dalam beberapa kerangka kerja yang didukung untuk LMI. Pada 0.25.0, TRTLLM, LMi-DIST dan DeepSpeed kerangka kerja menggunakan mesin MPI. vLLm, X, Optimum Neuron, Accelerate menggunakan mesin Python. Transform ersNeuron HuggingFa ce	`Python`,`MPI`, `DeepSpeed` (usang)
`option.mo del_dir`	Tidak	Jalur direktori untuk memuat model. Default diatur ke jalur saat ini dengan file model.	Default: `/opt/djl/ ml`

Item	Diperlukan	Deskripsi	Nilai contoh
		SageMakerAktif, ini diatur ke lokasi tempat SageMaker mengunduh objek model dari Amazon S3.	
`option.model_id`	Tidak	Nilai opsi ini adalah ID Wajah Pelukan model atau URL S3 dari artefak model. DJL Serving akan menggunakan ID untuk mengunduh model dari Hugging Face atau URL S3. DJL Serving menggunakan s5cmd untuk mengunduh model dari bucket, yang umumnya lebih cepat	`google/flan-t5-xl`, `s3://<my-bucket>/google/flan-t5-xl` Default: Tidak Ada
`option.dtype`	Tidak	Jenis data yang Anda rencanakan untuk mentransmisikan model. Defaultnya adalah fp16. Anda juga dapat mengatur ke bf16 jika Anda menggunakan G5, P4D dan mesin GPU yang lebih baru.	`fp16`,, `fp32bf16`, `int8` (hanya digunakan di LMI-dist)

Item	Diperlukan	Deskripsi	Nilai contoh
`option.tensor_parallel_degree`	Tidak	Jumlah GPU (nomor pengiris model) untuk membelah model. Jika Anda menggunakan LLM, Anda harus menetapkan nilai ini untuk mencapai kinerja terbaik. Jika Anda tidak tahu apa nilainya, mulailah mencoba dari “maks” (bagi model ke jumlah maksimum GPU pada mesin).	Standar untuk DeepSpeed, Transformer-NeuronX: 1 Default untuk HuggingFace Accelerate: -1 Standar untuk TrTllm Container: max

Item	Diperlukan	Deskripsi	Nilai contoh
`option.rolling_batch`	Tidak	Juga dikenal sebagai continuous batching. Memungkinkan batching tingkat iterasi menggunakan salah satu strategi yang didukung. Ini memungkinkan permintaan bersamaan yang tiba pada waktu yang berbeda untuk digabungkan sebagai batch untuk menjalankan inferensi dengan server model. Dinonaktifkan untuk DeepSpeed kontainer secara default mengingat ada banyak pilihan backend. Untuk wadah TensorRT, rolling batch diaktifkan secara default. Untuk TransformersNeuron X, ini dinonaktifkan secara default.	DeepSpeed wadah:`auto`,`scheduler`,`lmi-dist`,`vllm`, `deepspeed` Wadah neuron: `auto` Wadah TRTLLM: `trtllm`

Item	Diperlukan	Deskripsi	Nilai contoh
`option.max_rolling_batch_size`	Tidak	Permintaan atau batch bersamaan maksimum yang dapat diambil model. Server model akan memberi umpan ke proses python <max_rolling_batch_size>untuk mencegah GPU OOM. Pelanggan masih dapat mengirim lebih banyak permintaan ke server model. Permintaan lebih dari <max_rolling_batch_size>akan diantrian dan diumpankan ke Python sampai selesai. Catatan: Ini adalah konfigurasi khusus model. Jika Anda mengatur <max_rolling_batch_size>Model A, dan ada salinan N Model A di dalam wadah. Kemudian server model dapat menangani hingga <max_rolling_batch_size>permintaan x N.	Default: 32 (untuk semua mesin kecuali DeepSpeed) Default: 4 (untuk DeepSpeed)

Parameter lanjutan

Item	Diperlukan	Deskripsi	Nilai contoh
`option.trust_remote_code`	Tidak	Setel `true` untuk menggunakan model Hugging Face Hub dengan kode kustom. Kami menetapkan nilai default `false` untuk mencegah eksekusi kode berbahaya dari Hugging Face Hub.	Default: `false`
`option.revision`	Tidak	Gunakan versi tertentu atau hash komit dari model Hugging Face Hub	Default: Tidak Ada
`option.entryPoint`	Pilihan.EntryPoint	Mendefinisikan handler pemuatan model bawaan mana yang akan digunakan. Anda juga dapat menggunakan handler model kustom (model.py) di direktori model. Tentukan sebagai djl_python. <prebuilt handler>a tau jalur ke handler khusus Anda. DLC berbeda yang kami tawarkan memilih titik masuk untuk Anda. Misalnya,	`djl_python.deepspeed` , `djl_python.huggingface` , `djl_python.transformers_neuronx` , `djl_python.tensorrt_llm` , `djl_python.stable_diffusion`

Item	Diperlukan	Deskripsi	Nilai contoh
		jika Anda menggunakan TensorRTLLM DLC, ia menggunakan `djl_python.tensorrt_llm`	
`option.parallel_loading`	Tidak	Ini memuat pekerja secara paralel dan mengurangi waktu pemuatan model jika model Anda dapat masuk ke dalam memori CPU dengan beberapa proses. Catatan: Jika Anda memuat N salinan model pada saat yang sama, maka memori CPU puncak dapat naik ke N x model_size dan menyebabkan CPU OOM.	Default: `false`

Item	Diperlukan	Deskripsi	Nilai contoh
`option.model_loading_timeout`	Tidak	Menetapkan batas pada model waktu dapat memuat sebelum batas waktu server model. Waktu default adalah 30 menit (1800-an). <model_loading_timeout>Jika Anda menggunakan on SageMaker dan Anda berharap model membutuhkan waktu lebih lama untuk dimuat, Anda juga harus menyetel container_startup_health_check_timeout= aktif SageMaker	Default: `1800`
`job_queue_size`	Tidak	Menentukan ukuran antrian pekerjaan di tingkat model. Antrian pekerjaan biasanya digunakan untuk menangani permintaan bersamaan yang melampaui apa yang dapat diambil oleh server model backend. <job_queue_size>Server model akan mengantri permintaan hingga.	Default: `1000`

Item	Diperlukan	Deskripsi	Nilai contoh
`option.output_formatter`	Tidak	Gunakan hanya jika `option.rolling_batch` diaktifkan. Mendefinisikan format output mana server model akan dikirim sebagai hasilnya. Jika tidak ada nilai yang ditetapkan, kami mengirim kembali JSON dalam token.	`json`, `jsonlines` Default: `json`
`option.enable_streaming` (Usang sejak 0.25.0)	Tidak	Dari 0.25.0, kami mulai menghentikan parameter ini. Mengaktifkan streaming respons untuk batching statis. Gunakan `huggingface` untuk mengaktifkan output streaming HuggingFace -like. RollingBatch sudah melakukan streaming token secara default. Menyetel nilai ini untuk tidak RollingBatch berpengaruh.	`false`, `true`, `huggingface`

Parameter lanjutan: batching dinamis

Item	Diperlukan	Deskripsi	Nilai contoh
`batch_size`	Tidak	D ynamic-request-level batching. Ini biasanya digunakan dalam inferensi non-teks untuk menunggu permintaan datang untuk membangun batch. Ini membantu menangani permintaan bersamaan dengan lebih efisien. Batching dinamis tidak dapat digunakan dengan rolling batching mengingat algoritme batching berbeda yang ditentukan oleh server model.	Default: 1
`option.max_batch_delay`	Tidak	Penundaan maksimum untuk agregasi batch dalam milidetik. Kami akan menunggu <max_batch_delay>durasi milidetik untuk mengumpulkan permintaan hingga <batch_size>ukuran untuk dikirim ke server model.	Default: 100
`option.max_idle_time`	Tidak	Waktu idle maksimum dalam hitungan detik	Default: 60

Item	Diperlukan	Deskripsi	Nilai contoh
		sebelum thread pekerja diperkecil.	

Pengaturan handler Hugging Face

Jika Anda menggunakan handler Hugging Face yang disediakan oleh DJL Serving, Anda dapat mengonfigurasi opsi dalam tabel berikut di. `serving.properties`

Parameter umum

Item	Diperlukan	Deskripsi	Nilai contoh
`option.device_map`	Tidak	Memungkinkan pemasangan model di beberapa GPU.	`auto`, `balanced`, `balanced_low_0` , `sequential` Default: `auto` jika tp > 0 dan perangkat cuda tersedia
`option.low_cpu_mem_usage`	Tidak	Mengurangi penggunaan memori CPU saat memuat model. Kami menyarankan Anda mengatur ini ke`TRUE`.	`TRUE`
`option.quantize`	Tidak	Kuantisasi model dengan metode kuantisasi yang didukung.	`bitsandbytes4` , `bitsandbytes8` Default: Tidak Ada `bitsandbytes4` setara dengan `load_in_4bit` opsi.

Item	Diperlukan	Deskripsi	Nilai contoh
`option.task`	Tidak	Tugas yang digunakan dalam Hugging Face untuk jaringan pipa yang berbeda.	`text-generation`

Parameter lanjutan

Item	Diperlukan	Deskripsi	Nilai contoh
`option.decoding_strategy`	Tidak	Menentukan metode decoding untuk sampel, serakah dan pencarian kontrastif.	`sample`, `greedy`, `contrastive` Default: `greedy`
`option.disable_flash_attn`	Tidak	Digunakan untuk beralih antara menggunakan huggingface flash_attention atau tidak. Perhatikan bahwa perhatian flash huggingface dapat dipengaruhi oleh padding.	Default: `true`
`option.load_in_4bit` (Usang sejak 0.25.0)	Tidak	Menggunakan kuantisasi bitsandbytes. Didukung hanya pada model tertentu.	Telah usang. Gunakan `option.quantize=bitsandbytes4` sebagai gantinya. Default: `false`
`option.load_in_8bi`	Tidak	Menggunakan kuantisasi bitsandby	Telah usang. Gunakan `option.qu`

Item	Diperlukan	Deskripsi	Nilai contoh
`t` (Usang sejak 0.25.0)		tes. Didukung hanya pada model tertentu.	`antize=bi tsandbytes8` sebagai gantinya. Default: `false`
`option.ma x_sparsity`	Tidak	Digunakan dalam mekanisme ambang batas max-spars ity. Membatasi max_sparsity dalam urutan token yang disebabkan oleh padding.	`0.01`, `0.5` Default: `0.33`
`option.ma x_splits`	Tidak	Digunakan dalam mekanisme ambang batas max-spars ity. Membatasi jumlah maksimum pemisahan batch, di mana setiap split memiliki panggilan inferensi sendiri.	`1`, `5` Default: `3`

Pengaturan penangan LMI-dist

Jika Anda menggunakan LMI-Dist untuk opsi batch bergulir dengan DJL Serving, Anda dapat mengonfigurasi opsi dalam tabel berikut di. `serving.properties`

Parameter umum

Item	Diperlukan	Deskripsi	Nilai contoh
`option.quantize`	Tidak	Gunakan <option.q uantize>teknologi	`bitsandbytes8` , `gptq`

Item	Diperlukan	Deskripsi	Nilai contoh
		kuantisasi dengan model. gptq quantize membutuhkan pemuatan model gptq. bitsandbytes tidak digunakan lagi. Gunakan bitsandby tes8 sebagai gantinya. Kedua opsi itu sama.	Default: `None`

Parameter lanjutan

Item	Diperlukan	Deskripsi	Nilai contoh
`option.paged_attention`	Tidak	Gunakan PagedAtte ntion atau tidak. Default selalu digunakan. Nonaktifk an ini jika Anda berencana untuk menjalankan arsitektu r GPU G4 atau yang lebih lama	Default: `true`
`option.max_rolling_batch_prefill_tokens` (Usang sejak 0.25.0)	Tidak	Membatasi jumlah token untuk caching. Ini perlu disetel berdasarkan ukuran batch dan panjang urutan input untuk menghindari GPU OOM. Saat ini kami sedang menghitun	Default: Tidak Ada

Item	Diperlukan	Deskripsi	Nilai contoh
		g nilai terbaik untuk Anda. Dari 0.25.0, ini tidak lagi diperlukan. <max_rolling_batch _prefill_tokens>Ji ka Anda masih berencana untuk menghitung nilai ini secara manual, gunakan rumus ini: (max_input_tokens+ max_output_tokens) * batch_size =. Misalnya, 512 input, 512 output dan ukuran batch 16, akan menetapkan 16384. Menetapkan nilai ini tidak berarti Anda harus mengirim input 512 dan output 512. Anda masih dapat melakukan input 1024, output 512 dengan ukuran batch 10 untuk permintaa n Anda (16384/(1 024+512)) = 10.6	

Pengaturan handler VLLm

Jika Anda menggunakan Vllm untuk opsi batch bergulir dengan DJL Serving, Anda dapat mengonfigurasi opsi dalam tabel berikut di. `serving.properties`

Item	Diperlukan	Deskripsi	Nilai contoh
`option.quantize`	Tidak	Kuantisasi model dengan metode kuantisasi yang didukung	`awq` Default: Tidak Ada
`option.max_rolling_batch_prefill_tokens`	Tidak	Membatasi jumlah token untuk caching. Ini perlu disetel berdasarkan ukuran batch dan panjang urutan input untuk menghindari GPU OOM. Jika Anda tidak menyetel, VllM akan mencoba menemukan nomor yang bagus untuk disesuaikan. <max_rolling_batch_prefill_tokens>Jika Anda masih berencana untuk menghitung nilai ini secara manual, gunakan rumus ini: (max_input_tokens+max_output_tokens) * batch_size =. Misalnya, input 512, output 512 dan ukuran batch 16, akan menetapkan 16384. Menetapkan nilai ini tidak berarti Anda harus mengirim	Default: Tidak Ada

Item	Diperlukan	Deskripsi	Nilai contoh
		input 512 dan output 512. Anda masih dapat melakukan input 1024, output 512 dengan ukuran batch 10 untuk permintaa n Anda (16384/(1 024+512)) = 10.6	

DeepSpeed pengaturan handler

Jika Anda menggunakan DeepSpeed handler yang disediakan oleh DJL Serving, Anda dapat mengkonfigurasi opsi dalam tabel berikut di. `serving.properties`

Parameter umum

Item	Diperlukan	Deskripsi	Nilai contoh
`option.ma x_tokens`	Tidak	Jumlah total token (input dan output) yang DeepSpeed dapat digunakan. Jumlah token output adalah perbedaan antara jumlah total token dan jumlah token input. Secara default kami menetapkan nilai ke 1024. Jika Anda mencari generasi urutan panjang, Anda mungkin ingin menyetelnya ke nilai yang lebih tinggi,	`1024`

Item	Diperlukan	Deskripsi	Nilai contoh
		seperti 2048 atau 4096.	
`option.quantize`	Tidak	Tentukan opsi ini untuk mengukur model Anda menggunakan metode kuantisasi yang didukung di. DeepSpeed SmoothQuant adalah penawaran khusus kami untuk memberikan kuantisasi dengan kualitas yang lebih baik.	`dynamic_int8` , `smoothquant`
`option.task`	Tidak	Tugas yang digunakan dalam Hugging Face untuk jaringan pipa yang berbeda. Default-nya adalah `text-generation` .	`text-generation`

Parameter lanjutan

Item	Diperlukan	Deskripsi	Nilai contoh
`option.checkpoint`	Tidak	Jalur ke file pos pemeriksaan yang DeepSpeed kompatibel.	`ds_inference_checkpoint.json`

Item	Diperlukan	Deskripsi	Nilai contoh
`option.enable_cuda_graph`	Tidak	Mengaktifkan menangkap grafik CUDA dari forward pass untuk mempercepat.	`TRUE`
`option.low_cpu_mem_usage`	Tidak	Kurangi penggunaan memori CPU saat memuat model. Kami menyarankan Anda mengatur ini ke`TRUE`.	`TRUE`
`option.return_tuple`	Tidak	Apakah lapisan transformator perlu mengembalikan tupel atau tensor.	`FALSE`
`option.smoothquant_alpha`	Tidak	Jika smoothquant disediakan di option.quantize, Anda dapat memberikan nilai alfa ini. Jika tidak disediakan, DeepSpeed akan memilih satu untuk Anda	Setiap nilai float antara 0 dan 1

Item	Diperlukan	Deskripsi	Nilai contoh
`option.training_mp_size`	Tidak	Jika model dilatih DeepSpeed, ini menunjukkan derajat paralelisme tensor yang dengannya model dilatih. Bisa berbeda dari derajat paralel tensor yang diinginkan untuk inferensi.	2
`option.triangular_masking`	Tidak	Apakah akan menggunakan masking segitiga untuk masker perhatian. Ini adalah aplikasi atau model khusus.	`FALSE`

FasterTransformer pengaturan handler

Jika Anda menggunakan FasterTransformer handler yang disediakan oleh DJL Serving, Anda dapat mengkonfigurasi opsi dalam tabel berikut di. `serving.properties`

Item	Diperlukan	Deskripsi	Nilai contoh
`option.model_dir`	Tidak	Jalur direktori untuk memuat model. Default diatur ke jalur saat ini dengan file model.	Default: `/opt/djl/ml`
`option.model_id`	Tidak	Ini mengesampingkan `option.model_dir` jika	`google/flan-t5-xl`, `s3://my-b`

Item	Diperlukan	Deskripsi	Nilai contoh
		disetel. Digunakan untuk mengunduh model dari Hugging Face atau ember S3	`ucket/google/flan-t5-xl` Default: Tidak Ada

Pengaturan penangan Neuronx LMI

Jika Anda menggunakan handler Neuronx LMI yang disediakan oleh DJL Serving, Anda dapat mengonfigurasi opsi dalam tabel berikut di. `serving.properties`

Parameter umum

Item	Diperlukan	Deskripsi	Nilai contoh
`option.load_in_8bit`	Tidak	Tentukan opsi ini untuk mengukur model Anda menggunakan metode kuantisasi yang didukung di X. TransformerNeuron	`True`, `False` Default: `False`
`option.n_positions`	Tidak	Input maksimum ditambah panjang token keluaran per permintaan. Anda mungkin ingin menetapkan nilai ini lebih besar jika Anda berencana untuk melakukan inferensi dengan token input atau output yang panjang.	Default: `128`

Parameter lanjutan

Item	Diperlukan	Deskripsi	Nilai contoh
`option.context_length_esitmate`	Tidak	Perkiraan panjang input konteks untuk model Llama. Pelanggan dapat menentukan ukuran bucket yang berbeda untuk meningkatkan penggunaan kembali cache KV. Ini akan membantu meningkatkan latensi	Default: `None`
`option.load_split_model`	Tidak	Beralih ke `True` saat menggunakan artefak model yang telah dipecah untuk kompilasi atau pemuatan neuron.	Default: `False`
`option.low_cpu_mem_usage`	Tidak	Kurangi penggunaan memori CPU saat memuat model.	Default: `false`
`option.neuron_optimize_level`	Tidak	Tingkat pengoptimalan kompiler runtime neuron, menentukan jenis pengoptimalan yang diterapkan selama kompilasi. Semakin tinggi tingkat optimasi, semakin banyak waktu yang dihabiskan untuk kompilasi. Namun	1, 2, 3 Default: 2

Item	Diperlukan	Deskripsi	Nilai contoh
		sebagai gantinya, Anda mendapatkan latensi dan throughput yang lebih baik.	
`option.unroll`	Tidak	Buka gulungan grafik model untuk kompilasi . Dengan`unroll=None` , kompiler akan memiliki lebih banyak peluang mengoptimalkan seluruh lapisan.	Default: `None`

Pengaturan penangan Tensorrt-LLM

Jika Anda menggunakan handler TensorRT-LLM yang disediakan oleh DJL Serving, Anda dapat mengonfigurasi opsi dalam tabel berikut di. `serving.properties`

Parameter umum

Item	Diperlukan	Deskripsi	Nilai contoh
`option.max_input_len`	Tidak	Ukuran token input maksimum yang Anda harapkan untuk dimiliki model per permintaan. Ini adalah parameter kompilasi yang disetel ke model ke just-in-time kompilasi. Jika Anda menetapkan nilai ini terlalu rendah, model tidak akan	Nilai default untuk Llama adalah 512 Nilai default untuk Falcon adalah 1024

Item	Diperlukan	Deskripsi	Nilai contoh
		dapat mengkonsumsi input panjang.	
`option.max_output_len`	Tidak	Ukuran token keluaran maksimum yang Anda harapkan dimiliki model per permintaan. Ini adalah parameter kompilasi yang mengatur model ke just-in-time kompilasi. Jika Anda menetapkan nilai ini terlalu rendah, model tidak akan dapat menghasilkan token di luar nilai yang Anda tetapkan.	Nilai default untuk Llama adalah 512 Nilai default untuk Falcon adalah 1024
`option.use_custom_all_reduce`	Tidak	Gunakan kernel kustom all reduce untuk GPU yang mengaktifkan NVLink. Ini dapat membantu mempercepat kecepatan inferensi model dengan komunikasi yang lebih baik. Nyalakan ini dengan menyetel true pada P4D, P4De, P5, dan GPU lain yang terhubung dengan NVLink.	Default: `false`

Parameter lanjutan

Item	Diperlukan	Deskripsi	Nilai contoh
`option.tokens_per_block`	Tidak	Token per blok untuk digunakan dalam algoritma perhatian halaman.	Default: 64
`option.batch_scheduler_policy`	Tidak	Kebijakan penjadwal pengelola batch TensorRT-LLM.	`max_utilization` , `guaranteed_no_evict` Default: `max_utilization`
`option.kv_cache_free_gpu_mem_fraction`	Tidak	Fraksi memori GPU gratis yang dialokasikan untuk cache kv. Semakin besar nilai yang Anda tetapkan, semakin banyak memori yang akan dicoba diambil alih oleh model pada GPU. Semakin banyak memori yang diawetkan, semakin besar ukuran Cache KV yang dapat kita gunakan dan itu berarti urutan input +output yang lebih lama atau ukuran batch yang lebih besar.	Angka mengambang antara 0 dan 1. Defaultnya adalah 0,95

Item	Diperlukan	Deskripsi	Nilai contoh
`option.max_num_sequences`	Tidak	Jumlah maksimum permintaan input yang diproses dalam batch. Kami akan menerapkan max_rolling_batch_size sebagai nilai untuk itu jika Anda tidak mengatur ini. Umumnya Anda tidak perlu menyentuhnya kecuali Anda benar-benar ingin model dikompilasi ke ukuran batch yang tidak sama dengan set server model.	Integer lebih besar dari 0 Nilai default adalah ukuran batch yang ditetapkan saat membangun mesin TensorRT
`option.enable_trt_overlap`	Tidak	Parameter untuk tumpang tindih eksekusi batch permintaan. Ini mungkin berdampak negatif pada kinerja ketika jumlah permintaan terlalu kecil. Selama percobaan kami, kami melihat lebih banyak dampak negatif untuk mengaktifkannya daripada menonaktifkannya.	`True`, `False`. Default: `False`

Parameter lanjutan: kuantisasi

Item	Diperlukan	Deskripsi	Nilai contoh
`option.quantize`	Tidak	Saat ini hanya mendukung `smoothquant` untuk model Llama dengan mode just-in-time kompilasi.	`smoothquant`
`option.smoothquant_alpha`	Tidak	Parameter alfa smoothquant.	Default: 0.8
`option.smoothquant_per_token`	Tidak	Ini hanya diterapkan ketika `option.quantize` diatur kesmoothquant . Ini memungkinkan memilih, pada waktu berjalan, faktor penskalaan smoothquant khusus untuk setiap token. Ini biasanya sedikit lebih lambat, tetapi lebih akurat	`True`, `False`. Bawaan: `False`.
`option.smoothquant_per_channel`	Tidak	Ini hanya diterapkan ketika `option.quantize` diatur kesmoothquant . Ini memungkinkan memilih, pada waktu berjalan, faktor penskalaan smoothquant khusus	`True`, `False`. Bawaan: `False`.

Item	Diperlukan	Deskripsi	Nilai contoh
		untuk setiap token. Ini biasanya sedikit lebih lambat, tetapi lebih akurat	
`option.multi_query_mode`	Tidak	Ini hanya diperlukan ketika `option.quantize` diatur `kesmoothquant`. Ini harus diatur untuk model yang mendukung multi-query-attention, seperti llama-70b.	`True`, `False`. Bawaan: `False`.

Memilih jenis instans untuk inferensi model besar

Saat menerapkan model deep learning, kami biasanya menyeimbangkan biaya hosting model ini dengan kinerja dalam hal latensi, throughput, dan akurasi. Input inti untuk persamaan ini adalah jenis SageMaker instance. SageMakermenawarkan banyak jenis instans dengan perangkat GPU yang berbeda. Untuk model tertentu, ada kemungkinan beberapa instance yang cocok untuk hosting model untuk inferensi. Benchmarking dapat membantu Anda untuk memutuskan dengan contoh mana untuk melanjutkan.

Bagian berikut memberikan beberapa panduan yang dapat Anda ikuti untuk menentukan jenis instans mana yang akan dipilih untuk hosting model besar. Untuk menggunakan panduan ini, Anda harus mengetahui karakteristik berikut dari kasus penggunaan Anda:

- Arsitektur atau tipe model, seperti OPT, GPTJ, BLOOM, atau Bert
- Presisi tipe data, sepertifp32,fp16,bf16, atau `int8`
- Ukuran model dalam MB atau GB
- Ukuran token input dan output

Menentukan jenis instans yang mungkin

Saat memilih jenis instans, pertimbangkan ukuran model serta perangkat GPU yang tersedia. Container model inference (LMI) besar saat ini hanya didukung pada instans dengan GPU NVIDIA, dan tidak didukung pada instans Graviton. Untuk up-to-date informasi selengkapnya tentang instans GPU yang tersedia, lihat Instans GPU yang Disarankan.

Saat menerapkan model besar, situasi yang ideal adalah menyesuaikan model pada satu GPU. Ini adalah pilihan terbaik sehubungan dengan kinerja karena menghilangkan overhead komunikasi antara perangkat GPU. Untuk beberapa model, tidak mungkin menyesuaikannya dengan satu GPU karena ukuran model. Untuk model lain, mereka mungkin cocok pada satu GPU, tetapi mungkin lebih hemat biaya untuk mempartisi model di beberapa GPU yang lebih murah.

Bagian berikut menunjukkan bagaimana Anda dapat mengembangkan metrik untuk memfilter daftar GPU yang tersedia ke yang mungkin berfungsi untuk kasus penggunaan Anda.

Menentukan jenis instans yang mungkin berdasarkan tipe data

GPU yang tersedia SageMaker berbeda dalam dukungan asli mereka dari tipe data. Jika Anda berencana untuk menerapkan model`bf16`, pilih instans dengan perangkat GPU yang mendukung kemampuan komputasi 7.5+. Jika Anda berencana untuk menerapkan model`int8`, kami sangat menyarankan Anda memilih instans dengan perangkat GPU yang mendukung kemampuan komputasi 7.5+ karena GPU ini mengandung inti tensor. `int8` Namun, Anda masih dapat menggunakan HuggingFace Accelerate (`int8`menggunakan kuantisasi bitsandbytes) atau DeepSpeed (menggunakan ZeroQuant kuantisasi) pada kemampuan komputasi yang lebih rendah. Anda harus mengharapkan kinerja yang lebih rendah jika Anda pergi rute ini dibandingkan dengan menggunakan GPU dengan dukungan asli untuk `int8` matematika.

Untuk memverifikasi kemampuan komputasi GPU, lihat Kemampuan Komputasi GPU Anda di situs web NVIDIA.

Memperkirakan batas bawah untuk memori yang diperlukan untuk meng-host model

Untuk lebih memfilter daftar GPU, tentukan seberapa besar model Anda untuk presisi tipe data yang Anda inginkan untuk meng-host model. Jika ukuran model dapat muat pada satu GPU, dan latensi rendah adalah prioritas tertinggi, sebaiknya pilih instance dengan GPU yang memiliki memori yang cukup untuk meng-host model ini. Anda dapat memperkirakan batas bawah untuk memori yang diperlukan untuk meng-host model Anda berdasarkan jumlah parameter dalam model Anda dan tipe data. Ini lebih rendah estimasi memori terikat (LBME) mewakili memori minimum telanjang

dalam byte yang diperlukan untuk memuat parameter model ke dalam memori GPU. Hitung LBME menggunakan persamaan berikut:

- `int8` – `LBME = number of parameters`
- `fp16` dan `bf16` – `LBME = 2 × number of parameters`
- `fp32` – `LBME = 4 × number of parameters`

Menentukan kemungkinan contoh berdasarkan jumlah partisi

Menentukan tingkat partisi yang akan digunakan dengan model Anda bermuara pada faktor-faktor berikut:

- Ukuran model
- Biaya yang mau Anda bayarkan untuk instans
- Ketersediaan contoh tertentu
- Persyaratan latensi Anda

Misalnya, model Eleutherai/GPT-Neox-20B membutuhkan waktu sekitar 45 GB untuk dihosting. `fp16` Anda dapat menerapkan model ini menggunakan `p4de.24xlarge` instans tanpa berbagi karena memori GPU yang tersedia per perangkat adalah 80 GB. Ini adalah satu-satunya GPU saat ini AWS yang akan mendukung pemasangan model ini sepenuhnya pada perangkat. Dengan 8GPU pada contoh itu, Anda dapat meng-host 8 salinan model. Atau Anda juga dapat menerapkan model ini dengan partisi 2 arah pada `g5.12xlarge` Dengan 4 GPU, Anda dapat meng-host 2 salinan model. Menggunakan 4 `g5.12xlarge` instans untuk meng-host 8 salinan model ini dibandingkan dengan 1 `p4de.24xlarge` instans mendekati setengah biaya (meskipun sisa memori GPU pada `p4de.24xlarge` mendukung ukuran batch yang lebih besar). Sementara kinerja cenderung lebih rendah pada`g5.12xlarge`, mungkin lebih masuk akal dari perspektif biaya.

Beberapa contoh mungkin tidak tersedia di tertentu. Wilayah AWS Anda dapat memeriksa ketersediaan instans dengan Wilayah/Availability Zone menggunakan: AWS CLI

```
aws ec2 describe-instance-type-offerings --location-type "availability-zone" --filters
 Name=location,Values=us-east-2a --region us-east-2 --query "InstanceTypeOfferings[*].
[InstanceType]" --output text | sort
```

Lebih sedikit partisi umumnya menghasilkan latensi keseluruhan yang lebih rendah. Kinerja terbaik akan datang dari satu GPU run, tetapi kami sangat menyarankan Anda bereksperimen dengan partisi untuk memahami dampak latensi untuk model spesifik Anda.

Anda dapat mempersempit daftar kemungkinan instance dengan mempertimbangkan LBME yang Anda hitung pada langkah sebelumnya, dan jumlah partisi yang Anda inginkan. Jika memungkinkan, pertimbangkan beberapa nilai partisi untuk menjaga banyak jenis instance dalam pertimbangan. Misalnya, jika LBME adalah 30 GB, beberapa perkiraan untuk tingkat partisi yang berbeda adalah:

- Penyebaran GPU tunggal aktif `p4d.24xlarge` (40 GB per GPU) atau `p4de.24xlarge` (80 GB per GPU)
- 2 penyebaran GPU aktif `p3.8xlarge` (16 GB per GPU) atau `g5.12xlarge` (24 GB per GPU)

Memilih mesin

LMI deep learning container (DLC) memberikan dukungan untukDeepSpeed, FasterTransformer dan HuggingFace Mempercepat backend. Semua kerangka kerja ini dapat digunakan untuk menyebarkan dan meng-host model besar yang dipartisi di beberapa GPU. Anda biasanya dapat mengharapkan kinerja yang lebih tinggi (latensi lebih rendah atau throughput yang lebih tinggi) dengan DeepSpeed atauFasterTransformer, tetapi mesin ini tidak menawarkan inferensi yang dioptimalkan untuk semua arsitektur model. Semua kerangka kerja juga menerapkan model paralelisme secara berbeda. DeepSpeeddan FasterTransformer menggunakan paralelisme tensor, yang biasanya lebih berkinerja dengan biaya penggunaan memori GPU yang lebih tinggi. HuggingFaceAccelerate menggunakan paralelisme pipa yang menggunakan lebih sedikit memori tetapi juga kurang berkinerja.

Kami menyarankan Anda menggunakan DeepSpeed atau FasterTransformer bila memungkinkan karena kernel CUDA yang menyatu secara signifikan meningkatkan kinerja dibandingkan dengan menggunakan HuggingFace Accelerate. DeepSpeedsaat ini menawarkan implementasi kernel gabungan untuk arsitektur model berikut:

- Bert
- DistilBert
- GPT Neo, GPT Neo X, GPT2
- MEMILIH
- BERKEMBANG
- Megatron

- Difusi Stabil

FasterTransformermenawarkan implementasi kernel gabungan untuk arsitektur model berikut:

- GPT2
- MEMILIH
- BERKEMBANG
- T5

Untuk arsitektur model lain tidak ada perbedaan yang signifikan. Kami menyarankan Anda bereksperimen dengan keduanya DeepSpeed dan HuggingFace Accelerate untuk menentukan mesin mana yang paling cocok untuk model Anda. Untuk melihat model mana yang FasterTransformer mendukung, lihat matriks FasterTransformer dukungan.

Untuk memilih jenis instans, sesuaikan LBME Anda untuk mesin tertentu. Memperkirakan penggunaan GPU yang diharapkan juga berbeda antaraDeepSpeed, FasterTransformer dan HuggingFace Accelerate karena perbedaan dalam cara mereka menjalankan paralelisme model.

DeepSpeeddan FasterTransformer menggunakan paralelisme tensor. Beberapa modul, seperti embeddings, tidak mendukung paralelisme tensor. Modul-modul ini direplikasi di semua GPU. HuggingFaceMempercepat menggunakan partisi pipa naif, yang tidak menghasilkan modul direplikasi. Dengan demikian dengan DeepSpeed atau FasterTransformer Anda biasanya dapat mengharapkan kebutuhan memori meningkat dengan tingkat paralelisme tensor.

Bagian berikut menunjukkan bagaimana Anda dapat menyesuaikan LBME Anda untuk mendapatkan perkiraan memori model yang dimuat (LMME), yang merupakan perkiraan memori yang diperlukan per GPU untuk memuat model Anda.

HuggingFaceMempercepat

Dengan HuggingFace Accelerate, partisi pipa tidak secara signifikan meningkatkan memori yang dibutuhkan untuk memuat model. Tidak ada sub-modul yang perlu diduplikasi di seluruh GPU.

- Untuk penyebaran GPU tunggal,. `LMME = 1.15 × LBME`
- Untuk beberapa penyebaran GPU,. `LMME = (1.15 + 0.035 × number of partitions) × LBME` Di sini jumlah partisi diasumsikan kelipatan 2.

DeepSpeedatau FasterTransformer

Dengan DeepSpeed atauFasterTransformer, partisi parallel tensor menghasilkan beberapa sub-modul model yang diduplikasi karena tidak setiap modul mendukung paralelisme tensor. Tingkat replikasi modul sangat tergantung pada arsitektur model.

- Untuk penyebaran GPU tunggal,. `LMME = 1.20 × LBME`
- Untuk beberapa penyebaran GPU,`LMME = multiplier × LBME`, di mana pengganda sangat bergantung pada arsitektur model, dan tipe data. Untuk beberapa arsitektur populer, kami menyajikan pengganda yang direkomendasikan dalam tabel berikut:

Arsitektur model	**fp32**	**fp16**	**int8**
Varian GPT	1.45	1,55	1,55
Varian	1,5	1,55	1,55
Varian OPT	1,55	1,75	1,75
Model lainnya	1,5	1,5	1,5

Menyesuaikan estimasi memori model yang dimuat untuk panjang urutan dan ukuran batch

Selama inferensi, lebih banyak memori diperlukan untuk tensor input, tensor menengah, dan tensor keluaran, daripada yang disarankan LMME Anda. Anda juga perlu memperhitungkan memori tambahan yang diperlukan sebagai hasil dari partisi model.

Perkiraan memori model runtime (RMME), yang didefinisikan dalam persamaan berikut, menyumbang persyaratan memori tambahan berdasarkan panjang urutan (jumlah total token input dan output):

```
RMME per GPU = multiplier × LMME / number of partitions
```

Pengganda yang direkomendasikan yang tercantum di bawah ini didasarkan pada panjang urutan 1024. Jika Anda berencana untuk menggunakan urutan yang lebih pendek atau lebih panjang, Anda dapat lebih menyesuaikan pengganda ini dengan mengalikannya. `sequence length / 1024`

- Varian GPT - `1.1 + 0.05 × (number of partitions - 1)`

- Varian BLOOM - `1.15 + 0.05 × (number of partitions - 1)`
- varian OPT - `1.20 + 0.1 × (number of partitions - 1)`

Jika Anda berencana untuk mengakomodasi ukuran batch yang lebih besar dari 1, kalikan pengganda yang disebutkan di atas dengan ukuran batch yang Anda harapkan, untuk mendapatkan pengganda akhir untuk menghitung RMME per GPU.

Menyelesaikan kemungkinan jenis instans

Dengan RMME per GPU Anda dapat memfilter daftar GPU yang tersedia ke GPU yang mungkin berfungsi untuk kasus penggunaan Anda. RMME per GPU adalah metrik yang sengaja berhati-hati untuk memperhitungkan varians kinerja antara arsitektur model yang berbeda dengan tipe data dan versi kerangka kerja mesin yang berbeda. Jika Anda menemukan bahwa setiap instans hampir tidak dikecualikan oleh perkiraan Anda, kami sarankan untuk mencobanya.

Menyebarkan model yang tidak terkompresi

Saat menerapkan model ML, salah satu opsi adalah mengarsipkan dan mengompres artefak model ke dalam format. `tar.gz` Meskipun metode ini bekerja dengan baik untuk model kecil, mengompresi artefak model besar dengan ratusan miliar parameter dan kemudian mendekompresi pada titik akhir dapat memakan banyak waktu. Untuk inferensi model besar, kami menyarankan Anda menerapkan model yang tidak terkompresi. Panduan ini menunjukkan bagaimana Anda dapat menerapkan model yang tidak terkompresi.

Untuk menerapkan model ML-terkompresi, unggah semua artefak model ke Amazon S3 dan atur di bawah awalan Amazon S3 yang umum. Awalan Amazon S3 adalah serangkaian karakter di awal nama kunci objek Amazon S3, dipisahkan dari sisa nama oleh pembatas. Untuk informasi selengkapnya tentang awalan Amazon S3, lihat Mengatur objek menggunakan awalan.

Untuk menyebarkan denganSageMaker, Anda harus menggunakan garis miring (/) sebagai pembatas. Anda harus memastikan bahwa hanya artefak yang terkait dengan model ML-mu yang diatur dengan awalan. Untuk model ML dengan satu artefak tak terkompresi, awalan akan identik dengan nama kunci. Anda dapat memeriksa objek mana yang terkait dengan awalan Anda dengan: AWS CLI

```
aws s3 ls --recursive s3://bucket/prefix
```

Setelah mengunggah artefak model ke Amazon S3 dan mengaturnya di bawah awalan umum, Anda dapat menentukan lokasinya sebagai bagian dari ModelDataSourcebidang saat Anda memanggil permintaan. CreateModel SageMakerakan secara otomatis men-download artefak model terkompresi untuk `/opt/ml/model` untuk inferensi. Untuk informasi lebih lanjut tentang aturan yang SageMaker saat mengunduh artefak, lihat ModelDataSourceS3.

Cuplikan kode berikut menunjukkan bagaimana Anda dapat memanggil `CreateModel` API saat menerapkan model yang tidak terkompresi. Ganti *`teks pengguna yang dicetak miring dengan informasi`* Anda sendiri.

```
model_name = "model-name"
sagemaker_role = "arn:aws:iam::123456789012:role/SageMakerExecutionRole"
container = "123456789012.dkr.ecr.us-west-2.amazonaws.com/inference-image:latest"

create_model_response = sagemaker_client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = sagemaker_role,
    PrimaryContainer = {
        "Image": container,
        "ModelDataSource": {
            "S3DataSource": {
                "S3Uri": "s3://my-bucket/prefix/to/model/data/",
                "S3DataType": "S3Prefix",
                "CompressionType": "None",
            },
        },
    },
)
```

Contoh yang disebutkan di atas mengasumsikan bahwa artefak model Anda diatur di bawah awalan umum. Jika artefak model Anda adalah satu objek Amazon S3 yang tidak terkompresi, maka ubah `"S3Uri"` untuk menunjuk ke objek Amazon S3, dan ubah ke. `"S3DataType"` `"S3Object"`

> **Note**
>
> Saat ini Anda tidak dapat menggunakan `ModelDataSource` denganAWS Marketplace, SageMaker batch transform, titik akhir Inferensi SageMaker Tanpa Server, dan titik akhir multi-model. SageMaker

FAQ inferensi model besar

Lihat item FAQ berikut untuk jawaban atas pertanyaan umum tentang inferensi model besar (LMI) dengan. SageMaker

T: Kapan saya harus menggunakan LMI DLC?

J: Wadah pembelajaran mendalam inferensi model besar (LMI DLC) mencakup versi yang diuji dari paralelisasi model populer dan pustaka pengoptimalan untuk model hosting dengan puluhan atau ratusan miliar parameter. Anda harus menggunakan DLC LMI jika model pembelajaran mendalam Anda memerlukan beberapa akselerator untuk hosting atau jika Anda ingin mempercepat inferensi dengan model populer yang didukung seperti GPT, Bloom, dan OPT.

T: Siapa yang dapat saya hubungi jika ada yang tidak berhasil?

J: Jika, setelah meninjau dokumentasi, Anda terus mengalami masalah dalam menghosting model besar Anda, hubungiAWS Support.

T: Dapatkah saya menggunakan wadah pembelajaran mendalam inferensi model besar (LMI DLC) di luar? SageMaker

J: DLC AWS LMI telah diuji dan dirancang untuk SageMaker, tetapi dapat digunakan di Amazon EC2 dengan tipe instans yang didukung.

T: Format model apa yang dapat saya gunakan dengan DLC LMI?

AWSLMI DLC menyertakan fungsi kenyamanan untuk memuat format model Hugging Face. Anda dapat membawa model dalam format lain, seperti pos pemeriksaan Megatron, tetapi Anda perlu menulis logika untuk memuat format itu atau mengonversi ke format Hugging Face.

T: Bagaimana saya bisa beralih di antara pustaka paralelisasi model seperti Hugging Face Accelerate dan? DeepSpeed

Jika Anda menggunakan handler DJL (seperti dalam contoh tutorial), kemudian beralih mesin default dan handler dalam file. `serving.properties` Misalnya, ubah `option.entryPoint` pengaturan `engine` dan untuk beralih dari DeepSpeed ke Hugging Face. Secara umum, Anda dapat mengganti mesin dan menggunakan derajat paralel tensor yang sama.

Pemecahan masalah inferensi model besar

Jika Anda mengalami masalah atau kesalahan, Anda dapat mencoba menggunakan daftar berikut untuk memecahkan masalah. Jika masalah berlanjut, hubungi kami di LMI-DLC-feedback@amazon.com.

Topik

Mengunduh model ke instance saya membutuhkan waktu lama

Pertimbangkan untuk mengunggah artefak model Anda ke Amazon S3 dan sebutkan objek S3 `option.model_id` di file konfigurasi Anda. `serving.properties` Metode pengunduhan ini menggunakan `s5cmd` dan mempercepat pengunduhan data. Pastikan wadah memiliki izin untuk mengakses bucket S3 yang ditentukan.

Latensi inferensi atau kinerja throughput buruk

Bahkan dengan paralelisme model, model besar masih dapat memiliki kinerja yang buruk seperti latensi tinggi atau throughput rendah. Pertama, jika model Anda didukung, pertimbangkan untuk menggunakan DeepSpeed mesin yang menyertakan kernel yang dioptimalkan untuk inferensi. Ini dapat meningkatkan kinerja hingga 3 kali. Selanjutnya, pertimbangkan untuk menggunakan jenis instance yang lebih kuat, seperti `p4d.24xlarge` instance. Akhirnya, pertimbangkan untuk menggunakan model yang lebih kecil. Untuk mencapai kinerja terbaik dan biaya terendah, cobalah untuk menemukan model terkecil yang memenuhi bilah akurasi untuk kasus penggunaan Anda, karena lebih banyak parameter dapat menghasilkan latensi yang lebih tinggi, throughput yang lebih rendah, atau biaya yang lebih tinggi. Pertimbangkan untuk menyempurnakan model yang lebih kecil untuk meningkatkan akurasi.

SageMaker titik akhir gagal memulai dengan kesalahan batas waktu atau kehabisan memori

Model besar dapat membutuhkan lebih banyak waktu untuk memuat ke memori. Anda harus menyesuaikan konfigurasi titik akhir default untuk menangani waktu tambahan ini. Jika titik akhir gagal dalam pemeriksaan kesehatan atau kehabisan memori, periksa apakah Anda

telah mengonfigurasi titik akhir Anda dengan tepat. Misalnya, jika Amazon CloudWatch Logs menunjukkan batas waktu pemeriksaan kesehatan, Anda harus meningkatkan `ContainerStartupHealthCheckTimeoutInSeconds` parameter `ProductionVariants` selama `create_endpoint_config` langkah hosting aktif. SageMaker

Untuk informasi selengkapnya tentang parameter ini, lihat SageMaker parameter titik akhir untuk inferensi model besar.

Catatan rilis untuk wadah pembelajaran mendalam inferensi model besar

Anda dapat menemukan daftar wadah pembelajaran mendalam (DLC) yang didukung oleh SageMaker pada repositori AWS GitHub DLC. Di bawah ini adalah catatan rilis terkait.

Topik

Catatan Rilis LMI DLC: 27 November 2023 (1)

URI kontainer

- 763104351884.dkr. ecr.us-east-1.amazonaws.com /djl-inference:0.25.0-deepspeed0.11.0-cu11

Sorotan

- Menambahkan dukungan batching tingkat iterasi ke backend DeepSpeed
- Ditingkatkan ke vllm 2.0 ke lmi-dist backend
- Menambahkan fitur pemanasan dinamis ke backend lmi-dist

Pembaruan versi

- DJL Melayani 0.25.0
- DeepSpeed 0.11.0
- HuggingFace Mempercepat 0.23.0
- HuggingFace Trafo 4.34.0

Arsitektur model yang didukung

- Model didukung oleh DeepSpeed 0.11.0 menggunakan mesin DeepSpeed
- Model yang didukung oleh HuggingFace Accelerate menggunakan mesin Accelerate
- Dukungan batching tingkat iterasi untuk arsitektur model GPT, lLAMA, lLAMA 2, Falcon, dan T5 menggunakan mesin MPI
- Dukungan Paged Attention dan Flash Attention untuk arsitektur model GPT, Llama, Llama 2, Falcon dan T5 menggunakan mesin MPI

Masalah yang diketahui

- Saat menggunakan lmi-dist pada instance p5 di SageMaker Amazon, nonaktifkan aiccl dengan menggunakan `ENV USE_AICCL_BACKEND=false` variabel lingkungan untuk menghindari server model mogok saat pemuatan model.

Catatan Rilis LMI DLC: 27 November 2023 (2)

URI kontainer

- 763104351884.dkr. ecr.us-east-1.amazonaws.com /djl-inferensi:0.25.0-tensorrtllm0.5.0-cu122

Sorotan

- Mendukung pengelompokan permintaan tingkat iterasi/dalam pesawat
- Mendukung streaming respons
- Menambahkan dukungan untuk kompilasi model on-the-fly TensorRT-LLM sehingga pengguna hanya dapat memberikan id model Hugging Face untuk model Llama dan Falcon

- Menerima repositori model yang dibuat pengguna dengan artefak TensorRT-LLM yang dikompilasi untuk pemuatan yang lebih cepat

Pembaruan versi

- DJL Melayani 0.25.0
- Tensorrt-LLM 0.5.0
- PyTorch 2.0.1
- Python 3.10
- CUDA 12.2

Arsitektur model yang didukung

- Model LLAMA dan Falcon dengan dukungan untuk on-the-fly kompilasi model
- Kami belum menguji jenis model lainnya. Jika Anda ingin menggunakan selain model Llama dan Falcon, mereka harus dikompilasi sebelumnya dan diuji sebelum produksi.

Masalah yang diketahui

- Menggunakan parameter `repetition_penalty` sampling secara acak merusak program. Disarankan untuk tidak menggunakan `repetition_penalty` dalam versi ini.

Catatan Rilis LMI DLC: 27 November 2023 (3)

URI kontainer

- 763104351884.dkr. ecr.us-east-1.amazonaws.com /djl-inferensi:0.25.0-neuronx-sdk2.15.0

Sorotan

- Dukungan model yang dikompilasi sebelumnya

Pembaruan versi

- DJL Melayani 0.25.0
- NeuronSDK 2.15.0

- HuggingFace Mempercepat 0.23.0
- HuggingFace Trafo 4.34.0

Arsitektur model yang didukung

- GPT-2
- GPT-J
- GPT-Neox
- MEMILIH
- Mekar
- Llama
- Llama-2 (7b dan 13b)

Catatan Rilis LMI DLC: 4 Agustus 2023 (1)

URI kontainer

- `763104351884.dkr.ecr.region.amazonaws.com/djl-inference:0.23.0-fastertransformer5.3.0-cu118`

Sorotan

- Menambahkan dukungan untuk melayani model GPT-Neox.
- Wadah baru mendukung streaming respons.
- FasterTransformer handler diperbarui untuk mendukung PEFT dan LORA.
- DjlServing menyediakan integrasi ke backend inferensi Triton melalui java-cpp API.

Pembaruan versi

- DJL Melayani 0.23.0
- PyTorch 2.0.1
- FasterTransformer 5.3.0
- Akselerasi Hugging Face 0.20.3
- Trafo Hugging Face 4.30.1

- Diffuser Wajah Memeluk 0.12.0
- Python 3.9
- CUDA 11.8

Arsitektur model yang didukung

- GPT
- GPT-J
- GPT-Neox
- T5
- MEMILIH
- Mekar

Catatan Rilis LMI DLC: 4 Agustus 2023 (2)

URI kontainer

- `763104351884.dkr.ecr.region.amazonaws.com/djl-inference:0.23.0-deepspeed0.9.5-cu118`

Sorotan

- Upgrade kontainer DeepSpeed DLC ke 0.9.5. DeepSpeed
- Perbaikan dilakukan untuk mengaktifkan penghematan memori 2-3x melalui oss. DeepSpeed
- Menambahkan dukungan PyTorch -2.0.1, PEFT, dan LORA ke penangan default. DeepSpeed Rilis baru ini DeepSpeed menambahkan dukungan untuk model lLama v1.
- Menambahkan dukungan batching tingkat iterasi ke backend Hugging Face untuk arsitektur model GPT, Llama, Falcon, dan T5. Ini meningkatkan throughput penyajian model.

Pembaruan versi

- DJL Melayani 0.23.0
- DeepSpeed 0.9.5
- Akselerasi Hugging Face 0.20.3

- Trafo Hugging Face 4.30.2
- Python 3.9
- CUDA 11.8

Arsitektur model yang didukung

- Model didukung oleh DeepSpeed 0.9.5 menggunakan mesin DeepSpeed
- Model yang didukung oleh HuggingFace Accelerate menggunakan mesin Accelerate
- Dukungan batching tingkat iterasi untuk arsitektur model GPT, lLAMA, lLAMA 2, Falcon, dan T5 menggunakan mesin MPI
- Dukungan Paged Attention dan Flash Attention untuk arsitektur model GPT, Llama, Llama 2, Falcon dan T5 menggunakan mesin MPI

Catatan Rilis LMI DLC: 4 Agustus 2023 (3)

URI kontainer

- `763104351884.dkr.ecr.region.amazonaws.com/djl-inference:0.23.0-neuronx-sdk2.12.0`

Sorotan

- [Eksperimental] Menambahkan dukungan untuk model GPT-Neox.
- [Eksperimental] Menambahkan dukungan untuk model BLOOM.
- [Prototipe] Menambahkan dukungan untuk model LLAMA.
- Menambahkan dukungan untuk konfigurasi tensor-paralel yang lebih fleksibel ke GPT2, OPT, dan BLOOM.
- Menambahkan dukungan perhatian multi-kueri/multi-grup untuk GPT2.

Pembaruan versi

- DJL Melayani 0.23.0
- Obor Neuronx 1.13.1.1.9.0
- Transformer NeuronX 0.4.149

- Akselerasi Hugging Face 0.20.3
- Trafo Hugging Face 4.30.1
- Diffuser Wajah Memeluk 0.14.0

Arsitektur model yang didukung

- MEMILIH
- GPT2
- GPT-J
- GPT-Neox
- lLama
- Llama-2 7B dan 13B
- Mekar

Catatan Rilis LMI DLC: 7 Maret 2023

URI kontainer

- `763104351884.dkr.ecr.region.amazonaws.com/djl-inference:0.21.0-fastertransformer5.3.0-cu117`

Sorotan

- Pustaka paralel model FasterTransformer(FT) sekarang tersedia di DLC LMI, menambahkan dukungan untuk model populer seperti dan. `flan-t5-xxl` `flan-ul2` FT adalah perpustakaan sumber terbuka dari Nvidia yang menyediakan mesin akselerasi untuk menjalankan inferensi jaringan saraf berbasis transformator secara efisien. Ini dirancang untuk menangani model besar yang membutuhkan banyak GPU dan node secara terdistribusi. Pustaka mencakup versi blok transformator yang dioptimalkan, yang terdiri dari bagian encoder dan decoder, memungkinkan Anda menjalankan inferensi arsitektur encoder-decoder penuh seperti T5, serta model khusus encoder seperti BERT dan model khusus decoder seperti GPT.

Pembaruan versi

- DJL Melayani 0.21.0

- FasterTransformer 5.3.0
- Hugging Face Mempercepat 0.16.0
- Trafo Hugging Face 4.26.0
- Diffuser Wajah Memeluk 0.12.0
- Python 3.9
- CUDA 11.6

Masalah yang diketahui

- Dalam rilis pertama ini, Anda mungkin melihat waktu pemuatan model yang lama saat menggunakan pustaka FT. Pastikan untuk mengonfigurasi pemeriksaan kesehatan dan nilai batas waktu (lihatSageMaker parameter titik akhir untuk inferensi model besar) untuk memberikan waktu yang cukup untuk memuat model. Dalam pengujian kami, 1 jam sudah cukup. Rilis mendatang akan membahas waktu pemuatan model yang panjang ini dengan FT.

Catatan Rilis LMI DLC: 24 Februari 2023

URI kontainer

- `763104351884.dkr.ecr.region.amazonaws.com/djl-inference:0.21.0-deepspeed0.8.0-cu117`

Sorotan

- SageMaker Integrasi Python SDK baru dalam rilis ini, memungkinkan penerapan model bahasa besar tanpa kode

  Dukungan head-of-time partisi dengan DeepSpeed mengurangi waktu mulai titik akhir dengan memisahkan partisi model dari runtime

Pembaruan versi

- DJL Melayani 0.21.0
- DeepSpeed 0.8.0
- Hugging Face Mempercepat 0.16.0

- Trafo Hugging Face 4.26.0
- Diffuser Wajah Memeluk 0.12.0
- Python 3.9
- CUDA 11.6

Catatan Rilis DLC LMI: 16 Desember 2022

URI kontainer

- `763104351884.dkr.ecr.region.amazonaws.com/djl-inference:0.20.0-deepspeed0.7.5-cu116`

Sorotan

- Menambahkan dukungan presisi BF16 untuk model yang didukung oleh DeepSpeed inferensi (Bloom, GPT, OPT, dll.). Pelatihan di BF16 tidak memerlukan transformasi tambahan pada bobot model sebelum hosting. DeepSpeed
- Menambahkan dukungan untuk Hugging Face Diffusers StableDiffusion dan model.
- Termasuk versi terbaru dari DJL Serving, DeepSpeed, Hugging Face Transformers, dan driver CUDA, yang membawa peningkatan stabilitas, penanganan kesalahan yang lebih baik, cakupan model yang lebih baik, dan latensi.

Pembaruan versi

- DJL Melayani 0.20.0
- DeepSpeed 0.7.5
- Akselerasi Hugging Face 0.13.2
- Trafo Hugging Face 4.23.1
- Diffuser Wajah Memeluk 0.7.2
- CUDA 11.6

Catatan Rilis LMI DLC: 4 November 2022

Lampu Sorot

- Meluncurkan DLC LMI pada. AWS SageMaker dan Amazon EC2 mendukung DLC LMI untuk inferensi menggunakan paralelisme model.

URI kontainer

- `763104351884.dkr.ecr.region.amazonaws.com/djl-inference:0.19.0-deepspeed0.7.3-cu113`

Blog

- Untuk informasi selengkapnya, lihat Menerapkan BLOOM-176B dan OPT-30B di Amazon dengan inferensi SageMaker model besar Deep Learning Containers dan. DeepSpeed

Perbarui model dalam produksi

Pagar pembatas penerapan adalah serangkaian opsi penerapan model di Amazon SageMaker Inference untuk memperbarui model machine learning Anda dalam produksi. Dengan menggunakan opsi penyebaran yang dikelola sepenuhnya, Anda dapat mengontrol peralihan dari model saat ini dalam produksi ke yang baru. Mode pergeseran lalu lintas dalam penerapan biru/hijau, seperti kenari dan linier, memberi Anda kontrol terperinci atas proses pergeseran lalu lintas dari model Anda saat ini ke yang baru selama pembaruan. Ada juga perlindungan bawaan seperti pengembalian otomatis yang membantu Anda mengatasi masalah lebih awal dan secara otomatis mengambil tindakan korektif sebelum berdampak signifikan pada produksi.

Pagar pembatas penyebaran memberikan manfaat sebagai berikut:

- Keamanan penerapan saat memperbarui lingkungan produksi. Pembaruan regresif untuk lingkungan produksi dapat menyebabkan waktu henti yang tidak direncanakan dan dampak bisnis, seperti peningkatan latensi model dan tingkat kesalahan yang tinggi. Pagar pembatas penerapan membantu Anda mengurangi risiko tersebut dengan memberikan praktik terbaik dan pagar pembatas keselamatan operasional bawaan.
- Penyebaran yang dikelola sepenuhnya. SageMakermengurus pengaturan dan mengatur penerapan ini dan mengintegrasikannya dengan mekanisme pembaruan titik akhir. Anda tidak perlu membangun dan memelihara mekanisme orkestrasi, pemantauan, atau rollback. Anda dapat memanfaatkan SageMaker untuk mengatur dan mengatur penerapan ini dan fokus pada memanfaatkan ML untuk aplikasi Anda.

- Visibilitas. Anda dapat melacak kemajuan penerapan Anda melalui DescribeEndpointAPI atau melalui Amazon CloudWatch Events (untuk titik akhir yang didukung). Untuk mempelajari lebih lanjut tentang peristiwa diSageMaker, lihat bagian Perubahan status penerapan Endpoint di. Mengotomatisasi Amazon SageMaker dengan Amazon EventBridge Perhatikan bahwa jika titik akhir Anda menggunakan salah satu fitur di Pengecualian halaman, Anda tidak dapat menggunakan CloudWatch Acara.

> **Note**
>
> Pagar pembatas penerapan hanya berlaku untuk Inferensi asinkron dan Inferensi waktu nyata jenis titik akhir.

Cara memulai

Kami mendukung dua jenis penerapan untuk memperbarui model dalam produksi: penerapan biru/hijau dan penerapan bergulir.

- Deployment Biru/Hijau: Anda dapat menggeser lalu lintas dari armada lama Anda (armada biru) ke armada baru (armada hijau) dengan pembaruan. Penerapan biru/hijau menawarkan beberapa mode pergeseran lalu lintas. Mode pergeseran lalu lintas adalah konfigurasi yang menentukan cara SageMaker mengarahkan lalu lintas titik akhir ke armada baru yang berisi pembaruan Anda. Mode pergeseran lalu lintas berikut memberi Anda tingkat kontrol yang berbeda atas proses pembaruan titik akhir:
  - Semua Sekaligus Lalu Lintas Pergeseranmenggeser semua lalu lintas endpoint Anda dari armada biru ke armada hijau. Setelah lalu lintas bergeser ke armada hijau, CloudWatch alarm Amazon Anda yang telah ditentukan sebelumnya mulai memantau armada hijau untuk jangka waktu tertentu (periode pemanggangan). Jika tidak ada alarm perjalanan selama periode memanggang, kemudian SageMaker mengakhiri armada biru.
  - Pergeseran Canary Lalu Lintasmenggeser satu porsi kecil lalu lintas Anda (kenari) ke armada hijau dan memantaunya selama periode pemanggangan. Jika kenari berhasil di armada hijau, maka SageMaker menggeser sisa lalu lintas dari armada biru ke armada hijau sebelum mengakhiri armada biru.
  - Pergeseran lalu lintas linier memberikan lebih banyak penyesuaian atas jumlah langkah pergeseran lalu lintas dan persentase lalu lintas untuk bergeser untuk setiap langkah. Sementara

pergeseran kenari memungkinkan Anda menggeser lalu lintas dalam dua langkah, pergeseran linier memperluas ini ke n langkah spasi linear.

- Penyebaran Bergulir: Anda dapat memperbarui titik akhir sebagai kapasitas penyediaan SageMaker secara bertahap dan mengalihkan lalu lintas ke armada baru dalam langkah-langkah ukuran batch yang Anda tentukan. Instans pada armada baru diperbarui dengan konfigurasi penyebaran baru, dan jika tidak ada CloudWatch alarm yang tersandung selama periode pemanggangan, maka SageMaker bersihkan instance pada armada lama. Opsi ini memberi Anda kontrol terperinci atas jumlah instans atau persentase kapasitas yang digeser selama setiap langkah.

Anda dapat membuat dan mengelola penyebaran Anda melalui UpdateEndpointdan CreateEndpointSageMakerAPI dan AWS Command Line Interface perintah. Lihat halaman penyebaran individual untuk detail selengkapnya tentang cara mengatur penerapan Anda. Perhatikan bahwa jika titik akhir Anda menggunakan salah satu fitur yang tercantum di Pengecualian halaman, Anda tidak dapat menggunakan pagar pembatas penerapan.

Untuk mengikuti contoh terpandu yang menunjukkan cara menggunakan pagar pembatas penyebaran, lihat contoh notebook Jupyter untuk mode pergeseran lalu lintas kenari dan linier.

Konfigurasi dan Pemantauan Auto-Rollback

CloudWatch Alarm Amazon adalah prasyarat untuk menggunakan periode pemanggangan dalam pagar pembatas penerapan. Anda hanya dapat menggunakan fungsionalitas auto-rollback dalam pagar pembatas penerapan jika Anda mengatur CloudWatch alarm yang dapat memantau titik akhir. Jika salah satu alarm Anda berjalan selama periode pemantauan yang ditentukan, SageMaker memulai rollback lengkap ke titik akhir lama untuk melindungi aplikasi Anda. Jika Anda tidak memiliki CloudWatch alarm yang diatur untuk memantau titik akhir Anda, maka fungsionalitas auto-rollback tidak berfungsi selama penerapan Anda.

Untuk mempelajari lebih lanjut tentang Amazon CloudWatch, lihat Apa itu Amazon CloudWatch? di Panduan CloudWatch Pengguna Amazon.

> **Note**
>
> Pastikan peran eksekusi IAM Anda memiliki izin untuk melakukan`cloudwatch:DescribeAlarms` tindakan pada alarm rollback otomatis yang Anda tentukan.

Contoh Alarm

Untuk membantu Anda memulai, kami memberikan contoh berikut untuk menunjukkan kemampuan CloudWatch alarm. Selain menggunakan atau memodifikasi contoh berikut, Anda dapat membuat alarm sendiri dan mengonfigurasi alarm untuk memantau berbagai metrik pada armada yang ditentukan untuk jangka waktu tertentu. Untuk melihat lebih banyak SageMaker metrik dan dimensi yang dapat Anda tambahkan ke alarm, lihatPantau Amazon SageMaker dengan Amazon CloudWatch.

Topik

- Memantau kesalahan pemanggilan pada armada lama dan baru
- Monitor latensi model pada armada baru

Memantau kesalahan pemanggilan pada armada lama dan baru

CloudWatch Alarm berikut memonitor tingkat kesalahan rata-rata titik akhir. Anda dapat menggunakan alarm ini dengan jenis pemindahan lalu lintas pagar pembatas penyebaran untuk memberikan pemantauan keseluruhan pada armada lama dan baru. Jika alarm berjalan, maka SageMaker memulai rollback ke armada lama.

Kesalahan doa yang berasal dari armada lama dan armada baru berkontribusi pada tingkat kesalahan rata-rata. Jika tingkat kesalahan rata-rata melebihi ambang batas yang ditentukan, maka perjalanan alarm. Contoh khusus ini memonitor kesalahan 4xx (kesalahan klien) pada armada lama dan baru selama penyebaran. Anda juga dapat memantau kesalahan 5xx (kesalahan server) dengan menggunakan metrik`Invocation5XXErrors`.

> **Note**
>
> Untuk jenis alarm ini, jika armada lama Anda melakukan perjalanan alarm selama penyebaran, SageMaker menghentikan penyebaran Anda. Oleh karena itu, jika armada produksi Anda saat ini sudah menyebabkan kesalahan, pertimbangkan untuk menggunakan atau memodifikasi salah satu contoh berikut yang hanya memantau armada baru untuk kesalahan.

```
#Applied deployment type: all types
{
```

```
    "AlarmName": "EndToEndDeploymentHighErrorRateAlarm",
    "AlarmDescription": "Monitors the error rate of 4xx errors",
    "MetricName": "Invocation4XXErrors",
    "Namespace": "AWS/SageMaker",
    "Statistic": "Average",
    "Dimensions": [
        {
            "Name": "EndpointName",
            "Value": <your-endpoint-name>
        },
        {
            "Name": "VariantName",
            "Value": "AllTraffic"
        }
    ],
    "Period": 600,
    "EvaluationPeriods": 2,
    "Threshold": 1,
    "ComparisonOperator": "GreaterThanThreshold",
    "TreatMissingData": "notBreaching"
}
```

Di contoh sebelumnya, perhatikan nilai untuk bidang berikut:

- Untuk`AlarmName` dan`AlarmDescription`, masukkan nama dan deskripsi yang Anda pilih untuk alarm.
- Untuk`MetricName`, gunakan nilai`Invocation4XXErrors` untuk memantau kesalahan 4xx pada titik akhir
- Untuk`Namespace`, gunakan nilainya`AWS/SageMaker`. Anda juga dapat menentukan metrik kustom Anda sendiri, jika berlaku.
- Untuk`Statistic`, gunakan`Average`. Ini berarti bahwa alarm mengambil tingkat kesalahan rata-rata selama periode evaluasi saat menghitung apakah tingkat kesalahan telah melampaui ambang batas.
- Untuk dimensi`EndpointName`, gunakan nama titik akhir yang Anda perbarui sebagai nilainya.
- Untuk dimensi`VariantName`, gunakan nilai`AllTraffic` untuk menentukan semua lalu lintas titik akhir.
- Untuk`Period`, gunakan`600`. Ini menetapkan periode evaluasi alarm menjadi 10 menit.
- Untuk`EvaluationPeriods`, gunakan`2`. Nilai ini memberi tahu alarm untuk mempertimbangkan dua periode evaluasi terbaru saat menentukan status alarm.

Monitor latensi model pada armada baru

Contoh CloudWatch alarm berikut memonitor latensi model armada baru selama penerapan Anda. Anda dapat menggunakan alarm ini untuk memantau hanya armada baru dan mengecualikan armada lama. Alarm berlangsung untuk seluruh penyebaran. Contoh ini memberi Anda komprehensif, end-to-end pemantauan armada baru dan memulai rollback ke armada lama jika armada baru memiliki masalah waktu respons.

CloudWatch menerbitkan metrik dengan dimensi`EndpointConfigName:{New-Ep-Config}` setelah armada baru mulai menerima lalu lintas, dan metrik ini bertahan bahkan setelah penyebaran selesai.

Anda dapat menggunakan contoh alarm berikut dengan jenis deployment.

```
#Applied deployment type: all types
{
    "AlarmName": "NewEndpointConfigVersionHighModelLatencyAlarm",
    "AlarmDescription": "Monitors the model latency on new fleet",
    "MetricName": "ModelLatency",
    "Namespace": "AWS/SageMaker",
    "Statistic": "Average",
    "Dimensions": [
        {
            "Name": "EndpointName",
            "Value": <your-endpoint-name>
        },
        {
            "Name": "VariantName",
            "Value": "AllTraffic"
        },
        {
            "Name": "EndpointConfigName",
            "Value": <your-config-name>
    ],
    "Period": 300,
    "EvaluationPeriods": 2,
    "Threshold": 100000, # 100ms
    "ComparisonOperator": "GreaterThanThreshold",
    "TreatMissingData": "notBreaching"
}
```

Di contoh sebelumnya, perhatikan nilai untuk bidang berikut:

- Untuk`MetricName`, gunakan nilai`ModelLatency` untuk memantau waktu respons model.
- Untuk`Namespace`, gunakan nilainya`AWS/SageMaker`. Anda juga dapat menentukan metrik kustom Anda sendiri, jika berlaku.
- Untuk dimensi`EndpointName`, gunakan nama titik akhir yang Anda perbarui sebagai nilainya.
- Untuk dimensi`VariantName`, gunakan nilai`AllTraffic` untuk menentukan semua lalu lintas titik akhir.
- Untuk dimensi`EndpointConfigName`, nilainya harus mengacu pada nama konfigurasi titik akhir untuk titik akhir baru atau yang diperbarui.

> **Note**
>
> Jika Anda ingin memantau armada lama Anda alih-alih armada baru, Anda dapat mengubah dimensi`EndpointConfigName` untuk menentukan nama konfigurasi armada lama Anda.

Deployment Biru/Hijau

Saat Anda memperbarui titik akhir, Amazon SageMaker secara otomatis menggunakan penerapan biru/hijau untuk memaksimalkan ketersediaan titik akhir Anda. Dalam penyebaran biru/hijau, SageMaker ketentuan armada baru dengan pembaruan (armada hijau). Kemudian, SageMaker menggeser lalu lintas dari armada tua (armada biru) ke armada hijau. Setelah armada hijau beroperasi dengan lancar untuk periode evaluasi yang ditetapkan (disebut periode memanggang), SageMaker mengakhiri armada biru. Dengan kemampuan tambahan dalam penerapan biru/hijau, Anda dapat memanfaatkan mode pergeseran lalu lintas dan pemantauan pengembalian otomatis untuk melindungi titik akhir Anda dari dampak produksi yang signifikan.

Daftar berikut menjelaskan fitur utama penerapan biru/hijau di SageMaker:

- Mode pergeseran lalu lintas. Mode pergeseran lalu lintas untuk pagar pembatas penyebaran memungkinkan Anda mengontrol volume lalu lintas dan jumlah langkah pergeseran lalu lintas antara armada biru dan armada hijau. Kemampuan ini memberi Anda kemampuan untuk secara progresif mengevaluasi kinerja armada hijau tanpa sepenuhnya berkomitmen pada pergeseran lalu lintas 100%.
- Periode memanggang. Periode memanggang adalah jumlah waktu yang ditetapkan untuk memantau armada hijau sebelum melanjutkan ke tahap penyebaran berikutnya. Jika salah satu alarm yang telah ditentukan sebelumnya berjalan selama periode pemanggangan apa pun,

maka semua lalu lintas titik akhir kembali ke armada biru. Periode memanggang membantu Anda membangun kepercayaan pada pembaruan Anda sebelum membuat pergeseran lalu lintas permanen.

- Auto-rollback. Anda dapat menentukan CloudWatch alarm Amazon yang SageMaker digunakan untuk memantau armada hijau. Jika masalah dengan kode yang diperbarui membuat salah satu alarm, SageMaker memulai pengembalian otomatis ke armada biru untuk mempertahankan ketersediaan sehingga meminimalkan risiko.

Mode Pergeseran Lalu Lintas

Berbagai mode pergeseran lalu lintas dalam penyebaran biru/hijau memberi Anda kontrol yang lebih terperinci atas pergeseran lalu lintas antara armada biru dan armada hijau. Mode pergeseran lalu lintas yang tersedia untuk penyebaran biru/hijau semuanya sekaligus, kenari, dan linier. Tabel berikut menunjukkan perbandingan opsi.

> **Important**
>
> Untuk penyebaran biru/hijau yang melibatkan beberapa tahap lalu lintas pergeseran atau periode memanggang, Anda ditagih untuk kedua armada selama pembaruan, terlepas dari lalu lintas ke armada. Ini berbeda dengan penyebaran biru/hijau dengan pergeseran lalu lintas sekaligus dan tidak ada periode pemanggangan, di mana Anda hanya ditagih untuk satu armada selama pembaruan.

Nama	Apa itu?	Pro	Kontra	Rekomendasi
Semua sekaligus	Menggeser semua lalu lintas ke armada baru dalam satu langkah.	Meminimalkan durasi pembaruan secara keseluruhan.	Pembaruan regresif memengaruhi 100% lalu lintas.	Gunakan opsi ini untuk meminimalkan waktu dan biaya pembaruan.
Kenari	Lalu lintas bergeser dalam dua langkah. Langkah pertama (kenari)	Membatasi radius ledakan pembaruan regresif hanya	Kedua armada beroperasi secara parallel untuk seluruh penyebaran.	Gunakan opsi ini untuk menyeimbangkan antara meminimalkan

Nama	Apa itu?	Pro	Kontra	Rekomendasi
	menggeser sebagian kecil lalu lintas diikuti oleh langkah kedua, yang menggeser sisa lalu lintas.	untuk armada kenari.		radius ledakan pembaruan regresif dan meminimalkan waktu bahwa dua armada beroperasi.
Linear	Sebagian tetap dari pergesera n lalu lintas dalam jumlah langkah yang sama spasi yang telah ditentukan sebelumnya.	Meminimalkan risiko pembaruan regresif dengan menggeser lalu lintas melalui beberapa langkah.	Durasi dan biaya pembaruan sebanding dengan jumlah langkah.	Gunakan opsi ini untuk meminimalkan risiko dengan menyebarkan penyebaran di beberapa langkah.

Mulai

Setelah Anda menentukan konfigurasi penerapan yang diinginkan, SageMaker menangani penyediaan instans baru, menghentikan instans lama, dan mengalihkan lalu lintas untuk Anda. Anda dapat membuat dan mengelola penyebaran Anda melalui CreateEndpoint SageMaker API UpdateEndpointdanAWS Command Line Interface perintah yang ada dan. Perhatikan bahwa jika titik akhir Anda menggunakan salah satu fitur yang tercantum diPengecualian halaman, Anda tidak dapat menggunakan pagar pembatas penerapan. Lihat halaman penyebaran individual untuk detail selengkapnya tentang cara mengatur penerapan Anda:

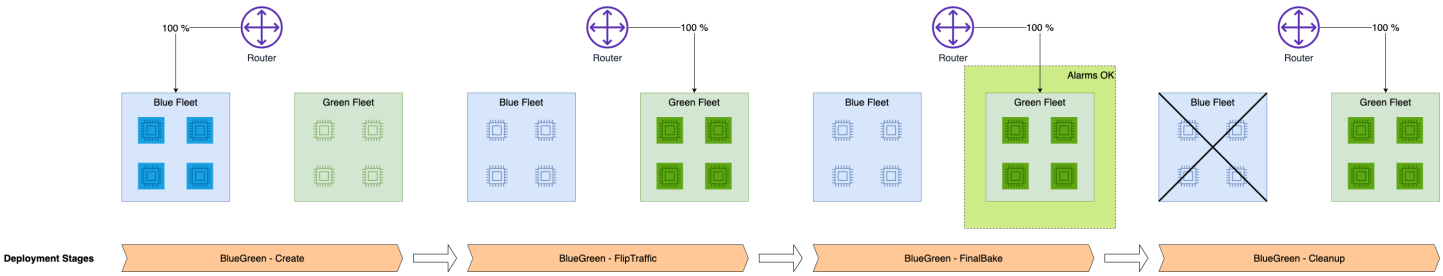
- Pembaruan Biru/Hijau dengan Pergeseran Lalu Lintas Sekaligus
- Pembaruan Biru/Hijau dengan Pergeseran Lalu Lintas Canary
- Pembaruan Biru/Hijau dengan Pergeseran Lalu Lintas Linear

Untuk mengikuti contoh terpandu yang menunjukkan cara menggunakan pagar pembatas penyebaran, lihat contoh notebook Jupyter untuk mode pergeseran lalu lintas kenari dan linier.

Semua Sekaligus Lalu Lintas Pergeseran

Dengan semua sekaligus lalu lintas bergeser, Anda dapat dengan cepat meluncurkan pembaruan endpoint menggunakan pagar keselamatan penyebaran biru/hijau. Anda dapat menggunakan opsi pergeseran lalu lintas ini untuk meminimalkan durasi pembaruan sambil tetap memanfaatkan jaminan ketersediaan penyebaran biru/hijau. Fitur periode baking membantu Anda untuk memantau kinerja dan fungsionalitas instans baru Anda sebelum mengakhiri instans lama Anda, memastikan bahwa armada baru Anda beroperasi penuh.

Diagram berikut menunjukkan bagaimana semua sekaligus lalu lintas bergeser mengelola armada lama dan baru.



Ketika Anda menggunakan semua sekaligus lalu lintas bergeser, SageMaker rute 100% dari lalu lintas ke armada baru (armada hijau). Setelah armada hijau mulai menerima lalu lintas, periode baking dimulai. Periode baking adalah jumlah waktu yang ditetapkan di mana alarm Amazon CloudWatch yang telah ditentukan sebelumnya memantau kinerja armada hijau. Jika tidak ada alarm perjalanan selama periode baking, SageMaker mengakhiri armada tua (armada biru). Jika ada alarm perjalanan selama periode baking, maka auto-rollback memulai dan 100% dari lalu lintas bergeser kembali ke armada biru.

Prasyarat

Sebelum menyiapkan penyebaran dengan semua sekaligus pergeseran lalu lintas, Anda harus membuat alarm Amazon CloudWatch untuk menonton metrik dari titik akhir Anda. Jika salah satu alarm perjalanan selama periode baking, maka lalu lintas berguling kembali ke armada biru Anda. Untuk mempelajari cara mengatur alarm CloudWatch di endpoint, lihat halaman prasyaratKonfigurasi dan Pemantauan Auto-Rollback. Untuk mempelajari selengkapnya tentang alarm CloudWatch, lihatMenggunakan alarm Amazon CloudWatchdiPanduan Pengguna Amazon CloudWatch.

Konfigurasi Semua Sekaligus Pergeseran Lalu Lintas

Setelah Anda siap untuk penyebaran Anda dan telah menyiapkan alarm CloudWatch untuk endpoint Anda, Anda dapat menggunakan SageMakerUpdateEndPointAPI atautitik akhirperintah diAWS Command Line Interfaceuntuk memulai penyebaran.

Topik

- Cara memperbarui endpoint (API)
- Cara memperbarui endpoint dengan kebijakan pembaruan biru/hijau yang ada (API)
- Cara memperbarui endpoint (CLI)

Cara memperbarui endpoint (API)

Contoh berikut menunjukkan cara memperbarui titik akhir Anda sekaligus pergeseran lalu lintas menggunakanUpdateEndPointdi Amazon SageMaker API.

```
import boto3
client = boto3.client("sagemaker")

response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-config-name>",
    DeploymentConfig={
        "BlueGreenUpdatePolicy": {
            "TrafficRoutingConfiguration": {
                "Type": "ALL_AT_ONCE"
            },
            "TerminationWaitInSeconds": 600,
            "MaximumExecutionTimeoutInSeconds": 1800
        },
        "AutoRollbackConfiguration": {
            "Alarms": [
                {
                    "AlarmName": "<your-cw-alarm>"
                },
            ]
        }
    }
)
```

Untuk mengkonfigurasi semua sekaligus opsi pergeseran lalu lintas, lakukan hal berikut:

- Untuk`EndpointName`, gunakan nama titik akhir yang ingin Anda perbarui.
- Untuk`EndpointConfigName`, gunakan nama konfigurasi titik akhir yang ingin Anda gunakan.

- Di bawah`DeploymentConfig`dan`BlueGreenUpdatePolicy`, di`TrafficRoutingConfiguration`, mengatur`Type`parameter`ALL_AT_ONCE`. Ini menentukan bahwa penyebaran menggunakan semua sekaligus lalu lintas modus pergeseran.
- Untuk`TerminationWaitInSeconds`, gunakan`600`. Parameter ini memberitahu SageMaker untuk menunggu jumlah waktu yang ditentukan (dalam detik) setelah armada hijau Anda sepenuhnya aktif sebelum mengakhiri instance dalam armada biru. Dalam contoh ini, SageMaker menunggu selama 10 menit setelah periode baking akhir sebelum mengakhiri armada biru.
- Untuk`MaximumExecutionTimeoutInSeconds`, gunakan`1800`. Parameter ini menetapkan jumlah maksimum waktu yang dapat dijalankan deployment sebelum waktu habis. Pada contoh sebelumnya, penyebaran Anda memiliki batas 30 menit untuk menyelesaikan.
- Masuk`AutoRollbackConfiguration`, dalam`Alarms`bidang, Anda dapat menambahkan alarm CloudWatch berdasarkan nama. Membuat satu`AlarmName: <your-cw-alarm>`entri untuk setiap alarm yang ingin Anda gunakan.

Cara memperbarui endpoint dengan kebijakan pembaruan biru/hijau yang ada (API)

Saat Anda menggunakanCreateEndPointAPI untuk membuat endpoint, Anda dapat menentukan konfigurasi penyebaran untuk digunakan kembali untuk pembaruan endpoint di masa mendatang. Anda dapat menggunakan`DeploymentConfig`pilihan sebagai contoh API UpdateEndPoint sebelumnya. Tidak ada perubahan pada perilaku CreateEndPoint API. Menentukan konfigurasi penyebaran tidak secara otomatis melakukan pembaruan biru/hijau pada titik akhir Anda.

Opsi untuk menggunakan konfigurasi penyebaran sebelumnya terjadi saat menggunakanUpdateEndPointAPI untuk memperbarui titik akhir Anda. Saat memperbarui titik akhir Anda, Anda dapat menggunakan`RetainDeploymentConfig`opsi untuk menjaga konfigurasi deployment yang Anda tentukan saat membuat titik akhir.

Saat memanggilUpdateEndPointAPI, mengatur`RetainDeploymentConfig`kepada`True`untuk menjaga`DeploymentConfig`pilihan dari konfigurasi titik akhir asli Anda.

```
response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-config-name>",
    RetainDeploymentConfig=True
)
```

Cara memperbarui endpoint (CLI)

Jika Anda menggunakanAWS CLI, contoh berikut menunjukkan bagaimana memulai biru/hijau sekaligus penyebaran menggunakantitik akhirperintah.

```
update-endpoint
--endpoint-name <your-endpoint-name>
--endpoint-config-name <your-config-name>
--deployment-config '"BlueGreenUpdatePolicy": {"TrafficRoutingConfiguration": {"Type":
 "ALL_AT_ONCE"},
    "TerminationWaitInSeconds": 600, "MaximumExecutionTimeoutInSeconds": 1800},
    "AutoRollbackConfiguration": {"Alarms": [{"AlarmName": "<your-alarm>"}]}'
```

Untuk mengkonfigurasi semua sekaligus opsi pergeseran lalu lintas, lakukan hal berikut:

- Untuk`endpoint-name`, gunakan nama titik akhir yang ingin Anda perbarui.
- Untuk`endpoint-config-name`, gunakan nama konfigurasi titik akhir yang ingin Anda gunakan.
- Untuk`deployment-config`, gunakanBlueGreenUpdatePolicyObjek JSON.

> **Note**
>
> Jika Anda lebih suka menyimpan objek JSON Anda dalam sebuah file, lihatMenghasilkanAWS CLIparameter kerangka dan inputdiAWS CLIPanduan Pengguna.

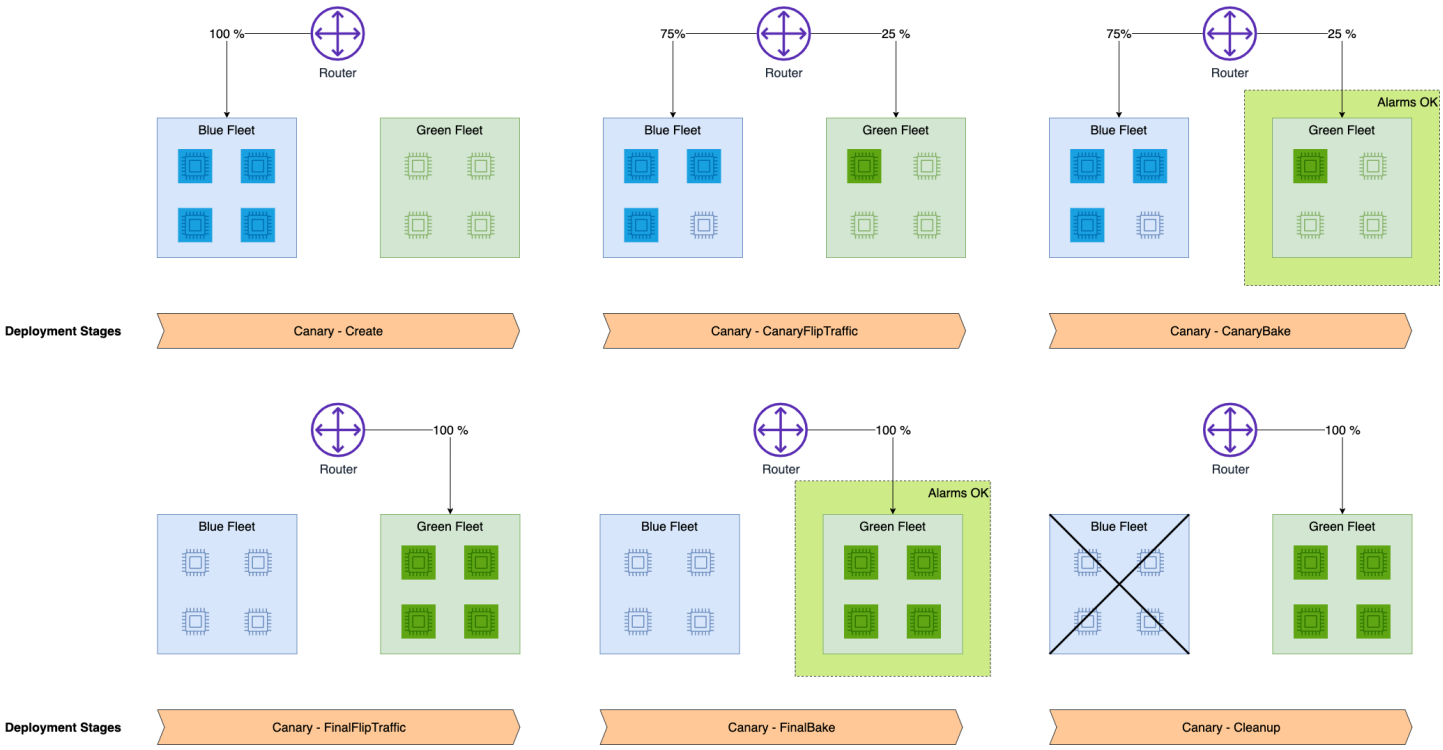
Pergeseran Canary Lalu Lintas

Dengan pergeseran lalu lintas kenari, Anda dapat menguji sebagian lalu lintas titik akhir Anda pada armada baru sementara armada lama melayani sisa lalu lintas. Langkah pengujian ini adalah pagar pembatas keselamatan yang memvalidasi fungsionalitas armada baru sebelum mengalihkan semua lalu lintas Anda ke armada baru. Anda masih memiliki manfaat dari penyebaran biru/hijau, dan fitur kenari yang ditambahkan memungkinkan Anda memastikan bahwa armada baru (hijau) dapat melayani kesimpulan sebelum membiarkannya menangani 100% lalu lintas.

Bagian armada hijau Anda yang menyala untuk menerima lalu lintas disebut kenari, dan Anda dapat memilih ukuran kenari ini. Perhatikan bahwa ukuran kenari harus kurang dari atau sama dengan 50% kapasitas armada baru. Setelah periode baking selesai dan tidak ada perjalanan alarm Amazon CloudWatch yang telah ditentukan sebelumnya, sisa lalu lintas bergeser dari armada lama (biru)

ke armada hijau. Pergeseran lalu lintas Canary memberi Anda lebih banyak keamanan selama penyebaran Anda karena masalah apa pun dengan model yang diperbarui hanya memengaruhi kenari.

Diagram berikut menunjukkan bagaimana pergeseran lalu lintas kenari mengelola distribusi lalu lintas antara armada biru dan hijau.



Setelah SageMaker menyediakan armada hijau, SageMaker mengarahkan sebagian lalu lintas yang masuk (misalnya, 25%) ke kenari. Kemudian periode baking dimulai, di mana alarm CloudWatch Anda memantau kinerja armada hijau. Selama waktu ini, armada biru dan armada hijau sebagian aktif dan menerima lalu lintas. Jika salah satu alarm perjalanan selama periode baking, maka SageMaker memulai rollback dan semua lalu lintas kembali ke armada biru. Jika tidak ada alarm perjalanan, maka semua lalu lintas bergeser ke armada hijau dan ada periode baking akhir. Jika periode baking akhir selesai tanpa tersandung alarm, maka armada hijau melayani semua lalu lintas dan SageMaker mengakhiri armada biru.

Prasyarat

Sebelum menyiapkan penyebaran dengan pergeseran lalu lintas canary, Anda harus membuat alarm Amazon CloudWatch untuk memantau metrik dari titik akhir Anda. Alarm aktif selama periode baking, dan jika ada alarm perjalanan, maka semua titik akhir lalu lintas gulungan kembali ke armada biru. Untuk mempelajari cara mengatur alarm CloudWatch di endpoint, lihat halaman

prasyaratKonfigurasi dan Pemantauan Auto-Rollback. Untuk mempelajari selengkapnya tentang alarm CloudWatchMenggunakan alarm Amazon CloudWatchdiPanduan Pengguna Amazon CloudWatch.

Mengkonfigurasi Pergeseran Lalu Lintas Canary

Setelah Anda siap untuk penyebaran Anda dan telah menyiapkan alarm Amazon CloudWatch untuk endpoint Anda, Anda dapat menggunakan Amazon SageMakerUpdateEndPointAPI atautitik akhir pembaruanperintah diAWS CLIuntuk memulai penyebaran.

Topik

- Cara memperbarui endpoint (API)
- Cara memperbarui endpoint dengan kebijakan pembaruan biru/hijau yang ada (API)
- Cara memperbarui endpoint (CLI)

Cara memperbarui endpoint (API)

Contoh berikutUpdateEndPointAPI menunjukkan bagaimana Anda dapat memperbarui endpoint dengan pergeseran lalu lintas kenari.

```
import boto3
client = boto3.client("sagemaker")

response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-config-name>",
    DeploymentConfig={
        "BlueGreenUpdatePolicy": {
            "TrafficRoutingConfiguration": {
                "Type": "CANARY",
                "CanarySize": {
                    "Type": "CAPACITY_PERCENT",
                    "Value": 30
                },
                "WaitIntervalInSeconds": 600
            },
            "TerminationWaitInSeconds": 600,
            "MaximumExecutionTimeoutInSeconds": 1800
        },
        "AutoRollbackConfiguration": {
```

```
            "Alarms": [
                {
                    "AlarmName": "<your-cw-alarm>"
                }
            ]
        }
    }
)
```

Untuk mengkonfigurasi opsi pergeseran lalu lintas kenari, lakukan hal berikut:

- Untuk`EndpointName`, gunakan nama titik akhir yang ingin Anda perbarui.
- Untuk`EndpointConfigName`, gunakan nama konfigurasi titik akhir yang ingin Anda gunakan.
- Di bawah`DeploymentConfig`dan`BlueGreenUpdatePolicy`, di`TrafficRoutingConfiguration`, mengatur`Type`parameter untuk`CANARY`. Ini menentukan bahwa penyebaran menggunakan pergeseran lalu lintas kenari.
- Di`CanarySize`lapangan, Anda dapat mengubah ukuran kenari dengan memodifikasi`Type`dan`Value`parameter. Untuk`Type`, gunakan`CAPACITY_PERCENT`, yang berarti persentase armada hijau Anda yang ingin Anda gunakan sebagai kenari, dan kemudian ditetapkan`Value`kepada`30`. Dalam contoh ini, Anda menggunakan 30% dari kapasitas armada hijau sebagai kenari. Perhatikan bahwa ukuran kenari harus sama dengan atau kurang dari 50% dari kapasitas armada hijau.
- Untuk`WaitIntervalInSeconds`, gunakan`600`. Parameter memberitahu SageMaker untuk menunggu jumlah waktu yang ditentukan (dalam detik) antara setiap shift interval. Interval ini adalah durasi periode kue kenari. Dalam contoh sebelumnya, SageMaker menunggu selama 10 menit setelah pergeseran kenari dan kemudian menyelesaikan pergeseran lalu lintas kedua dan terakhir.
- Untuk`TerminationWaitInSeconds`, gunakan`600`. Parameter ini memberitahu SageMaker untuk menunggu jumlah waktu yang ditentukan (dalam detik) setelah armada hijau Anda sepenuhnya aktif sebelum mengakhiri instance dalam armada biru. Dalam contoh ini, SageMaker menunggu selama 10 menit setelah periode baking akhir sebelum mengakhiri armada biru.
- Untuk`MaximumExecutionTimeoutInSeconds`, gunakan`1800`. Parameter ini menetapkan jumlah maksimum waktu yang dapat dijalankan deployment sebelum waktu habis. Pada contoh sebelumnya, penyebaran Anda memiliki batas 30 menit untuk menyelesaikan.
- Masuk`AutoRollbackConfiguration`, dalam`Alarms`bidang, Anda dapat menambahkan alarm CloudWatch berdasarkan nama. Membuat satu`AlarmName: <your-cw-alarm>`entri untuk setiap alarm yang ingin Anda gunakan.

Cara memperbarui endpoint dengan kebijakan pembaruan biru/hijau yang ada (API)

Saat Anda menggunakanCreateEndPointAPI untuk membuat endpoint, Anda dapat menentukan konfigurasi penyebaran untuk digunakan kembali untuk pembaruan endpoint di masa mendatang. Anda dapat menggunakan`DeploymentConfig`pilihan sebagai contoh API UpdateEndPoint sebelumnya. Tidak ada perubahan pada perilaku CreateEndPoint API. Menentukan konfigurasi penyebaran tidak secara otomatis melakukan pembaruan biru/hijau pada titik akhir Anda.

Opsi untuk menggunakan konfigurasi penyebaran sebelumnya terjadi saat menggunakanUpdateEndPointAPI untuk memperbarui titik akhir Anda. Saat memperbarui titik akhir Anda, Anda dapat menggunakan`RetainDeploymentConfig`opsi untuk menjaga konfigurasi deployment yang Anda tentukan saat membuat titik akhir.

Saat memanggilUpdateEndPointAPI, mengatur`RetainDeploymentConfig`kepada`True`menjaga`DeploymentConfig`pilihan dari konfigurasi titik akhir asli Anda.

```
response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-config-name>",
    RetainDeploymentConfig=True
)
```

Cara memperbarui endpoint (CLI)

Jika Anda menggunakanAWS CLI, contoh berikut menunjukkan bagaimana memulai penyebaran kenari biru/hijau menggunakantitik akhir pembaruanperintah.

```
update-endpoint
--endpoint-name <your-endpoint-name>
--endpoint-config-name <your-config-name>
--deployment-config '"BlueGreenUpdatePolicy": {"TrafficRoutingConfiguration": {"Type":
 "CANARY",
    "CanarySize": {"Type": "CAPACITY_PERCENT", "Value": 30}, "WaitIntervalInSeconds":
 600},
    "TerminationWaitInSeconds": 600, "MaximumExecutionTimeoutInSeconds": 1800},
    "AutoRollbackConfiguration": {"Alarms": [{"AlarmName": "<your-alarm>"}]}'
```

Untuk mengkonfigurasi opsi pergeseran lalu lintas kenari, lakukan hal berikut:

- Untuk`endpoint-name`, gunakan nama titik akhir yang ingin Anda perbarui.

- Untuk`endpoint-config-name`, gunakan nama konfigurasi titik akhir yang ingin Anda gunakan.
- Untuk`deployment-config`, gunakanBlueGreenUpdatePolicyObjek JSON.

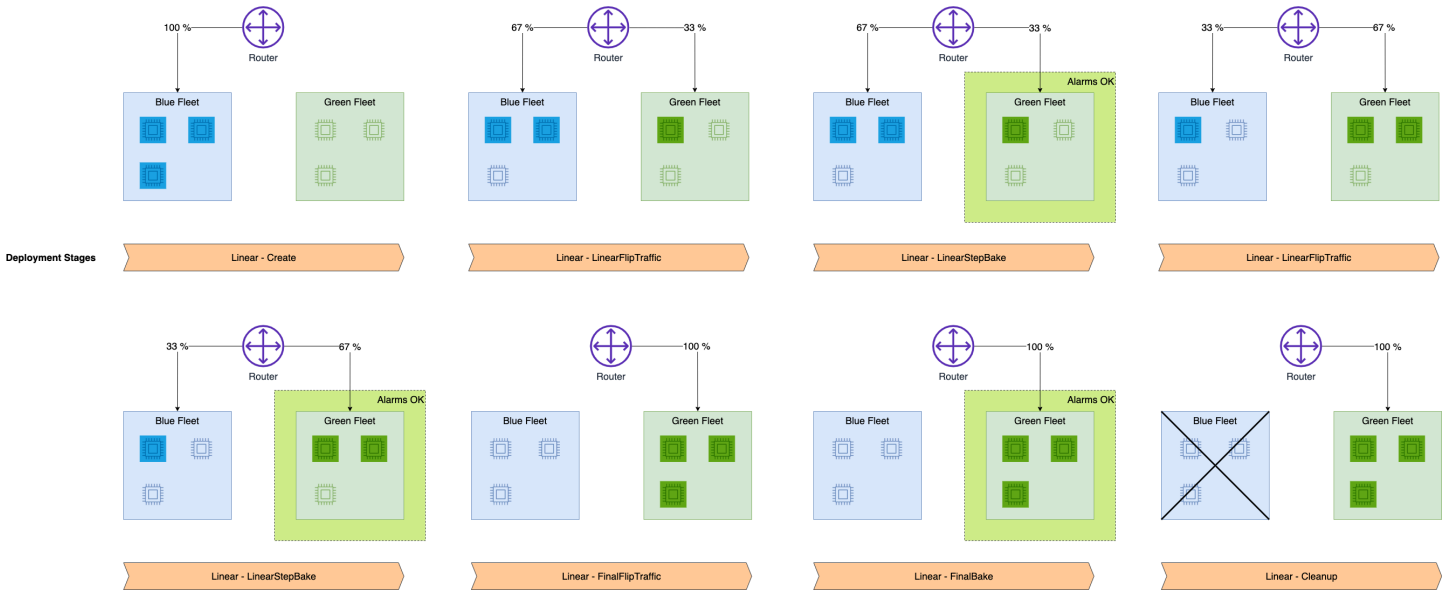
> **Note**
>
> Jika Anda lebih suka menyimpan objek JSON Anda dalam sebuah file, lihatMenghasilkanAWS CLIparameter kerangka dan inputdiAWS CLIPanduan Pengguna.

Pergeseran lalu lintas linier

Pergeseran lalu lintas linier memungkinkan Anda untuk secara bertahap menggeser lalu lintas dari armada lama Anda (armada biru) ke armada baru Anda (armada hijau). Dengan pergeseran lalu lintas linier, Anda dapat menggeser lalu lintas dalam beberapa langkah, meminimalkan kemungkinan gangguan pada titik akhir Anda. Opsi penyebaran biru/hijau ini memberi Anda kontrol paling terperinci atas pergeseran lalu lintas.

Anda dapat memilih jumlah instance atau persentase kapasitas armada hijau untuk diaktifkan selama setiap langkah. Setiap langkah linier hanya boleh antara 10-50% dari kapasitas armada hijau. Untuk setiap langkah, ada periode baking selama alarm Amazon CloudWatch Anda yang telah ditentukan sebelumnya memantau metrik pada armada hijau. Setelah periode baking selesai dan tidak ada alarm perjalanan, bagian aktif armada hijau Anda terus menerima lalu lintas dan langkah baru dimulai. Jika alarm perjalanan selama salah satu periode baking, 100% dari lalu lintas endpoint gulungan kembali ke armada biru.

Diagram berikut menunjukkan bagaimana lalu lintas linier pergeseran rute lalu lintas ke armada biru dan hijau.



Setelah SageMaker menyediakan armada baru, bagian pertama armada hijau menyala dan menerima lalu lintas. SageMaker menonaktifkan porsi ukuran armada biru yang sama, dan periode baking dimulai. Jika ada alarm perjalanan, semua lalu lintas endpoint gulungan kembali ke armada biru. Jika periode baking selesai, maka langkah selanjutnya dimulai. Bagian lain dari armada hijau mengaktifkan dan menerima lalu lintas, bagian dari armada biru menonaktifkan, dan periode baking lainnya dimulai. Proses yang sama berulang sampai armada biru sepenuhnya dinonaktifkan dan armada hijau sepenuhnya aktif dan menerima semua lalu lintas. Jika alarm padam pada titik mana pun, SageMaker mengakhiri proses pergeseran dan 100% lalu lintas kembali ke armada biru.

Prasyarat

Sebelum menyiapkan penyebaran dengan pergeseran lalu lintas linier, Anda harus membuat alarm CloudWatch untuk memantau metrik dari titik akhir. Alarm aktif selama periode baking, dan jika ada alarm perjalanan, maka semua titik akhir lalu lintas gulungan kembali ke armada biru. Untuk mempelajari cara mengatur alarm CloudWatch di endpoint, lihat halaman prasyaratKonfigurasi dan Pemantauan Auto-Rollback. Untuk mempelajari selengkapnya tentang alarm CloudWatch, lihatMenggunakan alarm Amazon CloudWatchdiPanduan Pengguna Amazon CloudWatch.

Konfigurasikan Pergeseran Lalu Linear

Setelah Anda siap untuk penyebaran Anda dan telah menyiapkan alarm CloudWatch untuk endpoint Anda, Anda dapat menggunakan Amazon SageMakerUpdateEndPointAPI autitik akhirperintah diAWS CLIuntuk memulai penyebaran.

Topik

- Cara memperbarui endpoint (API)
- Cara memperbarui endpoint dengan kebijakan pembaruan biru/hijau yang ada (API)
- Cara memperbarui endpoint (CLI)

Cara memperbarui endpoint (API)

Contoh berikutUpdateEndPointAPI menunjukkan bagaimana Anda dapat memperbarui endpoint dengan pergeseran lalu lintas linier.

```
import boto3
client = boto3.client("sagemaker")

response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-config-name>",
    DeploymentConfig={
        "BlueGreenUpdatePolicy": {
            "TrafficRoutingConfiguration": {
                "Type": "LINEAR",
                "LinearStepSize": {
                    "Type": "CAPACITY_PERCENT",
                    "Value": 20
                },
                "WaitIntervalInSeconds": 300
            },
            "TerminationWaitInSeconds": 300,
            "MaximumExecutionTimeoutInSeconds": 3600
        },
        "AutoRollbackConfiguration": {
            "Alarms": [
                {
                    "AlarmName": "<your-cw-alarm>"
                }
            ]
        }
    }
)
```

Untuk mengkonfigurasi opsi pergeseran lalu lintas linier, lakukan hal berikut:

- Untuk`EndpointName`, gunakan nama titik akhir yang ingin Anda perbarui.

- Untuk`EndpointConfigName`, gunakan nama konfigurasi titik akhir yang ingin Anda gunakan.
- Di bawah`DeploymentConfig`dan`BlueGreenUpdatePolicy`, di`TrafficRoutingConfiguration`, mengatur`Type`parameter`LINEAR`. Ini menentukan bahwa penyebaran menggunakan pergeseran lalu lintas linier.
- Di`LinearStepSize`bidang, Anda dapat mengubah ukuran langkah-langkah dengan memodifikasi`Type`dan`Value`parameter. Untuk`Type`, gunakan`CAPACITY_PERCENT`, yang berarti persentase armada hijau Anda yang ingin Anda gunakan sebagai ukuran langkah, dan kemudian atur`Value`kepada`20`. Dalam contoh ini, Anda mengaktifkan 20% kapasitas armada hijau untuk setiap langkah pergeseran lalu lintas. Perhatikan bahwa ketika menyesuaikan ukuran langkah linier Anda, Anda hanya harus menggunakan langkah-langkah yang 10-50% dari kapasitas armada hijau.
- Untuk`WaitIntervalInSeconds`, gunakan`300`. Parameter memberitahu SageMaker untuk menunggu jumlah waktu yang ditentukan (dalam detik) antara setiap shift lalu lintas. Interval ini adalah durasi periode baking antara setiap langkah linier. Pada contoh sebelumnya, SageMaker menunggu selama 5 menit di antara setiap shift lalu lintas.
- Untuk`TerminationWaitInSeconds`, gunakan`300`. Parameter ini memberitahu SageMaker untuk menunggu jumlah waktu yang ditentukan (dalam detik) setelah armada hijau Anda sepenuhnya aktif sebelum mengakhiri instance dalam armada biru. Dalam contoh ini, SageMaker menunggu selama 5 menit setelah periode baking akhir sebelum mengakhiri armada biru.
- Untuk`MaximumExecutionTimeoutInSeconds`, gunakan`3600`. Parameter ini menetapkan jumlah waktu maksimum yang dapat dijalankan deployment sebelum habis. Pada contoh sebelumnya, penyebaran Anda memiliki batas 1 jam untuk menyelesaikan.
- Masuk`AutoRollbackConfiguration`, dalam`Alarms`bidang, Anda dapat menambahkan alarm CloudWatch berdasarkan nama. Membuat satu`AlarmName: <your-cw-alarm>`entri untuk setiap alarm yang ingin Anda gunakan.

Cara memperbarui endpoint dengan kebijakan pembaruan biru/hijau yang ada (API)

Saat Anda menggunakanCreateEndPointAPI untuk membuat endpoint, Anda dapat menentukan konfigurasi penyebaran untuk digunakan kembali untuk pembaruan endpoint di masa mendatang. Anda dapat menggunakan`DeploymentConfig`pilihan sebagai contoh API UpdateEndPoint sebelumnya. Tidak ada perubahan pada perilaku CreateEndPoint API. Menentukan konfigurasi penyebaran tidak secara otomatis melakukan pembaruan biru/hijau pada titik akhir Anda.

Opsi untuk menggunakan konfigurasi penyebaran sebelumnya terjadi saat menggunakanUpdateEndPointAPI untuk memperbarui titik akhir Anda. Saat memperbarui titik akhir

Anda, Anda dapat menggunakan`RetainDeploymentConfig`opsi untuk menjaga konfigurasi deployment yang Anda tentukan saat membuat titik akhir.

Saat memanggilUpdateEndPointAPI, mengatur`RetainDeploymentConfig`kepada`True`untuk menjaga`DeploymentConfig`pilihan dari konfigurasi titik akhir asli Anda.

```
response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-config-name>",
    RetainDeploymentConfig=True
)
```

Cara memperbarui endpoint (CLI)

Jika Anda menggunakanAWS CLI, contoh berikut menunjukkan bagaimana memulai penyebaran linier biru/hijau menggunakantitik akhirperintah.

```
update-endpoint
--endpoint-name <your-endpoint-name>
--endpoint-config-name <your-config-name>
--deployment-config '{"BlueGreenUpdatePolicy": {"TrafficRoutingConfiguration": {"Type":
 "LINEAR",
    "LinearStepSize": {"Type": "CAPACITY_PERCENT", "Value": 20},
 "WaitIntervalInSeconds": 300},
    "TerminationWaitInSeconds": 300, "MaximumExecutionTimeoutInSeconds": 3600},
    "AutoRollbackConfiguration": {"Alarms": [{"AlarmName": "<your-alarm>"}]}'
```

Untuk mengkonfigurasi opsi pergeseran lalu lintas linier, lakukan hal berikut:

- Untuk`endpoint-name`, gunakan nama titik akhir yang ingin Anda perbarui.
- Untuk`endpoint-config-name`, gunakan nama konfigurasi titik akhir yang ingin Anda gunakan.
- Untuk`deployment-config`, gunakanBlueGreenUpdatePolicyObjek JSON.

> **Note**
>
> Jika Anda lebih suka menyimpan objek JSON Anda dalam sebuah file, lihatMenghasilkanAWS CLIparameter kerangka dan inputdiAWS CLIPanduan Pengguna.

Penyebaran Bergulir

Saat memperbarui titik akhir, Anda dapat menentukan penyebaran bergulir untuk secara bertahap mengalihkan lalu lintas dari armada lama Anda ke armada baru. Anda dapat mengontrol ukuran langkah pergeseran lalu lintas, serta menentukan periode evaluasi untuk memantau instance baru untuk masalah sebelum mengakhiri instance dari armada lama. Dengan penyebaran bergulir, instance pada armada lama dibersihkan setelah setiap pergeseran lalu lintas ke armada baru, mengurangi jumlah instance tambahan yang diperlukan untuk memperbarui titik akhir Anda. Hal ini berguna terutama untuk kasus dipercepat yang dalam permintaan tinggi.

Penyebaran bergulir secara bertahap menggantikan penerapan versi model Anda sebelumnya dengan versi baru dengan memperbarui titik akhir Anda dalam ukuran batch yang dapat dikonfigurasi. Perilaku pergeseran lalu lintas dari penyebaran bergulir mirip dengan mode pergeseran lalu lintas linier dalam penerapan biru/hijau, tetapi penerapan bergulir memberi Anda manfaat dari persyaratan kapasitas yang dikurangi jika dibandingkan dengan penerapan biru/hijau. Dengan penerapan bergulir, lebih sedikit instance yang aktif dalam satu waktu, dan Anda memiliki kontrol lebih terperinci atas berapa banyak instance yang ingin Anda perbarui di armada baru. Anda harus mempertimbangkan untuk menggunakan penyebaran bergulir alih-alih penyebaran biru/hijau jika Anda memiliki model besar atau titik akhir besar dengan banyak instance.

Daftar berikut menjelaskan fitur-fitur utama dari penerapan bergulir di Amazon: SageMaker

- Periode memanggang.Periode memanggang adalah jumlah waktu yang ditetapkan untuk memantau armada baru sebelum melanjutkan ke tahap penyebaran berikutnya. Jika salah satu alarm yang telah ditentukan sebelumnya berjalan selama periode pemanggangan apa pun, maka semua lalu lintas titik akhir kembali ke armada lama. Periode memanggang membantu Anda membangun kepercayaan pada pembaruan Anda sebelum membuat pergeseran lalu lintas permanen.
- Ukuran batch bergulir. Anda memiliki kontrol terperinci atas ukuran setiap batch untuk pergeseran lalu lintas, atau jumlah instance yang ingin Anda perbarui di setiap batch. Jumlah ini bisa berkisar 5-50% dari ukuran armada Anda. Anda dapat menentukan ukuran batch sebagai sejumlah instance atau sebagai persentase keseluruhan armada Anda.
- Auto-rollback.Anda dapat menentukan CloudWatch alarm Amazon yang SageMaker digunakan untuk memantau armada baru. Jika masalah dengan kode yang diperbarui membuat salah satu alarm, SageMaker memulai pengembalian otomatis ke armada lama untuk mempertahankan ketersediaan, sehingga meminimalkan risiko.

> **Note**
>
> Jika titik akhir Anda menggunakan salah satu fitur yang tercantum di halaman Pengecualian, Anda tidak dapat menggunakan penerapan bergulir.

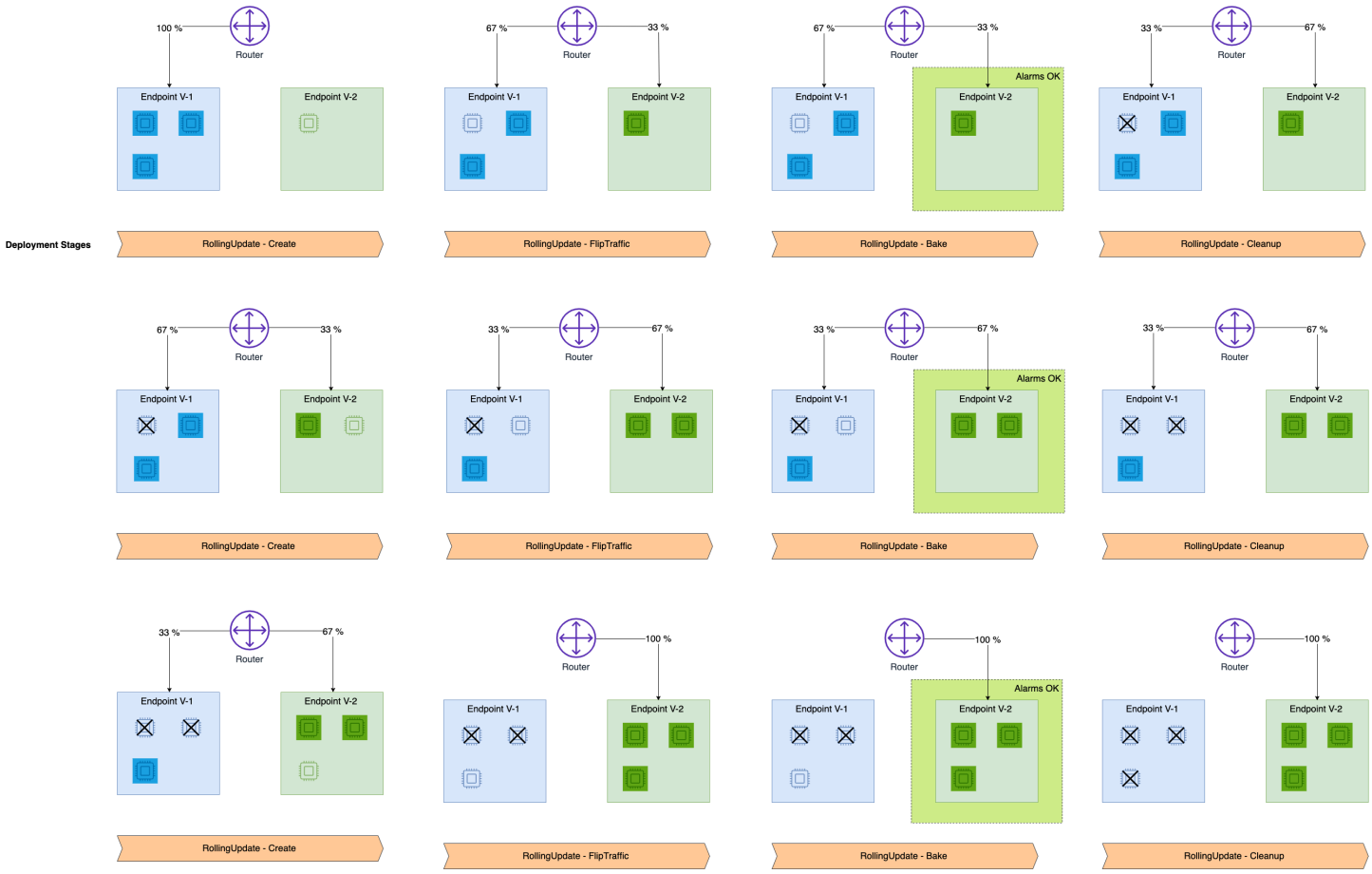
Cara kerjanya

Selama penyebaran bergulir, SageMaker menyediakan infrastruktur untuk mengalihkan lalu lintas dari armada lama ke armada baru tanpa harus menyediakan semua instance baru sekaligus. SageMakermenggunakan langkah-langkah berikut untuk menggeser lalu lintas:

1. SageMakerketentuan batch pertama contoh dalam armada baru.
2. Sebagian lalu lintas digeser dari instance lama ke batch pertama instance baru.
3. Setelah periode memanggang, jika tidak ada CloudWatch alarm Amazon yang tersandung, maka SageMaker bersihkan sekumpulan instance lama.
4. SageMakerterus menyediakan, menggeser, dan membersihkan instance dalam batch hingga penyebaran selesai.

Jika alarm tersandung selama salah satu periode memanggang, lalu lintas digulung kembali ke armada lama dalam batch ukuran yang Anda tentukan. Atau, Anda dapat menentukan penyebaran bergulir untuk menggeser 100% lalu lintas kembali ke armada lama jika alarm tersandung.

Diagram berikut menunjukkan perkembangan penyebaran bergulir yang berhasil, seperti yang dijelaskan pada langkah sebelumnya.



Untuk membuat penyebaran bergulir, Anda hanya perlu menentukan konfigurasi penyebaran yang Anda inginkan. Kemudian SageMaker menangani penyediaan instans baru, menghentikan instans lama, dan mengalihkan lalu lintas untuk Anda. Anda dapat membuat dan mengelola penyebaran Anda melalui CreateEndpointSageMakerAPI UpdateEndpointdan AWS Command Line Interface perintah yang ada dan.

Prasyarat

Sebelum menyiapkan penerapan bergulir, Anda harus membuat CloudWatch alarm Amazon untuk menonton metrik dari titik akhir Anda. Jika salah satu alarm berjalan selama periode memanggang, lalu lintas mulai bergulir kembali ke armada lama Anda. Untuk mempelajari cara mengatur CloudWatch alarm di titik akhir, lihat halaman prasyarat Konfigurasi dan Pemantauan Pengembalian Otomatis. Untuk mempelajari lebih lanjut tentang CloudWatch alarm, lihat Menggunakan CloudWatch alarm Amazon di CloudWatchPanduan Pengguna Amazon.

Juga, tinjau halaman Pengecualian untuk memastikan bahwa titik akhir Anda memenuhi persyaratan untuk penerapan bergulir.

Tentukan ukuran batch bergulir

Sebelum memperbarui titik akhir Anda, tentukan ukuran batch yang ingin Anda gunakan untuk mengalihkan lalu lintas ke armada baru secara bertahap.

Untuk rolling deployment, Anda dapat menentukan ukuran batch yaitu 5-50% dari kapasitas armada Anda. Jika Anda memilih ukuran batch yang besar, penyebaran selesai lebih cepat. Namun, perlu diingat bahwa titik akhir membutuhkan lebih banyak kapasitas saat memperbarui, kira-kira overhead ukuran batch. Jika Anda memilih ukuran batch yang lebih kecil, penyebaran membutuhkan waktu lebih lama, tetapi Anda menggunakan lebih sedikit kapasitas selama penerapan.

Mengkonfigurasi penerapan bergulir

Setelah Anda siap untuk deployment dan menyiapkan CloudWatch alarm untuk endpoint Anda, Anda dapat menggunakan SageMaker UpdateEndpointAPI atau perintah update-endpoint di untuk memulai penyebaran. AWS Command Line Interface

Cara memperbarui titik akhir

Contoh berikut menunjukkan bagaimana Anda dapat memperbarui titik akhir Anda dengan penyebaran bergulir menggunakan metode update_endpoint dari klien Boto3. SageMaker

Untuk mengkonfigurasi penyebaran bergulir, gunakan contoh dan bidang berikut:

- Untuk ituEndpointName, gunakan nama endpoint yang ada yang ingin Anda perbarui.
- Untuk ituEndpointConfigName, gunakan nama konfigurasi endpoint yang ingin Anda gunakan.
- Dalam AutoRollbackConfiguration objek, dalam Alarms bidang, Anda dapat menambahkan CloudWatch alarm Anda dengan nama. Buat satu AlarmName: <your-cw-alarm> entri untuk setiap alarm yang ingin Anda gunakan.
- Di bawahDeploymentConfig, untuk RollingUpdatePolicy objek, tentukan bidang berikut:
  - MaximumExecutionTimeoutInSeconds- Batas waktu untuk total penyebaran. Melebihi batas ini menyebabkan batas waktu. Nilai maksimum yang dapat Anda tentukan untuk bidang ini adalah 28800 detik, atau 8 jam.
  - WaitIntervalInSeconds— Panjang periode pemanggangan, di mana SageMaker memonitor alarm untuk setiap batch pada armada baru.
  - MaximumBatchSize- Tentukan Type batch yang ingin Anda gunakan (baik jumlah instance atau persentase keseluruhan armada Anda) danValue, atau ukuran setiap batch.

- `RollbackMaximumBatchSize`- Gunakan objek ini untuk menentukan strategi rollback jika alarm berjalan. Tentukan batch `Type` yang ingin Anda gunakan (baik jumlah instans atau persentase keseluruhan armada Anda), dan`Value`, atau ukuran setiap batch. Jika Anda tidak menentukan bidang ini, atau jika Anda menetapkan nilai ke 100% dari titik akhir Anda, SageMaker gunakan strategi rollback biru/hijau dan mengembalikan semua lalu lintas ke armada lama saat alarm berjalan.

```
import boto3
client = boto3.client("sagemaker")

response = client.update_endpoint(
    EndpointName="<your-endpoint-name>",
    EndpointConfigName="<your-config-name>",
    DeploymentConfig={
        "AutoRollbackConfiguration": {
            "Alarms": [
                {
                    "AlarmName": "<your-cw-alarm>"
                },
            ]
        },
        "RollingUpdatePolicy": {
            "MaximumExecutionTimeoutInSeconds": number,
            "WaitIntervalInSeconds": number,
            "MaximumBatchSize": {
                "Type": "INSTANCE_COUNT" | "CAPACITY_PERCENTAGE" (default),
                "Value": number
            },
            "RollbackMaximumBatchSize": {
                "Type": "INSTANCE_COUNT" | "CAPACITY_PERCENTAGE" (default),
                "Value": number
            },
        }
    }
)
```

Setelah memperbarui titik akhir Anda, Anda mungkin ingin memeriksa status penyebaran bergulir Anda dan memeriksa kesehatan titik akhir Anda. Anda dapat meninjau status endpoint Anda di SageMaker konsol, atau Anda dapat meninjau status titik akhir Anda dengan menggunakan API. DescribeEndpoint

Dalam `VariantStatus` objek yang dikembalikan oleh `DescribeEndpoint` API, `Status` bidang memberi tahu Anda penyebaran saat ini atau status operasional titik akhir Anda. Untuk informasi lebih lanjut tentang status yang mungkin dan apa artinya, lihat ProductionVariantStatus.

Jika Anda mencoba melakukan penerapan bergulir dan status titik akhir Anda`UpdateRollbackFailed`, lihat bagian berikut untuk bantuan pemecahan masalah.

Penanganan kegagalan

Jika penerapan bergulir Anda gagal dan rollback otomatis gagal juga, titik akhir Anda dapat dibiarkan dengan status. `UpdateRollbackFailed` Status ini berarti bahwa konfigurasi titik akhir yang berbeda diterapkan ke instans di belakang titik akhir Anda, dan titik akhir Anda dalam layanan dengan campuran konfigurasi endpoint lama dan baru.

Anda dapat melakukan panggilan lain ke UpdateEndpointAPI untuk mengembalikan titik akhir Anda ke keadaan sehat. Tentukan konfigurasi titik akhir dan konfigurasi penerapan yang Anda inginkan (baik sebagai penyebaran bergulir, penyebaran biru/hijau, atau keduanya) untuk memperbarui titik akhir Anda.

Anda dapat memanggil DescribeEndpointAPI untuk memeriksa kesehatan titik akhir Anda lagi, yang dikembalikan dalam `VariantStatus` objek sebagai `Status` bidang. Jika pembaruan Anda berhasil, titik akhir Anda `Status` kembali ke`InService`.

Pengecualian

Saat melakukan penyebaran biru/hijau atau bergulir, konfigurasi titik akhir baru Anda harus memiliki nama varian yang sama dengan konfigurasi titik akhir lama. Ada juga pengecualian berbasis fitur yang membuat titik akhir Anda tidak kompatibel dengan pagar pembatas penerapan saat ini. Jika titik akhir Anda menggunakan salah satu fitur berikut, Anda tidak dapat menggunakan pagar pembatas penerapan di titik akhir Anda, dan titik akhir Anda akan kembali menggunakan penerapan biru/hijau dengan pergeseran lalu lintas sekaligus dan tidak ada periode pemanggangan akhir:

- Kontainer Marketplace
- Endpoint yang menggunakan instans Inf1 (Inferentia-based)
- Titik akhir Amazon Elastic Inference

Jika Anda melakukan penerapan bergulir, ada pengecualian berbasis fitur tambahan:

- Titik akhir inferensi tanpa server

- Titik akhir inferensi multi-varian

Tes Bayangan

Dengan Amazon, SageMaker Anda dapat mengevaluasi perubahan apa pun pada infrastruktur penayangan model Anda dengan membandingkan kinerjanya dengan infrastruktur yang saat ini digunakan. Praktek ini dikenal sebagai pengujian bayangan. Pengujian bayangan dapat membantu Anda catch potensi kesalahan konfigurasi dan masalah kinerja sebelum memengaruhi pengguna akhir. Dengan SageMaker itu, Anda tidak perlu berinvestasi dalam membangun infrastruktur pengujian bayangan Anda, sehingga Anda dapat fokus pada pengembangan model.

Anda dapat menggunakan kemampuan ini untuk memvalidasi perubahan pada komponen apa pun dari varian produksi Anda, yaitu model, wadah, atau instans, tanpa dampak pengguna akhir. Hal ini berguna dalam situasi termasuk namun tidak terbatas pada hal-hal berikut:

- Anda mempertimbangkan untuk mempromosikan model baru yang telah divalidasi secara offline ke produksi, tetapi ingin mengevaluasi metrik kinerja operasional seperti latensi dan tingkat kesalahan sebelum membuat keputusan ini.
- Anda sedang mempertimbangkan perubahan pada wadah infrastruktur penayangan Anda, seperti menambal kerentanan atau meningkatkan ke versi yang lebih baru, dan ingin menilai dampak perubahan ini sebelum promosi ke produksi.
- Anda mempertimbangkan untuk mengubah instans ML-mu dan ingin mengevaluasi kinerja instans baru dengan permintaan inferensi langsung.

SageMakerKonsol memberikan pengalaman terpandu untuk mengelola alur kerja pengujian bayangan. Anda dapat mengatur pengujian bayangan untuk durasi waktu yang telah ditentukan, memantau kemajuan pengujian melalui dasbor langsung, membersihkan setelah selesai, dan menindaklanjuti hasilnya. Pilih varian produksi yang ingin Anda uji, dan SageMaker secara otomatis menerapkan varian baru dalam mode bayangan dan merutekan salinan permintaan inferensi ke sana secara real time dalam titik akhir yang sama. Hanya tanggapan varian produksi yang dikembalikan ke aplikasi panggilan. Anda dapat memilih untuk membuang atau mencatat respons varian bayangan untuk perbandingan offline. Untuk informasi lebih lanjut tentang varian produksi dan bayangan, lihatAman memvalidasi model dalam produksi. Perhatikan bahwa jika endpoint Anda menggunakan salah satu fitur yang tercantum dalam Pengecualian halaman, Anda tidak dapat menggunakan tes bayangan.

Lihat Buat tes bayangan petunjuk tentang membuat tes bayangan.

Buat tes bayangan

Anda dapat membuat tes bayangan untuk membandingkan kinerja varian bayangan dengan varian produksi. Anda dapat menjalankan pengujian pada titik akhir yang ada yang menyajikan permintaan inferensi atau Anda dapat membuat titik akhir baru untuk menjalankan pengujian.

Untuk membuat tes bayangan, Anda perlu menentukan yang berikut:

- Varian produksi yang menerima dan menanggapi 100 persen permintaan inferensi yang masuk.
- Varian bayangan yang menerima persentase permintaan yang masuk, direplikasi dari varian produksi, tetapi tidak mengembalikan tanggapan apa pun.

Untuk setiap varian, Anda dapat menggunakan SageMaker untuk mengontrol model, jenis instance, dan jumlah instance. Anda dapat mengonfigurasi persentase permintaan masuk, yang dikenal sebagai persentase sampling lalu lintas, yang ingin direplikasi ke varian bayangan Anda. SageMaker mengelola replikasi permintaan ke varian bayangan Anda dan Anda dapat memodifikasi persentase sampling lalu lintas saat pengujian dijadwalkan atau berjalan. Anda juga dapat mengaktifkan Pengambilan Data secara opsional untuk mencatat permintaan dan respons varian produksi dan bayangan Anda.

> **Note**
>
> SageMaker mendukung maksimal satu varian bayangan per titik akhir. Untuk titik akhir dengan varian bayangan, bisa ada maksimal satu varian produksi.

Anda dapat menjadwalkan tes untuk memulai kapan saja dan melanjutkan untuk durasi tertentu. Durasi default adalah 7 hari dan maksimum adalah 30 hari. Setelah pengujian selesai, titik akhir kembali ke keadaan sebelum memulai pengujian. Ini memastikan bahwa Anda tidak perlu membersihkan sumber daya secara manual setelah menyelesaikan tes.

Anda dapat memantau pengujian yang berjalan melalui dasbor di SageMaker konsol. Dasbor menyediakan perbandingan berdampingan metrik pemanggilan dan metrik instance antara varian produksi dan bayangan, bersama dengan tampilan tabel dengan statistik metrik yang relevan. Dasbor ini juga tersedia untuk pengujian yang diselesaikan. Setelah meninjau metrik, Anda dapat memilih untuk mempromosikan varian bayangan menjadi varian produksi baru atau mempertahankan varian produksi yang ada. Setelah Anda mempromosikan varian bayangan, itu merespons semua permintaan yang masuk. Untuk informasi selengkapnya, lihat Promosikan sebuah varian bayangan.

Prosedur berikut menjelaskan cara membuat tes bayangan melalui SageMaker konsol. Ada variasi dalam alur kerja tergantung pada apakah Anda ingin menggunakan titik akhir yang ada atau untuk membuat titik akhir baru untuk pengujian bayangan.

Topik

Prasyarat

Sebelum membuat tes bayangan dengan SageMaker konsol, Anda harus memiliki SageMaker model yang siap digunakan. Untuk informasi selengkapnya tentang cara membuat SageMaker model, lihatTerapkan model untuk inferensi waktu nyata.

Anda dapat memulai dengan pengujian bayangan dengan titik akhir yang ada dengan varian produksi dan varian bayangan, titik akhir yang ada hanya dengan varian produksi, atau hanya SageMaker model yang ingin Anda bandingkan. Tes bayangan mendukung pembuatan titik akhir dan menambahkan varian sebelum pengujian Anda dimulai.

Masukkan detail tes bayangan

Untuk mulai membuat tes bayangan Anda, isi halaman Enter shadow test details dengan melakukan hal berikut:

1. Buka konsol SageMaker .
2. Di panel navigasi kiri, pilih Inferensi, lalu pilih Tes bayangan.
3. Pilih Buat tes bayangan.
4. Di bawah Nama, masukkan nama untuk pengujian.
5. (Opsional) Di bawah Deskripsi, masukkan deskripsi untuk pengujian.
6. (Opsional) Tentukan Tag menggunakan pasangan Kunci dan Nilai.
7. Pilih Berikutnya.

Masukkan pengaturan uji bayangan

Setelah mengisi halaman Enter shadow test details, isi halaman Enter shadow test settings. Jika Anda sudah memiliki endpoint SageMaker Inference dan varian produksi, ikuti alur kerja Use an existing endpoint. Jika Anda belum memiliki endpoint, ikuti alur kerja Create a new endpoint.

Use an existing endpoint

Jika Anda ingin menggunakan endpoint yang ada untuk pengujian Anda, isi halaman Enter shadow test settings dengan melakukan hal berikut:

1. Pilih peran yang memiliki kebijakan `AmazonSageMakerFullAccess` IAM terlampir.
2. Pilih Gunakan titik akhir yang ada, lalu pilih salah satu titik akhir yang tersedia.
3. (Opsional) Untuk mengenkripsi volume penyimpanan pada titik akhir Anda, pilih kunci KMS yang ada atau pilih Masukkan ARN kunci KMS dari daftar tarik-turun di bawah Kunci enkripsi. Jika Anda memilih opsi kedua, bidang untuk memasukkan ARN kunci KMS muncul. Masukkan ARN kunci KMS di bidang itu.
4. Jika Anda memiliki beberapa varian produksi di belakang titik akhir itu, hapus yang tidak ingin Anda gunakan untuk pengujian. Anda dapat menghapus varian model dengan memilihnya dan kemudian memilih Hapus.
5. Jika Anda belum memiliki varian bayangan, tambahkan varian bayangan. Untuk menambahkan varian bayangan, lakukan hal berikut:

    a. Pilih Tambahkan.

    b. Pilih varian Shadow.

    c. Dalam kotak dialog Add model, pilih model yang ingin Anda gunakan untuk varian bayangan Anda.

    d. Pilih Simpan.
6. (Opsional) Pada langkah sebelumnya, varian bayangan ditambahkan dengan pengaturan default. Untuk mengubah pengaturan ini, pilih varian bayangan dan pilih Edit. Kotak dialog Edit shadow varian muncul. Untuk informasi selengkapnya tentang mengisi kotak dialog ini, lihatMengedit sebuah tes bayangan.
7. Di bagian Jadwal, masukkan durasi tes dengan melakukan hal berikut:

    a. Pilih kotak di bawah Durasi. Kalender popup muncul.

    b. Pilih tanggal mulai dan berakhir dari kalender, atau masukkan tanggal mulai dan berakhir di bidang untuk Tanggal mulai dan Tanggal akhir, masing-masing.

c. (Opsional) Untuk bidang Waktu mulai dan Waktu akhir, masukkan waktu mulai dan akhir, masing-masing, dalam format 24 jam.

d. Pilih Terapkan.

Durasi minimum adalah 1 jam, dan durasi maksimum adalah 30 hari.

8. (Opsional) Aktifkan pengambilan data untuk menyimpan permintaan inferensi dan informasi respons dari titik akhir Anda ke bucket Amazon S3, lalu masukkan lokasi bucket Amazon S3.
9. Pilih Buat tes bayangan.

Create a new endpoint

Jika Anda tidak memiliki titik akhir yang ada, atau Anda ingin membuat titik akhir baru untuk pengujian Anda, isi halaman Enter shadow test settings dengan melakukan hal berikut:

1. Pilih peran yang memiliki kebijakan `AmazonSageMakerFullAccess` IAM terlampir.
2. Pilih Buat titik akhir baru.
3. Di bawah Nama, masukkan nama untuk titik akhir.
4. Tambahkan satu varian produksi dan satu varian bayangan ke titik akhir:
   - Untuk menambahkan varian produksi pilih Tambah, lalu pilih varian Produksi. Dalam kotak dialog Tambah model, pilih model yang ingin Anda gunakan untuk varian produksi Anda, lalu pilih Simpan.
   - Untuk menambahkan varian bayangan pilih Tambah, lalu pilih varian Shadow. Dalam kotak dialog Tambah model, pilih model yang ingin Anda gunakan untuk varian bayangan Anda, lalu pilih Simpan.
5. (Opsional) Pada langkah sebelumnya, varian bayangan ditambahkan dengan pengaturan default. Untuk mengubah pengaturan ini, pilih varian bayangan dan pilih Edit. Kotak dialog Edit shadow varian muncul. Untuk informasi selengkapnya tentang mengisi kotak dialog ini, lihatMengedit sebuah tes bayangan.
6. Di bagian Jadwal, masukkan durasi tes dengan melakukan hal berikut:

   a. Pilih kotak di bawah Durasi. Kalender popup muncul.

   b. Pilih tanggal mulai dan berakhir dari kalender, atau masukkan tanggal mulai dan berakhir di bawah Tanggal mulai dan Tanggal akhir, masing-masing.

c. (Opsional) Di bawah Waktu mulai dan Waktu akhir, masukkan waktu mulai dan akhir, masing-masing, dalam format 24 jam.

d. Pilih Terapkan.

Durasi minimum adalah 1 jam, dan durasi maksimum adalah 30 hari.

7. (Opsional) Aktifkan pengambilan data untuk menyimpan permintaan inferensi dan informasi respons dari titik akhir Anda ke bucket Amazon S3, lalu masukkan lokasi bucket Amazon S3.
8. Pilih Buat tes bayangan.

Setelah menyelesaikan prosedur sebelumnya, Anda sekarang harus memiliki tes yang dijadwalkan untuk dimulai pada tanggal dan waktu mulai yang ditentukan. Anda dapat melihat kemajuan tes dari dasbor. Untuk informasi selengkapnya tentang melihat tes Anda dan tindakan yang dapat Anda lakukan, lihatMelihat, memantau, dan mengedit tes bayangan.

Melihat, memantau, dan mengedit tes bayangan

Anda dapat melihat status pengujian bayangan Anda, memantau kemajuannya dari dasbor, dan melakukan tindakan, seperti memulai atau menghentikan pengujian lebih awal atau menghapus pengujian. Bagian berikut menunjukkan bagaimana Anda dapat melihat dan memodifikasi pengujian bayangan Anda menggunakan SageMaker konsol.

Topik

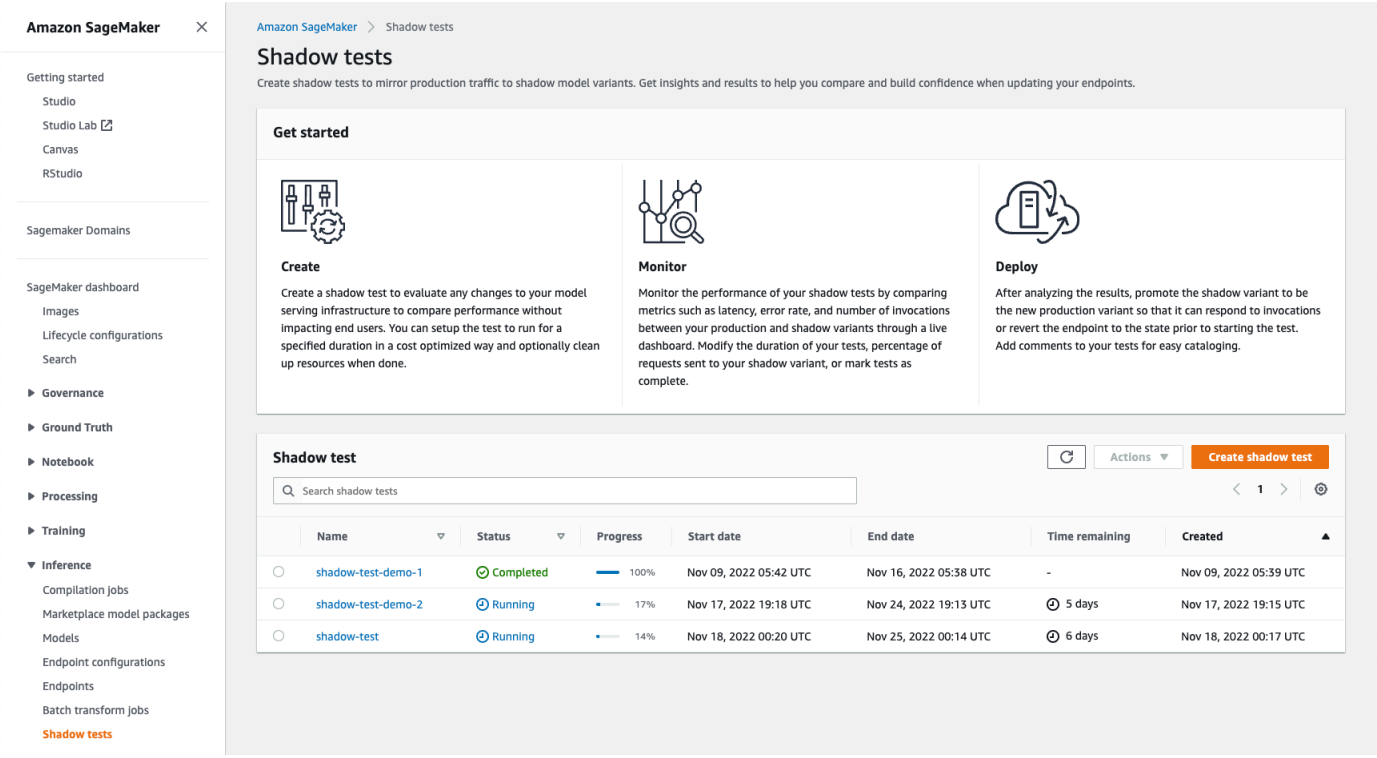
- Lihat tes bayangan
- Pantau tes bayangan
- Mulai tes bayangan lebih awal
- Selesaikan tes bayangan lebih awal
- Hapus sebuah tes bayangan
- Mengedit sebuah tes bayangan

Lihat tes bayangan

Anda dapat melihat status semua tes bayangan Anda pada halaman pengujian Bayangan di SageMaker konsol.

Untuk melihat pengujian Anda di konsol tersebut, lakukan hal berikut:

1. Buka konsol SageMaker .
2. Di panel navigasi, pilih Inferensi.
3. Pilih Tes bayangan untuk melihat halaman yang mencantumkan semua pengujian bayangan Anda. Halaman akan terlihat seperti screenshot berikut, dengan semua tes yang tercantum di bawah bagian uji Bayangan.



Anda dapat melihat status pengujian di konsol pada halaman pengujian Bayangan dengan memeriksa bidang Status untuk pengujian.

Status yang mungkin muncul adalah sebagai berikut:

- `Creating`- SageMaker adalah menciptakan tes Anda.
- `Created`— SageMaker telah selesai membuat tes Anda, dan itu akan dimulai pada waktu yang dijadwalkan.
- `Updating`- Saat Anda membuat perubahan pada pengujian Anda, pengujian Anda ditampilkan sebagai pembaruan.
- `Starting`— SageMaker adalah memulai tes Anda.
- `Running`— Tes Anda sedang berlangsung.
- `Stopping`- SageMaker adalah menghentikan tes Anda.

- `Completed`— Tes Anda telah selesai.
- `Cancelled`- Ketika Anda menyimpulkan tes Anda lebih awal, itu menunjukkan sebagai dibatalkan.

Pantau tes bayangan

Anda dapat melihat detail tes bayangan dan memantaunya saat sedang berlangsung atau setelah selesai. SageMaker menyajikan dasbor langsung yang membandingkan metrik operasional seperti latensi model, dan tingkat kesalahan yang dikumpulkan, dari varian produksi dan bayangan.

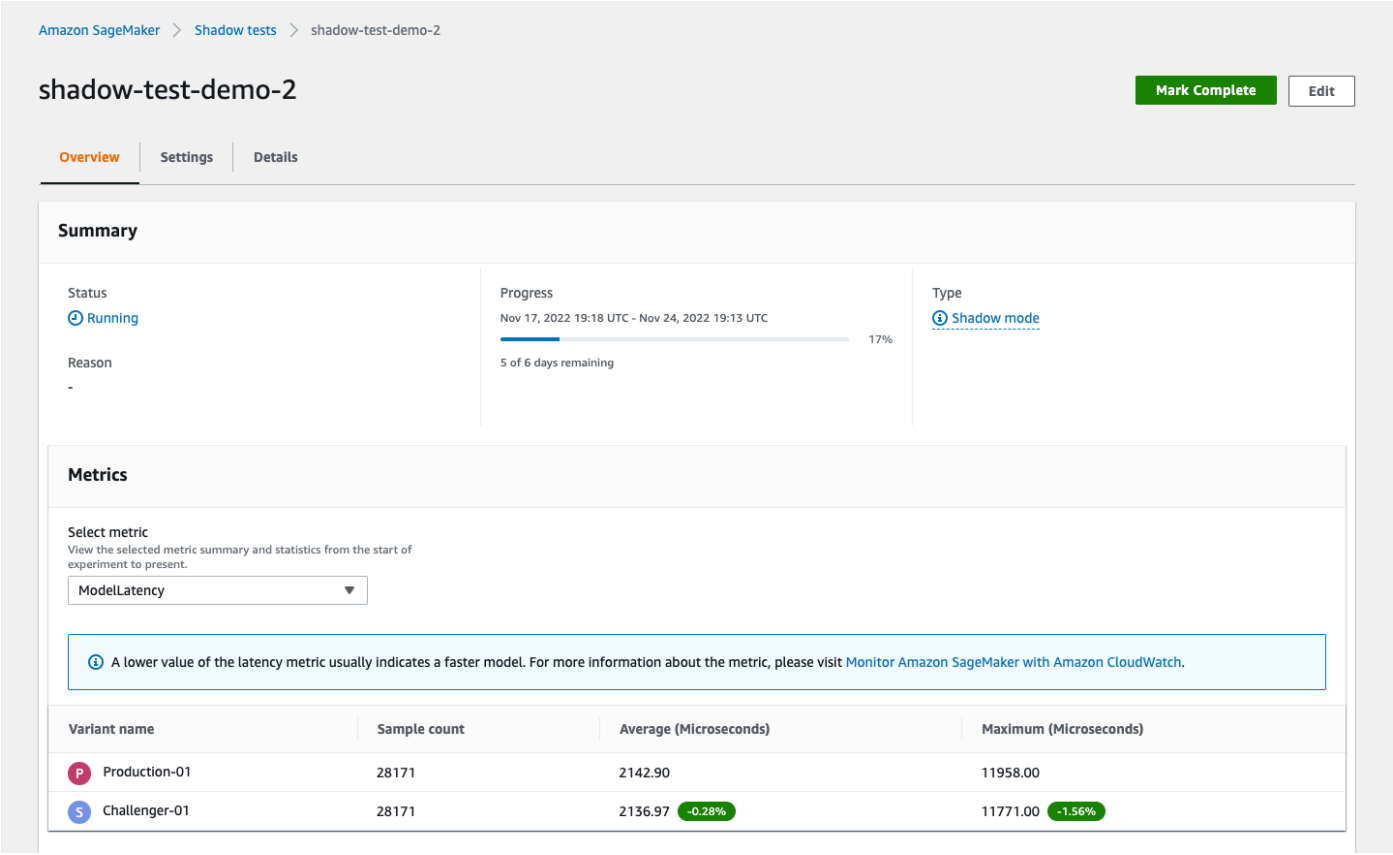
Untuk melihat bagan pengujian individual di konsol tersebut, lakukan hal berikut:

1. Pilih tes yang ingin Anda pantau dari bagian Shadow test pada halaman Shadow tests.
2. Dari daftar dropdown Tindakan, pilih Lihat. Halaman ikhtisar dengan detail pengujian dan dasbor metrik muncul.

Halaman ikhtisar memiliki tiga bagian berikut.

Ringkasan

Bagian ini merangkum kemajuan dan status tes. Ini juga menunjukkan statistik ringkasan metrik yang dipilih dari daftar dropdown Pilih metrik di ayat Metrik. Tangkapan layar berikut menunjukkan bagian ini.



Pada tangkapan layar sebelumnya, tab Pengaturan, dan Detail menampilkan pengaturan yang Anda pilih, dan detail yang Anda masukkan saat membuat pengujian.

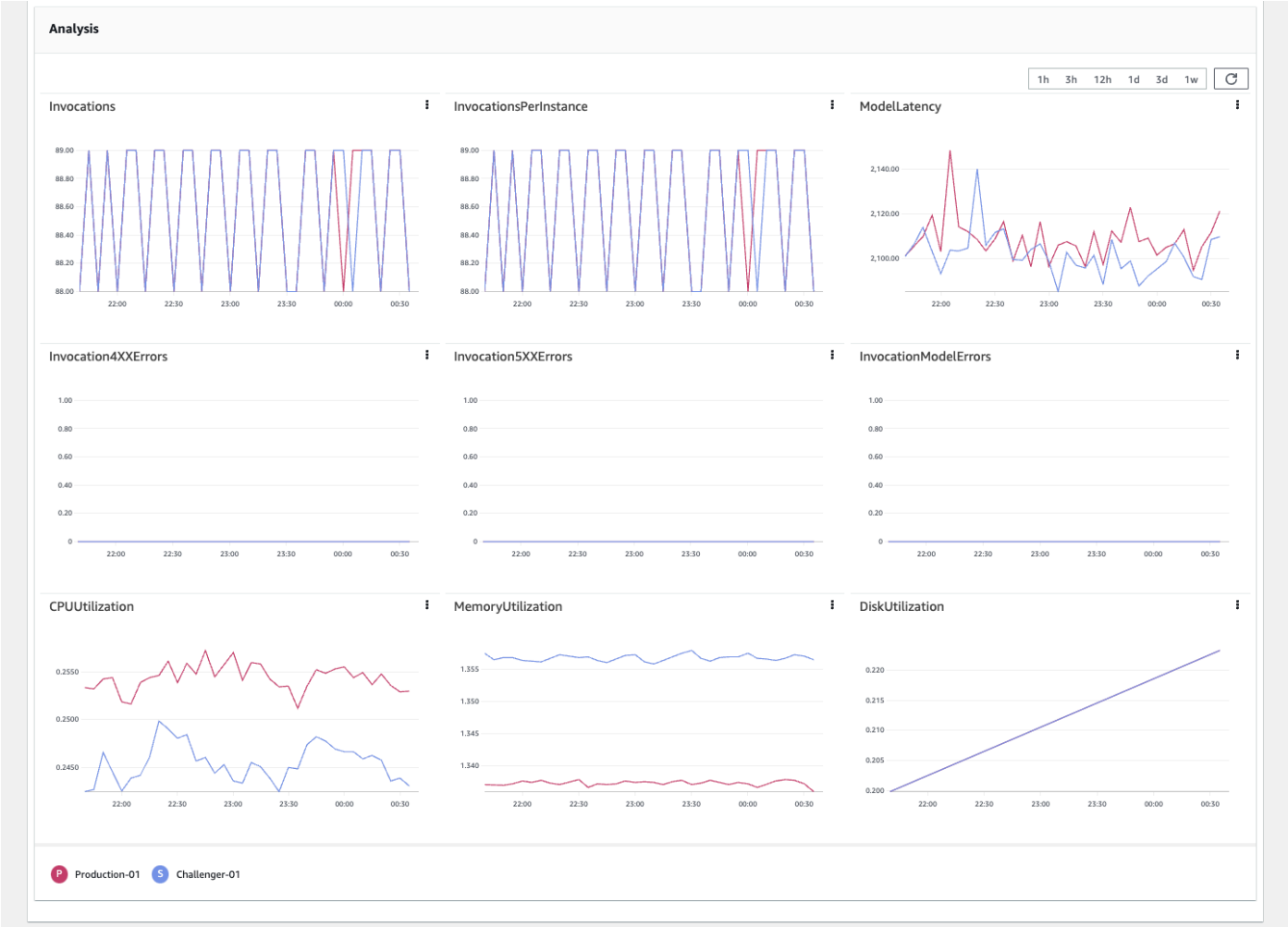
Analisa

Bagian ini menunjukkan dasbor metrik dengan bagan terpisah untuk metrik berikut ini:

- `Invocations`
- `InvocationsPerInstance`
- `ModelLatency`
- `Invocation4XXErrors`
- `Invocation5XXErrors`
- `InvocationModelErrors`
- `CPUUtilization`
- `MemoryUtilization`
- `DiskUtilization`

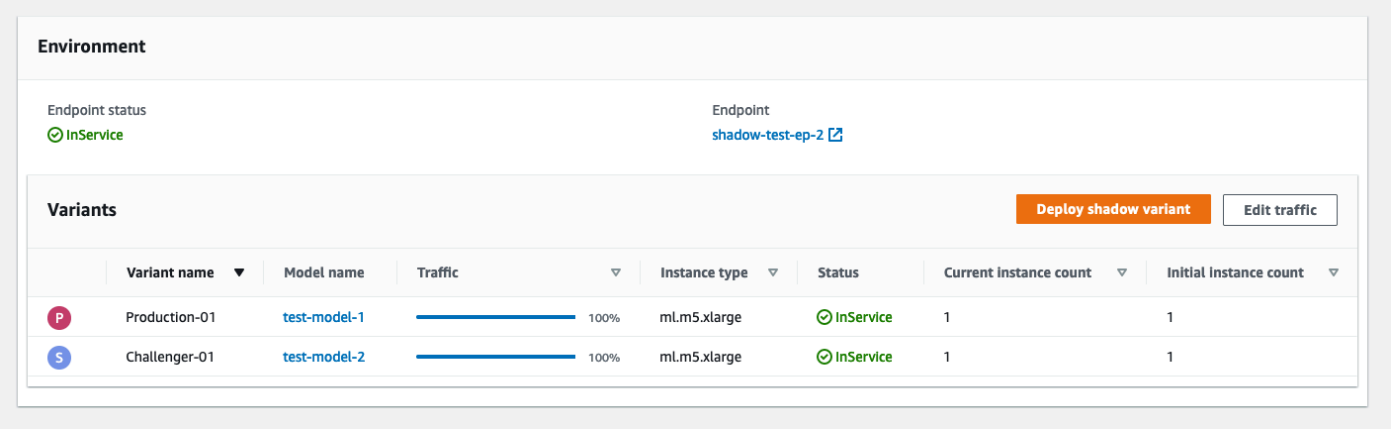
Tiga metrik terakhir memantau penggunaan sumber daya runtime kontainer model. Sisanya adalah CloudWatch metrik yang dapat Anda gunakan untuk menganalisis performa varian Anda.

Secara umum, lebih sedikit kesalahan menunjukkan model yang lebih stabil. Latensi yang lebih rendah menunjukkan model yang lebih cepat atau infrastruktur yang lebih cepat. Untuk informasi selengkapnya tentang CloudWatch metrik, lihatSageMaker Metrik Pemanggilan Titik Akhir. Tangkapan layar berikut menunjukkan dasbor metrik.



Environment

Bagian ini menunjukkan varian yang Anda bandingkan dalam pengujian. Jika Anda puas dengan performa varian bayangan, berdasarkan metrik yang disebutkan di atas, Anda dapat mempromosikan varian bayangan ke produksi, dengan memilih varian Deploy shadow. Untuk detail selengkapnya tentang menerapkan varian bayangan, lihatPromosikan sebuah varian bayangan. Anda juga dapat mengubah persentase pengambilan sampel lalu lintas, dan melanjutkan pengujian, dengan memilih Edit lalu lintas. Untuk detail selengkapnya tentang mengedit varian bayangan, lihatMengedit sebuah tes bayangan. Tangkapan layar berikut menunjukkan bagian ini.



Mulai tes bayangan lebih awal

Anda dapat memulai pengujian sebelum waktu mulai yang dijadwalkan. Jika durasi pengujian baru melebihi 30 hari, SageMaker secara otomatis menetapkan akhir pengujian menjadi 30 hari setelah waktu mulai yang baru. Tindakan ini segera memulai tes. Jika Anda ingin mengubah waktu mulai atau akhir tes, lihatMengedit sebuah tes bayangan.

Untuk segera memulai pengujian Anda, sebelum waktu mulai yang dijadwalkan, melalui konsol, lakukan hal berikut:

1. Pilih tes yang ingin Anda mulai segera dari bagian Shadow test pada halaman Shadow tests.
2. Dari daftar dropdown Tindakan, pilih Mulai. Tes bayangan Start? kotak dialog muncul.
3. Pilih Mulai sekarang.

Selesaikan tes bayangan lebih awal

Anda dapat menyelesaikan tes yang sedang berjalan sebelum akhir durasi yang dijadwalkan. Untuk informasi selengkapnya, lihat Selesaikan tes bayangan lebih awal.

Hapus sebuah tes bayangan

Anda dapat menghapus tes yang tidak lagi Anda perlukan. Menghapus pengujian Anda hanya akan menghapus metadata pengujian dan bukan titik akhir, varian, atau data yang diambil di Amazon S3. Jika Anda ingin endpoint Anda berhenti berjalan, Anda harus menghapus titik akhir Anda. Untuk informasi selengkapnya tentang penghapusan titik akhir, lihatHapus Titik Akhir dan Sumber Daya

Untuk menghapus tes melalui konsol, lakukan hal berikut:

1. Pilih tes yang ingin Anda hapus dari bagian Shadow test pada halaman Shadow tests.
2. Dari daftar dropdown Tindakan, pilih Hapus. Kotak dialog Delete shadow test muncul.
3. Dalam Untuk mengkonfirmasi penghapusan, ketik hapus di bidang. kotak teks, masukkan**`delete`**.
4. Pilih Delete (Hapus).

Mengedit sebuah tes bayangan

Anda dapat memodifikasi tes terjadwal dan dalam proses. Sebelum pengujian dimulai, Anda dapat mengubah deskripsi, konfigurasi varian bayangan, tanggal mulai, dan tanggal akhir pengujian. Anda juga dapat mengaktifkan atau mematikan pengambilan data.

Setelah pengujian dimulai, Anda hanya dapat mengubah deskripsi, persentase sampling lalu lintas untuk varian bayangan, dan tanggal akhir.

Untuk mengedit rincian pengujian Anda melalui konsol tersebut, lakukan hal berikut:

1. Pilih tes yang ingin Anda edit dari bagian Shadow test pada halaman Shadow tests.
2. Dari daftar dropdown Tindakan, pilih Edit. Halaman rincian uji Enter shadow akan muncul.
3. (Opsional) Di bawah Deskripsi, masukkan bagan pengujian Anda.
4. Pilih Selanjutnya. Halaman pengaturan uji Enter shadow muncul.
5. (Opsional) Untuk mengedit varian bayangan Anda, lakukan hal berikut:

    a. Pilih varian bayangan dan pilih Edit. Kotak dialog Edit bayangan varian muncul. Jika pengujian Anda sudah dimulai, maka Anda hanya dapat mengubah persentase pengambilan sampel lalu lintas.

    b. (Opsional) Di bawah Nama, masukkan nama baru untuk mengganti nama lama.

    c. (Opsional) Di bawah sampel Lalu Lintas, masukkan persentase pengambilan sampel lalu lintas baru untuk menggantikan persentase pengambilan sampel lalu lintas lama.

    d. (Opsional) Di bawah Jenis instans, pilih jenis instance baru dari daftar dropdown.

    e. (Opsional) Di bawah jumlah Instance, masukkan jumlah instance baru untuk menggantikan jumlah instance lama.

    f. Pilih Apply (Terapkan).

Anda tidak dapat mengubah model dalam varian bayangan Anda menggunakan prosedur di atas. Jika Anda ingin mengubah model, pertama-tama hapus varian bayangan dengan memilihnya dan memilih Remove. Kemudian tambahkan varian bayangan baru.

6. (Opsional) Untuk mengedit bagan tes, lakukan hal berikut:

   a. Pilih kotak di bawah Durasi di bagian Jadwal. Sebuah kalender popup muncul.

   b. Jika pengujian Anda belum dimulai, Anda dapat mengubah tanggal mulai dan berakhir. Pilih tanggal mulai dan akhir baru dari kalender, atau masukkan tanggal mulai dan akhir baru di bawah Tanggal mulai dan Tanggal akhir, masing-masing.

      Jika pengujian Anda sudah dimulai, Anda hanya dapat mengubah tanggal akhir. Masukkan tanggal akhir baru di bawah Tanggal akhir.

   c. (Opsional) Jika pengujian Anda belum dimulai, Anda dapat mengubah waktu mulai dan akhir. Masukkan waktu mulai dan akhir yang baru di bawah Waktu mulai, dan Waktu akhir, masing-masing, dalam format 24 jam.

      Jika tes Anda sudah dimulai, Anda hanya dapat mengubah waktu akhir. Masukkan waktu akhir baru di bawah Waktu akhir, dalam format 24 jam.

   d. Pilih Apply (Terapkan).

7. (Opsional) Aktifkan atau nonaktifkan Aktifkan pengambilan data.

8. Pilih Perbarui tes bayangan.

Menyelesaikan sebuah tes bayangan

Pengujian Anda secara otomatis selesai pada akhir durasi yang dijadwalkan, atau Anda dapat menghentikan pengujian yang sedang berlangsung lebih awal. Setelah pengujian Anda selesai, status pengujian di bagian Shadow tests pada halaman Shadow tests menunjukkan sebagai Complete. Kemudian Anda dapat meninjau dan menganalisis metrik akhir pengujian Anda.

Anda dapat menggunakan dasbor metrik untuk memutuskan apakah akan mempromosikan varian bayangan ke produksi. Untuk informasi selengkapnya tentang menganalisis dasbor metrik pengujian Anda, lihatPantau tes bayangan.

Untuk petunjuk tentang cara menyelesaikan pengujian Anda sebelum akhir waktu penyelesaian yang dijadwalkan, lihatSelesaikan tes bayangan lebih awal.

Untuk petunjuk tentang mempromosikan varian bayangan Anda ke produksi, lihatPromosikan sebuah varian bayangan.

Selesaikan tes bayangan lebih awal

Salah satu alasan Anda mungkin ingin menyelesaikan tes bayangan dalam proses adalah jika Anda telah memutuskan bahwa metrik untuk varian bayangan Anda terlihat bagus dan Anda ingin mempromosikannya ke produksi. Anda mungkin juga memutuskan untuk menyelesaikan tes jika satu atau lebih varian tidak berkinerja baik.

Untuk menyelesaikan tes sebelum tanggal akhir yang dijadwalkan, lakukan hal berikut:

1. Pilih tes yang ingin Anda tandai selesai dari bagian Shadow tests pada halaman Shadow tests.
2. Dari daftar dropdown Tindakan, pilih Complete, dan kotak dialog Complete shadow test muncul.
3. Di kotak dialog, pilih salah satu opsi berikut:

   - Ya, gunakan varian bayangan
   - Tidak, hapus varian bayangan
4. (Opsional) Dalam Komentar kotak teks, masukkan alasan Anda untuk menyelesaikan tes sebelum waktu akhir yang dijadwalkan.
5. 1. Jika Anda memutuskan untuk menyebarkan varian bayangan, pilih Complete dan lanjutkan untuk menyebarkan. Halaman varian Deploy shadow muncul. Untuk petunjuk tentang cara mengisi halaman ini, lihatPromosikan sebuah varian bayangan.
   2. Jika Anda memutuskan untuk menghapus varian bayangan, pilih Konfirmasi.

Promosikan sebuah varian bayangan

Jika Anda memutuskan ingin mengganti varian produksi dengan varian bayangan, Anda dapat memperbarui titik akhir dan mempromosikan varian bayangan untuk merespons permintaan inferensi. Ini menghapus varian produksi Anda saat ini dari produksi dan menggantikannya dengan varian bayangan Anda.

Jika tes bayangan Anda masih dalam proses, Anda harus terlebih dahulu menyelesaikan tes Anda. Untuk menyelesaikan tes bayangan Anda sebelum akhir yang dijadwalkan, ikuti petunjukSelesaikan tes bayangan lebih awal sebelum melanjutkan dengan bagian ini.

Saat Anda mempromosikan varian bayangan ke produksi, Anda memiliki opsi berikut untuk jumlah instance varian bayangan.

- Anda dapat mempertahankan jumlah dan jenis instans dari varian produksi. Jika Anda memilih opsi ini, maka varian bayangan Anda diluncurkan dalam produksi dengan jumlah instans saat ini, memastikan bahwa model Anda dapat terus memproses lalu lintas permintaan pada skala yang sama.
- Anda dapat mempertahankan jumlah instans dan jenis varian bayangan Anda. Jika Anda ingin menggunakan opsi ini, kami sarankan Anda melakukan tes bayangan dengan pengambilan sampel lalu lintas 100 persen untuk memastikan bahwa varian bayangan dapat memproses lalu lintas permintaan pada skala saat ini.
- Anda dapat menggunakan nilai kustom untuk jumlah dan jenis instance. Jika Anda ingin menggunakan opsi ini, kami sarankan Anda melakukan tes bayangan dengan pengambilan sampel lalu lintas 100 persen untuk memastikan bahwa varian bayangan dapat memproses lalu lintas permintaan pada skala saat ini.

Kecuali Anda memvalidasi jenis instans atau hitungan atau kedua varian bayangan, kami sangat menyarankan agar Anda mempertahankan jumlah dan jenis instans dari varian produksi saat mempromosikan varian bayangan Anda.

Untuk mempromosikan varian bayangan, lakukan hal berikut:

1. Jika tes Anda selesai, lakukan hal berikut:

   a. Pilih tes dari bagian Shadow test pada halaman Shadow tests.
   b. Dari daftar dropdown Tindakan, pilih Lihat. Dasbor akan muncul.
   c. Pilih varian Deploy shadow di bagian Environment. Halaman varian Deploy shadow muncul.

   Jika tes Anda belum selesai, lihatSelesaikan tes bayangan lebih awal untuk menyelesaikannya.
2. Di bagian Pengaturan varian, pilih salah satu opsi berikut:

   - Pertahankan pengaturan produksi
   - Pertahankan pengaturan bayangan
   - Pengaturan instans khusus

   Jika Anda memilih Pengaturan instans khusus, lakukan hal berikut:

   a. Pilih jenis instans dari daftar dropdown tipe Instance.
   b. Di bawah Jumlah instans, masukkan jumlah instans.

3. Dalam Masukkan 'deploy' untuk mengkonfirmasi kotak teks penyebaran, masukkan**`deploy`**.
4. Pilih varian Deploy shadow.

Titik akhir SageMaker Inferensi Anda sekarang menggunakan varian bayangan sebagai varian produksi Anda, dan varian produksi Anda telah dihapus dari titik akhir.

Praktik terbaik

Saat membuat eksperimen inferensi, perhatikan informasi berikut:

- Persentase pengambilan sampel lalu lintas — Mengambil sampel 100 persen permintaan inferensi memungkinkan Anda memvalidasi bahwa varian bayangan Anda dapat menangani lalu lintas produksi saat dipromosikan. Anda dapat memulai dengan persentase pengambilan sampel lalu lintas yang lebih rendah dan melakukan dial up saat Anda mendapatkan kepercayaan pada varian Anda, tetapi praktik terbaik adalah memastikan bahwa Anda telah meningkatkan lalu lintas menjadi 100 persen sebelum promosi.
- Jenis instans — Kecuali Anda menggunakan varian bayangan untuk mengevaluasi jenis atau ukuran instans alternatif, sebaiknya gunakan jenis, ukuran, dan hitungan instans yang sama sehingga Anda dapat yakin bahwa varian bayangan Anda dapat menangani volume permintaan inferensi setelah Anda mempromosikannya.
- Penskalaan otomatis - Untuk memastikan bahwa varian bayangan Anda dapat merespons lonjakan dalam jumlah permintaan inferensi atau perubahan pola permintaan inferensi, kami sangat menyarankan Anda mengonfigurasi penskalaan otomatis pada varian bayangan Anda. Untuk mempelajari cara mengonfigurasi penskalaan otomatis, lihatSecara Otomatis Menskalakan SageMaker Model Amazon. Jika telah mengonfigurasi penskalaan otomatis, Anda juga dapat memvalidasi perubahan pada kebijakan penskalaan otomatis tanpa menimbulkan dampak bagi pengguna.
- Pemantauan metrik — Setelah Anda memulai eksperimen bayangan dan memiliki pemanggilan yang memadai, pantau dasbor metrik untuk memastikan bahwa metrik seperti latensi dan tingkat kesalahan berada dalam batas yang dapat diterima. Ini membantu Anda catch kesalahan konfigurasi lebih awal dan mengambil tindakan korektif. Untuk informasi tentang cara memantau metrik eksperimen inferensi yang sedang berlangsung, lihatMelihat, memantau, dan mengedit tes bayangan.

Pengecualian

Ada pengecualian berbasis fitur yang membuat titik akhir Anda tidak kompatibel dengan pengujian bayangan saat ini. Jika titik akhir Anda menggunakan salah satu fitur berikut, Anda tidak dapat menggunakan pengujian bayangan di titik akhir, dan permintaan Anda untuk menyiapkan pengujian bayangan akan menyebabkan kesalahan validasi.

- Inferensi Inferensi tanpa server
- Inferensi Asinkron
- Kontainer Marketplace
- Titik akhir beberapa kontainer
- Titik akhir multi-model
- Titik akhir yang menggunakan instans Inf1 (berbasis Inferensi)
- Titik akhir inferensi Amazon Elastic Inference

Akses kontainer melalui SSM

Amazon SageMaker memungkinkan Anda terhubung dengan aman ke container Docker tempat model Anda digunakan untuk Inference menggunakan AWS Systems Manager (SSM). Ini memberi Anda akses tingkat shell ke wadah sehingga Anda dapat men-debug proses yang berjalan di dalam wadah dan mencatat perintah dan tanggapan dengan Amazon CloudWatch. Anda juga dapat mengatur AWS PrivateLink koneksi ke instance ML yang meng-host kontainer Anda untuk mengakses kontainer melalui SSM secara pribadi.

> **Warning**
>
> Mengaktifkan akses SSM dapat memengaruhi kinerja titik akhir Anda. Kami merekomendasikan penggunaan fitur ini dengan titik akhir dev atau pengujian Anda dan bukan dengan titik akhir dalam produksi. Selain itu, SageMaker secara otomatis menerapkan patch keamanan, dan mengganti atau menghentikan instans endpoint yang salah dalam waktu 10 menit. Namun untuk titik akhir dengan varian produksi yang diaktifkan SSM, SageMaker menunda penambalan keamanan dan mengganti atau menghentikan instans titik akhir yang salah dalam sehari, untuk memungkinkan Anda melakukan debug.

Bagian berikut merinci bagaimana Anda dapat menggunakan fitur ini.

Daftar Izinkan

Anda harus menghubungi dukungan pelanggan, dan daftar akun Anda diizinkan, untuk menggunakan fitur ini. Anda tidak dapat membuat titik akhir dengan akses SSM diaktifkan, jika akun Anda tidak diizinkan terdaftar untuk akses ini.

Aktifkan akses SSM

Untuk mengaktifkan akses SSM untuk kontainer yang ada pada titik akhir, perbarui titik akhir dengan konfigurasi titik akhir baru, dengan `EnableSSMAccess` parameter disetel ke `true` Contoh berikut menyediakan konfigurasi titik akhir sampel.

```
{
    "EndpointConfigName": "endpoint-config-name",
    "ProductionVariants": [
        {
            "InitialInstanceCount": 1,
            "InitialVariantWeight": 1.0,
            "InstanceType": "ml.t2.medium",
            "ModelName": model-name,
            "VariantName": variant-name,
            "EnableSSMAccess": true,
        },
    ]
}
```

Untuk informasi selengkapnya tentang mengaktifkan akses SSM, lihat EnablesSmAccess.

Konfigurasi IAM

Izin IAM titik akhir

Jika Anda telah mengaktifkan akses SSM untuk instance endpoint, SageMaker mulai dan mengelola agen SSM saat memulai instance endpoint. Untuk memungkinkan agen SSM berkomunikasi dengan layanan SSM, tambahkan kebijakan berikut ke peran eksekusi yang dijalankan endpoint.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        }
    ]
 }
```

Izin IAM pengguna

Tambahkan kebijakan berikut untuk memberikan izin sesi SSM pengguna IAM untuk terhubung ke target SSM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:TerminateSession"
            ],
            "Resource": "*"
        }
    ]
}
```

Anda dapat membatasi titik akhir yang dapat dihubungkan oleh pengguna IAM, dengan kebijakan berikut. Ganti *`teks placeholder yang dicetak miring`* dengan informasi Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:StartSession",
            ],
            "Resource": [
                "sagemaker-endpoint-arn"
            ]
        }
    ]
}
```

Akses SSM dengan AWS PrivateLink

Jika endpoint Anda berjalan dalam virtual private cloud (VPC) yang tidak terhubung ke internet publik, Anda dapat AWS PrivateLink menggunakannya untuk mengaktifkan SSM. AWS PrivateLinkmembatasi semua lalu lintas jaringan antara instans titik akhir Anda, SSM, dan Amazon EC2 ke jaringan Amazon. Untuk informasi selengkapnya tentang cara mengatur akses SSMAWS PrivateLink, lihat Menyiapkan titik akhir VPC untuk Pengelola Sesi.

Logging dengan Amazon CloudWatch Logs

Untuk titik akhir yang diaktifkan akses SSM, Anda dapat mencatat kesalahan dari agen SSM dengan Amazon Logs. CloudWatch Untuk informasi selengkapnya tentang cara mencatat kesalahan dengan CloudWatch Log, lihat Aktivitas sesi logging. Log tersedia di aliran log SSM,*variant-name*`/`*ec2-instance-id*`/ssm`, di bawah grup log titik akhir. `/aws/sagemaker/endpoints/`*endpoint-name* Untuk informasi selengkapnya tentang cara melihat log, lihat Melihat data log yang dikirim ke CloudWatch Log.

Varian produksi di belakang titik akhir Anda dapat memiliki beberapa wadah model. Log untuk setiap wadah model dicatat dalam aliran log. Setiap log didahului oleh`[sagemaker ssm logs] [container-name]`, di `container-name` mana nama yang Anda berikan ke wadah, atau nama default, seperti`container_0`, dan. `container_1`

Mengakses wadah model

Untuk mengakses wadah model pada instance endpoint Anda, Anda memerlukan ID targetnya. ID target ada dalam salah satu format berikut:

- `sagemaker-endpoint:`*endpoint-name*`_`*variant-name*`_`*ec2-instance-id*untuk wadah pada titik akhir kontainer tunggal

- `sagemaker-endpoint:`*`endpoint-name_variant-name_ec2-instance-id_container-name`*untuk wadah pada titik akhir multi-kontainer

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan AWS CLI untuk mengakses wadah model menggunakan ID target.

```
aws ssm start-session --target sagemaker-endpoint:prod-image-
classifier_variant1_i-003a121c1b21a90a9_container_1
```

Jika Anda mengaktifkan logging, seperti yang disebutkan dalamLogging dengan Amazon CloudWatch Logs, Anda dapat menemukan ID target untuk semua kontainer yang terdaftar di awal aliran log SSM.

> **Note**
>
> - Anda tidak dapat terhubung ke wadah algoritma 1P atau wadah model yang diperoleh dari SageMaker MarketPlace SSM. Namun Anda dapat terhubung ke wadah pembelajaran mendalam (DLC) yang disediakan oleh AWS atau wadah khusus apa pun yang Anda miliki.
> - Jika Anda telah mengaktifkan isolasi jaringan untuk wadah model yang mencegahnya melakukan panggilan jaringan keluar, Anda tidak dapat memulai sesi SSM untuk wadah tersebut.
> - Anda hanya dapat mengakses satu kontainer dari satu sesi SSM. Untuk mengakses wadah lain, meskipun berada di belakang titik akhir yang sama, mulailah sesi SSM baru dengan ID target titik akhir tersebut.

Menyebarkan model dengan server model

Konten berikut menunjukkan cara menerapkan model Anda SageMaker menggunakan server model populer, seperti TorchServe dan Triton.

Menyebarkan model dengan TorchServe

TorchServe adalah server model yang direkomendasikan untuk PyTorch, sudah diinstal sebelumnya di AWS PyTorch Deep Learning Container (DLC). Alat canggih ini menawarkan pelanggan pengalaman yang konsisten dan ramah pengguna, memberikan kinerja tinggi dalam menerapkan

beberapa PyTorch model di berbagai AWS instance, termasuk CPU, GPU, Neuron, dan Graviton, terlepas dari ukuran model atau distribusinya.

TorchServe mendukung beragam fitur canggih, termasuk batching dinamis, microbatching, pengujian model A/B, streaming, obor XLA, TensorRT, ONNX dan IPEX. Selain itu, ia dengan mulus mengintegrasikan PyTorch solusi model besar, PiPPy, memungkinkan penanganan model besar yang efisien. Selain itu, TorchServe memperluas dukungannya ke perpustakaan open-source populer seperti DeepSpeed, Accelerate, Fast Transformers, dan banyak lagi, memperluas kemampuannya lebih jauh. Dengan TorchServe, AWS pengguna dapat dengan percaya diri menerapkan dan melayani PyTorch model mereka, memanfaatkan keserbagunaan dan kinerja yang dioptimalkan di berbagai konfigurasi perangkat keras dan jenis model. Untuk informasi lebih rinci, Anda dapat merujuk ke PyTorchdokumentasi dan TorchServeseterusnya GitHub.

Tabel berikut mencantumkan AWS PyTorch DLC yang didukung oleh. TorchServe

Jenis instans	SageMaker PyTorch Tautan DLC
CPU dan GPU	SageMaker PyTorch kontainer
Neuron	PyTorch Wadah neuron
Graviton	SageMaker PyTorch Wadah graviton

Bagian berikut menjelaskan pengaturan untuk membangun dan menguji PyTorch DLC di Amazon. SageMaker

Memulai

Untuk memulai, pastikan Anda memiliki prasyarat berikut:

1. Pastikan Anda memiliki akses ke AWS akun. Siapkan lingkungan Anda sehingga AWS CLI dapat mengakses akun Anda melalui pengguna AWS IAM atau peran IAM. Kami merekomendasikan menggunakan peran IAM. Untuk tujuan pengujian di akun pribadi Anda, Anda dapat melampirkan kebijakan izin terkelola berikut ke peran IAM:
   - AmazonEC2 ContainerRegistryFullAccess
   - AmazonEC2 FullAccess
   - AWSServiceRoleForAmazonEksNodeGroup
   - AmazonSageMakerFullAccess

- AmazonS3 FullAccess

2. Konfigurasikan dependensi Anda secara lokal, seperti yang ditunjukkan pada contoh berikut:

```
from datetime import datetime
    import os
    import json
    import logging
    import time

    # External Dependencies:
    import boto3
    from botocore.exceptions import ClientError
    import sagemaker

    sess = boto3.Session()
    sm = sess.client("sagemaker")
    region = sess.region_name
    account = boto3.client("sts").get_caller_identity().get("Account")

    smsess = sagemaker.Session(boto_session=sess)
    role = sagemaker.get_execution_role()

    # Configuration:
    bucket_name = smsess.default_bucket()
    prefix = "torchserve"
    output_path = f"s3://{bucket_name}/{prefix}/models"
    print(f"account={account}, region={region}, role={role}")
```

3. Ambil gambar PyTorch DLC, seperti yang ditunjukkan pada contoh berikut.

   SageMaker PyTorch Gambar DLC tersedia di semua AWS wilayah. Untuk informasi selengkapnya, lihat daftar gambar kontainer DLC.

```
baseimage = sagemaker.image_uris.retrieve(
        framework="pytorch",
        region="<region>",
        py_version="py310",
        image_scope="inference",
        version="2.0.1",
        instance_type="ml.g4dn.16xlarge",
    )
```

4. Buat ruang kerja lokal.

```
mkdir -p workspace/
```

Menambahkan paket

Bagian berikut menjelaskan cara menambahkan dan menginstal paket ke gambar PyTorch DLC Anda.

Kasus penggunaan BYOC

Langkah-langkah berikut menguraikan cara menambahkan paket ke gambar PyTorch DLC Anda. Untuk informasi selengkapnya tentang menyesuaikan container, lihat Membangun Gambar Kustom AWS Deep Learning Containers.

1. Misalkan Anda ingin menambahkan paket ke image PyTorch docker DLC. Buat Dockerfile di bawah `docker` direktori, seperti yang ditunjukkan pada contoh berikut:

```
mkdir -p workspace/docker
    cat workspace/docker/Dockerfile

    ARG BASE_IMAGE

    FROM $BASE_IMAGE

    #Install any additional libraries
    RUN pip install transformers==4.28.1
```

2. Buat dan publikasikan image docker yang disesuaikan dengan menggunakan skrip build_and_push.sh berikut.

```
# Download script build_and_push.sh to workspace/docker
    ls workspace/docker
    build_and_push.sh  Dockerfile

    # Build and publish your docker image
    reponame = "torchserve"
    versiontag = "demo-0.1"

    ./build_and_push.sh {reponame} {versiontag} {baseimage} {region} {account}
```

SageMaker kasus penggunaan prainstal

Contoh berikut menunjukkan cara menginstal paket ke wadah PyTorch DLC Anda. Anda harus membuat `requirements.txt` file secara lokal di bawah direktori`workspace/code`.

```
mkdir -p workspace/code
    cat workspace/code/requirements.txt

    transformers==4.28.1
```

Buat artefak TorchServe model

Dalam contoh berikut, kami menggunakan model MNIST yang telah dilatih sebelumnya. Kami membuat direktori`workspace/mnist`, menerapkan mnist_handler.py dengan mengikuti instruksi layanan TorchServe khusus, dan mengkonfigurasi parameter model (seperti ukuran batch dan pekerja) di model-config.yaml. Kemudian, kami menggunakan TorchServe alat `torch-model-archiver` untuk membangun artefak model dan mengunggah ke Amazon S3.

1. Konfigurasikan parameter model di`model-config.yaml`.

   ```
   ls -al workspace/mnist-dev

       mnist.py
       mnist_handler.py
       mnist_cnn.pt
       model-config.yaml

       # config the model
       cat workspace/mnist-dev/model-config.yaml
       minWorkers: 1
       maxWorkers: 1
       batchSize: 4
       maxBatchDelay: 200
       responseTimeout: 300
   ```

2. Bangun artefak model dengan menggunakan torch-model-archiver .

   ```
   torch-model-archiver --model-name mnist --version 1.0 --model-file workspace/
   mnist-dev/mnist.py --serialized-file workspace/mnist-dev/mnist_cnn.pt --handler
   ```

```
 workspace/mnist-dev/mnist_handler.py --config-file workspace/mnist-dev/model-
config.yaml --archive-format tgz
```

Jika Anda ingin menginstal paket, Anda harus menyertakan `code` direktori dalam `tar.gz` file.

```
cd workspace
    torch-model-archiver --model-name mnist --version 1.0 --model-file mnist-
dev/mnist.py --serialized-file mnist-dev/mnist_cnn.pt --handler mnist-dev/
mnist_handler.py --config-file mnist-dev/model-config.yaml --archive-format no-
archive

    cd mnist
    mv ../code .
    tar cvzf mnist.tar.gz .
```

3. Unggah `mnist.tar.gz` ke Amazon S3.

```
# upload mnist.tar.gz to S3
    output_path = f"s3://{bucket_name}/{prefix}/models"
    aws s3 cp mnist.tar.gz {output_path}/mnist.tar.gz
```

Menggunakan titik akhir model tunggal untuk diterapkan dengan TorchServe

Contoh berikut menunjukkan cara membuat titik akhir inferensi real-time model tunggal, menerapkan model ke titik akhir, dan menguji titik akhir dengan menggunakan Amazon Python SDK. SageMaker

```
from sagemaker.model import Model
    from sagemaker.predictor import Predictor

    # create the single model endpoint and deploy it on SageMaker
    model = Model(model_data = f'{output_path}/mnist.tar.gz',
                  image_uri = baseimage,
                  role = role,
                  predictor_cls = Predictor,
                  name = "mnist",
                  sagemaker_session = smsess)

    endpoint_name = 'torchserve-endpoint-' + time.strftime("%Y-%m-%d-%H-%M-%S",
 time.gmtime())
    predictor = model.deploy(instance_type='ml.g4dn.xlarge',
                             initial_instance_count=1,
```

```
                            endpoint_name = endpoint_name,
                            serializer=JSONSerializer(),
                            deserializer=JSONDeserializer())

    # test the endpoint
    import random
    import numpy as np
    dummy_data = {"inputs": np.random.rand(16, 1, 28, 28).tolist()}

    res = predictor.predict(dummy_data)
```

Menggunakan titik akhir multi-model untuk diterapkan TorchServe

Titik akhir multi-model adalah solusi yang dapat diskalakan dan hemat biaya untuk menampung sejumlah besar model di belakang satu titik akhir. Mereka meningkatkan pemanfaatan titik akhir dengan berbagi armada sumber daya yang sama dan melayani wadah untuk menampung semua model Anda. Mereka juga mengurangi overhead penerapan karena SageMaker mengelola model bongkar muat secara dinamis, serta penskalaan sumber daya berdasarkan pola lalu lintas. Titik akhir multi-model sangat berguna untuk pembelajaran mendalam dan model AI generatif yang membutuhkan daya komputasi yang dipercepat.

Dengan menggunakan TorchServe pada titik akhir SageMaker multi-model, Anda dapat mempercepat pengembangan Anda dengan menggunakan tumpukan penyajian yang Anda kenal sambil memanfaatkan berbagi sumber daya dan manajemen model yang disederhanakan yang disediakan oleh titik akhir SageMaker multi-model.

Contoh berikut menunjukkan cara membuat endpoint multi-model, menerapkan model ke endpoint, dan menguji endpoint dengan menggunakan Amazon Python SDK. SageMaker Rincian tambahan dapat ditemukan dalam contoh notebook ini.

```
from sagemaker.multidatamodel import MultiDataModel
    from sagemaker.model import Model
    from sagemaker.predictor import Predictor

    # create the single model endpoint and deploy it on SageMaker
    model = Model(model_data = f'{output_path}/mnist.tar.gz',
                  image_uri = baseimage,
                  role = role,
                  sagemaker_session = smsess)
```

```
    endpoint_name = 'torchserve-endpoint-' + time.strftime("%Y-%m-%d-%H-%M-%S",
time.gmtime())
    mme = MultiDataModel(
        name = endpoint_name,
        model_data_prefix = output_path,
        model = model,
        sagemaker_session = smsess)

    mme.deploy(
        initial_instance_count = 1,
        instance_type = "ml.g4dn.xlarge",
        serializer=sagemaker.serializers.JSONSerializer(),
        deserializer=sagemaker.deserializers.JSONDeserializer())

    # list models
    list(mme.list_models())

    # create mnist v2 model artifacts
    cp mnist.tar.gz mnistv2.tar.gz

    # add mnistv2
    mme.add_model(mnistv2.tar.gz)

    # list models
    list(mme.list_models())

    predictor = Predictor(endpoint_name=mme.endpoint_name, sagemaker_session=smsess)

    # test the endpoint
    import random
    import numpy as np
    dummy_data = {"inputs": np.random.rand(16, 1, 28, 28).tolist()}

    res = predictor.predict(date=dummy_data, target_model="mnist.tar.gz")
```

Metrik

TorchServe mendukung metrik tingkat sistem dan tingkat model. Anda dapat mengaktifkan metrik dalam mode format log atau mode Prometheus melalui variabel lingkungan. `TS_METRICS_MODE` Anda dapat menggunakan file konfigurasi metrik TorchServe pusat `metrics.yaml` untuk menentukan jenis metrik yang akan dilacak, seperti jumlah permintaan, latensi, penggunaan memori, pemanfaatan GPU, dan banyak lagi. Dengan mengacu pada file ini, Anda dapat memperoleh

wawasan tentang kinerja dan kesehatan model yang diterapkan dan secara efektif memantau perilaku TorchServe server secara real-time. Untuk informasi lebih rinci, lihat dokumentasi TorchServe metrik.

Anda dapat mengakses log TorchServe metrik yang mirip dengan format StatSD melalui filter log CloudWatch Amazon. Berikut ini adalah contoh log TorchServe metrik:

```
CPUUtilization.Percent:0.0|#Level:Host|#hostname:my_machine_name,timestamp:1682098185
    DiskAvailable.Gigabytes:318.0416717529297|#Level:Host|
#hostname:my_machine_name,timestamp:1682098185
```

Menyebarkan model dengan DJL Serving

DJL Serving adalah solusi penyajian model berdiri sendiri universal berkinerja tinggi. Dibutuhkan model pembelajaran mendalam, beberapa model, atau alur kerja dan membuatnya tersedia melalui titik akhir HTTP.

Anda dapat menggunakan salah satu DJL Serving Deep Learning Containers (DLC) untuk melayani model Anda. AWS Untuk mempelajari tentang jenis model dan kerangka kerja yang didukung, lihat repositori DJL Serving GitHub.

DJL Serving menawarkan banyak fitur yang membantu Anda untuk menyebarkan model Anda dengan kinerja tinggi:

- Kemudahan penggunaan - DJL Serving dapat melayani sebagian besar model tanpa modifikasi apa pun. Anda membawa artefak model Anda, dan DJL Serving dapat meng-host mereka.
- Beberapa perangkat dan dukungan akselerator — DJL Serving mendukung penerapan model pada CPU, GPU, dan Inferentia. AWS
- Kinerja - DJL Serving menjalankan inferensi multithreaded dalam satu mesin virtual Java (JVM) untuk meningkatkan throughput.
- Batching dinamis - DJL Serving mendukung batching dinamis untuk meningkatkan throughput.
- Penskalaan otomatis - DJL Serving secara otomatis menskalakan pekerja naik atau turun berdasarkan beban lalu lintas.
- Dukungan multi-engine - DJL Serving dapat secara bersamaan meng-host model menggunakan kerangka kerja yang berbeda (misalnya, PyTorch dan). TensorFlow
- Model ensemble dan alur kerja — DJL Serving mendukung penerapan alur kerja kompleks yang terdiri dari beberapa model dan dapat mengeksekusi bagian alur kerja pada CPU dan bagian lain pada GPU. Model dalam alur kerja dapat memanfaatkan kerangka kerja yang berbeda.

Bagian berikut menjelaskan cara mengatur endpoint dengan DJL Serving pada. SageMaker

Memulai

Untuk memulai, pastikan Anda memiliki prasyarat berikut:

1. Pastikan Anda memiliki akses ke AWS akun. Siapkan lingkungan Anda sehingga AWS CLI dapat mengakses akun Anda melalui pengguna AWS IAM atau peran IAM. Kami merekomendasikan menggunakan peran IAM. Untuk tujuan pengujian di akun pribadi Anda, Anda dapat melampirkan kebijakan izin terkelola berikut ke peran IAM:
   - AmazonEC2 ContainerRegistryFullAccess
   - AmazonEC2 FullAccess
   - AmazonSageMakerFullAccess
   - AmazonS3 FullAccess
2. Pastikan Anda telah menyiapkan klien docker di sistem Anda.
3. Masuk ke Amazon Elastic Container Registry dan atur variabel lingkungan berikut:

   ```
   export ACCOUNT_ID=<your_account_id>
   export REGION=<your_region>
   aws ecr get-login-password --region $REGION | docker login --username AWS --password-
   stdin $ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com
   ```

4. Tarik gambar docker.

   ```
   docker pull 763104351884.dkr.ecr.us-west-2.amazonaws.com/djl-inference:0.22.1-
   deepspeed0.9.2-cu118
   ```

   Untuk semua gambar kontainer DJL Serving yang tersedia, lihat wadah inferensi model besar dan wadah inferensi CPU Serving DJL. Saat memilih gambar dari tabel di tautan sebelumnya, ganti AWS wilayah di kolom URL contoh dengan wilayah tempat Anda berada. DLC tersedia di wilayah yang tercantum dalam tabel di bagian atas halaman Available Deep Learning Containers Images.

Sesuaikan wadah Anda

Anda dapat menambahkan paket ke gambar DLC dasar untuk menyesuaikan wadah Anda. Misalkan Anda ingin menambahkan paket ke image `763104351884.dkr.ecr.us-west-2.amazonaws.com/djl-inference:0.22.1-deepspeed0.9.2-cu118` docker.

Anda harus membuat dockerfile dengan gambar yang Anda inginkan sebagai gambar dasar, menambahkan paket yang diperlukan, dan mendorong gambar ke Amazon ECR.

Untuk menambahkan paket, selesaikan langkah-langkah berikut:

1. Tentukan instruksi untuk menjalankan pustaka atau paket yang Anda inginkan di dockerfile gambar dasar.

```
FROM 763104351884.dkr.ecr.us-west-2.amazonaws.com/djl-inference:0.22.1-
deepspeed0.9.2-cu118

## add custom packages/libraries
RUN git clone https://github.com/awslabs/amazon-sagemaker-examples
```

2. Bangun image Docker dari dockerfile. Tentukan repositori Amazon ECR Anda, nama gambar dasar, dan tag untuk gambar. Jika Anda tidak memiliki repositori Amazon ECR, lihat Menggunakan Amazon ECR dengan di AWS CLI Panduan Pengguna Amazon ECR untuk petunjuk tentang cara membuatnya.

```
docker build -f Dockerfile -t <registry>/<image_name>:<image_tag>
```

3. Dorong gambar Docker ke repositori Amazon ECR Anda.

```
docker push $ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com/<image_name>:<image_tag>
```

Anda sekarang harus memiliki gambar kontainer khusus yang dapat Anda gunakan untuk penyajian model. Untuk contoh penyesuaian kontainer lainnya, lihat Membangun Gambar Kustom AWS Deep Learning Containers.

Siapkan artefak model Anda

Sebelum menerapkan model Anda SageMaker, Anda harus mengemas artefak model Anda dalam sebuah `.tar.gz` file. DJL Serving menerima artefak berikut dalam arsip Anda:

- Pos pemeriksaan model: File yang menyimpan bobot model Anda.
- `serving.properties`: File konfigurasi yang dapat Anda tambahkan untuk setiap model. Tempatkan `serving.properties` di direktori yang sama dengan file model Anda.
- `model.py`: Kode penangan inferensi. Ini hanya berlaku saat menggunakan mode Python. Jika Anda tidak menentukan`model.py`, djl-serving menggunakan salah satu penangan default.

Berikut ini adalah contoh `model.tar.gz` struktur:

```
- model_root_dir # root directory
   - serving.properties
   - model.py # your custom handler file for Python, if you choose not to use the
default handlers provided by DJL Serving
   - model binary files # used for Java mode, or if you don't want to use
option.model_id and option.s3_url for Python mode
```

DJL Serving mendukung mesin Java didukung oleh mesin DJL atau Python. Tidak semua artefak sebelumnya diperlukan; artefak yang diperlukan bervariasi berdasarkan mode yang Anda pilih. Misalnya, dalam mode Python, Anda hanya perlu menentukan `option.model_id` dalam `serving.properties` file; Anda tidak perlu menentukan pos pemeriksaan model di dalam wadah LMI. Dalam mode Java, Anda diminta untuk mengemas pos pemeriksaan model. Untuk detail selengkapnya tentang cara mengkonfigurasi `serving.properties` dan mengoperasikan dengan mesin yang berbeda, lihat Mode Operasi Penyajian DJL.

Gunakan titik akhir model tunggal untuk menerapkan dengan DJL Serving

Setelah menyiapkan artefak model Anda, Anda dapat menerapkan model Anda ke titik akhir. SageMaker Bagian ini menjelaskan cara menerapkan model tunggal ke titik akhir dengan DJL Serving. Jika Anda menerapkan beberapa model, lewati bagian ini dan buka. Gunakan titik akhir multi-model untuk menerapkan dengan DJL Serving

Contoh berikut menunjukkan metode untuk membuat objek model menggunakan Amazon SageMaker Python SDK. Anda harus menentukan bidang berikut:

- `image_uri`: Anda dapat mengambil salah satu gambar DJL Serving dasar seperti yang ditunjukkan dalam contoh ini, atau Anda dapat menentukan gambar Docker kustom dari repositori Amazon ECR Anda, jika Anda mengikuti instruksi di. Sesuaikan wadah Anda
- `model_s3_url`: Ini harus URI Amazon S3 yang menunjuk ke file Anda`.tar.gz`.
- `model_name`: Tentukan nama untuk objek model.

```
import boto3
 import sagemaker
from sagemaker.model import Model
from sagemaker import image_uris, get_execution_role
```

```
aws_region = "aws-region"
sagemaker_session =
 sagemaker.Session(boto_session=boto3.Session(region_name=aws_region))
role = get_execution_role()

def create_model(model_name, model_s3_url):
    # Get the DJL DeepSpeed image uri
    image_uri = image_uris.retrieve(
        framework="djl-deepspeed",
        region=sagemaker_session.boto_session.region_name,
        version="0.20.0"
    )
    model = Model(
        image_uri=image_uri,
        model_data=model_s3_url,
        role=role,
        name=model_name,
        sagemaker_session=sagemaker_session,
    )
    return model
```

Setelah Anda membuat objek model, Anda dapat menerapkan model ke titik akhir dan membuat prediksi. Untuk end-to-end contoh cara melakukan langkah-langkah ini, lihat tutorialnyaInferensi model besar dengan DeepSpeed dan DJL Serving.

Gunakan titik akhir multi-model untuk menerapkan dengan DJL Serving

Jika Anda ingin menerapkan beberapa model ke titik akhir, SageMaker menawarkan titik akhir multi-model, yang merupakan solusi terukur dan hemat biaya untuk menerapkan sejumlah besar model. DJL Serving juga mendukung pemuatan beberapa model secara bersamaan dan menjalankan inferensi pada masing-masing model secara bersamaan. Kontainer Penyajian DJL mematuhi kontrak titik akhir SageMaker multi-model dan dapat digunakan untuk menyebarkan titik akhir multi-model.

Setiap artefak model individu perlu dikemas dengan cara yang sama seperti yang dijelaskan di bagian sebelumnya. Siapkan artefak model Anda Anda dapat mengatur konfigurasi khusus model dalam `serving.properties` file dan kode pengendali inferensi khusus model di. `model.py` Untuk titik akhir multi-model, model perlu diatur dengan cara berikut:

```
 root_dir
       |-- model_1.tar.gz
       |-- model_2.tar.gz
```

```
        |-- model_3.tar.gz
            .
            .
            .
```

Amazon SageMaker Python SDK menggunakan MultiDataModelobjek untuk membuat instance endpoint multi-model. URI Amazon S3 untuk direktori root harus diteruskan sebagai `model_data_prefix` argumen ke konstruktor. `MultiDataModel`

DJL Serving juga menyediakan beberapa parameter konfigurasi untuk mengelola persyaratan memori model, seperti `required_memory_mb` dan`reserved_memory_mb`, yang dapat dikonfigurasi untuk setiap model dalam file serving.properties. Parameter ini berguna untuk menangani kesalahan memori dengan lebih anggun. Untuk semua parameter yang dapat dikonfigurasi, lihat OutofMemory penanganan di djl-serving.

Fitur penskalaan otomatis dari DJL Serving memudahkan untuk memastikan bahwa model diskalakan dengan tepat untuk lalu lintas masuk. Secara default, DJL Serving menentukan jumlah maksimum pekerja untuk model yang dapat didukung berdasarkan perangkat keras yang tersedia (seperti core CPU atau perangkat GPU). Anda dapat mengatur batas bawah dan atas untuk setiap model untuk memastikan bahwa tingkat lalu lintas minimum selalu dapat dilayani, dan bahwa satu model tidak mengkonsumsi semua sumber daya yang tersedia. Anda dapat mengatur properti berikut dalam file serving.properties:

- `gpu.minWorkers`: Jumlah minimum pekerja untuk GPU.
- `gpu.maxWorkers`: Jumlah maksimum pekerja untuk GPU.
- `cpu.minWorkers`: Jumlah minimum pekerja untuk CPU.
- `cpu.maxWorkers`: Jumlah maksimum pekerja untuk CPU.

Untuk end-to-end contoh cara menerapkan titik akhir multi-model saat SageMaker menggunakan kontainer Penyajian DJL, lihat contoh notebook Multi-model-inference-demo.ipynb.

Menerapkan model dengan Triton Inference Server

Triton Inference Server adalah perangkat lunak penyajian inferensi open source yang merampingkan inferensi AI. Dengan Triton, Anda dapat menerapkan model apa pun yang dibangun dengan beberapa kerangka pembelajaran mendalam dan pembelajaran mesin, termasuk TensorRT,,, ONNX, OpenVINO, Python, RAPIDS FIL TensorFlow PyTorch, dan banyak lagi.

Kontainer SageMaker Triton membantu Anda menyebarkan Triton Inference Server pada platform SageMaker Hosting untuk melayani model terlatih dalam produksi. Ini mendukung berbagai mode di mana SageMaker beroperasi. Untuk daftar container Triton Inference Server yang tersedia SageMaker, lihat NVIDIA Triton Inference Containers (hanya dukungan SM).

Untuk contoh end-to-end notebook, kami sarankan untuk melihat amazon-sagemaker-examples repositori.

Mode hosting

Mode SageMaker Hosting berikut didukung oleh kontainer Triton:

- Titik akhir model tunggal
  - Ini SageMaker adalah mode operasi default. Dalam mode ini, wadah Triton dapat memuat satu model, atau model ansambel tunggal.
  - Nama model harus diteruskan sebagai properti lingkungan kontainer, yang merupakan bagian dari panggilan `CreateModel` SageMaker API. Variabel lingkungan yang digunakan untuk meneruskan nama model adalah`SAGEMAKER_TRITON_DEFAULT_MODEL_NAME`.
- Titik akhir model tunggal dengan ansambel
  - Triton Inference Server mendukung ansambel, yang merupakan pipeline, atau DAG (grafik asiklik terarah) model. Sementara ansambel secara teknis terdiri dari beberapa model, dalam mode titik akhir model tunggal default, SageMaker dapat memperlakukan ansambel dengan tepat (meta-model yang mewakili pipa) sebagai model utama untuk dimuat, dan selanjutnya dapat memuat model terkait.
  - Nama model ansambel yang tepat harus digunakan untuk memuat model. Itu harus diteruskan sebagai properti lingkungan kontainer, yang merupakan bagian dari panggilan `CreateModel` SageMaker API. Variabel lingkungan yang digunakan untuk meneruskan nama model adalah`SAGEMAKER_TRITON_DEFAULT_MODEL_NAME`.
- Titik akhir multi-model
  - Dalam mode ini, SageMaker dapat melayani beberapa model pada satu titik akhir. Anda dapat menggunakan mode ini dengan menentukan variabel lingkungan `'MultiModel': true` sebagai properti lingkungan kontainer, yang merupakan bagian dari panggilan `CreateModel` SageMaker API.
  - Secara default, tidak ada model yang dimuat saat instance dimulai. Untuk menjalankan permintaan inferensi terhadap model tertentu, tentukan `*.tar.gz` file model yang sesuai sebagai argumen ke `TargetModel` properti panggilan `InvokeEndpoint` SageMaker API.

- Titik akhir multi-model dengan ansambel
  - Dalam mode ini, SageMaker berfungsi seperti yang dijelaskan untuk titik akhir multi-model. Namun, kontainer SageMaker Triton dapat memuat beberapa model ansambel, yang berarti bahwa beberapa pipeline model dapat berjalan pada instance yang sama. SageMaker memperlakukan setiap ansambel sebagai satu model, dan ansambel yang tepat dari setiap model dapat dipanggil dengan menentukan arsip yang sesuai sebagai. `*.tar.gz` `TargetModel`
  - Untuk manajemen memori yang lebih baik selama memori dinamis `LOAD` dan`UNLOAD`, kami sarankan Anda menjaga ukuran ansambel kecil.

Jenis muatan inferensi

Triton mendukung dua metode pengiriman muatan inferensi melalui jaringan - `json` dan `binary +json` (atau json yang dikodekan biner). Muatan JSON dalam kedua kasus mencakup tipe data, bentuk, dan tensor permintaan inferensi yang sebenarnya. Tensor permintaan harus berupa tensor biner.

Dengan `binary+json` formatnya, Anda harus menentukan panjang metadata permintaan di header untuk memungkinkan Triton mengurai muatan biner dengan benar. Dalam wadah SageMaker Triton, ini dilakukan dengan menggunakan `Content-Type` header khusus:. `application/ vnd.sagemaker-triton.binary+json;json-header-size={}` Ini berbeda dengan menggunakan `Inference-Header-Content-Length` header pada Triton Inference Server yang berdiri sendiri karena header khusus tidak diizinkan masuk. SageMaker

Menggunakan config.pbtxt untuk mengatur konfigurasi model

Untuk Server Inferensi Triton SageMaker aktif, setiap model harus menyertakan `config.pbtxt` file yang menentukan, setidaknya, konfigurasi berikut untuk model:

- `name`: Meskipun ini opsional untuk model yang berjalan di luar SageMaker, kami menyarankan Anda selalu memberikan nama untuk model yang akan dijalankan di Triton on. SageMaker
- `platform`dan/atau `backend`: Menyetel backend sangat penting untuk menentukan jenis model. Beberapa backend memiliki klasifikasi lebih lanjut, seperti `tensorflow_savedmodel` atau. `tensorflow_graphdef` Opsi tersebut dapat ditentukan sebagai bagian dari `platform` kunci selain `backend` kunci. Backend yang paling umum adalah`tensorrt`,,`onnxruntime`,`tensorflow`,,`pytorch`, `python` `dalifil`, dan. `openvino`
- `input`: Tentukan tiga atribut untuk input:`name`, `data_type` dan `dims` (bentuk).

- `output`: Tentukan tiga atribut untuk output:`name`, `data_type` dan `dims` (bentuk).
- `max_batch_size`: Atur ukuran batch ke nilai yang lebih besar dari atau sama dengan 1 yang menunjukkan ukuran batch maksimum yang harus digunakan Triton dengan model.

Untuk detail lebih lanjut tentang konfigurasi`config.pbtxt`, lihat repositori Triton. GitHub Triton menyediakan beberapa konfigurasi untuk mengubah perilaku model. Beberapa opsi konfigurasi yang paling umum dan penting adalah:

- `instance_groups`: Grup instance membantu menentukan nomor dan lokasi untuk model tertentu. Mereka memiliki atribut`count`,`kind`, dan `gpus` (`kind`digunakan kapan`KIND_GPU`). `count`Atributnya setara dengan jumlah pekerja. Untuk penyajian model reguler, setiap pekerja memiliki salinan modelnya sendiri. Demikian pula, di Triton, `count` menentukan jumlah salinan model per perangkat. Misalnya, jika `instance_group` jenisnya`KIND_CPU`, maka CPU memiliki `count` jumlah salinan model.

  > **Note**
  >
  > Pada instance GPU, `instance_group` konfigurasi berlaku per perangkat GPU. Misalnya, `count` jumlah salinan model ditempatkan pada setiap perangkat GPU kecuali Anda secara eksplisit menentukan perangkat GPU mana yang harus memuat model.

- `dynamic_batching`dan `sequence_batching`: Batching dinamis digunakan untuk model stateless, dan batch urutan digunakan untuk model stateful (di mana Anda ingin merutekan permintaan ke instance model yang sama setiap saat). Penjadwal batching mengaktifkan antrian per model, yang membantu meningkatkan throughput, tergantung pada konfigurasi batching.
- `ensemble`: Model ansambel mewakili pipa dari satu atau lebih model dan koneksi tensor input dan output antara model-model tersebut. Itu dapat dikonfigurasi dengan menentukan `platform` sebagai`ensemble`. Konfigurasi ansambel hanyalah representasi dari pipa model. Pada SageMaker, semua model di bawah ansambel diperlakukan sebagai tanggungan model ansambel dan dihitung sebagai model tunggal untuk metrik, seperti. SageMaker `LoadedModelCount`

Menerbitkan metrik Triton default ke Amazon CloudWatch

NVIDIA Triton Inference Container memaparkan metrik pada port 8002 (dapat dikonfigurasi) untuk berbagai model dan GPU yang digunakan di Triton Inference Server. Untuk detail lengkap metrik

default yang tersedia, lihat GitHub halaman untuk metrik Triton Inference Server. Metrik ini dalam format Prometheus dan dapat dikikis menggunakan konfigurasi scraper Prometheus.

Dimulai dengan versi v23.07 dan seterusnya, wadah SageMaker Triton mendukung penerbitan metrik ini ke Amazon CloudWatch dengan menentukan beberapa variabel lingkungan. Untuk mengikis metrik Prometheus, wadah Triton memanfaatkan agen Amazon. SageMaker CloudWatch

Variabel lingkungan yang diperlukan yang harus Anda tentukan untuk mengumpulkan metrik adalah sebagai berikut:

Variabel lingkungan	Deskripsi	Nilai contoh
`SAGEMAKER_TRITON_ALLOW_METRICS`	Tentukan opsi ini untuk memungkinkan Triton mempublikasikan metrik ke titik akhir Prometheus-nya.	“benar”
`SAGEMAKER_TRITON_PUBLISH_METRICS_TO_CLOUDWATCH`	Tentukan opsi ini untuk memulai pra-pemeriksaan yang diperlukan untuk mempublikasikan metrik ke Amazon. CloudWatch	“benar”
`SAGEMAKER_TRITON_CLOUDWATCH_LOG_GROUP`	Tentukan opsi ini untuk menunjuk ke grup log tempat metrik ditulis.	“/aws//titik akhir//SageMaker” TritonMetrics SageMaker TwoEnsemblesTest
`SAGEMAKER_TRITON_CLOUDWATCH_METRIC_NAMESPACE`	Tentukan opsi ini untuk menunjuk ke namespace metrik tempat Anda ingin melihat dan memplot metrik.	“/aws//titik akhir//SageMaker” TritonMetrics SageMaker TwoEnsemblesPublicTest
`SAGEMAKER_TRITON_METRICS_PORT`	Tentukan ini sebagai 8002, atau port lainnya. Jika SageMaker belum memblokir port yang ditentukan, itu digunakan. Jika tidak, port	“8002"

Variabel lingkungan	Deskripsi	Nilai contoh
	lain yang tidak diblokir dipilih secara otomatis.	

Saat menerbitkan metrik dengan Triton aktif SageMaker, ingatlah batasan berikut:

- Meskipun Anda dapat menghasilkan metrik khusus melalui backend C-API dan Python (v23.05 dan seterusnya), metrik ini saat ini tidak didukung untuk dipublikasikan ke Amazon. CloudWatch
- Dalam mode SageMaker multi-model endpoint (MME), Triton berjalan di lingkungan yang memerlukan ruang nama model untuk diaktifkan karena setiap model (kecuali model ansambel) diperlakukan seolah-olah mereka berada dalam repositori model mereka sendiri. Saat ini, ini menciptakan batasan untuk metrik. Saat ruang nama model diaktifkan, Triton tidak membedakan metrik antara dua model dengan nama yang sama milik ansambel yang berbeda. Sebagai solusinya, pastikan setiap model yang digunakan memiliki nama yang unik. Ini juga membuatnya lebih mudah untuk mencari metrik Anda. CloudWatch

Variabel-variabel lingkungan

Tabel berikut mencantumkan variabel lingkungan yang didukung untuk Triton pada. SageMaker

Variabel lingkungan	Deskripsi	Jenis	Kemungkinan nilai
`SAGEMAKER_MULTI_MODEL`	Memungkinkan Triton beroperasi dalam mode titik akhir SageMaker multi-model.	Boolean	`true`, `false`
`SAGEMAKER_TRITON_DEFAULT_MODEL_NAME`	Tentukan model yang akan dimuat dalam mode model SageMaker tunggal (default). Untuk mode ansambel, tentukan	String	<model_name>seperti yang ditentukan dalam config.pbtxt

Variabel lingkungan	Deskripsi	Jenis	Kemungkinan nilai
	nama ansambel yang tepat.		
SAGEMAKER_TRITON_PING_MODE	'ready'adalah mode default dalam SageMaker mode model tunggal, dan 'live' merupakan default dalam SageMaker mode titik akhir multi-model.	String	ready, live
SAGEMAKER_TRITON_DISABLE_MODEL_NAMESPACING	Dalam wadah SageMaker Triton, ini diatur ke secara true default.	Boolean	true, false
SAGEMAKER_BIND_TO_PORT	Saat aktif SageMaker, port defaultnya adalah 8080. Anda dapat menyesuaikan ke port yang berbeda dalam skenario multi-kontainer.	String	<port_number>
SAGEMAKER_SAFE_PORT_RANGE	Ini diatur oleh SageMaker platform saat menggunakan mode multi-kontainer.	String	<port_1>— <port_2>

Variabel lingkungan	Deskripsi	Jenis	Kemungkinan nilai
`SAGEMAKER_TRITON_ALLOW_GRPC`	Meskipun SageMaker tidak mendukung GRPC saat ini, jika Anda menggunakan Triton di depan proxy terbalik khusus, Anda dapat memilih untuk mengaktifkan GRPC.	Boolean	`true`, `false`
`SAGEMAKER_TRITON_GRPC_PORT`	Port default untuk GRPC adalah 8001, tetapi Anda dapat mengubahnya.	String	<port_number>
`SAGEMAKER_TRITON_THREAD_COUNT`	Anda dapat mengatur jumlah thread handler permintaan HTTP default.	String	<number>
`SAGEMAKER_TRITON_LOG_VERBOSE`	`true`secara default aktif SageMaker, tetapi Anda dapat menonaktifkan opsi ini secara selektif.	Boolean	`true`, `false`
`SAGEMAKER_TRITON_LOG_INFO`	`false`secara default aktif SageMaker.	Boolean	`true`, `false`
`SAGEMAKER_TRITON_LOG_WARNING`	`false`secara default aktif SageMaker.	Boolean	`true`, `false`
`SAGEMAKER_TRITON_LOG_ERROR`	`false`secara default aktif SageMaker.	Boolean	`true`, `false`

Variabel lingkungan	Deskripsi	Jenis	Kemungkinan nilai
SAGEMAKER_TRITON_SHM_DEFAULT_BYTE_SIZE	Tentukan ukuran shm untuk backend Python, dalam byte. Nilai defaultnya adalah 16 MB tetapi dapat ditingkatkan.	String	<number>
SAGEMAKER_TRITON_SHM_GROWTH_BYTE_SIZE	Tentukan ukuran pertumbuhan shm untuk backend Python, dalam byte. Nilai defaultnya adalah 1 MB tetapi dapat ditingkatkan untuk memungkinkan peningkatan yang lebih besar.	String	<number>
SAGEMAKER_TRITON_TENSORFLOW_VERSION	Nilai default-nya adalah 2. Triton tidak lagi mendukung Tensorflow 2 dari Triton v23.04. Anda dapat mengonfigurasi variabel ini untuk versi sebelumnya.	String	<number>

Variabel lingkungan	Deskripsi	Jenis	Kemungkinan nilai
SAGEMAKER_TRITON_MODEL_LOAD_GPU_LIMIT	Batasi persentase memori GPU maksimum yang digunakan untuk pemuatan model, memungkinkan sisanya digunakan untuk permintaan inferensi.	String	<number>
SAGEMAKER_TRITON_ALLOW_METRICS	false secara default aktif SageMaker.	Boolean	true, false
SAGEMAKER_TRITON_METRICS_PORT	Port default adalah 8002.	String	<number>
SAGEMAKER_TRITON_PUBLISH_METRICS_TO_CLOUDWATCH	false secara default aktif SageMaker. Setel variabel ini true untuk memungkinkan mendorong metrik default Triton ke Amazon. CloudWatch Jika opsi ini diaktifkan, Anda bertanggung jawab atas CloudWatch biaya saat metrik dipublikasikan ke akun Anda.	Boolean	true, false

Variabel lingkungan	Deskripsi	Jenis	Kemungkinan nilai
`SAGEMAKER_TRITON_CLOUDWATCH_LOG_GROUP`	Diperlukan jika Anda telah mengaktifkan penerbitan metrik. CloudWatch	String	<cloudwatch_log_group_name>
`SAGEMAKER_TRITON_CLOUDWATCH_METRIC_NAMESPACE`	Diperlukan jika Anda telah mengaktifkan penerbitan metrik. CloudWatch	String	<cloudwatch_metric_namespace>
`SAGEMAKER_TRITON_ADDITIONAL_ARGS`	Menambahkan argumen tambahan saat memulai Triton Server.	String	<additional_args>

Terapkan model di tepi dengan SageMaker Edge Manager

> **Warning**
>
> SageMaker Edge Manager dihentikan pada 26 April 2024. Untuk informasi selengkapnya tentang melanjutkan penerapan model Anda ke perangkat edge, lihatSageMaker Edge Manager akhir hidup.

Amazon SageMaker Edge Manager menyediakan manajemen model untuk perangkat edge sehingga Anda dapat mengoptimalkan, mengamankan, memantau, dan memelihara model pembelajaran mesin pada armada perangkat edge seperti kamera pintar, robot, komputer pribadi, dan perangkat seluler.

Mengapa Menggunakan Edge Manager?

Banyak kasus penggunaan machine learning (ML) memerlukan menjalankan model ML pada armada perangkat edge, yang memungkinkan Anda mendapatkan prediksi secara real-time, menjaga privasi pengguna akhir, dan menurunkan biaya konektivitas jaringan. Dengan meningkatnya ketersediaan

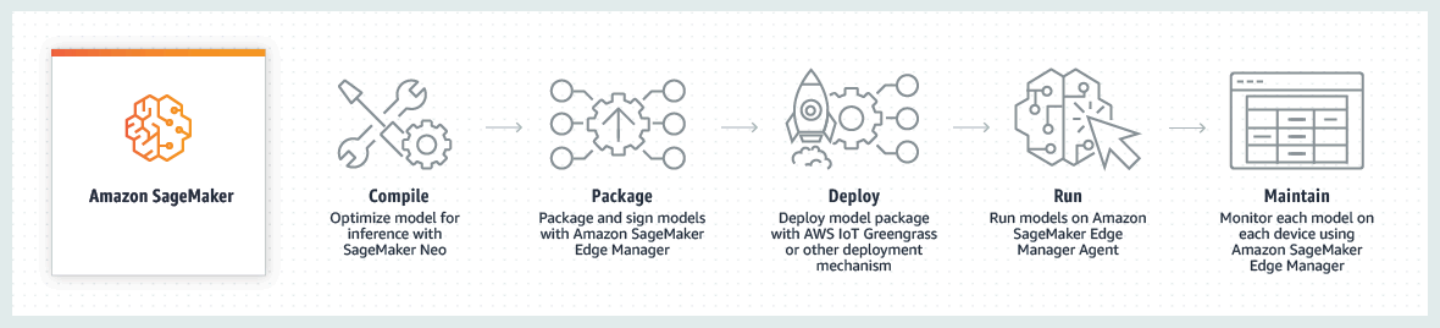
perangkat keras tepi berdaya rendah yang dirancang untuk ML, sekarang dimungkinkan untuk menjalankan beberapa model jaringan saraf kompleks pada perangkat edge.

Namun, mengoperasikan model ML pada perangkat edge sangat menantang, karena perangkat, tidak seperti instance cloud, memiliki komputasi, memori, dan konektivitas yang terbatas. Setelah model dikerahkan, Anda perlu terus memantau model, karena penyimpangan model dapat menyebabkan kualitas model membusuk dari waktu ke waktu. Memantau model di seluruh armada perangkat Anda sulit karena Anda perlu menulis kode khusus untuk mengumpulkan sampel data dari perangkat Anda dan mengenali kecondongan dalam prediksi. Selain itu, model sering dikodekan dengan keras ke dalam aplikasi. Untuk memperbarui model, Anda harus membangun kembali dan memperbarui seluruh aplikasi atau firmware perangkat, yang dapat mengganggu operasi Anda.

Dengan SageMaker Edge Manager, Anda dapat mengoptimalkan, menjalankan, memantau, dan memperbarui model pembelajaran mesin di seluruh armada perangkat di edge.

Bagaimana Cara Kerjanya?

Pada tingkat tinggi, ada lima komponen utama dalam alur kerja SageMaker Edge Manager: menyusun model dengan SageMaker Neo, mengemas model yang dikompilasi NEO, menerapkan model ke perangkat Anda, menjalankan model pada mesin SageMaker inferensi (agen Edge Manager), dan memelihara model pada perangkat.



SageMaker Edge Manager menggunakan SageMaker Neo untuk mengoptimalkan model Anda untuk perangkat keras target dalam satu klik, lalu menandatangani model Anda secara kriptografis sebelum penerapan. Dengan menggunakan SageMaker Edge Manager, Anda dapat mengambil sampel data input dan output model dari perangkat edge dan mengirimkannya ke cloud untuk pemantauan dan analisis, dan melihat dasbor yang melacak dan melaporkan secara visual tentang pengoperasian model yang diterapkan di dalam SageMaker konsol.

SageMaker Edge Manager memperluas kemampuan yang sebelumnya hanya tersedia di cloud hingga ke edge, sehingga pengembang dapat terus meningkatkan kualitas model dengan

menggunakan Amazon SageMaker Model Monitor untuk deteksi drift, kemudian memberi label ulang data dengan SageMaker Ground Truth dan melatih kembali model. SageMaker

Bagaimana Cara Menggunakan SageMaker Edge Manager?

Jika Anda adalah pengguna pertama kali SageMaker Edge Manager, kami sarankan Anda melakukan hal berikut:

1. Baca bagian Memulai - Bagian ini memandu Anda melalui pengaturan pekerjaan pengemasan tepi pertama Anda dan membuat armada pertama Anda.
2. Jelajahi contoh notebook Edge Manager Jupyter - Contoh notebook disimpan di amazon-sagemaker-examples GitHub repositori di folder sagemaker_edge_manager.

Memulai

Panduan ini menunjukkan cara menyelesaikan langkah-langkah yang diperlukan untuk mendaftar, menyebarkan, dan mengelola armada perangkat, dan cara memenuhi prasyarat Amazon SageMaker Edge Manager.

Topik

- Mengatur
- Latih, Kompilasi, dan Package Model Anda
- Membuat dan Mendaftarkan Armada dan Otentikasi Perangkat
- Unduh dan Atur Edge Manager
- Jalankan Agen

Mengatur

Sebelum Anda mulai menggunakan SageMaker Edge Manager untuk mengelola model pada armada perangkat Anda, Anda harus terlebih dahulu membuat Peran IAM untuk keduanya dan SageMaker . AWS IoT Anda juga ingin membuat setidaknya satu bucket Amazon S3 di mana Anda akan menyimpan model pra-terlatih Anda, output dari pekerjaan kompilasi SageMaker Neo Anda, serta data input dari perangkat edge Anda.

Mendaftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk secara online.

   Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

   Saat Anda mendaftar Akun AWS, Pengguna root akun AWS akan dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, tetapkan akses administratif ke pengguna administratif, dan hanya gunakan pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Membuat pengguna administratif

Setelah mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat sebuah pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke AWS Management Console sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Di halaman berikutnya, masukkan kata sandi Anda.

   Untuk bantuan masuk menggunakan pengguna root, lihat Masuk sebagai pengguna root dalam Panduan Pengguna AWS Sign-In.
2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuknya, silakan lihat Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda (konsol) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center.

2. Di Pusat Identitas IAM, berikan akses administratif ke sebuah pengguna administratif.

   Untuk mendapatkan tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, silakan lihat Mengonfigurasi akses pengguna dengan Direktori Pusat Identitas IAM default di Panduan Pengguna AWS IAM Identity Center.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal akses AWS dalam Panduan Pengguna AWS Sign-In.

Buat peran dan penyimpanan

SageMaker Edge Manager memerlukan akses ke URI bucket Amazon S3 Anda. Untuk memfasilitasi ini, buat peran IAM yang dapat berjalan SageMaker dan memiliki izin untuk mengakses Amazon S3. Menggunakan peran ini, SageMaker dapat berjalan di bawah akun Anda dan akses ke bucket Amazon S3 Anda.

Anda dapat membuat peran IAM dengan menggunakan konsol IAM, AWS SDK for Python (Boto3), atau. AWS CLI Berikut ini adalah contoh cara membuat peran IAM, melampirkan kebijakan yang diperlukan dengan konsol IAM, dan membuat bucket Amazon S3.

1. Buat peran IAM untuk Amazon SageMaker.

   a. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

   b. Di panel navigasi konsol IAM, pilih Peran, dan lalu pilih Buat peran.

   c. Untuk Pilih jenis entitas tepercaya, pilih layanan AWS.

   d. Pilih layanan yang ingin Anda perbolehkan untuk mengasumsikan peran ini. Dalam hal ini, pilih SageMaker. Kemudian, pilih Selanjutnya: Izin.

- Ini secara otomatis membuat kebijakan IAM yang memberikan akses ke layanan terkait seperti Amazon S3, Amazon CloudWatch ECR, dan Log.

e. Pilih Selanjutnya: Tanda.

f. (Opsional) Tambahkan metadata ke dalam peran dengan melampirkan tanda sebagai pasangan nilai-kunci. Untuk informasi lebih lanjut tentang penggunaan tanda dan IAM, lihat Penandaan sumber daya IAM.

g. Pilih Berikutnya: Peninjauan.

h. Ketik nama Peran.

i. Jika memungkinkan, ketikkan nama peran atau akhiran nama peran. Nama peran harus unik di akun AWS Anda. Grup tidak dibedakan berdasarkan huruf besar-kecil. Misalnya, Anda tidak dapat membuat peran dengan nama `PRODROLE` dan `prodrole`. Karena sumber daya AWS lainnya mungkin merujuk peran tersebut, Anda tidak dapat mengubah nama peran tersebut setelah nama dibuat.

j. (Opsional) Untuk Deskripsi peran, ketikkan deskripsi untuk peran baru tersebut.

k. Tinjau peran dan kemudian pilih Buat peran.

   Perhatikan ARN SageMaker Peran, yang Anda gunakan untuk membuat pekerjaan kompilasi dengan SageMaker Neo dan pekerjaan pengemasan dengan Edge Manager. Untuk mengetahui peran ARN menggunakan konsol, lakukan hal berikut:

   i. Pergi ke IamConsole: https://console.aws.amazon.com/iam/

   ii. Pilih Peran.

   iii. Cari peran yang baru saja Anda buat dengan mengetikkan nama peran di bidang pencarian.

   iv. Pilih peran.

   v. Peran ARN ada di bagian atas halaman Ringkasan.

2. Buat peran IAM untukAWS IoT.

   Peran AWS IoT IAM yang Anda buat digunakan untuk mengotorisasi objek benda Anda. Anda juga menggunakan peran IAM ARN untuk membuat dan mendaftarkan armada perangkat dengan SageMaker objek klien.

   Konfigurasikan peran IAM di AWS akun Anda untuk diambil oleh penyedia kredensi atas nama perangkat di armada perangkat Anda. Kemudian, lampirkan kebijakan untuk mengizinkan perangkat Anda berinteraksi dengan AWS IoT layanan.

Buat peran AWS IoT baik untuk pemrograman atau dengan konsol IAM, mirip dengan apa yang Anda lakukan ketika Anda membuat peran untuk. SageMaker

a. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

b. Di panel navigasi konsol IAM, pilih Peran, dan lalu pilih Buat peran.

c. Untuk Pilih jenis entitas tepercaya, pilih layanan AWS.

d. Pilih layanan yang ingin Anda perbolehkan untuk mengasumsikan peran ini. Dalam hal ini, pilih IoT. Pilih IoT sebagai Kasus Penggunaan.

e. Pilih Selanjutnya: Izin.

f. Pilih Selanjutnya: Tanda.

g. (Opsional) Tambahkan metadata ke dalam peran dengan melampirkan tanda sebagai pasangan nilai-kunci. Untuk informasi lebih lanjut tentang penggunaan tanda dan IAM, lihat Penandaan sumber daya IAM.

h. Pilih Berikutnya: Peninjauan.

i. Ketik nama Peran. Nama peran harus dimulai dengan`SageMaker`.

j. (Opsional) Untuk Deskripsi peran, ketikkan deskripsi untuk peran baru tersebut.

k. Tinjau peran dan kemudian pilih Buat peran.

l. Setelah peran dibuat, pilih Peran di konsol IAM. Cari peran yang Anda buat dengan mengetikkan nama peran di kolom Pencarian.

m. Pilih peran Anda.

n. Selanjutnya, pilih Lampirkan Kebijakan.

o. Cari `AmazonSageMakerEdgeDeviceFleetPolicy` di bidang Pencarian. Pilih `AmazonSageMakerEdgeDeviceFleetPolicy`.

p. Pilih Lampirkan kebijakan.

q. Tambahkan pernyataan kebijakan berikut ke hubungan kepercayaan:

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
        "Effect": "Allow",
        "Principal": {"Service": "credentials.iot.amazonaws.com"},
        "Action": "sts:AssumeRole"
```

```
      },
      {
        "Effect": "Allow",
        "Principal": {"Service": "sagemaker.amazonaws.com"},
        "Action": "sts:AssumeRole"
      }
  ]
}
```

Kebijakan kepercayaan adalah dokumen kebijakan JSON di mana Anda menentukan prinsip yang Anda percayai untuk mengambil peran tersebut. Untuk informasi selengkapnya tentang kebijakan kepercayaan, lihat Istilah dan konsep peran.

r. Perhatikan AWS IoT peran ARN. Anda menggunakan ARN AWS IoT Peran untuk membuat dan mendaftarkan armada perangkat. Untuk menemukan peran IAM ARN dengan konsol:

   i. Buka konsol IAM: https://console.aws.amazon.com/iam/

   ii. Pilih Peran.

   iii. Cari peran yang Anda buat dengan mengetikkan nama peran di bidang Pencarian.

   iv. Pilih peran.

   v. Peran ARN ada di halaman Ringkasan.

3. Buat bucket Amazon S3.

   SageMaker Neo dan Edge Manager mengakses model yang telah dikompilasi sebelumnya dan model yang dikompilasi dari bucket Amazon S3. Edge Manager juga menyimpan data sampel dari armada perangkat Anda di Amazon S3.

   a. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.

   b. Pilih Buat bucket.

   c. Dalam nama Bucket, masukkan nama untuk bucket Anda.

   d. Di Wilayah, pilih AWS Wilayah tempat Anda ingin ember berada.

   e. Di pengaturan Bucket untuk Blokir Akses Publik, pilih pengaturan yang ingin Anda terapkan ke bucket.

   f. Pilih Buat bucket.

   Untuk informasi selengkapnya tentang membuat bucket Amazon S3, lihat Memulai Amazon S3.

Latih, Kompilasi, dan Package Model Anda

Di bagian ini Anda akan membuat SageMaker dan objek AWS IoT klien, mengunduh model pembelajaran mesin yang telah dilatih sebelumnya, mengunggah model Anda ke bucket Amazon S3 Anda, mengkompilasi model Anda untuk perangkat target Anda SageMaker dengan Neo, dan mengemas model Anda sehingga dapat digunakan dengan agen Edge Manager.

1. Impor pustaka dan buat objek klien.

   Tutorial ini menggunakan AWS SDK for Python (Boto3) untuk membuat klien untuk berinteraksi dengan SageMaker, Amazon S3, dan. AWS IoT

   Impor Boto3, tentukan Wilayah Anda, dan inisialisasi objek klien yang Anda butuhkan seperti yang ditunjukkan pada contoh berikut:

   ```
   import boto3
   import json
   import time

   AWS_REGION = 'us-west-2'# Specify your Region
   bucket = 'bucket-name'

   sagemaker_client = boto3.client('sagemaker', region_name=AWS_REGION)
   iot_client = boto3.client('iot', region_name=AWS_REGION)
   ```

   Tentukan variabel dan tetapkan peran ARN yang Anda buat SageMaker untuk AWS IoT dan sebagai string:

   ```
   # Replace with the role ARN you created for SageMaker
   sagemaker_role_arn = "arn:aws:iam::<account>:role/*"

   # Replace with the role ARN you created for AWS IoT.
   # Note: The name must start with 'SageMaker'
   iot_role_arn = "arn:aws:iam::<account>:role/SageMaker*"
   ```

2. Latih model pembelajaran mesin.

   Lihat Melatih Model dengan Amazon SageMaker untuk informasi lebih lanjut tentang cara melatih model pembelajaran mesin menggunakan SageMaker. Anda dapat mengunggah model terlatih lokal secara opsional langsung ke bucket Amazon S3 URI.

Jika Anda belum memiliki model, Anda dapat menggunakan model pra-terlatih untuk langkah selanjutnya dalam tutorial ini. Misalnya, Anda dapat menyimpan model MobileNet V2 dari TensorFlow kerangka kerja. MobileNet V2 adalah model klasifikasi gambar yang dioptimalkan untuk aplikasi seluler. Untuk informasi lebih lanjut tentang MobileNet V2, lihat MobileNet GitHub README.

Ketik berikut ini ke Notebook Jupyter Anda untuk menyimpan model V2 yang telah dilatih sebelumnya MobileNet :

```
# Save the MobileNet V2 model to local storage
   import tensorflow as tf
   model = tf.keras.applications.MobileNetV2()
   model.save("mobilenet_v2.h5")
```

> **Note**
>
> - Jika Anda belum TensorFlow menginstal, Anda dapat melakukannya dengan menjalankan `pip install tensorflow=2.4`
> - Gunakan TensorFlow versi 2.4 atau lebih rendah untuk tutorial ini.

Model akan disimpan ke dalam `mobilenet_v2.h5` file. Sebelum mengemas model, Anda harus terlebih dahulu mengkompilasi model Anda menggunakan SageMaker Neo. Lihat Kerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukung untuk memeriksa apakah versi Anda TensorFlow (atau kerangka kerja pilihan lainnya) saat ini didukung oleh SageMaker Neo.

SageMaker Neo membutuhkan model untuk disimpan sebagai file TAR terkompresi. Kemas ulang sebagai file TAR terkompresi (*.tar.gz):

```
# Package MobileNet V2 model into a TAR file
   import tarfile

   tarfile_name='mobilenet-v2.tar.gz'

   with tarfile.open(tarfile_name, mode='w:gz') as archive:
       archive.add('mobilenet-v2.h5')
```

3. Unggah model Anda ke Amazon S3.

Setelah Anda memiliki model pembelajaran mesin, simpan di ember Amazon S3. Contoh berikut menggunakan AWS CLI perintah untuk mengunggah model ke bucket Amazon S3 yang Anda buat sebelumnya di direktori bernama models. Ketik berikut ini ke Notebook Jupyter Anda:

```
!aws s3 cp mobilenet-v2.tar.gz s3://{bucket}/models/
```

4. Kompilasi model Anda dengan SageMaker Neo.

   Kompilasi model pembelajaran mesin Anda dengan SageMaker Neo untuk perangkat edge. Anda perlu mengetahui URI bucket Amazon S3 tempat Anda menyimpan model terlatih, kerangka kerja pembelajaran mesin yang Anda gunakan untuk melatih model, bentuk input model, dan perangkat target Anda.

   Untuk model MobileNet V2, gunakan yang berikut ini:

   ```
   framework = 'tensorflow'
   target_device = 'jetson_nano'
   data_shape = '{"data":[1,3,224,224]}'
   ```

   SageMaker Neo membutuhkan bentuk input model tertentu dan format model berdasarkan kerangka pembelajaran mendalam yang Anda gunakan. Untuk informasi selengkapnya tentang cara menyimpan model Anda, lihatBentuk data input apa yang diharapkan SageMaker Neo?. Untuk informasi selengkapnya tentang perangkat dan kerangka kerja yang didukung oleh Neo, lihatKerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukung.

   Gunakan `CreateCompilationJob` API untuk membuat pekerjaan kompilasi dengan SageMaker Neo. Berikan nama untuk pekerjaan kompilasi, ARN SageMaker Peran, URI Amazon S3 tempat model Anda disimpan, bentuk input model, nama kerangka kerja, URI Amazon S3 tempat Anda SageMaker ingin menyimpan model yang dikompilasi, dan target perangkat edge Anda.

   ```
   # Specify the path where your model is stored
   model_directory = 'models'
   s3_model_uri = 's3://{}/{}/{}'.format(bucket, model_directory, tarfile_name)

   # Store compiled model in S3 within the 'compiled-models' directory
   compilation_output_dir = 'compiled-models'
   s3_output_location = 's3://{}/{}/'.format(bucket, compilation_output_dir)
   ```

```
# Give your compilation job a name
compilation_job_name = 'getting-started-demo'

sagemaker_client.create_compilation_job(CompilationJobName=compilation_job_name,
                                        RoleArn=sagemaker_role_arn,
                                        InputConfig={
                                            'S3Uri': s3_model_uri,
                                            'DataInputConfig': data_shape,
                                            'Framework' : framework.upper()},
                                        OutputConfig={
                                            'S3OutputLocation': s3_output_location,
                                            'TargetDevice': target_device},
                                        StoppingCondition={'MaxRuntimeInSeconds':
 900})
```

5. Package model kompilasi Anda.

   Pekerjaan pengemasan membutuhkan model yang SageMaker dikompilasi Neo dan membuat perubahan apa pun yang diperlukan untuk menerapkan model dengan mesin inferensi, agen Edge Manager. Untuk mengemas model Anda, buat pekerjaan pengemasan tepi dengan `create_edge_packaging` API atau SageMaker konsol.

   Anda perlu memberikan nama yang Anda gunakan untuk pekerjaan kompilasi Neo Anda, nama untuk pekerjaan pengemasan, peran ARN (lihat Mengatur bagian), nama untuk model, versi model, dan URI bucket Amazon S3 untuk output pekerjaan pengemasan. Perhatikan bahwa nama pekerjaan pengemasan Edge Manager peka huruf besar/kecil. Berikut ini adalah contoh cara membuat pekerjaan pengemasan menggunakan API.

   ```
   edge_packaging_name='edge-packaging-demo'
   model_name="sample-model"
   model_version="1.1"
   ```

   Tentukan URI Amazon S3 tempat Anda ingin menyimpan model paket.

   ```
   # Output directory where you want to store the output of the packaging job
   packaging_output_dir = 'packaged_models'
   packaging_s3_output = 's3://{}/{}'.format(bucket, packaging_output_dir)
   ```

   Gunakan `CreateEdgePackagingJob` untuk mengemas model yang dikompilasi NEO Anda. Berikan nama untuk pekerjaan pengemasan tepi Anda dan nama yang

Anda berikan untuk pekerjaan kompilasi Anda (dalam contoh ini, itu disimpan dalam variabel`compilation_job_name`). Berikan juga nama untuk model Anda, versi untuk model Anda (ini digunakan untuk membantu Anda melacak versi model apa yang Anda gunakan), dan URI S3 tempat Anda SageMaker ingin menyimpan model yang dikemas.

```
sagemaker_client.create_edge_packaging_job(
                    EdgePackagingJobName=edge_packaging_name,
                    CompilationJobName=compilation_job_name,
                    RoleArn=sagemaker_role_arn,
                    ModelName=model_name,
                    ModelVersion=model_version,
                    OutputConfig={
                        "S3OutputLocation": packaging_s3_output
                        }
                    )
```

Membuat dan Mendaftarkan Armada dan Otentikasi Perangkat

Di bagian ini Anda akan membuat objek AWS IoT benda Anda, membuat armada perangkat, mendaftarkan armada perangkat Anda sehingga dapat berinteraksi dengan cloud, membuat sertifikat X.509 untuk mengautentikasi perangkat AndaAWS IoT Core, mengaitkan alias peran dengan AWS IoT yang dihasilkan saat Anda membuat armada Anda, dapatkan titik akhir AWS khusus akun Anda untuk penyedia kredensi, dapatkan file Amazon Root CA resmi, dan unggah file Amazon CA ke Amazon S3.

1. Buat AWS IoT sesuatu.

   SageMaker Edge Manager memanfaatkan AWS IoT Core layanan untuk memfasilitasi koneksi antara perangkat edge dan titik akhir di AWS cloud. Anda dapat memanfaatkan AWS IoT fungsionalitas yang ada setelah mengatur perangkat agar berfungsi dengan Edge Manager.

   Untuk menghubungkan perangkat AndaAWS IoT, Anda perlu membuat objek AWS IoT benda, membuat dan mendaftarkan sertifikat klien dengan AWS IoT, dan membuat dan mengonfigurasi peran IAM untuk perangkat Anda.

   Pertama, buat AWS IoT benda benda dengan AWS IoT klien (`iot_client`) yang Anda buat sebelumnya dengan Boto3. Contoh berikut menunjukkan cara membuat dua benda benda:

   ```
   iot_thing_name = 'sample-device'
   ```

```
iot_thing_type = 'getting-started-demo'

iot_client.create_thing_type(
    thingTypeName=iot_thing_type
)

# Create an AWS IoT thing objects
iot_client.create_thing(
    thingName=iot_thing_name,
    thingTypeName=iot_thing_type
)
```

2. Buat armada perangkat Anda.

   Buat armada perangkat dengan objek SageMaker klien yang ditentukan pada langkah sebelumnya. Anda juga dapat menggunakan SageMaker konsol untuk membuat armada perangkat.

   ```
   import time
   device_fleet_name="demo-device-fleet" + str(time.time()).split('.')[0]
   device_name="sagemaker-edge-demo-device" + str(time.time()).split('.')[0]
   ```

   Tentukan peran IoT Anda ARN. Ini memungkinkan AWS IoT memberikan kredensil sementara ke perangkat.

   ```
   device_model_directory='device_output'
   s3_device_fleet_output = 's3://{}/{}'.format(bucket, device_model_directory)

   sagemaker_client.create_device_fleet(
       DeviceFleetName=device_fleet_name,
       RoleArn=iot_role_arn, # IoT Role ARN specified in previous step
       OutputConfig={
           'S3OutputLocation': s3_device_fleet_output
       }
   )
   ```

   Alias AWS IoT peran dibuat saat Anda membuat armada perangkat. Alias peran ini dikaitkan dengan AWS IoT penggunaan `iot_client` objek di langkah selanjutnya.

3. Daftarkan armada perangkat Anda.

Untuk berinteraksi dengan cloud, Anda perlu mendaftarkan perangkat Anda ke SageMaker Edge Manager. Dalam contoh ini, Anda mendaftarkan satu perangkat dengan armada yang Anda buat. Untuk mendaftarkan perangkat, Anda perlu memberikan nama perangkat dan nama AWS IoT benda seperti yang ditunjukkan pada contoh berikut:

```
# Device name should be 36 characters
device_name = "sagemaker-edge-demo-device" + str(time.time()).split('.')[0]

sagemaker_client.register_devices(
    DeviceFleetName=device_fleet_name,
    Devices=[
        {
            "DeviceName": device_name,
            "IotThingName": iot_thing_name
        }
    ]
)
```

4. Buat sertifikat X.509.

Setelah membuat objek AWS IoT benda, Anda harus membuat sertifikat perangkat X.509 untuk objek benda Anda. Sertifikat ini mengautentikasi perangkat Anda keAWS IoT Core.

Gunakan berikut ini untuk membuat kunci pribadi, kunci publik, dan file sertifikat X.509 menggunakan AWS IoT client defined () `iot_client` sebelumnya.

```
# Creates a 2048-bit RSA key pair and issues an X.509 # certificate
# using the issued public key.
create_cert = iot_client.create_keys_and_certificate(
    setAsActive=True
)

# Get certificate from dictionary object and save in its own
with open('./device.pem.crt', 'w') as f:
    for line in create_cert['certificatePem'].split('\n'):
        f.write(line)
        f.write('\n')
# Get private key from dictionary object and save in its own
with open('./private.pem.key', 'w') as f:
    for line in create_cert['keyPair']['PrivateKey'].split('\n'):
        f.write(line)
```

```
        f.write('\n')
# Get a private key from dictionary object and save in its own
with open('./public.pem.key', 'w') as f:
    for line in create_cert['keyPair']['PublicKey'].split('\n'):
        f.write(line)
        f.write('\n')
```

5. Kaitkan alias peran denganAWS IoT.

   Saat Anda membuat armada perangkat dengan SageMaker (`sagemaker_client.create_device_fleet()`), alias peran dibuat untuk Anda. Alias AWS IoT peran menyediakan mekanisme untuk perangkat yang terhubung untuk mengautentikasi AWS IoT menggunakan sertifikat X.509, dan kemudian mendapatkan AWS kredensil berumur pendek dari peran IAM yang terkait dengan alias peran. AWS IoT Alias peran memungkinkan Anda untuk mengubah peran perangkat tanpa harus memperbarui perangkat. Gunakan `DescribeDeviceFleet` untuk mendapatkan nama alias peran dan ARN.

   ```
   # Print Amazon Resource Name (ARN) and alias that has access
   # to AWS Internet of Things (IoT).
   sagemaker_client.describe_device_fleet(DeviceFleetName=device_fleet_name)

   # Store iot role alias string in a variable
   # Grabs role ARN
   full_role_alias_name =
    sagemaker_client.describe_device_fleet(DeviceFleetName=device_fleet_name)
   ['IotRoleAlias']
   start_index = full_role_alias_name.find('SageMaker') # Find beginning of role name

   role_alias_name = full_role_alias_name[start_index:]
   ```

   Gunakan `iot_client` untuk memfasilitasi mengaitkan alias peran yang dihasilkan dari pembuatan armada perangkat dengan: AWS IoT

   ```
   role_alias = iot_client.describe_role_alias(
                       roleAlias=role_alias_name)
   ```

   Untuk informasi selengkapnya tentang alias peran IAM, lihat Alias peran memungkinkan akses ke layanan yang tidak digunakan.

Anda membuat dan mendaftarkan sertifikat dengan AWS IoT sebelumnya untuk otentikasi perangkat Anda berhasil. Sekarang, Anda perlu membuat dan melampirkan kebijakan ke sertifikat untuk mengotorisasi permintaan token keamanan.

```
alias_policy = {
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iot:AssumeRoleWithCertificate",
    "Resource": role_alias['roleAliasDescription']['roleAliasArn']
  }
}

policy_name = 'aliaspolicy-'+ str(time.time()).split('.')[0]
aliaspolicy = iot_client.create_policy(policyName=policy_name,
                                       policyDocument=json.dumps(alias_policy))

# Attach policy
iot_client.attach_policy(policyName=policy_name,
                         target=create_cert['certificateArn'])
```

6. Dapatkan titik akhir AWS khusus akun Anda untuk penyedia kredensional.

   Perangkat edge membutuhkan titik akhir untuk mengasumsikan kredensil. Dapatkan titik akhir AWS khusus akun Anda untuk penyedia kredensi.

   ```
   # Get the unique endpoint specific to your AWS account that is making the call.
   iot_endpoint = iot_client.describe_endpoint(
       endpointType='iot:CredentialProvider'
   )

   endpoint="https://{}/role-aliases/{}/
   credentials".format(iot_endpoint['endpointAddress'],role_alias_name)
   ```

7. Dapatkan file CA root Amazon resmi dan unggah ke bucket Amazon S3.

   Gunakan yang berikut ini di Notebook Jupyter Anda atau AWS CLI (jika Anda menggunakan terminal Anda, hapus '!' fungsi sihir):

   ```
   !wget https://www.amazontrust.com/repository/AmazonRootCA1.pem
   ```

Gunakan titik akhir untuk membuat permintaan HTTPS ke penyedia kredensi untuk mengembalikan token keamanan. Contoh perintah berikut menggunakan`curl`, tetapi Anda dapat menggunakan klien HTTP apa pun.

```
!curl --cert device.pem.crt --key private.pem.key --cacert AmazonRootCA1.pem
 $endpoint
```

Jika sertifikat diverifikasi, unggah kunci dan sertifikat ke URI bucket Amazon S3 Anda:

```
!aws s3 cp private.pem.key s3://{bucket}/authorization-files/
!aws s3 cp device.pem.crt s3://{bucket}/authorization-files/
!aws s3 cp AmazonRootCA1.pem s3://{bucket}/authorization-files/
```

Bersihkan direktori kerja Anda dengan memindahkan kunci dan sertifikat Anda ke direktori yang berbeda:

```
# Optional - Clean up working directory
!mkdir authorization-files
!mv private.pem.key device.pem.crt AmazonRootCA1.pem authorization-files/
```

Unduh dan Atur Edge Manager

Agen Edge Manager adalah mesin inferensi untuk perangkat edge Anda. Gunakan agen untuk membuat prediksi dengan model yang dimuat ke perangkat tepi Anda. Agen juga mengumpulkan metrik model dan menangkap data pada interval tertentu.

Di bagian ini Anda akan mengatur perangkat Anda dengan agen. Untuk melakukannya, pertama-tama salin artefak rilis dan tandatangani sertifikat root dari bucket rilis secara lokal ke mesin Anda. Setelah Anda membuka zip artefak rilis, unggah ke Amazon S3. Selanjutnya, tentukan dan simpan file konfigurasi untuk agen. Template disediakan untuk Anda salin dan tempel. Terakhir, salin artefak rilis, file konfigurasi, dan kredensil ke perangkat Anda.

1. Unduh agen SageMaker Edge Manager.

   Agen dirilis dalam format biner untuk sistem operasi yang didukung. Contoh ini menjalankan inferensi pada Jetson Nano yang menggunakan sistem operasi Linux dan memiliki arsitektur

ARM64. Untuk informasi selengkapnya tentang apa yang digunakan oleh sistem operasi dan arsitektur perangkat yang didukung, lihatPerangkat yang Didukung, Arsitektur Chip, dan Sistem.

Ambil versi binari terbaru dari bucket rilis SageMaker Edge Manager dari Wilayah us-barat-2.

```
!aws s3 ls s3://sagemaker-edge-release-store-us-west-2-linux-armv8/Releases/ | sort
 -r
```

Ini mengembalikan artefak rilis yang diurutkan berdasarkan versinya.

```
                           PRE 1.20210512.96da6cc/
                           PRE 1.20210305.a4bc999/
                           PRE 1.20201218.81f481f/
                           PRE 1.20201207.02d0e97/
```

Versi ini memiliki format berikut:<MAJOR_VERSION>.<YYYY-MM-DD>.<SHA-7>. Ini terdiri dari tiga komponen:

- <MAJOR_VERSION>: Versi rilis. Versi rilis saat ini diatur ke1.
- <YYYY-MM-DD>: Cap waktu pelepasan artefak.
- <SHA-7>: ID komit repositori tempat rilis dibuat.

Salin file TAR zip secara lokal atau ke perangkat Anda secara langsung. Contoh berikut menunjukkan cara menyalin artefak rilis terbaru pada saat dokumen ini dirilis.

```
!aws s3 cp s3://sagemaker-edge-release-store-us-west-2-linux-x64/
Releases/1.20201218.81f481f/1.20201218.81f481f.tgz ./
```

Setelah Anda memiliki artefak, unzip file TAR zip. Berikut ini membuka ritsleting file TAR dan menyimpannya dalam direktori bernama: agent_demo

```
!mkdir agent_demo
!tar -xvzf 1.20201218.81f481f.tgz -C ./agent_demo
```

Unggah artefak rilis agen ke bucket Amazon S3 Anda. Contoh kode berikut menyalin konten di dalamnya agent_demo dan mengunggahnya ke direktori dalam bucket Amazon S3 Anda yang disebut: agent_demo

```
!aws s3 cp --recursive ./agent_demo s3://{bucket}/agent_demo
```

Anda juga memerlukan sertifikat root penandatanganan dari ember rilis:

```
!aws s3 cp s3://sagemaker-edge-release-store-us-west-2-linux-x64/Certificates/us-
west-2/us-west-2.pem ./
```

Unggah sertifikat root penandatanganan ke bucket Amazon S3 Anda:

```
!aws s3 cp us-west-2.pem s3://{bucket}/authorization-files/
```

2. Tentukan file konfigurasi agen SageMaker Edge Manager.

   Pertama, tentukan file konfigurasi agen sebagai berikut:

```
sagemaker_edge_config = {
    "sagemaker_edge_core_device_name": "device_name",
    "sagemaker_edge_core_device_fleet_name": "device_fleet_name",
    "sagemaker_edge_core_capture_data_buffer_size": 30,
    "sagemaker_edge_core_capture_data_push_period_seconds": 4,
    "sagemaker_edge_core_folder_prefix": "demo_capture",
    "sagemaker_edge_core_region": "us-west-2",
    "sagemaker_edge_core_root_certs_path": "/agent_demo/certificates",
    "sagemaker_edge_provider_aws_ca_cert_file": "/agent_demo/iot-credentials/
AmazonRootCA1.pem",
    "sagemaker_edge_provider_aws_cert_file": "/agent_demo/iot-credentials/
device.pem.crt",
    "sagemaker_edge_provider_aws_cert_pk_file": "/agent_demo/iot-credentials/
private.pem.key",
    "sagemaker_edge_provider_aws_iot_cred_endpoint": "endpoint",
    "sagemaker_edge_provider_provider": "Aws",
    "sagemaker_edge_provider_s3_bucket_name": bucket,
    "sagemaker_edge_core_capture_data_destination": "Cloud"
}
```

   Ganti yang berikut ini:

   - "device_name"dengan nama perangkat Anda (string ini disimpan di langkah sebelumnya dalam variabel bernamadevice_name).

- `"device_fleet_name"`dengan nama armada perangkat Anda (string ini disimpan langkah sebelumnya dalam variabel bernama`device_fleet_name`)
- `"endpoint"`dengan titik akhir AWS khusus akun Anda untuk penyedia kredensional (string ini disimpan di langkah sebelumnya dalam variabel bernama). `endpoint`

Selanjutnya, simpan sebagai file JSON:

```
edge_config_file = open("sagemaker_edge_config.json", "w")
json.dump(sagemaker_edge_config, edge_config_file, indent = 6)
edge_config_file.close()
```

Unggah file konfigurasi ke bucket Amazon S3 Anda:

```
!aws s3 cp sagemaker_edge_config.json s3://{bucket}/
```

3. Salin artefak rilis, file konfigurasi, dan kredensional ke perangkat Anda.

Instruksi berikut dilakukan pada perangkat tepi itu sendiri.

> **Note**
>
> Anda harus terlebih dahulu menginstal Python, theAWS SDK for Python (Boto3), dan AWS CLI on your edge device.

Buka terminal di perangkat Anda. Buat folder untuk menyimpan artefak rilis, kredensialmu, dan file konfigurasi.

```
mkdir agent_demo
cd agent_demo
```

Salin konten artefak rilis yang disimpan di bucket Amazon S3 ke perangkat:

```
# Copy release artifacts
aws s3 cp s3://<bucket-name>/agent_demo/ ./ --recursive
```

(Isi artefak rilis disimpan dalam direktori yang disebut `agent_demo` pada langkah sebelumnya). Ganti `<bucket-name>` dan `agent_demo` dengan nama bucket Amazon S3 Anda dan jalur file ke artefak rilis Anda, masing-masing.

Buka `/bin` direktori dan buat file biner dapat dieksekusi:

```
cd bin

chmod +x sagemaker_edge_agent_binary
chmod +x sagemaker_edge_agent_client_example

cd agent_demo
```

Buat direktori untuk menyimpan AWS IoT kredensil Anda dan salin kredensional Anda dari bucket Amazon S3 ke perangkat edge Anda (gunakan yang sama pada yang Anda tentukan dalam variabel: `bucket`

```
mkdir iot-credentials
cd iot-credentials

aws s3 cp s3://<bucket-name>/authorization-files/AmazonRootCA1.pem ./
aws s3 cp s3://<bucket-name>/authorization-files/device.pem.crt ./
aws s3 cp s3://<bucket-name>/authorization-files/private.pem.key ./

cd ../
```

Buat direktori untuk menyimpan sertifikat root penandatanganan model Anda:

```
mkdir certificates

cd certificates

aws s3 cp s3://<bucket-name>/authorization-files/us-west-2.pem ./

cd agent_demo
```

Salin file konfigurasi Anda ke perangkat Anda:

```
#Download config file from S3
aws s3 cp s3://<bucket-name>/sagemaker_edge_config.json ./
```

```
cd agent_demo
```

`agent_demo`Direktori Anda di perangkat edge Anda akan terlihat mirip dengan yang berikut ini:

```
###agent_demo
|    ### bin
|        ### sagemaker_edge_agent_binary
|        ### sagemaker_edge_agent_client_example
|    ### sagemaker_edge_config.json
|    ### certificates
|        ###us-west-2.pem
|    ### iot-credentials
|        ### AmazonRootCA1.pem
|        ### device.pem.crt
|        ### private.pem.key
|    ### docs
|        ### api
|        ### examples
|    ### ATTRIBUTIONS.txt
|    ### LICENSE.txt
|    ### RELEASE_NOTES.md
```

Jalankan Agen

Pada bagian ini Anda akan menjalankan agen sebagai biner menggunakan gRPC, dan memeriksa apakah perangkat dan armada Anda bekerja dan mengumpulkan data sampel.

1. Luncurkan agen.

   Agen SageMaker Edge Manager dapat dijalankan sebagai proses mandiri dalam bentuk biner executable Executable dan Linkable Format (ELF) atau dapat ditautkan sebagai Dynamic Shared Object (.dll). Berjalan sebagai biner yang dapat dieksekusi mandiri adalah mode yang disukai dan didukung di Linux.

   Contoh ini menggunakan gRPC untuk menjalankan agen. gRPC adalah kerangka kerja Remote Procedure Call (RPC) berkinerja tinggi open source yang dapat berjalan di lingkungan apa pun. Untuk informasi selengkapnya tentang gRPC, lihat dokumentasi gRPC.

   Untuk menggunakan gRPC, lakukan langkah-langkah berikut:

a. Tentukan layanan dalam file.proto.

b. Hasilkan kode server dan klien menggunakan compiler buffer protokol.

c. Gunakan Python (atau bahasa lain yang didukung oleh gRPC) gRPC API untuk menulis server untuk layanan Anda.

d. Gunakan Python (atau bahasa lain yang didukung oleh gRPC) gRPC API untuk menulis klien untuk layanan Anda.

Artefak rilis yang Anda unduh berisi aplikasi gRPC yang siap untuk Anda jalankan agen. Contoh ini terletak di dalam `/bin` direktori artefak rilis Anda. `sagemaker_edge_agent_binary`Biner executable ada di direktori ini.

Untuk menjalankan agen dengan contoh ini, berikan path ke file socket Anda (.sock) dan file JSON .config:

```
./bin/sagemaker_edge_agent_binary -a /tmp/sagemaker_edge_agent_example.sock -c
 sagemaker_edge_config.json
```

2. Periksa perangkat Anda.

Periksa apakah perangkat Anda terhubung dan mengambil sampel data. Melakukan pemeriksaan berkala, secara manual atau otomatis, memungkinkan Anda memeriksa apakah perangkat atau armada Anda berfungsi dengan baik.

Berikan nama armada tempat perangkat berada dan pengidentifikasi perangkat unik. Dari mesin lokal Anda, jalankan yang berikut ini:

```
sagemaker_client.describe_device(
    DeviceName=device_name,
    DeviceFleetName=device_fleet_name
)
```

Untuk model yang diberikan, Anda dapat melihat nama, versi model, waktu sampel terbaru, dan kapan inferensi terakhir dibuat.

```
{
  "DeviceName": "sample-device",
  "DeviceFleetName": "demo-device-fleet",
  "IoTThingName": "sample-thing-name-1",
```

```
  "RegistrationTime": 1600977370,
  "LatestHeartbeat": 1600977370,
  "Models":[
    {
        "ModelName": "mobilenet_v2.tar.gz",
        "ModelVersion": "1.1",
        "LatestSampleTime": 1600977370,
        "LatestInference": 1600977370
    }
  ]
}
```

Stempel waktu yang disediakan oleh `LastetHeartbeat` menunjukkan sinyal terakhir yang diterima dari perangkat. `LatestSampleTime`dan `LatestInference` jelaskan cap waktu dari sampel data terakhir dan inferensi, masing-masing.

3. Periksa armada Anda.

   Periksa apakah armada Anda bekerja sama`GetDeviceFleetReport`. Berikan nama armada yang dimiliki perangkat tersebut.

```
sagemaker_client.get_device_fleet_report(
    DeviceFleetName=device_fleet_name
)
```

Untuk model tertentu, Anda dapat melihat nama, versi model, waktu sampel terbaru, dan kapan inferensi terakhir dibuat, bersama dengan URI bucket Amazon S3 tempat sampel data disimpan.

```
# Sample output
{
 "DeviceFleetName": "sample-device-fleet",
 "DeviceFleetArn": "arn:aws:sagemaker:us-west-2:9999999999:device-fleet/sample-
fleet-name",
 "OutputConfig": {
             "S3OutputLocation": "s3://fleet-bucket/package_output",
  },
 "AgentVersions":[{"Version": "1.1", "AgentCount": 2}]}
 "DeviceStats": {"Connected": 2, "Registered": 2},
 "Models":[{
            "ModelName": "sample-model",
            "ModelVersion": "1.1",
            "OfflineDeviceCount": 0,
```

```
            "ConnectedDeviceCount": 2,
            "ActiveDeviceCount": 2,
            "SamplingDeviceCount": 100
            }]
}
```

Mengatur Perangkat dan Armada

Armada adalah kumpulan perangkat yang dikelompokkan secara logis yang dapat Anda gunakan untuk mengumpulkan dan menganalisis data. Anda dapat menggunakan SageMaker Edge Manager untuk mengoperasikan model pembelajaran mesin pada armada kamera pintar, speaker pintar, robot, dan perangkat tepi lainnya.

Buat armada dan daftarkan perangkat Anda secara terprogram dengan AWS SDK for Python (Boto3) atau melalui konsol. SageMaker

Topik

- Buat Armada
- Daftarkan Perangkat
- Periksa Status

Buat Armada

Anda dapat membuat armada secara terprogram dengan AWS SDK for Python (Boto3) atau melalui SageMaker konsol https://console.aws.amazon.com/sagemaker.

Buat Armada (Boto3)

Gunakan `CreateDeviceFleet` API untuk membuat armada. Tentukan nama untuk armada, ARN AWS IoT Peran Anda untuk `RoleArn` bidang, serta URI Amazon S3 tempat perangkat menyimpan data sampel.

Anda dapat secara opsional menyertakan deskripsi armada, tag, dan ID AWS KMS Kunci.

```
import boto3

# Create SageMaker client so you can interact and manage SageMaker resources
sagemaker_client = boto3.client("sagemaker", region_name="aws-region")
```

```
sagemaker_client.create_device_fleet(
    DeviceFleetName="sample-fleet-name",
    RoleArn="arn:aws:iam::999999999:role/rolename", # IoT Role ARN
    Description="fleet description",
    OutputConfig={
        S3OutputLocation="s3://bucket/",
        KMSKeyId: "1234abcd-12ab-34cd-56ef-1234567890ab",
    },
        Tags=[
        {
            "Key": "string",
            "Value" : "string"
         }
     ],
)
```

Alias AWS IoT Peran dibuat untuk Anda saat Anda membuat armada perangkat. Alias AWS IoT peran menyediakan mekanisme untuk perangkat yang terhubung untuk mengautentikasi AWS IoT menggunakan sertifikat X.509 dan kemudian mendapatkan AWS kredensil berumur pendek dari peran IAM yang terkait dengan alias peran. AWS IoT

Gunakan `DescribeDeviceFleet` untuk mendapatkan nama alias peran dan ARN.

```
# Print Amazon Resource Name (ARN) and alias that has access
# to AWS Internet of Things (IoT).
sagemaker_client.describe_device_fleet(DeviceFleetName=device_fleet_name)
['IotRoleAlias']
```

Gunakan `DescribeDeviceFleet` API untuk mendapatkan deskripsi armada yang Anda buat.

```
sagemaker_client.describe_device_fleet(
    DeviceFleetName="sample-fleet-name"
)
```

Secara default, ia mengembalikan nama armada, armada perangkat ARN, URI bucket Amazon S3, peran IAM, alias peran yang dibuat, stempel waktu saat armada dibuatAWS IoT, dan stempel waktu kapan armada terakhir dimodifikasi.

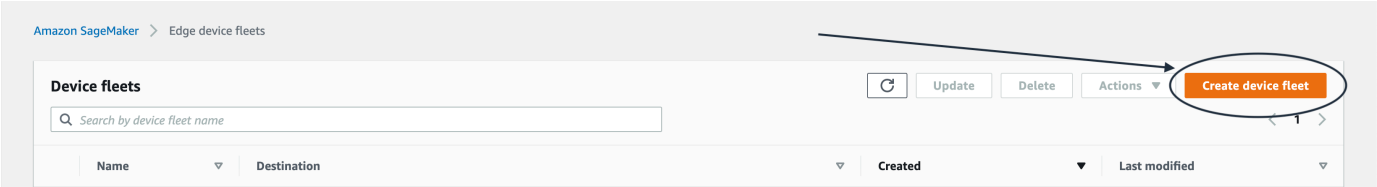
```
{ "DeviceFleetName": "sample-fleet-name",
  "DeviceFleetArn": "arn:aws:sagemaker:us-west-2:9999999999:device-fleet/sample-fleet-
name",
```

```
  "IAMRole": "arn:aws:iam::999999999:role/rolename",
  "Description": "this is a sample fleet",
  "IoTRoleAlias": "arn:aws:iot:us-west-2:9999999999:rolealias/SagemakerEdge-sample-
fleet-name"
  "OutputConfig": {
              "S3OutputLocation": "s3://bucket/folder",
              "KMSKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
   },
   "CreationTime": "1600977370",
   "LastModifiedTime": "1600977370"}
```

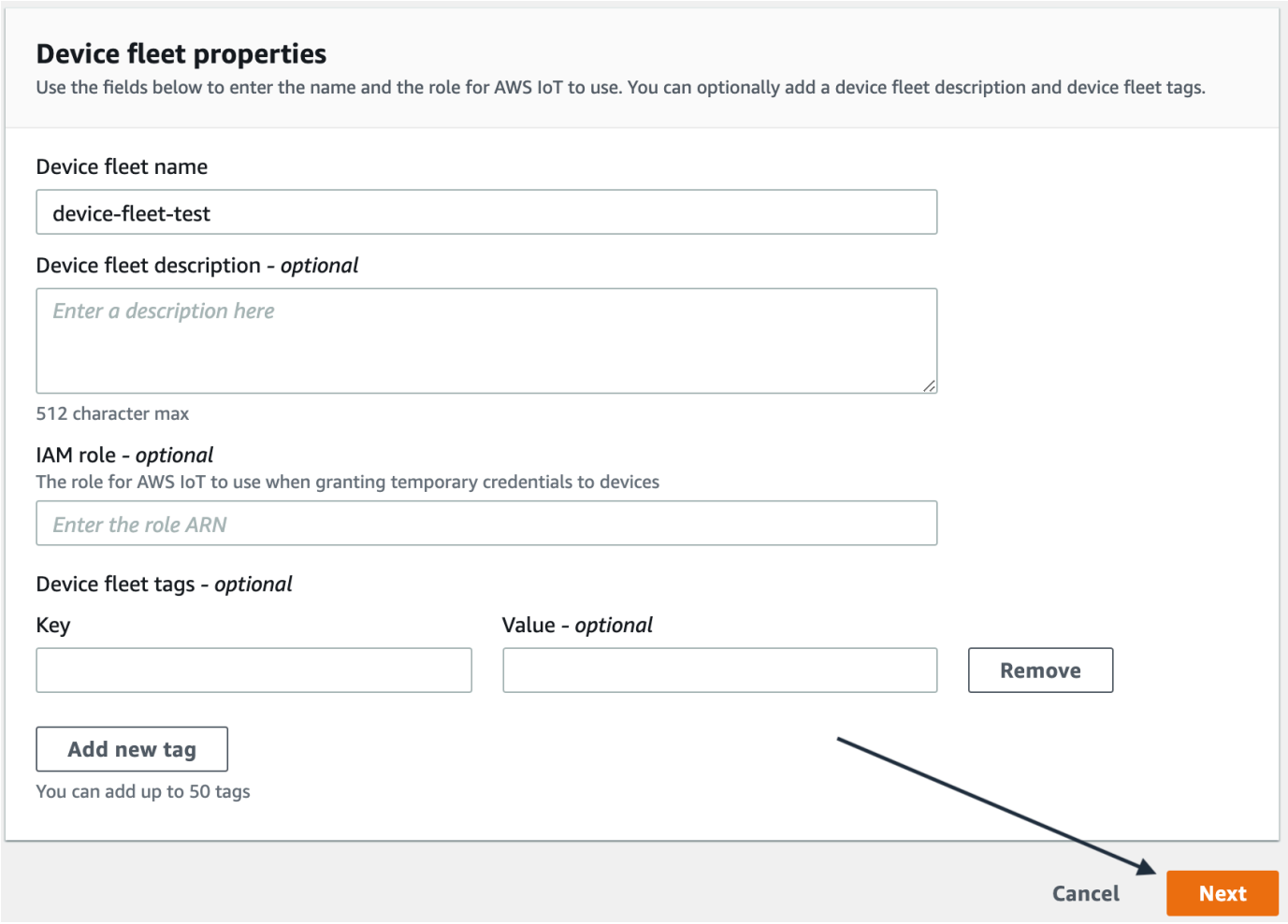
Buat Armada (Konsol)

Anda dapat membuat pekerjaan pengemasan Edge Manager menggunakan SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker.

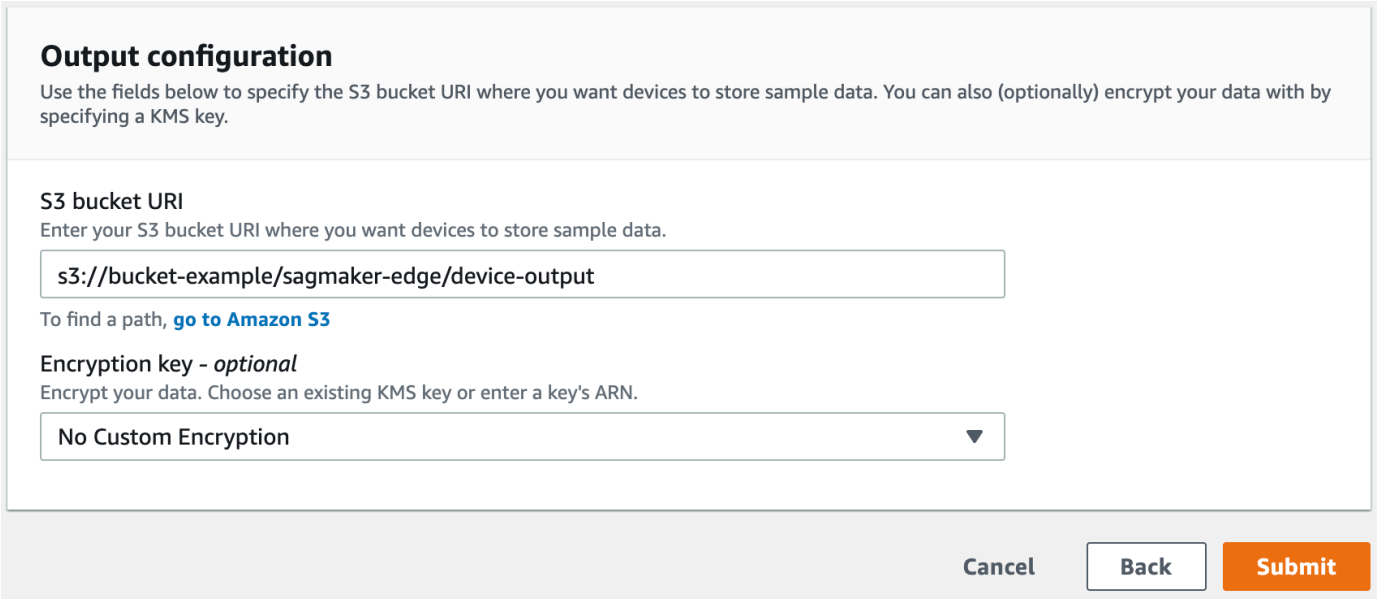
1. Di SageMaker konsol, pilih Edge Manager dan kemudian pilih armada perangkat Edge.
2. Pilih Buat armada perangkat.



3. Masukkan nama untuk armada perangkat di bidang Nama armada perangkat. Pilih Berikutnya.



4. Pada halaman konfigurasi Output, tentukan URI bucket Amazon S3 tempat Anda ingin menyimpan data sampel dari armada perangkat. Anda dapat menambahkan kunci enkripsi juga dengan memilih kunci yang ada AWS KMS dari daftar dropdown atau dengan memasukkan ARN kunci. Pilih Kirim.



5. Pilih nama armada perangkat Anda untuk diarahkan ke detail armada perangkat. Halaman ini menampilkan nama armada perangkat, ARN, deskripsi (jika Anda memberikannya), tanggal armada dibuat, terakhir kali armada diubah, URI bucket Amazon S3, ID kunci (jika disediakan)AWS KMS, alias (jika disediakan)AWS IoT, dan peran IAM. Jika Anda menambahkan tag, tag akan muncul di bagian Tag armada perangkat.

Daftarkan Perangkat

> **Important**
>
> Pendaftaran perangkat diperlukan untuk menggunakan bagian mana pun dari SageMaker Edge Manager.

Anda dapat membuat armada secara terprogram dengan AWS SDK for Python (Boto3) atau melalui SageMaker konsol di https://console.aws.amazon.com/sagemaker.

Daftarkan Perangkat (Boto3)

Untuk mendaftarkan perangkat Anda, pertama-tama buat dan daftarkan objek AWS IoT benda dan konfigurasikan peran IAM. SageMaker Edge Manager memanfaatkan AWS IoT Core layanan untuk memfasilitasi koneksi antara perangkat edge dan cloud. Anda dapat memanfaatkan AWS IoT fungsionalitas yang ada setelah mengatur perangkat agar berfungsi dengan Edge Manager.

Untuk menghubungkan perangkat Anda ke AWS IoT Anda perlu membuat objek AWS IoT benda, membuat dan mendaftarkan sertifikat klien denganAWS IoT, dan membuat dan mengkonfigurasi peran IAM untuk perangkat Anda.

Lihat Panduan Memulai untuk contoh mendalam atau layanan Jelajahi AWS IoT Core dalam tutorial langsung.

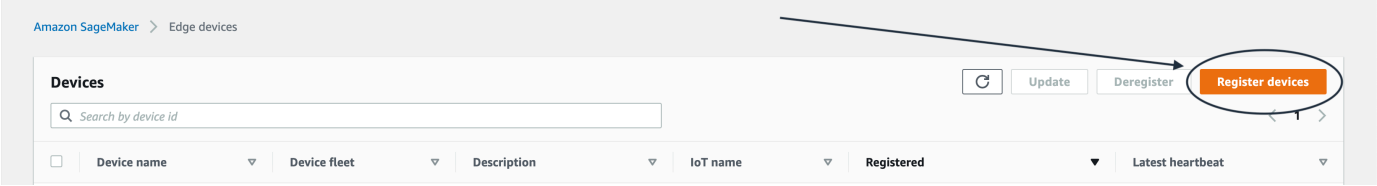
Gunakan `RegisterDevices` API untuk mendaftarkan perangkat Anda. Berikan nama armada yang Anda inginkan perangkat menjadi bagiannya, serta nama untuk perangkat tersebut. Anda secara opsional dapat menambahkan deskripsi ke perangkat, tag, dan nama AWS IoT benda yang terkait dengan perangkat.

```
sagemaker_client.register_devices(
    DeviceFleetName="sample-fleet-name",
    Devices=[
        {
            "DeviceName": "sample-device-1",
            "IotThingName": "sample-thing-name-1",
            "Description": "Device #1"
        }
     ],
     Tags=[
        {
            "Key": "string",
            "Value" : "string"
         }
     ],
)
```

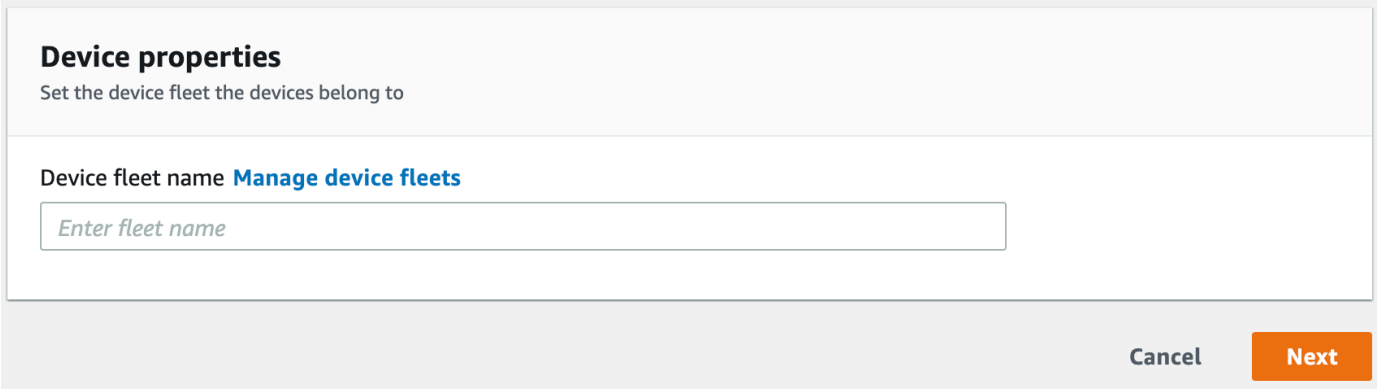
Daftarkan Perangkat (Konsol)

Anda dapat mendaftarkan perangkat Anda menggunakan SageMaker konsol di https://console.aws.amazon.com/sagemaker.

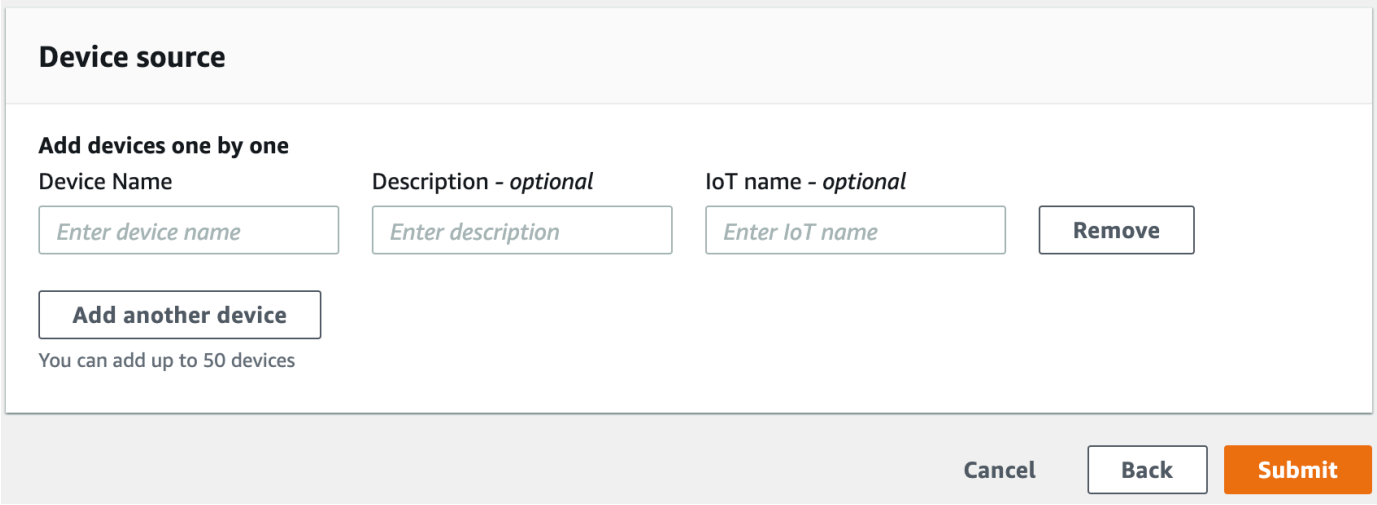
1. Di SageMaker konsol, pilih Edge Inference dan kemudian pilih pilih perangkat Edge.
2. Pilih Daftarkan perangkat.



3. Di bagian Properti perangkat, masukkan nama armada yang dimiliki perangkat di bawah bidang Nama armada perangkat. Pilih Berikutnya.



4. Di bagian Sumber perangkat, tambahkan perangkat Anda satu per satu. Anda harus menyertakan Nama Perangkat untuk setiap perangkat di armada Anda. Anda secara opsional dapat memberikan deskripsi (di bidang Deskripsi) dan nama objek Internet of Things (IoT) (di bidang nama IoT). Pilih Kirim setelah Anda menambahkan semua perangkat Anda.



Halaman Perangkat menampilkan nama perangkat yang telah Anda tambahkan, armada yang dimilikinya, kapan terdaftar, detak jantung terakhir, dan deskripsi dan AWS IoT nama, jika Anda memberikannya.

Pilih perangkat untuk melihat detail perangkat, termasuk nama perangkat, armada, ARN, deskripsi, nama IoT Thing, saat perangkat terdaftar, dan detak jantung terakhir.

Periksa Status

Periksa apakah perangkat atau armada Anda terhubung dan mengambil sampel data. Melakukan pemeriksaan berkala, secara manual atau otomatis, memungkinkan Anda memeriksa apakah perangkat atau armada Anda berfungsi dengan baik.

Gunakan konsol Amazon S3 di https://console.aws.amazon.com/s3/ untuk secara interaktif memilih armada untuk pemeriksaan status. Anda juga dapat menggunakanAWS SDK for Python (Boto3). Berikut ini menjelaskan berbagai API dari Boto3 yang dapat Anda gunakan untuk memeriksa status perangkat atau armada Anda. Gunakan API yang paling sesuai dengan kasus penggunaan Anda.

- Periksa perangkat individual.

  Untuk memeriksa status perangkat individual, gunakan `DescribeDevice` API. Daftar yang berisi satu atau beberapa model disediakan jika model telah diterapkan ke perangkat.

  ```
  sagemaker_client.describe_device(
      DeviceName="sample-device-1",
      DeviceFleetName="sample-fleet-name"
  )
  ```

  Menjalankan `DescribeDevice` pengembalian:

  ```
  { "DeviceName": "sample-device".
    "Description": "this is a sample device",
    "DeviceFleetName": "sample-device-fleet",
    "IoTThingName": "SampleThing",
    "RegistrationTime": 1600977370,
    "LatestHeartbeat": 1600977370,
    "Models":[
          {
           "ModelName": "sample-model",
           "ModelVersion": "1.1",
           "LatestSampleTime": 1600977370,
           "LatestInference": 1600977370
          }
     ]
  }
  ```

- Periksa armada perangkat.

Untuk memeriksa status armada, gunakan `GetDeviceFleetReport` API. Berikan nama armada perangkat untuk mendapatkan ringkasan armada.

```
sagemaker_client.get_device_fleet_report(
    DeviceFleetName="sample-fleet-name"
)
```

- Periksa detak jantung.

  Setiap perangkat dalam armada secara berkala menghasilkan sinyal, atau “detak jantung”. Detak jantung dapat digunakan untuk memeriksa apakah perangkat berkomunikasi dengan Edge Manager. Jika stempel waktu detak jantung terakhir tidak diperbarui, perangkat mungkin gagal.

  Periksa detak jantung terakhir dengan dibuat oleh perangkat dengan `DescribeDevice` API. Tentukan nama perangkat dan armada tempat perangkat tepi berada.

```
sagemaker_client.describe_device(
    DeviceName="sample-device-1",
    DeviceFleetName="sample-fleet-name"
)
```

Package Model

SageMaker Pekerjaan pengemasan Edge Manager mengambil model yang SageMaker dikompilasi Amazon Neo dan membuat perubahan apa pun yang diperlukan untuk menerapkan model dengan mesin inferensi, agen Edge Manager.

Topik

- Prasyarat
- Package a Model (Amazon SageMaker Console)
- Package a Model (Boto3)

Prasyarat

Untuk mengemas model, Anda harus melakukan hal berikut:

1. Kompilasi model pembelajaran mesin Anda dengan SageMaker Neo.

Jika Anda belum melakukannya, kompilasi model Anda dengan SageMaker Neo. Untuk informasi selengkapnya tentang cara mengkompilasi model Anda, lihat Mengompilasi dan Menerapkan Model dengan Neo. Jika Anda adalah pengguna pertama kali SageMaker Neo, lakukan Memulai dengan Perangkat Neo Edge.

2. Dapatkan nama pekerjaan kompilasi Anda.

   Berikan nama nama pekerjaan kompilasi yang Anda gunakan saat Anda menyusun model Anda dengan SageMaker Neo. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/ dan pilih Pekerjaan kompilasi untuk menemukan daftar kompilasi yang telah dikirimkan ke AWS akun Anda. Nama-nama pekerjaan kompilasi yang dikirimkan ada di kolom Nama.

3. Dapatkan IAM ARN Anda.

   Anda memerlukan Nama Sumber Daya Amazon (ARN) dari peran IAM yang dapat Anda gunakan untuk mengunduh dan mengunggah model dan menghubungi Neo. SageMaker

   Gunakan salah satu metode berikut untuk mendapatkan ARN IAM Anda:

   - Secara terprogram dengan Python SageMaker SDK

     ```
     import sagemaker

     # Initialize SageMaker Session object so you can interact with AWS resources
     sess = sagemaker.Session()

     # Get the role ARN
     role = sagemaker.get_execution_role()

     print(role)
     >> arn:aws:iam::<your-aws-account-id>:role/<your-role-name>
     ```

     Untuk informasi selengkapnya tentang penggunaan SageMaker Python SDK, lihat Python SageMaker SDK API.

   - Menggunakan konsol AWS Identity and Access Management (IAM)

     Arahkan ke konsol IAM di https://console.aws.amazon.com/iam/. Di bagian Sumber Daya IAM, pilih Peran untuk melihat daftar peran di AWS akun Anda. Pilih atau buat peran yang memiliki`AmazonSageMakerFullAccess`,`AWSIoTFullAccess`, dan`AmazonS3FullAccess`.

Untuk informasi lebih lanjut tentang IAM, lihat Apa itu IAM?

4. Memiliki URI bucket S3.

   Anda harus memiliki setidaknya satu URI bucket Amazon Simple Storage Service (Amazon S3) untuk menyimpan model yang dikompilasi NEO, output dari pekerjaan pengemasan Edge Manager, dan data sampel dari armada perangkat Anda.

   Gunakan salah satu metode berikut untuk membuat bucket Amazon S3:

   - Secara terprogram dengan Python SageMaker SDK

     Anda dapat menggunakan bucket Amazon S3 default selama sesi berlangsung. Bucket default dibuat berdasarkan format berikut:`sagemaker-{region}-{aws-account-id}`. Untuk membuat bucket default dengan SageMaker Python SDK, gunakan yang berikut ini:

     ```
     import sagemaker

     session=sagemaker.create_session()

     bucket=session.default_bucket()
     ```

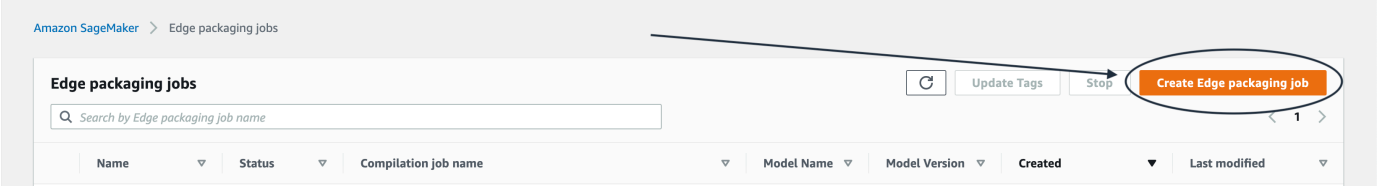
   - Menggunakan konsol Amazon S3

     Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/ dan lihat Bagaimana cara membuat Bucket S3? untuk step-by-step instruksi.

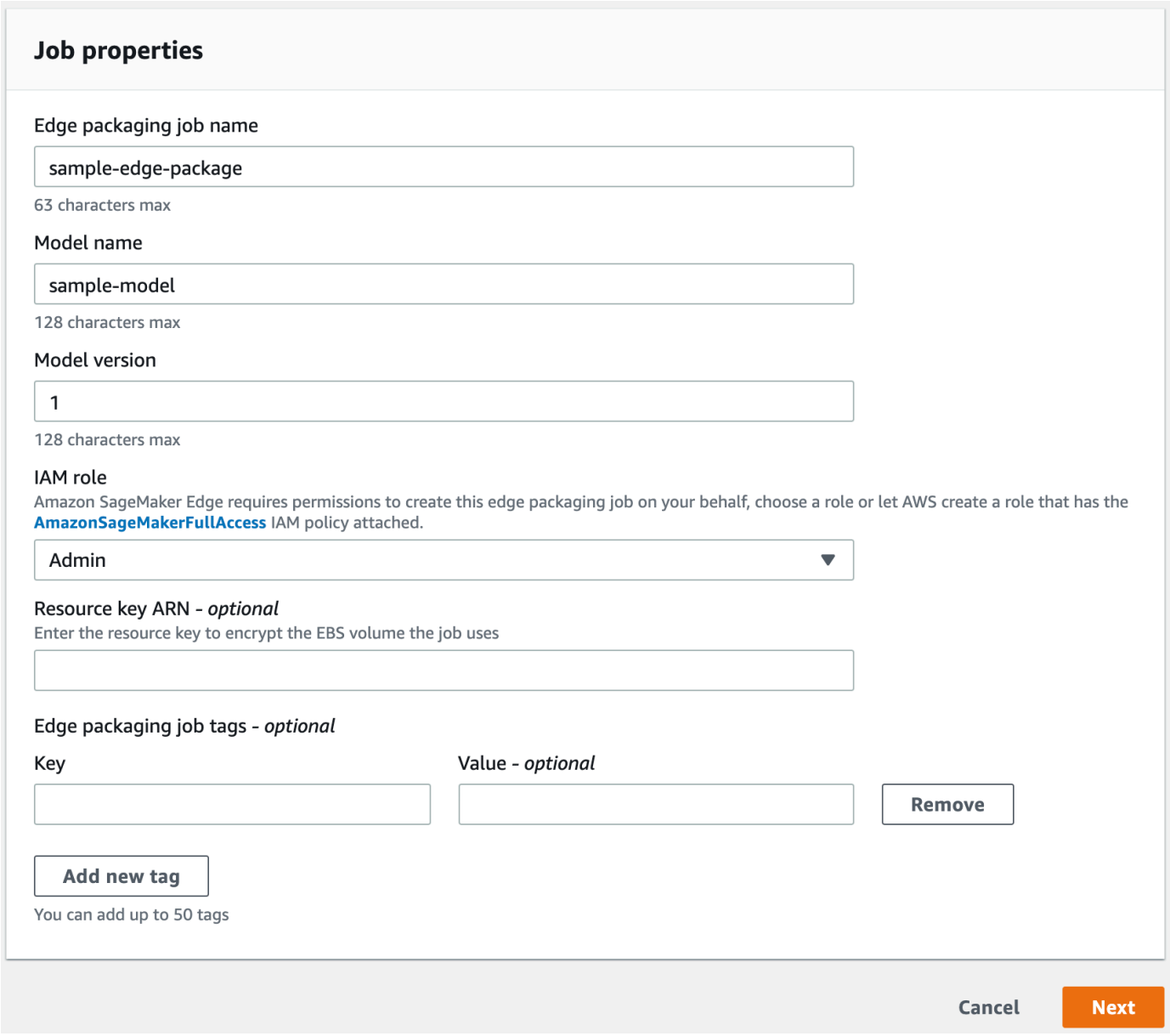
Package a Model (Amazon SageMaker Console)

Anda dapat membuat pekerjaan pengemasan SageMaker Edge Manager menggunakan SageMaker konsol di https://console.aws.amazon.com/sagemaker/. Sebelum melanjutkan, pastikan Anda sudah puasPrasyarat.

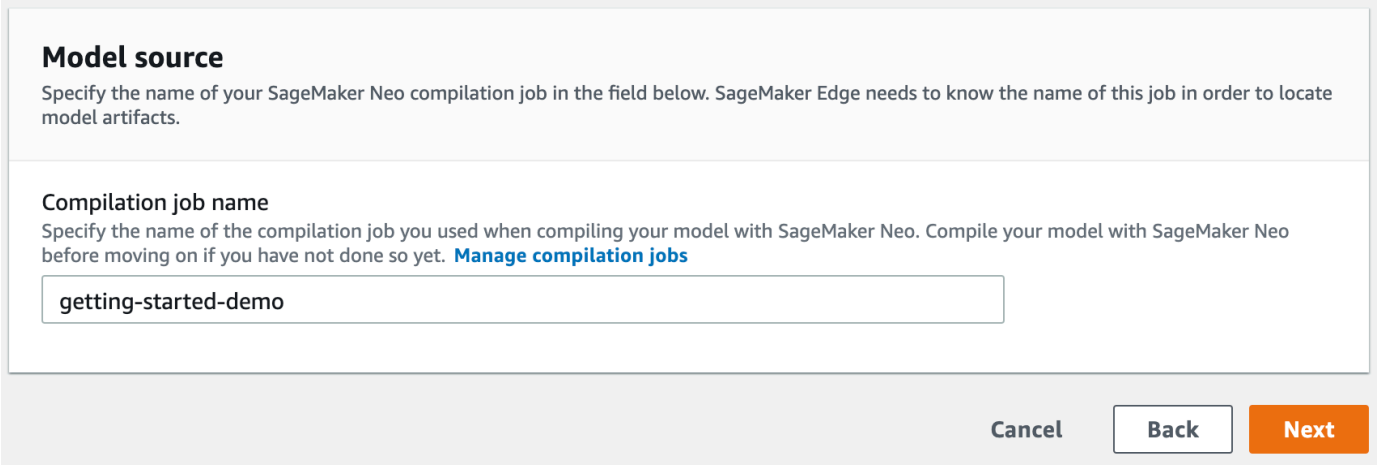
1. Di SageMaker konsol, pilih Edge Inference dan kemudian pilih pekerjaan kemasan Edge, seperti yang ditunjukkan pada gambar berikut.



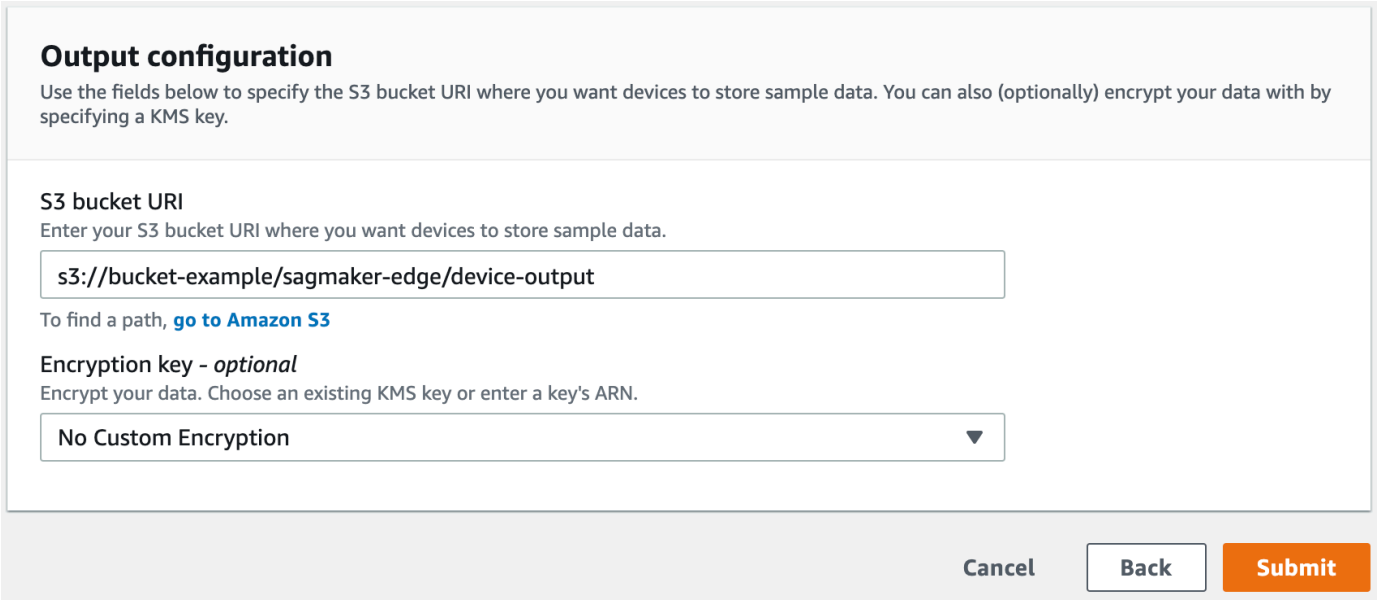
2. Pada halaman Properti Job, masukkan nama untuk pekerjaan pengemasan Anda di bawah nama pekerjaan pengemasan Edge. Perhatikan bahwa nama pekerjaan pengemasan Edge Manager peka huruf besar/kecil. Beri nama model Anda dan berikan versi: masukkan ini di bawah Nama model dan versi Model, masing-masing.
3. Selanjutnya, pilih peran IAM. Anda dapat memilih peran atau membiarkan AWS membuat peran untuk Anda. Anda dapat secara opsional menentukan ARN kunci sumber daya dan tag pekerjaan.
4. Pilih Berikutnya.



5. Tentukan nama pekerjaan kompilasi yang Anda gunakan saat mengkompilasi model Anda dengan SageMaker Neo di bidang Nama pekerjaan kompilasi. Pilih Berikutnya.



6. Pada halaman konfigurasi Output, masukkan URI bucket Amazon S3 tempat Anda ingin menyimpan output dari pekerjaan pengemasan.



Kolom Status pada halaman pekerjaan pengemasan Edge harus dibaca DALAM PROSES. Setelah pekerjaan pengemasan selesai, status diperbarui menjadi SELESAI.

Memilih pekerjaan pengemasan mengarahkan Anda ke pengaturan pekerjaan itu. Bagian Pengaturan Job menampilkan nama pekerjaan, ARN, status, waktu pembuatan, waktu modifikasi terakhir, durasi pekerjaan pengemasan, dan peran ARN.

Bagian konfigurasi Input menampilkan lokasi artefak model, konfigurasi input data, dan kerangka pembelajaran mesin model.

Bagian konfigurasi Output menampilkan lokasi output dari pekerjaan pengemasan, perangkat target yang modelnya dikompilasi, dan tag apa pun yang Anda buat.

7. Pilih nama armada perangkat Anda untuk diarahkan ke detail armada perangkat. Halaman ini menampilkan nama armada perangkat, ARN, deskripsi (jika Anda memberikannya), tanggal armada dibuat, terakhir kali armada diubah, URI bucket Amazon S3, ID kunci (jika disediakan)AWS KMS, alias (jika disediakan)AWS IoT, dan peran IAM. Jika Anda menambahkan tag, tag akan muncul di bagian Tag armada perangkat.

Package a Model (Boto3)

Anda dapat membuat pekerjaan pengemasan SageMaker Edge Manager denganAWS SDK for Python (Boto3). Sebelum melanjutkan, pastikan Anda sudah puasPrasyarat.

Untuk meminta pekerjaan pengemasan tepi, gunakan`CreateEdgePackagingJob`. Anda perlu memberikan nama untuk pekerjaan pengemasan tepi Anda, nama pekerjaan kompilasi SageMaker Neo Anda, nama sumber daya Amazon (ARN) peran Anda, nama untuk model Anda, versi untuk model Anda, dan URI bucket Amazon S3 tempat Anda ingin menyimpan output dari pekerjaan pengemasan Anda. Perhatikan bahwa nama pekerjaan pengemasan Edge Manager dan nama pekerjaan kompilasi SageMaker Neo peka huruf besar/kecil.

```
# Import AWS SDK for Python (Boto3)
import boto3

# Create Edge client so you can submit a packaging job
sagemaker_client = boto3.client("sagemaker", region_name='aws-region')

sagemaker_client.create_edge_packaging_job(
    EdgePackagingJobName="edge-packaging-name",
    CompilationJobName="neo-compilation-name",
    RoleArn="arn:aws:iam::99999999999:role/rolename",
    ModelName="sample-model-name",
    ModelVersion="model-version",
    OutputConfig={
        "S3OutputLocation": "s3://your-bucket/",
    }
)
```

Anda dapat memeriksa status pekerjaan pengemasan tepi menggunakan `DescribeEdgePackagingJob` dan memberikan nama pekerjaan pengemasan tepi peka huruf besar/kecil:

```
response = sagemaker_client.describe_edge_packaging_job(
                                       EdgePackagingJobName="edge-packaging-name")
```

Ini mengembalikan kamus yang dapat digunakan untuk polling status pekerjaan pengemasan:

```
# Optional - Poll every 30 sec to check completion status
import time

while True:
    response = sagemaker_client.describe_edge_packaging_job(
                                           EdgePackagingJobName="edge-packaging-name")

    if response['EdgePackagingJobStatus'] == 'Completed':
        break
    elif response['EdgePackagingJobStatus'] == 'Failed':
        raise RuntimeError('Packaging job failed')
    print('Packaging model...')
    time.sleep(30)
print('Done!')
```

Untuk daftar pekerjaan pengemasan, gunakan`ListEdgePackagingJobs`. Anda dapat menggunakan API ini untuk mencari pekerjaan pengemasan tertentu. Berikan sebagian nama untuk memfilter nama pekerjaan pengemasan`NameContains`, nama sebagian `ModelNameContains` untuk memfilter pekerjaan di mana nama model berisi nama yang Anda berikan. Juga tentukan dengan kolom mana yang akan diurutkan`SortBy`, dan dengan arah mana untuk mengurutkan `SortOrder` (salah satu `Ascending` atau`Descending`).

```
sagemaker_client.list_edge_packaging_jobs(
    "NameContains": "sample",
    "ModelNameContains": "sample",
    "SortBy": "column-name",
    "SortOrder": "Descending"
)
```

Untuk menghentikan pekerjaan pengemasan, gunakan `StopEdgePackagingJob` dan berikan nama pekerjaan pengemasan tepi Anda.

```
sagemaker_client.stop_edge_packaging_job(
        EdgePackagingJobName="edge-packaging-name"
)
```

Untuk daftar lengkap API Edge Manager, lihat dokumentasi Boto3.

Agen Manajer Edge

Agen Edge Manager adalah mesin inferensi untuk perangkat edge Anda. Gunakan agen untuk membuat prediksi dengan model yang dimuat ke perangkat tepi Anda. Agen juga mengumpulkan metrik model dan menangkap data pada interval tertentu. Data sampel disimpan di bucket Amazon S3 Anda.

Ada dua metode untuk menginstal dan menyebarkan agen Edge Manager ke perangkat edge Anda:

1. Unduh agen sebagai biner dari ember rilis Amazon S3. Untuk informasi selengkapnya, lihat Unduh dan Atur Agen Edge Manager Secara Manual.
2. Gunakan konsol AWS IoT Greengrass V2 atau AWS CLI untuk menyebarkan`aws.greengrass.SageMakerEdgeManager`. Lihat Buat Komponen AWS IoT Greengrass V2.

Unduh dan Atur Agen Edge Manager Secara Manual

Unduh agen Edge Manager berdasarkan sistem operasi, arsitektur, dan AWS Wilayah Anda. Agen diperbarui secara berkala, sehingga Anda memiliki opsi untuk memilih agen Anda berdasarkan tanggal dan versi rilis. Setelah Anda memiliki agen, buat file konfigurasi JSON. Tentukan nama benda IoT perangkat, nama armada, kredenal perangkat, dan pasangan nilai kunci lainnya. Lihat Menjalankan agen Edge Manager untuk lengkap daftar kunci yang harus Anda tentukan dalam file konfigurasi. Anda dapat menjalankan agen sebagai biner yang dapat dieksekusi atau menautkannya sebagai objek bersama dinamis (DSO).

Bagaimana agen bekerja

Agen berjalan pada CPU perangkat Anda. Agen menjalankan inferensi pada kerangka kerja dan perangkat keras perangkat target yang Anda tentukan selama pekerjaan kompilasi. Misalnya, jika Anda mengkompilasi model untuk Jetson Nano, agen mendukung GPU dalam Deep Learning Runtime (DLR) yang disediakan.

Agen dirilis dalam format biner untuk sistem operasi yang didukung. Periksa apakah sistem operasi Anda didukung dan memenuhi persyaratan OS minimum dalam tabel berikut:

Linux

Versi: Ubuntu 18.04

Format Biner yang Didukung: x86-64 bit (biner ELF) dan ARMv8 64 bit (biner ELF)

Windows

Versi: Windows 10 versi 1909

Format Biner yang Didukung: x86-32 bit (DLL) dan x86-64 bit (DLL)

Menginstal agen Edge Manager

Untuk menggunakan agen Edge Manager, Anda harus terlebih dahulu mendapatkan artefak rilis dan sertifikat root. Artefak rilis disimpan dalam ember Amazon S3 di `us-west-2` Wilayah. Untuk mengunduh artefak, tentukan sistem operasi Anda (`<OS>`) dan file. `<VERSION>`

Berdasarkan sistem operasi Anda, ganti `<OS>` dengan salah satu dari berikut ini:

Windows 32-bit	Windows 64-bit	Linux x86-64	Linux ARMv8
jendela-x86	jendela-x64	linux-x64	linux-armv8

`VERSION`Ini dipecah menjadi tiga komponen:`<MAJOR_VERSION>.<YYYY-MM-DD>-<SHA-7>`, di mana:

- `<MAJOR_VERSION>`: Versi rilis. Versi rilis saat ini diatur ke`1`.
- `<YYYY-MM-DD>`: Cap waktu pelepasan artefak.
- `<SHA-7>`: ID komit repositori tempat rilis dibuat.

Anda harus memberikan `<MAJOR_VERSION>` dan cap waktu dalam `YYYY-MM-DD` format. Kami menyarankan Anda menggunakan stempel waktu rilis artefak terbaru.

Jalankan yang berikut ini di baris perintah Anda untuk mendapatkan cap waktu terbaru. Ganti `<OS>` dengan sistem operasi Anda:

```
aws s3 ls s3://sagemaker-edge-release-store-us-west-2-<OS>/Releases/ | sort -r
```

Misalnya, jika Anda memiliki OS Windows 32-bit, jalankan:

```
aws s3 ls s3://sagemaker-edge-release-store-us-west-2-windows-x86/Releases/ | sort -r
```

Ini mengembalikan:

```
2020-12-01 23:33:36 0

                   PRE 1.20201218.81f481f/
                   PRE 1.20201207.02d0e97/
```

Output kembali dalam contoh ini menunjukkan dua artefak rilis. File artefak rilis pertama mencatat bahwa versi rilis memiliki versi rilis utama1, cap waktu 20201218 (dalam format YYYY-MM-DD), dan ID komit SHA-7. 81f481f

> **Note**
>
> Perintah sebelumnya mengasumsikan Anda telah mengonfigurasi file. AWS Command Line Interface Untuk informasi selengkapnya, tentang cara mengonfigurasi pengaturan yang AWS CLI digunakan untuk berinteraksiAWS, lihat Mengonfigurasi AWS CLI.

Berdasarkan sistem operasi Anda, gunakan perintah berikut untuk menginstal artefak:

Windows 32-bit

```
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-windows-x86/
Releases/<VERSION>/<VERSION>.zip .
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-windows-x86/
Releases/<VERSION>/sha256_hex.shasum .
```

Windows 64-bit

```
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-windows-x64/
Releases/<VERSION>/<VERSION>.zip .
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-windows-x64/
Releases/<VERSION>/sha256_hex.shasum .
```

Linux x86-64

```
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-linux-x64/
Releases/<VERSION>/<VERSION>.tgz .
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-linux-x64/Releases/<VERSION>/
sha256_hex.shasum .
```

Linux ARMv8

```
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-linux-armv8/
Releases/<VERSION>/<VERSION>.tgz .
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-linux-armv8/
Releases/<VERSION>/sha256_hex.shasum .
```

Anda juga harus mengunduh sertifikat root. Sertifikat ini memvalidasi artefak model yang ditandatangani AWS sebelum memuatnya ke perangkat edge Anda.

Ganti <OS> yang sesuai dengan platform Anda dari daftar sistem operasi yang didukung dan ganti <REGION> dengan AWS Wilayah Anda.

```
aws s3 cp s3://sagemaker-edge-release-store-us-west-2-<OS>/
Certificates/<REGION>/<REGION>.pem .
```

Menjalankan agen Edge Manager

Anda dapat menjalankan agen SageMaker Edge Manager sebagai proses mandiri dalam bentuk biner executable Executable and Linkable Format (ELF) yang dapat dieksekusi atau Anda dapat menautkannya sebagai objek bersama dinamis (.dll). Linux mendukung menjalankannya sebagai biner yang dapat dieksekusi mandiri dan merupakan mode yang disukai. Windows mendukung menjalankannya sebagai objek bersama (.dll).

Di Linux, kami menyarankan Anda menjalankan biner melalui layanan yang merupakan bagian dari sistem inisialisasi (`init`) Anda. Jika Anda ingin menjalankan biner secara langsung, Anda dapat melakukannya di terminal seperti yang ditunjukkan pada contoh berikut. Jika Anda memiliki OS modern, tidak ada instalasi lain yang diperlukan sebelum menjalankan agen, karena semua persyaratan dibangun secara statis ke dalam executable. Ini memberi Anda fleksibilitas untuk menjalankan agen di terminal, sebagai layanan, atau di dalam wadah.

Untuk menjalankan agen, pertama buat file konfigurasi JSON. Tentukan pasangan kunci-nilai berikut:

- `sagemaker_edge_core_device_name`: Nama perangkat. Nama perangkat ini harus didaftarkan bersama dengan armada perangkat di konsol SageMaker Edge Manager.
- `sagemaker_edge_core_device_fleet_name`: Nama armada tempat perangkat tersebut berada.
- `sagemaker_edge_core_region`: AWS Wilayah yang terkait dengan perangkat, armada, dan ember Amazon S3. Ini sesuai dengan Wilayah tempat perangkat terdaftar dan tempat bucket Amazon S3 dibuat (diharapkan sama). Model itu sendiri dapat dikompilasi dengan SageMaker Neo di Wilayah yang berbeda, konfigurasi ini tidak terkait dengan kompilasi model Region.
- `sagemaker_edge_core_root_certs_path`: Jalur folder absolut ke root certificate. Ini digunakan untuk memvalidasi perangkat dengan AWS akun yang relevan.
- `sagemaker_edge_provider_aws_ca_cert_file`: Jalur absolut ke sertifikat Amazon Root CA (AmazonRootCa1.PEM). Ini digunakan untuk memvalidasi perangkat dengan AWS akun yang relevan. `AmazonCA`adalah sertifikat yang dimiliki oleh`AWS`.
- `sagemaker_edge_provider_aws_cert_file`: Jalur absolut untuk AWS IoT menandatangani root certificate (`*.pem.crt`).
- `sagemaker_edge_provider_aws_cert_pk_file`: Jalur absolut ke kunci AWS IoT pribadi. (`*.pem.key`).
- `sagemaker_edge_provider_aws_iot_cred_endpoint`*: Titik akhir AWS IoT kredensial (identifier .iot. wilayah* .amazonaws.com). Endpoint ini digunakan untuk validasi kredensi. Lihat Menghubungkan perangkat AWS IoT untuk informasi selengkapnya.
- `sagemaker_edge_provider_provider`: Ini menunjukkan implementasi antarmuka penyedia yang digunakan. Antarmuka penyedia berkomunikasi dengan layanan jaringan akhir untuk unggahan, detak jantung, dan validasi pendaftaran. Secara default ini diatur ke`"Aws"`. Kami mengizinkan implementasi khusus dari antarmuka penyedia. Ini dapat diatur ke `None` tanpa penyedia atau `Custom` untuk implementasi khusus dengan jalur objek bersama yang relevan yang disediakan.
- `sagemaker_edge_provider_provider_path`: Menyediakan jalur absolut ke objek bersama implementasi penyedia. (.so atau berkas .dll). File .dll atau .so `"Aws"` penyedia disediakan dengan rilis agen. Bidang ini wajib.
- `sagemaker_edge_provider_s3_bucket_name`: Nama bucket Amazon S3 Anda (bukan URI bucket Amazon S3). Bucket harus memiliki `sagemaker` string dalam namanya.
- `sagemaker_edge_log_verbose`(Boolean.): Opsional. Ini menetapkan log debug. Pilih salah satu `True` atau`False`.

- `sagemaker_edge_telemetry_libsystemd_path`: Hanya untuk Linux, `systemd` mengimplementasikan metrik penghitung kerusakan agen. Setel jalur absolut libsystemd untuk mengaktifkan metrik penghitung kerusakan. Anda dapat menemukan jalur libsystemd default dapat ditemukan dengan menjalankan `whereis libsystemd` di terminal perangkat.
- `sagemaker_edge_core_capture_data_destination`: Tujuan untuk mengunggah data pengambilan. Pilih salah satu `"Cloud"` atau`"Disk"`. Default diatur ke `"Disk"`. Mengaturnya untuk `"Disk"` menulis tensor input dan output dan data tambahan ke sistem file lokal di lokasi pilihan Anda. Saat menulis untuk `"Cloud"` menggunakan nama bucket Amazon S3 yang disediakan dalam konfigurasi. `sagemaker_edge_provider_s3_bucket_name`
- `sagemaker_edge_core_capture_data_disk_path`: Atur jalur absolut dalam sistem file lokal, di mana file data pengambilan `"Disk"` ditulis kapan tujuannya. Bidang ini tidak digunakan ketika `"Cloud"` ditentukan sebagai tujuan.
- `sagemaker_edge_core_folder_prefix`: Awalan induk di Amazon S3 tempat data yang diambil disimpan saat Anda `"Cloud"` menentukan sebagai tujuan data pengambilan (. `sagemaker_edge_core_capture_data_disk_path`) Data yang diambil disimpan dalam sub-folder di bawah `sagemaker_edge_core_capture_data_disk_path` if `"Disk"` ditetapkan sebagai tujuan data.
- `sagemaker_edge_core_capture_data_buffer_size`(Nilai integer): Ukuran buffer melingkar data tangkapan. Ini menunjukkan jumlah maksimum permintaan yang disimpan dalam buffer.
- `sagemaker_edge_core_capture_data_batch_size`(Nilai integer): Ukuran batch data pengambilan. Ini menunjukkan ukuran batch permintaan yang ditangani dari buffer. Nilai ini harus kurang dari`sagemaker_edge_core_capture_data_buffer_size`. Maksimal setengah ukuran buffer direkomendasikan untuk ukuran batch.
- `sagemaker_edge_core_capture_data_push_period_seconds`(Nilai integer): Periode push data capture dalam hitungan detik. Sejumlah permintaan dalam buffer ditangani ketika ada permintaan ukuran batch di buffer, atau ketika periode waktu ini telah selesai (mana yang lebih dulu). Konfigurasi ini menetapkan periode waktu tersebut.
- `sagemaker_edge_core_capture_data_base64_embed_limit`: Batas untuk mengunggah data tangkapan dalam byte. Nilai integer.

File konfigurasi Anda akan terlihat mirip dengan contoh berikut (dengan nilai spesifik Anda ditentukan). Contoh ini menggunakan AWS penyedia default (`"Aws"`) dan tidak menentukan unggahan berkala.

```
{
```

```
    "sagemaker_edge_core_device_name": "device-name",
    "sagemaker_edge_core_device_fleet_name": "fleet-name",
    "sagemaker_edge_core_region": "region",
    "sagemaker_edge_core_root_certs_path": "<Absolute path to root certificates>",
    "sagemaker_edge_provider_provider": "Aws",
    "sagemaker_edge_provider_provider_path" : "/path/to/libprovider_aws.so",
    "sagemaker_edge_provider_aws_ca_cert_file": "<Absolute path to Amazon Root CA
 certificate>/AmazonRootCA1.pem",
    "sagemaker_edge_provider_aws_cert_file": "<Absolute path to AWS IoT signing root
 certificate>/device.pem.crt",
    "sagemaker_edge_provider_aws_cert_pk_file": "<Absolute path to AWS IoT private
 key.>/private.pem.key",
    "sagemaker_edge_provider_aws_iot_cred_endpoint": "https://<AWS IoT Endpoint
 Address>",
    "sagemaker_edge_core_capture_data_destination": "Cloud",
    "sagemaker_edge_provider_s3_bucket_name": "sagemaker-bucket-name",
    "sagemaker_edge_core_folder_prefix": "Amazon S3 folder prefix",
    "sagemaker_edge_core_capture_data_buffer_size": 30,
    "sagemaker_edge_core_capture_data_batch_size": 10,
    "sagemaker_edge_core_capture_data_push_period_seconds": 4000,
    "sagemaker_edge_core_capture_data_base64_embed_limit": 2,
    "sagemaker_edge_log_verbose": false
}
```

Artefak rilis mencakup executable biner yang disebut `sagemaker_edge_agent_binary` dalam direktori. `/bin` Untuk menjalankan biner, gunakan `-a` bendera untuk membuat deskriptor file soket (.sock) di direktori yang Anda pilih dan tentukan jalur file konfigurasi agen JSON yang Anda buat dengan bendera. `-c`

```
./sagemaker_edge_agent_binary -a <ADDRESS_TO_SOCKET> -c <PATH_TO_CONFIG_FILE>
```

Contoh berikut menunjukkan cuplikan kode dengan direktori dan jalur file yang ditentukan:

```
./sagemaker_edge_agent_binary -a /tmp/sagemaker_edge_agent_example.sock -c
 sagemaker_edge_config.json
```

Dalam contoh ini, deskriptor file soket bernama `sagemaker_edge_agent_example.sock` dibuat di `/tmp` direktori dan menunjuk ke file konfigurasi yang ada di direktori kerja yang sama dengan agen yang dipanggil`sagemaker_edge_config.json`.

Menyebarkan Model Package dan Edge Manager Agent dengan AWS IoT Greengrass

SageMaker Edge Manager mengintegrasikan AWS IoT Greengrass versi 2 untuk menyederhanakan akses, pemeliharaan, dan penerapan agen dan model Edge Manager ke perangkat Anda. Tanpa AWS IoT Greengrass V2, menyiapkan perangkat dan armada untuk menggunakan SageMaker Edge Manager mengharuskan Anda menyalin agen Edge Manager secara manual dari bucket rilis Amazon S3. Anda menggunakan agen untuk membuat prediksi dengan model yang dimuat ke perangkat tepi Anda. Dengan integrasi AWS IoT Greengrass V2 dan SageMaker Edge Manager, Anda dapat menggunakan komponen AWS IoT Greengrass V2. Komponen adalah modul perangkat lunak pra-bangun yang dapat menghubungkan perangkat edge Anda ke AWS layanan atau layanan pihak ketiga melaluiAWS IoT Greengrass.

Anda harus menginstal perangkat lunak AWS IoT Greengrass Core ke perangkat Anda jika Anda ingin menggunakan AWS IoT Greengrass V2 untuk menyebarkan agen Edge Manager dan model Anda. Untuk informasi selengkapnya tentang persyaratan perangkat dan cara mengatur perangkat, lihat Menyiapkan perangkat AWS IoT Greengrass inti dalam AWS IoT Greengrass dokumentasi.

Anda menggunakan tiga komponen berikut untuk menyebarkan agen Edge Manager:

- Komponen publik pra-bangun: SageMaker memelihara komponen Edge Manager publik.
- Komponen pribadi yang dibuat secara otomatis: Komponen pribadi dibuat secara otomatis saat Anda mengemas model pembelajaran mesin dengan `CreateEdgePackagingJob`API dan menentukan bidang `GreengrassV2Component` Edge Manager API. `PresetDeploymentType`
- Komponen khusus: Ini adalah aplikasi inferensi yang bertanggung jawab untuk memproses dan membuat kesimpulan di perangkat Anda. Anda harus membuat komponen ini. Lihat Buat komponen kustom Hello World di dokumentasi SageMaker Edge Manager atau Buat AWS IoT Greengrass komponen kustom dalam AWS IoT Greengrass dokumentasi untuk informasi selengkapnya tentang cara membuat komponen kustom.

Prasyarat

SageMaker Edge Manager menggunakan AWS IoT Greengrass V2 untuk menyederhanakan penerapan agen Edge Manager, model pembelajaran mesin Anda, dan aplikasi inferensi Anda ke perangkat Anda dengan menggunakan komponen. Untuk mempermudah mempertahankan peran AWS IAM Anda, Edge Manager memungkinkan Anda untuk menggunakan kembali alias AWS IoT peran yang ada. Jika Anda belum memilikinya, Edge Manager menghasilkan alias peran sebagai bagian dari pekerjaan pengemasan Edge Manager. Anda tidak perlu lagi mengaitkan alias peran

yang dihasilkan dari pekerjaan pengemasan SageMaker Edge Manager dengan AWS IoT peran Anda.

Sebelum Anda mulai, Anda harus menyelesaikan prasyarat berikut:

1. Instal perangkat lunak AWS IoT Greengrass Core. Untuk informasi lebih lanjut, lihat Menginstal perangkat lunak AWS IoT Greengrass Inti.
2. Mengatur AWS IoT Greengrass V2. Untuk informasi selengkapnya, lihat Menginstal perangkat lunak AWS IoT Greengrass Inti dengan penyediaan sumber daya manual.

   > **Note**
   >
   > - Pastikan nama AWS IoT benda semuanya huruf kecil dan tidak mengandung karakter kecuali (opsional) tanda hubung (). -
   > - Peran IAM harus dimulai dengan `SageMaker*`

3. Lampirkan izin dan kebijakan sebaris berikut ke peran IAM yang dibuat selama penyiapan AWS IoT Greengrass V2.
   - Arahkan ke konsol IAM https://console.aws.amazon.com/iam/.
   - Cari peran yang Anda buat dengan mengetikkan nama peran di kolom Pencarian.
   - Pilih peran Anda.
   - Selanjutnya, pilih Lampirkan kebijakan.
   - Cari AmazonSageMakerEdgeDeviceFleetPolicy.
   - Pilih AmazonSageMakerFullAccess(Ini adalah langkah opsional yang memudahkan Anda untuk menggunakan kembali peran IAM ini dalam kompilasi dan pengemasan model).
   - Tambahkan izin yang diperlukan ke kebijakan izin peran, jangan lampirkan kebijakan sebaris ke pengguna IAM.

```
{
    "Version":"2012-10-17",
    "Statement":[
      {
        "Sid":"GreengrassComponentAccess",
        "Effect":"Allow",
        "Action":[
            "greengrass:CreateComponentVersion",
            "greengrass:DescribeComponent"
        ],
```

```
        "Resource":"*"
       }
    ]
}
```

- Pilih Lampirkan kebijakan.
- Pilih Hubungan kepercayaan.
- Pilih Edit trust relationship (Edit Hubungan Kepercayaan).
- Ganti konten dengan yang berikut ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "credentials.iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

4. Buat armada perangkat Edge Manager. Untuk informasi tentang cara membuat armada, lihatMengatur Perangkat dan Armada.
5. Daftarkan perangkat Anda dengan nama yang sama dengan nama AWS IoT benda Anda yang dibuat selama pengaturan AWS IoT Greengrass V2.
6. Buat setidaknya satu AWS IoT Greengrass komponen pribadi khusus. Komponen ini adalah aplikasi yang menjalankan inferensi pada perangkat. Lihat informasi yang lebih lengkap di Buat komponen kustom Hello World

> **Note**
>
> - SageMaker Edge Manager dan AWS IoT Greengrass integrasi hanya berfungsi untuk AWS IoT Greengrass v2.
> - Baik nama AWS IoT benda Anda dan nama perangkat Edge Manager harus sama.
> - SageMaker Edge Manager tidak memuat AWS IoT sertifikat lokal dan memanggil titik akhir penyedia AWS IoT kredensi secara langsung. Sebagai gantinya, SageMaker Edge Manager menggunakan AWS IoT Greengrass v2 TokenExchangeService dan mengambil kredensi sementara dari titik akhir TES.

Buat Komponen AWS IoT Greengrass V2

AWS IoT Greengrassmenggunakan komponen, modul perangkat lunak yang digunakan dan berjalan pada perangkat AWS IoT Greengrass inti. Anda membutuhkan (minimal) tiga komponen:

1. AWS IoT GreengrassKomponen Edge Manager Agent publik yang menyebarkan agentbinary Edge Manager.
2. Komponen model yang dibuat secara otomatis saat Anda mengemas model pembelajaran mesin dengan AWS SDK for Python (Boto3) API atau dengan SageMaker konsol. Untuk informasi, lihat Buat komponen yang dibuat secara otomatis.
3. Komponen pribadi dan kustom untuk mengimplementasikan aplikasi klien agen Edge Manager, dan melakukan pra-pemrosesan dan pasca-pemrosesan hasil inferensi. Untuk informasi selengkapnya tentang cara membuat komponen kustom, lihat Buat komponen yang dibuat secara otomatis atau Membuat AWS IoT Greengrass komponen kustom.

Buat komponen yang dibuat secara otomatis

Buat komponen model dengan `CreateEdgePackagingJob`API dan tentukan `GreengrassV2Component` bidang API pekerjaan pengemasan SageMaker Edge Manager`PresetDeploymentType`. Saat Anda memanggil `CreateEdgePackagingJob` API, Edge Manager mengambil model yang SageMaker dikompilasi Neo Anda di Amazon S3 dan membuat komponen model. Komponen model secara otomatis disimpan di akun Anda. Anda dapat melihat salah satu komponen Anda dengan menavigasi ke AWS IoT konsol https://console.aws.amazon.com/iot/. Pilih Greengrass lalu pilih Perangkat inti. Halaman ini memiliki daftar perangkat AWS IoT Greengrass inti yang terkait dengan akun Anda. Jika nama komponen model

tidak ditentukan dalam`PresetDeploymentConfig`, nama default yang dihasilkan terdiri dari `"SagemakerEdgeManager"` dan nama pekerjaan pengemasan agen Edge Manager Anda. Contoh berikut menunjukkan cara menentukan ke Edge Manager untuk membuat komponen AWS IoT Greengrass V2 dengan `CreateEdgePackagingJob` API.

```
import sagemaker
import boto3

# Create a SageMaker client object to make it easier to interact with other AWS
 services.
sagemaker_client = boto3.client('sagemaker', region=<YOUR_REGION>)

# Replace with your IAM Role ARN
sagemaker_role_arn = "arn:aws:iam::<account>:role/*"

# Replace string with the name of your already created S3 bucket.
bucket = 'edge-manager-demo-bucket'

# Specify a name for your edge packaging job.
edge_packaging_name = "edge_packag_job_demo"

# Replace the following string with the name you used for the SageMaker Neo compilation
 job.
compilation_job_name = "getting-started-demo"

# The name of the model and the model version.
model_name = "sample-model"
model_version = "1.1"

# Output directory in S3 where you want to store the packaged model.
packaging_output_dir = 'packaged_models'
packaging_s3_output = 's3://{}/{}'.format(bucket, packaging_output_dir)

# The name you want your Greengrass component to have.
component_name = "SagemakerEdgeManager" + edge_packaging_name

sagemaker_client.create_edge_packaging_job(
                    EdgePackagingJobName=edge_packaging_name,
                    CompilationJobName=compilation_job_name,
                    RoleArn=sagemaker_role_arn,
                    ModelName=model_name,
                    ModelVersion=model_version,
                    OutputConfig={
```

```
                        "S3OutputLocation": packaging_s3_output,
                        "PresetDeploymentType":"GreengrassV2Component",
                        "PresetDeploymentConfig":"{\"ComponentName\":\"sample-
component-name\", \"ComponentVersion\":\"1.0.2\"}"
                        }
                    )
```

Anda juga dapat membuat komponen yang dibuat secara otomatis dengan SageMaker konsol. Ikuti langkah 1-6 di Package a Model (Amazon SageMaker Console)

Masukkan URI bucket Amazon S3 tempat Anda ingin menyimpan output pekerjaan pengemasan dan kunci enkripsi opsional.

Lengkapi yang berikut ini untuk membuat komponen model:

1. Pilih Penerapan prasetel.
2. Tentukan nama komponen untuk bidang Nama komponen.
3. Secara opsional, berikan deskripsi komponen, versi komponen, OS platform, atau arsitektur platform untuk deskripsi Komponen, versi Komponen, OS Platform, dan arsitektur Platform.
4. Pilih Kirim.

Buat komponen kustom Hello World

Komponen aplikasi khusus digunakan untuk melakukan inferensi pada perangkat tepi. Komponen bertanggung jawab untuk memuat model ke SageMaker Edge Manager, memanggil agen Edge Manager untuk inferensi, dan membongkar model saat komponen dimatikan. Sebelum Anda membuat komponen Anda, pastikan agen dan aplikasi dapat berkomunikasi dengan Edge Manager. Untuk melakukan ini, konfigurasikan gRPC. Agen Edge Manager menggunakan metode yang didefinisikan dalam Protobuf Buffers dan server gRPC untuk menjalin komunikasi dengan aplikasi klien di perangkat edge dan cloud.

Untuk menggunakan gRPC, Anda harus:

1. Buat rintisan gRPC menggunakan file.proto yang disediakan saat Anda mengunduh agen Edge Manager dari bucket rilis Amazon S3.
2. Tulis kode klien dengan bahasa yang Anda inginkan.

Anda tidak perlu mendefinisikan layanan dalam file.proto. File.proto layanan disertakan dalam file TAR terkompresi saat Anda mengunduh biner rilis agen Edge Manager dari bucket rilis Amazon S3.

Instal gRPC dan alat lain yang diperlukan di mesin host Anda dan buat `agent_pb2_grpc.py` rintisan gRPC dan dengan Python. `agent_pb2.py` Pastikan Anda memiliki `agent.proto` direktori lokal Anda.

```
%%bash
pip install grpcio
pip install grpcio-tools
python3 -m grpc_tools.protoc --proto_path=. --python_out=. --grpc_python_out=.
 agent.proto
```

Kode sebelumnya menghasilkan klien gRPC dan antarmuka server dari definisi layanan.proto Anda. Dengan kata lain, itu menciptakan model gRPC dengan Python. Direktori API berisi spesifikasi Protobuf untuk berkomunikasi dengan agen.

Selanjutnya, gunakan gRPC API untuk menulis klien dan server untuk layanan Anda (2). Contoh skrip berikut,`edge_manager_python_example.py`, menggunakan Python untuk memuat, daftar, dan membongkar `yolov3` model ke perangkat edge.

```
import grpc
from PIL import Image
import agent_pb2
import agent_pb2_grpc
import os


model_path = '<PATH-TO-SagemakerEdgeManager-COMPONENT>'

agent_socket = 'unix:///tmp/aws.greengrass.SageMakerEdgeManager.sock'

agent_channel = grpc.insecure_channel(agent_socket, options=(('grpc.enable_http_proxy',
 0),))

agent_client = agent_pb2_grpc.AgentStub(agent_channel)


def list_models():
    return agent_client.ListModels(agent_pb2.ListModelsRequest())


def list_model_tensors(models):
    return {
        model.name: {
```

```
            'inputs': model.input_tensor_metadatas,
            'outputs': model.output_tensor_metadatas
        }
        for model in list_models().models
    }


def load_model(model_name, model_path):
    load_request = agent_pb2.LoadModelRequest()
    load_request.url = model_path
    load_request.name = model_name
    return agent_client.LoadModel(load_request)


def unload_model(name):
    unload_request = agent_pb2.UnLoadModelRequest()
    unload_request.name = name
    return agent_client.UnLoadModel(unload_request)


def predict_image(model_name, image_path):
    image_tensor = agent_pb2.Tensor()
    image_tensor.byte_data = Image.open(image_path).tobytes()
    image_tensor_metadata = list_model_tensors(list_models())[model_name]['inputs'][0]
    image_tensor.tensor_metadata.name = image_tensor_metadata.name
    image_tensor.tensor_metadata.data_type = image_tensor_metadata.data_type
    for shape in image_tensor_metadata.shape:
        image_tensor.tensor_metadata.shape.append(shape)
    predict_request = agent_pb2.PredictRequest()
    predict_request.name = model_name
    predict_request.tensors.append(image_tensor)
    predict_response = agent_client.Predict(predict_request)
    return predict_response

def main():
    try:
        unload_model('your-model')
    except:
        pass

    print('LoadModel...', end='')
    try:
        load_model('your-model', model_path)
        print('done.')
```

```
    except Exception as e:
        print()
        print(e)
        print('Model already loaded!')

    print('ListModels...', end='')
    try:
        print(list_models())
        print('done.')

    except Exception as e:
        print()
        print(e)
        print('List model failed!')

    print('Unload model...', end='')
    try:
        unload_model('your-model')
        print('done.')
    except Exception as e:
        print()
        print(e)
        print('unload model failed!')

if __name__ == '__main__':
    main()
```

Pastikan `model_path` menunjuk ke nama AWS IoT Greengrass komponen yang berisi model jika Anda menggunakan contoh kode klien yang sama.

Anda dapat membuat komponen Hello World AWS IoT Greengrass V2 Anda setelah Anda membuat stub gRPC Anda dan Anda memiliki kode Hello World Anda siap. Untuk melakukannya:

- Unggah bucket Anda `edge_manager_python_example.pyagent_pb2_grpc.py`,, dan `agent_pb2.py` ke Amazon S3 Anda dan catat jalur Amazon S3 mereka.
- Buat komponen pribadi di konsol AWS IoT Greengrass V2 dan tentukan resep untuk komponen Anda. Tentukan URI Amazon S3 ke aplikasi Hello World dan rintisan gRPC Anda dalam resep berikut.

```
---
RecipeFormatVersion: 2020-01-25
ComponentName: com.sagemaker.edgePythonExample
```

```
ComponentVersion: 1.0.0
ComponentDescription: Sagemaker Edge Manager Python example
ComponentPublisher: Amazon Web Services, Inc.
ComponentDependencies:
  aws.greengrass.SageMakerEdgeManager:
    VersionRequirement: '>=1.0.0'
    DependencyType: HARD
Manifests:
  - Platform:
      os: linux
      architecture: "/amd64|x86/"
    Lifecycle:
      install: |-
        apt-get install python3-pip
        pip3 install grpcio
        pip3 install grpcio-tools
        pip3 install protobuf
        pip3 install Pillow
      run:
        script: |-
          python3 {artifacts:path}/edge_manager_python_example.py
    Artifacts:
      - URI: <code-s3-path>
      - URI: <pb2-s3-path>
      - URI: <pb2-grpc-s3-path>
```

Untuk informasi rinci tentang membuat resep Hello World, lihat Membuat komponen pertama Anda dalam AWS IoT Greengrass dokumentasi.

Menerapkan komponen ke perangkat Anda

Terapkan komponen Anda dengan AWS IoT konsol atau dengan. AWS CLI

Untuk men-deploy komponen Anda (konsol)

Terapkan AWS IoT Greengrass komponen Anda dengan AWS IoT konsol.

1. Di AWS IoT Greengrass konsol di menu navigasi https://console.aws.amazon.com/iot/, pilih Deployment.
2. Pada halaman Komponen, pada tab Komponen publik, pilih `aws.greengrass.SageMakerEdgeManager`.
3. Pada halaman `aws.greengrass.SageMakerEdgeManager` pilih Deploy.

4. Dari`Add to deployment`, pilih salah satu dari berikut ini:
   a. Untuk menggabungkan komponen ini ke deployment yang ada pada perangkat target Anda, pilih Tambahkan ke deployment yang ada, lalu pilih deployment yang ingin Anda revisi.
   b. Untuk membuat deployment baru di perangkat target Anda, pilih Buat deployment baru. Jika Anda memiliki deployment yang ada di perangkat, dengan memilih langkah ini Anda akan menggantikan deployment yang ada.
5. Di halaman Tentukan target, lakukan hal berikut:
   a. Di bawah informasi Deployment, masukkan atau ubah nama yang ramah untuk deployment Anda.
   b. Di bawah Target deployment, pilih target untuk deployment Anda, dan pilih Selanjutnya. Anda tidak dapat mengubah target deployment jika Anda merevisi deployment yang ada.
6. Pada halaman Pilih komponen, di bawah Komponen saya, pilih:
   - com. <CUSTOM-COMPONENT-NAME>
   - `aws.greengrass.SageMakerEdgeManager`
   - SagemakerEdgeManager. <YOUR-PACKAGING-JOB>
7. Pada halaman Configure components, pilih com.greengrass. SageMakerEdgeManager, dan lakukan hal berikut.
   a. Pilih Konfigurasi komponen.
   b. Di bawah Pembaruan konfigurasi, di Konfigurasi untuk menggabungkan, masukkan konfigurasi berikut.

      ```
      {
          "DeviceFleetName": "device-fleet-name",
          "BucketName": "DOC-EXAMPLE-BUCKET"
      }
      ```

      Ganti *`device-fleet-name`*dengan nama armada perangkat edge yang Anda buat, dan ganti *`DOC-EXAMPLE-BUCKET`*dengan nama bucket Amazon S3 yang terkait dengan armada perangkat Anda.
   c. Pilih Konfirmasi dan kemudian pilih Selanjutnya.
8. Pada halaman Konfigurasikan pengaturan lanjutan, simpan pengaturan konfigurasi default tersebut, dan pilih Selanjutnya.
9. Di halaman Tinjau, pilih Deploy.

Untuk men-deploy komponen Anda (AWS CLI)

1. Buat `deployment.json` file untuk menentukan konfigurasi penerapan untuk komponen SageMaker Edge Manager Anda. File ini akan terlihat seperti contoh berikut.

```
{
  "targetArn":"targetArn",
  "components": {
    "aws.greengrass.SageMakerEdgeManager": {
      "componentVersion": 1.0.0,
      "configurationUpdate": {
        "merge": {
          "DeviceFleetName": "device-fleet-name",
          "BucketName": "DOC-EXAMPLE-BUCKET"
        }
      }
    },
    "com.greengrass.SageMakerEdgeManager.ImageClassification": {
      "componentVersion": 1.0.0,
      "configurationUpdate": {
      }
    },
    "com.greengrass.SageMakerEdgeManager.ImageClassification.Model": {
      "componentVersion": 1.0.0,
      "configurationUpdate": {
      }
    },
  }
}
```

   - Di kolom `targetArn`, ganti *`targetArn`* dengan Amazon Resource Name (ARN) dari grup objek atau objek yang ditargetkan untuk deployment tersebut, dalam format berikut:
     - Objek: `arn:aws:iot:`*`region`*`:`*`account-id`*`:thing/`*`thingName`*
     - Grup objek: `arn:aws:iot:`*`region`*`:`*`account-id:`*`thinggroup/`*`thingGroupName`*
   - Di `merge` bidang, ganti *`device-fleet-name`*dengan nama armada perangkat edge yang Anda buat, dan ganti *`DOC-EXAMPLE-BUCKET`*dengan nama bucket Amazon S3 yang terkait dengan armada perangkat Anda.
   - Ganti versi komponen untuk setiap komponen dengan versi terbaru yang tersedia.
2. Jalankan perintah berikut untuk men-deploy komponen pada perangkat:

```
aws greengrassv2 create-deployment \
    --cli-input-json file://path/to/deployment.json
```

Deployment ini dapat memakan waktu beberapa menit hingga selesai. Pada langkah berikutnya, periksa log komponen untuk memverifikasi bahwa deployment tersebut berhasil diselesaikan dan untuk melihat hasil inferensi.

Untuk informasi selengkapnya tentang penerapan komponen ke perangkat individual atau grup perangkat, lihat Menerapkan AWS IoT Greengrass komponen ke perangkat.

Menerapkan Paket Model Langsung dengan SageMaker Edge Manager Deployment API

SageMaker Edge Manager menyediakan API penerapan yang dapat Anda gunakan untuk menerapkan model ke target perangkat tanpa. AWS IoT Greengrass Ini berguna dalam situasi di mana Anda ingin memperbarui model secara independen dari pembaruan firmware atau mekanisme penyebaran aplikasi. Anda dapat menggunakan API untuk mengintegrasikan penerapan edge Anda ke dalam alur kerja CI/CD untuk menerapkan model secara otomatis setelah Anda memvalidasi model Anda untuk akurasi. API juga memiliki opsi rollback dan peluncuran bertahap yang nyaman bagi Anda untuk memastikan model bekerja dengan baik di lingkungan tertentu sebelum peluncuran yang lebih luas.

Untuk menggunakan API penerapan Edge Manager, pertama-tama kompilasi dan paket model Anda. Untuk informasi tentang cara mengkompilasi dan mengemas model Anda, lihatLatih, Kompilasi, dan Package Model Anda. Bagian berikut dari panduan ini menunjukkan bagaimana Anda dapat membuat penerapan tepi menggunakan SageMaker API, setelah Anda mengkompilasi dan mengemas model Anda.

Topik

- Buat rencana penyebaran tepi
- Mulai penyebaran tepi
- Periksa status penyebaran

Buat rencana penyebaran tepi

Anda dapat membuat rencana penerapan tepi dengan CreateEdgeDeploymentPlanAPI. Rencana penyebaran dapat memiliki beberapa tahap. Anda dapat mengonfigurasi setiap tahap untuk meluncurkan penerapan ke subset perangkat edge (berdasarkan persentase, atau berdasarkan nama perangkat). Anda juga dapat mengonfigurasi bagaimana kegagalan peluncuran ditangani pada setiap tahap.

Cuplikan kode berikut menunjukkan bagaimana Anda dapat membuat rencana penyebaran tepi dengan 1 tahap untuk menerapkan model yang dikompilasi dan paket ke 2 perangkat tepi tertentu:

```
import boto3

client = boto3.client("sagemaker")

client.create_edge_deployment_plan(
    EdgeDeploymentPlanName="edge-deployment-plan-name",
    DeviceFleetName="device-fleet-name",
    ModelConfigs=[
        {
            "EdgePackagingJobName": "edge-packaging-job-name",
            "ModelHandle": "model-handle"
        }
    ],
    Stages=[
        {
            "StageName": "stage-name",
            "DeviceSelectionConfig": {
                "DeviceSubsetType": "SELECTION",
                "DeviceNames": ["device-name-1", "device-name-2"]
            },
            "DeploymentConfig": {
                "FailureHandlingPolicy": "ROLLBACK_ON_FAILURE"
            }
        }
    ]
)
```

Alih-alih perangkat tertentu, jika Anda ingin menyebarkan ke model ke persentase perangkat di armada Anda, maka tetapkan nilai `DeviceSubsetType` to `"PERCENTAGE"` dan ganti

`"DeviceNames": ["device-name-1", "device-name-2"]` dengan contoh `"Percentage": desired-percentage` di atas.

Tahapan dapat ditambahkan setelah rencana penerapan dibuat dengan CreateEdgeDeploymentStageAPI, jika Anda ingin mulai meluncurkan tahapan baru setelah memvalidasi keberhasilan peluncuran pengujian Anda. Untuk informasi selengkapnya tentang tahapan penerapan, lihatDeploymentStage.

Mulai penyebaran tepi

Setelah membuat rencana penerapan dan tahapan penerapan, Anda dapat memulai penerapan dengan API. `StartEdgeDeploymentStage`

```
client.start_edge_deployment_stage(
    EdgeDeploymentPlanName="edge-deployment-plan-name",
    StageName="stage-name"
)
```

Periksa status penyebaran

Anda dapat memeriksa status penerapan tepi dengan DescribeEdgeDeploymentPlanAPI.

```
client.describe_edge_deployment_plan(
    EdgeDeploymentPlanName="edge-deployment-plan-name"
)
```

Kelola Model

Agen Edge Manager dapat memuat beberapa model sekaligus dan membuat kesimpulan dengan model yang dimuat pada perangkat edge. Jumlah model yang dapat dimuat agen ditentukan oleh memori yang tersedia pada perangkat. Agen memvalidasi tanda tangan model dan memuat ke dalam memori semua artefak yang dihasilkan oleh pekerjaan pengemasan tepi. Langkah ini mengharuskan semua sertifikat yang diperlukan yang dijelaskan dalam langkah-langkah sebelumnya untuk diinstal bersama dengan sisa instalasi biner. Jika tanda tangan model tidak dapat divalidasi, maka pemuatan model gagal dengan kode pengembalian dan alasan yang sesuai.

SageMaker Agen Edge Manager menyediakan daftar API Manajemen Model yang mengimplementasikan bidang kontrol dan API bidang data pada perangkat edge. Seiring dengan dokumentasi ini, kami merekomendasikan untuk melalui implementasi klien sampel yang menunjukkan penggunaan kanonik dari API yang dijelaskan di bawah ini.

`proto`File tersedia sebagai bagian dari artefak rilis (di dalam tarball rilis). Dalam dokumen ini, kami mencantumkan dan menjelaskan penggunaan API yang tercantum dalam `proto` file ini.

> **Note**
>
> Ada one-to-one pemetaan untuk API ini pada rilis Windows dan kode contoh untuk aplikasi yang diterapkan di C # dibagikan dengan artefak rilis untuk Windows. Petunjuk di bawah ini adalah untuk menjalankan Agen sebagai proses mandiri, berlaku untuk artefak rilis untuk Linux.

Ekstrak arsip berdasarkan OS Anda. `VERSION`Dimana dipecah menjadi tiga komponen:`<MAJOR_VERSION>.<YYYY-MM-DD>-<SHA-7>`. Lihat Menginstal agen Edge Manager untuk informasi tentang cara mendapatkan versi rilis (`<MAJOR_VERSION>`), cap waktu artefak rilis (`<YYYY-MM-DD>`), dan ID komit repositori () `SHA-7`

Linux

Arsip zip dapat diekstraksi dengan perintah:

```
tar -xvzf <VERSION>.tgz
```

Windows

Arsip zip dapat diekstraksi dengan UI atau perintah:

```
unzip <VERSION>.tgz
```

Hirarki artefak rilis (setelah mengekstrak `tar/zip` arsip) ditunjukkan di bawah ini. `proto`File agen tersedia di bawah`api/`.

```
0.20201205.7ee4b0b
### bin
```

```
#         ### sagemaker_edge_agent_binary
#         ### sagemaker_edge_agent_client_example
### docs
### api
#         ### agent.proto
### attributions
#         ### agent.txt
#         ### core.txt
### examples
### ipc_example
### CMakeLists.txt
### sagemaker_edge_client.cc
### sagemaker_edge_client_example.cc
### sagemaker_edge_client.hh
### sagemaker_edge.proto
### README.md
### shm.cc
### shm.hh
### street_small.bmp
```

Topik

Model Beban

Agen Edge Manager mendukung pemuatan beberapa model. API ini memvalidasi tanda tangan model dan memuat ke dalam memori semua artefak yang dihasilkan oleh operasi. `EdgePackagingJob` Langkah ini mengharuskan semua sertifikat yang diperlukan untuk diinstal bersama dengan instalasi biner agen lainnya. Jika tanda tangan model tidak dapat divalidasi maka langkah ini gagal dengan kode pengembalian yang sesuai dan pesan kesalahan di log.

```
// perform load for a model
```

```
// Note:
// 1. currently only local filesystem paths are supported for loading models.
// 2. multiple models can be loaded at the same time, as limited by available device
 memory
// 3. users are required to unload any loaded model to load another model.
// Status Codes:
// 1. OK - load is successful
// 2. UNKNOWN - unknown error has occurred
// 3. INTERNAL - an internal error has occurred
// 4. NOT_FOUND - model doesn't exist at the url
// 5. ALREADY_EXISTS - model with the same name is already loaded
// 6. RESOURCE_EXHAUSTED - memory is not available to load the model
// 7. FAILED_PRECONDITION - model is not compiled for the machine.
//
rpc LoadModel(LoadModelRequest) returns (LoadModelResponse);
```

Input

```
//
// request for LoadModel rpc call
//
message LoadModelRequest {
  string url = 1;
  string name = 2;  // Model name needs to match regex "^[a-zA-Z0-9](-*[a-zA-Z0-9])*
$"
}
```

Output

```
//
//
// response for LoadModel rpc call
//
message LoadModelResponse {
  Model model = 1;
}

//
// Model represents the metadata of a model
//  url - url representing the path of the model
//  name - name of model
//  input_tensor_metadatas - TensorMetadata array for the input tensors
//  output_tensor_metadatas - TensorMetadata array for the output tensors
```

```
//
// Note:
//  1. input and output tensor metadata could empty for dynamic models.
//
message Model {
  string url = 1;
  string name = 2;
  repeated TensorMetadata input_tensor_metadatas = 3;
  repeated TensorMetadata output_tensor_metadatas = 4;
}
```

Model Bongkar

Membongkar model yang dimuat sebelumnya. Hal ini diidentifikasi melalui alias model yang disediakan selama`loadModel`. Jika alias tidak ditemukan atau model tidak dimuat maka mengembalikan kesalahan.

```
//
// perform unload for a model
// Status Codes:
// 1. OK - unload is successful
// 2. UNKNOWN - unknown error has occurred
// 3. INTERNAL - an internal error has occurred
// 4. NOT_FOUND - model doesn't exist
//
rpc UnLoadModel(UnLoadModelRequest) returns (UnLoadModelResponse);
```

Input

```
//
// request for UnLoadModel rpc call
//
message UnLoadModelRequest {
 string name = 1; // Model name needs to match regex "^[a-zA-Z0-9](-*[a-zA-Z0-9])*$"
}
```

Output

```
//
// response for UnLoadModel rpc call
//
```

```
message UnLoadModelResponse {}
```

Daftar Model

Daftar semua model yang dimuat dan aliasnya.

```
//
// lists the loaded models
// Status Codes:
// 1. OK - unload is successful
// 2. UNKNOWN - unknown error has occurred
// 3. INTERNAL - an internal error has occurred
//
rpc ListModels(ListModelsRequest) returns (ListModelsResponse);
```

Input

```
//
// request for ListModels rpc call
//
message ListModelsRequest {}
```

Output

```
//
// response for ListModels rpc call
//
message ListModelsResponse {
 repeated Model models = 1;
}
```

Jelaskan Model

Menjelaskan model yang dimuat pada agen.

```
//
// Status Codes:
// 1. OK - load is successful
// 2. UNKNOWN - unknown error has occurred
// 3. INTERNAL - an internal error has occurred
```

```
// 4. NOT_FOUND - model doesn't exist at the url
//
rpc DescribeModel(DescribeModelRequest) returns (DescribeModelResponse);
```

Input

```
//
// request for DescribeModel rpc call
//
message DescribeModelRequest {
  string name = 1;
}
```

Output

```
//
// response for DescribeModel rpc call
//
message DescribeModelResponse {
  Model model = 1;
}
```

Tangkap Data

Memungkinkan aplikasi klien menangkap tensor input dan output di bucket Amazon S3, dan opsional tambahan. Aplikasi klien diharapkan untuk meneruskan ID tangkapan unik bersama dengan setiap panggilan ke API ini. Ini nantinya dapat digunakan untuk menanyakan status tangkapan.

```
//
// allows users to capture input and output tensors along with auxiliary data.
// Status Codes:
// 1. OK - data capture successfully initiated
// 2. UNKNOWN - unknown error has occurred
// 3. INTERNAL - an internal error has occurred
// 5. ALREADY_EXISTS - capture initiated for the given capture_id
// 6. RESOURCE_EXHAUSTED - buffer is full cannot accept any more requests.
// 7. OUT_OF_RANGE - timestamp is in the future.
// 8. INVALID_ARGUMENT - capture_id is not of expected format.
//
rpc CaptureData(CaptureDataRequest) returns (CaptureDataResponse);
```

Input

```
enum Encoding {
 CSV = 0;
 JSON = 1;
 NONE = 2;
 BASE64 = 3;
}

//
// AuxilaryData represents a payload of extra data to be capture along with inputs
 and outputs of inference
// encoding - supports the encoding of the data
// data - represents the data of shared memory, this could be passed in two ways:
// a. send across the raw bytes of the multi-dimensional tensor array
// b. send a SharedMemoryHandle which contains the posix shared memory segment id
 and
// offset in bytes to location of multi-dimensional tensor array.
//
message AuxilaryData {
 string name = 1;
 Encoding encoding = 2;
 oneof data {
 bytes byte_data = 3;
 SharedMemoryHandle shared_memory_handle = 4;
 }
}

//
// Tensor represents a tensor, encoded as contiguous multi-dimensional array.
// tensor_metadata - represents metadata of the shared memory segment
// data_or_handle - represents the data of shared memory, this could be passed in
 two ways:
// a. send across the raw bytes of the multi-dimensional tensor array
// b. send a SharedMemoryHandle which contains the posix shared memory segment
// id and offset in bytes to location of multi-dimensional tensor array.
//
message Tensor {
 TensorMetadata tensor_metadata = 1; //optional in the predict request
 oneof data {
 bytes byte_data = 4;
 // will only be used for input tensors
 SharedMemoryHandle shared_memory_handle = 5;
 }
```

```
}

//
// request for CaptureData rpc call
//
message CaptureDataRequest {
 string model_name = 1;
 string capture_id = 2; //uuid string
 Timestamp inference_timestamp = 3;
 repeated Tensor input_tensors = 4;
 repeated Tensor output_tensors = 5;
 repeated AuxilaryData inputs = 6;
 repeated AuxilaryData outputs = 7;
}
```

Output

```
//
// response for CaptureData rpc call
//
message CaptureDataResponse {}
```

Dapatkan Status Capture

Tergantung pada model dimuat input dan output tensor bisa besar (untuk banyak perangkat tepi). Menangkap ke cloud bisa memakan waktu. Jadi `CaptureData()` diimplementasikan sebagai operasi asinkron. ID tangkapan adalah pengidentifikasi unik yang disediakan klien selama pengambilan panggilan data, ID ini dapat digunakan untuk menanyakan status panggilan asinkron.

```
//
// allows users to query status of capture data operation
// Status Codes:
// 1. OK - data capture successfully initiated
// 2. UNKNOWN - unknown error has occurred
// 3. INTERNAL - an internal error has occurred
// 4. NOT_FOUND - given capture id doesn't exist.
//
rpc GetCaptureDataStatus(GetCaptureDataStatusRequest) returns
 (GetCaptureDataStatusResponse);
```

Input

```
//
// request for GetCaptureDataStatus rpc call
//
message GetCaptureDataStatusRequest {
  string capture_id = 1;
}
```

Output

```
enum CaptureDataStatus {
  FAILURE = 0;
  SUCCESS = 1;
  IN_PROGRESS = 2;
  NOT_FOUND = 3;
}

//
// response for GetCaptureDataStatus rpc call
//
message GetCaptureDataStatusResponse {
  CaptureDataStatus status = 1;
}
```

Memprediksi

`predict`API melakukan inferensi pada model yang dimuat sebelumnya. Ini menerima permintaan dalam bentuk tensor yang langsung dimasukkan ke dalam jaringan saraf. Outputnya adalah tensor keluaran (atau skalar) dari model. Ini adalah panggilan blokir.

```
//
// perform inference on a model.
//
// Note:
// 1. users can chose to send the tensor data in the protobuf message or
// through a shared memory segment on a per tensor basis, the Predict
// method with handle the decode transparently.
// 2. serializing large tensors into the protobuf message can be quite expensive,
// based on our measurements it is recommended to use shared memory of
// tenors larger than 256KB.
```

```
// 3. SMEdge IPC server will not use shared memory for returning output tensors,
// i.e., the output tensor data will always send in byte form encoded
// in the tensors of PredictResponse.
// 4. currently SMEdge IPC server cannot handle concurrent predict calls, all
// these call will be serialized under the hood. this shall be addressed
// in a later release.
// Status Codes:
// 1. OK - prediction is successful
// 2. UNKNOWN - unknown error has occurred
// 3. INTERNAL - an internal error has occurred
// 4. NOT_FOUND - when model not found
// 5. INVALID_ARGUMENT - when tenors types mismatch
//
rpc Predict(PredictRequest) returns (PredictResponse);
```

Input

```
// request for Predict rpc call
//
message PredictRequest {
string name = 1;
repeated Tensor tensors = 2;
}

//
// Tensor represents a tensor, encoded as contiguous multi-dimensional array.
//    tensor_metadata - represents metadata of the shared memory segment
//    data_or_handle - represents the data of shared memory, this could be passed in
 two ways:
//                        a. send across the raw bytes of the multi-dimensional
 tensor array
//                        b. send a SharedMemoryHandle which contains the posix
 shared memory segment
//                            id and offset in bytes to location of multi-
dimensional tensor array.
//
message Tensor {
  TensorMetadata tensor_metadata = 1; //optional in the predict request
  oneof data {
    bytes byte_data = 4;
    // will only be used for input tensors
    SharedMemoryHandle shared_memory_handle = 5;
  }
```

```
}

//
// Tensor represents a tensor, encoded as contiguous multi-dimensional array.
//    tensor_metadata - represents metadata of the shared memory segment
//    data_or_handle - represents the data of shared memory, this could be passed in
 two ways:
//                        a. send across the raw bytes of the multi-dimensional
 tensor array
//                        b. send a SharedMemoryHandle which contains the posix
 shared memory segment
//                            id and offset in bytes to location of multi-
dimensional tensor array.
//
message Tensor {
  TensorMetadata tensor_metadata = 1; //optional in the predict request
  oneof data {
    bytes byte_data = 4;
    // will only be used for input tensors
    SharedMemoryHandle shared_memory_handle = 5;
  }
}

//
// TensorMetadata represents the metadata for a tensor
//    name - name of the tensor
//    data_type  - data type of the tensor
//    shape - array of dimensions of the tensor
//
message TensorMetadata {
  string name = 1;
  DataType data_type = 2;
  repeated int32 shape = 3;
}

//
// SharedMemoryHandle represents a posix shared memory segment
//    offset - offset in bytes from the start of the shared memory segment.
//    segment_id - shared memory segment id corresponding to the posix shared memory
 segment.
//    size - size in bytes of shared memory segment to use from the offset position.
//
message SharedMemoryHandle {
  uint64 size = 1;
```

```
  uint64 offset = 2;
  uint64 segment_id = 3;
}
```

Output

> **Note**
>
> `PredictResponse`Satu-satunya yang kembali `Tensors` dan tidak`SharedMemoryHandle`.

```
// response for Predict rpc call
//
message PredictResponse {
   repeated Tensor tensors = 1;
}
```

SageMaker Edge Manager akhir hidup

Mulai 26 April 2024, Anda tidak dapat lagi mengakses Amazon SageMaker Edge Manager melalui konsol AWS manajemen, membuat pekerjaan pengemasan tepi, dan mengelola armada perangkat edge.

FAQ

Gunakan bagian berikut untuk mendapatkan jawaban atas pertanyaan umum tentang SageMaker Edge Manager end of life (EOL).

T: Apa yang terjadi pada Amazon SageMaker Edge Manager saya setelah tanggal EOL?

J: Setelah 26 April 2024, semua referensi ke pekerjaan pengemasan tepi, perangkat, dan armada perangkat dihapus dari layanan Edge Manager. Anda tidak dapat lagi menemukan atau mengakses layanan Edge Manager dari AWS konsol Anda dan aplikasi yang memanggil API layanan Edge Manager tidak lagi berfungsi.

T: Apakah saya akan ditagih untuk sumber daya Edge Manager yang tersisa di akun saya setelah tanggal EOL?

J: Sumber daya yang dibuat oleh Edge Manager, seperti paket edge di dalam bucket Amazon S3, AWS IoT things, dan peran AWS IAM, terus ada di layanan masing-masing setelah 26 April 2024. Agar tidak ditagih setelah Edge Manager tidak lagi didukung, hapus sumber daya Anda. Untuk informasi selengkapnya tentang menghapus sumber daya Anda, lihatHapus sumber daya Edge Manager.

T: Bagaimana cara menghapus sumber daya Amazon SageMaker Edge Manager saya?

J: Sumber daya yang dibuat oleh Edge Manager, seperti paket edge di dalam bucket Amazon S3, AWS IoT things, dan peran AWS IAM, terus ada di layanan masing-masing setelah 26 April 2024. Agar tidak ditagih setelah Edge Manager tidak lagi didukung, hapus sumber daya Anda. Untuk informasi selengkapnya tentang menghapus sumber daya Anda, lihatHapus sumber daya Edge Manager.

T: Bagaimana saya bisa terus menerapkan model di edge?

A: Kami sarankan Anda mencoba salah satu alat pembelajaran mesin berikut. Untuk runtime edge lintas platform, gunakan ONNX. ONNX adalah solusi open-source yang populer dan terpelihara dengan baik yang menerjemahkan model Anda ke dalam instruksi yang dapat dijalankan oleh banyak jenis perangkat keras, dan kompatibel dengan kerangka kerja HTML terbaru. ONNX dapat diintegrasikan ke dalam SageMaker alur kerja Anda sebagai langkah otomatis untuk penerapan edge Anda.

Untuk penggunaan penggunaan edge dan pemantauan. AWS IoT Greengrass V2 AWS IoT Greengrass V2memiliki mekanisme pengemasan dan penyebaran yang dapat diperluas yang dapat menyesuaikan model dan aplikasi di tepi. Anda dapat menggunakan saluran MQTT bawaan untuk mengirim kembali telemetri model ke SageMaker Amazon Model Monitor atau menggunakan sistem izin bawaan untuk mengirim data yang diambil dari model kembali ke Amazon Simple Storage Service (Amazon S3). Jika Anda tidak atau tidak dapat menggunakanAWS IoT Greengrass V2, kami sarankan menggunakan MQTT dan IoT Jobs (pustaka C/C++) untuk membuat mekanisme OTA ringan untuk menghadirkan model.

Kami telah menyiapkan kode sampel yang tersedia di GitHub repositori ini untuk membantu Anda beralih ke alat yang disarankan ini.

Hapus sumber daya Edge Manager

Sumber daya yang dibuat oleh Edge Manager terus ada setelah 26 April 2024. Untuk menghindari penagihan, hapus sumber daya ini.

Untuk menghapus AWS IoT Greengrass sumber daya, lakukan hal berikut:

1. Di AWS IoT Core konsol, pilih Perangkat Greengrass di bawah Kelola.
2. Pilih Komponen.
3. Di bawah komponen Saya, komponen yang dibuat Edge Manager ada dalam format SageMakerEdge (EdgePackagingJobName). Pilih komponen yang ingin Anda hapus.
4. Kemudian pilih Hapus versi.

Untuk menghapus alias AWS IoT peran, lakukan hal berikut:

1. Di AWS IoT Core konsol, pilih Keamanan di bawah Kelola.
2. Pilih alias Peran.
3. Alias peran yang dibuat Edge Manager ada dalam format SageMakerEdge- {DeviceFleetName}. Pilih peran yang ingin Anda hapus.
4. Pilih Hapus.

Untuk menghapus pekerjaan pengemasan di bucket Amazon S3, lakukan hal berikut:

1. Di SageMaker konsol, pilih Edge Inference.
2. Pilih pekerjaan pengemasan Edge.
3. Pilih salah satu pekerjaan pengemasan tepi. Salin URI Amazon S3 di bawah Artefak model di bagian konfigurasi Output.
4. Di konsol Amazon S3, navigasikan ke lokasi yang sesuai, dan periksa apakah Anda perlu menghapus artefak model. Untuk menghapus artefak model, pilih objek Amazon S3 dan pilih Hapus.

Optimalkan kinerja model menggunakan Neo

Memulai dengan SageMaker yang memungkinkan model pembelajaran mesin untuk berlatih sekali dan berjalan di mana saja di cloud dan di tepi.

Jika Anda adalah pengguna pertama kali SageMaker Neo, kami sarankan Anda memeriksaMemulai denganbagian untuk mendapatkan step-by-step petunjuk tentang cara mengkompilasi dan menyebarkan ke perangkat edge.

Memulai dengan SageMaker Neo?

Umumnya, mengoptimalkan model pembelajaran mesin untuk inferensi pada berbagai platform sulit karena Anda perlu menyesuaikan model untuk konfigurasi perangkat keras dan perangkat lunak tertentu dari setiap platform. Jika Anda ingin mendapatkan kinerja optimal untuk beban kerja tertentu, Anda perlu mengetahui arsitektur perangkat keras, set instruksi, pola akses memori, dan bentuk data input, di antara faktor-faktor lainnya. Untuk pengembangan perangkat lunak tradisional, alat seperti kompiler dan profiler menyederhanakan proses. Untuk pembelajaran mesin, sebagian besar alat khusus untuk kerangka kerja atau perangkat keras. Ini memaksa Anda menjadi manual trial-and-error proses yang tidak dapat diandalkan dan tidak produktif.

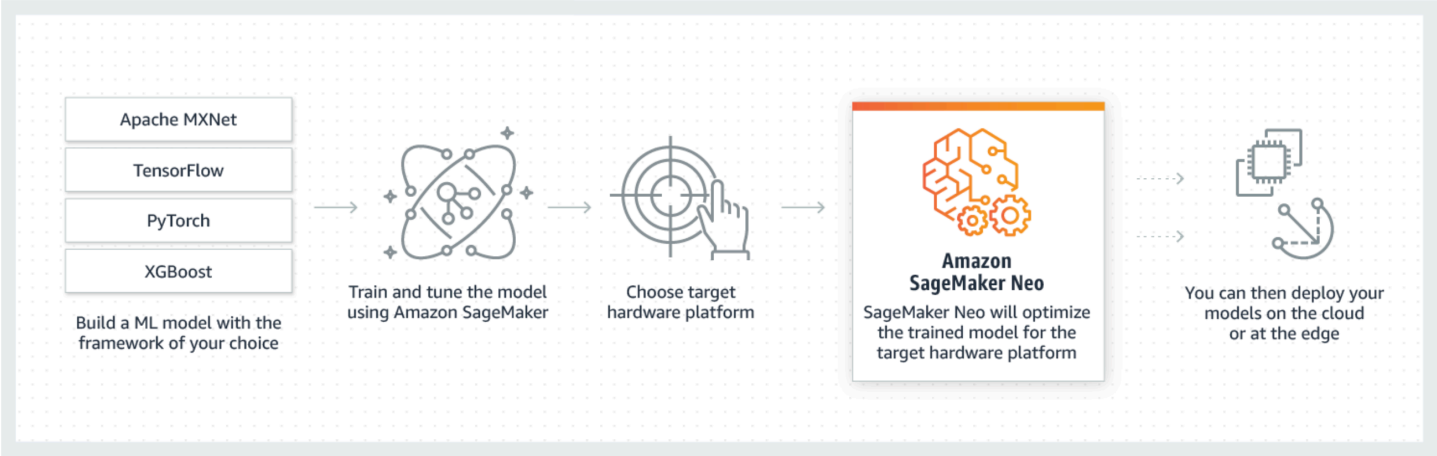
Neo secara otomatis mengoptimalkan Gluon, Keras, MXNet, PyTorch, TensorFlow, TensorFlowModel -Lite, dan ONNX untuk inferensi pada mesin Android, Linux, dan Windows berdasarkan prosesor dari Ambarella, ARM, Intel, Nvidia, NXP, Qualcomm, Texas Instruments, dan Xilinx. Neo diuji dengan model visi komputer yang tersedia di kebun binatang model di seluruh kerangka kerja. SageMaker Neo mendukung kompilasi dan penyebaran untuk dua platform utama: instance cloud (termasuk Inferentia) dan perangkat edge.

Untuk informasi selengkapnya tentang kerangka kerja yang didukung dan jenis instans cloud yang dapat Anda gunakan, lihatJenis dan Kerangka Instance yang Didukunguntuk contoh cloud.

Untuk informasi lebih lanjut tentang kerangka kerja yang didukung, perangkat edge, sistem operasi, arsitektur chip, dan model pembelajaran mesin umum yang diuji oleh SageMaker Neo untuk perangkat tepi, lihatKerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukunguntuk perangkat tepi.

Cara Kerjanya

Neo terdiri dari compiler dan runtime. Pertama, API kompilasi Neo membaca model yang diekspor dari berbagai kerangka kerja. Ini mengubah fungsi dan operasi khusus kerangka kerja menjadi representasi perantara kerangka kerja agnostik. Selanjutnya, ia melakukan serangkaian pengoptimalan. Kemudian menghasilkan kode biner untuk operasi yang dioptimalkan, menulisnya ke perpustakaan objek bersama, dan menyimpan definisi model dan parameter ke dalam file terpisah. Neo juga menyediakan runtime untuk setiap platform target yang memuat dan mengeksekusi model yang dikompilasi.



Anda dapat membuat pekerjaan kompilasi Neo dari salah satu SageMaker konsolAWS Command Line Interface(AWS CLI), notebook Python, atau SageMaker sdk.Untuk informasi tentang cara mengkompilasi model, lihatGunakan Neo untuk Mengompilasi Model. Dengan beberapa perintah CLI, pemanggilan API, atau beberapa klik, Anda dapat mengonversi model untuk platform pilihan Anda. Anda dapat menerapkan model ke SageMaker titik akhir atau padaAWS IoT Greengrassperangkat dengan cepat.

Neo dapat mengoptimalkan model dengan parameter baik dalam FP32 atau dikuantisasi ke INT8 atau FP16 bit-width.

Topik

- Gunakan Neo untuk Mengompilasi Model
- Instans Cloud
- Perangkat Edge
- Memecahkan Masalah Kesalahan

Gunakan Neo untuk Mengompilasi Model

Bagian ini menunjukkan cara membuat, mendeskripsikan, menghentikan, dan membuat daftar pekerjaan kompilasi. Opsi berikut tersedia di Amazon SageMaker Neo untuk mengelola pekerjaan kompilasi untuk model pembelajaran mesin: SageMaker konsol Amazon, atau Amazon SageMaker SDK. AWS Command Line Interface

Topik

- Mempersiapkan Model untuk Kompilasi
- Kompilasi Model () AWS Command Line Interface

- Kompilasi Model (SageMaker Konsol Amazon)
- Kompilasi Model (Amazon SageMaker SDK)

Mempersiapkan Model untuk Kompilasi

SageMaker Neo membutuhkan model pembelajaran mesin untuk memenuhi bentuk data input tertentu. Bentuk input yang diperlukan untuk kompilasi tergantung pada kerangka pembelajaran mendalam yang Anda gunakan. Setelah bentuk input model Anda diformat dengan benar, simpan model Anda sesuai dengan persyaratan di bawah ini. Setelah Anda memiliki model yang disimpan, kompres artefak model.

Topik

- Bentuk data input apa yang diharapkan SageMaker Neo?
- Menyimpan Model untuk SageMaker Neo

Bentuk data input apa yang diharapkan SageMaker Neo?

Sebelum Anda mengkompilasi model Anda, pastikan model Anda diformat dengan benar. Neo mengharapkan nama dan bentuk input data yang diharapkan untuk model terlatih Anda dengan format JSON atau format daftar. Input yang diharapkan adalah kerangka kerja khusus.

Di bawah ini adalah bentuk input yang diharapkan SageMaker Neo:

Keras

Tentukan nama dan bentuk (format NCHW) dari input data yang diharapkan menggunakan format kamus untuk model terlatih Anda. Perhatikan bahwa sementara artefak model Keras harus diunggah dalam format NHWC (channel-last), DataInputConfig harus ditentukan dalam format NCHW (channel-first). Format kamus yang diperlukan adalah sebagai berikut:

- Untuk satu masukan: `{'input_1':[1,3,224,224]}`
- Untuk dua input: `{'input_1': [1,3,224,224], 'input_2':[1,3,224,224]}`

MXNET/ONNX

Tentukan nama dan bentuk (format NCHW) dari input data yang diharapkan menggunakan format kamus untuk model terlatih Anda. Format kamus yang diperlukan adalah sebagai berikut:

- Untuk satu masukan: `{'data':[1,3,1024,1024]}`
- Untuk dua input: `{'var1': [1,1,28,28], 'var2':[1,1,28,28]}`

PyTorch

Untuk PyTorch model, Anda tidak perlu memberikan nama dan bentuk input data yang diharapkan jika Anda memenuhi kedua kondisi berikut:

- Anda membuat file definisi model Anda dengan menggunakan PyTorch 2.0 atau yang lebih baru. Untuk informasi selengkapnya tentang cara membuat file definisi, lihat PyTorch bagian di bawah Menyimpan Model untuk SageMaker Neo.
- Anda sedang mengkompilasi model Anda untuk instance cloud. Untuk informasi selengkapnya tentang jenis instance yang didukung SageMaker Neo, lihatJenis dan Kerangka Instance yang Didukung.

Jika Anda memenuhi kondisi ini, SageMaker Neo mendapatkan konfigurasi input dari file definisi model (.pt atau.pth) yang Anda buat dengan. PyTorch

Jika tidak, Anda harus melakukan hal berikut:

Tentukan nama dan bentuk (format NCHW) dari input data yang diharapkan menggunakan format kamus untuk model terlatih Anda. Atau, Anda dapat menentukan bentuk hanya menggunakan format daftar. Format kamus yang diperlukan adalah sebagai berikut:

- Untuk satu masukan dalam format kamus: `{'input0':[1,3,224,224]}`
- Untuk satu masukan dalam format daftar: `[[1,3,224,224]]`
- Untuk dua input dalam format kamus: `{'input0':[1,3,224,224], 'input1': [1,3,224,224]}`
- Untuk dua input dalam format daftar: `[[1,3,224,224], [1,3,224,224]]`

TensorFlow

Tentukan nama dan bentuk (format NHWC) dari input data yang diharapkan menggunakan format kamus untuk model terlatih Anda. Format kamus yang diperlukan adalah sebagai berikut:

- Untuk satu masukan: `{'input':[1,1024,1024,3]}`
- Untuk dua input: `{'data1': [1,28,28,1], 'data2':[1,28,28,1]}`

TFLite

Tentukan nama dan bentuk (format NHWC) dari input data yang diharapkan menggunakan format kamus untuk model terlatih Anda. Format kamus yang diperlukan adalah sebagai berikut:

- Untuk satu masukan: `{'input':[1,224,224,3]}`

> **Note**
>
> SageMaker Neo hanya mendukung TensorFlow Lite untuk target perangkat edge. Untuk daftar target perangkat SageMaker Neo edge yang didukung, lihat Perangkat halaman SageMaker Neo. Untuk daftar target instans SageMaker Neo cloud yang didukung, lihat Jenis dan Kerangka Instance yang Didukung halaman SageMaker Neo.

XGBoost

Nama dan bentuk data input tidak diperlukan.

Menyimpan Model untuk SageMaker Neo

Contoh kode berikut menunjukkan cara menyimpan model Anda agar kompatibel dengan Neo. Model harus dikemas sebagai file tar terkompresi ()`*.tar.gz`.

Keras

Model Keras memerlukan satu file definisi model (`.h5`).

Ada dua opsi untuk menyimpan model Keras Anda agar kompatibel untuk SageMaker Neo:

1. Ekspor ke `.h5` format dengan`model.save("<model-name>", save_format="h5")`.
2. Bekukan `SavedModel` setelah mengekspor.

Di bawah ini adalah contoh cara mengekspor `tf.keras` model sebagai grafik beku (opsi dua):

```
import os
import tensorflow as tf
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras import backend

tf.keras.backend.set_learning_phase(0)
```

```
model = tf.keras.applications.ResNet50(weights='imagenet', include_top=False,
 input_shape=(224, 224, 3), pooling='avg')
model.summary()

# Save as a SavedModel
export_dir = 'saved_model/'
model.save(export_dir, save_format='tf')

# Freeze saved model
input_node_names = [inp.name.split(":")[0] for inp in model.inputs]
output_node_names = [output.name.split(":")[0] for output in model.outputs]
print("Input names: ", input_node_names)
with tf.Session() as sess:
    loaded = tf.saved_model.load(sess, export_dir=export_dir, tags=["serve"])
    frozen_graph = tf.graph_util.convert_variables_to_constants(sess,

 sess.graph.as_graph_def(),
                                                                 output_node_names)
    tf.io.write_graph(graph_or_graph_def=frozen_graph, logdir=".",
 name="frozen_graph.pb", as_text=False)

import tarfile
tar = tarfile.open("frozen_graph.tar.gz", "w:gz")
tar.add("frozen_graph.pb")
tar.close()
```

Warning

Jangan mengekspor model Anda dengan `SavedModel` kelas menggunakan`model.save(<path>, save_format='tf')`. Format ini cocok untuk pelatihan, tetapi tidak cocok untuk inferensi.

MXNet

Model MXNet harus disimpan sebagai `*-symbol.json` file simbol tunggal dan satu parameter. `*.params files`

Gluon Models

Tentukan jaringan saraf menggunakan `HybridSequential` Kelas. Ini akan menjalankan kode dalam gaya pemrograman simbolik (sebagai lawan dari pemrograman imperatif).

```
from mxnet import nd, sym
from mxnet.gluon import nn

def get_net():
    net = nn.HybridSequential()  # Here we use the class HybridSequential.
    net.add(nn.Dense(256, activation='relu'),
            nn.Dense(128, activation='relu'),
            nn.Dense(2))
    net.initialize()
    return net

# Define an input to compute a forward calculation.
x = nd.random.normal(shape=(1, 512))
net = get_net()

# During the forward calculation, the neural network will automatically infer
# the shape of the weight parameters of all the layers based on the shape of
# the input.
net(x)

# hybridize model
net.hybridize()
net(x)

# export model
net.export('<model_name>') # this will create model-symbol.json and
 model-0000.params files

import tarfile
tar = tarfile.open("<model_name>.tar.gz", "w:gz")
for name in ["<model_name>-0000.params", "<model_name>-symbol.json"]:
    tar.add(name)
tar.close()
```

Untuk informasi selengkapnya tentang model hibridisasi, lihat dokumentasi hibridisasi MXNet.

Gluon Model Zoo (GluonCV)

Model kebun binatang model GluonCV datang pra-hibridisasi. Jadi Anda bisa mengekspornya.

```
import numpy as np
import mxnet as mx
import gluoncv as gcv
from gluoncv.utils import export_block
```

```
import tarfile

net = gcv.model_zoo.get_model('<model_name>', pretrained=True) # For example, choose
 <model_name> as resnet18_v1
export_block('<model_name>', net, preprocess=True, layout='HWC')

tar = tarfile.open("<model_name>.tar.gz", "w:gz")

for name in ["<model_name>-0000.params", "<model_name>-symbol.json"]:
    tar.add(name)
tar.close()
```

Non Gluon Models

Semua model non-Gluon saat disimpan ke penggunaan disk `*-symbol` dan `*.params` file. Karena itu mereka sudah dalam format yang benar untuk Neo.

```
# Pass the following 3 parameters: sym, args, aux
mx.model.save_checkpoint('<model_name>',0,sym,args,aux) # this will create
 <model_name>-symbol.json and <model_name>-0000.params files

import tarfile
tar = tarfile.open("<model_name>.tar.gz", "w:gz")

for name in ["<model_name>-0000.params", "<model_name>-symbol.json"]:
    tar.add(name)
tar.close()
```

PyTorch

PyTorch model harus disimpan sebagai file definisi (`.pt`atau`.pth`) dengan tipe data input dari. `float32`

Untuk menyimpan model Anda, gunakan `torch.jit.trace` metode yang diikuti oleh `torch.save` metode. Proses ini menyimpan objek ke file disk dan secara default menggunakan python pickle (`pickle_module=pickle`) untuk menyimpan objek dan beberapa metadata. Selanjutnya, konversi model yang disimpan ke file tar terkompresi.

```
import torchvision
import torch
```

```
model = torchvision.models.resnet18(pretrained=True)
model.eval()
inp = torch.rand(1, 3, 224, 224)
model_trace = torch.jit.trace(model, inp)

# Save your model. The following code saves it with the .pth file extension
model_trace.save('model.pth')

# Save as a compressed tar file
import tarfile
with tarfile.open('model.tar.gz', 'w:gz') as f:
    f.add('model.pth')
f.close()
```

Jika Anda menyimpan model Anda dengan PyTorch 2.0 atau yang lebih baru, SageMaker Neo memperoleh konfigurasi input untuk model (nama dan bentuk untuk inputnya) dari file definisi. Dalam hal ini, Anda tidak perlu menentukan konfigurasi input data SageMaker saat Anda mengkompilasi model.

Jika Anda ingin mencegah SageMaker Neo menurunkan konfigurasi input, Anda dapat mengatur `_store_inputs` parameter `torch.jit.trace` ke`False`. Jika Anda melakukan ini, Anda harus menentukan konfigurasi input data SageMaker saat Anda mengkompilasi model.

Untuk informasi selengkapnya tentang `torch.jit.trace` metode ini, lihat TORCH.JIT.TRACE di dokumentasi. PyTorch

TensorFlow

TensorFlow membutuhkan satu `.pb` atau satu `.pbtxt` file dan direktori variabel yang berisi variabel. Untuk model beku, hanya satu `.pb` atau `.pbtxt` file yang diperlukan.

Contoh kode berikut menunjukkan cara menggunakan perintah tar Linux untuk mengompres model Anda. Jalankan yang berikut ini di terminal Anda atau di notebook Jupyter (jika Anda menggunakan notebook Jupyter, masukkan perintah `!` ajaib di awal pernyataan):

```
# Download SSD_Mobilenet trained model
!wget http://download.tensorflow.org/models/object_detection/
ssd_mobilenet_v2_coco_2018_03_29.tar.gz

# unzip the compressed tar file
!tar xvf ssd_mobilenet_v2_coco_2018_03_29.tar.gz
```

```
# Compress the tar file and save it in a directory called 'model.tar.gz'
!tar czvf model.tar.gz ssd_mobilenet_v2_coco_2018_03_29/frozen_inference_graph.pb
```

Bendera perintah yang digunakan dalam contoh ini mencapai hal berikut:

- `c`: Buat arsip
- `z`: Kompres arsip dengan gzip
- `v`: Tampilkan kemajuan arsip
- `f`: Tentukan nama file arsip

Estimator Bawaan

Estimator bawaan dibuat oleh wadah khusus kerangka kerja atau wadah khusus algoritme. Objek penaksir untuk algoritme bawaan dan estimator khusus kerangka kerja menyimpan model dalam format yang benar untuk Anda saat Anda melatih model menggunakan metode bawaan. `.fit`

Misalnya, Anda dapat menggunakan a `sagemaker.TensorFlow` untuk mendefinisikan TensorFlow estimator:

```
from sagemaker.tensorflow import TensorFlow

estimator = TensorFlow(entry_point='mnist.py',
                       role=role,  #param role can be arn of a sagemaker execution
 role
                       framework_version='1.15.3',
                       py_version='py3',
                       training_steps=1000,
                       evaluation_steps=100,
                       instance_count=2,
                       instance_type='ml.c4.xlarge')
```

Kemudian latih model dengan metode `.fit` bawaan:

```
estimator.fit(inputs)
```

Sebelum akhirnya mengkompilasi model dengan `compile_model` metode build in:

```
# Specify output path of the compiled model
output_path = '/'.join(estimator.output_path.split('/')[:-1])
```

```
# Compile model
optimized_estimator = estimator.compile_model(target_instance_family='ml_c5',
                            input_shape={'data':[1, 784]},  # Batch size 1, 3
 channels, 224x224 Images.
                            output_path=output_path,
                            framework='tensorflow', framework_version='1.15.3')
```

Anda juga dapat menggunakan `sagemaker.estimator.Estimator` Class untuk menginisialisasi objek estimator untuk pelatihan dan mengkompilasi algoritma bawaan dengan `compile_model` metode dari Python SageMaker SDK:

```
import sagemaker
from sagemaker.image_uris import retrieve
sagemaker_session = sagemaker.Session()
aws_region = sagemaker_session.boto_region_name

# Specify built-in algorithm training image
training_image = retrieve(framework='image-classification',
                          region=aws_region, image_scope='training')

training_image = retrieve(framework='image-classification', region=aws_region,
 image_scope='training')

# Create estimator object for training
estimator = sagemaker.estimator.Estimator(image_uri=training_image,
                                          role=role,  #param role can be arn of a
 sagemaker execution role
                                          instance_count=1,
                                          instance_type='ml.p3.8xlarge',
                                          volume_size = 50,
                                          max_run = 360000,
                                          input_mode= 'File',
                                          output_path=s3_training_output_location,
                                          base_job_name='image-classification-training'
                                          )

# Setup the input data_channels to be used later for training.

train_data = sagemaker.inputs.TrainingInput(s3_training_data_location,
                                            content_type='application/x-recordio',
                                            s3_data_type='S3Prefix')
validation_data = sagemaker.inputs.TrainingInput(s3_validation_data_location,
```

```
                                                content_type='application/x-recordio',
                                                s3_data_type='S3Prefix')
data_channels = {'train': train_data, 'validation': validation_data}


# Train model
estimator.fit(inputs=data_channels, logs=True)

# Compile model with Neo

optimized_estimator = estimator.compile_model(target_instance_family='ml_c5',
                                              input_shape={'data':[1, 3, 224, 224],
 'softmax_label':[1]},
                                              output_path=s3_compilation_output_location,
                                              framework='mxnet',
                                              framework_version='1.7')
```

Untuk informasi selengkapnya tentang mengkompilasi model dengan SageMaker Python SDK, lihat. Kompilasi Model (Amazon SageMaker SDK)

Kompilasi Model () AWS Command Line Interface

Bagian ini menunjukkan cara mengelola pekerjaan kompilasi Amazon SageMaker Neo untuk model pembelajaran mesin menggunakan AWS Command Line Interface (CLI). Anda dapat membuat, mendeskripsikan, menghentikan, dan membuat daftar pekerjaan kompilasi.

1. Buat Job Kompilasi

   Dengan operasi CreateCompilationJobAPI, Anda dapat menentukan format input data, bucket S3 untuk menyimpan model Anda, bucket S3 untuk menulis model yang dikompilasi, dan perangkat atau platform perangkat keras target.

   Tabel berikut menunjukkan cara mengonfigurasi `CreateCompilationJob` API berdasarkan apakah target Anda adalah perangkat atau platform.

   Device Example

   ```
   {
       "CompilationJobName": "neo-compilation-job-demo",
       "RoleArn": "arn:aws:iam::<your-account>:role/service-role/AmazonSageMaker-
   ExecutionRole-yyyymmddThhmmss",
       "InputConfig": {
   ```

```
        "S3Uri": "s3://<your-bucket>/sagemaker/neo-compilation-job-demo-data/
train",
        "DataInputConfig":  "{'data': [1,3,1024,1024]}",
        "Framework": "MXNET"
    },
    "OutputConfig": {
        "S3OutputLocation": "s3://<your-bucket>/sagemaker/neo-compilation-job-
demo-data/compile",
        # A target device specification example for a ml_c5 instance family
        "TargetDevice": "ml_c5"
    },
    "StoppingCondition": {
        "MaxRuntimeInSeconds": 300
    }
}
```

Anda dapat secara opsional menentukan versi kerangka kerja yang Anda gunakan dengan `FrameworkVersion`bidang jika Anda menggunakan PyTorch kerangka kerja untuk melatih model Anda dan perangkat target Anda adalah `ml_*` target.

```
{
    "CompilationJobName": "neo-compilation-job-demo",
    "RoleArn": "arn:aws:iam::<your-account>:role/service-role/AmazonSageMaker-
ExecutionRole-yyyymmddThhmmss",
    "InputConfig": {
        "S3Uri": "s3://<your-bucket>/sagemaker/neo-compilation-job-demo-data/
train",
        "DataInputConfig":  "{'data': [1,3,1024,1024]}",
        "Framework": "PYTORCH",
        "FrameworkVersion": "1.6"
    },
    "OutputConfig": {
        "S3OutputLocation": "s3://<your-bucket>/sagemaker/neo-compilation-job-
demo-data/compile",
        # A target device specification example for a ml_c5 instance family
        "TargetDevice": "ml_c5",
        # When compiling for ml_* instances using PyTorch framework, use the
 "CompilerOptions" field in
        # OutputConfig to provide the correct data type ("dtype") of the model's
 input. Default assumed is "float32"
        "CompilerOptions": "{'dtype': 'long'}"
    },
    "StoppingCondition": {
```

```
        "MaxRuntimeInSeconds": 300
    }
}
```

> Catatan:
>
> - Jika Anda menyimpan model Anda dengan menggunakan PyTorch versi 2.0 atau yang lebih baru, `DataInputConfig` bidang ini opsional. SageMakerNeo mendapatkan konfigurasi input dari file definisi model yang Anda buat dengan PyTorch. Untuk informasi selengkapnya tentang cara membuat file definisi, lihat PyTorch bagian di bawah Menyimpan Model untuk SageMaker Neo.
> - Bidang API ini hanya didukung untuk PyTorch.

Platform Example

```
{
    "CompilationJobName": "neo-test-compilation-job",
    "RoleArn": "arn:aws:iam::<your-account>:role/service-role/AmazonSageMaker-
ExecutionRole-yyyymmddThhmmss",
    "InputConfig": {
        "S3Uri": "s3://<your-bucket>/sagemaker/neo-compilation-job-demo-data/
train",
        "DataInputConfig":  "{'data': [1,3,1024,1024]}",
        "Framework": "MXNET"
    },
    "OutputConfig": {
        "S3OutputLocation": "s3://<your-bucket>/sagemaker/neo-compilation-job-
demo-data/compile",
        # A target platform configuration example for a p3.2xlarge instance
        "TargetPlatform": {
            "Os": "LINUX",
            "Arch": "X86_64",
            "Accelerator": "NVIDIA"
        },
        "CompilerOptions": "{'cuda-ver': '10.0', 'trt-ver': '6.0.1', 'gpu-code':
 'sm_70'}"
    },
    "StoppingCondition": {
        "MaxRuntimeInSeconds": 300
```

```
    }
}
```

> **Note**
>
> Untuk operasi `OutputConfig` API, operasi `TargetPlatform` API `TargetDevice` dan API saling eksklusif. Anda harus memilih salah satu dari dua opsi.

Untuk menemukan contoh string JSON `DataInputConfig` tergantung pada kerangka kerja, lihat Apa bentuk data input yang diharapkan Neo.

Untuk informasi selengkapnya tentang menyiapkan konfigurasi, lihat operasi InputConfigOutputConfig, dan TargetPlatformAPI dalam referensi SageMaker API.

2. Setelah Anda mengkonfigurasi file JSON, jalankan perintah berikut untuk membuat pekerjaan kompilasi:

```
aws sagemaker create-compilation-job \
--cli-input-json file://job.json \
--region us-west-2

# You should get CompilationJobArn
```

3. Jelaskan pekerjaan kompilasi dengan menjalankan perintah berikut:

```
aws sagemaker describe-compilation-job \
--compilation-job-name $JOB_NM \
--region us-west-2
```

4. Hentikan pekerjaan kompilasi dengan menjalankan perintah berikut:

```
aws sagemaker stop-compilation-job \
--compilation-job-name $JOB_NM \
--region us-west-2

# There is no output for compilation-job operation
```

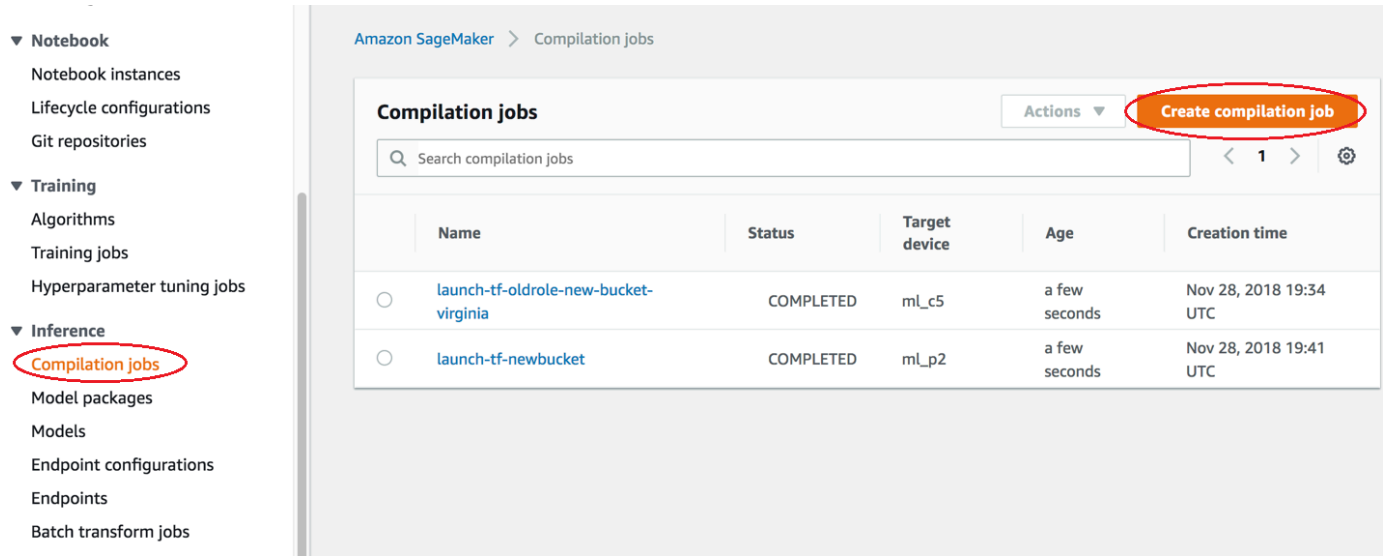
5. Buat daftar pekerjaan kompilasi dengan menjalankan perintah berikut:

```
aws sagemaker list-compilation-jobs \
--region us-west-2
```

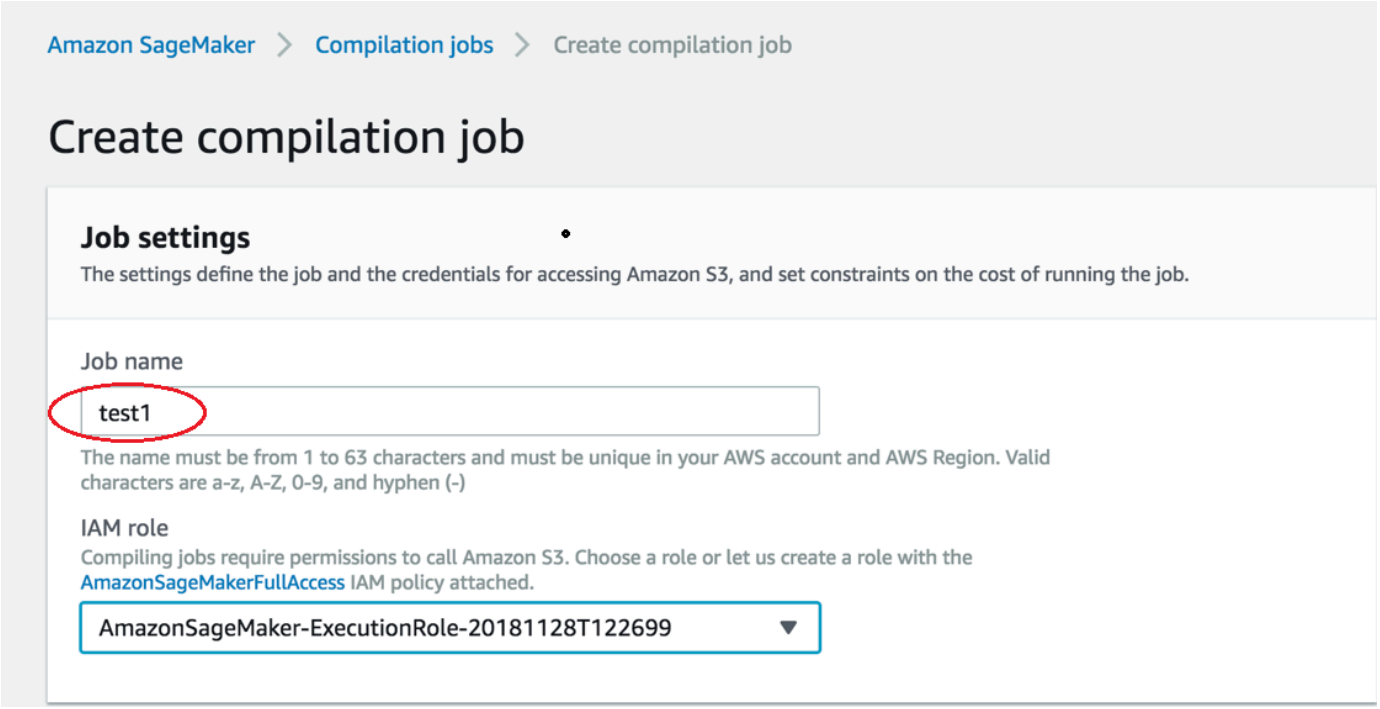
Kompilasi Model (SageMaker Konsol Amazon)

Anda dapat membuat pekerjaan kompilasi Amazon SageMaker Neo di SageMaker konsol Amazon.

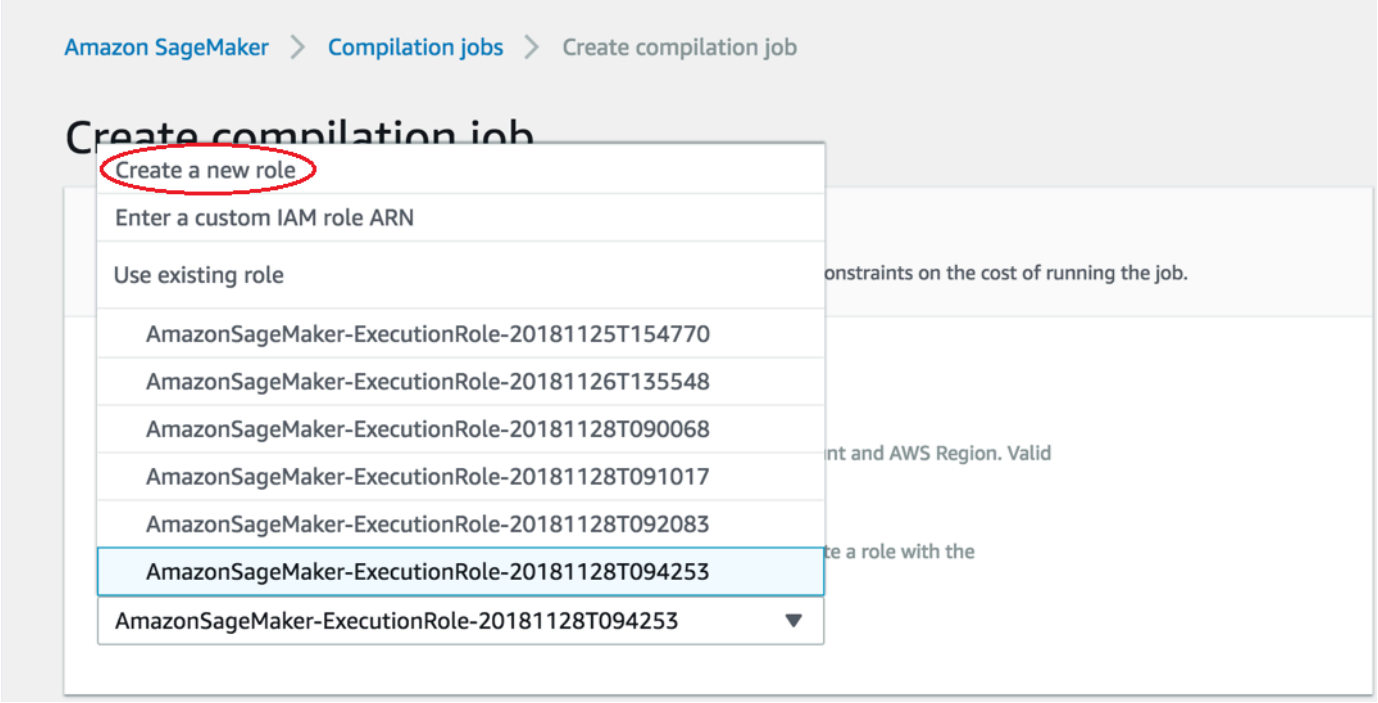
1. Di SageMaker konsol Amazon, pilih Pekerjaan kompilasi, lalu pilih Buat pekerjaan kompilasi.



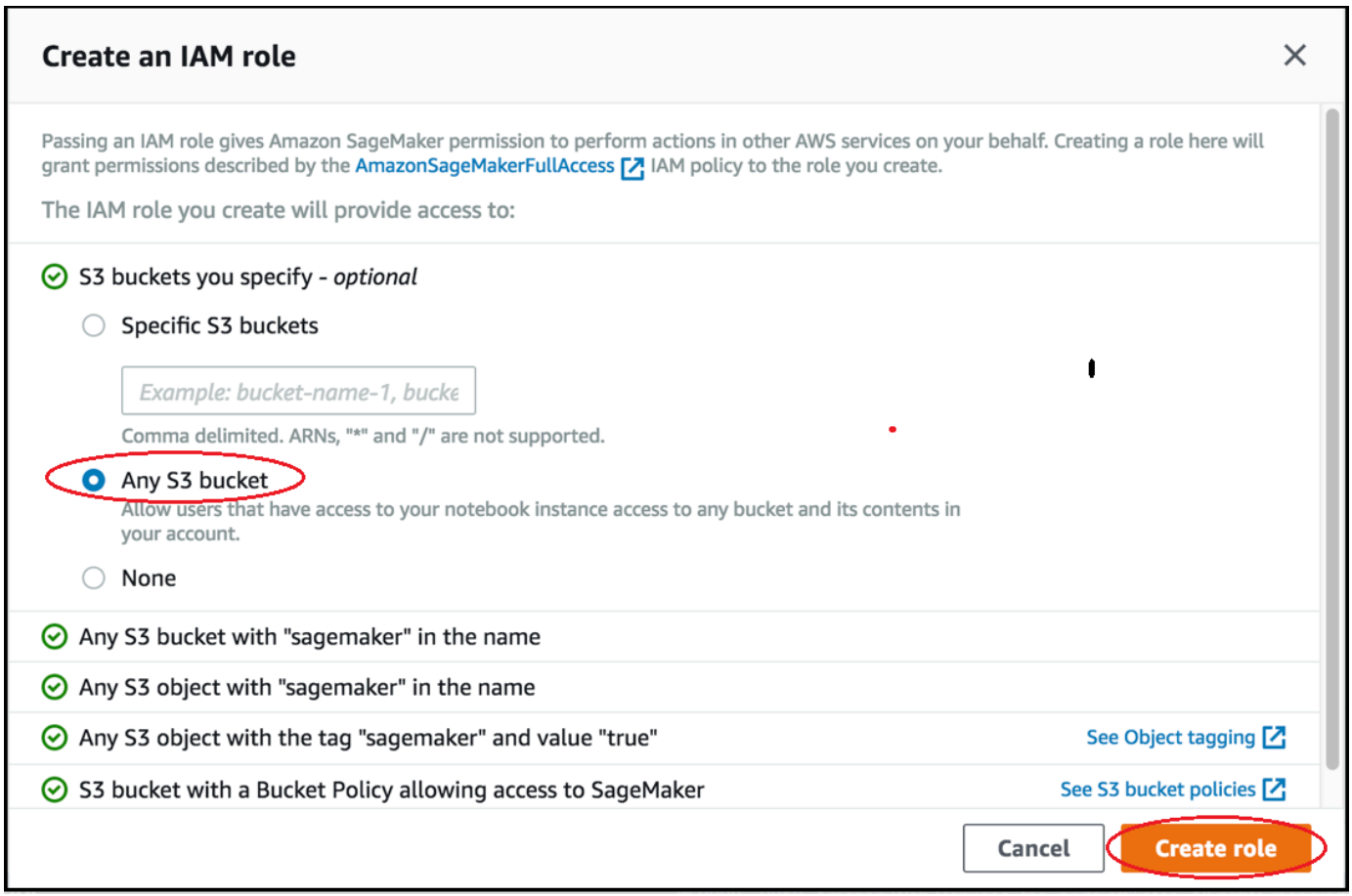
2. Pada halaman Create compilation job, di bawah Job name, masukkan nama. Kemudian pilih peran IAM.



3. Jika Anda tidak memiliki peran IAM, pilih Buat peran baru.



4. Pada halaman Create an IAM role, pilih bucket Any S3, lalu pilih Create role.

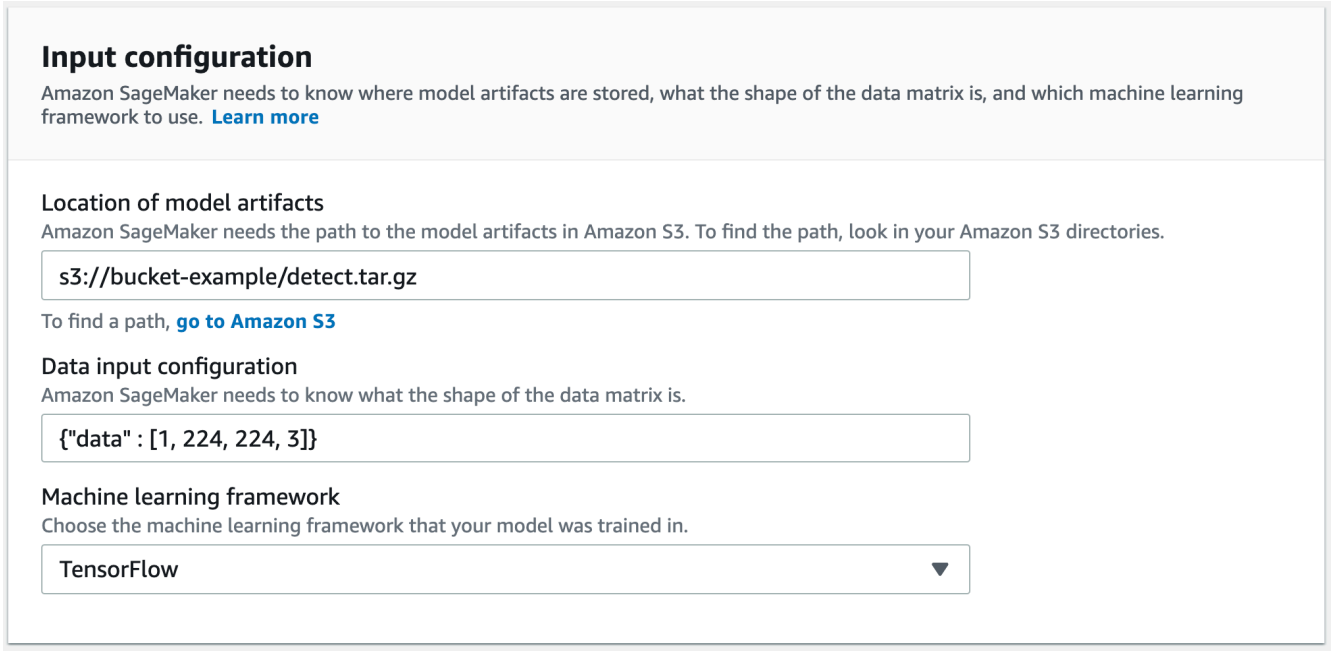


5. Non PyTorch Frameworks

   Dalam bagian Konfigurasi input, masukkan path lengkap URI bucket Amazon S3 yang berisi artefak model Anda di bidang input Lokasi artefak model. Artefak model Anda harus dalam format file tarball terkompresi (). `.tar.gz`

   Untuk bidang konfigurasi input Data, masukkan string JSON yang menentukan bentuk data input.

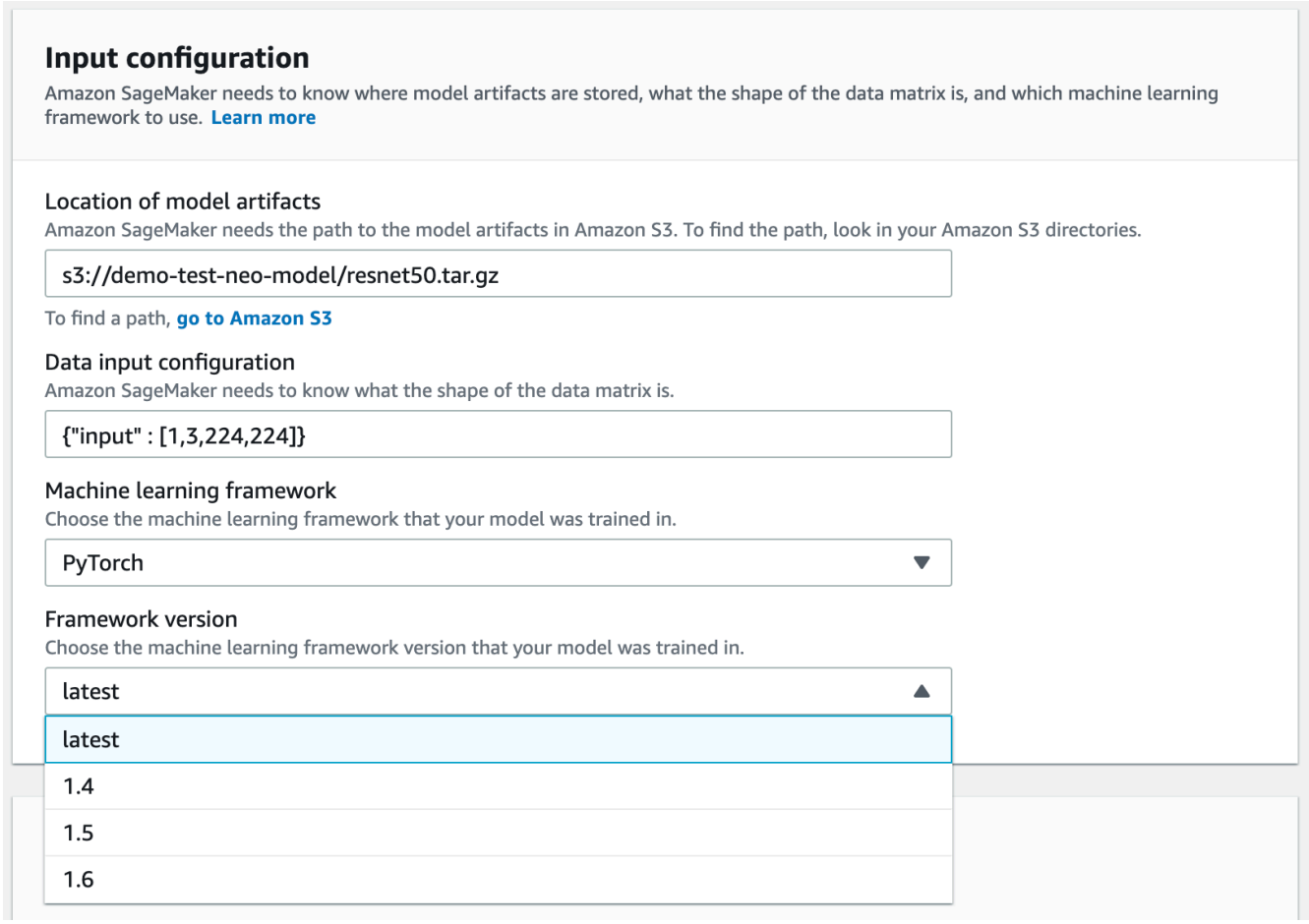
   Untuk kerangka pembelajaran mesin, pilih kerangka kerja pilihan Anda.



Untuk menemukan contoh string JSON dari bentuk data input tergantung pada kerangka kerja, lihat Apa bentuk data input yang diharapkan Neo.

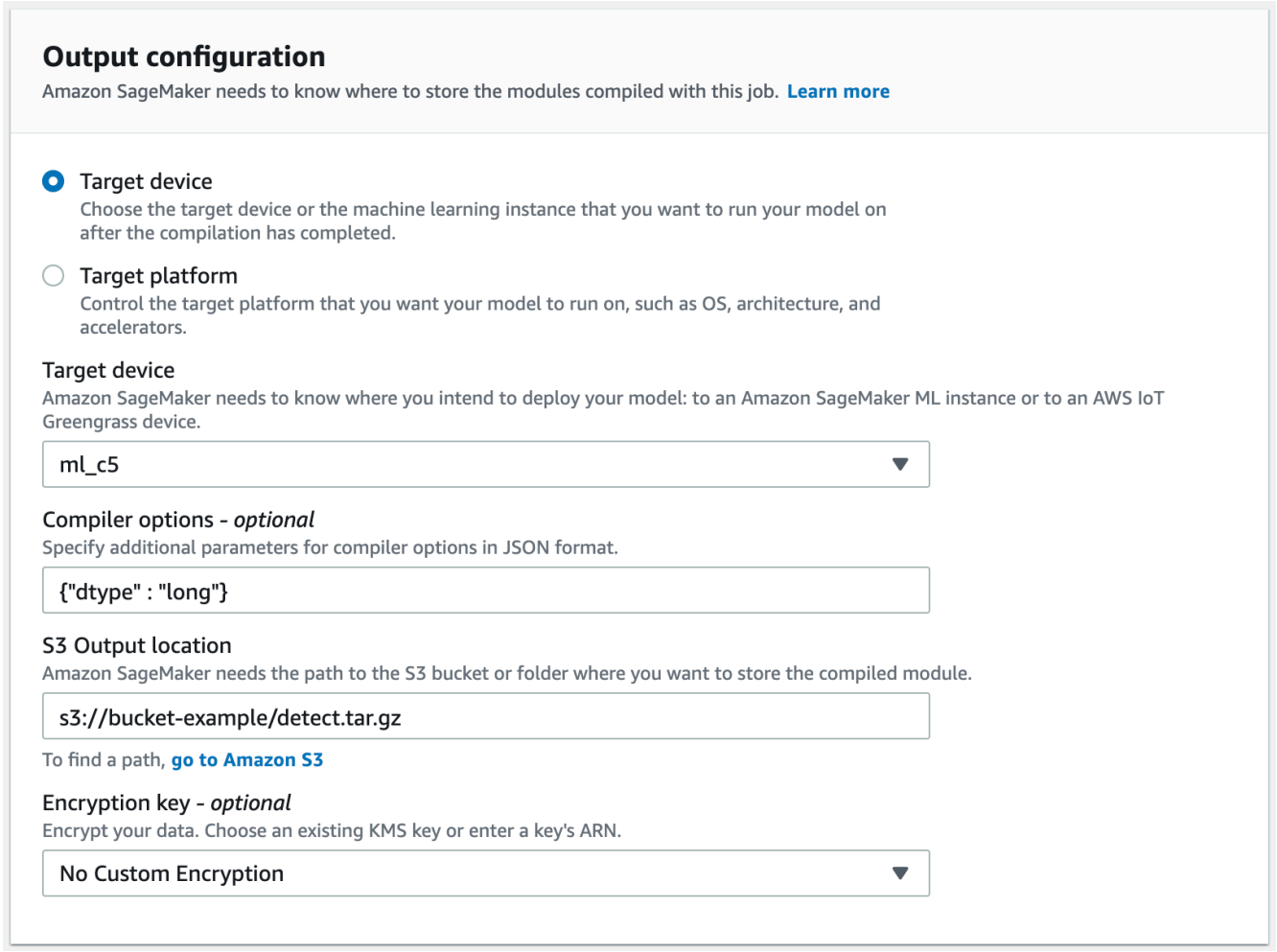
PyTorch Framework

Instruksi serupa berlaku untuk menyusun PyTorch model. Namun, jika Anda berlatih dengan PyTorch dan mencoba mengkompilasi model untuk `ml_*` (kecuali`ml_inf`) target, Anda dapat menentukan versi yang PyTorch Anda gunakan secara opsional.



Untuk menemukan contoh string JSON dari bentuk data input tergantung pada kerangka kerja, lihat Apa bentuk data input yang diharapkan Neo.

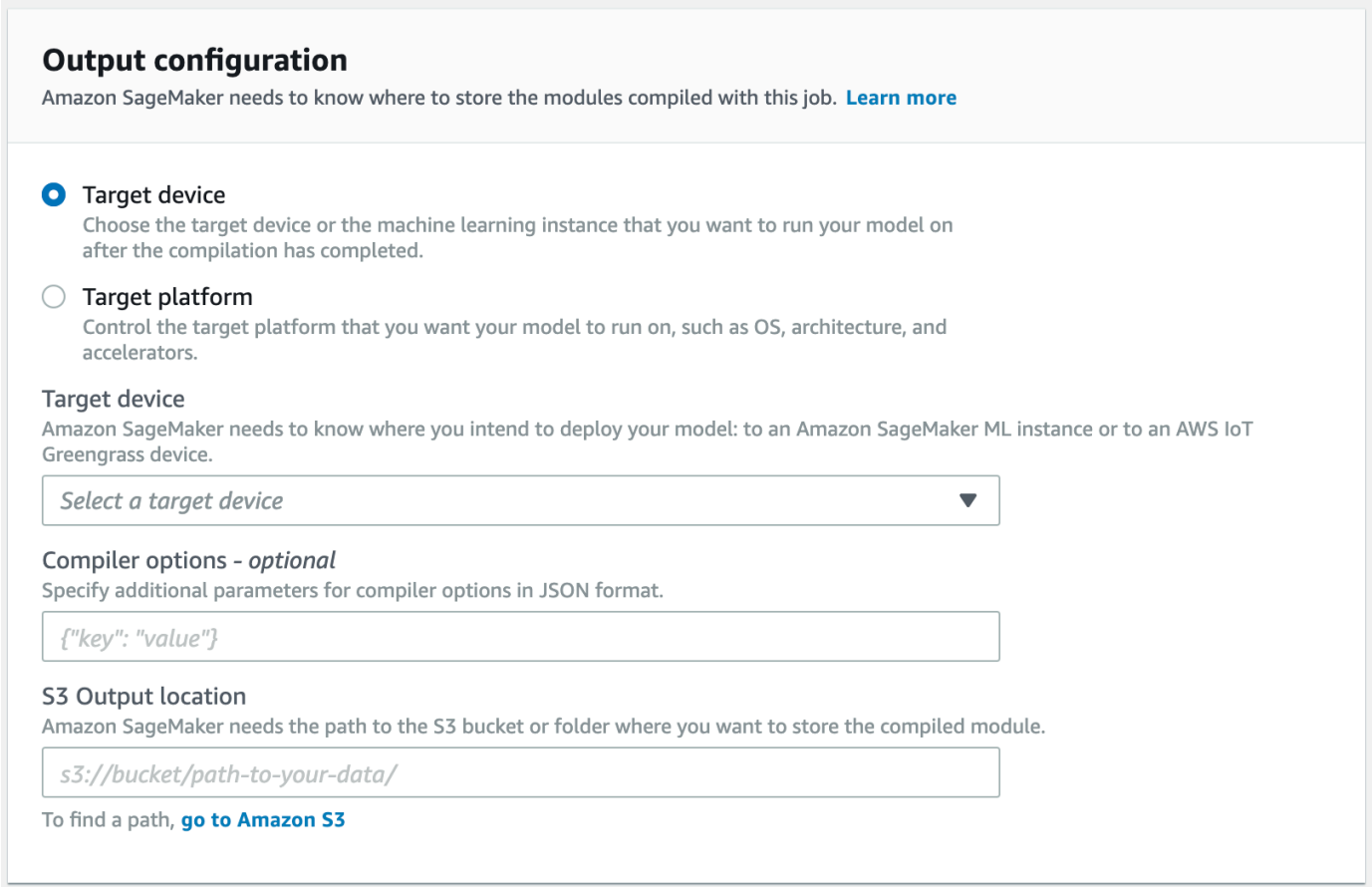
> **Catatan**
>
> - Jika Anda menyimpan model Anda dengan menggunakan PyTorch versi 2.0 atau yang lebih baru, bidang konfigurasi input data adalah opsional. SageMaker Neo mendapatkan konfigurasi input dari file definisi model yang Anda buat dengan PyTorch. Untuk informasi selengkapnya tentang cara membuat file definisi, lihat PyTorch bagian di bawah Menyimpan Model untuk SageMaker Neo.
> - Saat mengompilasi `ml_*` instance menggunakan PyTorch framework, gunakan bidang opsi Compiler di Output Configuration untuk memberikan tipe data (`dtype`) yang benar dari input model. Default diatur ke `"float32"`.



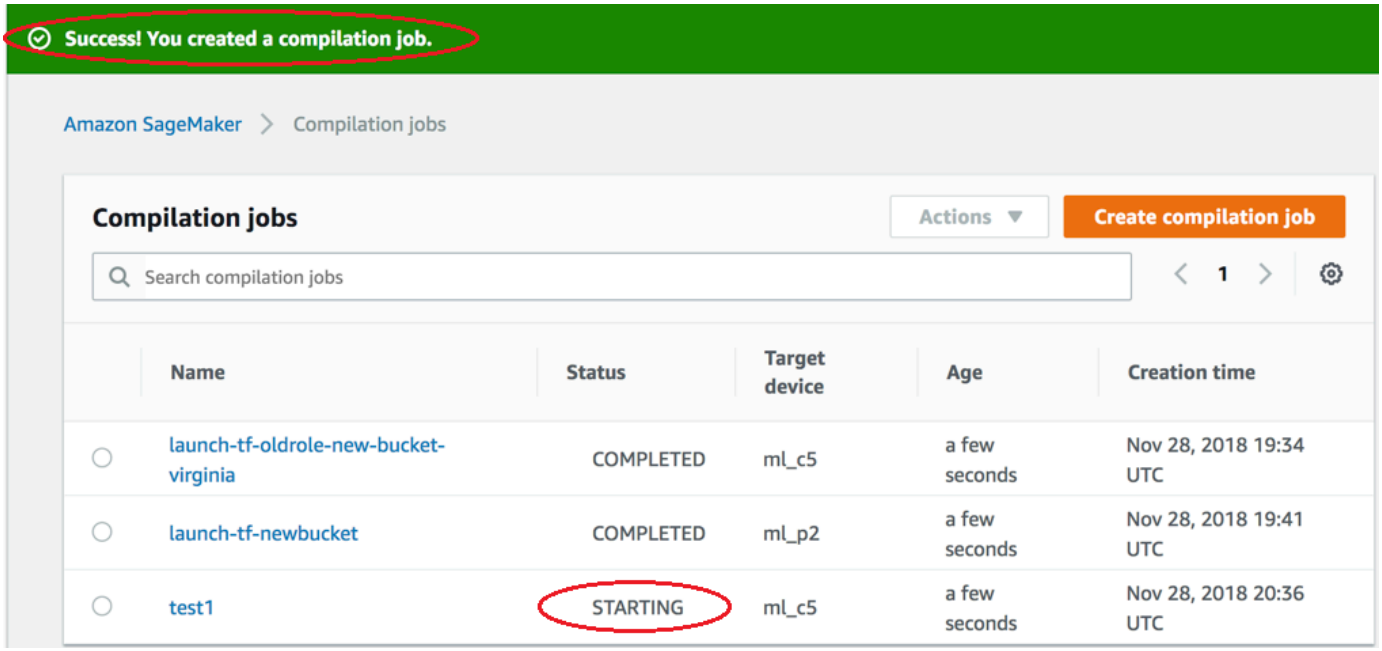
> **Warning**
>
> Jika Anda menentukan jalur URI bucket Amazon S3 yang mengarah ke `.pth` file, Anda akan menerima kesalahan berikut setelah memulai kompilasi: `ClientError: InputConfiguration: Unable to untar input model.Please confirm the model is a tar.gz file`

6. Pergi ke bagian konfigurasi Output. Pilih di mana Anda ingin menerapkan model Anda. Anda dapat menerapkan model Anda ke perangkat Target atau platform Target. Perangkat target termasuk perangkat cloud dan edge. Platform target mengacu pada OS, arsitektur, dan akselerator tertentu yang Anda inginkan agar model Anda berjalan.

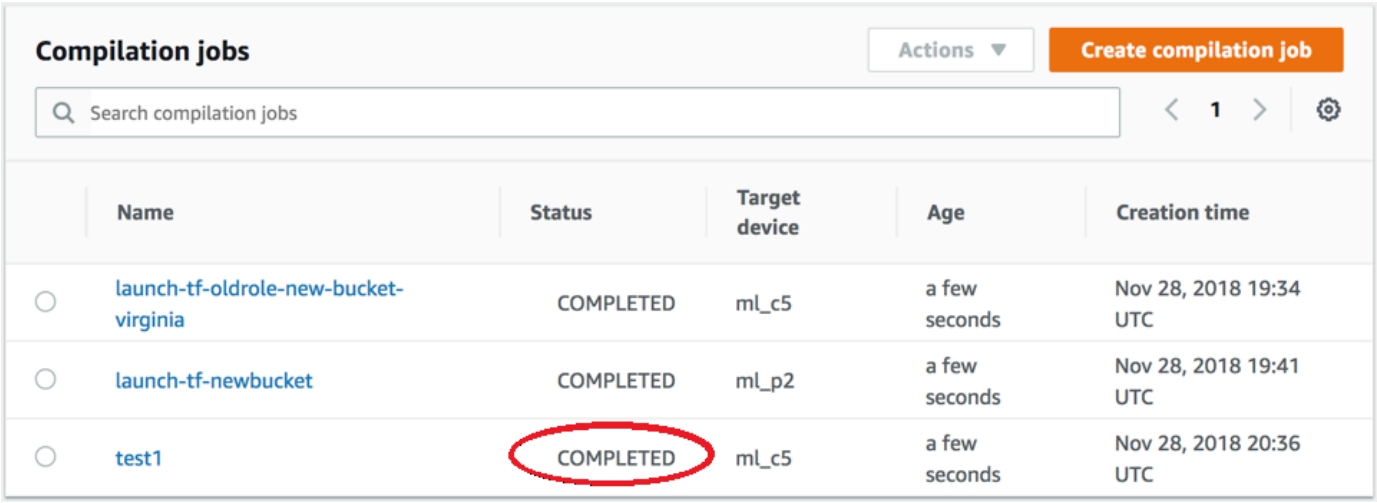
Untuk lokasi Output S3, masukkan path ke bucket S3 tempat Anda ingin menyimpan model. Anda secara opsional dapat menambahkan opsi kompiler dalam format JSON di bawah bagian Opsi kompiler.



7. Periksa status pekerjaan kompilasi saat dimulai. Status pekerjaan ini dapat ditemukan di bagian atas halaman Pekerjaan Kompilasi, seperti yang ditunjukkan pada gambar berikut. Anda juga dapat memeriksa statusnya di kolom Status.



8. Periksa status pekerjaan kompilasi setelah selesai. Anda dapat memeriksa status di kolom Status seperti yang ditunjukkan pada gambar berikut.



Kompilasi Model (Amazon SageMaker SDK)

Anda dapat menggunakan `compile_model`API di Amazon SageMaker SDK for Python untuk mengkompilasi model terlatih dan mengoptimalkannya untuk perangkat keras target tertentu. API harus dipanggil pada objek estimator yang digunakan selama pelatihan model.

> **Note**
>
> Anda harus mengatur variabel `MMS_DEFAULT_RESPONSE_TIMEOUT` lingkungan `500` saat mengkompilasi model dengan MXNet atau. PyTorch Variabel lingkungan tidak diperlukan untuk TensorFlow.

Berikut ini adalah contoh bagaimana Anda dapat mengkompilasi model menggunakan `trained_model_estimator` objek:

```
# Replace the value of expected_trained_model_input below and
# specify the name & shape of the expected inputs for your trained model
# in json dictionary form
expected_trained_model_input = {'data':[1, 784]}

# Replace the example target_instance_family below to your preferred
 target_instance_family
compiled_model = trained_model_estimator.compile_model(target_instance_family='ml_c5',
        input_shape=expected_trained_model_input,
        output_path='insert s3 output path',
        env={'MMS_DEFAULT_RESPONSE_TIMEOUT': '500'})
```

Kode mengkompilasi model, menyimpan model yang dioptimalkan di`output_path`, dan membuat SageMaker model yang dapat digunakan ke titik akhir. Contoh notebook menggunakan SDK untuk Python disediakan di bagian Notebook Contoh Kompilasi Model Neo.

Instans Cloud

Amazon SageMaker Neo menyediakan dukungan kompilasi for kerangka kerja machine learning populer seperti TensorFlow, PyTorch, MXNet, dan banyak lagi. Anda dapat menerapkan model yang dikompilasi ke instance cloud danAWSInferensia contoh. Untuk daftar lengkap kerangka kerja dan jenis instans yang didukung, lihatJenis dan Framework Instans yang Didukung.

Anda dapat mengkompilasi model Anda dengan salah satu dari tiga cara: melaluiAWS CLI, yang SageMaker Konsol, atau SageMaker SDK for Python. LihatGunakan Neo untuk Mengkompilasi ModelUntuk informasi lebih lanjut. Setelah dikompilasi, artefak model Anda disimpan dalam URI bucket Amazon S3 yang Anda tentukan selama pekerjaan kompilasi. Anda dapat menerapkan model yang dikompilasi ke instance cloud danAWSInferensia contoh menggunakan SageMaker SDK for Python,AWS SDK for Python (Boto3),AWS CLI, atauAWSkonsol.

Jika Anda menerapkan model Anda menggunakanAWS CLI, konsol, atau Boto3, Anda harus memilih URI Amazon ECR image Docker untuk wadah utama Anda. LihatGambar Kontainer Neo Inferensiuntuk daftar URI Amazon ECR.

Topik

- Jenis dan Kerangka Instance yang Didukung
- Menyebarkan Model
- Minta Inferensi dari Layanan yang Diterahkan
- Gambar Kontainer

Jenis dan Kerangka Instance yang Didukung

Amazon SageMaker Neo mendukung kerangka kerja deep learning populer untuk kompilasi dan penerapan. Anda dapat menerapkan model Anda ke instans cloud, jenis instans AWS Inferentia, atau akselerator Amazon Elastic Inference.

Berikut ini menjelaskan kerangka kerja yang didukung SageMaker Neo dan instance cloud target yang dapat Anda kompilasi dan terapkan. Untuk informasi tentang cara menerapkan model yang dikompilasi ke instans cloud atau Inferentia, lihat Menerapkan Model dengan Instans Cloud. Untuk informasi tentang cara menerapkan model yang dikompilasi dengan akselerator Elastic Inference, lihat. Gunakan EI di Titik Akhir yang SageMaker Dihosting Amazon

Instans Cloud

SageMakerNeo mendukung kerangka kerja deep learning berikut untuk instans cloud CPU dan GPU:

Kerangka Kerja	Kerangka Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
MXNet	1.8.0	Mendukung 1.8.0 atau sebelumnya	Klasifikasi Gambar, Deteksi Objek, Segmentasi Semantik,	Satu file simbol (.json) dan satu file parameter (.params)	GluonCV

Kerangka Kerja	Kerangka Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
			Estimasi Pose, Pengenalan Aktivitas		
ONNX	1.7.0	Mendukung 1.7.0 atau sebelumnya	Klasifikasi Gambar, SVM	Satu file model (.onnx)	
Keras	2.2.4	Mendukung 2.2.4 atau sebelumnya	Klasifikasi Gambar	Satu file definisi model (.h5)	
PyTorch	1.4, 1.5, 1.6, 1.7, 1.8, 1.12, 1.13, atau 2.0	Mendukung 1.4, 1.5, 1.6, 1.7, 1.8, 1.12, 1.13, dan 2.0	Klasifikasi Gambar Versi 1.13 dan 2.0 mendukung Object Detection , Vision Transform er, dan HuggingFace	Satu file definisi model (.pt atau.pth) dengan masukan dtype float32	

Kerangka Kerja	Kerangka Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
TensorFlow	1.15.3 atau 2.9	Mendukung 1.15.3 dan 2.9	Klasifikasi Gambar	Untuk model yang disimpan, satu file.pb atau satu file.pbtxt dan direktori variabel yang berisi variabel Untuk model beku, hanya satu file.pb atau .pbtxt	
XGBoost	1.3.3	Mendukung 1.3.3 atau sebelumnya	Pohon Keputusan	Satu file model XGBoost (.model) di mana jumlah node dalam pohon kurang dari 2 ^ 31	

> **Note**
>
> “Model Version” adalah versi kerangka kerja yang digunakan untuk melatih dan mengekspor model.

Tipe instans

Anda dapat menerapkan model yang SageMaker dikompilasi ke salah satu instance cloud yang tercantum di bawah ini:

Instans	Jenis Komputasi				
`ml_c4`	Standar				
`ml_c5`	Standar				
`ml_m4`	Standar				
`ml_m5`	Standar				
`ml_p2`	Komputasi yang dipercepat				
`ml_p3`	Komputasi yang dipercepat				
`ml_g4dn`	Komputasi yang dipercepat				

Untuk informasi tentang vCPU, memori, dan harga per jam yang tersedia untuk setiap jenis instans, lihat SageMakerHarga Amazon.

> **Note**
>
> Saat mengkompilasi untuk `ml_*` instance menggunakan PyTorch framework, gunakan kolom Compiler options di Output Configuration untuk menyediakan tipe data yang benar (`dtype`) dari input model.
> Default diatur ke `"float32"`.

AWSInferensia

SageMakerNeo mendukung kerangka kerja deep learning berikut untuk Inf1:

Kerangka Kerja	Kerangka Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
MXNet	1.5 atau 1,8	Mendukung 1.8, 1.5 dan sebelumnya	Klasifikasi Gambar, Deteksi Objek, Segmentasi Semantik, Estimasi Pose, Pengenalan Aktivitas	Satu file simbol (.json) dan satu file parameter (.params)	GluonCV
PyTorch	1.7, 1.8 atau 1.9	Mendukung 1.9 dan sebelumnya	Klasifikasi Gambar	Satu file definisi model (.pt atau.pth) dengan masukan dtype float32	
TensorFlow	1.15 atau 2,5	Mendukung 2.5, 1.15 dan sebelumnya	Klasifikasi Gambar	Untuk model yang disimpan, satu file.pb atau satu file.pbtxt dan direktori variabel yang berisi variabel	

Kerangka Kerja	Kerangka Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
				Untuk model beku, hanya satu file.pb atau .pbtxt	

> **Note**
>
> “Model Version” adalah versi kerangka kerja yang digunakan untuk melatih dan mengekspor model.

Anda dapat menerapkan model yang SageMaker dikompilasi NEO ke instans Amazon EC2 Inf1 berbasis AWS Inferensia. AWS Inferentia adalah chip silikon khusus pertama Amazon yang dirancang untuk mempercepat pembelajaran mendalam. Saat ini, Anda dapat menggunakan `ml_inf1` instance untuk menerapkan model yang dikompilasi.

AWSInferentia2 dan Trainium AWS

Saat ini, Anda dapat menerapkan model yang SageMaker dikompilasi NEO ke instans Amazon EC2 Inf2 AWS berbasis Inferentia2 (di Wilayah AS Timur (Ohio)), dan ke instans Amazon EC2 Trn1 berbasis Trn1 AWS berbasis Trainium (di Wilayah AS Timur (Virginia Utara)). Untuk informasi selengkapnya tentang model yang didukung pada instance ini, lihat Pedoman Sesuai Arsitektur Model dalam dokumentasi AWS Neuron, dan contoh di repositori Neuron Github.

Amazon Elastic Inference

SageMakerNeo mendukung kerangka kerja deep learning berikut untuk Elastic Inference:

Kerangka Kerja	Kerangka Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)
TensorFlow	2.3.2	Mendukung 2.3	Klasifikasi Gambar, Deteksi Objek, Segmentasi Semantik, Estimasi Pose, Pengenalan Aktivitas	Untuk model yang disimpan, satu file.pb atau satu file.pbtxt dan direktori variabel yang berisi variabel. Untuk model beku, hanya satu file.pb atau .pbtxt.

Anda dapat menerapkan model SageMaker NEO-compiled Anda ke Elastic Inference Accelerator. Untuk informasi selengkapnya, lihat Gunakan EI di Titik Akhir yang SageMaker Dihosting Amazon.

Menyebarkan Model

Menetapkan Amazon SageMaker Model yang dikompilasi neo ke titik akhir HTTPS, Anda harus mengonfigurasi dan membuat titik akhir untuk model menggunakan Amazon SageMaker layanan hosting. Saat ini, pengembang dapat menggunakan Amazon SageMaker API untuk menerapkan modul ke instance ml.c5, ml.c4, ml.m5, ml.m4, ml.p3, ml.p2, dan ml.inf1.

UntukInferensiadanTrainiumcontoh, model perlu dikompilasi secara khusus untuk instance tersebut. Model yang dikompilasi untuk jenis instance lain tidak dijamin berfungsi dengan instans Inferentia atau Trainium.

UntukAkselerator Elastic Inference, model perlu dikompilasi khusus untuk perangkat ml_eia2. Untuk informasi tentang cara menerapkan model yang dikompilasi ke akselerator Elastic Inference, lihatGunakan EI di Titik Akhir yang SageMaker Dihosting Amazon.

Saat menerapkan model yang dikompilasi, Anda perlu menggunakan instance yang sama untuk target yang Anda gunakan untuk kompilasi. Ini menciptakan SageMaker endpoint yang dapat Anda gunakan untuk melakukan inferensi. Anda dapat menggunakan model yang dikompilasi Neo

menggunakan model berikut:Amazon SageMaker SDK for Python,SDK untuk Python (Boto3),AWS Command Line Interface, danSageMakerkonsol.

> **Note**
>
> Untuk menerapkan model menggunakanAWS CLI, konsol, atau Boto3, lihatGambar Wadah Inferensi Neountuk memilih URI gambar inferensi untuk wadah utama Anda.

Topik

Prasyarat

> **Note**
>
> Ikuti petunjuk di bagian ini jika Anda menyusun model Anda menggunakanAWS SDK for Python (Boto3),AWS CLI, atau SageMaker konsol

Untuk membuat SageMaker Model yang dikompilasi neo, Anda memerlukan yang berikut:

1. Gambar Docker Amazon ECR URI. Anda dapat memilih salah satu yang memenuhi kebutuhan Anda daridaftar ini.
2. File skrip titik masuk:

    a. Untuk PyTorch dan model MXNet:

    Jika Anda melatih model Anda menggunakan SageMaker, skrip pelatihan harus mengimplementasikan fungsi yang dijelaskan di bawah ini. Skrip pelatihan berfungsi sebagai skrip titik masuk selama inferensi. Dalam contoh yang dirinci diPelatihan, Kompilasi, dan Penerapan MNIST dengan Modul MXNet dan SageMaker Neo, naskah pelatihan (`mnist.py`) mengimplementasikan fungsi yang diperlukan.

Jika Anda tidak melatih model Anda menggunakan SageMaker, Anda perlu memberikan skrip titik masuk (`inference.py`) file yang dapat digunakan pada saat inferensi. Berdasarkan Framework—MXNet atau PyTorch—lokasi skrip inferensi harus sesuai dengan SageMaker SDK PythonStruktur Direktori Model untuk MxNetatauStruktur Direktori Model untuk PyTorch.

Saat menggunakan gambar Neo Inference Optimized Container denganPyTorchdanMxNetpada jenis instance CPU dan GPU, skrip inferensi harus mengimplementasikan fungsi-fungsi berikut:

- `model_fn`: Menetapkan model (Opsional)
- `input_fn`: Mengkonversi payload permintaan masuk ke array numpy.
- `predict_fn`: Melakukan prediksi.
- `output_fn`: Mengkonversi output prediksi ke payload respon.
- Atau, Anda dapat mendefinisikan`transform_fn`untuk menggabungkan`input_fn`,`predict_fn`, dan`output_fn`.

Berikut ini adalah contoh`inference.py`script dalam direktori bernama`code`(`code/inference.py`) untukPyTorch dan MxNet (Gluon dan Modul). Contoh pertama memuat model dan kemudian menyajikannya pada data gambar pada GPU:

MXNet Module

```
import numpy as np
import json
import mxnet as mx
import neomx  # noqa: F401
from collections import namedtuple

Batch = namedtuple('Batch', ['data'])

# Change the context to mx.cpu() if deploying to a CPU endpoint
ctx = mx.gpu()

def model_fn(model_dir):
    # The compiled model artifacts are saved with the prefix 'compiled'
    sym, arg_params, aux_params = mx.model.load_checkpoint('compiled', 0)
    mod = mx.mod.Module(symbol=sym, context=ctx, label_names=None)
```

```
    exe = mod.bind(for_training=False,
                   data_shapes=[('data', (1,3,224,224))],
                   label_shapes=mod._label_shapes)
    mod.set_params(arg_params, aux_params, allow_missing=True)

    # Run warm-up inference on empty data during model load (required for
 GPU)
    data = mx.nd.empty((1,3,224,224), ctx=ctx)
    mod.forward(Batch([data]))
    return mod


def transform_fn(mod, image, input_content_type, output_content_type):
    # pre-processing
    decoded = mx.image.imdecode(image)
    resized = mx.image.resize_short(decoded, 224)
    cropped, crop_info = mx.image.center_crop(resized, (224, 224))
    normalized = mx.image.color_normalize(cropped.astype(np.float32) / 255,
                                   mean=mx.nd.array([0.485, 0.456, 0.406]),
                                   std=mx.nd.array([0.229, 0.224, 0.225]))
    transposed = normalized.transpose((2, 0, 1))
    batchified = transposed.expand_dims(axis=0)
    casted = batchified.astype(dtype='float32')
    processed_input = casted.as_in_context(ctx)

    # prediction/inference
    mod.forward(Batch([processed_input]))

    # post-processing
    prob = mod.get_outputs()[0].asnumpy().tolist()
    prob_json = json.dumps(prob)
    return prob_json, output_content_type
```

MXNet Gluon

```
import numpy as np
import json
import mxnet as mx
import neomx  # noqa: F401

# Change the context to mx.cpu() if deploying to a CPU endpoint
ctx = mx.gpu()
```

```
def model_fn(model_dir):
    # The compiled model artifacts are saved with the prefix 'compiled'
    block = mx.gluon.nn.SymbolBlock.imports('compiled-symbol.json',
['data'],'compiled-0000.params', ctx=ctx)

    # Hybridize the model & pass required options for Neo: static_alloc=True
 & static_shape=True
    block.hybridize(static_alloc=True, static_shape=True)

    # Run warm-up inference on empty data during model load (required for
 GPU)
    data = mx.nd.empty((1,3,224,224), ctx=ctx)
    warm_up = block(data)
    return block


def input_fn(image, input_content_type):
    # pre-processing
    decoded = mx.image.imdecode(image)
    resized = mx.image.resize_short(decoded, 224)
    cropped, crop_info = mx.image.center_crop(resized, (224, 224))
    normalized = mx.image.color_normalize(cropped.astype(np.float32) / 255,
                                   mean=mx.nd.array([0.485, 0.456, 0.406]),
                                   std=mx.nd.array([0.229, 0.224, 0.225]))
    transposed = normalized.transpose((2, 0, 1))
    batchified = transposed.expand_dims(axis=0)
    casted = batchified.astype(dtype='float32')
    processed_input = casted.as_in_context(ctx)
    return processed_input


def predict_fn(processed_input_data, block):
    # prediction/inference
    prediction = block(processed_input_data)
    return prediction

def output_fn(prediction, output_content_type):
    # post-processing
    prob = prediction.asnumpy().tolist()
    prob_json = json.dumps(prob)
    return prob_json, output_content_type
```

PyTorch 1.4 and Older

```
import os
import torch
import torch.nn.parallel
import torch.optim
import torch.utils.data
import torch.utils.data.distributed
import torchvision.transforms as transforms
from PIL import Image
import io
import json
import pickle


def model_fn(model_dir):
    """Load the model and return it.
    Providing this function is optional.
    There is a default model_fn available which will load the model
    compiled using SageMaker Neo. You can override it here.

    Keyword arguments:
    model_dir -- the directory path where the model artifacts are present
    """

    # The compiled model is saved as "compiled.pt"
    model_path = os.path.join(model_dir, 'compiled.pt')
    with torch.neo.config(model_dir=model_dir, neo_runtime=True):
        model = torch.jit.load(model_path)
        device = torch.device("cuda" if torch.cuda.is_available() else
 "cpu")
        model = model.to(device)

    # We recommend that you run warm-up inference during model load
    sample_input_path = os.path.join(model_dir, 'sample_input.pkl')
    with open(sample_input_path, 'rb') as input_file:
        model_input = pickle.load(input_file)
    if torch.is_tensor(model_input):
        model_input = model_input.to(device)
        model(model_input)
    elif isinstance(model_input, tuple):
        model_input = (inp.to(device) for inp in model_input if
 torch.is_tensor(inp))
```

```
        model(*model_input)
    else:
        print("Only supports a torch tensor or a tuple of torch tensors")
        return model


def transform_fn(model, request_body, request_content_type,
                 response_content_type):
    """Run prediction and return the output.
    The function
    1. Pre-processes the input request
    2. Runs prediction
    3. Post-processes the prediction output.
    """
    # preprocess
    decoded = Image.open(io.BytesIO(request_body))
    preprocess = transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize(
            mean=[
                0.485, 0.456, 0.406], std=[
                0.229, 0.224, 0.225]),
    ])
    normalized = preprocess(decoded)
    batchified = normalized.unsqueeze(0)
    # predict
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    batchified = batchified.to(device)
    output = model.forward(batchified)

    return json.dumps(output.cpu().numpy().tolist()), response_content_type
```

PyTorch 1.5 and Newer

```
import os
import torch
import torch.nn.parallel
import torch.optim
import torch.utils.data
import torch.utils.data.distributed
import torchvision.transforms as transforms
```

```
from PIL import Image
import io
import json
import pickle


def model_fn(model_dir):
    """Load the model and return it.
    Providing this function is optional.
    There is a default_model_fn available, which will load the model
    compiled using SageMaker Neo. You can override the default here.
    The model_fn only needs to be defined if your model needs extra
    steps to load, and can otherwise be left undefined.

    Keyword arguments:
    model_dir -- the directory path where the model artifacts are present
    """

    # The compiled model is saved as "model.pt"
    model_path = os.path.join(model_dir, 'model.pt')
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    model = torch.jit.load(model_path, map_location=device)
    model = model.to(device)

    return model


def transform_fn(model, request_body, request_content_type,
                 response_content_type):
    """Run prediction and return the output.
    The function
    1. Pre-processes the input request
    2. Runs prediction
    3. Post-processes the prediction output.
    """
    # preprocess
    decoded = Image.open(io.BytesIO(request_body))
    preprocess = transforms.Compose([
                                transforms.Resize(256),
                                transforms.CenterCrop(224),
                                transforms.ToTensor(),
                                transforms.Normalize(
                                    mean=[
                                        0.485, 0.456, 0.406], std=[
```

```
                                        0.229, 0.224, 0.225]),
                                ])
    normalized = preprocess(decoded)
    batchified = normalized.unsqueeze(0)

    # predict
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    batchified = batchified.to(device)
    output = model.forward(batchified)
    return json.dumps(output.cpu().numpy().tolist()), response_content_type
```

b. Untuk instance inf1 atau onnx, xgboost, gambar kontainer keras

Untuk semua gambar kontainer yang dioptimalkan Neo Inference lainnya, atau jenis instance inferentia, skrip titik masuk harus mengimplementasikan fungsi berikut untuk Neo Deep Learning Runtime:

- `neo_preprocess`: Mengkonversi payload permintaan masuk ke array numpy.
- `neo_postprocess`: Mengubah output prediksi dari Neo Deep Learning Runtime menjadi badan respons.

> **Note**
>
> Dua fungsi sebelumnya tidak menggunakan salah satu fungsi MXNet, PyTorch, atau TensorFlow.

Contoh cara menggunakan fungsi ini, lihatNotebook Contoh Kompilasi Model Neo.

c. Untuk TensorFlow model

Jika model Anda memerlukan logika pra-dan pasca-pemrosesan khusus sebelum data dikirim ke model, maka Anda harus menentukan skrip titik masuk`inference.py`file yang dapat digunakan pada saat inferensi. Skrip harus mengimplementasikan salah satu dari sepasang`input_handler`dan`output_handler`fungsi atau fungsi handler tunggal.

> **Note**
>
> Perhatikan bahwa jika fungsi handler diimplementasikan,`input_handler`dan`output_handler`diabaikan.

Berikut ini adalah contoh kode`inference.py`skrip yang dapat Anda kumpulkan dengan model kompilasi untuk melakukan pra-dan sesudah pemrosesan khusus pada model klasifikasi gambar. The SageMaker klien mengirimkan file gambar sebagai`application/x-image`jenis konten ke`input_handler`fungsi, di mana ia dikonversi ke JSON. File gambar yang dikonversi kemudian dikirim keServer Model Tensorflow (TFX)menggunakan REST API.

```
import json
import numpy as np
import json
import io
from PIL import Image

def input_handler(data, context):
    """ Pre-process request input before it is sent to TensorFlow Serving REST
 API

    Args:
    data (obj): the request data, in format of dict or string
    context (Context): an object containing request and configuration details

    Returns:
    (dict): a JSON-serializable dict that contains request body and headers
    """
    f = data.read()
    f = io.BytesIO(f)
    image = Image.open(f).convert('RGB')
    batch_size = 1
    image = np.asarray(image.resize((512, 512)))
    image = np.concatenate([image[np.newaxis, :, :]] * batch_size)
    body = json.dumps({"signature_name": "serving_default", "instances":
 image.tolist()})
    return body
```

```
def output_handler(data, context):
    """Post-process TensorFlow Serving output before it is returned to the
 client.

    Args:
    data (obj): the TensorFlow serving response
    context (Context): an object containing request and configuration details

    Returns:
    (bytes, string): data to return to client, response content type
    """
    if data.status_code != 200:
        raise ValueError(data.content.decode('utf-8'))

    response_content_type = context.accept_header
    prediction = data.content
    return prediction, response_content_type
```

Jika tidak ada pra-atau pasca-pemrosesan khusus, SageMaker klien mengonversi gambar file ke JSON dengan cara yang sama sebelum mengirimnya ke SageMaker titik akhir.

Untuk informasi lebih lanjut, lihatMenetapkan TensorFlow Melayani Endpoint di SageMaker SDK Python.

3. URI bucket Amazon S3 yang berisi artefak model yang dikompilasi.

Menerapkan Model Kompilasi Menggunakan SageMaker SDK

Anda harus memuaskanprasyaratbagian jika model dikompilasi menggunakanAWS SDK for Python (Boto3),AWS CLI, atau Amazon SageMaker konsol Ikuti salah satu kasus penggunaan berikut untuk menerapkan model yang dikompilasi dengan SageMaker Neo berdasarkan bagaimana Anda menyusun model Anda.

Topik

- Jika Anda mengkompilasi model Anda menggunakan SageMaker SDK
- Jika Anda mengkompilasi model Anda menggunakan MXNet atau PyTorch
- Jika Anda mengkompilasi model Anda menggunakan Boto3, SageMaker konsol, atau CLI untuk TensorFlow

Jika Anda mengkompilasi model Anda menggunakan SageMaker SDK

TheSagemaker.modelpegangan objek untuk model yang dikompilasi memasokmenyebarkan ()fungsi, yang memungkinkan Anda untuk membuat endpoint untuk melayani permintaan inferensi. Fungsi ini memungkinkan Anda mengatur jumlah dan jenis instance yang digunakan untuk titik akhir. Anda harus memilih contoh yang telah Anda kompilasi model Anda. Misalnya, dalam pekerjaan yang dikompilasi diKompilasi Model (Amazon SageMaker SDK)bagian, ini adalah`ml_c5`.

```
predictor = compiled_model.deploy(initial_instance_count = 1, instance_type =
 'ml.c5.4xlarge')

# Print the name of newly created endpoint
print(predictor.endpoint_name)
```

Jika Anda mengkompilasi model Anda menggunakan MXNet atau PyTorch

Buat SageMaker memodelkan dan menerapkannya menggunakan API deploy () di bawah API Model khusus kerangka kerja. Untuk MxNet, ituMXNetModeldan untuk PyTorch, itu PyTorchModel. Saat Anda membuat dan menerapkan SageMaker model, Anda harus mengaturMMS_DEFAULT_RESPONSE_TIMEOUTvariabel lingkungan500Menetapkan`entry_point`parameter sebagai skrip inferensi (`inference.py`) dan`source_dir`parameter sebagai lokasi direktori (`code`) dari skrip inferensi. Menetapkan skrip inferensi (`inference.py`) ikuti langkah Prasyarat.

Contoh berikut menunjukkan cara menggunakan fungsi ini untuk menggunakan fungsi ini untuk menggunakan fungsi ini untuk menggunakan fungsi ini untuk menggunakan fungsi ini untuk menggunakan fungsi ini untuk menggunakan fungsi ini untuk menggunakan fungsi untuk menggunakan fungsi SageMaker SDK untuk Python:

MXNet

```
from sagemaker.mxnet import MXNetModel

# Create SageMaker model and deploy an endpoint
sm_mxnet_compiled_model = MXNetModel(
    model_data='insert S3 path of compiled MXNet model archive',
    role='AmazonSageMaker-ExecutionRole',
    entry_point='inference.py',
    source_dir='code',
    framework_version='1.8.0',
```

```
    py_version='py3',
    image_uri='insert appropriate ECR Image URI for MXNet',
    env={'MMS_DEFAULT_RESPONSE_TIMEOUT': '500'},
)

# Replace the example instance_type below to your preferred instance_type
predictor = sm_mxnet_compiled_model.deploy(initial_instance_count = 1, instance_type
 = 'ml.p3.2xlarge')

# Print the name of newly created endpoint
print(predictor.endpoint_name)
```

PyTorch 1.4 and Older

```
from sagemaker.pytorch import PyTorchModel

# Create SageMaker model and deploy an endpoint
sm_pytorch_compiled_model = PyTorchModel(
    model_data='insert S3 path of compiled PyTorch model archive',
    role='AmazonSageMaker-ExecutionRole',
    entry_point='inference.py',
    source_dir='code',
    framework_version='1.4.0',
    py_version='py3',
    image_uri='insert appropriate ECR Image URI for PyTorch',
    env={'MMS_DEFAULT_RESPONSE_TIMEOUT': '500'},
)

# Replace the example instance_type below to your preferred instance_type
predictor = sm_pytorch_compiled_model.deploy(initial_instance_count = 1,
 instance_type = 'ml.p3.2xlarge')

# Print the name of newly created endpoint
print(predictor.endpoint_name)
```

PyTorch 1.5 and Newer

```
from sagemaker.pytorch import PyTorchModel

# Create SageMaker model and deploy an endpoint
sm_pytorch_compiled_model = PyTorchModel(
    model_data='insert S3 path of compiled PyTorch model archive',
    role='AmazonSageMaker-ExecutionRole',
```

```
    entry_point='inference.py',
    source_dir='code',
    framework_version='1.5',
    py_version='py3',
    image_uri='insert appropriate ECR Image URI for PyTorch',
)

# Replace the example instance_type below to your preferred instance_type
predictor = sm_pytorch_compiled_model.deploy(initial_instance_count = 1,
 instance_type = 'ml.p3.2xlarge')

# Print the name of newly created endpoint
print(predictor.endpoint_name)
```

> **Note**
>
> TheAmazonSageMakerFullAccessdanAmazonS3ReadOnlyAccessKebijakan harus dilampirkan padaAmazonSageMaker-ExecutionRolePeran IAM.

Jika Anda mengkompilasi model Anda menggunakan Boto3, SageMaker konsol, atau CLI untuk TensorFlow

MenetapkanTensorFlowModelobjek, lalu panggil deploy:

```
role='AmazonSageMaker-ExecutionRole'
model_path='S3 path for model file'
framework_image='inference container arn'
tf_model = TensorFlowModel(model_data=model_path,
                framework_version='1.15.3',
                role=role,
                image_uri=framework_image)
instance_type='ml.c5.xlarge'
predictor = tf_model.deploy(instance_type=instance_type,
                    initial_instance_count=1)
```

LihatMenyebarkan langsung dari artefak modeluntuk informasi lebih lanjut.

Anda dapat memilih gambar Docker Amazon ECR URI yang memenuhi kebutuhan Anda daridaftar ini.

Untuk informasi lebih lanjut tentang cara membangun`TensorFlowModel`objek, lihatSageMaker SDK.

> **Note**
>
> Permintaan inferensi pertama Anda mungkin memiliki latensi tinggi jika Anda menerapkan model Anda pada GPU. Ini karena kernel komputasi yang dioptimalkan dibuat pada permintaan inferensi pertama. Kami menyarankan Anda membuat file pemanasan permintaan inferensi dan menyimpannya di samping file model Anda sebelum mengirimkannya ke TFX. Ini dikenal sebagai “pemanasan” model.

Cuplikan kode berikut menunjukkan cara menghasilkan file pemanasan untuk contoh klasifikasi gambar diprasyaratbagian:

```
import tensorflow as tf
from tensorflow_serving.apis import classification_pb2
from tensorflow_serving.apis import inference_pb2
from tensorflow_serving.apis import model_pb2
from tensorflow_serving.apis import predict_pb2
from tensorflow_serving.apis import prediction_log_pb2
from tensorflow_serving.apis import regression_pb2
import numpy as np

with tf.python_io.TFRecordWriter("tf_serving_warmup_requests") as writer:
    img = np.random.uniform(0, 1, size=[224, 224, 3]).astype(np.float32)
    img = np.expand_dims(img, axis=0)
    test_data = np.repeat(img, 1, axis=0)
    request = predict_pb2.PredictRequest()
    request.model_spec.name = 'compiled_models'
    request.model_spec.signature_name = 'serving_default'
    request.inputs['Placeholder:0'].CopyFrom(tf.compat.v1.make_tensor_proto(test_data,
 shape=test_data.shape, dtype=tf.float32))
    log = prediction_log_pb2.PredictionLog(
    predict_log=prediction_log_pb2.PredictLog(request=request))
    writer.write(log.SerializeToString())
```

Untuk informasi lebih lanjut cara “menghangatkan” model Anda, lihat cara menggunakanTensorFlow Halaman TFX.

Menyebarkan Model Kompilasi Menggunakan Boto3

Anda harus memuaskanprasyaratbagian jika model dikompilasi menggunakanAWS SDK for Python (Boto3),AWS CLI, atau Amazon SageMaker konsol Ikuti langkah-langkah berikut untuk membuat dan menggunakan cara menggunakan SageMaker Model yang dikompilasi neo menggunakanSDK Layanan Web Amazon untuk Python (Boto3).

Topik

- Menetapkan Model

Menetapkan Model

Setelah Anda puasprasyarat, gunakan`create_model`,`create_enpoint_config`, dan`create_endpoint`API.

Contoh berikut menunjukkan cara menggunakan API untuk menerapkan model yang dikompilasi dengan Neo:

```
import boto3
client = boto3.client('sagemaker')

# create sagemaker model
create_model_api_response = client.create_model(
                                    ModelName='my-sagemaker-model',
                                    PrimaryContainer={
                                        'Image': <insert the ECR Image URI>,
                                        'ModelDataUrl': 's3://path/to/model/artifact/
model.tar.gz',
                                        'Environment': {}
                                    },
                                    ExecutionRoleArn='ARN for AmazonSageMaker-
ExecutionRole'
                            )

print ("create_model API response", create_model_api_response)

# create sagemaker endpoint config
create_endpoint_config_api_response = client.create_endpoint_config(
                                            EndpointConfigName='sagemaker-neomxnet-
endpoint-configuration',
                                            ProductionVariants=[
```

```
                                                {
                                                    'VariantName': <provide your
 variant name>,
                                                    'ModelName': 'my-sagemaker-model',
                                                    'InitialInstanceCount': 1,
                                                    'InstanceType': <provide your
 instance type here>
                                                },
                                            ]
                                        )

print ("create_endpoint_config API response", create_endpoint_config_api_response)

# create sagemaker endpoint
create_endpoint_api_response = client.create_endpoint(
                                    EndpointName='provide your endpoint name',
                                    EndpointConfigName=<insert your endpoint config
 name>,
                                )

print ("create_endpoint API response", create_endpoint_api_response)
```

> **Note**
>
> TheAmazonSageMakerFullAccessdanAmazonS3ReadOnlyAccessKebijakan harus dilampirkan padaAmazonSageMaker-ExecutionRolePeran IAM.

Untuk sintaks lengkapcreate_model,create_endpoint_config, dancreate_endpointAPI, lihatcreate_model,create_endpoint_config, dancreate_endpoint, masing-masing.

Jika Anda tidak melatih model Anda menggunakan SageMaker, tentukan variabel lingkungan berikut:

MXNet and PyTorch

```
"Environment": {
    "SAGEMAKER_PROGRAM": "inference.py",
    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code",
    "SAGEMAKER_CONTAINER_LOG_LEVEL": "20",
    "SAGEMAKER_REGION": "insert your region",
    "MMS_DEFAULT_RESPONSE_TIMEOUT": "500"
}
```

TensorFlow

```
"Environment": {
    "SAGEMAKER_PROGRAM": "inference.py",
    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code",
    "SAGEMAKER_CONTAINER_LOG_LEVEL": "20",
    "SAGEMAKER_REGION": "insert your region"
}
```

Jika Anda melatih model Anda menggunakan SageMakerMenetapkan variabel lingkunganSAGEMAKER_SUBMIT_DIRECTORYsebagai URI bucket Amazon S3 lengkap yang berisi skrip pelatihan.

Menerapkan Model Kompilasi MenggunakanAWS CLI

Anda harus memuaskanprasyaratbagian jika model dikompilasi menggunakanAWS SDK for Python (Boto3),AWS CLI, atau Amazon SageMaker konsol Ikuti langkah-langkah berikut untuk membuat dan menggunakan cara menggunakan SageMaker Model yang dikompilasi neo menggunakanAWS CLI.

Topik

- Menetapkan Model

Menetapkan Model

Setelah Anda puasprasyarat, gunakancreate-model,create-enpoint-config, dancreate-endpoint AWS CLIperintah Langkah-langkah berikut menjelaskan cara menggunakan perintah ini untuk menggunakan perintah ini untuk menerapkan model yang dikompilasi dengan Neo:

Menetapkan Model

DariGambar Wadah Inferensi Neo, pilih URI gambar inferensi dan kemudian gunakancreate-modelAPI untuk membuat SageMaker model Anda dapat melakukan ini dengan dua langkah:

1. Buatcreate_model.jsonberkas. Di dalam file, tentukan nama model, URI gambar, jalur kemodel.tar.gzfile dalam ember Amazon S3 Anda, dan SageMakerperan eksekusi:

```
{
    "ModelName": "insert model name",
    "PrimaryContainer": {
```

```
        "Image": "insert the ECR Image URI",
        "ModelDataUrl": "insert S3 archive URL",
        "Environment": {"See details below"}
    },
    "ExecutionRoleArn": "ARN for AmazonSageMaker-ExecutionRole"
}
```

Jika Anda melatih model Anda menggunakan SageMaker, tentukan variabel lingkungan berikut:

```
"Environment": {
    "SAGEMAKER_SUBMIT_DIRECTORY" : "[Full S3 path for *.tar.gz file containing the
 training script]"
}
```

Jika Anda tidak melatih model Anda menggunakan SageMaker, tentukan variabel lingkungan berikut:

MXNet and PyTorch

```
"Environment": {
    "SAGEMAKER_PROGRAM": "inference.py",
    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code",
    "SAGEMAKER_CONTAINER_LOG_LEVEL": "20",
    "SAGEMAKER_REGION": "insert your region",
    "MMS_DEFAULT_RESPONSE_TIMEOUT": "500"
}
```

TensorFlow

```
"Environment": {
    "SAGEMAKER_PROGRAM": "inference.py",
    "SAGEMAKER_SUBMIT_DIRECTORY": "/opt/ml/model/code",
    "SAGEMAKER_CONTAINER_LOG_LEVEL": "20",
    "SAGEMAKER_REGION": "insert your region"
}
```

> **Note**
>
> TheAmazonSageMakerFullAccessdanAmazonS3ReadOnlyAccessKebijakan harus dilampirkan padaAmazonSageMaker-ExecutionRolePeran IAM.

2. Jalankan perintah berikut:

```
aws sagemaker create-model --cli-input-json file://create_model.json
```

Untuk sintaks lengkap`create-model`API, lihat`create-model`.

Menetapkan Konfigurasi Endpoint

Setelah membuat SageMaker model, buat konfigurasi titik akhir menggunakan`create-endpoint-config`API. Untuk melakukan ini, buat file JSON dengan spesifikasi konfigurasi endpoint. Misalnya, Anda dapat menggunakan template kode berikut dan menyimpannya sebagai:`create_config.json`:

```
{
    "EndpointConfigName": "<provide your endpoint config name>",
    "ProductionVariants": [
        {
            "VariantName": "<provide your variant name>",
            "ModelName": "my-sagemaker-model",
            "InitialInstanceCount": 1,
            "InstanceType": "<provide your instance type here>",
            "InitialVariantWeight": 1.0
        }
    ]
}
```

Sekarang jalankan yang berikutAWS CLIperintah untuk membuat konfigurasi titik akhir Anda:

```
aws sagemaker create-endpoint-config --cli-input-json file://create_config.json
```

Untuk sintaks lengkap`create-endpoint-config`API, lihat`create-endpoint-config`.

Menetapkan Endpoint

Setelah Anda membuat konfigurasi titik akhir Anda, buat titik akhir menggunakan`create-endpoint`API:

```
aws sagemaker create-endpoint --endpoint-name '<provide your endpoint name>' --
endpoint-config-name '<insert your endpoint config name>'
```

Untuk sintaks lengkap`create-endpoint`API, lihat`create-endpoint`.

Menerapkan Model Terkompilasi Menggunakan Konsol

Anda harus memuaskanprasyaratbagian jika model dikompilasi menggunakanAWS SDK for Python (Boto3),AWS CLI, atau Amazon SageMaker konsol Ikuti langkah-langkah berikut untuk membuat dan menggunakan cara menggunakan SageMakerModel yang dikompilasi neo menggunakan SageMakerkonsolhttps://console.aws.amazon.com/ SageMaker.

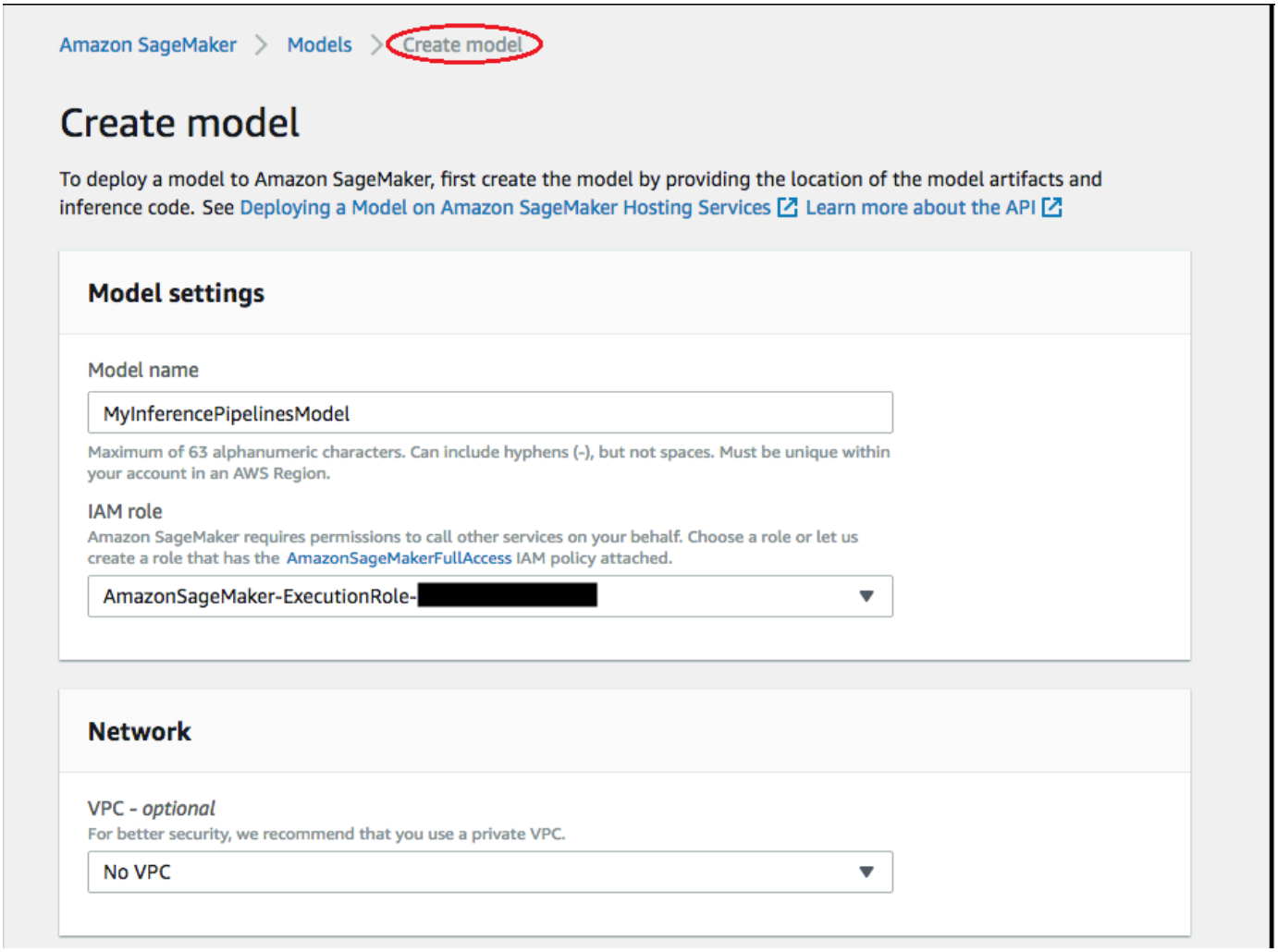
Topik

- Menetapkan Model

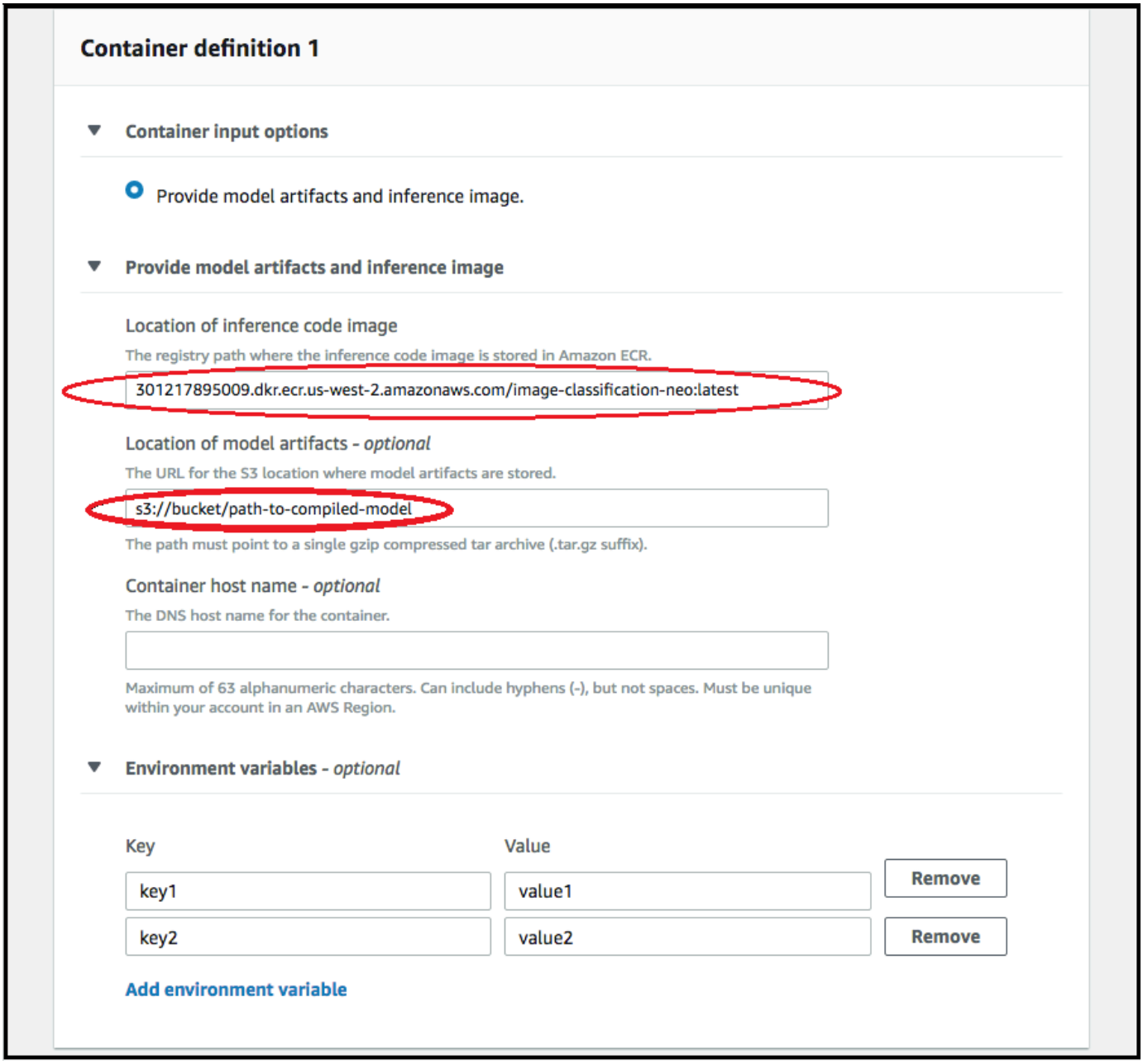
Menetapkan Model

Setelah Anda puasprasyarat, gunakan langkah-langkah berikut untuk menerapkan model yang dikompilasi dengan Neo:

1. PilihModel, dan kemudian pilihBuat modeldariInferensikelompok. PadaBuat modelhalaman, selesaikanNama model,Peran IAM, danVPCbidang (opsional), jika diperlukan.



2. Untuk menambahkan informasi tentang wadah yang digunakan untuk menerapkan model Anda, pilihTambahkan wadahwadah, lalu pilihBerikutnya. MenetapkanMenetapkan opsi input kontainer,Lokasi gambar kode inferensi, danLokasi artefak model, dan secara opsional,Nama host kontainer, danVariabel lingkunganladang.



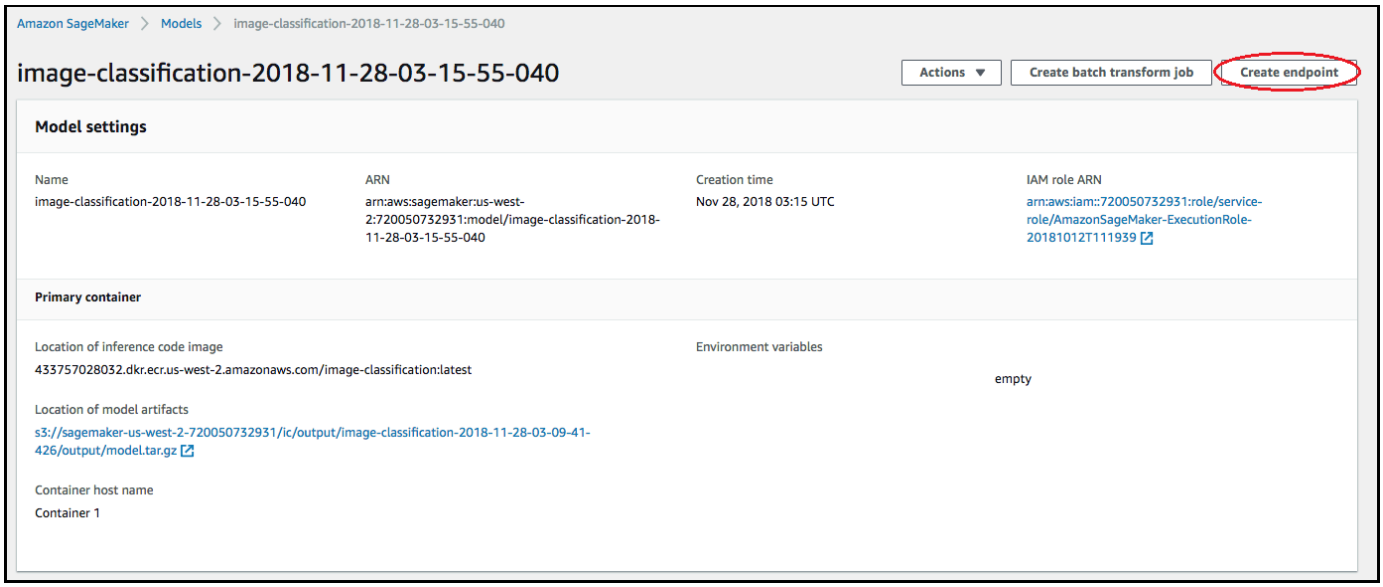
3. Untuk menerapkan model yang dikompilasi NEO, pilih yang berikut ini:

   - Menetapkan opsi input kontainer: PilihBerikan artefak model dan gambar inferensi.
   - Lokasi gambar kode inferensi: Pilih URI gambar inferensi dariGambar Wadah Inferensi Neo, tergantung padaAWSWilayah dan jenis aplikasi.
   - Lokasi artefak model: Masukkan URI bucket Amazon S3 dari artefak model terkompilasi yang dihasilkan oleh API kompilasi Neo.
   - Variabel lingkungan:
     - Menetapkan bidang iniSageMakerXGBoost.

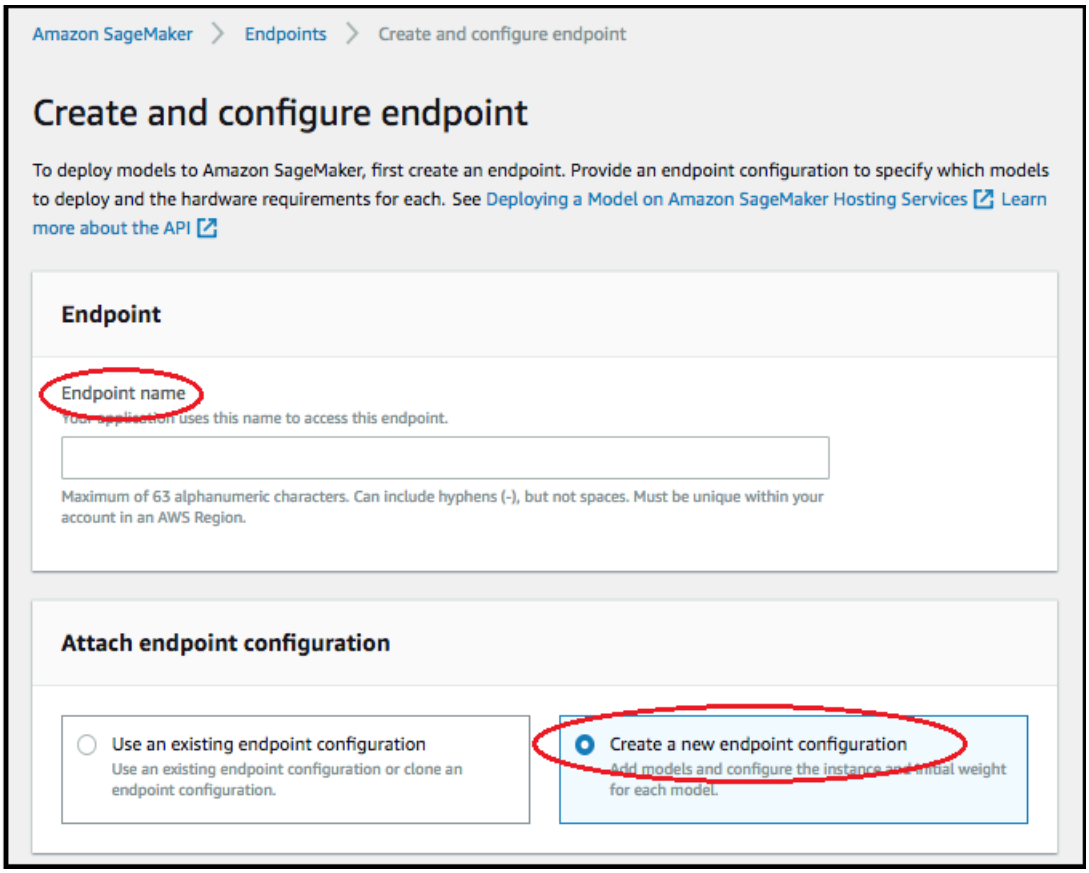
- Jika Anda melatih model Anda menggunakan SageMaker, tentukan variabel lingkunganSAGEMAKER_SUBMIT_DIRECTORYsebagai URI bucket Amazon S3 yang berisi skrip pelatihan.
- Jika Anda tidak melatih model Anda menggunakan SageMaker, tentukan variabel lingkungan berikut:

Key	Nilai untuk MXNet dan PyTorch	Nilai TensorFlow
SAGEMAKER_PROGRAM	inference.py	inference.py
SAGEMAKER_SUBMIT_DIRECTORY	/opt/ml/model/kode	/opt/ml/model/kode
SAGEMAKER_CONTAINER_LOG_LEVEL	20	20
SAGEMAKER_REGION	<your region>	<your region>
MMS_DEFAULT_RESPONSE_TIMEOUT	500	Menetapkan bidang ini untuk TF

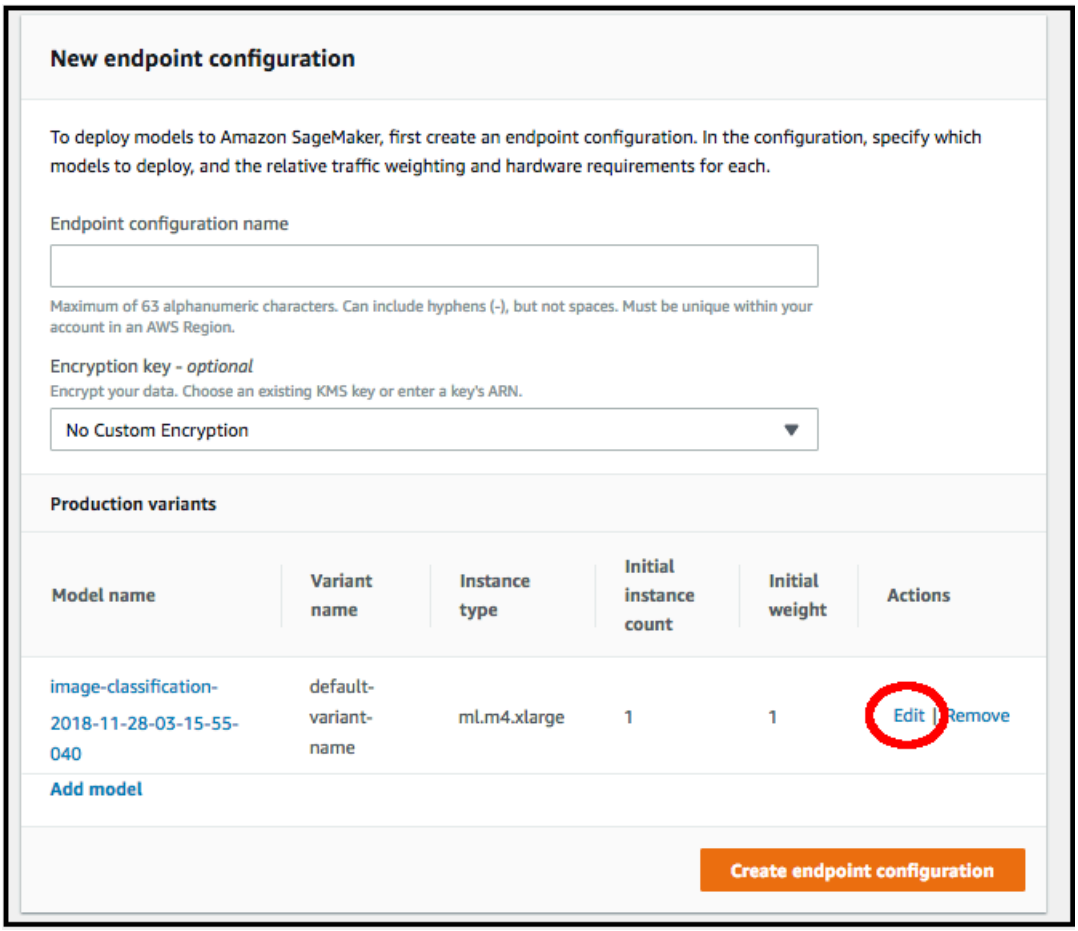
4. Konfirmasikan bahwa informasi untuk wadah akurat, lalu pilihBuat model. PadaBuat halaman arahan model, pilihMenetapkan titik akhir.



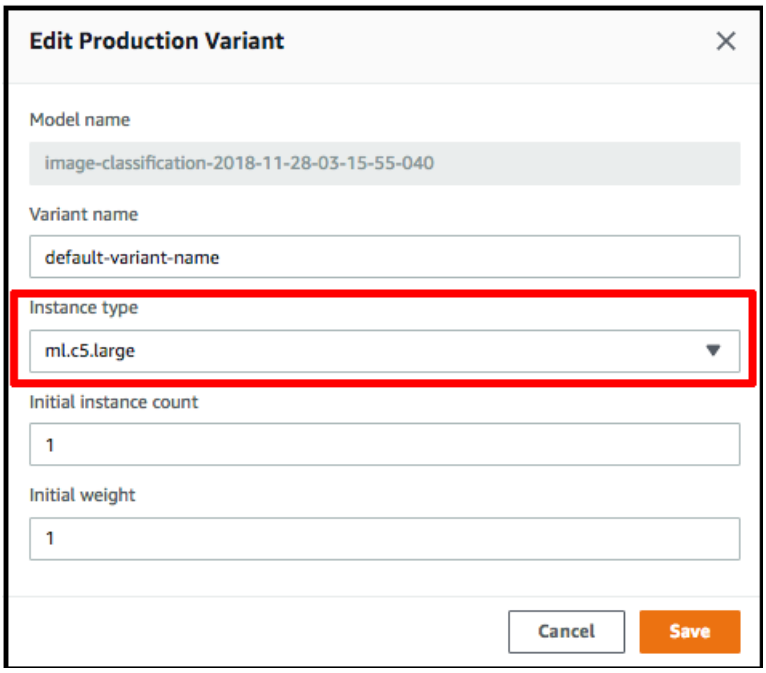
5. DiBuat dan konfigurasikan titik akhirdiagram, tentukanNama titik akhir. UntukLampirkan konfigurasi titik akhir, pilihMembuat konfigurasi endpoint baru.



6. DiKonfigurasi titik akhir baruhalaman, tentukanNama konfigurasi titik akhir.



7. PilihSuntingdi samping nama model dan tentukan yang benarJenis instanspadaEdit Varian Produksihalaman. Sangat penting bahwaJenis instansnilai cocok dengan yang ditentukan dalam pekerjaan kompilasi Anda.



8. Pilih Save (Simpan).
9. PadaKonfigurasi titik akhir baruhalaman, pilihMenetapkan konfigurasi titik akhir, dan kemudian pilihMenetapkan titik akhir.

Minta Inferensi dari Layanan yang Diterahkan

Jika Anda telah mengikuti instruksiMenyebarkan Model, Anda harus memiliki SageMaker endpoint mengatur dan jalankan. Terlepas dari bagaimana Anda menerapkan model NEO-compiled Anda, ada tiga cara Anda dapat mengirimkan permintaan inferensi:

Topik

- Minta Inferensi dari Layanan yang Diterapkan (Amazon SageMaker SDK)
- Minta Inferensi dari Layanan yang Dikerahkan (Boto3)
- Minta Inferensi dari Layanan yang Dikerahkan (AWSCLI)

Minta Inferensi dari Layanan yang Diterapkan (Amazon SageMaker SDK)

Gunakan contoh kode berikut untuk meminta kesimpulan dari layanan yang Anda gunakan berdasarkan kerangka kerja yang Anda gunakan untuk melatih model Anda. Contoh kode untuk kerangka kerja yang berbeda serupa. Perbedaan utamanya adalah TensorFlow memerlukan`application/json`sebagai jenis konten.

PyTorch dan MXNet

Jika Anda menggunakanPyTorch v1.4 atau yang lebih baruatauMXNet 1.7.0 atau yang lebih barudan Anda memiliki Amazon SageMaker titik akhir`InService`, Anda dapat membuat permintaan inferensi menggunakan`predictor`paket SageMaker SDK for Python.

> **Note**
>
> API bervariasi berdasarkan SageMaker Versi SDK for Python:
>
> - Untuk versi 1.x, gunakan`RealTimePredictor`dan`Predict`API.
> - Untuk versi 2.x, gunakan`Predictor`dan`Predict`API.

Contoh kode berikut menunjukkan cara menggunakan API ini untuk mengirim image for inference:

SageMaker Python SDK v1.x

```
from sagemaker.predictor import RealTimePredictor

endpoint = 'insert name of your endpoint here'

# Read image into memory
payload = None
with open("image.jpg", 'rb') as f:
    payload = f.read()

predictor = RealTimePredictor(endpoint=endpoint, content_type='application/x-image')
inference_response = predictor.predict(data=payload)
print (inference_response)
```

SageMaker Python SDK v2.x

```
from sagemaker.predictor import Predictor

endpoint = 'insert name of your endpoint here'

# Read image into memory
payload = None
with open("image.jpg", 'rb') as f:
    payload = f.read()
```

```
predictor = Predictor(endpoint)
inference_response = predictor.predict(data=payload)
print (inference_response)
```

TensorFlow

Contoh kode berikut menunjukkan cara menggunakan SageMaker Python SDK API untuk mengirim gambar untuk inferensi:

```
from sagemaker.predictor import Predictor
from PIL import Image
import numpy as np
import json

endpoint = 'insert the name of your endpoint here'

# Read image into memory
image = Image.open(input_file)
batch_size = 1
image = np.asarray(image.resize((224, 224)))
image = image / 128 - 1
image = np.concatenate([image[np.newaxis, :, :]] * batch_size)
body = json.dumps({"instances": image.tolist()})

predictor = Predictor(endpoint)
inference_response = predictor.predict(data=body)
print(inference_response)
```

Minta Inferensi dari Layanan yang Dikerahkan (Boto3)

Anda dapat mengirimkan permintaan inferensi menggunakan SageMaker SDK for Python (Boto3) client dan`invoke_endpoint()`API setelah Anda memiliki SageMaker titik akhir`InService`. Contoh kode berikut menunjukkan cara mengirim gambar untuk inferensi:

PyTorch and MXNet

```
import boto3

import json

endpoint = 'insert name of your endpoint here'
```

```
runtime = boto3.Session().client('sagemaker-runtime')

# Read image into memory
with open(image, 'rb') as f:
    payload = f.read()
# Send image via InvokeEndpoint API
response = runtime.invoke_endpoint(EndpointName=endpoint, ContentType='application/
x-image', Body=payload)

# Unpack response
result = json.loads(response['Body'].read().decode())
```

TensorFlow

Untuk TensorFlow kirimkan masukan dengan`application/json`untuk jenis konten.

```
from PIL import Image
import numpy as np
import json
import boto3

client = boto3.client('sagemaker-runtime')
input_file = 'path/to/image'
image = Image.open(input_file)
batch_size = 1
image = np.asarray(image.resize((224, 224)))
image = image / 128 - 1
image = np.concatenate([image[np.newaxis, :, :]] * batch_size)
body = json.dumps({"instances": image.tolist()})
ioc_predictor_endpoint_name = 'insert name of your endpoint here'
content_type = 'application/json'
ioc_response = client.invoke_endpoint(
    EndpointName=ioc_predictor_endpoint_name,
    Body=body,
    ContentType=content_type
 )
```

XGBoost

Untuk aplikasi XGBoost, Anda harus mengirimkan teks CSV sebagai gantinya:

```
import boto3
import json
```

```
endpoint = 'insert your endpoint name here'

runtime = boto3.Session().client('sagemaker-runtime')

csv_text = '1,-1.0,1.0,1.5,2.6'
# Send CSV text via InvokeEndpoint API
response = runtime.invoke_endpoint(EndpointName=endpoint, ContentType='text/csv',
 Body=csv_text)
# Unpack response
result = json.loads(response['Body'].read().decode())
```

Perhatikan bahwa BYOM memungkinkan untuk jenis konten kustom. Untuk informasi selengkapnya, lihat `runtime_InvokeEndpoint`.

Minta Inferensi dari Layanan yang Dikerahkan (AWSCLI)

Permintaan inferensi dapat dibuat dengan`sagemaker-runtime invoke-endpoint`setelah Anda memiliki Amazon SageMakertitik akhir`InService`. Anda dapat membuat permintaan inferensi denganAWS Command Line Interface(AWS CLI). Contoh berikut menunjukkan cara mengirim gambar untuk inferensi:

```
aws sagemaker-runtime invoke-endpoint --endpoint-name 'insert name of your endpoint
 here' --body fileb://image.jpg --content-type=application/x-image output_file.txt
```

Sesi`output_file.txt`dengan informasi tentang permintaan inferensi Anda dibuat jika inferensi berhasil.

Untuk TensorFlow kirimkan masukan dengan`application/json`sebagai jenis konten.

```
aws sagemaker-runtime invoke-endpoint --endpoint-name 'insert name of your endpoint
 here' --body fileb://input.json --content-type=application/json output_file.txt
```

Gambar Kontainer

SageMaker Neo sekarang menyediakan informasi URI gambar inferensi untuk`ml_*`target. Untuk informasi lebih lanjut lihatDescribeCompilationJob.

Berdasarkan kasus penggunaan Anda, ganti bagian yang disorot dalam template URI gambar inferensi yang disediakan di bawah ini dengan nilai yang sesuai.

Amazon SageMaker XGBoost

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/xgboost-neo:latest
```

Ganti*aws_account_id*dari tabel di akhir halaman ini berdasarkan*aws_region*Anda menggunakan.

Keras

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-neo-keras:fx_version-
instance_type-py3
```

Ganti*aws_account_id*dari tabel di akhir halaman ini berdasarkan*aws_region*Anda menggunakan.

Ganti*fx_versi*bersama`2.2.4`.

Ganti*instance_type*dengan salah satu`cpu`atau`gpu`.

MXNet

CPU or GPU instance types

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-inference-
mxnet:fx_version-instance_type-py3
```

Ganti*aws_account_id*dari tabel di akhir halaman ini berdasarkan*aws_region*Anda menggunakan.

Ganti*fx_versi*bersama`1.8.0`.

Ganti*instance_type*dengan salah satu`cpu`atau`gpu`.

Inferentia1

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-neo-
mxnet:fx_version-instance_type-py3
```

Ganti*aws_region*dengan salah satu`us-east-1`atau`us-west-2`.

Ganti*aws_account_id*dari tabel di akhir halaman ini berdasarkan*aws_region*Anda menggunakan.

Ganti*fx_versi* bersama `1.5.1`.

Ganti*instance_type* bersama `inf`.

ONNX

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-neo-onnx:fx_version-
instance_type-py3
```

Ganti*aws_account_id* dari tabel di akhir halaman ini berdasarkan*aws_region* Anda menggunakan.

Ganti*fx_versi* bersama `1.5.0`.

Ganti*instance_type* dengan salah satu `cpu` atau `gpu`.

PyTorch

CPU or GPU instance types

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-inference-
pytorch:fx_version-instance_type-py3
```

Ganti*aws_account_id* dari tabel di akhir halaman ini berdasarkan*aws_region* Anda menggunakan.

Ganti*fx_versi* bersama `1.4`, `1.5`, `1.6`, `1.7`, `1.8`, `1.12`, `1.13`, atau `2.0`.

Ganti*instance_type* dengan salah satu `cpu` atau `gpu`.

Inferentia1

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-neo-
pytorch:fx_version-instance_type-py3
```

Ganti*aws_region* dengan salah satu `us-east-1` atau `us-west-2`.

Ganti*aws_account_id* dari tabel di akhir halaman ini berdasarkan*aws_region* Anda menggunakan.

Ganti*fx_versi* bersama `1.5.1`.

Ganti*instance_type*bersamainf.

Inferentia2 and Trainium1

```
763104351884.dkr.ecr.aws_region.amazonaws.com/pytorch-inference-neuronx:1.13.1-
neuronx-py38-sdk2.10.0-ubuntu20.04
```

Ganti*aws_region*bersamaus-east-2untuk Inferensia2, danus-east-1untuk Trainium1.

TensorFlow

CPU or GPU instance types

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-inference-
tensorflow:fx_version-instance_type-py3
```

Ganti*aws_account_id*dari tabel di akhir halaman ini berdasarkan*aws_region*Anda menggunakan.

Ganti*fx_versi*bersama1.15.3atau2.9.

Ganti*instance_type*dengan salah satucpuataugpu.

Inferentia1

```
aws_account_id.dkr.ecr.aws_region.amazonaws.com/sagemaker-neo-
tensorflow:fx_version-instance_type-py3
```

Ganti*aws_account_id*dari tabel di akhir halaman ini berdasarkan*aws_region*Anda menggunakan. Perhatikan bahwa misalnya jenisinfcumaus-east-1danus-west-2didukung.

Ganti*fx_versi*bersama1.15.0

Ganti*instance_type*bersamainf.

Inferentia2 and Trainium1

```
763104351884.dkr.ecr.aws_region.amazonaws.com/tensorflow-inference-neuronx:2.10.1-
neuronx-py38-sdk2.10.0-ubuntu20.04
```

Ganti*aws_region*bersamaus-east-2untuk Inferensia2, danus-east-1untuk Trainium1.

Berikut peta tabel*aws_account_id*bersama*aws_region*. Gunakan tabel ini untuk menemukan URI gambar inferensi yang benar yang Anda butuhkan untuk aplikasi Anda.

aws_account_id	aws_region
785573368785	us-east-1
007439368137	us–east-2
710691900526	us-west-1
301217895009	us-west-2
802834080501	eu-west-1
205493899709	eu-west-2
254080097072	eu-west-3
601324751636	eu-north-1
966458181534	eu-south-1
746233611703	eu-central-1
110948597952	ap-east-1
763008648453	ap-south-1
941853720454	ap-northeast-1
151534178276	ap-northeast-2
925152966179	ap-northeast-3
324986816169	ap-southeast-1
355873309152	ap-southeast-2
474822919863	cn-northwest-1
472730292857	cn-north-1

aws_account_id	aws_region
756306329178	sa-east-1
464438896020	ca-central-1
836785723513	me-south-1
774647643957	af-south-1
275950707576	Il-sentral-1

Perangkat Edge

Amazon SageMaker Neo memberikan dukungan kompilasi untuk kerangka pembelajaran mesin yang populer. Anda dapat menggunakan perangkat tepi yang dikompilasi Neo seperti Raspberry Pi 3, Texas Instruments 'Sitara, Jetson TX1, dan banyak lagi. Untuk daftar kerangka kerja dan perangkat edge yang didukung, lihatKerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukung.

Anda harus mengkonfigurasi perangkat edge Anda sehingga dapat digunakanAWSjasa. Salah satu cara untuk melakukannya adalah dengan menginstal DLR dan Boto3 ke perangkat Anda. Untuk melakukan ini, Andaharus mengatur kredensi otentikasi. LihatBoto3AWSKonfigurasiUntuk informasi lebih lanjut. Setelah model Anda dikompilasi dan perangkat edge Anda dikonfigurasi, Anda dapat mengunduh model dari Amazon S3 ke perangkat edge Anda. Dari sana, Anda bisa menggunakanWaktu Aktif Deep Learning (DLR)untuk membaca model yang dikompilasi dan membuat kesimpulan.

Untuk pengguna pertama kali, kami sarankan Anda memeriksaMemulaipanduan. Panduan ini memandu Anda melalui cara mengatur kredensi Anda, menyusun model, menyebarkan model Anda ke Raspberry Pi 3, dan membuat kesimpulan pada gambar.

Topik

- Kerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukung
- Deploy Model
- Memulai Neo di Perangkat Edge

Kerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukung

Amazon SageMaker Neo mendukung kerangka kerja machine learning umum, perangkat edge, sistem operasi, dan arsitektur chip. Cari tahu apakah Neo mendukung kerangka kerja, perangkat tepi, OS, dan arsitektur chip Anda dengan memilih salah satu topik di bawah ini.

Anda dapat menemukan daftar model yang telah diuji oleh Tim Amazon SageMaker Neo diModel yang Diuji bagian tersebut.

> **Note**
>
> - Perangkat Ambarella memerlukan file tambahan untuk dimasukkan dalam file TAR terkompresi sebelum dikirim untuk kompilasi. Untuk informasi selengkapnya, lihat Memecahkan Masalah Kesalahan Ambarella.
> - TIM-VX (libtim-vx.so) diperlukan untuk i.MX 8M Plus. Untuk informasi tentang cara membangun TIM-VX, lihat GitHub repositori TIM-VX.

Topik

Kerangka Kerja yang Didukung

Amazon SageMaker Neo mendukung kerangka kerja berikut.

Kerangka Kerja	Framework Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
MXNet	1.8	Mendukung 1.8 atau yang lebih baru	Klasifikasi Gambar, Deteksi Objek, Segmentas	Satu file simbol (.json) dan satu file parameter (.params)	GluonCV

Kerangka Kerja	Framework Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
			i Semantik, Estimasi Pose, Pengenalan Aktivitas		
ONNX	1.7	Mendukung 1.7 atau yang lebih baru	Klasifikasi Citra Citra, SVM	Satu file model (.onnx)	
Keras	2.2	Mendukung 2.2 atau yang lebih baru	Klasifikasi Citra	Satu file definisi model (.h5)	
PyTorch	1.7, 1,8	Mendukung 1.7, 1.8 atau sebelumnya	Klasifikasi Gambar, Deteksi Objek	Satu file definisi model (.pth)	
TensorFlow	1.15, 2.4, 2.5 (hanya untuk instans ml.inf1.*)	Mendukung 1.15, 2.4, 2.5 (hanya untuk instans ml.inf1.*) atau sebelumnya	Klasifikasi Gambar, Deteksi Objek	* Untuk model yang disimpan, satu file.pb atau satu file.pbtxt dan direktori variabel yang berisi variabel* Untuk model beku, hanya satu file.pb atau .pbtxt	

Kerangka Kerja	Framework Versi	Versi Model	Model	Format Model (dikemas dalam *.tar.gz)	Toolkit
TensorFlow-Ringan	1.15	Mendukung 1.15 atau yang lebih baru	Klasifikasi Gambar, Deteksi Objek	Satu model definisi flatbuffer file (.tflite)	
XGBoost	1.3	Mendukung 1.3 atau yang lebih baru	Pohon Keputusan	Satu file model XGBoost (.model) di mana jumlah node dalam pohon kurang dari 2 ^ 31	
DARKNET			Klasifikasi Gambar, Deteksi Objek (model Yolo tidak didukung)	Satu file konfigurasi (.cfg) dan satu file bobot (.weights)	

Perangkat yang Didukung, Arsitektur Chip, dan Sistem

Amazon SageMaker Neo mendukung perangkat, arsitektur chip, dan sistem operasi berikut.

Perangkat

Anda dapat memilih perangkat menggunakan daftar dropdown di SageMaker konsol Amazon atau dengan menentukan konfigurasi keluaran `CreateCompilationJob`API.`TargetDevice`

Anda dapat memilih salah satu perangkat tepi berikut:

Daftar Perangkat	Sistem pada Chip (SoC)	Sistem Operasi	Arsitektur	Akselerator	Opsi Compiler Contoh
lorong		Linux	ARM64	Mali	
amba_cv2	CV2	Arch Linux	ARM64	cvflow	
amba_cv22	CV22	Arch Linux	ARM64	cvflow	
amba_cv25	CV25	Arch Linux	ARM64	cvflow	
coreml		iOS, macOS			`{"class_labels": "imagenet_labels_1000.txt"}`
imx8qm	NXP	Linux	ARM64		
imx8mplus	i.MX 8M Ditambah	Linux	ARM64	NPU	
jacinto_tda4vm	TDA4VM	Linux	LENGAN	TDA4VM	
jetson_nano		Linux	ARM64	NVIDIA	`{'gpu-code': 'sm_53', 'trt-ver': '5.0.6', 'cuda-ver': '10.0'}` UntukTensorFlow2 ,`{'JETPACK`

Daftar Perangkat	Sistem pada Chip (SoC)	Sistem Operasi	Arsitektur	Akselerator	Opsi Compiler Contoh
					`_VERSION' : '4.6', 'gpu_code ': 'sm_72'}`
jetson_tx1		Linux	ARM64	NVIDIA	`{'gpu-code': 'sm_53', 'trt-ver': '6.0.1', 'cuda-ver': '10.0'}`
jetson_tx2		Linux	ARM64	NVIDIA	`{'gpu-code': 'sm_62', 'trt-ver': '6.0.1', 'cuda-ver': '10.0'}`

Daftar Perangkat	Sistem pada Chip (SoC)	Sistem Operasi	Arsitektur	Akselerator	Opsi Compiler Contoh
jetson_xavier		Linux	ARM64	NVIDIA	`{'gpu-code': 'sm_72', 'trt-ver': '5.1.6', 'cuda-ver': '10.0'}`
qcs605		Android	ARM64	Mali	`{'ANDROID_PLATFORM': 27}`
qcs603		Android	ARM64	Mali	`{'ANDROID_PLATFORM': 27}`
rasp3b	LENGAN A56	Linux	EABIHF		`{'mattr': ['+neon']}`
rasp4b	LENGAN A72				
rk3288		Linux	EABIHF	Mali	
rk3399		Linux	ARM64	Mali	
sbe_c		Linux	x86_64		`{'mcpu': 'core-avx2'}`
sitara_am57x	AM57X	Linux	ARM64	EVE dan/atau C66x DSP	

Daftar Perangkat	Sistem pada Chip (SoC)	Sistem Operasi	Arsitektur	Akselerator	Opsi Compiler Contoh
x86_win32		Windows 10	X86_32		
x86_win64		Windows 10	X86_32		

Untuk informasi selengkapnya tentang opsi kompiler kunci-nilai JSON untuk setiap perangkat target, lihat`CompilerOptions` bidang di tipe data `OutputConfig`API.

Sistem dan Arsitektur Chip

Tabel pencarian berikut memberikan informasi mengenai sistem operasi dan arsitektur yang tersedia untuk pekerjaan kompilasi model Neo.

Linux

	X86_64	X86	ARM64	EABIHF	EABI
Tidak ada akselerator (CPU)	X		X	X	X
GPU NVIDIA	X		X		
Intel_Graphics	X				
LENGAN Mali			X	X	X

Android

	X86_64	X86	ARM64	EABIHF	EABI
Tidak ada akselerator (CPU)	X	X	X		X
GPU NVIDIA					
Intel_Graphics	X	X			
LENGAN Mali			X		X

Windows

	X86_64	X86	ARM64	EABIHF	EABI
Tidak ada akselerator (CPU)	X	X			

Model yang Diuji

Bagian yang dapat dilipat berikut memberikan informasi tentang model pembelajaran mesin yang diuji oleh tim Amazon SageMaker Neo. Perluas bagian yang dapat dilipat berdasarkan kerangka kerja Anda untuk memeriksa apakah model diuji.

> **Note**
>
> Ini bukan daftar lengkap daftar lengkap Neo.

LihatKerangka Kerja yang Didukung dan SageMaker Neo Operator yang Didukung untuk mengetahui apakah Anda dapat mengkompilasi model Anda dengan SageMaker Neo.

DarkNet

Model	LENGA V8	LENGA Mali	Ambare	Nvidia	Panora	TDA4V	Qualco QCS6(	X86_Li	X86_W ws		
Alexne											
Resnet	X	X		X	X	X		X	X		
YoloV2				X	X	X		X	X		
Yolov2_ cil	X	X		X	X	X		X	X		
Yolov3_ 6				X	X	X		X	X		
Yolov3_ ngil	X	X		X	X	X		X	X		

MXNet

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
Alexnet			X						
Densene 21			X						
DenseNe 01	X	X	X	X	X	X		X	X
GoogLeN	X	X		X	X	X		X	X
Inceptior V3				X	X	X		X	X

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
MobileNe 0,75	X	X		X	X	X			X
MobileNe 1.0	X	X	X	X	X	X			X
MobileNe V2_0.5	X	X		X	X	X			X
MobileNe V2_1.0	X	X	X	X	X	X	X	X	X
MobileNe V3_Besa	X	X	X	X	X	X	X	X	X
MobileNe V3_Kecil	X	X	X	X	X	X	X	X	X
ResNeSt				X	X			X	X
ResNet1 v1	X	X	X	X	X	X			X
ResNet1 v2	X	X		X	X	X			X
ResNet5 v1	X	X	X	X	X	X		X	X
ResNet5 v2	X	X	X	X	X	X		X	X
ResNext 1_32x4d									
ResNext _32x4d	X		X	X	X			X	X

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
Senet_1				X	X	X		X	X
SE_Resl t 50_32x4	X	X		X	X	X		X	X
Squeeze t1.0	X	X	X	X	X	X			X
Squeeze t1.1	X	X	X	X	X	X		X	X
VGG11	X	X	X	X	X			X	X
Xception	X	X	X	X	X	X		X	X
gelapnet 3	X	X		X	X	X		X	X
resnet18 v1b_0.8	X	X		X	X	X			X
resnet50 v1d_0.11	X	X		X	X	X			X
resnet50 v1d_0.86	X	X	X	X	X	X		X	X
ssd_512_ obilenet1 .0_coco	X		X	X	X	X		X	X
ssd_512_ obilenet1 .0_voc	X		X	X	X	X		X	X

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
ssd_resn t50_v1	X		X	X	X			X	X
yolo3_da knet53_c co	X			X	X			X	X
yolo3_m ilenet1.0 _coco	X	X		X	X	X		X	X
deeplab_ esnet50			X						

Keras

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
densene 21	X	X	X	X	X	X		X	X
densene 01	X	X	X	X	X	X			X
inception _v3	X	X		X	X	X		X	X
mobilene _v1	X	X	X	X	X	X		X	X
mobilene _v2	X	X	X	X	X	X		X	X

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
resnet15 _v1				X	X				X
resnet15 _v2				X	X				X
resnet50 v1	X	X	X	X	X			X	X
resnet50 v2	X	X	X	X	X	X		X	X
vgg16			X	X	X			X	X

ONNX

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
alexnet			X						
mobilene v2-1.0	X	X	X	X	X	X		X	X
resnet18 1	X			X	X				X
resnet18 2	X			X	X				X
resnet50 1	X		X	X	X			X	X
resnet50 2	X		X	X	X			X	X

Model	LENGAN V8	LENGAN Mali	Ambarell	Nvidia	Panoram	TDA4VM	Qualcom QCS603	X86_Linu	X86_Windo ws
resnet15 v1				X	X	X			X
resnet15 v2				X	X	X			X
meremas 1	X		X	X	X	X		X	X
vgg19			X						X

PyTorch (FP32)

Model	LENGA V8	LENGA Mali	Ambare	Ambare	Nvidia	Panorar	TDA4VI	Qualcor QCS60	X86_Lin	X86_Windo ws
densene 21	X	X	X	X	X	X	X		X	X
inceptio _v3		X			X	X	X		X	X
resnet1					X	X	X			X
resnet1	X	X			X	X	X			X
resnet5	X	X	X	X	X	X			X	X
pemera 1.0	X	X			X	X	X			X
meremа 1	X	X	X	X	X	X	X		X	X
yolov4					X	X				

Model	LENGA V8	LENGA Mali	Ambare	Ambare	Nvidia	Panorar	TDA4VI	Qualcor QCS60	X86_Lir	X86_Windo ws
yolov5				X	X	X				
fasterrc n_resne 0_fpn					X	X				
maskrcr resnet5 fpn					X	X				

TensorFlow

TensorFlow

Model	LENGAN V8	LENGAN Mali	Ambarell	Ambarell	Nvidia	Panoram	TDA4VM	Qua QC	X86	X86_Windo ws
densenet 01	X	X	X	X	X	X	X		X	X
inception _v3	X	X	X		X	X	X		X	X
mobilene 100_v1	X	X	X		X	X	X			X
mobilene 100_v2.0	X	X	X		X	X	X		X	X
mobilene 130_v2	X	X			X	X	X			X
mobilene 140_v2	X	X	X		X	X	X		X	X

Model	LENGAN V8	LENGAN Mali	Ambarell	Ambarell	Nvidia	Panoram	TDA4VM	Qua QC	X86	X86 ws	Windo
resnet50_ v1.5	X	X			X	X	X		X	X	
resnet50_ v2	X	X	X	X	X	X	X		X	X	
pemeras	X	X	X	X	X	X	X		X	X	
mask_rcr _inceptio n_resnet_ v2					X						
ssd_mob enet_v2					X	X					
faster_rc nn_resne 50_lowpr posals					X						
rfcn_resr et101					X						

TensorFlow.Keras

Model	LENGAN V8	LENGAN Mali	Ambarella	Nvidia	Panorama	TDA4VM	Qua QCS	X86_	X86 ws	Windo
DenseNet 21	X	X		X	X	X		X	X	
DenseNet 01	X	X		X	X	X			X	

Model	LENGAN V8	LENGAN Mali	Ambarella	Nvidia	Panorama	TDA4VM	Qua QCS	X86_	X86_ Windo ws
Inception V3	X	X		X	X	X		X	X
MobileNet	X	X		X	X	X		X	X
MobileNet v2	X	X		X	X	X		X	X
NASNetLa ge				X	X			X	X
NASNetM ile	X	X		X	X	X		X	X
ResNet10				X	X	X			X
ResNet10 V2				X	X	X			X
ResNet15				X	X				X
ResNet15 v2				X	X				X
ResNet50	X	X		X	X			X	X
ResNet50 2	X	X		X	X	X		X	X
VGG16				X	X			X	X
Xception	X	X		X	X	X		X	X

TensorFlow-Ringan

TensorFlow-Lite (FP32)

Model	LENGA V8	LENGA Mali	Ambare	Nvidia	Panora	TDA4V	Qualco QCS60	X86_Lir	X86_Wi ws	i.MX 8M Ditambah
densen 2018_0 7	X			X	X	X			X	
inceptic _resnet 2_2018 _27				X	X	X			X	
inceptic _v3_20 04_27				X	X	X			X	X
inceptic _v4_20 04_27				X	X	X			X	X
mnasne .5_224_ _07_20	X			X	X	X			X	
mnasne .0_224_ _07_20	X			X	X	X			X	
mnasne .3_224_ _07_20	X			X	X	X			X	

Model	LENGA V8	LENGA Mali	Ambare	Nvidia	Panora	TDA4V	Qualco QCS60	X86_Lir	X86_Wi ws	i.MX 8M Ditambah
mobiler _v1_0.2 128	X			X	X	X			X	X
mobiler _v1_0.2 224	X			X	X	X			X	X
mobiler _v1_0.5 28	X			X	X	X			X	X
mobiler _v1_0.5 24	X			X	X	X			X	X
mobiler _v1_0.7 128	X			X	X	X			X	X
mobiler _v1_0.7 224	X			X	X	X			X	X
mobiler _v1_1.0 28	X			X	X	X			X	X
mobiler _v1_1.0 92	X			X	X	X			X	X
mobiler _v2_1.0 24	X			X	X	X			X	X

Model	LENGA V8	LENGA Mali	Ambare	Nvidia	Panora	TDA4VI	Qualcor QCS60	X86_Lir	X86_Wi ws	i.MX 8M Ditambah
resnet_ _101				X	X	X			X	
squeez t_2018_ _27	X			X	X	X			X	

TensorFlow-Lite (INT8)

Model	LENGA V8	LENGA Mali	Ambare	Nvidia	Panora	TDA4VI	Qualcor QCS60	X86_Lir	X86_Wi ws	i.MX 8M Ditambah
inceptic _v1							X			X
inceptic _v2							X			X
inceptic _v3	X					X	X		X	X
inceptic _v4_29	X					X	X		X	X
mobiler _v1_0.2 128	X					X			X	X
mobiler _v1_0.2 224	X					X			X	X

Model	LENGA V8	LENGA Mali	Ambare	Nvidia	Panora	TDA4VI	Qualcor QCS60	X86_Lir	X86_Wi ws	i.MX 8M Ditambah
mobiler _v1_0.5 28	X					X			X	X
mobiler _v1_0.5 24	X					X			X	X
mobiler _v1_0.7 128	X					X			X	X
mobiler _v1_0.7 224	X					X	X		X	X
mobiler _v1_1.0 28	X					X			X	X
mobiler _v1_1.0 24	X					X	X		X	X
mobiler _v2_1.0 24	X					X	X		X	X
dalam- v3_ 513							X			

Deploy Model

Anda dapat menerapkan modul komputasi ke perangkat edge yang dibatasi sumber daya dengan: mengunduh model yang dikompilasi dari Amazon S3 ke perangkat Anda dan menggunakanDLR, atau Anda dapat menggunakanAWSIoT Greengrass.

Sebelum melanjutkan, pastikan perangkat edge Anda harus didukung oleh SageMaker Neo. Lihat,Kerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukunguntuk mengetahui perangkat tepi apa yang didukung. Pastikan bahwa Anda menentukan perangkat tepi target Anda ketika Anda mengirimkan pekerjaan kompilasi, lihatGunakan Neo untuk Mengkompilasi Model.

Menerapkan Model Terkompilasi (DLR)

DLRadalah runtime umum yang ringkas untuk model pembelajaran mendalam dan model pohon keputusan. DLR menggunakanTVMwaktu aktif,Treeliteruntime, NVIDIA TensorRT™, dan dapat menyertakan waktu operasi khusus perangkat keras lainnya. DLR menyediakan API Python/C ++ terpadu untuk memuat dan menjalankan model yang dikompilasi di berbagai perangkat.

Anda dapat menginstal rilis terbaru dari paket DLR menggunakan perintah pip berikut:

```
pip install dlr
```

Untuk pemasangan DLR pada target GPU atau perangkat tepi non-x86, silakan lihatRilisuntuk biner prebuilt, atauMenginstal DLRuntuk membangun DLR dari sumber. Misalnya, untuk menginstal DLR untuk Raspberry Pi 3, Anda dapat menggunakan:

```
pip install https://neo-ai-dlr-release.s3-us-west-2.amazonaws.com/v1.3.0/pi-armv7l-
raspbian4.14.71-glibc2_24-libstdcpp3_4/dlr-1.3.0-py3-none-any.whl
```

Menyebarkan Model (AWSIoT Greengrass)

AWSIoT Greengrassmemperluas kemampuan cloud ke perangkat lokal. Hal ini memungkinkan perangkat untuk mengumpulkan dan menganalisis data lebih dekat ke sumber informasi, bereaksi secara mandiri terhadap acara lokal, dan berkomunikasi secara aman satu sama lain di jaringan lokal. DenganAWSIoT Greengrass, Anda dapat melakukan inferensi machine learning di edge pada data yang dihasilkan secara lokal dengan menggunakan model yang terlatih cloud. Saat ini, Anda dapat menerapkan model ke semuaAWSPerangkat IoT Greengrass berbasis prosesor seri ARM Cortex-A, Intel Atom, dan Nvidia Jetson. Untuk informasi lebih lanjut tentang penerapan aplikasi inferensi Lambda untuk melakukan inferensi machine learning denganAWSIoT Greengrass,

lihatCara mengonfigurasi kesimpulan machine learning yang dioptimalkan menggunakanAWSKonsol Manajemen.

Memulai Neo di Perangkat Edge

Panduan ini untuk memulai dengan Amazon SageMaker Neo menunjukkan cara mengompilasi model, menyiapkan perangkat, dan membuat kesimpulan di perangkat. Sebagian besar contoh kode menggunakan Boto3. Kami menyediakan perintah menggunakanAWS CLImana yang berlaku, serta petunjuk tentang cara untuk memenuhi prasyarat untuk Neo.

> **Note**
>
> Anda dapat menjalankan cuplikan kode berikut di komputer lokal Anda, dalam SageMaker notebook, dalam SageMaker Studio, atau (tergantung pada perangkat edge Anda) pada perangkat edge Anda. Penyiapannya serupa; namun, ada dua pengecualian utama jika Anda menjalankan panduan ini di dalam SageMaker contoh notebook atau SageMaker Sesi Studio:
>
> - Anda tidak perlu menginstal Boto3.
> - Anda tidak perlu menambahkan`'AmazonSageMakerFullAccess'`Kebijakan IAM

Panduan ini mengasumsikan Anda menjalankan instruksi berikut di perangkat edge Anda.

Prasyarat

1. Menginstal Boto3

   Jika Anda menjalankan perintah ini di perangkat edge Anda, Anda harus menginstalAWS SDK for Python (Boto3). Dalam lingkungan Python (sebaiknya lingkungan virtual), jalankan yang berikut secara lokal di terminal perangkat edge Anda atau dalam instance notebook Jupyter:

   Terminal

   ```
   pip install boto3
   ```

   Jupyter Notebook

   ```
   !pip install boto3
   ```

2. MengaturAWSKredensial

Anda perlu menyiapkan kredensi Amazon Web Services di perangkat Anda untuk menjalankan SDK for Python (Boto3). Secara default,AWSkredensi harus disimpan dalam file`~/.aws/credentials`pada perangkat tepi Anda. Di dalam file kredensil, Anda akan melihat dua variabel lingkungan:`aws_access_key_id`dan`aws_secret_access_key`.

Di terminal Anda, jalankan:

```
$ more ~/.aws/credentials

[default]
aws_access_key_id = YOUR_ACCESS_KEY
aws_secret_access_key = YOUR_SECRET_KEY
```

KlasterAWSPanduan Referensi Ummemiliki petunjuk tentang cara untuk mendapatkan`aws_access_key_id`dan`aws_secret_access_key`. Untuk informasi lebih lanjut tentang cara mengatur kredensi di perangkat Anda, lihatBoto3dokumentasi.

3. Siapkan Peran IAM dan lampirkan kebijakan.

Neo memerlukan akses ke URI bucket S3 Anda. Buat peran IAM yang dapat dijalankan SageMakerdan memiliki izin untuk mengakses URI S3. Anda dapat membuat IAM role baik dengan menggunakan SDK for Python (Boto3), konsol, atauAWS CLI. Contoh berikut menggambarkan cara membuat IAM role menggunakan SDK for Python (Boto3):

```
import boto3

AWS_REGION = 'aws-region'

# Create an IAM client to interact with IAM
iam_client = boto3.client('iam', region_name=AWS_REGION)
role_name = 'role-name'
```

Untuk informasi lebih lanjut tentang cara membuat IAM role dengan konsol,AWS CLI, atau melaluiAWSAPI, lihatMembuat pengguna IAM diAWSakun.

Buat kamus yang menjelaskan kebijakan IAM yang Anda lampirkan. Kebijakan ini digunakan untuk membuat peran IAM baru.

```
policy = {
    'Statement': [
```

```
        {
            'Action': 'sts:AssumeRole',
            'Effect': 'Allow',
            'Principal': {'Service': 'sagemaker.amazonaws.com'},
        }],
     'Version': '2012-10-17'
}
```

Buat peran IAM baru menggunakan kebijakan yang Anda tetapkan di atas:

```
import json

new_role = iam_client.create_role(
    AssumeRolePolicyDocument=json.dumps(policy),
    Path='/',
    RoleName=role_name
)
```

Anda perlu tahu apa Amazon Resource Name (ARN) Anda ketika Anda membuat pekerjaan kompilasi di langkah selanjutnya, jadi simpan dalam variabel juga.

```
role_arn = new_role['Role']['Arn']
```

Sekarang setelah Anda membuat peran baru, lampirkan izin yang dibutuhkan untuk berinteraksi dengan Amazon SageMaker dan Amazon S3:

```
iam_client.attach_role_policy(
    RoleName=role_name,
    PolicyArn='arn:aws:iam::aws:policy/AmazonSageMakerFullAccess'
)

iam_client.attach_role_policy(
    RoleName=role_name,
    PolicyArn='arn:aws:iam::aws:policy/AmazonS3FullAccess'
);
```

4. Buat bucket Amazon S3 untuk menyimpan artefak model Anda

   SageMaker Neo akan mengakses artefak model Anda dari Amazon S3

Boto3

```
# Create an S3 client
s3_client = boto3.client('s3', region_name=AWS_REGION)

# Name buckets
bucket='name-of-your-bucket'

# Check if bucket exists
if boto3.resource('s3').Bucket(bucket) not in
 boto3.resource('s3').buckets.all():
    s3_client.create_bucket(
        Bucket=bucket,
        CreateBucketConfiguration={
            'LocationConstraint': AWS_REGION
        }
    )
else:
    print(f'Bucket {bucket} already exists. No action needed.')
```

CLI

```
aws s3 mb s3://'name-of-your-bucket' --region specify-your-region

# Check your bucket exists
aws s3 ls s3://'name-of-your-bucket'/
```

5. Latih model machine learning

LihatLatih Model dengan Amazon SageMakeruntuk informasi lebih lanjut tentang cara melatih model machine learning menggunakan Amazon SageMaker. Anda dapat mengunggah model terlatih lokal secara opsional langsung ke bucket URI Amazon S3.

> **Note**
>
> Pastikan model diformat dengan benar tergantung pada kerangka kerja yang Anda gunakan. LihatApa bentuk data masukan tidak SageMaker Neo harapkan?

Jika Anda belum memiliki model, gunakan`curl`perintah untuk mendapatkan salinan lokal`coco_ssd_mobilenet`model dari TensorFlowSitus web. Model yang baru saja Anda salin adalah model deteksi objek yang dilatih dariSet data COCO. Ketik yang berikut ini ke dalam buku catatan Jupyter Anda:

```
model_zip_filename = './coco_ssd_mobilenet_v1_1.0.zip'
!curl http://storage.googleapis.com/download.tensorflow.org/models/tflite/
coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip \
    --output {model_zip_filename}
```

Perhatikan bahwa contoh khusus ini telah dipaketkan dalam file .zip. Unzip file ini dan paket ulang sebagai tarfile terkompresi (`.tar.gz`) sebelum menggunakannya di langkah selanjutnya. Ketik yang berikut ini ke dalam buku catatan Jupyter Anda:

```
# Extract model from zip file
!unzip -u {model_zip_filename}

model_filename = 'detect.tflite'
model_name = model_filename.split('.')[0]

# Compress model into .tar.gz so SageMaker Neo can use it
model_tar = model_name + '.tar.gz'
!tar -czf {model_tar} {model_filename}
```

6. Unggah model terlatih ke bucket S3

   Setelah Anda melatih mode pembelajaran mesin, simpan dalam bucket S3.

   Boto3

   ```
   # Upload model
   s3_client.upload_file(Filename=model_filename, Bucket=bucket,
    Key=model_filename)
   ```

   CLI

   Ganti`your-model-filename`dan`your-S3-bucket`dengan nama ember S3 Anda.

   ```
   aws s3 cp your-model-filename s3://your-S3-bucket
   ```

Langkah 1: Kompilasi Model

Setelah Anda puasPrasyarat, Anda dapat mengkompilasi model Anda dengan Amazon SageMakerNeo. Anda dapat mengkompilasi model Anda menggunakanAWS CLI, konsol atauAmazon Web Services SDK for Python (Boto3), LihatGunakan Neo untuk Mengkompilasi Model. Dalam contoh ini, Anda akan mengkompilasi model Anda dengan Boto3.

Untuk mengkompilasi model, SageMaker Neo memerlukan informasi berikut:

1. URI bucket Amazon S3 tempat Anda menyimpan model terlatih.

   Jika Anda mengikuti prasyarat, nama bucket Anda disimpan dalam variabel bernama`bucket`. Potongan kode berikut menunjukkan cara membuat daftar semua bucket Anda menggunakanAWS CLI:

   ```
   aws s3 ls
   ```

   Misalnya:

   ```
   $ aws s3 ls
   2020-11-02 17:08:50 bucket
   ```

2. URI bucket Amazon S3 tempat Anda ingin menyimpan model yang dikompilasi.

   Potongan kode di bawah menggabungkan URI bucket Amazon S3 dengan nama direktori keluaran yang disebut`output`:

   ```
   s3_output_location = f's3://{bucket}/output'
   ```

3. Kerangka pembelajaran mesin yang Anda gunakan untuk melatih model Anda.

   Tentukan kerangka kerja yang Anda gunakan untuk melatih model Anda.

   ```
   framework = 'framework-name'
   ```

   Misalnya, jika Anda ingin mengkompilasi model yang dilatih menggunakan TensorFlow, Anda dapat menggunakan`tflite`atau`tensorflow`. Gunakan`tflite`jika Anda ingin menggunakan versi yang lebih ringan TensorFlow yang menggunakan lebih sedikit memori penyimpanan.

   ```
   framework = 'tflite'
   ```

Untuk daftar lengkap kerangka kerja yang didukung Neo, lihatKerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukung.

4. Bentuk input model Anda.

   Neo membutuhkan nama dan bentuk tensor input Anda. Nama dan bentuk diteruskan sebagai pasangan nilai kunci.`value`adalah daftar dimensi integer dari tensor input dan`key`adalah nama yang tepat dari tensor input dalam model.

   ```
   data_shape = '{"name": [tensor-shape]}'
   ```

   Misalnya:

   ```
   data_shape = '{"normalized_input_image_tensor":[1, 300, 300, 3]}'
   ```

   > **Note**
   >
   > Pastikan model diformat dengan benar tergantung pada kerangka kerja yang Anda gunakan. LihatApa bentuk data masukan tidak SageMaker Neo harapkan? Kunci dalam kamus ini harus diubah menjadi nama tensor masukan baru.

5. Baik nama perangkat target yang akan dikompilasi atau detail umum platform perangkat keras

   ```
   target_device = 'target-device-name'
   ```

   Misalnya, jika Anda ingin menyebarkan ke Raspberry Pi 3, gunakan:

   ```
   target_device = 'rasp3b'
   ```

   Anda dapat menemukan seluruh daftar perangkat edge yang didukung diKerangka Kerja, Perangkat, Sistem, dan Arsitektur yang Didukung.

Sekarang setelah Anda menyelesaikan langkah-langkah sebelumnya, Anda dapat mengirimkan pekerjaan kompilasi ke Neo.

```
# Create a SageMaker client so you can submit a compilation job
sagemaker_client = boto3.client('sagemaker', region_name=AWS_REGION)
```

```
# Give your compilation job a name
compilation_job_name = 'getting-started-demo'
print(f'Compilation job for {compilation_job_name} started')

response = sagemaker_client.create_compilation_job(
    CompilationJobName=compilation_job_name,
    RoleArn=role_arn,
    InputConfig={
        'S3Uri': s3_input_location,
        'DataInputConfig': data_shape,
        'Framework': framework.upper()
    },
    OutputConfig={
        'S3OutputLocation': s3_output_location,
        'TargetDevice': target_device
    },
    StoppingCondition={
        'MaxRuntimeInSeconds': 900
    }
)

# Optional - Poll every 30 sec to check completion status
import time

while True:
    response =
 sagemaker_client.describe_compilation_job(CompilationJobName=compilation_job_name)
    if response['CompilationJobStatus'] == 'COMPLETED':
        break
    elif response['CompilationJobStatus'] == 'FAILED':
        raise RuntimeError('Compilation failed')
    print('Compiling ...')
    time.sleep(30)
print('Done!')
```

Jika Anda ingin informasi tambahan untuk debugging, termasuk pernyataan cetak berikut:

```
print(response)
```

Jika pekerjaan kompilasi berhasil, model yang dikompilasi disimpan dalam bucket Amazon S3 keluaran yang Anda tentukan sebelumnya (`s3_output_location`). Unduh model yang Anda kompilasi secara lokal:

```
object_path = f'output/{model}-{target_device}.tar.gz'
neo_compiled_model = f'compiled-{model}.tar.gz'
s3_client.download_file(bucket, object_path, neo_compiled_model)
```

Langkah 2: Siapkan Perangkat Anda

Anda perlu menginstal paket di perangkat edge Anda sehingga perangkat Anda dapat membuat kesimpulan. Anda juga perlu menginstalAWSIoT Greengrassinti atauWaktu Aktif Deep Learning (DLR). Dalam contoh ini, Anda akan menginstal paket yang diperlukan untuk membuat kesimpulan untuk`coco_ssd_mobilenet`algoritma deteksi objek dan Anda akan menggunakan DLR.

1. Menginstal paket tambahan

   Selain Boto3, Anda harus menginstal pustaka tertentu di perangkat edge Anda. Perpustakaan apa yang Anda instal bergantung pada kasus penggunaan Anda.

   Misalnya, untuk`coco_ssd_mobilenet`algoritma deteksi objek yang Anda download sebelumnya, Anda perlu menginstalNumPyuntuk manipulasi data dan statistik,PILuntuk memuat gambar, danMatplotlibuntuk menghasilkan plot. Anda juga memerlukan salinan TensorFlow jika Anda ingin mengukur dampak kompilasi dengan Neo versus garis dasar.

   ```
   !pip3 install numpy pillow tensorflow matplotlib
   ```

2. Instal mesin inferensi di perangkat Anda

   Untuk menjalankan model NEO-compiled Anda, instalWaktu Aktif Deep Learning (DLR)di perangkat Anda. DLR adalah runtime umum yang ringkas untuk model pembelajaran mendalam dan model pohon keputusan. Pada target CPU x86_64 yang menjalankan Linux, Anda dapat menginstal rilis terbaru paket DLR menggunakan yang berikut`pip`perintah:

   ```
   !pip install dlr
   ```

   Untuk pemasangan DLR pada target GPU atau perangkat tepi non-x86, lihatRilisuntuk biner prebuilt, atauMenginstal DLRuntuk membangun DLR dari sumber. Misalnya, untuk menginstal DLR untuk Raspberry Pi 3, Anda dapat menggunakan:

   ```
   !pip install https://neo-ai-dlr-release.s3-us-west-2.amazonaws.com/v1.3.0/pi-
   armv7l-raspbian4.14.71-glibc2_24-libstdcpp3_4/dlr-1.3.0-py3-none-any.whl
   ```

Langkah 3: Membuat Inferensi di Perangkat Anda

Dalam contoh ini, Anda akan menggunakan Boto3 untuk mengunduh output pekerjaan kompilasi Anda ke perangkat edge Anda. Anda kemudian akan mengimpor DLR, mengunduh contoh gambar dari dataset, mengubah ukuran gambar ini agar sesuai dengan input asli model, dan kemudian Anda akan membuat prediksi.

1. Unduh model yang dikompilasi dari Amazon S3 ke perangkat Anda dan ekstrak dari tarfile terkompresi.

```
# Download compiled model locally to edge device
object_path = f'output/{model_name}-{target_device}.tar.gz'
neo_compiled_model = f'compiled-{model_name}.tar.gz'
s3_client.download_file(bucket_name, object_path, neo_compiled_model)

# Extract model from .tar.gz so DLR can use it
!mkdir ./dlr_model # make a directory to store your model (optional)
!tar -xzvf ./compiled-detect.tar.gz --directory ./dlr_model
```

2. Impor DLR dan diinisialisasi**`DLRModel`**objek.

```
import dlr

device = 'cpu'
model = dlr.DLRModel('./dlr_model', device)
```

3. Unduh gambar untuk inferensi dan format berdasarkan bagaimana model Anda dilatih.

   Untuk`coco_ssd_mobilenet`contoh, Anda dapat men-download gambar dariSet data COCOdan kemudian mereformasi gambar menjadi`300x300`:

```
from PIL import Image

# Download an image for model to make a prediction
input_image_filename = './input_image.jpg'
!curl https://farm9.staticflickr.com/8325/8077197378_79efb4805e_z.jpg --output
 {input_image_filename}

# Format image so model can make predictions
resized_image = image.resize((300, 300))

# Model is quantized, so convert the image to uint8
```

```
x = np.array(resized_image).astype('uint8')
```

4. Gunakan DLR untuk membuat kesimpulan.

   Terakhir, Anda dapat menggunakan DLR untuk membuat prediksi pada gambar yang baru saja Anda unduh:

   ```
   out = model.run(x)
   ```

Untuk contoh selengkapnya menggunakan DLR untuk membuat kesimpulan dari model yang dikompilasi NEO pada perangkat edge, lihatneo-ai-dlr Repositori Github.

Memecahkan Masalah Kesalahan

Bagian ini berisi informasi tentang cara memahami dan mencegah kesalahan umum, pesan kesalahan yang mereka hasilkan, dan panduan tentang cara mengatasi kesalahan ini. Sebelum melanjutkan, tanyakan pada diri Anda pertanyaan-pertanyaan berikut:

Apakah Anda mengalami kesalahan sebelum menerapkan model Anda? Jika ya, lihatMemecahkan Masalah Kesalahan Kompilasi Neo.

Apakah Anda mengalami kesalahan setelah mengkompilasi model Anda? Jika ya, lihatMemecahkan Masalah Kesalahan Inferensi Neo.

Apakah Anda mengalami kesalahan saat mencoba mengkompilasi model Anda untuk perangkat Ambarella? Jika ya, lihatMemecahkan Masalah Kesalahan Ambarella.

Jenis Klasifikasi Kesalahan

Daftar ini mengklasifikasikankesalahan penggunaAnda dapat menerima dari Neo. Ini termasuk kesalahan akses dan izin dan kesalahan pemuatan untuk setiap kerangka kerja yang didukung. Semua kesalahan lainnya adalahkesalahan sistem.

Kesalahan izin klien

Neo melewati kesalahan untuk ini langsung dari layanan dependen.

- Akses Ditolaksaat memanggil sts:AssumeRole
- Setiap 400kesalahan saat memanggil Amazon S3 untuk mengunduh atau mengunggah model klien

- PassRole kesalahan

Kesalahan beban

Dengan asumsi bahwa kompiler Neo berhasil memuat .tar.gz dari Amazon S3, periksa apakah tarball berisi file yang diperlukan untuk kompilasi. Kriteria pemeriksaan khusus kerangka kerja:

- TensorFlow: Hanya mengharapkan file protobuf (*.pb atau*.pbtxt). Untuk model yang disimpan, mengharapkan satu folder variabel.
- Pytorch: Harapkan hanya satu file pytorch (*.pth).
- MXNET: Harapkan hanya satu file simbol (*.json) dan satu file parameter (*.params).
- XGBoost: Harapkan hanya satu file model XGBoost (*.model). Model input memiliki batasan ukuran.

Kesalahan kompilasi

Dengan asumsi bahwa kompiler Neo berhasil memuat .tar.gz dari Amazon S3, dan tarball berisi file yang diperlukan untuk kompilasi. Kriteria pemeriksaan adalah:

- OperatorNotImplemented: Operator belum diimplementasikan.
- OperatorAttributeNotImplemented: Atribut di operator yang ditentukan belum diimplementasikan.
- OperatorAttributeRequired: Atribut diperlukan untuk grafik simbol internal, tetapi tidak tercantum dalam grafik model input pengguna.
- OperatorAttributeValueNotValid: Nilai atribut di operator tertentu tidak valid.

Topik

- Memecahkan Masalah Kesalahan Kompilasi Neo
- Memecahkan Masalah Kesalahan Inferensi Neo
- Memecahkan Masalah Kesalahan Ambarella

Memecahkan Masalah Kesalahan Kompilasi Neo

Bagian ini berisi informasi tentang cara memahami dan mencegah kesalahan kompilasi umum, pesan kesalahan yang mereka hasilkan, dan panduan tentang cara mengatasi kesalahan ini.

Topik

Cara Menggunakan Halaman Ini

Mencoba untuk menyelesaikan kesalahan Anda dengan melalui bagian-bagian ini dalam urutan berikut:

1. Periksa apakah input pekerjaan kompilasi Anda memenuhi persyaratan input. Lihat Bentuk data input apa yang diharapkan SageMaker Neo?
2. Cek umumkesalahan khusus kerangka kerja.
3. Periksa apakah kesalahan Anda adalahkesalahan infrastruktur.
4. Cek Andalog kompilasi.

Kesalahan Terkait Kerangka Kerja

Keras

Kesalahan	Solusi
`InputConfiguration: No h5 file provided in <model path>`	Periksa file h5 Anda ada di Amazon S3 URI yang Anda tentukan. Atau Periksa bahwafile h5 diformat dengan benar.
`InputConfiguration: Multiple h5 files provided, <model path>, when only one is allowed`	Periksa bahwa Anda hanya menyediakan satuh5berkas.

Kesalahan	Solusi
`ClientError: InputConfiguration: Unable to load provided Keras model. Error: 'sample_weight_mode'`	Periksa versi Keras yang Anda tentukan didukung. Lihat, kerangka kerja yang didukung untukcontoh clouddanperangkat tepi.
`ClientError: InputConfiguration: Input input has wrong shape in Input Shape dictionary. Input shapes should be provided in NCHW format.`	Periksa apakah input model Anda mengikuti format NCHW. LihatApa yang dilakukan bentuk data input SageMaker Neo berharap?

MXNet

Kesalahan	Solusi
`ClientError: InputConfiguration: Only one parameter file is allowed for MXNet model. Please make sure the framework you select is correct.`	SageMaker Neo akan memilih file parameter pertama yang diberikan untuk kompilasi.

TensorFlow

Kesalahan	Solusi
`InputConfiguration: Exactly one .pb file is allowed for TensorFlow models.`	Pastikan Anda hanya menyediakan satu file.pb atau .pbtxt.
`InputConfiguration: Exactly one .pb or .pbtxt file is allowed for TensorFlow models.`	Pastikan Anda hanya menyediakan satu file.pb atau .pbtxt.

Kesalahan	Solusi
`ClientError: InputConfiguration: TVM cannot convert <model zoo> model. Please make sure the framework you selected is correct. The following operators are not implemented: {<operator name>}`	Periksa operator yang Anda pilih didukung. LihatSageMaker Kerangka Kerja dan Operator yang Didukung Neo.

PyTorch

Kesalahan	Solusi
`InputConfiguration: We are unable to extract DataInputConfig from the model due to` *`input_config_derivation_error`* `. Please override by providing a DataInputConfig during compilation job creation.`	Lakukan salah satu dari langkah berikut: • Tentukan nama dan bentuk input yang diharapkan dengan memberikan`DataInput Config` definisi dalam permintaan kompilasi Anda. • Selidiki kesalahan di Amazon CloudWatch Log. Cek`/aws/sagemaker/ CompilationJobs` log group dan cari log stream bernama*`compilati onJobName`* `/model-in fo-extraction` .

Kesalahan Terkait Infrastruktur

Kesalahan	Solusi
`ClientError: InputConfiguration: S3 object does not exist. Bucket: <bucket>, Key: <bucket key>`	Periksa URI Amazon S3 yang Anda berikan.
`ClientError: InputConfiguration: Bucket <bucket name> is in region <region name> which is different from AWS Sagemaker service region <service region>`	Buat bucket Amazon S3 yang berada di wilayah yang sama dengan layanan.
`ClientError: InputConfiguration: Unable to untar input model. Please confirm the model is a tar.gz file`	Periksa apakah model Anda di Amazon S3 dikompresi menjadi `atar.gz`berkas.

Periksa log kompilasi Anda

1. Navigasikan ke Amazon CloudWatch padahttps://console.aws.amazon.com/cloudwatch/.
2. Pilih wilayah tempat Anda membuat pekerjaan kompilasi dariWilayahdaftar dropdown di kanan atas.
3. Di panel navigasi Amazon CloudWatch, pilihLog. PilihGrup log.
4. Cari grup log yang disebut`/aws/sagemaker/CompilationJobs`. Pilih grup log .
5. Cari logstream yang dinamai setelah nama pekerjaan kompilasi. Pilih aliran log.

Memecahkan Masalah Kesalahan Inferensi Neo

Bagian ini berisi informasi tentang cara mencegah dan mengatasi beberapa kesalahan umum yang mungkin Anda temui saat menerapkan dan/atau memanggil titik akhir. Bagian ini berlaku untukPyTorch 1.4.0 atau lebih barudanMXNet v1.7.0 atau lebih baru.

- Pastikan inferensi pertama (inferensi pemanasan) pada data input yang valid dilakukan di`model_fn()`, jika Anda mendefinisikan`model_fn`dalam skrip inferensi Anda, jika tidak, pesan kesalahan berikut dapat dilihat di terminal saat`predict API`disebut:

```
An error occurred (ModelError) when calling the InvokeEndpoint operation: Received
 server error (0) from <users-sagemaker-endpoint> with message "Your invocation timed
 out while waiting for a response from container model. Review the latency metrics
 for each container in Amazon CloudWatch, resolve the issue, and try again."
```

- Pastikan bahwa variabel lingkungan dalam tabel berikut diatur. Jika tidak disetel, pesan galat berikut mungkin muncul:

  Di terminal:

  ```
  An error occurred (ModelError) when calling the InvokeEndpoint operation: Received
   server error (503) from <users-sagemaker-endpoint> with message "{ "code": 503,
   "type": "InternalServerException", "message": "Prediction failed" } ".
  ```

  Di CloudWatch:

  ```
  W-9001-model-stdout com.amazonaws.ml.mms.wlm.WorkerLifeCycle - AttributeError:
   'NoneType' object has no attribute 'transform'
  ```

Kunci	Nilai
SAGEMAKER_PROGRAM	inference.py
SAGEMAKER_SUBMIT_DIRECTORY	/opt/ml/model/kode
SAGEMAKER_CONTAINER_LOG_LEVEL	20
SAGEMAKER_REGION	<your region>

- Pastikan bahwa`MMS_DEFAULT_RESPONSE_TIMEOUT`variabel lingkungan diatur ke 500 atau nilai yang lebih tinggi saat membuat Amazon SageMaker model; jika tidak, pesan kesalahan berikut dapat dilihat di terminal:

  ```
  An error occurred (ModelError) when calling the InvokeEndpoint operation: Received
   server error (0) from <users-sagemaker-endpoint> with message "Your invocation timed
   out while waiting for a response from container model. Review the latency metrics
   for each container in Amazon CloudWatch, resolve the issue, and try again."
  ```

Memecahkan Masalah Kesalahan Ambarella

SageMaker Neo membutuhkan model untuk dikemas dalam file TAR terkompresi (`*.tar.gz`). Perangkat Ambarella memerlukan file tambahan untuk dimasukkan dalam file TAR terkompresi sebelum dikirim untuk kompilasi. Sertakan file berikut dalam file TAR terkompresi Anda jika Anda ingin mengkompilasi model untuk target Ambarella dengan SageMaker Neo:

- Model terlatih menggunakan kerangka kerja yang didukung oleh SageMaker Neo
- File konfigurasi JSON
- Gambar kalibrasi

Misalnya, file TAR Anda akan terlihat serupa dengan contoh berikut:

```
###amba_config.json
###calib_data
|    ### data1
|    ### data2
|    ### .
|    ### .
|    ### .
|    ### data500
###mobilenet_v1_1.0_0224_frozen.pb
```

Direktori dikonfigurasi sebagai berikut:

- `amba_config.json`: File konfigurasi
- `calib_data`: Folder berisi gambar kalibrasi
- `mobilenet_v1_1.0_0224_frozen.pb`: TensorFlow model disimpan sebagai grafik beku

Untuk informasi tentang kerangka kerja yang didukung oleh SageMaker Neo, lihatKerangka Kerja yang Didukung.

Menyiapkan File Konfigurasi

File konfigurasi menyediakan informasi yang diperlukan oleh rantai alat Ambarella untuk mengkompilasi model. File konfigurasi harus disimpan sebagai file JSON dan nama file harus diakhiri dengan`*config.json`. Bagan berikut menunjukkan isi file konfigurasi.

Key	Deskripsi	Contoh
masukan	Kamus pemetaan lapisan masukan untuk atribut.	`{inputs:{"data":{...},"data1":{...}}}`
“data”	Nama lapisan masukan. Catatan: “data” adalah contoh nama yang dapat Anda gunakan untuk memberi label pada lapisan input.	“data”
bentuk	Menjelaskan bentuk input ke model. Ini mengikuti konvensi yang sama yang SageMaker Neo menggunakan.	“bentuk”: “1,3,224,224"
filepath	Jalur relatif ke direktori yang berisi gambar kalibrasi. Ini bisa berupa file biner atau gambar seperti JPG atau PNG.	“filepath”: “calib_data/”
format warna	Format warna yang diharapkan model. Ini akan digunakan saat mengonversi gambar menjadi biner. Nilai yang didukung: [RGB, BGR]. Defaultnya adalah RGB.	“format warna” :"RGB”
kejam	Nilai rata-rata yang akan dikurangi dari input. Bisa berupa nilai tunggal atau daftar nilai. Ketika mean diberikan sebagai daftar, jumlah entri harus sesuai dengan dimensi saluran input.	“berarti” :128.0

Key	Deskripsi	Contoh
skala	Nilai skala yang akan digunakan untuk menormalkan input. Bisa berupa nilai tunggal atau daftar nilai. Ketika skala diberikan sebagai daftar, jumlah entri harus sesuai dengan dimensi saluran input.	“skala”: 255.0

Berikut adalah file konfigurasi contoh:

```
{
    "inputs": {
        "data": {
                "shape": "1, 3, 224, 224",
                "filepath": "calib_data/",
                "colorformat": "RGB",
                "mean":[128,128,128],
                "scale":[128.0,128.0,128.0]
        }
    }
}
```

Gambar Kalibrasi

Kuantisasi model terlatih Anda dengan memberikan gambar kalibrasi. Mengukur model Anda meningkatkan kinerja mesin CVFlow pada Sistem Ambarella pada Chip (SoC). Rantai alat Ambarella menggunakan gambar kalibrasi untuk menentukan bagaimana setiap lapisan dalam model harus dikuantisasi untuk mencapai kinerja dan akurasi yang optimal. Setiap lapisan dikuantisasi secara independen ke format INT8 atau INT16. Model akhir memiliki campuran lapisan INT8 dan INT16 setelah kuantisasi.

Berapa banyak gambar yang harus Anda gunakan?

Disarankan agar Anda menyertakan antara 100-200 gambar yang mewakili jenis adegan yang diharapkan ditangani oleh model. Waktu kompilasi model meningkat secara linier ke jumlah gambar kalibrasi dalam file input.

Apa format gambar yang direkomendasikan?

Gambar kalibrasi dapat dalam format biner mentah atau format gambar seperti JPG dan PNG.

Folder kalibrasi Anda dapat berisi campuran gambar dan file biner. Jika folder kalibrasi berisi gambar dan file biner, rantai alat pertama-tama mengonversi gambar menjadi file biner. Setelah konversi selesai, ia menggunakan file biner yang baru dihasilkan bersama dengan file biner yang awalnya ada di folder.

Bisakah saya mengonversi gambar menjadi format biner terlebih dahulu?

Ya. Anda dapat mengonversi gambar ke format biner dengan paket sumber terbuka seperti OpenCV atau PIL. Pangkas dan ubah ukuran gambar sehingga memenuhi lapisan input model terlatih Anda.

Mean dan Skala

Anda dapat menentukan opsi pra-pemrosesan rata-rata dan penskalaan ke rantai alat Amberalla. Operasi ini tertanam ke dalam jaringan dan diterapkan selama inferensi pada setiap input. Jangan berikan data yang diproses jika Anda menentukan mean atau skala. Lebih khusus lagi, jangan berikan data yang telah Anda kurangi dari rata-rata atau terapkan penskalaan.

Periksa log kompilasi Anda

Untuk informasi tentang memeriksa log kompilasi untuk perangkat Ambarella, lihat Periksa log kompilasi Anda.

Gunakan Amazon SageMaker Elastic Inference (EI)

Mulai 15 April 2023, AWS tidak akan memasukkan pelanggan baru ke Amazon Elastic Inference (EI), dan akan membantu pelanggan saat ini memigrasikan beban kerja mereka ke opsi yang menawarkan harga dan performa yang lebih baik. Setelah 15 April 2023, pelanggan baru tidak akan dapat meluncurkan instans dengan akselerator Amazon EI di Amazon, Amazon ECS, atau SageMaker Amazon EC2.

Machine learning (ML) on AWS membantu Anda berinovasi lebih cepat dengan rangkaian layanan dan infrastruktur ML terlengkap yang tersedia dalam model as-you-go penggunaan berbayar berbiaya rendah. AWSterus memberikan infrastruktur berkinerja lebih baik dan biaya lebih rendah untuk beban kerja inferensi ML. AWSmeluncurkan Amazon Elastic Inference (EI) pada tahun 2018 untuk memungkinkan pelanggan melampirkan akselerasi bertenaga GPU berbiaya rendah ke Amazon EC2, SageMaker instans Amazon, atau Amazon Elastic Container Service (ECS) untuk

mengurangi biaya menjalankan inferensi pembelajaran mendalam hingga 75% dibandingkan dengan instans berbasis GPU mandiri seperti Amazon EC2 P4d dan Amazon EC2 G5. Pada tahun 2019, AWS meluncurkan AWS Inferentia, silikon khusus pertama Amazon yang dirancang untuk mempercepat beban kerja pembelajaran mendalam dengan memberikan inferensi kinerja tinggi di cloud. Instans Amazon EC2 Inf1 berdasarkan chip AWS Inferentia menghasilkan throughput 2,3x lebih tinggi dan biaya per inferensi hingga 70% lebih rendah daripada instans Amazon EC2 berbasis GPU generasi saat ini yang sebanding. Dengan ketersediaan opsi komputasi baru yang dipercepat seperti instans AWS Inferentia dan Amazon EC2 G5, manfaat melampirkan GPU fraksional ke instans host CPU menggunakan Amazon EI telah berkurang. Misalnya, pelanggan yang menghosting model di Amazon EI yang pindah ke `ml.inf1.xlarge` instans bisa mendapatkan penghematan biaya hingga 56% dan peningkatan kinerja 2x.

Pelanggan dapat menggunakan Amazon SageMaker Inference Recommender untuk membantu mereka memilih instans alternatif terbaik untuk Amazon EI untuk menerapkan model ML-nya.

Pertanyaan yang sering diajukan

1. Mengapa Amazon mendorong pelanggan untuk memindahkan beban kerja dari Amazon Elastic Inference (EI) ke opsi akselerasi perangkat keras yang lebih baru seperti Inferentia? AWS

   Pelanggan mendapatkan kinerja yang lebih baik dengan harga yang jauh lebih baik daripada Amazon EI dengan opsi akselerator perangkat keras baru seperti AWSInferentia untuk beban kerja inferensi mereka. AWS Inferentia dirancang untuk memberikan inferensi kinerja tinggi di cloud, untuk menurunkan total biaya inferensi, dan untuk memudahkan pengembang mengintegrasikan pembelajaran mesin ke dalam aplikasi bisnis mereka. Untuk memungkinkan pelanggan mendapatkan keuntungan dari akselerator perangkat keras generasi baru tersebut, kami tidak akan memasukkan pelanggan baru ke Amazon EI setelah 15 April 2023.

2. AWSLayanan apa yang terpengaruh oleh langkah untuk menghentikan orientasi pelanggan baru ke Amazon Elastic Inference (EI)?

   Pengumuman ini akan memengaruhi akselerator Amazon EI yang melekat pada tugas Amazon EC2, instans SageMaker Amazon, atau Amazon Elastic Container Service (ECS). Di Amazon SageMaker, ini berlaku untuk titik akhir dan kernel notebook menggunakan akselerator Amazon EI.

3. Apakah saya dapat membuat akselerator Amazon Elastic Inference (EI) baru setelah 15 April 2023?

   Tidak, jika Anda adalah pelanggan baru dan belum menggunakan Amazon EI dalam 30 hari terakhir, maka Anda tidak akan dapat membuat instans Amazon EI baru di AWS akun Anda

setelah 15 April 2023. Namun, jika Anda telah menggunakan akselerator Amazon EI setidaknya sekali dalam 30 hari terakhir, Anda dapat memasang akselerator Amazon EI baru ke instans Anda.

4. Bagaimana cara mengevaluasi opsi instans alternatif untuk Titik Akhir SageMaker Inferensi Amazon saya saat ini?

   Amazon SageMaker Inference Recommender dapat membantu Anda mengidentifikasi penerapan hemat biaya untuk memigrasikan beban kerja yang ada dari Amazon Elastic Inference (EI) ke instans ML yang sesuai yang didukung oleh. SageMaker

5. Bagaimana cara mengubah jenis instans untuk titik akhir saya yang ada di Amazon? SageMaker

   Anda dapat mengubah jenis instans untuk titik akhir yang ada dengan melakukan hal berikut:

   1. Pertama, buat yang baru EndpointConfig yang menggunakan tipe instance baru. Jika Anda memiliki kebijakan penskalaan otomatis, hapus kebijakan penskalaan otomatis yang ada.
   2. Panggil UpdateEndpointsambil menentukan yang baru Anda buat EndpointConfig.
   3. Tunggu hingga titik akhir Anda mengubah status menjadi`InService`. Ini akan memakan waktu sekitar 10-15 menit.
   4. Terakhir, jika Anda memerlukan penskalaan otomatis untuk titik akhir baru Anda, buat kebijakan penskalaan otomatis baru untuk titik akhir baru ini dan. ProductionVariant

6. Bagaimana cara mengubah jenis instans untuk Instans SageMaker Notebook Amazon yang sudah ada menggunakan Amazon Elastic Inference (EI)?

   Pilih instance Notebook di SageMaker konsol, lalu pilih Instans Notebook yang ingin Anda perbarui. Pastikan Instance Notebook memiliki `Stopped` status. Terakhir, Anda dapat memilih Edit dan mengubah jenis instans Anda. Pastikan bahwa, ketika Instans Notebook Anda dimulai, Anda memilih kernel yang tepat untuk instans baru Anda.

7. Apakah ada jenis instance tertentu yang merupakan alternatif yang baik untuk Amazon Elastic Inference (EI)?

   Setiap beban kerja pembelajaran mesin adalah unik. Sebaiknya gunakan Amazon SageMaker Inference Recommender untuk membantu Anda mengidentifikasi jenis instans yang tepat untuk beban kerja, persyaratan kinerja, dan anggaran Anda. AWS Inferentia, khususnya`inf1.xlarge`, adalah alternatif kinerja tinggi dan biaya rendah terbaik untuk pelanggan Amazon EI.

Bermigrasi dari Amazon Elastic Inference ke instans lain

Informasi berikut dapat membantu Anda memigrasikan titik akhir yang SageMaker di-host dari instans yang menggunakan akselerator Amazon Elastic Inference ke instans lain. Saran bervariasi tergantung pada kerangka kerja Anda.

PyTorch

Jika Anda bermigrasi dari PyTorch, gunakan panduan berikut.

1. Pilih jenis instans yang tepat

Setiap beban kerja pembelajaran mesin adalah unik. Sebaiknya gunakan Amazon SageMaker Inference Recommender untuk membantu Anda mengidentifikasi jenis instans yang tepat untuk beban kerja, persyaratan kinerja, dan anggaran Anda. AWS Inferentia, khususnya`inf1.xlarge`, adalah alternatif kinerja tinggi dan biaya rendah terbaik untuk pelanggan Amazon Elastic Inference.

Dalam pengujian beban kami dengan Inference Recommender, `g4dn.xlarge` instans berkinerja lebih baik daripada `m5.large` instance dengan lampiran. `eia.2large` Dengan Amazon Elastic Inference, Anda harus membayar biaya tambahan instans ML tempat akselerator terpasang. Amazon Elastic Inference juga hanya mendukung PyTorch 1,5 dan TensorFlow 2,3. Jika Anda bermigrasi ke `ml.g4dn` instance, Anda dapat menggunakan versi terbaru PyTorch 1.11 dan 2.9. TensorFlow Selain itu, `ml.g4dn` dan AWS Inferensia tersedia di semua AWS Wilayah, sedangkan Amazon Elastic Inference hanya tersedia di 6 Wilayah. Baik AWS Inferentia dan `ml.g4dn` menawarkan kinerja yang lebih baik dengan harga lebih rendah untuk sebagian besar beban kerja inferensi ML.

2. Memodifikasi **`inference.py`**

Ubah `inference.py` file Anda untuk menghapus perubahan yang diperlukan khusus Inferensi Elastis dan gunakan penangan default. Berdasarkan kasus pengguna yang berbeda, Anda mungkin memiliki penangan input dan output yang berbeda, tetapi perubahan utama yang harus Anda lakukan ada dalam fungsi `model_fn` handler pemuatan model dan. `predict_fn` Hapus pengendali prediksi khusus Inferensi Elastis `predict_fn` dan kembalikan handler pemuatan model ke format `model_fn` default. Contoh berikut menunjukkan bagaimana melakukan ini, dengan bagian-bagian yang harus Anda hapus dari `inference.py` komentar:

```
from __future__ import print_function

import os
```

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import numpy as np

def model_fn(model_dir, context):
    model = {customer_model}
    # if torch.__version__ in VERSIONS_USE_NEW_API:
        # import torcheia
        # loaded_model = loaded_model.eval()
        # loaded_model = torcheia.jit.attach_eia(loaded_model, 0)
    with open(os.path.join(model_dir, 'model.pth'), 'rb') as f:
        model.load_state_dict(torch.load(f))
    return model


# def predict_fn(input_data, model):
#     logger.info(
#         "Performing EIA inference with Torch JIT context with input of size
 {}".format(
#             input_data.shape
#         )
#     )
#     device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
#     input_data = input_data.to(device)
#     with torch.no_grad():
#        if torch.__version__ in VERSIONS_USE_NEW_API:
#             import torcheia
#
#             torch._C._jit_set_profiling_executor(False)
#             with torch.jit.optimized_execution(True):
#                 return model.forward(input_data)
#         else:
#             with torch.jit.optimized_execution(True, {"target_device": "eia:0"}):
#                 return model(input_data)

def predict_fn(input_data, model):
    return model(input_data)
```

3. Buat model

Buat model baru yang menunjuk ke `inference.py` file Anda yang dimodifikasi. Anda dapat menyimpan `inference.py` file secara lokal dan mengarahkannya dengan menentukan

`source_dir` dan `entry_point` atau memasukkan `inference.py` file ke dalam model tarball. Contoh berikut menunjukkan kasus sebelumnya:

```
from sagemaker.pytorch import PyTorchModel

pytorch = PyTorchModel(
    model_data={model_data_url},
    role=role,
    entry_point="inference.py",
    source_dir="code",
    framework_version="1.5.1",
    py_version="py3",
    sagemaker_session=sagemaker_session,
)
```

4. Terapkan model ke titik akhir dan panggil

Anda dapat menggunakan salah satu opsi berikut untuk menerapkan model Anda setelah membuat perubahan sebelumnya.

Opsi 1: Menyebarkan dari awal

Anda dapat menerapkan model ke titik akhir baru dengan instance yang direkomendasikan dari kategori Accelerated Computing, seperti G4.

```
predictor = pytorch.deploy(
        ...
        # instance_type = "ml.c5.xlarge",
        instance_type="ml.g4dn.2xlarge",
        ...
response = predictor.predict(payload)
```

Opsi 2: Perbarui titik akhir yang ada

Selesaikan langkah-langkah berikut untuk memperbarui titik akhir yang ada:

1. Panggilan `CreateEndpointConfig` untuk membuat baru `EndpointConfig` yang menggunakan jenis instance baru. Jika Anda memiliki kebijakan penskalaan otomatis, hapus kebijakan penskalaan otomatis yang ada.

   ```
   endpoint_config_response = sagemaker_client.create_endpoint_config(
       EndpointConfigName=endpoint_config_name,
   ```

```
    ProductionVariants=[
        {
            "VariantName": "variant1", # The name of the production variant.
            "ModelName": model_name, # The name of new created model
            "InstanceType": instance_type, # Specify the right-sized instance type.
            "InitialInstanceCount": 1 # Number of instances to launch initially.
        }
    ]
)
```

2. Panggil `UpdateEndpoint` dan tentukan yang baru Anda buat`EndpointConfig`.

```
endpoint_config_response = sagemaker_client.update_endpoint(
    EndpointConfigName=endpoint_config_name, # The name of the new endpoint config
 just created
    EndpointName=endpoint_name # The name of the existing endpoint you want to
 update
 )
```

3. Tunggu hingga titik akhir Anda mengubah status menjadi`InService`. Ini membutuhkan waktu sekitar 10-15 menit.
4. Terakhir, jika Anda memerlukan penskalaan otomatis untuk titik akhir baru Anda, buat kebijakan penskalaan otomatis baru untuk titik akhir baru Anda dan. `ProductionVariant`

TensorFlow

Jika Anda bermigrasi dari TensorFlow, gunakan panduan berikut.

1. Pilih jenis instans yang tepat

Lihat 1. Pilih panduan jenis instans yang tepat di PyTorch bagian ini.

2. Terapkan model ke titik akhir dan panggil

Anda dapat menggunakan salah satu opsi berikut untuk menerapkan model Anda.

Opsi 1: Menyebarkan dari awal

Anda dapat bermigrasi dari Elastic Inference dengan menerapkan ulang model ke titik akhir baru dengan menghapus **`accelerator_type`** bidang dan menentukan jenis instance berukuran tepat dari kategori Accelerated Computing, seperti G4. Dalam contoh berikut, baris komentar menyebabkan Anda menerapkan tanpa menggunakan akselerator Elastic Inference.

```
predictor = tensorflow_model.deploy(
    ...
    instance_type="ml.g4dn.2xlarge"
    # instance_type="ml.c5.xlarge",
    # accelerator_type="ml.eia1.medium"
    ...
)
```

Opsi 2: Perbarui titik akhir yang ada

Lihat Opsi 2. Perbarui panduan titik akhir yang ada di Langkah 4 PyTorch bagian.

MXNet

Jika Anda bermigrasi dari MxNet, gunakan panduan berikut.

1. Pilih jenis instans yang tepat

Lihat 1. Pilih panduan jenis instans yang tepat di PyTorch bagian ini.

2. Terapkan model ke titik akhir dan panggil

Anda dapat menggunakan salah satu opsi berikut untuk menerapkan model Anda.

Opsi 1: Menyebarkan dari awal

Anda dapat bermigrasi dari Elastic Inference dengan menerapkan ulang model ke titik akhir baru dengan menghapus **`accelerator_type`** bidang dan menentukan jenis instance berukuran tepat dari kategori Accelerated Computing, seperti G4. Dalam contoh berikut, baris komentar menyebabkan Anda menerapkan tanpa menggunakan akselerator Elastic Inference.

```
predictor = mxnet_model.deploy(
        ...
        # instance_type="ml.c5.xlarge",
        instance_type="ml.g4dn.2xlarge"
        ...
)
```

Opsi 2: Perbarui titik akhir yang ada

Lihat Opsi 2: Perbarui panduan titik akhir yang ada di Langkah 4 PyTorch bagian.

Topik

Bagaimana EI Bekerja

Akselerator Amazon Elastic Inference adalah perangkat terpasang jaringan yang bekerja bersama SageMaker instans di titik akhir Anda untuk mempercepat panggilan inferensi Anda. Elastic Inference mempercepat inferensi dengan memungkinkan Anda memasang GPU fraksional ke instance apa pun. SageMaker Anda dapat memilih instance klien untuk menjalankan aplikasi Anda dan melampirkan akselerator Elastic Inference untuk menggunakan jumlah akselerasi GPU yang tepat untuk kebutuhan inferensi Anda. Elastic Inference membantu Anda menurunkan biaya saat tidak sepenuhnya memanfaatkan instans GPU Anda untuk inferensi. Sebaiknya coba Elastic Inference dengan model Anda menggunakan instans CPU dan ukuran akselerator yang berbeda.

Jenis akselerator EI berikut tersedia. Anda dapat mengonfigurasi titik akhir atau instance notebook dengan jenis akselerator EI apa pun.

Dalam tabel, throughput dalam teraflops (TFLOPS) terdaftar untuk operasi floating-point presisi tunggal (F32) dan floating-point setengah presisi (F16). Memori dalam GB juga terdaftar.

Jenis Akselerator	Throughput F32 di TFLOPS	Throughput F16 di TFLOPS	Memori dalam GB
ml.eia2.sedang	1	8	2
ml.eia2.large	2	16	4
ml.eia2.xlarge	4	32	8

Jenis Akselerator	Throughput F32 di TFLOPS	Throughput F16 di TFLOPS	Memori dalam GB
ml.eia1.sedang	1	8	1
ml.eia1.large	2	16	2
ml.eia1.xlarge	4	32	4

Pilih Jenis Akselerator EI

Pertimbangkan faktor-faktor berikut saat memilih jenis akselerator untuk model yang dihosting:

- Model, tensor input, dan ukuran batch memengaruhi jumlah memori akselerator yang Anda butuhkan. Mulailah dengan jenis akselerator yang menyediakan setidaknya memori sebanyak ukuran file model terlatih Anda. Faktor bahwa model mungkin menggunakan memori secara signifikan lebih banyak daripada ukuran file saat runtime.
- Tuntutan pada sumber daya komputasi CPU, memori sistem utama, dan akselerasi berbasis GPU dan memori akselerator bervariasi secara signifikan antara berbagai jenis model pembelajaran mendalam. Persyaratan latensi dan throughput aplikasi juga menentukan jumlah komputasi dan akselerasi yang Anda butuhkan. Uji secara menyeluruh berbagai konfigurasi tipe instans dan ukuran akselerator EI untuk memastikan Anda memilih konfigurasi yang paling sesuai dengan kebutuhan kinerja aplikasi Anda.

Untuk informasi selengkapnya tentang memilih akselerator EI, lihat:

- Ikhtisar Amazon Elastic Inference
- Memilih Instance dan Jenis Akselerator untuk Model Anda
- Mengoptimalkan biaya di Amazon Elastic Inference dengan TensorFlow

Menggunakan EI dalam Instance SageMaker Notebook

Biasanya, Anda membuat dan menguji model pembelajaran mesin di buku SageMaker catatan sebelum Anda menerapkannya untuk produksi. Anda dapat melampirkan EI ke instance notebook saat membuat instance notebook. Anda dapat menyiapkan titik akhir yang di-host secara lokal pada instance notebook dengan menggunakan mode lokal yang didukung oleh, TensorFlow MXNet, serta

PyTorch estimator serta model di Amazon SageMaker Python SDK untuk menguji kinerja inferensi. Elastic Inference diaktifkan saat PyTorch ini tidak didukung pada instance notebook. Untuk petunjuk tentang cara melampirkan EI ke instance notebook dan menyiapkan titik akhir lokal untuk inferensi, lihat. Lampirkan EI ke Instance Notebook Ada juga kernel SageMaker Notebook Jupyter yang mendukung Inference Elastic untuk versi Elastic Inference-enabled dan Apache MXNet. TensorFlow Untuk informasi tentang menggunakan instance SageMaker notebook, lihat Menggunakan Instans SageMaker Notebook Amazon

Gunakan EI pada Endpoint yang Dihosting

Ketika Anda siap untuk menerapkan model Anda untuk produksi untuk memberikan kesimpulan, Anda membuat titik akhir yang SageMaker dihosting. Anda dapat melampirkan EI ke instance tempat titik akhir Anda di-host untuk meningkatkan kinerjanya dalam memberikan kesimpulan. Untuk petunjuk tentang cara melampirkan EI ke instance endpoint yang dihosting, lihatGunakan EI di Titik Akhir yang SageMaker Dihosting Amazon.

Framework yang Mendukung EI

Amazon Elastic Inference dirancang untuk digunakan dengan versi Apache MXNet TensorFlow, atau kerangka kerja pembelajaran mesin yang AWS disempurnakan. PyTorch Versi kerangka kerja yang disempurnakan ini secara otomatis dibangun ke dalam wadah saat Anda menggunakan Amazon SageMaker Python SDK, atau Anda dapat mengunduhnya sebagai file biner dan mengimpornya di wadah Docker Anda sendiri.

Anda dapat mengunduh file TensorFlow biner berkemampuan EI dari bucket Amazon S3 amazonei-tensorflow publik ke wadah penyajian. TensorFlow Untuk informasi selengkapnya tentang membuat container yang menggunakan versi EI-enabled TensorFlow, lihat Amazon Elastic TensorFlow Inference with in. SageMaker

Anda dapat mengunduh file biner MXNet berkemampuan EI dari bucket Amazon S3 amazonei-apachemxnet publik ke wadah penyajian MXNet. Untuk informasi selengkapnya tentang membuat container yang menggunakan MxNet versi EI-enabled, lihat Amazon Elastic Inference with MxNet in. SageMaker

Anda dapat mengunduh biner yang diaktifkan Elastic Inference untuk. PyTorch Untuk informasi selengkapnya tentang membuat container yang menggunakan versi EI-enabled PyTorch, lihat Amazon Elastic PyTorch Inference with in. SageMaker

Untuk menggunakan Elastic Inference di endpoint yang di-host, Anda dapat memilih salah satu kerangka kerja berikut tergantung pada kebutuhan Anda.

- SageMaker Python SDK - Menyebarkan model TensorFlow
- SageMaker Python SDK - Menyebarkan model MXNet
- SageMaker Python SDK - Menyebarkan model PyTorch

Jika Anda perlu membuat wadah khusus untuk menerapkan model Anda yang kompleks dan memerlukan ekstensi ke kerangka kerja yang tidak didukung oleh kontainer SageMaker pra-bangun, gunakan AWSSDK tingkat rendah untuk Python (Boto 3).

Gunakan EI dengan Algoritma SageMaker Built-in

Saat ini, algoritma Klasifikasi Gambar - MXNet dan Deteksi built-in mendukung EI. Untuk contoh yang menggunakan algoritma Klasifikasi Gambar dengan EI, lihat Contoh Klasifikasi Gambar Multiclass End-to-End.

Contoh Notebook EI

Contoh notebook berikut memberikan contoh penggunaan EI di SageMaker:

- Menggunakan Amazon Elastic Inference dengan MXNet di Amazon SageMaker
- Menggunakan Amazon Elastic Inference dengan MXNet di Instans Notebook Amazon SageMaker
- Menggunakan Amazon Elastic Inference dengan model yang dikompilasi TensorFlow NEO SageMaker
- Menggunakan Amazon Elastic Inference dengan model Serving yang sudah terlatih TensorFlow SageMaker

Siapkan untuk Menggunakan EI

Gunakan instruksi dalam topik ini hanya jika salah satu dari berikut ini berlaku untuk Anda:

- Anda ingin menggunakan peran atau kebijakan izin yang disesuaikan.
- Anda ingin menggunakan VPC untuk model yang dihosting atau instance notebook Anda.

> **Note**
>
> Jika Anda sudah memiliki peran eksekusi yang memiliki kebijakan `AmazonSageMakerFullAccess` terkelola yang dilampirkan (ini berlaku untuk setiap peran

IAM yang Anda buat saat membuat instance notebook, tugas pelatihan, atau model di konsol) dan Anda tidak tersambung ke model EI atau instance notebook di VPC, Anda tidak perlu membuat perubahan ini untuk menggunakan EI di Amazon. SageMaker

Topik

Mengatur Izin yang Diperlukan

Untuk menggunakan EI SageMaker, peran yang Anda gunakan untuk membuka instance notebook atau membuat model deployable harus memiliki kebijakan dengan izin yang diperlukan terlampir. Anda dapat melampirkan kebijakan `AmazonSageMakerFullAccess` terkelola, yang berisi izin yang diperlukan, ke peran, atau Anda dapat menambahkan kebijakan khusus yang memiliki izin yang diperlukan. Untuk informasi tentang membuat peran IAM, lihat Membuat Peran untuk AWS Layanan (Konsol) di Panduan AWS Identity and Access Management Pengguna. Untuk informasi tentang melampirkan kebijakan ke peran, lihat Menambahkan dan Menghapus Kebijakan IAM.

Tambahkan izin ini secara khusus untuk menghubungkan EI dalam kebijakan IAM.

```
{
    "Effect": "Allow",
    "Action": [
        "elastic-inference:Connect",
        "ec2:DescribeVpcEndpoints"
    ],
    "Resource": "*"
}
```

Kebijakan IAM berikut adalah daftar lengkap izin yang diperlukan untuk menggunakan EI di. SageMaker

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
                "elastic-inference:Connect",
                "ec2:DescribeVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "cloudwatch:PutMetricData",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DeleteAlarms",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeleteScheduledAction",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:DescribeScheduledActions",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:PutScheduledAction",
                "application-autoscaling:RegisterScalableTarget",
                "logs:CreateLogGroup",
```

```
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "s3:ExistingObjectTag/SageMaker": "true"
                }
            }
        },
        {
            "Action": "iam:CreateServiceLinkedRole",
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/aws-service-role/sagemaker.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "sagemaker.application-
autoscaling.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        }
    ]
}
```

Gunakan VPC Kustom untuk Connect ke EI

Untuk menggunakan EI dengan SageMaker VPC, Anda perlu membuat dan mengonfigurasi dua grup keamanan, dan menyiapkan titik akhir antarmuka PrivateLink VPC. EI menggunakan titik akhir antarmuka VPC untuk berkomunikasi dengan titik SageMaker akhir di VPC Anda. Grup keamanan yang Anda buat digunakan untuk terhubung ke titik akhir antarmuka VPC.

Mengatur Grup Keamanan untuk Connect ke EI

Untuk menggunakan EI dalam VPC, Anda perlu membuat dua grup keamanan:

- Grup keamanan untuk mengontrol akses ke titik akhir antarmuka VPC yang akan Anda atur untuk EI.
- Grup keamanan yang memungkinkan SageMaker untuk memanggil ke grup keamanan pertama.

Untuk mengkonfigurasi dua grup keamanan

1. Buat grup keamanan tanpa koneksi keluar. Anda akan melampirkan ini ke antarmuka titik akhir VPC yang Anda buat di bagian berikutnya.
2. Buat grup keamanan kedua tanpa koneksi masuk, tetapi dengan koneksi keluar ke grup keamanan pertama.
3. Edit grup keamanan pertama untuk mengizinkan koneksi masuk hanya ke grup keamanan kedua dan semua koneksi keluar.

Untuk informasi selengkapnya tentang grup keamanan VPC, lihat Grup Keamanan untuk VPC Anda di Panduan Pengguna Amazon Virtual Private Cloud.

Siapkan Endpoint Antarmuka VPC untuk Connect ke EI

Untuk menggunakan EI dengan SageMaker VPC kustom, Anda perlu menyiapkan titik akhir antarmuka VPC (PrivateLink) untuk layanan EI.

- Siapkan titik akhir antarmuka VPC (PrivateLink) untuk EI. Ikuti petunjuk di Membuat Endpoint Antarmuka. Dalam daftar layanan, pilih com.amazonaws. <region>.elastic-inference.runtime. Untuk grup Keamanan, pastikan Anda memilih grup keamanan pertama yang Anda buat di bagian sebelumnya ke titik akhir.
- Saat Anda mengatur titik akhir antarmuka, pilih semua Availability Zone tempat EI tersedia. EI gagal jika Anda tidak menyiapkan setidaknya dua Availability Zone. Untuk informasi tentang subnet VPC, lihat VPC dan Subnet.

Lampirkan EI ke Instance Notebook

Untuk menguji dan mengevaluasi kinerja inferensi menggunakan EI, Anda dapat melampirkan EI ke instance notebook saat membuat atau memperbarui instance notebook. Anda kemudian dapat menggunakan EI dalam mode lokal untuk meng-host model di titik akhir yang dihosting pada instance notebook. Anda harus menguji berbagai ukuran instans notebook dan akselerator EI untuk mengevaluasi konfigurasi yang paling sesuai untuk kasus penggunaan Anda.

Siapkan untuk Menggunakan EI

Untuk menggunakan EI secara lokal dalam instance notebook, buat instance notebook dengan instance EI.

Untuk membuat instance notebook dengan instance EI

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilih Instance Notebook.
3. Pilih Buat instans notebook.
4. Untuk nama instance Notebook, berikan nama unik untuk instance notebook Anda.
5. Untuk jenis instance notebook, pilih instance CPU seperti ml.t2.medium.
6. Untuk Elastic Inference (EI), pilih instance dari daftar, seperti ml.eia2.medium.
7. Untuk peran IAM, pilih peran IAM yang memiliki izin yang diperlukan untuk digunakan SageMaker dan EI.
8. (Opsional) Untuk VPC - Opsional, jika Anda ingin instance notebook menggunakan VPC, pilih salah satu dari daftar yang tersedia. Jika tidak, biarkan sebagai No VPC. Jika Anda menggunakan VPC, ikuti instruksi di. Gunakan VPC Kustom untuk Connect ke EI
9. (Opsional) Untuk konfigurasi Siklus Hidup - opsional, biarkan sebagai Tidak ada konfigurasi atau pilih konfigurasi siklus hidup. Untuk informasi selengkapnya, lihat Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup.
10. (Opsional) Untuk kunci Enkripsi - opsional, Opsional) Jika Anda SageMaker ingin menggunakan kunci AWS Key Management Service (AWS KMS) untuk mengenkripsi data dalam volume penyimpanan ML yang dilampirkan ke instance notebook, tentukan kuncinya.
11. (Opsional) Untuk Ukuran Volume Dalam GB - opsional, biarkan nilai default 5.
12. (Opsional) Untuk Tag, tambahkan tag ke instance notebook. Tag adalah label yang Anda tetapkan untuk membantu mengelola instance notebook Anda. Setiap tanda terdiri atas sebuah kunci dan sebuah nilai, yang keduanya Anda tentukan.
13. Pilih Buat Instance Notebook.

Setelah membuat instance notebook dengan EI terlampir, Anda dapat membuat notebook Jupyter dan menyiapkan titik akhir EI yang di-host secara lokal pada instance notebook.

Topik

- Gunakan EI dalam Mode Lokal di SageMaker

Gunakan EI dalam Mode Lokal di SageMaker

Untuk menggunakan EI secara lokal di titik akhir yang dihosting pada instance notebook, gunakan mode lokal dengan versi Amazon SageMaker Python SDK dari TensorFlow MXNet, atau estimator atau model. PyTorch Untuk informasi selengkapnya tentang dukungan mode lokal di SageMaker Python SDK, lihat https://github.com/aws/ #. sagemaker-python-sdk sagemaker-python-sdk-overview

Topik

- Gunakan EI dalam Mode Lokal dengan SageMaker TensorFlow Estimator dan Model
- Gunakan EI dalam Mode Lokal dengan Estimator dan SageMaker Model Apache MXNet
- Gunakan EI dalam Mode Lokal dengan SageMaker PyTorch Estimator dan Model

Gunakan EI dalam Mode Lokal dengan SageMaker TensorFlow Estimator dan Model

Untuk menggunakan EI dengan TensorFlow dalam mode lokal, tentukan `local` `local_sagemaker_notebook` untuk `instance_type` dan untuk `accelerator_type` saat Anda memanggil `deploy` metode estimator atau objek model. Untuk informasi selengkapnya tentang TensorFlow estimator dan model Amazon SageMaker Python SDK, lihat https://sagemaker.readthedocs.io/en/stable/frameworks/tensorflow/index.html.

Kode berikut menunjukkan cara menggunakan mode lokal dengan objek estimator. Untuk memanggil `deploy` metode ini, Anda harus memiliki sebelumnya:

- Melatih model dengan memanggil `fit` metode estimator.
- Lewati artefak model saat Anda menginisialisasi objek model.

```
# Deploys the model to a local endpoint
tf_predictor = tf_model.deploy(initial_instance_count=1,
                               instance_type='local',
                               accelerator_type='local_sagemaker_notebook')
```

Gunakan EI dalam Mode Lokal dengan Estimator dan SageMaker Model Apache MXNet

Untuk menggunakan EI dengan MxNet dalam mode lokal, `local` tentukan `instance_type` untuk `local_sagemaker_notebook` dan `accelerator_type` untuk saat Anda memanggil `deploy` metode estimator atau objek model. Untuk informasi selengkapnya tentang estimator dan model MXNet Amazon SageMaker Python SDK, lihat https://sagemaker.readthedocs.io/en/stable/frameworks/mxnet/index.html.

Kode berikut menunjukkan cara menggunakan mode lokal dengan objek estimator. Anda sebelumnya harus memanggil `fit` metode estimator untuk melatih model.

```
# Deploys the model to a local endpoint
mxnet_predictor = mxnet_estimator.deploy(initial_instance_count=1,
                                         instance_type='local',
                                         accelerator_type='local_sagemaker_notebook')
```

Untuk contoh lengkap menggunakan EI dalam mode lokal dengan MxNet, lihat contoh notebook di https://sagemaker-examples.readthedocs.io/en/latest/ /mxnet_mnist/ mxnet_mnist_elastic_inference_local.html. sagemaker-python-sdk

Gunakan EI dalam Mode Lokal dengan SageMaker PyTorch Estimator dan Model

Untuk menggunakan EI dengan PyTorch dalam mode lokal, saat Anda memanggil `deploy` metode estimator atau objek model, tentukan `local` untuk `instance_type` dan `local_sagemaker_notebook` untuk`accelerator_type`. Untuk informasi selengkapnya tentang estimator dan model Amazon SageMaker Python SDK, lihat PyTorch SageMaker PyTorch Estimator dan Model.

Kode berikut menunjukkan cara menggunakan mode lokal dengan objek estimator. Anda sebelumnya harus memanggil `fit` metode estimator untuk melatih model.

```
# Deploys the model to a local endpoint
pytorch_predictor = pytorch_estimator.deploy(initial_instance_count=1,
                                             instance_type='local',

 accelerator_type='local_sagemaker_notebook')
```

Gunakan EI di Titik Akhir yang SageMaker Dihosting Amazon

Untuk menggunakan Elastic Inference (EI) di Amazon SageMaker dengan titik akhir yang dihosting untuk inferensi real-time, tentukan akselerator EI saat Anda membuat model deployable yang akan di-host pada titik akhir tersebut. Anda dapat melakukannya dengan salah satu cara berikut:

- Gunakan versi Amazon SageMaker Python SDK dari, TensorFlow MxNet, atau dan container yang sudah dibuat sebelumnya SageMaker untuk, MxNet, PyTorch dan TensorFlow PyTorch
- Bangun wadah Anda sendiri, dan gunakan SageMaker API tingkat rendah (Boto 3). Anda harus mengimpor versi EI-enabled dari TensorFlow MXNet, atau dari lokasi Amazon S3 yang PyTorch

disediakan ke dalam wadah Anda, dan menggunakan salah satu versi tersebut untuk menulis skrip pelatihan Anda.

- Gunakan algoritme Klasifikasi Gambar - MXNet atau Deteksi bawaan, dan gunakan algoritme AWS SDK for Python (Boto3) untuk menjalankan tugas pelatihan Anda dan buat model yang dapat diterapkan dan titik akhir yang dihosting.

Topik

Gunakan EI dengan SageMaker TensorFlow Container

Untuk menggunakan TensorFlow EI in SageMaker, Anda perlu memanggil `deploy` metode objek Estimator atau Model. Anda kemudian menentukan tipe akselerator menggunakan argumen input accelerator_type. Untuk informasi tentang penggunaan TensorFlow di SDK SageMaker Python, lihat: https://sagemaker.readthedocs.io/en/stable/frameworks/tensorflow/index.html.

SageMaker menyediakan pelatihan model default dan kode inferensi untuk kenyamanan Anda. Untuk format file kustom, Anda mungkin perlu menerapkan pelatihan model kustom dan kode inferensi.

Gunakan Objek Estimator

Untuk menggunakan objek estimator dengan EI, saat Anda menggunakan metode penerapan, sertakan argumen `accelerator_type` input. Estimator mengembalikan objek prediktor, yang kita sebut metode penyebaran, seperti yang ditunjukkan dalam kode contoh.

```
# Deploy an estimator using EI (using the accelerator_type input argument)
predictor = estimator.deploy(initial_instance_count=1,
                             instance_type='ml.m4.xlarge',
                             accelerator_type='ml.eia2.medium')
```

Gunakan Objek Model

Untuk menggunakan objek model dengan EI, saat Anda menggunakan metode penerapan, sertakan argumen `accelerator_type` input. Estimator mengembalikan objek prediktor, yang kita sebut metode penyebaran, seperti yang ditunjukkan dalam kode contoh.

```
# Deploy a model using EI (using the accelerator_type input argument)
predictor = model.deploy(initial_instance_count=1,
                         instance_type='ml.m4.xlarge',
                         accelerator_type='ml.eia2.medium')
```

Gunakan EI dengan SageMaker MxNet Container

Untuk menggunakan MxNet dengan EI SageMaker in, Anda perlu memanggil `deploy` metode objek Estimator atau Model. Anda kemudian menentukan jenis akselerator menggunakan argumen `accelerator_type` masukan. Untuk informasi tentang penggunaan MxNet di Amazon SageMaker Python SDK, lihat https://sagemaker.readthedocs.io/en/stable/frameworks/mxnet/index.html

Untuk kenyamanan Anda, SageMaker berikan pelatihan model default dan kode inferensi. Untuk format file kustom, Anda mungkin perlu menulis pelatihan model kustom dan kode inferensi.

Gunakan Objek Estimator

Untuk menggunakan objek estimator dengan EI, saat Anda menggunakan metode penerapan, sertakan argumen `accelerator_type` input. Estimator mengembalikan objek prediktor, yang kita sebut metode penyebaran, seperti yang ditunjukkan dalam kode contoh.

```
# Deploy an estimator using EI (using the accelerator_type input argument)
predictor = estimator.deploy(initial_instance_count=1,
                         instance_type='ml.m4.xlarge',
                         accelerator_type='ml.eia2.medium')
```

Gunakan Objek Model

Untuk menggunakan objek model dengan EI, saat Anda menggunakan metode penerapan, sertakan argumen `accelerator_type` input. Estimator mengembalikan objek prediktor, yang kita sebut metode penyebaran, seperti yang ditunjukkan dalam kode contoh.

```
# Deploy a model using EI (using the accelerator_type input argument)
predictor = model.deploy(initial_instance_count=1,
                         instance_type='ml.m4.xlarge',
                         accelerator_type='ml.eia2.medium')
```

Untuk contoh lengkap penggunaan EI dengan MxNet di, lihat contoh notebook di https://github.com/awslabs/ /blob/master/ SageMaker /mxnet_mnist/mxnet_mnist_elastic_inference.ipynb amazon-sagemaker-examples sagemaker-python-sdk

Gunakan EI dengan SageMaker PyTorch Container

Untuk menggunakan PyTorch EI in SageMaker, Anda perlu memanggil `deploy` metode objek Estimator atau Model. Anda kemudian menentukan jenis akselerator menggunakan argumen `accelerator_type` masukan. Untuk informasi tentang penggunaan PyTorch di Amazon SageMaker Python SDK, lihat SageMaker PyTorch Estimator dan Model.

Untuk kenyamanan Anda, SageMaker berikan pelatihan model default dan kode inferensi. Untuk format file kustom, Anda mungkin perlu menulis pelatihan model kustom dan kode inferensi.

Gunakan Objek Estimator

Untuk menggunakan objek estimator dengan EI, saat Anda menggunakan metode penerapan, sertakan argumen `accelerator_type` input. Estimator mengembalikan objek prediktor, yang kita sebut metode penyebaran, seperti yang ditunjukkan dalam kode contoh ini.

```
# Deploy an estimator using EI (using the accelerator_type input argument)
predictor = estimator.deploy(initial_instance_count=1,
                             instance_type='ml.m4.xlarge',
                             accelerator_type='ml.eia2.medium')
```

Gunakan Objek Model

Untuk menggunakan objek model dengan EI, saat Anda menggunakan metode penerapan, sertakan argumen `accelerator_type` input. Model mengembalikan objek prediktor, yang kita sebut metode penyebaran, seperti yang ditunjukkan dalam kode contoh ini.

```
# Deploy a model using EI (using the accelerator_type input argument)
predictor = model.deploy(initial_instance_count=1,
                             instance_type='ml.m4.xlarge',
                             accelerator_type='ml.eia2.medium')
```

Gunakan EI dengan Wadah Anda Sendiri

Untuk menggunakan EI dengan model dalam wadah khusus yang Anda buat, gunakan AWS SDK tingkat rendah untuk Python (Boto 3). unduh dan impor versi AWS EI-enabled dari TensorFlow, Apache MXNet, PyTorch atau kerangka kerja pembelajaran mesin, dan tulis skrip pelatihan Anda menggunakan kerangka kerja tersebut.

Impor Versi EI dari TensorFlow, MxNet, PyTorch atau ke Container Docker Anda

Untuk menggunakan EI dengan container Anda sendiri, Anda perlu mengimpor library Amazon EI TensorFlow Serving, library Amazon EI Apache MXNet, atau library yang diaktifkan Elastic Inference ke dalam container Anda. PyTorch Versi EI-enabled dari dan TensorFlow MXNet saat ini tersedia sebagai file biner yang disimpan di lokasi Amazon S3. Anda dapat mengunduh biner berkemampuan EI dari bucket Amazon S3 di TensorFlow console.aws.amazon.com/s3/buckets/amazonei-tensorflow. Untuk informasi tentang membangun kontainer yang menggunakan versi EI-enabled dari TensorFlow, lihat https://github.com/aws/ sagemaker-tensorflow-container # building-the-sagemaker-elastic -. inference-tensorflow-serving-container Anda dapat mengunduh biner berkemampuan EI untuk Apache MXNet dari bucket Amazon S3 publik di console.aws.amazon.com/s3/buckets/amazonei-apachemxnet. Untuk informasi tentang membangun sebuah container yang menggunakan versi EI-enabled MxNet, lihat https://github.com/aws/ # -. sagemaker-mxnet-container building-the-sagemaker-elastic inference-mxnet-container Anda dapat mengunduh biner yang diaktifkan Elastic Inference untuk. PyTorch Untuk informasi tentang membuat kontainer yang menggunakan versi Elastic Inference yang diaktifkan PyTorch, lihat Membangun gambar Anda.

Buat Endpoint EI dengan AWS SDK untuk Python (Boto 3)

Untuk membuat endpoint dengan menggunakan AWS SDK untuk Python (Boto 3), pertama-tama Anda membuat konfigurasi endpoint. Konfigurasi titik akhir menentukan satu atau lebih model (disebut varian produksi) yang ingin Anda host di titik akhir. Untuk melampirkan EI ke satu atau beberapa varian produksi yang dihosting di titik akhir, Anda menentukan salah satu jenis instance EI sebagai `AcceleratorType` bidang untuk itu`ProductionVariant`. Anda kemudian meneruskan konfigurasi titik akhir itu saat Anda membuat titik akhir.

Buat Konfigurasi Endpoint

Untuk menggunakan EI, Anda perlu menentukan jenis akselerator dalam konfigurasi titik akhir.

```
# Create Endpoint Configuration
from time import gmtime, strftime

endpoint_config_name = 'ImageClassificationEndpointConfig-' + strftime("%Y-%m-%d-%H-%M-
%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = sagemaker.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType':'ml.m4.xlarge',
```

```
        'InitialInstanceCount':1,
        'ModelName':model_name,
        'VariantName':'AllTraffic',
        'AcceleratorType':'ml.eia2.medium'}])

print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
```

Buat Endpoint

Setelah Anda membuat konfigurasi titik akhir dengan tipe akselerator, Anda dapat membuat titik akhir.

```
endpoint_name = 'ImageClassificationEndpoint-' + strftime("%Y-%m-%d-%H-%M-%S",
 gmtime())
endpoint_response = sagemaker.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name)
```

Setelah membuat titik akhir, Anda dapat memanggilnya menggunakan `invoke_endpoint` metode dalam objek runtime Boto3, seperti yang Anda lakukan pada titik akhir lainnya.

Praktik terbaik

Topik berikut memberikan panduan tentang praktik terbaik untuk menerapkan model pembelajaran mesin di Amazon SageMaker.

Topik

- Praktik terbaik untuk menerapkan model pada SageMaker Layanan Hosting
- Pantau Praktik Terbaik Keamanan
- Inferensi real-time latensi rendah dengan AWS PrivateLink
- Migrasikan beban kerja inferensi dari x86 ke Graviton AWS
- Memecahkan masalah penerapan model Amazon SageMaker
- Praktik terbaik pengoptimalan biaya inferensi
- Praktik terbaik untuk meminimalkan gangguan selama peningkatan driver GPU
- Praktik terbaik untuk keamanan dan kesehatan titik akhir dengan Amazon SageMaker

Praktik terbaik untuk menerapkan model pada SageMaker Layanan Hosting

Saat menghosting model menggunakan layanan SageMaker hosting, pertimbangkan hal berikut:

- Biasanya, aplikasi klien mengirimkan permintaan ke titik akhir SageMaker HTTPS untuk mendapatkan kesimpulan dari model yang diterapkan. Anda juga dapat mengirim permintaan ke titik akhir ini dari buku catatan Jupyter Anda selama pengujian.
- Anda dapat menerapkan model yang dilatih dengan SageMaker target penyebaran Anda sendiri. Untuk melakukan itu, Anda perlu mengetahui format khusus algoritme dari artefak model yang dihasilkan oleh pelatihan model. Untuk informasi selengkapnya tentang format keluaran, lihat bagian yang sesuai dengan algoritme yang Anda gunakanFormat Data Umum untuk Pelatihan.
- Anda dapat menerapkan beberapa varian model ke titik akhir SageMaker HTTPS yang sama. Ini berguna untuk menguji variasi model dalam produksi. Misalnya, misalkan Anda telah menerapkan model ke dalam produksi. Anda ingin menguji variasi model dengan mengarahkan sejumlah kecil lalu lintas, katakanlah 5%, ke model baru. Untuk melakukan ini, buat konfigurasi titik akhir yang menjelaskan kedua varian model. Anda menentukan `ProductionVariant` dalam permintaan Anda ke`CreateEndPointConfig`. Untuk informasi selengkapnya, lihat `ProductionVariant`.
- Anda dapat mengonfigurasi `ProductionVariant` untuk menggunakan Application Auto Scaling. Untuk informasi tentang mengonfigurasi penskalaan otomatis, lihat. Secara Otomatis Menskalakan SageMaker Model Amazon
- Anda dapat memodifikasi titik akhir tanpa mengambil model yang sudah digunakan ke produksi di luar layanan. Misalnya, Anda dapat menambahkan varian model baru, memperbarui konfigurasi instans Komputasi ML dari varian model yang ada, atau mengubah distribusi lalu lintas antar varian model. Untuk memodifikasi endpoint, Anda menyediakan konfigurasi endpoint baru. SageMaker mengimplementasikan perubahan tanpa downtime. Untuk informasi lebih lanjut lihat, `UpdateEndpoint`dan `UpdateEndpointWeightsAndCapacities`.
- Mengubah atau menghapus artefak model atau mengubah kode inferensi setelah menerapkan model menghasilkan hasil yang tidak terduga. Jika Anda perlu mengubah atau menghapus artefak model atau mengubah kode inferensi, ubah titik akhir dengan menyediakan konfigurasi titik akhir baru. Setelah Anda memberikan konfigurasi endpoint baru, Anda dapat mengubah atau menghapus artefak model yang sesuai dengan konfigurasi endpoint lama.
- Jika Anda ingin mendapatkan kesimpulan tentang seluruh kumpulan data, pertimbangkan untuk menggunakan transformasi batch sebagai alternatif layanan hosting. Untuk informasi, lihat Gunakan Batch Transform

Terapkan Beberapa Instance di Seluruh Availability Zone

Buat titik akhir yang kuat saat menghosting model Anda. SageMakerendpoint dapat membantu melindungi aplikasi Anda dari pemadaman Availability Zone dan kegagalan instans. Jika terjadi pemadaman atau instans gagal, SageMaker secara otomatis mencoba mendistribusikan instans Anda di seluruh Availability Zone. Untuk alasan ini, kami sangat menyarankan agar Anda menerapkan beberapa instance untuk setiap titik akhir produksi.

Jika Anda menggunakan Amazon Virtual Private Cloud (VPC), konfigurasikan VPC dengan setidaknya dua `Subnets`, masing-masing di Availability Zone yang berbeda. Jika terjadi pemadaman atau instans gagal, Amazon SageMaker secara otomatis mencoba mendistribusikan instans Anda di seluruh Availability Zone.

Secara umum, untuk mencapai kinerja yang lebih andal, gunakan lebih banyak Jenis Instance kecil di Availability Zone yang berbeda untuk meng-host endpoint Anda.

Menyebarkan komponen inferensi untuk ketersediaan tinggi. Selain rekomendasi di atas untuk nomor instans, untuk mencapai ketersediaan 99,95%, pastikan komponen inferensi Anda dikonfigurasi untuk memiliki lebih dari dua salinan. Selain itu, dalam kebijakan penskalaan otomatis terkelola, tetapkan jumlah minimum instans menjadi dua juga.

Pantau Praktik Terbaik Keamanan

Pantau penggunaan SageMaker Anda terkait dengan praktik terbaik keamanan dengan menggunakan AWSSecurity Hub. Hub Keamanan menggunakan kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan guna membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya tentang menggunakan Security Hub untuk mengevaluasi SageMaker sumber daya, lihat SageMaker kontrol Amazon di Panduan Pengguna AWS Security Hub.

Inferensi real-time latensi rendah dengan AWS PrivateLink

Amazon SageMaker menyediakan latensi rendah untuk inferensi waktu nyata sambil mempertahankan ketersediaan dan ketahanan tinggi menggunakan penerapan Multi-AZ. Latensi aplikasi terdiri dari dua komponen utama: infrastruktur atau latensi overhead dan latensi inferensi model. Pengurangan latensi overhead membuka kemungkinan baru seperti menerapkan model yang lebih kompleks, mendalam, dan akurat atau membagi aplikasi monolitik menjadi modul layanan mikro yang dapat diskalakan dan dapat dipelihara. Anda dapat mengurangi latensi untuk inferensi

real-time dengan SageMaker menggunakan penerapan. AWS PrivateLink DenganAWS PrivateLink, Anda dapat mengakses semua operasi SageMaker API secara pribadi dari Virtual Private Cloud (VPC) Anda dengan cara yang dapat diskalakan dengan menggunakan titik akhir VPC antarmuka. Endpoint VPC antarmuka adalah elastic network interface di subnet Anda dengan alamat IP pribadi yang berfungsi sebagai titik masuk untuk semua panggilan API. SageMaker

Secara default, SageMaker titik akhir dengan 2 instans atau lebih diterapkan di setidaknya 2 AWS Availability Zones (AZ) dan instance di AZ mana pun dapat memproses pemanggilan. Ini menghasilkan satu atau lebih “hop” AZ yang berkontribusi pada latensi overhead. AWS PrivateLinkPenerapan dengan `privateDNSEnabled` opsi yang ditetapkan sebagai `true` meringankan hal ini dengan mencapai dua tujuan:

- Itu menyimpan semua lalu lintas inferensi dalam VPC Anda.
- Itu membuat lalu lintas pemanggilan di AZ yang sama dengan klien yang memulainya saat menggunakan Runtime. SageMaker Ini menghindari “lompatan” antara AZ yang mengurangi latensi overhead.

Bagian berikut dari panduan ini menunjukkan bagaimana Anda dapat mengurangi latensi untuk inferensi waktu nyata dengan AWS PrivateLink penerapan.

Topik

- Menyebarkan AWS PrivateLink
- Menerapkan SageMaker titik akhir di VPC
- Memanggil titik akhir SageMaker

Menyebarkan AWS PrivateLink

Untuk menerapkanAWS PrivateLink, pertama-tama buat titik akhir antarmuka untuk VPC tempat Anda terhubung ke titik akhir. SageMaker Ikuti langkah-langkah di Akses AWS layanan menggunakan titik akhir VPC antarmuka untuk membuat titik akhir antarmuka. Saat membuat titik akhir, pilih pengaturan berikut di antarmuka konsol:

- Pilih kotak centang Aktifkan nama DNS di bawah Pengaturan Tambahan
- Pilih grup keamanan yang sesuai dan subnet yang akan digunakan dengan titik SageMaker akhir.

Pastikan juga bahwa VPC mengaktifkan nama host DNS. Untuk informasi selengkapnya tentang cara mengubah atribut DNS untuk VPC Anda, lihat Melihat dan memperbarui atribut DNS untuk VPC Anda.

Menerapkan SageMaker titik akhir di VPC

Untuk mencapai latensi overhead rendah, buat SageMaker titik akhir menggunakan subnet yang sama dengan yang Anda tentukan saat menerapkan. AWS PrivateLink Subnet ini harus cocok dengan AZ aplikasi klien Anda, seperti yang ditunjukkan dalam cuplikan kode berikut.

```
model_name = '<the-name-of-your-model>'

vpc = 'vpc-0123456789abcdef0'
subnet_a = 'subnet-0123456789abcdef0'
subnet_b = 'subnet-0123456789abcdef1'
security_group = 'sg-0123456789abcdef0'

create_model_response = sagemaker_client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = sagemaker_role,
    PrimaryContainer = {
        'Image': container,
        'ModelDataUrl': model_url
    },
    VpcConfig = {
        'SecurityGroupIds': [security_group],
        'Subnets': [subnet_a, subnet_b],
    },
)
```

Cuplikan kode yang disebutkan di atas mengasumsikan bahwa Anda telah mengikuti langkah-langkahnya. Sebelum Anda memulai

Memanggil titik akhir SageMaker

Terakhir, tentukan klien SageMaker Runtime dan panggil SageMaker titik akhir seperti yang ditunjukkan pada cuplikan kode berikut.

```
endpoint_name = '<endpoint-name>'

runtime_client = boto3.client('sagemaker-runtime')
```

```
response = runtime_client.invoke_endpoint(EndpointName=endpoint_name,
                                          ContentType='text/csv',
                                          Body=payload)
```

Untuk informasi selengkapnya tentang konfigurasi titik akhir, lihatTerapkan model untuk inferensi waktu nyata.

Migrasikan beban kerja inferensi dari x86 ke Graviton AWS

AWSGraviton adalah serangkaian prosesor berbasis ARM yang dirancang oleh. AWS Mereka lebih hemat energi daripada prosesor berbasis x86 dan menawarkan rasio harga-kinerja yang menarik. Amazon SageMaker menawarkan instance berbasis Graviton sehingga Anda dapat memanfaatkan prosesor canggih ini untuk kebutuhan inferensi Anda.

Anda dapat memigrasikan beban kerja inferensi yang ada dari instance berbasis x86 ke instance berbasis Graviton, dengan menggunakan gambar kontainer yang kompatibel dengan ARM atau gambar kontainer multi-arsitektur. Panduan ini mengasumsikan bahwa Anda menggunakan gambar kontainer AWSDeep Learning, atau gambar kontainer yang kompatibel dengan ARM Anda sendiri. Untuk informasi lebih lanjut tentang membuat gambar Anda sendiri, periksa Membangun gambar Anda.

Pada tingkat tinggi, memigrasikan beban kerja inferensi dari instance berbasis x86 ke instance berbasis Graviton adalah proses empat langkah:

1. Dorong gambar kontainer ke Amazon Elastic Container Registry (Amazon ECR), registri kontainer AWS terkelola.
2. Buat SageMaker Model.
3. Buat konfigurasi titik akhir.
4. Buat titik akhir.

Bagian berikut dari panduan ini memberikan rincian lebih lanjut mengenai langkah-langkah di atas. Ganti *`teks placeholder pengguna`* dalam contoh kode dengan informasi Anda sendiri.

Topik

- Dorong gambar wadah ke Amazon ECR
- Buat SageMaker Model
- Buat konfigurasi titik akhir

- Buat titik akhir

Dorong gambar wadah ke Amazon ECR

Anda dapat mendorong gambar kontainer Anda ke Amazon ECR dengan file. AWS CLI Saat menggunakan gambar yang kompatibel dengan ARM, verifikasi bahwa itu mendukung arsitektur ARM:

```
docker inspect deep-learning-container-uri
```

Respons `"Architecture": "arm64"` menunjukkan bahwa gambar mendukung arsitektur ARM. Anda dapat mendorongnya ke Amazon ECR dengan `docker push` perintah. Untuk informasi selengkapnya, periksa Mendorong gambar Docker.

Gambar kontainer multi-arsitektur pada dasarnya adalah sekumpulan gambar kontainer yang mendukung arsitektur atau sistem operasi yang berbeda, yang dapat Anda rujuk dengan nama manifes umum. Jika Anda menggunakan gambar wadah multi-arsitektur, maka selain mendorong gambar ke Amazon ECR, Anda juga harus mendorong daftar manifes ke Amazon ECR. Daftar manifes memungkinkan penyertaan bersarang dari manifes gambar lain, di mana setiap gambar yang disertakan ditentukan oleh arsitektur, sistem operasi, dan atribut platform lainnya. Contoh berikut membuat daftar manifes, dan mendorongnya ke Amazon ECR.

1. Buat daftar manifes.

   ```
   docker manifest create aws-account-id.dkr.ecr.aws-region.amazonaws.com/my-
   repository \
     aws-account-id.dkr.ecr.aws-account-id.amazonaws.com/my-repository:amd64 \
    aws-account-id.dkr.ecr.aws-account-id.amazonaws.com/my-repository:arm64 \
   ```

2. Beri anotasi daftar manifes, sehingga benar mengidentifikasi gambar mana untuk arsitektur mana.

   ```
   docker manifest annotate --arch arm64 aws-account-id.dkr.ecr.aws-
   region.amazonaws.com/my-repository \
     aws-account-id.dkr.ecr.aws-region.amazonaws.com/my-repository:arm64
   ```

3. Dorong manifes.

```
docker manifest push aws-account-id.dkr.ecr.aws-region.amazonaws.com/my-repository
```

Untuk informasi selengkapnya tentang membuat dan mendorong daftar manifes ke Amazon ECR, periksa Memperkenalkan gambar wadah multi-arsitektur untuk Amazon ECR, dan Mendorong gambar multi-arsitektur.

Buat SageMaker Model

Buat SageMaker Model dengan memanggil `CreateModel`API.

```
import boto3
from sagemaker import get_execution_role


aws_region = "aws-region"
sagemaker_client = boto3.client("sagemaker", region_name=aws_region)

role = get_execution_role()

sagemaker_client.create_model(
    ModelName = "model-name",
    PrimaryContainer = {
        "Image": "deep-learning-container-uri",
        "ModelDataUrl": "model-s3-location",
        "Environment": {
            "SAGEMAKER_PROGRAM": "inference.py",
            "SAGEMAKER_SUBMIT_DIRECTORY": "inference-script-s3-location",
            "SAGEMAKER_CONTAINER_LOG_LEVEL": "20",
            "SAGEMAKER_REGION": aws_region,
        }
    },
    ExecutionRoleArn = role
)
```

Buat konfigurasi titik akhir

Buat konfigurasi titik akhir dengan memanggil `CreateEndpointConfig`API. Untuk daftar instance berbasis Graviton, periksa Instance yang dioptimalkan Compute.

```
sagemaker_client.create_endpoint_config(
    EndpointConfigName = "endpoint-config-name",
    ProductionVariants = [
        {
            "VariantName": "variant-name",
            "ModelName": "model-name",
            "InitialInstanceCount": 1,
            "InstanceType": "ml.c7g.xlarge", # Graviton-based instance
       }
    ]
)
```

Buat titik akhir

Buat titik akhir dengan memanggil `CreateEndpoint`API.

```
sagemaker_client.create_endpoint(
    EndpointName = "endpoint-name",
    EndpointConfigName = "endpoint-config-name"
)
```

Memecahkan masalah penerapan model Amazon SageMaker

Jika Anda mengalami masalah saat menerapkan model pembelajaran mesin di Amazon SageMaker, lihat panduan berikut.

Topik

- Kesalahan Deteksi dalam Hitungan CPU Aktif
- Masalah dengan penerapan file model.tar.gz
- Kontainer primer tidak lulus pemeriksaan kesehatan ping

Kesalahan Deteksi dalam Hitungan CPU Aktif

Jika Anda menerapkan SageMaker model dengan Linux Java Virtual Machine (JVM), Anda mungkin mengalami kesalahan deteksi yang mencegah penggunaan sumber daya CPU yang tersedia. Masalah ini mempengaruhi beberapa JVM yang mendukung Java 8 dan Java 9, dan sebagian besar yang mendukung Java 10 dan Java 11. JVM ini menerapkan mekanisme yang mendeteksi

dan menangani jumlah CPU dan memori maksimum yang tersedia saat menjalankan model dalam wadah Docker, dan, lebih umum, dalam `taskset` perintah Linux atau grup kontrol (cgroups). SageMakerpenerapan memanfaatkan beberapa pengaturan yang digunakan JVM untuk mengelola sumber daya ini. Saat ini, ini menyebabkan wadah salah mendeteksi jumlah CPU yang tersedia.

SageMaker tidak membatasi akses ke CPU pada sebuah instance. Namun, JVM mungkin mendeteksi jumlah CPU seperti 1 ketika lebih banyak CPU tersedia untuk wadah. Akibatnya, JVM menyesuaikan semua pengaturan internalnya untuk berjalan seolah-olah hanya inti 1 CPU yang tersedia. Pengaturan ini memengaruhi pengumpulan sampah, kunci, utas kompiler, dan internal JVM lainnya yang berdampak negatif pada konkurensi, throughput, dan latensi wadah.

Untuk contoh kesalahan deteksi, dalam wadah yang dikonfigurasi untuk SageMaker itu digunakan dengan JVM yang didasarkan pada Java8_191 dan yang memiliki empat CPU yang tersedia pada instance, jalankan perintah berikut untuk memulai JVM Anda:

```
java -XX:+UnlockDiagnosticVMOptions -XX:+PrintActiveCpus -version
```

Ini menghasilkan output berikut:

```
active_processor_count: sched_getaffinity processor count: 4
active_processor_count: determined by OSContainer: 1
active_processor_count: sched_getaffinity processor count: 4
active_processor_count: determined by OSContainer: 1
active_processor_count: sched_getaffinity processor count: 4
active_processor_count: determined by OSContainer: 1
active_processor_count: sched_getaffinity processor count: 4
active_processor_count: determined by OSContainer: 1
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-8u191-b12-2ubuntu0.16.04.1-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
```

Banyak JVM yang terpengaruh oleh masalah ini memiliki opsi untuk menonaktifkan perilaku ini dan membangun kembali akses penuh ke semua CPU pada instance. Nonaktifkan perilaku yang tidak diinginkan dan buat akses penuh ke semua CPU instance dengan memasukkan `-XX:-UseContainerSupport` parameter saat memulai aplikasi Java. Misalnya, jalankan `java` perintah untuk memulai JVM Anda sebagai berikut:

```
java -XX:-UseContainerSupport -XX:+UnlockDiagnosticVMOptions -XX:+PrintActiveCpus -
version
```

Ini menghasilkan output berikut:

```
active_processor_count: sched_getaffinity processor count: 4
active_processor_count: sched_getaffinity processor count: 4
active_processor_count: sched_getaffinity processor count: 4
active_processor_count: sched_getaffinity processor count: 4
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-8u191-b12-2ubuntu0.16.04.1-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
```

Periksa apakah JVM yang digunakan dalam wadah Anda mendukung parameter. `-XX:-UseContainerSupport` Jika ya, selalu berikan parameter saat Anda memulai JVM Anda. Ini menyediakan akses ke semua CPU dalam instans Anda.

Anda mungkin juga mengalami masalah ini saat secara tidak langsung menggunakan JVM dalam wadah. SageMaker Misalnya, saat menggunakan JVM untuk mendukung SparkMl Scala. `-XX:-UseContainerSupport`Parameter juga mempengaruhi output yang dikembalikan oleh Java `Runtime.getRuntime().availableProcessors()` API.

Masalah dengan penerapan file model.tar.gz

Saat Anda menerapkan model menggunakan `model.tar.gz` file, tarball model tidak boleh menyertakan symlink apa pun. Symlink menyebabkan pembuatan model gagal. Selain itu, kami menyarankan Anda untuk tidak memasukkan file yang tidak perlu di tarball.

Kontainer primer tidak lulus pemeriksaan kesehatan ping

Jika penampung utama Anda gagal melakukan ping pemeriksaan kesehatan dengan pesan kesalahan berikut, ini menunjukkan bahwa ada masalah dengan penampung atau skrip Anda:

```
The primary container for production variant beta did not pass the ping health check.
 Please check CloudWatch Logs logs for this endpoint.
```

Untuk memecahkan masalah ini, Anda harus memeriksa CloudWatch log Log untuk titik akhir yang dimaksud untuk melihat apakah ada kesalahan atau masalah yang mencegah penampung merespons atau. `/ping` `/invocations` Log dapat memberikan pesan kesalahan yang dapat menunjukkan masalah. Setelah Anda mengidentifikasi alasan kesalahan dan kegagalan, Anda harus menyelesaikan kesalahan.

Ini juga merupakan praktik yang baik untuk menguji penerapan model secara lokal sebelum membuat titik akhir.

- Gunakan mode lokal di SageMaker SDK untuk meniru lingkungan yang dihosting dengan menerapkan model ke titik akhir lokal. Untuk informasi selengkapnya, lihat Mode Lokal.
- Gunakan perintah vanilla docker untuk menguji container merespons /ping dan /invocations. Untuk informasi selengkapnya, lihat local_test.

Praktik terbaik pengoptimalan biaya inferensi

Konten berikut menyediakan teknik dan pertimbangan untuk mengoptimalkan biaya titik akhir. Anda dapat menggunakan rekomendasi ini untuk mengoptimalkan biaya untuk titik akhir baru dan yang sudah ada.

Praktik terbaik

Untuk mengoptimalkan biaya SageMaker Inferensi Anda, ikuti praktik terbaik ini.

Pilih opsi inferensi terbaik untuk pekerjaan itu.

SageMaker menawarkan 4 opsi inferensi yang berbeda untuk memberikan opsi inferensi terbaik untuk pekerjaan itu. Anda mungkin dapat menghemat biaya dengan memilih opsi inferensi yang paling sesuai dengan beban kerja Anda.

- Gunakan inferensi real-time untuk beban kerja latensi rendah dengan pola lalu lintas yang dapat diprediksi yang harus memiliki karakteristik latensi yang konsisten dan selalu tersedia. Anda membayar untuk menggunakan instance.
- Gunakan inferensi tanpa server untuk beban kerja sinkron yang memiliki pola lalu lintas runcing dan dapat menerima variasi dalam latensi p99. Inferensi tanpa server secara otomatis menskalakan untuk memenuhi lalu lintas beban kerja Anda sehingga Anda tidak membayar sumber daya idle apa pun. Anda hanya membayar untuk durasi permintaan inferensi. Model dan wadah yang sama dapat digunakan dengan inferensi real-time dan tanpa server sehingga Anda dapat beralih di antara dua mode ini jika kebutuhan Anda berubah.
- Gunakan inferensi asinkron untuk beban kerja asinkron yang memproses hingga 1 GB data (seperti korpus teks, gambar, video, dan audio) yang tidak sensitif terhadap latensi dan sensitif biaya. Dengan inferensi asinkron, Anda dapat mengontrol biaya dengan menentukan jumlah instans tetap untuk tingkat pemrosesan optimal alih-alih menyediakan untuk puncak. Anda juga dapat menurunkan skala ke nol untuk menghemat biaya tambahan.
- Gunakan inferensi batch untuk beban kerja yang Anda perlukan inferensi untuk sekumpulan besar data untuk proses yang terjadi secara offline (yaitu, Anda tidak memerlukan titik akhir yang persisten). Anda membayar instance untuk durasi pekerjaan inferensi batch.

Ikut serta dalam SageMaker Savings Plan.

- Jika Anda memiliki tingkat penggunaan yang konsisten di semua SageMaker layanan, Anda dapat ikut serta dalam SageMaker Savings Plan untuk membantu mengurangi biaya hingga 64%.
- Amazon SageMaker Savings Plans menyediakan model harga yang fleksibel untuk Amazon SageMaker, dengan imbalan komitmen terhadap jumlah penggunaan yang konsisten (diukur dalam $/jam) untuk jangka waktu satu tahun atau tiga tahun. Paket ini secara otomatis berlaku untuk penggunaan instans MS SageMaker yang memenuhi syarat termasuk SageMaker Studio Classic Notebook, SageMaker On-Demand Notebook, SageMaker Processing, SageMaker Data Wrangler, SageMaker Training, SageMaker Real-Time Inference, dan SageMaker Batch Transform terlepas dari keluarga instans, ukuran, atau Wilayah. Misalnya, Anda dapat mengubah penggunaan dari instance CPU ml.c5.xlarge yang berjalan di US East (Ohio) ke instans ML.INF1 di US West (Oregon) untuk beban kerja inferensi kapan saja dan secara otomatis terus membayar harga Savings Plans.

Optimalkan model Anda agar berjalan lebih baik.

- Model yang tidak dioptimalkan dapat menyebabkan waktu berjalan lebih lama dan menggunakan lebih banyak sumber daya. Anda dapat memilih untuk menggunakan lebih banyak atau lebih besar contoh untuk meningkatkan kinerja; Namun, ini mengarah pada biaya yang lebih tinggi.
- Dengan mengoptimalkan model Anda agar lebih berkinerja, Anda mungkin dapat menurunkan biaya dengan menggunakan instance yang lebih sedikit atau lebih kecil sambil mempertahankan karakteristik kinerja yang sama atau lebih baik. Anda dapat menggunakan SageMaker Neo dengan SageMaker Inferensi untuk mengoptimalkan model secara otomatis. Untuk detail dan sampel lebih lanjut, lihatOptimalkan kinerja model menggunakan Neo.

Gunakan jenis dan ukuran instans yang paling optimal untuk inferensi waktu nyata.

- SageMaker Inferensi memiliki lebih dari 70 jenis dan ukuran instans yang dapat digunakan untuk menerapkan model ML termasuk chipset AWS Inferentia dan Graviton yang dioptimalkan untuk ML. Memilih instance yang tepat untuk model Anda membantu memastikan Anda memiliki instance berkinerja terbaik dengan biaya terendah untuk model Anda.

- Dengan menggunakan Inference Recommender, Anda dapat dengan cepat membandingkan instans yang berbeda untuk memahami kinerja model dan biaya. Dengan hasil ini, Anda dapat memilih instans untuk digunakan dengan laba atas investasi terbaik.

Tingkatkan efisiensi dan biaya dengan menggabungkan beberapa titik akhir menjadi satu titik akhir untuk inferensi waktu nyata.

- Biaya dapat bertambah dengan cepat saat Anda menerapkan beberapa titik akhir, terutama jika titik akhir tidak sepenuhnya memanfaatkan instance yang mendasarinya. Untuk mengetahui apakah instans kurang digunakan, periksa metrik penggunaan (CPU, GPU, dll) di Amazon untuk instance Anda. CloudWatch Jika Anda memiliki lebih dari satu titik akhir ini, Anda dapat menggabungkan model atau wadah pada beberapa titik akhir ini menjadi satu titik akhir.
- Dengan menggunakan Multi-model endpoint (MME) atau Multi-container endpoint (MCE), Anda dapat menerapkan beberapa model atau kontainer ML dalam satu titik akhir untuk berbagi instance di beberapa model atau kontainer dan meningkatkan laba atas investasi Anda. Untuk mempelajari lebih lanjut, lihat ini Menghemat biaya inferensi dengan menggunakan titik akhir SageMaker multi-model Amazon atau Menerapkan beberapa kontainer penyajian pada satu instance menggunakan titik akhir SageMaker multi-container Amazon di blog Machine Learning. AWS

Siapkan penskalaan otomatis agar sesuai dengan persyaratan beban kerja Anda untuk inferensi real-time dan asinkron.

- Tanpa penskalaan otomatis, Anda perlu menyediakan lalu lintas puncak atau tidak tersedianya model risiko. Kecuali lalu lintas ke model Anda stabil sepanjang hari, akan ada kelebihan kapasitas yang tidak terpakai. Hal ini menyebabkan rendahnya pemanfaatan dan sumber daya yang terbuang.
- Autoscaling adalah out-of-the-box fitur yang memantau beban kerja Anda dan secara dinamis menyesuaikan kapasitas untuk mempertahankan kinerja yang stabil dan dapat diprediksi dengan biaya serendah mungkin. Ketika beban kerja meningkat, penskalaan otomatis membawa lebih banyak contoh online. Ketika beban kerja berkurang, penskalaan otomatis menghapus instans yang tidak perlu, membantu Anda mengurangi biaya komputasi. Untuk mempelajari lebih lanjut, lihat Mengonfigurasi titik akhir inferensi penskalaan otomatis di Amazon di blog Machine Learning SageMaker. AWS

Praktik terbaik untuk meminimalkan gangguan selama peningkatan driver GPU

SageMaker Model Deployment memutakhirkan driver GPU pada instans ML untuk opsi Inferensi Real-time, Batch, dan Asynchronous dari waktu ke waktu untuk memberi pelanggan akses ke peningkatan dari penyedia driver. Di bawah ini Anda dapat melihat versi GPU yang didukung untuk

setiap opsi Inferensi. Versi driver yang berbeda dapat mengubah cara model Anda berinteraksi dengan GPU. Di bawah ini adalah beberapa strategi untuk membantu Anda memahami cara kerja aplikasi Anda dengan versi driver yang berbeda.

Versi saat ini dan keluarga instans yang didukung

Amazon SageMaker Inference mendukung driver dan keluarga instans berikut:

Layanan	GPU	Versi Driver	Tipe instans
Real-time	NVIDIA	470.57.02	ml.p2.*, ml.p3.*, ml.p4d.*, ml.p4de.*, ml.g4dn.*, ml.g5. *
		535.54.03	ml.p5. *
Batch	NVIDIA	470.57.02	ml.p2.*, ml.p3.*, ml.p4d.*, ml.p4de.*, ml.g4dn.*, ml.g5*
Inferensi Asinkron	NVIDIA	470.57.02	ml.p2.*, ml.p3.*, ml.p4d.*, ml.p4de.*, ml.g4dn.*, ml.g5*
		535.54.03	ml.p5. *

Memecahkan masalah wadah model Anda dengan kemampuan GPU

Jika Anda mengalami masalah saat menjalankan beban kerja GPU, lihat panduan berikut:

Kegagalan deteksi kartu GPU atau kesalahan inisialisasi NVIDIA

Jalankan perintah `nvidia-smi` (NVIDIA System Management Interface) dari dalam wadah Docker. Jika Antarmuka Manajemen Sistem NVIDIA mendeteksi kesalahan deteksi GPU atau kesalahan inisialisasi NVIDIA, itu akan mengembalikan pesan kesalahan berikut:

```
Failed to initialize NVML: Driver/library version mismatch
```

Berdasarkan kasus penggunaan Anda, ikuti praktik terbaik berikut ini untuk mengatasi kegagalan atau kesalahan:

- Ikuti rekomendasi praktik terbaik yang dijelaskan dalam Jika Anda membawa wadah model (BYO) Anda sendiri dropdown.
- Ikuti rekomendasi praktik terbaik yang dijelaskan dalam Jika Anda menggunakan lapisan kompatibilitas CUDA dropdown.

Lihat halaman Antarmuka Manajemen Sistem NVIDIA di situs web NVIDIA untuk informasi lebih lanjut.

`CannotStartContainerError`

Jika instans GPU Anda menggunakan versi driver NVIDIA yang tidak kompatibel dengan versi CUDA di wadah Docker, maka penerapan titik akhir akan gagal dengan pesan kesalahan berikut:

```
Failure reason CannotStartContainerError. Please ensure the model container for
variant <variant_name> starts correctly when invoked with 'docker run <image> serve'
```

Berdasarkan kasus penggunaan Anda, ikuti praktik terbaik berikut ini untuk mengatasi kegagalan atau kesalahan:

- Ikuti rekomendasi praktik terbaik yang dijelaskan dalam Driver yang bergantung pada wadah saya lebih besar dari versi pada instance GPU ML dropdown.
- Ikuti rekomendasi praktik terbaik yang dijelaskan dalam Jika Anda menggunakan lapisan kompatibilitas CUDA dropdown.

Praktik terbaik untuk bekerja dengan versi driver yang tidak cocok

Berikut ini memberikan informasi tentang cara memperbarui driver GPU Anda:

Driver yang bergantung pada wadah saya lebih rendah dari versi pada instance GPU ML

Tidak ada tindakan yang diperlukan. NVIDIA menyediakan kompatibilitas mundur.

Driver yang bergantung pada wadah saya lebih besar dari versi pada instance GPU ML

Jika ini adalah perbedaan versi kecil, tidak ada tindakan yang diperlukan. NVIDIA menyediakan kompatibilitas maju versi minor.

Jika itu adalah perbedaan versi utama, CUDA Compatibility Package perlu diinstal. Silakan merujuk ke CUDA Compatibility Package dalam dokumentasi NVIDIA.

> **Important**
>
> CUDA Compatibility Package tidak kompatibel ke belakang sehingga perlu dinonaktifkan jika versi driver pada instance lebih besar dari versi CUDA Compatibility Package.

Jika Anda membawa wadah model (BYO) Anda sendiri

Pastikan tidak ada paket driver NVIDIA yang dibundel dalam gambar yang dapat menyebabkan konflik dengan versi driver NVIDIA host.

Jika Anda menggunakan lapisan kompatibilitas CUDA

Untuk memverifikasi apakah platform versi driver Nvidia mendukung versi CUDA Compatibility Package yang diinstal dalam wadah model, lihat dokumentasi CUDA. Jika platform versi driver Nvidia tidak mendukung versi CUDA Compatibility Package, Anda dapat menonaktifkan atau menghapus CUDA Compatibility Package dari gambar wadah model. Jika versi libs kompatibilitas CUDA didukung oleh versi driver Nvidia terbaru, kami sarankan Anda mengaktifkan CUDA Compatibility Package berdasarkan versi driver Nvidia yang terdeteksi untuk kompatibilitas masa depan dengan menambahkan cuplikan kode di bawah ini ke dalam skrip shell start up container (di skrip). `ENTRYPOINT`

Skrip menunjukkan cara mengganti penggunaan CUDA Compatibility Package secara dinamis berdasarkan versi driver Nvidia yang terdeteksi pada host yang digunakan untuk wadah model Anda. Saat SageMaker merilis versi driver Nvidia yang lebih baru, CUDA Compatibility Package yang diinstal dapat dimatikan secara otomatis jika aplikasi CUDA didukung secara native pada driver baru.

```
#!/bin/bash

verlte() {
    [ "$1" = "$2" ] && return 1 || [ "$2" = "`echo -e "$1\n$2" | sort -V | head -n1`" ]
}

if [ -f /usr/local/cuda/compat/libcuda.so.1 ]; then
    cat /usr/local/cuda/version.txt
    CUDA_COMPAT_MAX_DRIVER_VERSION=$(readlink /usr/local/cuda/compat/libcuda.so.1 |cut
 -d'.' -f 3-)
```

```
    echo "CUDA compat package requires Nvidia driver #
${CUDA_COMPAT_MAX_DRIVER_VERSION}"
    NVIDIA_DRIVER_VERSION=$(sed -n 's/^NVRM.*Kernel Module *\([0-9.]*\).*$/\1/p' /proc/
driver/nvidia/version 2>/dev/null || true)
    echo "Current installed Nvidia driver version is ${NVIDIA_DRIVER_VERSION}"
    if [ $(verlte $CUDA_COMPAT_MAX_DRIVER_VERSION $NVIDIA_DRIVER_VERSION) ]; then
        echo "Setup CUDA compatibility libs path to LD_LIBRARY_PATH"
        export LD_LIBRARY_PATH=/usr/local/cuda/compat:$LD_LIBRARY_PATH
        echo $LD_LIBRARY_PATH
    else
        echo "Skip CUDA compat libs setup as newer Nvidia driver is installed"
    fi
else
    echo "Skip CUDA compat libs setup as package not found"
fi
```

Praktik terbaik untuk keamanan dan kesehatan titik akhir dengan Amazon SageMaker

Untuk mengatasi masalah keamanan terbaru, Amazon SageMaker secara otomatis menambal titik akhir ke perangkat lunak terbaru dan paling aman. Namun, jika Anda salah mengubah dependensi titik akhir Anda, Amazon tidak SageMaker dapat secara otomatis menambal titik akhir Anda atau mengganti instance yang tidak sehat Anda. Untuk memastikan titik akhir Anda tetap memenuhi syarat untuk pembaruan otomatis, terapkan praktik terbaik berikut.

Jangan hapus sumber daya saat titik akhir Anda menggunakannya

Hindari menghapus salah satu sumber daya berikut jika Anda memiliki titik akhir yang menggunakannya:

- Definisi model yang Anda buat dengan CreateModeltindakan di Amazon SageMaker API.
- Artefak model apa pun yang Anda tentukan untuk `ModelDataUrl`parameter.
- Peran IAM dan izin yang Anda tentukan untuk parameter. `ExecutionRoleArn`

> **Pengingat**
>
> Dalam definisi model yang digunakan endpoint Anda, pastikan bahwa peran IAM yang Anda tentukan memiliki izin yang benar. Untuk informasi selengkapnya tentang izin

yang diperlukan untuk SageMaker titik akhir Amazon, lihat. CreateModel API: Izin Peran Eksekusi

- Gambar inferensi yang Anda tentukan untuk `Image`parameter, jika Anda menggunakan kode inferensi Anda sendiri.

> **Pengingat**
>
> Jika Anda menggunakan fitur registri pribadi, pastikan Amazon SageMaker dapat mengakses registri pribadi selama Anda menggunakan titik akhir.

- Subnet Amazon VPC dan grup keamanan yang Anda tentukan untuk parameter. `VpcConfig`
- Konfigurasi titik akhir yang Anda buat dengan CreateEndpointConfigtindakan di Amazon SageMaker API.
- Kunci KMS atau bucket Amazon S3 apa pun yang Anda tentukan dalam konfigurasi titik akhir.

> **Pengingat**
>
> Pastikan Anda tidak menonaktifkan kunci KMS ini.

Ikuti prosedur ini untuk memperbarui titik akhir Anda

Saat memperbarui SageMaker titik akhir Amazon, gunakan salah satu prosedur berikut yang berlaku untuk kebutuhan Anda.

Untuk memperbarui setelan definisi model Anda

1. Buat definisi model baru dengan pengaturan yang diperbarui dengan menggunakan CreateModel tindakan di Amazon SageMaker API.
2. Buat konfigurasi endpoint baru yang menggunakan definisi model baru. Untuk melakukan ini, gunakan CreateEndpointConfig tindakan di Amazon SageMaker API.
3. Perbarui titik akhir Anda dengan konfigurasi titik akhir baru sehingga pengaturan definisi model yang diperbarui diterapkan.
4. (Opsional) Hapus konfigurasi titik akhir lama jika Anda tidak menggunakannya dengan titik akhir lainnya. Anda juga dapat menghapus sumber daya yang Anda tentukan dalam definisi model

jika Anda tidak menggunakannya dengan titik akhir lainnya. Sumber daya ini mencakup artefak model di Amazon S3 dan gambar inferensi.

Untuk memperbarui konfigurasi titik akhir Anda

1. Buat konfigurasi titik akhir baru dengan pengaturan yang diperbarui.
2. Perbarui titik akhir Anda dengan konfigurasi baru sehingga pembaruan Anda berlaku.
3. (Opsional) Hapus konfigurasi titik akhir lama jika Anda tidak menggunakannya dengan titik akhir lainnya. Anda juga dapat menghapus sumber daya yang Anda tentukan dalam definisi model jika Anda tidak menggunakannya dengan titik akhir lainnya. Sumber daya ini mencakup artefak model di Amazon S3 dan gambar inferensi.

Setiap kali Anda membuat definisi model baru atau konfigurasi titik akhir, kami sarankan Anda menggunakan nama yang unik. Jika Anda ingin memperbarui sumber daya ini dan mempertahankan nama aslinya, gunakan prosedur berikut.

Untuk memperbarui pengaturan model Anda dan mempertahankan nama model asli

1. Hapus definisi model yang ada. Pada titik ini, titik akhir apa pun yang menggunakan model rusak, tetapi Anda memperbaikinya dalam langkah-langkah berikut.
2. Buat definisi model lagi dengan pengaturan yang diperbarui, dan gunakan nama model yang sama.
3. Buat konfigurasi endpoint baru yang menggunakan definisi model yang diperbarui.
4. Perbarui titik akhir Anda dengan konfigurasi titik akhir yang baru sehingga pembaruan Anda berlaku.

Untuk memperbarui konfigurasi titik akhir Anda dan mempertahankan nama konfigurasi asli

1. Hapus konfigurasi endpoint yang ada.
2. Buat konfigurasi titik akhir baru dengan pengaturan yang diperbarui, dan gunakan nama aslinya.
3. Perbarui titik akhir Anda dengan konfigurasi baru sehingga pembaruan Anda berlaku.

Fitur yang didukung

Amazon SageMaker menawarkan empat opsi berikut untuk menerapkan model untuk inferensi.

- Inferensi real-time untuk beban kerja inferensi dengan persyaratan latensi rendah real-time, interaktif, dan rendah.
- Batch transform untuk inferensi offline dengan dataset besar.
- Inferensi asinkron untuk near-real-time inferensi dengan input besar yang membutuhkan waktu preprocessing lebih lama.
- Inferensi tanpa server untuk beban kerja inferensi yang memiliki periode idle antara semburan lalu lintas.

Tabel berikut merangkum fitur platform inti yang didukung oleh setiap opsi inferensi. Ini tidak menunjukkan fitur yang dapat disediakan oleh kerangka kerja, kontainer Docker kustom, atau melalui rantai layanan yang berbeda. AWS

Fitur	Inferensi waktu nyata	Transformasi batch	Inferensi asinkron	Inferensi tanpa server	kontainer Docker
Dukungan penskalaan otomatis	✓	T/A	✓	✓	T/A
Dukungan GPU	✓ [1]	✓ [1]	✓ [1]		1P, pra-dibangun, BYOC
Model tunggal	✓	✓	✓	✓	T/A
Titik akhir multi-model	✓				K-nn, XGBoost, Pelajar Linear, RCF,, Apache MXNet,, scikit-belajar 2 TensorFlow PyTorch
Titik akhir multi-kontainer	✓				1P, pra-dibangun, Perpanjang pra-dibangun, BYOC

Fitur	Inferensi waktu nyata	Transformasi batch	Inferensi asinkron	Inferensi tanpa server	kontainer Docker
Pipa inferensi serial	✓	✓			1P, pra-dibangun, Perpanjang pra-dibangun, BYOC
Inferensi Recommender	✓				1P, pra-dibangun, Perpanjang pra-dibangun, BYOC
Dukungan tautan pribadi	✓	✓	✓		T/A
Dukungan penangkapan data/Model monitor	✓	✓			T/A
DLC didukung	1P, pra-dibangun, Perpanjang pra-dibangun, BYOC	1P, pra-dibangun, Perpanjang pra-dibangun, BYOC	1P, pra-dibangun, Perpanjang pra-dibangun, BYOC	1P, pra-dibangun, Perpanjang pra-dibangun, BYOC	T/A
Protokol didukung	HTTP	HTTP	HTTP	HTTP	T/A
Ukuran muatan	< 6 MB	≤ 100 MB	≤ 1 GB	≤ 4 MB	

Fitur	Inferensi waktu nyata	Transformasi batch	Inferensi asinkron	Inferensi tanpa server	kontainer Docker
HTTP chunked encoding	Framework dependen, 1P tidak didukung	T/A	Framework dependen, 1P tidak didukung	Framework dependen, 1P tidak didukung	T/A
Batas waktu permintaan	<60 detik	Hari	< 1 jam	<60 detik	T/A
Pagar pembatas penyebaran: penyebaran biru/hijau	✓	T/A	✓		T/A
Pagar pembatas penyebaran: penyebaran bergulir	✓	T/A	✓		T/A
Pengujian bayangan	✓				T/A
Skala ke nol		T/A	✓	✓	T/A
Dukungan paket model pasar	✓	✓			T/A
Dukungan cloud pribadi virtual	✓	✓	✓		T/A

Fitur	Inferensi waktu nyata	Transformasi batch	Inferensi asinkron	Inferensi tanpa server	kontainer Docker
Beberapa varian produksi mendukung	✓				T/A
Isolasi jaringan	✓		✓		T/A
Dukungan penyajian paralel model	✓ [3]	✓	✓ [3]		✓ [3]
Enkripsi volume	✓	✓	✓	✓	T/A
Pelanggan AWS KMS	✓	✓	✓	✓	T/A
d dukungan misalnya	✓	✓	✓		T/A
dukungan inf1	✓				✓

Dengan SageMaker itu, Anda dapat menerapkan satu model, atau beberapa model di belakang titik akhir inferensi tunggal untuk inferensi waktu nyata. Tabel berikut merangkum fitur inti yang didukung oleh berbagai opsi hosting yang datang dengan inferensi waktu nyata.

Fitur	Titik akhir model tunggal	Titik akhir multi-model	Pipa inferensi serial	Titik akhir multi-kontainer
Dukungan penskalaan otomatis	✓	✓	✓	✓

Fitur	Titik akhir model tunggal	Titik akhir multi-model	Pipa inferensi serial	Titik akhir multi-kontainer
Dukungan GPU	✓ [1]	✓	✓	
Model tunggal	✓	✓	✓	✓
Titik akhir multi-model		✓	✓	T/A
Titik akhir multi-kontainer	✓			T/A
Pipa inferensi serial	✓	✓	T/A	
Inferensi Recommender	✓			
Dukungan tautan pribadi	✓	✓	✓	✓
Dukungan penangkapan data/Model monitor	✓	T/A	N/A	T/A
DLC didukung	1P, pra-dibangun, Perpanjang pra-dibangun, BYOC	K-nn, XGBoost, Pelajar Linear, RCF,, Apache MXNet,, scikit-belajar 2 TensorFlow PyTorch	1P, pra-dibangun, Perpanjang pra-dibangun, BYOC	1P, pra-dibangun, Perpanjang pra-dibangun, BYOC
Protokol didukung	HTTP	HTTP	HTTP	HTTP
Ukuran muatan	< 6 MB	< 6 MB	< 6 MB	< 6 MB

Fitur	Titik akhir model tunggal	Titik akhir multi-model	Pipa inferensi serial	Titik akhir multi-kontainer
Batas waktu permintaan	<60 detik	<60 detik	<60 detik	<60 detik
Pagar pembatas penyebaran: penyebaran biru/hijau	✓	✓	✓	✓
Pagar pembatas penyebaran: penyebaran bergulir	✓	✓	✓	✓
Pengujian bayangan	✓			
Dukungan paket model pasar	✓			
Dukungan cloud pribadi virtual	✓	✓	✓	✓
Beberapa varian produksi mendukung	✓		✓	✓
Isolasi jaringan	✓	✓	✓	✓
Dukungan penyajian paralel model	✓ [3]		✓ [3]	
Enkripsi volume	✓	✓	✓	✓
Pelanggan AWS KMS	✓	✓	✓	✓

Fitur	Titik akhir model tunggal	Titik akhir multi-model	Pipa inferensi serial	Titik akhir multi-kontainer
d dukungan misalnya	✓	✓	✓	✓
dukungan inf1	✓			

[1] Ketersediaan jenis instans Amazon EC2 bergantung pada AWS Wilayah. Untuk ketersediaan instans khususAWS, lihat SageMakerHarga Amazon.

[2] Untuk menggunakan kerangka kerja atau algoritme lainnya, gunakan toolkit SageMaker Inferensi untuk membuat wadah yang mendukung titik akhir multi-model.

[3] DenganSageMaker, Anda dapat menggunakan model besar (hingga 500 GB) untuk inferensi. Anda dapat mengonfigurasi pemeriksaan kesehatan kontainer dan mengunduh kuota batas waktu hingga 60 menit. Ini akan memungkinkan Anda memiliki lebih banyak waktu untuk mengunduh dan memuat model Anda dan sumber daya terkait. Untuk informasi selengkapnya, lihat SageMaker parameter titik akhir untuk inferensi model besar. Anda dapat menggunakan wadah Inferensi model besar yang SageMaker kompatibel. Anda juga dapat menggunakan perpustakaan paralelisasi model pihak ketiga, seperti Triton dengan dan. FasterTransformer DeepSpeed Anda harus memastikan bahwa mereka kompatibel denganSageMaker.

Sumber daya

Gunakan sumber daya berikut untuk pemecahan masalah dan referensi, menjawab FAQ, dan mempelajari lebih lanjut tentang AmazonSageMaker.

Topik

- Blog, contoh buku catatan, dan sumber daya tambahan
- Pemecahan masalah dan referensi
- FAQ Hosting Model

Blog, contoh buku catatan, dan sumber daya tambahan

Bagian berikut berisi contoh dan sumber daya tambahan bagi Anda untuk mempelajari lebih lanjut tentang AmazonSageMaker.

Blog dan studi kasus

Lihat tabel berikut untuk daftar blog dan studi kasus untuk berbagai fitur di dalamnyaSageMakerKesimpulan. Anda dapat menggunakan blog untuk membantu menyusun solusi yang sesuai untuk kasus penggunaan Anda.

Fitur	Sumber daya
Inferensi Real-Time	• Memulai penerapan model real-time di AmazonSageMaker • Menerapkan BLOOM-176B dan OPT-30B di AmazonSageMakerdengan model besar inferensi Deep Learning Containers danDeepSpeed • Membuat REST API yang didukung machine learning dengan template pemetaan Amazon API Gateway dan AmazonSageMaker
Penskalaan otomatis	• Mengonfigurasi titik akhir inferensi penskalaan otomatis di AmazonSageMaker
Inferensi Tanpa Server	• AmazonSageMakerInferensi Tanpa Server - Inferensi Pembelajaran Mesin tanpa Khawatir tentang Server • Host Memeluk model transformator Wajah menggunakan AmazonSageMakerInferensi Tanpa Server • Memperkenalkan AmazonSageMakerToolkit Pembandingan Inferensi Tanpa Server
Inferensi Asinkron	• Menjalankan inferensi penglihatan komputer pada video besar dengan AmazonSageMakertitik akhir asinkron • Buat solusi pemeliharaan prediktif dengan Amazon Kinesis,AWS Glue, dan AmazonSageMaker

Fitur	Sumber daya
	• Tingkatkan penelitian bernilai tinggi dengan Hugging Face dan AmazonSageMakerTitik akhir Inferensi Asinkron
Transformasi Batch	• Mengaitkan hasil prediksi dengan data input menggunakan AmazonSageMakerTransformasi Batch
Titik Akhir Multi-Model	• Menghemat biaya inferensi dengan menggunakan AmazonSageMakertitik akhir multi-model • Menjalankan beberapa model deep learning di GPU dengan AmazonSageMakertitik akhir multi-model • Cara mengukur inferensi pembelajaran mesin untuk kasus penggunaan SaaS multi-penyewa • Menjalankan dan mengoptimalkan inferensi multi-model dengan AmazonSageMakertitik akhir multi-model
Pipa Inferensi Serial	• Pola desain untuk inferensi serial di AmazonSageMaker
Titik Akhir Multi-Kontainer	• Inferensi ML-efisien biaya dengan model multi-framework di AmazonSageMaker
Menjalankan Ansambel Model	• Jalankan model ENSEMBLE di AmazonSageMaker

Fitur	Sumber daya
Inferensi Recommender	• SageMakerContoh notebook Inference Recommender • SageMakerInferensi Recommender untukHuggingFaceContoh notebook Analisis Sentimen BERT • Mencapai kinerja hyperscale untuk penyajian model menggunakan NVIDIA Triton Inference Server di AmazonSageMaker
Model lanjutan hosting seri blog	• Bagian 1: Pola desain umum untuk membangun aplikasi ML di AmazonSag eMaker • Bagian 2: Memulai dengan menerapkan model waktu nyataSageMaker • Bagian 3: Menjalankan dan mengoptimalkan inferensi multi-model dengan AmazonSag eMakertitik akhir multi-model • Bagian 4: Pola desain untuk inferensi serial di AmazonSageMaker • Bagian 5: Inferensi ML hemat biaya dengan model multi-framework di AmazonSag eMaker • Bagian 6: Praktik terbaik dalam menguji dan memperbarui modelSageMaker • Bagian 7: Jalankan model mL ansambel di AmazonSageMaker

Notebook contoh

Lihat tabel berikut misalnya notebook yang dapat membantu Anda mempelajari lebih lanjut tentangSageMakerKesimpulan.

Fitur	Notebook contoh
Inferensi Recommender	• SageMakerContoh notebook Inference Recommender • SageMakerInferensi Recommender untukHuggingFaceContoh notebook Analisis Sentimen BERT
Optimalkan model bahasa besar (LLMs) untukSageMaker	Lokakarya Generatif AI LLMs

Sumber daya tambahan

Untuk informasi lebih lanjut tentang masing-masingSageMakerOpsi inferensi secara rinci, Anda dapat menonton video berikut.

Menerapkan model ML untuk inferensi dengan kinerja tinggi dan biaya rendah

Pemecahan masalah dan referensi

Anda dapat menggunakan sumber daya berikut dan dokumentasi referensi untuk memahami praktik terbaik saat menggunakanSageMakerInferensi dan untuk memecahkan masalah dengan penerapan model:

- Untuk mengatasi masalah penerapan model, lihatMemecahkan masalah penerapan model Amazon SageMaker .
- Untuk praktik terbaik penerapan model, lihatPraktik terbaik.
- Untuk informasi referensi tentang ukuran volume penyimpanan yang disediakan untuk berbagai ukuran instans hosting, lihatVolume volume penyimpanan volume penyimpanan instans volume penyimpanan instans.
- Untuk informasi referensi tentangSageMakerbatas dan kuota, lihatAmazonSageMakerendpoint dan kuota.
- Untuk pertanyaan yang sering diajukan tentangSageMaker, lihatFAQ Hosting Model.

FAQ Hosting Model

Lihat item FAQ berikut untuk jawaban atas pertanyaan umum tentangSageMakerHosting inferensi.

Hosting Umum

Item FAQ berikut menjawab pertanyaan umum umum untukSageMakerKesimpulan.

T: Opsi penerapan apa yang dilakukan AmazonSageMakermenyediakan?

A: Setelah Anda membangun dan melatih model, AmazonSageMakermenyediakan empat opsi untuk menerapkannya sehingga Anda dapat mulai membuat prediksi. Inferensi Real-Time cocok untuk beban kerja dengan persyaratan latensi milidetik, ukuran muatan hingga 6 MB, dan waktu pemrosesan hingga 60 detik. Batch Transform sangat ideal untuk prediksi offline pada batch besar data yang tersedia di depan. Inferensi Asinkron dirancang untuk beban kerja yang tidak memiliki persyaratan latensi sub-detik, ukuran muatan hingga 1 GB, dan waktu pemrosesan hingga 15 menit. Dengan Inferensi Tanpa Server, Anda dapat dengan cepat menerapkan model machine learning untuk inferensi tanpa harus mengkonfigurasi atau mengelola infrastruktur yang mendasarinya, dan Anda hanya membayar untuk kapasitas komputasi yang digunakan untuk memproses permintaan inferensi, yang ideal untuk beban kerja intermiten.

T: Bagaimana cara memilih opsi penerapan modelSageMaker?

A: Diagram berikut dapat membantu Anda memilihSageMakerOpsi penyebaran model hosting.

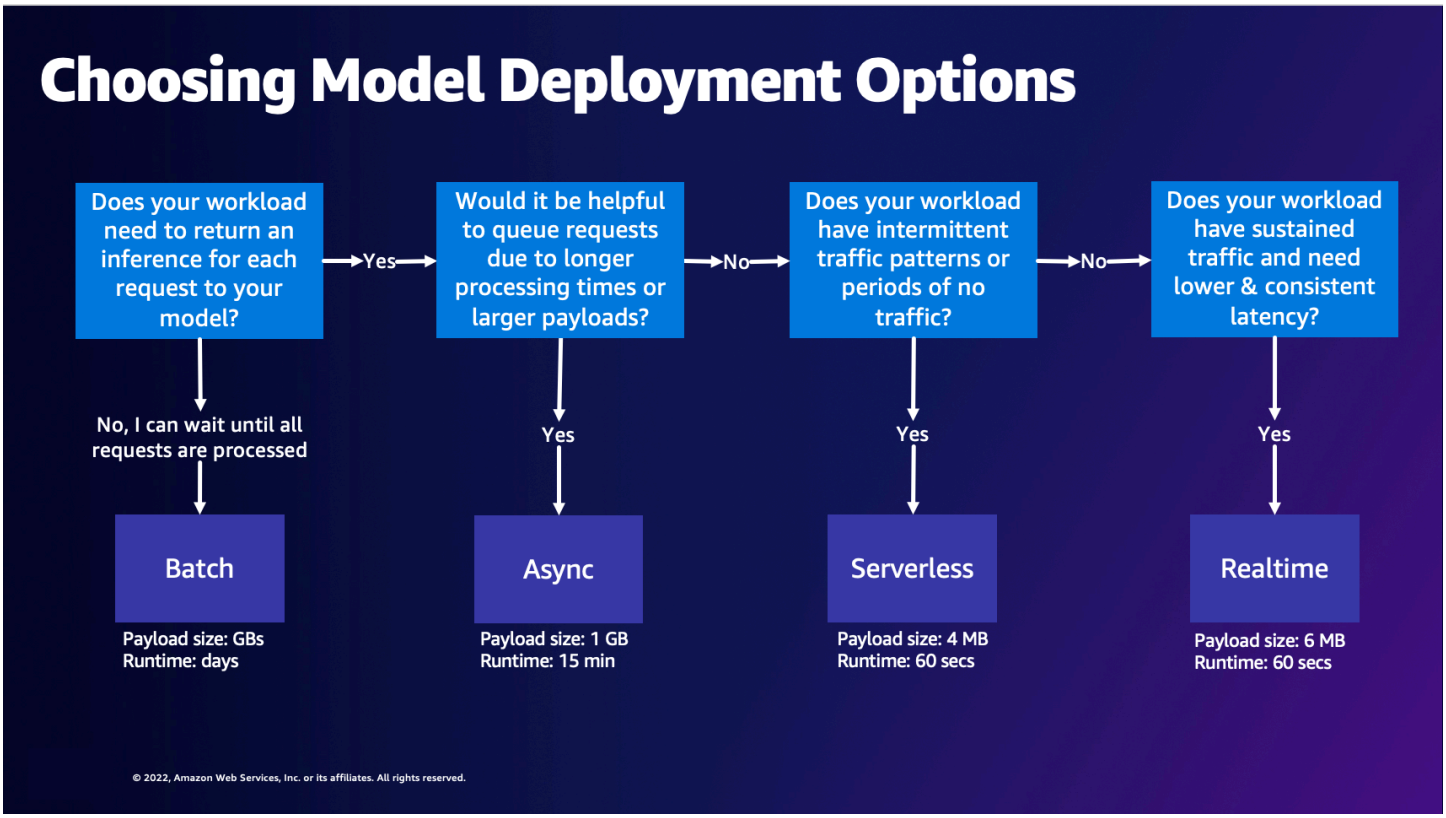


Diagram sebelumnya memandu Anda melalui proses keputusan berikut. Jika Anda ingin memproses permintaan dalam batch, Anda mungkin ingin memilih Batch Transform. Jika tidak, jika Anda ingin menerima inferensi untuk setiap permintaan ke model Anda, Anda mungkin ingin memilih Inferensi Asinkron, Inferensi Tanpa Server, atau Inferensi Real-Time. Anda dapat memilih Inferensi Asinkron jika Anda memiliki waktu pemrosesan yang lama atau muatan besar dan ingin mengantri permintaan. Anda dapat memilih Inferensi Tanpa Server jika beban kerja Anda memiliki lalu lintas yang tidak dapat diprediksi atau terputus-putus. Anda dapat memilih Inferensi Real-Time jika Anda memiliki lalu lintas berkelanjutan dan memerlukan latensi yang lebih rendah dan konsisten untuk permintaan Anda.

T: Saya pernah mendengarSageMakerInferensi itu mahal. Apa cara terbaik untuk mengoptimalkan biaya saya saat hosting model?

A: Untuk mengoptimalkan biaya AndaSageMakerInferensi, Anda harus memilih opsi hosting yang tepat untuk kasus penggunaan Anda. Anda juga dapat menggunakan fitur Inferensi seperti AmazonSageMakerPaket Tabungan, optimasi model dengan SageMakerNeo, Titik Akhir Multi-Model dan Titik Akhir Multi-Kontainer, atau autoscaling. Untuk tips tentang cara mengoptimalkan biaya Inferensi Anda, lihat Praktik terbaik pengoptimalan biaya inferensi.

T: Mengapa saya harus menggunakan AmazonSageMakerInferensi Recommender?

A: Anda harus menggunakan AmazonSageMakerInference Recommender jika Anda memerlukan rekomendasi untuk konfigurasi endpoint yang tepat untuk meningkatkan kinerja dan mengurangi biaya. Sebelumnya, ilmuwan data yang ingin menggunakan model mereka harus menjalankan tolok ukur manual untuk memilih konfigurasi titik akhir yang tepat. Pertama, mereka harus memilih jenis instance pembelajaran mesin yang tepat dari lebih dari 70 jenis instans yang tersedia berdasarkan persyaratan sumber daya model dan muatan sampel mereka, dan kemudian mengoptimalkan model untuk memperhitungkan perangkat keras yang berbeda. Kemudian, mereka harus melakukan uji beban ekstensif untuk memvalidasi bahwa persyaratan latensi dan throughput dipenuhi dan biayanya rendah. Inference Recommender menghilangkan kompleksitas ini dengan membantu Anda melakukan hal berikut:

- Mulai dalam hitungan menit dengan rekomendasi instans.
- Lakukan pengujian beban di seluruh jenis instans untuk mendapatkan rekomendasi tentang konfigurasi titik akhir Anda dalam beberapa jam.
- Secara otomatis menyetel parameter server kontainer dan model serta melakukan optimasi model untuk jenis instance tertentu.

T: Apa itu server model?

SEBUAH:SageMakerendpoint adalah endpoint HTTP REST yang menggunakan server web kemas, yang mencakup server model. Kontainer ini bertanggung jawab untuk memuat dan melayani permintaan untuk model pembelajaran mesin. Mereka mengimplementasikan server web yang merespons`/invocations`dan`/ping`pada port 8080.

Server model umum termasukTensorFlowMelayani,TorchServedan Multi Model Server. SageMakerwadah kerangka memiliki server model ini dibangun di.

T: Apa yang Membawa Kontainer Anda Sendiri dengan AmazonSageMaker?

A: Segala sesuatu diSageMakerInferensi dalam peti kemas. SageMakermenyediakan kontainer terkelola untuk kerangka kerja populer sepertiTensorFlow, SKLearn, danHuggingFace. Untuk daftar terbaru yang komprehensif dari gambar-gambar tersebut, lihatGambar yang Tersedia.

Terkadang ada kerangka kerja khusus yang mungkin Anda perlukan untuk membangun wadah. Pendekatan ini dikenal sebagaiBawa Kontainer Anda SendiriatauBYOC. Dengan pendekatan BYOC, Anda menyediakan image Docker untuk mengatur kerangka kerja atau pustaka Anda. Kemudian, Anda mendorong gambar ke Amazon Elastic Container Registry (Amazon ECR) sehingga Anda

dapat menggunakan gambar denganSageMaker. Untuk contoh pendekatan BYOC, lihatOverivew Kontainer untuk AmazonSageMaker.

Atau, alih-alih membangun gambar dari awal, Anda dapat memperpanjang wadah. Anda dapat mengambil salah satu gambar dasar ituSageMakermenyediakan dan menambahkan dependensi Anda di atasnya di Dockerfile Anda.

T: Apakah saya perlu melatih model sayaSageMakeruntuk menjadi tuan rumah merekaSageMakertitik akhir?

SEBUAH:SageMakermenawarkan kapasitas untuk membawa model kerangka kerja terlatih Anda sendiri yang telah Anda latih di luarSageMakerdan menyebarkannya pada salah satuSageMakerpilihan hosting.

SageMakermengharuskan Anda untuk mengemas model dalam`model.tar.gz`file dan memiliki struktur direktori tertentu. Setiap kerangka memiliki struktur model sendiri (lihat pertanyaan berikut misalnya struktur). Untuk informasi lebih lanjut, lihatSageMakerDokumentasi Python SDK untukTensorFlow,PyTorch, danMXNet.

Meskipun Anda dapat memilih dari gambar kerangka prebuilt sepertiTensorFlow,PyTorch, dan MXNet untuk menjadi tuan rumah model terlatih Anda, Anda juga dapat membangun wadah Anda sendiri untuk meng-host model terlatih AndaSageMakertitik akhir. Untuk panduan, lihat contoh notebook JupyterMembangun wadah algoritme Anda sendiri.

T: Bagaimana saya harus menyusun model saya jika saya ingin menggunakanSageMakertapi tidak melatihSageMaker?

SEBUAH:SageMakermembutuhkan artefak model Anda untuk dikompresi dalam`.tar.gz`file, atautarball. SageMakersecara otomatis mengekstrak ini`.tar.gz`file ke`/opt/ml/model/`direktori dalam wadah Anda. Tarball tidak boleh berisi symlink atau file unncessary. Jika Anda menggunakan salah satu wadah kerangka kerja, sepertiTensorFlow,PyTorch, atau MXNet, wadah mengharapkan struktur TAR Anda menjadi sebagai berikut:

TensorFlow

```
model.tar.gz/
             |--[model_version_number]/
                                      |--variables
                                      |--saved_model.pb
            code/
                |--inference.py
```

```
                |--requirements.txt
```

PyTorch

```
model.tar.gz/
             |- model.pth
             |- code/
                     |- inference.py
                     |- requirements.txt  # only for versions 1.3.1 and higher
```

MXNet

```
model.tar.gz/
            |- model-symbol.json
            |- model-shapes.json
            |- model-0000.params
            |- code/
                    |- inference.py
                    |- requirements.txt # only for versions 1.6.0 and higher
```

T: Saat memohonSageMakerendpoint, saya dapat memberikan**`ContentType`**dan**`Accept`**Jenis MIME. Yang mana yang digunakan untuk mengidentifikasi tipe data yang dikirim dan diterima?

SEBUAH:`ContentType`adalah tipe MIME dari data input dalam badan permintaan (tipe MIME dari data yang Anda kirim ke titik akhir Anda). Server model menggunakan`ContentType`untuk menentukan apakah dapat menangani jenis yang disediakan atau tidak.

`Accept`adalah tipe MIME dari respons inferensi (tipe MIME dari data yang dikembalikan titik akhir Anda). Server model menggunakan`Accept`ketik untuk menentukan apakah dapat menangani mengembalikan jenis yang disediakan atau tidak.

Jenis MIME yang umum termasuk`text/csv`,`application/json`, dan`application/jsonlines`.

T: Untuk apa format data yang didukungSageMakerInferensi?

SEBUAH:SageMakermelewati permintaan apa pun ke wadah model tanpa modifikasi. Wadah harus berisi logika untuk deserialize permintaan. Untuk informasi tentang format yang ditetapkan untuk algoritma bawaan, lihatFormat Data Umum untuk Inferensi. Jika Anda sedang membangun wadah Anda sendiri atau menggunakanSageMakerFramework container, Anda dapat menyertakan logika untuk menerima format permintaan pilihan Anda.

Demikian pula,SageMakerjuga mengembalikan respon tanpa modifikasi, dan kemudian klien harus deserialize respon. Dalam kasus algoritma bawaan, mereka mengembalikan tanggapan dalam format tertentu. Jika Anda sedang membangun wadah Anda sendiri atau menggunakanSageMakerFramework container, Anda dapat menyertakan logika untuk mengembalikan respons dalam format yang Anda pilih.

T: Bagaimana cara memanggil titik akhir saya dengan data biner seperti video atau gambar?

GunakanMemanggil EndpointPanggilan API untuk membuat inferensi terhadap titik akhir Anda.

Ketika melewati masukan Anda sebagai payload ke`InvokeEndpoint`API, Anda harus memberikan jenis data input yang benar yang diharapkan model Anda. Saat melewati muatan di`InvokeEndpoint`API panggilan, byte permintaan diteruskan langsung ke wadah model. Misalnya, untuk gambar, Anda dapat menggunakan`application/jpeg`untuk`ContentType`, dan pastikan bahwa model Anda dapat melakukan inferensi pada jenis data ini. Ini berlaku untuk JSON, CSV, video, atau jenis masukan lain yang mungkin Anda hadapi.

Faktor lain yang perlu dipertimbangkan adalah batas ukuran payload. Dalam hal titik akhir real-time dan tanpa server, batas payload adalah 6 MB. Anda dapat membagi video Anda menjadi beberapa bingkai dan memanggil titik akhir dengan setiap bingkai satu per satu. Atau, jika kasus penggunaan memungkinkan, Anda dapat mengirim seluruh video dalam payload menggunakan titik akhir asinkron, yang mendukung muatan hingga 1 GB.

Untuk contoh yang menampilkan cara menjalankan inferensi penglihatan komputer pada video berukuran besar dengan Asynchronous Inference, lihat iniposting blog.

Inferensi Real-Time

Item FAQ berikut menjawab pertanyaan umum untukSageMakerInferensi Real-Time.

T: Bagaimana cara membuatSageMakertitik akhir?

A: Anda dapat membuatSageMakerendpoint melaluiAWS-didukung perkakas sepertiAWSSDK,SageMakerPython SDK,AWS Management Console,AWS CloudFormation, danAWS Cloud Development Kit (AWS CDK).

Ada tiga entitas kunci dalam pembuatan endpoint: aSageMakermodel, sebuahSageMakerkonfigurasi endpoint, danSageMakertitik akhir. YangSageMakermodel menunjuk ke arah data model dan gambar yang Anda gunakan. Konfigurasi endpoint menentukan varian produksi Anda, yang mungkin mencakup jenis instans dan jumlah instans. Anda kemudian dapat

menggunakancreate_endpointPanggilan API atau.menyebarkan ()panggilan untukSageMakeruntuk membuat endpoint menggunakan metadata dari model dan konfigurasi endpoint Anda.

T: Apakah saya perlu menggunakanSageMakerPython SDK untuk membuat/memanggil titik akhir?

A: Tidak, Anda dapat menggunakan berbagaiAWSSDK (lihatMemanggil/Buatuntuk SDK yang tersedia) atau bahkan memanggil API web yang sesuai secara langsung.

T: Apa perbedaan antara Multi-Model Endpoints (MME) dan Multi Model Server (MMS)?

A: Endpoint Multi-Model adalah opsi Inferensi Real-Time yangSageMakermenyediakan. Dengan Endpoint Multi-Model, Anda dapat menampung ribuan model di belakang satu titik akhir. Server Multi Modeladalah kerangka open-source untuk melayani model machine learning. Ini menyediakan kemampuan manajemen front-end dan model HTTP yang diperlukan oleh titik akhir multi-model untuk meng-host beberapa model dalam satu kontainer, memuat model ke dalam dan membongkar model keluar dari wadah secara dinamis, dan melakukan inferensi pada model dimuat tertentu.

T: Apa saja arsitektur penerapan model yang berbeda yang didukung oleh Inferensi Real-Time?

SEBUAH:SageMakerInferensi Real-Time mendukung berbagai arsitektur penerapan model seperti Endpoint Multi-Model, Endpoint Multi-Container, dan Serial Inference Pipelines.

Titik Akhir Multi-Model (MME)- MME memungkinkan pelanggan untuk menyebarkan 1000s model hiper-personalisasi dengan cara yang hemat biaya. Semua model dikerahkan pada armada sumber daya bersama. MME bekerja paling baik ketika model memiliki ukuran dan latensi yang sama dan termasuk dalam kerangka kerja yang sama. Endpoint ini ideal ketika Anda tidak perlu memanggil model yang sama setiap saat. Anda dapat memuat model masing-masing secara dinamis keSageMakerendpoint untuk melayani permintaan Anda.

Titik Akhir Multi-Kontainer (MCE)- MCE memungkinkan pelanggan untuk menyebarkan 15 kontainer berbeda dengan kerangka kerja dan fungsi yang beragam tanpa cold start sementara hanya menggunakan satuSageMakertitik akhir. Anda dapat langsung memanggil kontainer ini. MCE adalah yang terbaik ketika Anda ingin menyimpan semua model dalam memori.

Pipa Inferensi Serial (SIP)- Anda dapat menggunakan SIP untuk menghubungkan 2-15 kontainer pada satu titik akhir. SIP sebagian besar cocok untuk menggabungkan preprocessing dan inferensi model dalam satu titik akhir dan untuk operasi latensi rendah.

Inferensi Tanpa Server

Item FAQ berikut menjawab pertanyaan umum untuk AmazonSageMakerInferensi Tanpa Server.

T: Apa itu AmazonSageMakerInferensi Tanpa Server?

SEBUAH:Inferensi Tanpa Serveradalah opsi penyajian model tanpa server yang dibuat khusus yang membuatnya mudah untuk menerapkan dan menskalakan model mL. Titik akhir Inferensi Tanpa Server secara otomatis memulai sumber daya komputasi dan menskalakannya masuk dan keluar tergantung pada lalu lintas, menghilangkan kebutuhan Anda untuk memilih jenis instans, menjalankan kapasitas yang disediakan, atau mengelola penskalaan. Anda dapat secara opsional menentukan persyaratan memori untuk titik akhir tanpa server Anda. Anda hanya membayar selama menjalankan kode inferensi dan jumlah data yang diproses, bukan untuk periode idle.

T: Mengapa saya harus menggunakan Inferensi Tanpa Server?

J: Inferensi Tanpa Server menyederhanakan pengalaman pengembang dengan menghilangkan kebutuhan untuk menyediakan kapasitas di depan dan mengelola kebijakan penskalaan. Inferensi Tanpa Server dapat menskalakan secara instan dari puluhan hingga ribuan inferensi dalam hitungan detik berdasarkan pola penggunaan, sehingga ideal untuk aplikasi ML dengan lalu lintas intermiten atau tidak dapat diprediksi. Misalnya, layanan chatbot yang digunakan oleh perusahaan pengolah gaji mengalami peningkatan pertanyaan pada akhir bulan sementara lalu lintas terputus-putus untuk sisa bulan itu. Menyediakan instans untuk seluruh bulan dalam skenario seperti itu tidak hemat biaya, karena Anda akhirnya membayar untuk periode idle.

Inferensi Tanpa Server membantu mengatasi jenis kasus penggunaan ini dengan memberi Anda penskalaan otomatis dan cepat di luar kotak tanpa perlu memperkirakan lalu lintas di depan atau mengelola kebijakan penskalaan. Selain itu, Anda hanya membayar waktu komputasi untuk menjalankan kode inferensi dan pemrosesan data, sehingga ideal untuk beban kerja dengan lalu lintas intermiten.

T: Bagaimana cara memilih ukuran memori yang tepat untuk endpoint tanpa server saya?

J: Endpoint tanpa server Anda memiliki ukuran RAM minimum 1024 MB (1 GB), dan ukuran RAM maksimum yang dapat Anda pilih adalah 6144 MB (6 GB). Ukuran memori yang dapat Anda pilih adalah 1024 MB, 2048 MB, 3072 MB, 4096 MB, 5120 MB, atau 6144 MB. Inferensi Tanpa Server secara otomatis menetapkan sumber daya komputasi yang sebanding dengan memori yang Anda pilih. Jika Anda memilih ukuran memori yang lebih besar, container Anda memiliki akses ke lebih banyak vCPU.

Pilih ukuran memori endpoint Anda sesuai dengan ukuran model Anda. Umumnya, ukuran memori harus setidaknya sebesar ukuran model Anda. Anda mungkin perlu melakukan benchmark untuk memilih pilihan memori yang tepat untuk model Anda berdasarkan SLA latensi Anda. Peningkatan

ukuran memori memiliki harga yang berbeda; lihatAmazonSageMakerHalaman hargauntuk informasi lebih lanjut.

Transformasi Batch

Item FAQ berikut menjawab pertanyaan umum untukSageMakerBatch Transform.

T: Bagaimana Batch Transform membagi data saya?

A: Untuk format file tertentu seperti CSV, RecordO dan TFRecord,SageMakerdapat membagi data Anda menjadi batch mini rekaman tunggal atau multi-record dan mengirimkannya sebagai payload ke wadah model Anda. Ketika nilai`BatchStrategy`adalah`MultiRecord`,SageMakermengirimkan jumlah maksimum catatan dalam setiap permintaan, hingga`MaxPayloadInMB`batas. Ketika nilai`BatchStrategy`adalah`SingleRecord`,SageMakermengirimkan catatan individu dalam setiap permintaan.

T: Berapa batas waktu maksimum untuk Batch Transform dan batas payload untuk satu record?

J: Batas waktu maksimum untuk Batch Transform adalah 3600 detik. Yangukuran muatan maksimumuntuk catatan (per batch mini) adalah 100 MB.

T: Bagaimana cara mempercepat pekerjaan Batch Transform?

A: Jika Anda menggunakan`CreateTransformJob`API, Anda dapat mengurangi waktu yang diperlukan untuk menyelesaikan pekerjaan transformasi batch dengan menggunakan nilai optimal untuk parameter seperti`MaxPayloadInMB`,`MaxConcurrentTransforms`, atau`BatchStrategy`. Nilai ideal untuk`MaxConcurrentTransforms`sama dengan jumlah pekerja komputasi dalam pekerjaan batch transform. Jika Anda menggunakanSageMakerkonsol, Anda dapat menentukan nilai parameter optimal ini diKonfigurasi tambahanbagian dariBatch mengubah konfigurasi pekerjaanhalaman. SageMakersecara otomatis menemukan pengaturan parameter optimal untuk algoritma bawaan. Untuk algoritma kustom, memberikan nilai-nilai ini melaluieksekusi-parametertitik akhir.

T: Apa format data yang didukung secara native di Batch Transform?

A: Batch Transform mendukung CSV dan JSON.

Inferensi Asinkron

Item FAQ berikut menjawab pertanyaan umum umum untukSageMakerInferensi Asinkron.

T: Apa itu AmazonSageMakerInferensi Asynchronous?

J: Inferensi Asinkron mengantri permintaan masuk dan memprosesnya secara asinkron. Opsi ini sangat ideal untuk permintaan dengan ukuran muatan besar atau waktu pemrosesan yang lama yang perlu diproses saat tiba. Secara opsional, Anda dapat mengonfigurasi pengaturan penskalaan otomatis untuk menurunkan jumlah instans menjadi nol saat tidak memproses permintaan secara aktif.

T: Bagaimana cara menskalakan titik akhir saya menjadi 0 saat tidak ada lalu lintas?

SEBUAH: AmazonSageMakermendukung penskalaan otomatis (autoscaling) titik akhir asinkron Anda. Penskalaan otomatis secara dinamis menyesuaikan jumlah instance yang disediakan untuk model sebagai respons terhadap perubahan beban kerja Anda. Tidak seperti model host lainnyaSageMakermendukung, dengan Inferensi Asinkron, Anda juga dapat menurunkan skala instans titik akhir asinkron Anda menjadi nol. Permintaan yang diterima saat ada nol instans akan diantrekan untuk diproses setelah titik akhir bertambah. Untuk informasi lebih lanjut, lihatSkala otomatis titik akhir asinkron.

AmazonSageMakerInferensi Tanpa Server juga secara otomatis menurunkan skala ke nol. Anda tidak akan melihat ini karenaSageMakermengelola penskalaan titik akhir tanpa server Anda, tetapi jika Anda tidak mengalami lalu lintas, infrastruktur yang sama berlaku.

Melaksanakan

Amazon SageMaker mendukung fitur untuk mengimplementasikan model pembelajaran mesin di lingkungan produksi dengan integrasi dan penerapan berkelanjutan. Topik berikut memberikan informasi tentang cara menyiapkan infrastruktur MLOP saat menggunakan. SageMaker

Topik

Mengapa Anda Harus Menggunakan MLOP?

Saat Anda beralih dari menjalankan proyek kecerdasan buatan dan pembelajaran mesin (AI/ML) individu ke menggunakan AI/ML untuk mengubah bisnis Anda dalam skala besar, disiplin Operasi ML/MLOP dapat membantu. MLOP menjelaskan aspek unik proyek AI/ML dalam manajemen proyek, CI/CD, dan jaminan kualitas, membantu Anda meningkatkan waktu pengiriman, mengurangi cacat, dan membuat ilmu data lebih produktif. MLOPs mengacu pada metodologi yang dibangun di atas penerapan DevOps praktik untuk beban kerja pembelajaran mesin. Untuk pembahasan tentang DevOps prinsip, lihat white paper Pengantar DevOps tentang AWS. Untuk mempelajari lebih lanjut tentang implementasi menggunakan AWS layanan, lihat Mempraktikkan CI/CD AWS dan Infrastruktur sebagai Kode.

Seperti DevOps, MLOP bergantung pada pendekatan kolaboratif dan efisien untuk siklus hidup pengembangan pembelajaran mesin di mana persimpangan orang, proses, dan teknologi mengoptimalkan end-to-end aktivitas yang diperlukan untuk mengembangkan, membangun, dan mengoperasikan beban kerja pembelajaran mesin.

MLOPs berfokus pada persimpangan ilmu data dan rekayasa data dalam kombinasi dengan DevOps praktik yang ada untuk merampingkan pengiriman model di seluruh siklus hidup pengembangan pembelajaran mesin. MLOPs adalah disiplin mengintegrasikan beban kerja ML ke dalam manajemen rilis, CI/CD, dan operasi. MLOP membutuhkan integrasi pengembangan perangkat lunak, operasi, rekayasa data, dan ilmu data.

Tantangan dengan MLOP

Meskipun MLOP dapat menyediakan alat yang berharga untuk membantu Anda meningkatkan skala bisnis Anda, Anda mungkin menghadapi masalah tertentu saat Anda mengintegrasikan MLOP ke dalam beban kerja pembelajaran mesin Anda.

Manajemen proyek

- Proyek ML melibatkan ilmuwan data, peran yang relatif baru, dan yang tidak sering diintegrasikan ke dalam tim lintas fungsi. Anggota tim baru ini sering berbicara bahasa teknis yang sangat berbeda dari pemilik produk dan insinyur perangkat lunak, menambah masalah yang biasa dalam menerjemahkan persyaratan bisnis ke dalam persyaratan teknis.

Komunikasi dan kolaborasi

- Membangun visibilitas pada proyek ML dan memungkinkan kolaborasi di berbagai pemangku kepentingan seperti insinyur data, ilmuwan data, insinyur ML, dan DevOps menjadi semakin penting untuk memastikan hasil yang sukses.

Semuanya adalah kode

- Penggunaan data produksi dalam kegiatan pengembangan, siklus hidup eksperimen yang lebih lama, ketergantungan pada jalur data, pelatihan ulang jalur pipa penyebaran, dan metrik unik dalam mengevaluasi kinerja model.
- Model sering memiliki siklus hidup independen dari aplikasi dan sistem yang terintegrasi dengan model tersebut.
- Seluruh end-to-end sistem dapat direproduksi melalui kode dan artefak berversi. DevOps proyek menggunakan Infrastructure-as-Code (IAc) dan Configuration-as-Code (CAC) untuk membangun lingkungan, dan Pipelines-as-Code (PAC) untuk memastikan pola CI/CD yang konsisten. Pipeline harus terintegrasi dengan alur kerja pelatihan Big Data dan ML. Itu sering berarti bahwa pipa adalah kombinasi dari alat CI/CD tradisional dan mesin alur kerja lainnya. Ada masalah kebijakan

penting untuk banyak proyek ML, sehingga pipeline mungkin juga perlu menegakkan kebijakan tersebut. Data input yang bias menghasilkan hasil yang bias, kekhawatiran yang meningkat bagi para pemangku kepentingan bisnis.

CI/CD

- Dalam MLOPs, data sumber adalah input kelas satu, bersama dengan kode sumber. Itu sebabnya MLOPs menyerukan pembuatan versi data sumber dan memulai pipeline berjalan saat data sumber atau inferensi berubah.
- Pipelines juga harus versi model ML, bersama dengan input dan output lainnya, untuk menyediakan ketertelusuran.
- Pengujian otomatis harus mencakup validasi yang tepat dari model ML selama fase pembuatan dan ketika model dalam produksi.
- Fase membangun dapat mencakup pelatihan model dan pelatihan ulang, proses yang memakan waktu dan intensif sumber daya. Pipa harus cukup granular untuk hanya melakukan siklus pelatihan penuh ketika data sumber atau kode ML berubah, bukan ketika komponen terkait berubah.
- Karena kode pembelajaran mesin biasanya merupakan bagian kecil dari solusi keseluruhan, pipeline penerapan juga dapat menggabungkan langkah-langkah tambahan yang diperlukan untuk mengemas model untuk konsumsi sebagai API oleh aplikasi dan sistem lain.

Pencatatan dan pemantauan

- Fase rekayasa fitur dan pelatihan model yang diperlukan untuk menangkap metrik pelatihan model serta eksperimen model. Menyetel model ML membutuhkan manipulasi bentuk data input serta hiperparameter algoritma, dan secara sistematis menangkap eksperimen tersebut. Pelacakan eksperimen membantu ilmuwan data bekerja lebih efektif dan memberikan gambaran yang dapat direproduksi dari pekerjaan mereka.
- Model ML yang diterapkan memerlukan pemantauan data yang diteruskan ke model untuk inferensi, bersama dengan stabilitas titik akhir standar dan metrik kinerja. Sistem pemantauan juga harus menangkap kualitas output model, sebagaimana dievaluasi oleh metrik ML yang sesuai.

Manfaat

Mengadopsi praktik MLOP memberi Anda lebih cepat time-to-market untuk proyek ML dengan memberikan manfaat berikut.

- Produktivitas: Menyediakan lingkungan swalayan dengan akses ke kumpulan data yang dikuratori memungkinkan insinyur data dan ilmuwan data bergerak lebih cepat dan membuang lebih sedikit waktu dengan data yang hilang atau tidak valid.
- Pengulangan: Mengotomatiskan semua langkah dalam MDC membantu Anda memastikan proses yang dapat diulang, termasuk bagaimana model dilatih, dievaluasi, diversi, dan digunakan.
- Keandalan: Menggabungkan praktik CI/CD memungkinkan kemampuan untuk tidak hanya menyebarkan dengan cepat tetapi dengan peningkatan kualitas dan konsistensi.
- Auditabilitas: Membuat versi semua input dan output, dari eksperimen ilmu data hingga sumber data hingga model terlatih, berarti bahwa kami dapat menunjukkan dengan tepat bagaimana model dibangun dan di mana model itu digunakan.
- Kualitas data dan model: MLOPs memungkinkan kami menegakkan kebijakan yang menjaga terhadap bias model dan melacak perubahan pada properti statistik data dan kualitas model dari waktu ke waktu.

SageMaker Eksperimen

Pembuatan model ML membutuhkan banyak iterasi pelatihan saat Anda menyetel algoritme, arsitektur model, dan parameter untuk mencapai akurasi prediksi yang tinggi. Anda dapat melacak input dan output di seluruh iterasi pelatihan ini untuk meningkatkan pengulangan uji coba dan kolaborasi dalam tim Anda menggunakan Eksperimen Amazon. SageMaker Anda juga dapat melacak parameter, metrik, kumpulan data, dan artefak lain yang terkait dengan pekerjaan pelatihan model Anda. SageMaker Eksperimen menawarkan satu antarmuka di mana Anda dapat memvisualisasikan pekerjaan pelatihan yang sedang berlangsung, berbagi eksperimen dalam tim Anda, dan menerapkan model langsung dari eksperimen.

Untuk mempelajari tentang SageMaker Eksperimen, lihatKelola Machine Learning dengan Amazon SageMaker Experiments.

SageMaker Alur kerja

Saat Anda menskalakan operasi machine learning (ML), Anda dapat menggunakan layanan alur kerja Amazon yang dikelola SageMaker sepenuhnya untuk menerapkan praktik integrasi dan penerapan berkelanjutan (CI/CD) untuk siklus hidup ML Anda. Dengan SageMaker Pipelines SDK, Anda memilih dan mengintegrasikan langkah-langkah pipeline ke dalam solusi terpadu yang mengotomatiskan proses pembuatan model dari persiapan data hingga penerapan model. Untuk arsitektur berbasis Kubernetes, Anda dapat menginstal SageMaker Operator di klaster Kubernetes Anda untuk membuat SageMaker pekerjaan secara native menggunakan API Kubernetes dan alat Kubernetes baris perintah seperti. `kubectl` Dengan SageMaker komponen untuk pipeline Kubeflow, Anda dapat membuat dan memantau SageMaker pekerjaan asli dari Pipelines Kubeflow Anda. Parameter pekerjaan, status, dan output dari dapat diakses dari SageMaker UI Pipelines Kubeflow. Terakhir, jika Anda ingin menjadwalkan batch run non-interaktif notebook Jupyter Anda, gunakan layanan alur kerja berbasis notebook untuk memulai proses mandiri atau reguler pada jadwal yang Anda tentukan.

Singkatnya, SageMaker menawarkan teknologi alur kerja berikut:

- Pipa Bangunan SageMaker Model Amazon: Alat untuk membangun dan mengelola jaringan pipa ML.
- Orkestrasi Kubernetes: operator SageMaker kustom untuk klaster Kubernetes Anda dan komponen untuk Pipelines Kubeflow.
- SageMaker Lowongan Notebook: Sesuai permintaan atau batch non-interaktif terjadwal dari notebook Jupyter Anda.

Anda juga dapat memanfaatkan layanan lain yang terintegrasi SageMaker untuk membangun alur kerja Anda. Opsi mencakup layanan berikut:

- Alur Kerja Alur Udara: SageMaker API untuk mengekspor konfigurasi untuk membuat dan mengelola alur kerja Alur Udara.
- AWS Step Functions: Alur kerja multi-langkah di Python yang mengatur SageMaker infrastruktur tanpa harus menyediakan sumber daya Anda secara terpisah.

Untuk informasi selengkapnya tentang mengelola SageMaker pelatihan dan inferensi, lihat Alur Kerja SDK Amazon SageMaker Python.

Topik

Pipa Bangunan SageMaker Model Amazon

Amazon SageMaker Model Building Pipelines adalah alat untuk membangun pipa pembelajaran mesin yang memanfaatkan integrasi langsung SageMaker . Karena integrasi ini, Anda dapat membuat pipeline dan menyiapkan SageMaker Proyek untuk orkestrasi menggunakan alat yang menangani banyak pembuatan dan manajemen langkah untuk Anda. Anda dapat membangun pipeline menggunakan SageMaker Python SDK, atau Anda dapat membuat pipeline menggunakan Pipeline Definition JSON SageMaker Schema.

SageMaker Pipelines memberikan keuntungan berikut dibandingkan penawaran AWS alur kerja lainnya:

SageMaker Integrasi

SageMaker Pipelines terintegrasi langsung dengan SageMaker, sehingga Anda tidak perlu berinteraksi dengan AWS layanan lain. Anda juga tidak perlu mengelola sumber daya apa pun karena SageMaker Pipelines adalah layanan yang dikelola sepenuhnya, yang berarti ia menciptakan dan mengelola sumber daya untuk Anda.

SageMaker Integrasi Python SDK

Karena SageMaker Pipelines terintegrasi dengan SageMaker Python SDK, Anda dapat membuat pipeline Anda secara terprogram menggunakan antarmuka Python tingkat tinggi yang mungkin sudah Anda kenal. Untuk melihat referensi SageMaker Python SDK API, lihat Pipelines. Untuk contoh kode SDK SageMaker Python, lihat Pipa Pembuatan SageMaker Model Amazon.

SageMaker Integrasi Studio

SageMaker Studio menawarkan lingkungan untuk mengelola pengalaman end-to-end SageMaker Pipelines. Menggunakan Studio, Anda dapat melewati AWS konsol untuk seluruh manajemen alur kerja Anda. Untuk informasi selengkapnya tentang mengelola SageMaker Pipelines dari SageMaker Studio, lihatLihat, Lacak, dan Jalankan SageMaker Pipelines di Studio SageMaker .

Pelacakan Silsilah Data

Dengan SageMaker Pipelines Anda dapat melacak riwayat data Anda dalam eksekusi pipeline. Amazon SageMaker ML Lineage Tracking memungkinkan Anda menganalisis dari mana datanya berasal, dari mana data tersebut digunakan sebagai input, dan output yang dihasilkan darinya. Misalnya, Anda dapat melihat model yang dibuat dari kumpulan data individual, dan Anda dapat melihat kumpulan data yang digunakan untuk membuat model individual. Untuk informasi selengkapnya, lihat Pelacakan SageMaker Silsilah Amazon.

Langkah Penggunaan Kembali

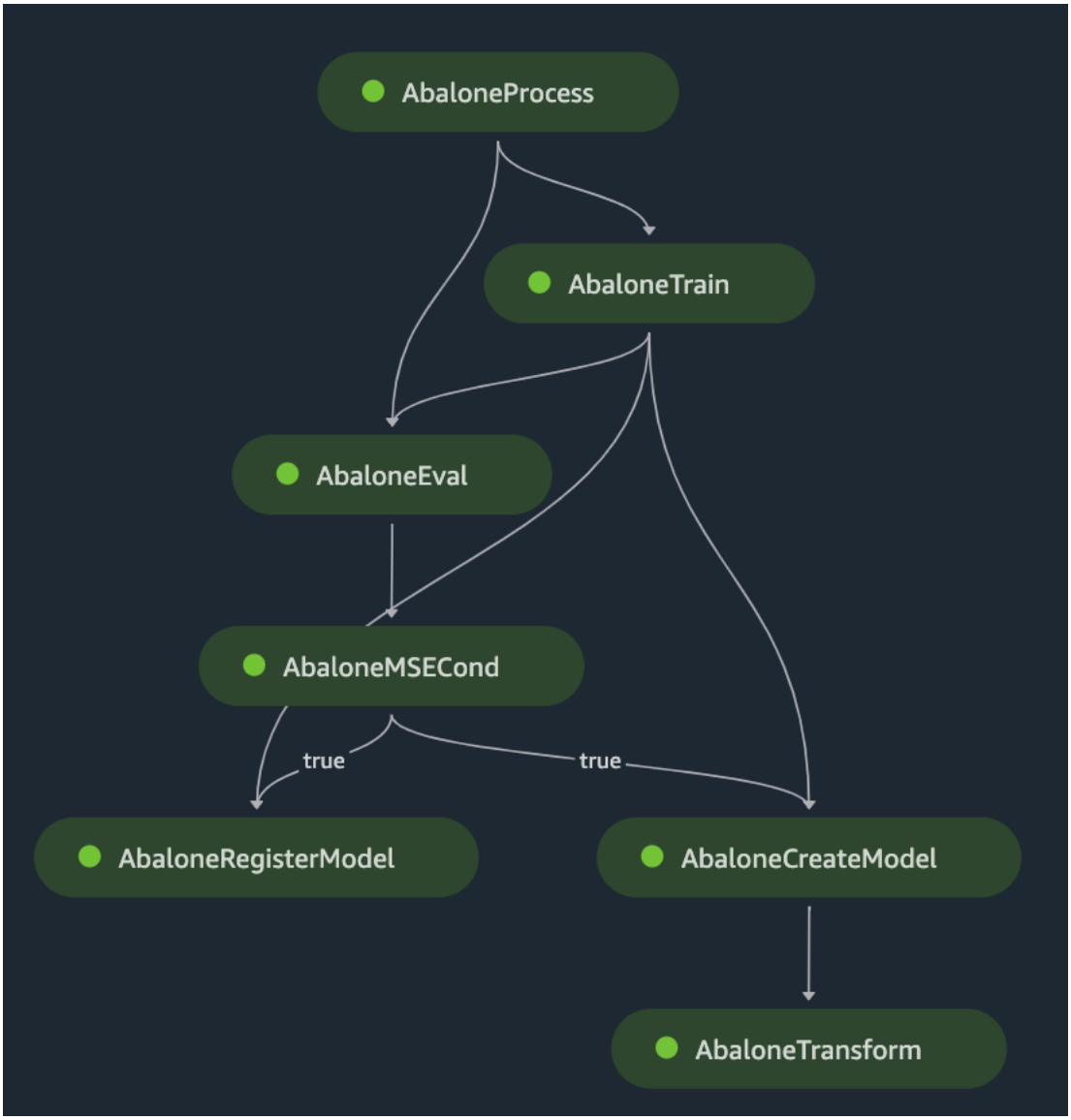
Dengan SageMaker Pipelines, Anda dapat menentukan langkah-langkah untuk caching. Ketika sebuah langkah di-cache, itu diindeks untuk digunakan kembali nanti jika langkah yang sama dijalankan lagi. Akibatnya, Anda dapat menggunakan kembali output dari eksekusi langkah sebelumnya dari langkah yang sama di pipeline yang sama tanpa harus menjalankan langkah lagi. Untuk informasi selengkapnya tentang langkah caching, lihatLangkah Alur Caching.

Topik

- SageMaker Ikhtisar Pipelines
- Membuat dan Mengelola SageMaker Pipelines

SageMaker Ikhtisar Pipelines

Pipeline Amazon SageMaker Model Building Pipelines adalah serangkaian langkah yang saling berhubungan yang didefinisikan menggunakan Pipelines SDK. Anda juga dapat membangun pipeline tanpa SDK menggunakan skema JSON definisi pipeline. Definisi pipeline ini mengkodekan pipeline menggunakan grafik asiklik terarah (DAG) yang dapat diekspor sebagai definisi JSON. DAG ini memberikan informasi tentang persyaratan dan hubungan antara setiap langkah pipa Anda. Struktur DAG pipa ditentukan oleh dependensi data antar langkah. Dependensi data ini dibuat ketika properti output langkah diteruskan sebagai input ke langkah lain. Gambar berikut adalah contoh dari pipa DAG:



Topik berikut menjelaskan konsep dasar SageMaker Pipelines. Untuk tutorial yang menjelaskan implementasi konsep-konsep ini, lihatMembuat dan Mengelola SageMaker Pipelines.

Topik

- Struktur dan Eksekusi Pipa
- Manajemen Akses IAM
- Dukungan Lintas Akun untuk Pipelines SageMaker
- Parameter Alur
- Membuat Alur
- L Kode ift-and-shift Python dengan dekorator @step

- Lulus Data Antar Langkah
- Langkah Alur Caching
- Coba lagi Kebijakan untuk Langkah-langkah Pipa
- Eksekusi selektif langkah-langkah pipa
- Perhitungan dasar, deteksi drift, dan siklus hidup dengan serta ClarifyCheck langkah-langkah QualityCheck di Amazon Model Building Pipelines SageMaker
- Jadwalkan Alur Berjalan
- Integrasi SageMaker Eksperimen Amazon
- Mode
- Memecahkan Masalah Pipa Pembuatan SageMaker Model Amazon

Struktur dan Eksekusi Pipa

Topik

- Struktur Alur
- Eksekusi Pipeline menggunakan Konfigurasi Paralelisme

Struktur Alur

Instans Amazon SageMaker Model Building Pipelines terdiri dari`name`,`parameters`, dan`steps`. Nama pipa harus unik dalam `(account, region)` pasangan. Semua parameter yang digunakan dalam definisi langkah harus didefinisikan dalam pipeline. Langkah-langkah pipeline yang tercantum secara otomatis menentukan urutan eksekusi mereka berdasarkan dependensi data mereka satu sama lain. Layanan SageMaker Pipelines menyelesaikan hubungan antara langkah-langkah dalam dependensi data DAG untuk membuat serangkaian langkah yang diselesaikan eksekusi. Berikut ini adalah contoh struktur pipa.

```
from sagemaker.workflow.pipeline import Pipeline

  pipeline_name = f"AbalonePipeline"
  pipeline = Pipeline(
      name=pipeline_name,
      parameters=[
          processing_instance_type,
          processing_instance_count,
          training_instance_type,
```

```
            model_approval_status,
            input_data,
            batch_data,
        ],
        steps=[step_process, step_train, step_eval, step_cond],
    )
```

Eksekusi Pipeline menggunakan Konfigurasi Paralelisme

Secara default, pipeline melakukan semua langkah yang tersedia untuk dijalankan secara paralel. Anda dapat mengontrol perilaku ini dengan menggunakan `ParallelismConfiguration` properti saat membuat atau memperbarui pipeline, serta saat memulai atau mencoba ulang eksekusi pipeline.

Konfigurasi paralelisme diterapkan per eksekusi. Misalnya, jika dua eksekusi dimulai, mereka masing-masing dapat menjalankan maksimum 50 langkah secara bersamaan, dengan total 100 langkah yang berjalan secara bersamaan. Juga, `ParallelismConfiguration` yang ditentukan saat memulai, mencoba kembali, atau memperbarui eksekusi lebih diutamakan daripada konfigurasi paralelisme yang ditentukan dalam pipeline.

Example Membuat eksekusi pipeline dengan **`ParallelismConfiguration`**

```
pipeline = Pipeline(
        name="myPipeline",
        steps=[step_process, step_train]
    )

  pipeline.create(role, parallelism_config={"MaxParallelExecutionSteps": 50})
```

Manajemen Akses IAM

Bagian berikut menjelaskan persyaratan AWS Identity and Access Management (IAM) untuk Amazon SageMaker Model Building Pipelines. Untuk contoh bagaimana Anda dapat menerapkan izin ini, lihatPrasyarat.

Topik

- Izin Peran Alur
- Izin Langkah Pipeline
- Sesuaikan manajemen akses untuk pekerjaan SageMaker Pipelines
- Kebijakan Kontrol Layanan dengan Pipa

Izin Peran Alur

Pipeline Anda memerlukan peran eksekusi pipeline IAM yang diteruskan ke SageMaker Pipelines saat Anda membuat pipeline. Peran untuk SageMaker instance yang membuat pipeline harus memiliki `iam:PassRole` izin untuk peran eksekusi pipeline untuk melewatinya. Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM.

Peran eksekusi pipeline Anda memerlukan izin berikut:

- Untuk meneruskan peran apa pun ke SageMaker pekerjaan dalam pipeline, `iam:PassRole` izin untuk peran yang sedang dilalui.
- `Create`dan `Describe` izin untuk masing-masing jenis pekerjaan di pipeline.
- Izin Amazon S3 untuk menggunakan fungsi ini. `JsonGet` Anda mengontrol akses ke sumber daya Amazon S3 Anda menggunakan kebijakan berbasis sumber daya dan kebijakan berbasis identitas. Kebijakan berbasis sumber daya diterapkan pada bucket Amazon S3 Anda dan memberikan SageMaker akses Pipelines ke bucket. Kebijakan berbasis identitas memberi pipeline Anda kemampuan untuk melakukan panggilan Amazon S3 dari akun Anda. Untuk informasi selengkapnya tentang kebijakan berbasis sumber daya dan kebijakan berbasis identitas, lihat Kebijakan berbasis identitas dan kebijakan berbasis sumber daya.

```
{
    "Action": [
        "s3:GetObject"
    ],
    "Resource": "arn:aws:s3:::<your-bucket-name>/*",
    "Effect": "Allow"
}
```

Izin Langkah Pipeline

SageMaker Pipelines termasuk langkah-langkah yang menjalankan SageMaker pekerjaan. Agar langkah pipeline dapat menjalankan pekerjaan ini, mereka memerlukan peran IAM di akun Anda yang menyediakan akses untuk sumber daya yang dibutuhkan. Peran ini diteruskan ke kepala SageMaker layanan oleh pipeline Anda. Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM.

Secara default, setiap langkah mengambil peran eksekusi pipeline. Anda secara opsional dapat meneruskan peran yang berbeda ke salah satu langkah dalam pipeline Anda. Ini memastikan bahwa kode di setiap langkah tidak memiliki kemampuan untuk memengaruhi sumber daya yang

digunakan dalam langkah lain kecuali ada hubungan langsung antara dua langkah yang ditentukan dalam definisi pipeline. Anda melewati peran ini saat mendefinisikan prosesor atau estimator untuk langkah Anda. Untuk contoh cara memasukkan peran ini dalam definisi ini, lihat dokumentasi SageMakerPython SDK.

Sesuaikan manajemen akses untuk pekerjaan SageMaker Pipelines

Anda dapat menyesuaikan kebijakan IAM lebih lanjut sehingga anggota yang dipilih di organisasi Anda dapat menjalankan salah satu atau semua langkah pipeline. Misalnya, Anda dapat memberikan izin kepada pengguna tertentu untuk membuat pekerjaan pelatihan, dan grup pengguna lain izin untuk membuat pekerjaan pemrosesan, dan semua pengguna Anda mengizinkan untuk menjalankan langkah-langkah yang tersisa. Untuk menggunakan fitur ini, Anda memilih string kustom yang awalan nama pekerjaan Anda. Admin Anda menambahkan ARN yang diizinkan dengan awalan sementara ilmuwan data Anda menyertakan awalan ini dalam instantiasi pipeline. Karena kebijakan IAM untuk pengguna yang diizinkan berisi ARN pekerjaan dengan awalan yang ditentukan, pekerjaan berikutnya dari langkah pipeline Anda memiliki izin yang diperlukan untuk melanjutkan. Awalan Job tidak aktif secara default—Anda harus mengaktifkan opsi ini di kelas Anda untuk menggunakannya. `Pipeline`

Untuk pekerjaan dengan awalan dimatikan, nama pekerjaan diformat seperti yang ditunjukkan dan merupakan rangkaian bidang yang dijelaskan dalam tabel berikut:

`pipelines-<executionId>-<stepNamePrefix>-<entityToken>-<failureCount>`

Bidang	Definisi
jaringan pipa	String statis selalu ditambahkan. String ini mengidentifikasi layanan orkestrasi pipeline sebagai sumber pekerjaan.
ExecutionID	Buffer acak untuk instance pipeline yang sedang berjalan.
stepNamePrefix	Nama langkah yang ditentukan pengguna (diberikan dalam `name` argumen langkah pipeline), terbatas pada 20 karakter pertama.

Bidang	Definisi
EntityToken	Token acak untuk memastikan idempotensi entitas langkah.
FailureCount	Jumlah percobaan ulang saat ini berusaha menyelesaikan pekerjaan.

Dalam hal ini, tidak ada awalan khusus yang ditambahkan ke nama pekerjaan, dan kebijakan IAM yang sesuai harus cocok dengan string ini.

Untuk pengguna yang mengaktifkan awalan pekerjaan, nama pekerjaan yang mendasarinya mengambil formulir berikut, dengan awalan kustom ditentukan sebagai: `MyBaseJobName`

< MyBaseJobName > <executionId>- <entityToken>- <failureCount>

Awalan kustom menggantikan `pipelines` string statis untuk membantu Anda mempersempit pilihan pengguna yang dapat menjalankan SageMaker pekerjaan sebagai bagian dari pipeline.

Pembatasan panjang awalan

Nama pekerjaan memiliki batasan panjang internal khusus untuk masing-masing langkah pipa. Kendala ini juga membatasi panjang awalan yang diizinkan. Persyaratan panjang awalan adalah sebagai berikut:

Membuat Alur	Panjang awalan
`TrainingStep` , `ModelStep` , `TransformStep` , `ProcessingStep` , `ClarifyCheckStep` , `QualityCheckStep` , `RegisterModelStep`	38
`TuningStep` , `AutoML`	6

Terapkan awalan pekerjaan ke kebijakan IAM

Admin Anda membuat kebijakan IAM yang memungkinkan pengguna awalan tertentu untuk membuat pekerjaan. Contoh kebijakan berikut memungkinkan ilmuwan data untuk membuat pekerjaan pelatihan jika mereka menggunakan `MyBaseJobName` awalan.

```
{
    "Action": "sagemaker:CreateTrainingJob",
    "Effect": "Allow",
    "Resource": [
        "arn:aws:sagemaker:region:account-id:*/MyBaseJobName-*"
    ]
}
```

Terapkan awalan pekerjaan ke instantiasi pipeline

Anda menentukan awalan Anda dengan `*base_job_name` argumen kelas instance pekerjaan.

> **Note**
>
> Anda meneruskan awalan pekerjaan Anda dengan `*base_job_name` argumen ke instance pekerjaan sebelum membuat langkah pipeline. Contoh pekerjaan ini berisi informasi yang diperlukan agar pekerjaan dapat dijalankan sebagai langkah dalam pipeline. Argumen ini bervariasi tergantung pada instance pekerjaan yang digunakan. Daftar berikut menunjukkan argumen mana yang akan digunakan untuk setiap jenis langkah pipeline:
>
> - `base_job_name`untuk kelas `Estimator` (`TrainingStep`), `Processor` (`ProcessingStep`), dan `AutoML` (`AutoMLStep`)
> - `tuning_base_job_name`untuk `Tuner` kelas (`TuningStep`)
> - `transform_base_job_name`untuk `Transformer` kelas (`TransformStep`)
> - `base_job_name``CheckJobConfig`untuk kelas `QualityCheckStep` (Quality Check) dan `ClarifyCheckstep` (Clarify Check)
> - Untuk `Model` kelas, argumen yang digunakan tergantung pada apakah Anda menjalankan `create` atau `register` pada model Anda sebelum meneruskan hasilnya ke `ModelStep`
>   - Jika Anda memanggil`create`, awalan kustom berasal dari `name` argumen saat Anda membuat model Anda (yaitu,) `Model(name=)`
>   - Jika Anda menelepon`register`, awalan kustom berasal dari `model_package_name` argumen panggilan Anda ke `register` (yaitu,*`my_model`*`.register(model_package_name=)`)

Contoh berikut menunjukkan cara menentukan awalan untuk instans pekerjaan pelatihan baru.

```
# Create a job instance
```

```
xgb_train = Estimator(
    image_uri=image_uri,
    instance_type="ml.m5.xlarge",
    instance_count=1,
    output_path=model_path,
    role=role,
    subnets=["subnet-0ab12c34567de89f0"],
    base_job_name="MyBaseJobName"
    security_group_ids=["sg-1a2bbcc3bd4444e55"],
    tags = [ ... ]
    encrypt_inter_container_traffic=True,
)

# Attach your job instance to a pipeline step
step_train = TrainingStep(
    name="TestTrainingJob",
    estimator=xgb_train,
    inputs={
        "train": TrainingInput(...),
        "validation": TrainingInput(...)
    }
)
```

Awalan tugas berada dalam kondisi mati secara default. Untuk memilih fitur ini, gunakan `use_custom_job_prefix` opsi `PipelineDefinitionConfig` seperti yang ditunjukkan pada cuplikan berikut:

```
from sagemaker.workflow.pipeline_definition_config import PipelineDefinitionConfig

# Create a definition configuration and toggle on custom prefixing
definition_config = PipelineDefinitionConfig(use_custom_job_prefix=True);

# Create a pipeline with a custom prefix
 pipeline = Pipeline(
     name="MyJobPrefixedPipeline",
     parameters=[...]
     steps=[...]
     pipeline_definition_config=definition_config
)
```

Buat dan jalankan pipeline Anda. Contoh berikut membuat dan menjalankan pipeline, dan juga menunjukkan bagaimana Anda dapat mematikan awalan pekerjaan dan menjalankan kembali pipeline Anda.

```
pipeline.create(role_arn=sagemaker.get_execution_role())

# Optionally, call definition() to confirm your prefixed job names are in the built
 JSON
pipeline.definition()
pipeline.start()

# To run a pipeline without custom-prefixes, toggle off use_custom_job_prefix, update
 the pipeline
# via upsert() or update(), and start a new run
definition_config = PipelineDefinitionConfig(use_custom_job_prefix=False)
pipeline.pipeline_definition_config = definition_config
pipeline.update()
execution = pipeline.start()
```

Demikian pula, Anda dapat mengaktifkan fitur untuk pipeline yang ada dan memulai proses baru yang menggunakan awalan pekerjaan.

```
definition_config = PipelineDefinitionConfig(use_custom_job_prefix=True)
pipeline.pipeline_definition_config = definition_config
pipeline.update()
execution = pipeline.start()
```

Terakhir, Anda dapat melihat pekerjaan yang diawali khusus dengan memanggil eksekusi `list_steps` pipeline.

```
steps = execution.list_steps()

prefixed_training_job_name = steps['PipelineExecutionSteps'][0]['Metadata']
['TrainingJob']['Arn']
```

Kebijakan Kontrol Layanan dengan Pipa

Kebijakan kontrol layanan (SCP) adalah jenis kebijakan organisasi yang dapat Anda gunakan untuk mengelola izin dalam organisasi Anda. SCP menawarkan kontrol pusat atas izin maksimum yang tersedia untuk semua akun di organisasi Anda. Dengan menggunakan SageMaker Pipelines dalam

organisasi Anda, Anda dapat memastikan bahwa ilmuwan data mengelola eksekusi pipeline Anda tanpa harus berinteraksi dengan konsol. AWS

Jika Anda menggunakan VPC dengan SCP yang membatasi akses ke Amazon S3, Anda perlu mengambil langkah-langkah untuk mengizinkan pipeline mengakses sumber daya Amazon S3 lainnya.

Untuk mengizinkan SageMaker Pipelines mengakses Amazon S3 di luar VPC Anda dengan fungsi `JsonGet` tersebut, perbarui SCP organisasi Anda untuk memastikan bahwa peran yang menggunakan SageMaker Pipelines dapat mengakses Amazon S3. Untuk melakukan ini, buat pengecualian untuk peran yang digunakan oleh pelaksana SageMaker Pipelines melalui peran eksekusi pipeline menggunakan tag utama dan kunci kondisi.

Untuk mengizinkan SageMaker Pipelines mengakses Amazon S3 di luar VPC Anda

1. Buat tag unik untuk peran eksekusi pipeline Anda mengikuti langkah-langkah dalam Menandai pengguna dan peran IAM.
2. Berikan pengecualian di SCP Anda menggunakan kunci `Aws:PrincipalTag IAM` kondisi untuk tag yang Anda buat. Untuk informasi lebih lanjut, lihat Membuat, memperbarui, dan menghapus kebijakan kontrol layanan.

Dukungan Lintas Akun untuk Pipelines SageMaker

Anda dapat menggunakan dukungan lintas akun untuk Amazon SageMaker Model Building Pipelines untuk berbagi entitas pipeline di seluruh AWS akun dan mengakses pipeline bersama melalui panggilan API langsung.

Mengatur pembagian pipa lintas akun

SageMaker menggunakan AWSResource Access Manager (AWSRAM) untuk membantu Anda berbagi entitas pipeline dengan aman di seluruh akun.

Buat berbagi sumber daya

1. Pilih Buat berbagi sumber daya melalui AWS RAMkonsol.
2. Saat menentukan detail berbagi sumber daya, pilih jenis sumber daya SageMaker Pipelines dan pilih satu atau beberapa saluran pipa yang ingin Anda bagikan. Saat Anda berbagi pipeline dengan akun lain, semua eksekusinya juga dibagikan secara implisit.

3. Kaitkan izin dengan pembagian sumber daya Anda. Pilih kebijakan izin hanya-baca default atau kebijakan izin eksekusi pipeline yang diperluas. Untuk informasi selengkapnya, lihat Kebijakan izin untuk sumber daya SageMaker Pipelines.

   > **Note**
   >
   > Jika Anda memilih kebijakan eksekusi pipeline yang diperluas, perhatikan bahwa perintah start, stop, dan retry yang dipanggil oleh akun bersama menggunakan sumber daya di AWS akun yang berbagi pipeline.

4. Gunakan ID AWS akun untuk menentukan akun yang ingin Anda beri akses ke sumber daya bersama Anda.
5. Tinjau konfigurasi berbagi sumber daya Anda dan pilih Buat berbagi sumber daya. Perlu waktu beberapa menit agar pembagian sumber daya dan asosiasi utama selesai.

Untuk informasi selengkapnya, lihat Berbagi AWS sumber daya Anda di Panduan Pengguna AWS Resource Access Manager.

Dapatkan tanggapan atas undangan berbagi sumber daya Anda

Setelah pembagian sumber daya dan asosiasi prinsipal ditetapkan, AWS akun yang ditentukan menerima undangan untuk bergabung dengan pembagian sumber daya. AWSAkun harus menerima undangan untuk mendapatkan akses ke sumber daya bersama.

Untuk informasi selengkapnya tentang menerima undangan berbagi sumber dayaAWS RAM, lihat Menggunakan AWS sumber daya bersama di Panduan Pengguna AWS Resource Access Manager.

Kebijakan izin untuk sumber daya SageMaker Pipelines

Saat membuat pembagian sumber daya, pilih salah satu dari dua kebijakan izin yang didukung untuk dikaitkan dengan jenis sumber daya SageMaker pipeline. Kedua kebijakan tersebut memberikan akses ke saluran pipa yang dipilih dan semua pelaksanaannya.

Izin hanya-baca default

`AWSRAMDefaultPermissionSageMakerPipeline`Kebijakan ini memungkinkan tindakan hanya-baca berikut:

```
"sagemaker:DescribePipeline"
```

```
"sagemaker:DescribePipelineDefinitionForExecution"
"sagemaker:DescribePipelineExecution"
"sagemaker:ListPipelineExecutions"
"sagemaker:ListPipelineExecutionSteps"
"sagemaker:ListPipelineParametersForExecution"
"sagemaker:Search"
```

Izin eksekusi pipa yang diperluas

`AWSRAMPermissionSageMakerPipelineAllowExecution`Kebijakan ini mencakup semua izin hanya-baca dari kebijakan default dan juga memungkinkan akun bersama untuk memulai, menghentikan, dan mencoba lagi eksekusi pipeline.

> **Note**
>
> Perhatikan penggunaan AWS sumber daya saat menggunakan kebijakan izin eksekusi pipeline yang diperluas. Dengan kebijakan ini, akun bersama diizinkan untuk memulai, menghentikan, dan mencoba lagi eksekusi pipeline. Sumber daya apa pun yang digunakan untuk eksekusi pipa bersama dikonsumsi oleh akun pemilik.

Kebijakan izin eksekusi pipeline yang diperluas memungkinkan tindakan berikut:

```
"sagemaker:DescribePipeline"
"sagemaker:DescribePipelineDefinitionForExecution"
"sagemaker:DescribePipelineExecution"
"sagemaker:ListPipelineExecutions"
"sagemaker:ListPipelineExecutionSteps"
"sagemaker:ListPipelineParametersForExecution"
"sagemaker:StartPipelineExecution"
"sagemaker:StopPipelineExecution"
"sagemaker:RetryPipelineExecution"
"sagemaker:Search"
```

Akses entitas pipeline bersama melalui panggilan API langsung

Setelah berbagi pipeline lintas akun disiapkan, Anda dapat memanggil tindakan SageMaker API berikut menggunakan ARN pipeline:

> **Note**
>
> Anda hanya dapat memanggil perintah API jika perintah tersebut disertakan dalam izin yang terkait dengan pembagian sumber daya Anda. Jika Anda memilih `AWSRAMPermissionSageMakerPipelineAllowExecution` kebijakan, perintah mulai, berhenti, dan coba lagi menggunakan sumber daya di AWS akun yang membagikan pipeline.

- DescribePipeline
- DescribePipelineDefinitionForExecution
- DescribePipelineExecution
- ListPipelineExecutions
- ListPipelineExecutionSteps
- ListPipelineParametersForExecution
- StartPipelineExecution
- StopPipelineExecution
- RetryPipelineExecution

Parameter Alur

Anda dapat memasukkan variabel ke dalam definisi pipeline Anda menggunakan parameter. Anda dapat mereferensikan parameter yang Anda tentukan di seluruh definisi pipeline Anda. Parameter memiliki nilai default, yang dapat Anda timpa dengan menentukan nilai parameter saat memulai eksekusi pipeline. Nilai default harus berupa instance yang cocok dengan tipe parameter. Semua parameter yang digunakan dalam definisi langkah harus didefinisikan dalam definisi pipeline Anda. Pipeles Bangunan SageMaker Model Amazon mendukung tipe parameter berikut:

- `ParameterString`— Mewakili parameter string.
- `ParameterInteger`— Mewakili parameter integer.
- `ParameterFloat`— Mewakili parameter float.
- `ParameterBoolean`— Mewakili tipe Boolean Python.

Parameter mengambil format berikut:

```
<parameter> = <parameter_type>(
    name="<parameter_name>",
    default_value=<default_value>
)
```

Contoh berikut menunjukkan implementasi parameter sampel.

```
from sagemaker.workflow.parameters import (
    ParameterInteger,
    ParameterString,
    ParameterFloat,
    ParameterBoolean
)

processing_instance_count = ParameterInteger(
    name="ProcessingInstanceCount",
    default_value=1
)
```

Anda melewatkan parameter saat membuat alur Anda seperti yang ditunjukkan dalam contoh berikut ini.

```
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[
        processing_instance_count
    ],
    steps=[step_process]
)
```

Anda juga dapat meneruskan nilai parameter yang berbeda dari nilai default ke eksekusi pipeline, seperti yang ditunjukkan pada contoh berikut.

```
execution = pipeline.start(
    parameters=dict(
        ProcessingInstanceCount="2",
        ModelApprovalStatus="Approved"
    )
)
```

Anda dapat memanipulasi parameter dengan fungsi SageMaker Python SDK seperti. `sagemaker.workflow.functions.Join` Untuk informasi selengkapnya tentang parameter, lihat Parameter SageMaker Pipelines.

Untuk batasan Parameter SageMaker Pipelines yang diketahui, lihat Batasan - Parameterisasi di Amazon Python SageMaker SDK.

Membuat Alur

SageMaker Pipa terdiri dari langkah-langkah. Langkah-langkah ini menentukan tindakan yang diambil pipeline dan hubungan antara langkah-langkah menggunakan properti.

Topik

- Jenis Langkah
- Lab
- Paralelisme Langkah
- Ketergantungan Data Antar Langkah
- Ketergantungan Kustom Antar Langkah
- Gunakan Gambar Kustom dalam Satu Langkah

Jenis Langkah

Berikut ini menjelaskan persyaratan dari setiap jenis langkah dan memberikan contoh implementasi langkah. Ini bukan implementasi fungsional karena mereka tidak menyediakan sumber daya dan input yang dibutuhkan. Untuk tutorial yang mengimplementasikan langkah-langkah ini, lihatMembuat dan Mengelola SageMaker Pipelines.

> **Note**
>
> Anda juga dapat membuat langkah dari kode pembelajaran mesin lokal Anda dengan mengubahnya menjadi langkah SageMaker Pipelines dengan dekorator. `@step` Untuk informasi selengkapnya, lihat @step dekorator.

Amazon SageMaker Model Building Pipelines mendukung jenis langkah berikut:

- Pemrosesan
- Pelatihan

- Penyetelan
- AutoML
- `Model`
- `CreateModel`
- `RegisterModel`
- Transformasi
- Ketentuan
- `Callback`
- Lambda
- `ClarifyCheck`
- `QualityCheck`
- EMR
- Contoh Pelatihan
- Gagal

@step dekorator

Anda dapat membuat langkah dari kode pembelajaran mesin lokal menggunakan `@step` dekorator. Setelah Anda menguji kode Anda, Anda dapat mengonversi fungsi ke langkah SageMaker pipeline dengan membubuhi keterangan dengan dekorator. `@step` SageMaker Pipelines membuat dan menjalankan pipa ketika Anda melewati output dari fungsi `@step` -decorated sebagai langkah ke pipeline Anda. Anda juga dapat membuat pipa DAG multi-langkah yang mencakup satu atau lebih fungsi `@step` yang didekorasi serta langkah-langkah SageMaker pipa tradisional. Untuk detail selengkapnya tentang cara membuat langkah dengan `@step` dekorator, lihatL Kode ift-and-shift Python dengan dekorator @step.

Langkah Pemrosesan

Gunakan langkah pemrosesan untuk membuat pekerjaan pemrosesan untuk pemrosesan data. Untuk informasi selengkapnya tentang pekerjaan pemrosesan, lihat Memproses Data dan Mengevaluasi Model.

Langkah pemrosesan membutuhkan prosesor, skrip Python yang mendefinisikan kode pemrosesan, output untuk pemrosesan, dan argumen pekerjaan. Contoh berikut menunjukkan cara membuat `ProcessingStep` definisi.

```
from sagemaker.sklearn.processing import SKLearnProcessor

sklearn_processor = SKLearnProcessor(framework_version='1.0-1',
                                     role=<role>,
                                     instance_type='ml.m5.xlarge',
                                     instance_count=1)
```

```
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker.workflow.steps import ProcessingStep

inputs = [
    ProcessingInput(source=<input_data>, destination="/opt/ml/processing/input"),
]

outputs = [
    ProcessingOutput(output_name="train", source="/opt/ml/processing/train"),
    ProcessingOutput(output_name="validation", source="/opt/ml/processing/validation"),
    ProcessingOutput(output_name="test", source="/opt/ml/processing/test")
]

step_process = ProcessingStep(
    name="AbaloneProcess",
    step_args = sklearn_processor.run(inputs=inputs, outputs=outputs,
        code="abalone/preprocessing.py")
)
```

Lulus parameter runtime

Contoh berikut menunjukkan cara melewati parameter waktu aktif dari PySpark prosesor ke. `ProcessingStep`

```
from sagemaker.workflow.pipeline_context import PipelineSession
from sagemaker.spark.processing import PySparkProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker.workflow.steps import ProcessingStep

pipeline_session = PipelineSession()

pyspark_processor = PySparkProcessor(
    framework_version='2.4',
    role=<role>,
    instance_type='ml.m5.xlarge',
```

```
    instance_count=1,
    sagemaker_session=pipeline_session,
)

step_args = pyspark_processor.run(
    inputs=[ProcessingInput(source=<input_data>, destination="/opt/ml/processing/
input"),],
    outputs=[
        ProcessingOutput(output_name="train", source="/opt/ml/processing/train"),
        ProcessingOutput(output_name="validation", source="/opt/ml/processing/
validation"),
        ProcessingOutput(output_name="test", source="/opt/ml/processing/test")
    ],
    code="preprocess.py",
    arguments=None,
)


step_process = ProcessingStep(
    name="AbaloneProcess",
    step_args=step_args,
)
```

Untuk informasi selengkapnya tentang persyaratan langkah pemrosesan, lihat sagemaker.workflow.steps. ProcessingStepdokumentasi. Untuk contoh mendalam, lihat Menentukan Langkah Pemrosesan untuk Rekayasa Fitur di Buku catatan contoh Orkestrasi Pekerjaan untuk Melatih dan Mengevaluasi Model dengan Amazon SageMaker Pipelines.

Langkah Pelatihan

Anda menggunakan langkah pelatihan untuk membuat pekerjaan pelatihan untuk melatih model. Untuk informasi selengkapnya tentang pekerjaan pelatihan, lihat Melatih Model dengan Amazon SageMaker.

Langkah pelatihan membutuhkan estimator, serta input data pelatihan dan validasi. Contoh berikut menunjukkan cara membuat `TrainingStep` definisi. Untuk informasi selengkapnya tentang persyaratan langkah pelatihan, lihat sagemaker.workflow.steps. TrainingStepdokumentasi.

```
from sagemaker.workflow.pipeline_context import PipelineSession

from sagemaker.inputs import TrainingInput
from sagemaker.workflow.steps import TrainingStep
```

```
from sagemaker.xgboost.estimator import XGBoost

pipeline_session = PipelineSession()

xgb_estimator = XGBoost(..., sagemaker_session=pipeline_session)

step_args = xgb_estimator.fit(
    inputs={
        "train": TrainingInput(
            s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
                "train"
            ].S3Output.S3Uri,
            content_type="text/csv"
        ),
        "validation": TrainingInput(
            s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
                "validation"
            ].S3Output.S3Uri,
            content_type="text/csv"
        )
    }
)

step_train = TrainingStep(
    name="TrainAbaloneModel",
    step_args=step_args,
)
```

Langkah 1: Lab

Anda menggunakan langkah tuning untuk membuat pekerjaan tuning hyperparameter, juga dikenal sebagai optimasi hyperparameter (HPO). Pekerjaan tuning hyperparameter menjalankan beberapa pekerjaan pelatihan, masing-masing menghasilkan versi model. Untuk informasi lebih lanjut tentang penyetelan hyperparameter, lihat. Lakukan Penyetelan Model Otomatis dengan SageMaker

Pekerjaan penyetelan dikaitkan dengan SageMaker eksperimen untuk pipa, dengan pekerjaan pelatihan yang dibuat sebagai uji coba. Untuk informasi selengkapnya, lihat Integrasi Eksperimen.

Langkah penyetelan membutuhkan input HyperparameterTunerdan pelatihan. Anda dapat melatih kembali pekerjaan penyetelan sebelumnya dengan menentukan `warm_start_config` parameter. `HyperparameterTuner` Untuk informasi lebih lanjut tentang penyetelan hyperparameter dan start hangat, lihat. Jalankan Pekerjaan Tuning Hyperparameter Mulai yang Hangat

Anda menggunakan metode get_top_model_s3_uri dari sagemaker.workflow.steps. TuningStepkelas untuk mendapatkan artefak model dari salah satu versi model berkinerja terbaik. Untuk buku catatan yang menunjukkan cara menggunakan langkah penyetelan dalam SageMaker pipeline, lihat sagemaker-pipelines-tuning-step.ipynb.

> **Important**
>
> Langkah-langkah penyetelan diperkenalkan di Amazon SageMaker Python SDK v2.48.0 dan Amazon Studio Classic v3.8.0. SageMaker Anda harus memperbarui Studio Classic sebelum menggunakan langkah penyetelan atau DAG pipeline tidak ditampilkan. Untuk memperbarui Studio Classic, lihatMatikan dan Perbarui SageMaker Studio Classic.

Contoh berikut menunjukkan cara membuat `TuningStep` definisi.

```
from sagemaker.workflow.pipeline_context import PipelineSession

from sagemaker.tuner import HyperparameterTuner
from sagemaker.inputs import TrainingInput
from sagemaker.workflow.steps import TuningStep

tuner = HyperparameterTuner(..., sagemaker_session=PipelineSession())

step_tuning = TuningStep(
    name = "HPTuning",
    step_args = tuner.fit(inputs=TrainingInput(s3_data="s3://my-bucket/my-data"))
)
```

Dapatkan versi model terbaik

Contoh berikut menunjukkan cara mendapatkan versi model terbaik dari pekerjaan tuning menggunakan `get_top_model_s3_uri` metode ini. Paling-paling, 50 versi berkinerja teratas tersedia berdasarkan peringkat HyperParameterTuningJobObjective. `top_k`Argumennya adalah indeks ke dalam versi, di mana `top_k=0` versi berkinerja terbaik dan `top_k=49` merupakan versi berkinerja terburuk.

```
best_model = Model(
    image_uri=image_uri,
    model_data=step_tuning.get_top_model_s3_uri(
        top_k=0,
```

```
        s3_bucket=sagemaker_session.default_bucket()
    ),
    ...
)
```

Untuk informasi selengkapnya tentang persyaratan langkah penyetelan, lihat sagemaker.workflow.steps. TuningStepdokumentasi.

Langkah AutoMl

Gunakan AutoML API untuk membuat pekerjaan AutoML untuk melatih model secara otomatis. Untuk informasi selengkapnya tentang pekerjaan AutoML, lihat Mengotomatiskan pengembangan model dengan Amazon Autopilot. SageMaker

> **Note**
>
> Saat ini, langkah AutoML hanya mendukung mode pelatihan ensembling.

Contoh berikut menunjukkan cara membuat definisi menggunakan`AutoMLStep`.

```
from sagemaker.workflow.pipeline_context import PipelineSession
from sagemaker.workflow.automl_step import AutoMLStep

pipeline_session = PipelineSession()

auto_ml = AutoML(...,
    role="<role>",
    target_attribute_name="my_target_attribute_name",
    mode="ENSEMBLING",
    sagemaker_session=pipeline_session)

input_training = AutoMLInput(
    inputs="s3://my-bucket/my-training-data",
    target_attribute_name="my_target_attribute_name",
    channel_type="training",
)
input_validation = AutoMLInput(
    inputs="s3://my-bucket/my-validation-data",
    target_attribute_name="my_target_attribute_name",
    channel_type="validation",
)
```

```
step_args = auto_ml.fit(
    inputs=[input_training, input_validation]
)

step_automl = AutoMLStep(
    name="AutoMLStep",
    step_args=step_args,
)
```

Dapatkan versi model terbaik

Langkah AutoML secara otomatis melatih beberapa kandidat model. Anda bisa mendapatkan model dengan metrik objektif terbaik dari pekerjaan AutoML menggunakan `get_best_auto_ml_model` metode dan IAM `role` untuk mengakses artefak model sebagai berikut.

```
best_model = step_automl.get_best_auto_ml_model(role=<role>)
```

Untuk informasi selengkapnya, lihat langkah AutoML di Python SageMaker SDK.

Contoh

Gunakan a `ModelStep` untuk membuat atau mendaftarkan SageMaker model. Untuk informasi selengkapnya tentang `ModelStep` persyaratan, lihat sagemaker.workflow.model_step. ModelStepdokumentasi.

Membuat Alur

Anda dapat menggunakan a `ModelStep` untuk membuat SageMaker model. A `ModelStep` membutuhkan artefak model dan informasi tentang jenis SageMaker instance yang perlu Anda gunakan untuk membuat model. Untuk informasi selengkapnya tentang SageMaker model, lihat Melatih Model dengan Amazon SageMaker.

Contoh berikut menunjukkan cara membuat `ModelStep` definisi.

```
from sagemaker.workflow.pipeline_context import PipelineSession
from sagemaker.model import Model
from sagemaker.workflow.model_step import ModelStep

step_train = TrainingStep(...)
model = Model(
    image_uri=pytorch_estimator.training_image_uri(),
```

```
    model_data=step_train.properties.ModelArtifacts.S3ModelArtifacts,
    sagemaker_session=PipelineSession(),
    role=role,
)

step_model_create = ModelStep(
   name="MyModelCreationStep",
   step_args=model.create(instance_type="ml.m5.xlarge"),
)
```

Daftarkan Model

Anda dapat menggunakan a `ModelStep` untuk mendaftarkan a `sagemaker.model.Model` atau `sagemaker.pipeline.PipelineModel` dengan registri SageMaker model Amazon. A `PipelineModel` mewakili pipa inferensi, yang merupakan model yang terdiri dari urutan linier kontainer yang memproses permintaan inferensi. Untuk informasi lebih lanjut tentang cara mendaftar model, lihat[Daftarkan dan Terapkan Model dengan Registri Model](#).

Contoh berikut menunjukkan cara membuat daftar `ModelStep` yang `PipelineModel` mencatat.

```
import time

from sagemaker.workflow.pipeline_context import PipelineSession
from sagemaker.sklearn import SKLearnModel
from sagemaker.xgboost import XGBoostModel

pipeline_session = PipelineSession()

code_location = 's3://{0}/{1}/code'.format(bucket_name, prefix)

sklearn_model = SKLearnModel(

 model_data=processing_step.properties.ProcessingOutputConfig.Outputs['model'].S3Output.S3Uri,
   entry_point='inference.py',
   source_dir='sklearn_source_dir/',
   code_location=code_location,
   framework_version='1.0-1',
   role=role,
   sagemaker_session=pipeline_session,
   py_version='py3'
)

xgboost_model = XGBoostModel(
```

```
    model_data=training_step.properties.ModelArtifacts.S3ModelArtifacts,
    entry_point='inference.py',
    source_dir='xgboost_source_dir/',
    code_location=code_location,
    framework_version='0.90-2',
    py_version='py3',
    sagemaker_session=pipeline_session,
    role=role
)

from sagemaker.workflow.model_step import ModelStep
from sagemaker import PipelineModel

pipeline_model = PipelineModel(
    models=[sklearn_model, xgboost_model],
    role=role,sagemaker_session=pipeline_session,
)

register_model_step_args = pipeline_model.register(
     content_types=["application/json"],
    response_types=["application/json"],
    inference_instances=["ml.t2.medium", "ml.m5.xlarge"],
    transform_instances=["ml.m5.xlarge"],
    model_package_group_name='sipgroup',
)

step_model_registration = ModelStep(
    name="AbaloneRegisterModel",
    step_args=register_model_step_args,
)
```

CreateModel Langkah

> **Important**
>
> Kami merekomendasikan penggunaan Contoh untuk membuat model pada v2.90.0 dari Python SDK. SageMaker `CreateModelStep`akan terus bekerja di versi SDK SageMaker Python sebelumnya, tetapi tidak lagi didukung secara aktif.

Anda menggunakan `CreateModel` langkah untuk membuat SageMaker model. Untuk informasi selengkapnya tentang SageMaker model, lihat Melatih Model dengan Amazon SageMaker.

Langkah membuat model memerlukan artefak model dan informasi tentang jenis SageMaker instance yang perlu Anda gunakan untuk membuat model. Contoh berikut menunjukkan cara membuat definisi `CreateModel` langkah. Untuk informasi selengkapnya tentang persyaratan `CreateModel` langkah, lihat sagemaker.workflow.steps. CreateModelStepdokumentasi.

```
from sagemaker.workflow.steps import CreateModelStep

step_create_model = CreateModelStep(
    name="AbaloneCreateModel",
    model=best_model,
    inputs=inputs
)
```

RegisterModel Langkah

> **Important**
>
> Kami merekomendasikan penggunaan Contoh untuk mendaftarkan model pada v2.90.0 dari Python SDK. SageMaker `RegisterModel`akan terus bekerja di versi SDK SageMaker Python sebelumnya, tetapi tidak lagi didukung secara aktif.

Anda menggunakan `RegisterModel` langkah untuk mendaftarkan Sagemaker.model.Model atau sagemaker.pipeline. PipelineModeldengan registri SageMaker model Amazon. A `PipelineModel` mewakili pipa inferensi, yang merupakan model yang terdiri dari urutan linier kontainer yang memproses permintaan inferensi.

Untuk informasi lebih lanjut tentang cara mendaftar model, lihatDaftarkan dan Terapkan Model dengan Registri Model. Untuk informasi selengkapnya tentang persyaratan `RegisterModel` langkah, lihat sagemaker.workflow.step_collections. RegisterModeldokumentasi.

Contoh berikut menunjukkan cara membuat `RegisterModel` langkah yang `PipelineModel` mencatat.

```
import time
from sagemaker.sklearn import SKLearnModel
from sagemaker.xgboost import XGBoostModel

code_location = 's3://{0}/{1}/code'.format(bucket_name, prefix)
```

```
sklearn_model =
 SKLearnModel(model_data=processing_step.properties.ProcessingOutputConfig.Outputs['model'].S3O
 entry_point='inference.py',
 source_dir='sklearn_source_dir/',
 code_location=code_location,
 framework_version='1.0-1',
 role=role,
 sagemaker_session=sagemaker_session,
 py_version='py3')

xgboost_model =
 XGBoostModel(model_data=training_step.properties.ModelArtifacts.S3ModelArtifacts,
 entry_point='inference.py',
 source_dir='xgboost_source_dir/',
 code_location=code_location,
 framework_version='0.90-2',
 py_version='py3',
 sagemaker_session=sagemaker_session,
 role=role)

from sagemaker.workflow.step_collections import RegisterModel
from sagemaker import PipelineModel
pipeline_model =
 PipelineModel(models=[sklearn_model,xgboost_model],role=role,sagemaker_session=sagemaker_sessi

step_register = RegisterModel(
 name="AbaloneRegisterModel",
 model=pipeline_model,
 content_types=["application/json"],
 response_types=["application/json"],
 inference_instances=["ml.t2.medium", "ml.m5.xlarge"],
 transform_instances=["ml.m5.xlarge"],
 model_package_group_name='sipgroup',
)
```

Jika `model` tidak disediakan, langkah model register membutuhkan estimator seperti yang ditunjukkan pada contoh berikut.

```
from sagemaker.workflow.step_collections import RegisterModel

step_register = RegisterModel(
    name="AbaloneRegisterModel",
    estimator=xgb_train,
```

```
    model_data=step_train.properties.ModelArtifacts.S3ModelArtifacts,
    content_types=["text/csv"],
    response_types=["text/csv"],
    inference_instances=["ml.t2.medium", "ml.m5.xlarge"],
    transform_instances=["ml.m5.xlarge"],
    model_package_group_name=model_package_group_name,
    approval_status=model_approval_status,
    model_metrics=model_metrics
)
```

Pembuatan

Anda menggunakan langkah transformasi untuk transformasi batch untuk menjalankan inferensi pada seluruh kumpulan data. Untuk informasi lebih lanjut tentang transformasi batch, lihatJalankan Transformasi Batch dengan Pipa Inferensi.

Langkah transformasi membutuhkan transformator dan data untuk menjalankan transformasi batch. Contoh berikut menunjukkan cara membuat definisi `Transform` langkah. Untuk informasi selengkapnya tentang persyaratan `Transform` langkah, lihat sagemaker.workflow.steps. TransformStepdokumentasi.

```
from sagemaker.workflow.pipeline_context import PipelineSession

from sagemaker.transformer import Transformer
from sagemaker.inputs import TransformInput
from sagemaker.workflow.steps import TransformStep

transformer = Transformer(..., sagemaker_session=PipelineSession())

step_transform = TransformStep(
    name="AbaloneTransform",
    step_args=transformer.transform(data="s3://my-bucket/my-data"),
)
```

Langkah Kondisi

Anda menggunakan langkah kondisi untuk mengevaluasi kondisi properti langkah untuk menilai tindakan mana yang harus diambil selanjutnya dalam pipeline.

Langkah kondisi memerlukan daftar kondisi, daftar langkah yang harus dijalankan jika kondisi dievaluasi`true`, dan daftar langkah yang harus dijalankan jika kondisi dievaluasi. `false` Contoh berikut menunjukkan cara membuat `ConditionStep` definisi.

Batasan

- SageMaker Pipelines tidak mendukung penggunaan langkah-langkah kondisi bersarang. Anda tidak dapat melewati langkah kondisi sebagai input untuk langkah kondisi lain.
- Langkah kondisi tidak dapat menggunakan langkah yang identik di kedua cabang. Jika Anda membutuhkan fungsionalitas langkah yang sama di kedua cabang, duplikat langkahnya dan beri nama yang berbeda.

```
from sagemaker.workflow.conditions import ConditionLessThanOrEqualTo
from sagemaker.workflow.condition_step import ConditionStep
from sagemaker.workflow.functions import JsonGet

cond_lte = ConditionLessThanOrEqualTo(
    left=JsonGet(
        step_name=step_eval.name,
        property_file=evaluation_report,
        json_path="regression_metrics.mse.value"
    ),
    right=6.0
)

step_cond = ConditionStep(
    name="AbaloneMSECond",
    conditions=[cond_lte],
    if_steps=[step_register, step_create_model, step_transform],
    else_steps=[]
)
```

Untuk informasi selengkapnya tentang `ConditionStep` persyaratan, lihat sagemaker.workflow.condition_step. ConditionStepReferensi API. Untuk informasi selengkapnya tentang kondisi yang didukung, lihat Amazon SageMaker Model Building Pipelines - Ketentuan dalam dokumentasi SageMaker Python SDK.

Langkah

Anda menggunakan `Callback` langkah untuk menggabungkan proses dan AWS layanan tambahan ke dalam alur kerja Anda yang tidak langsung disediakan oleh Amazon SageMaker Model Building Pipelines. Ketika sebuah `Callback` langkah berjalan, prosedur berikut terjadi:

- SageMaker Pipelines mengirimkan pesan ke antrean Amazon Simple Queue Service (Amazon SQS). Pesan tersebut berisi token yang SageMaker dihasilkan pipeline dan daftar parameter input yang disediakan pelanggan. Setelah mengirim pesan, SageMaker Pipelines menunggu tanggapan dari pelanggan.
- Pelanggan mengambil pesan dari antrian Amazon SQS dan memulai proses kustom mereka.
- Ketika proses selesai, pelanggan memanggil salah satu API berikut dan mengirimkan token yang dihasilkan SageMaker pipeline:
  - SendPipelineExecutionStepSuccess, bersama dengan daftar parameter output
  - SendPipelineExecutionStepFailure, bersama dengan alasan kegagalan
- Panggilan API menyebabkan SageMaker Pipelines melanjutkan proses pipeline atau gagal dalam proses.

Untuk informasi selengkapnya tentang persyaratan `Callback` langkah, lihat sagemaker.workflow.callback_step. CallbackStepdokumentasi. Untuk solusi selengkapnya, lihat Memperluas SageMaker Pipelines untuk menyertakan langkah-langkah kustom menggunakan langkah panggilan balik.

> **Important**
>
> `Callback`langkah-langkah diperkenalkan di Amazon SageMaker Python SDK v2.45.0 dan Amazon Studio Classic v3.6.2. SageMaker Anda harus memperbarui Studio Classic sebelum menggunakan `Callback` langkah atau DAG pipeline tidak ditampilkan. Untuk memperbarui Studio Classic, lihatMatikan dan Perbarui SageMaker Studio Classic.

Sampel berikut menunjukkan implementasi dari prosedur sebelumnya.

```
from sagemaker.workflow.callback_step import CallbackStep

step_callback = CallbackStep(
    name="MyCallbackStep",
    sqs_queue_url="https://sqs.us-east-2.amazonaws.com/012345678901/MyCallbackQueue",
    inputs={...},
    outputs=[...]
)

callback_handler_code = '
    import boto3
```

```
    import json

    def handler(event, context):
        sagemaker_client=boto3.client("sagemaker")

        for record in event["Records"]:
            payload=json.loads(record["body"])
            token=payload["token"]

            # Custom processing

            # Call SageMaker to complete the step
            sagemaker_client.send_pipeline_execution_step_success(
                CallbackToken=token,
                OutputParameters={...}
            )
'
```

> **Note**
>
> Parameter keluaran untuk tidak `CallbackStep` boleh bersarang. Misalnya, jika Anda menggunakan kamus bersarang sebagai parameter keluaran Anda, maka kamus diperlakukan sebagai string tunggal (mis. `{"output1": "{\"nested_output1\": \"my-output\"}"}`). Jika Anda memberikan nilai bersarang, maka ketika Anda mencoba merujuk ke parameter keluaran tertentu, kesalahan klien yang tidak dapat dicoba ulang akan dilemparkan.

Menghentikan perilaku

Proses pipeline tidak berhenti saat `Callback` langkah sedang berjalan.

Saat Anda memanggil StopPipelineExecutionproses pipeline dengan `Callback` langkah yang sedang berjalan, SageMaker Pipelines mengirimkan pesan Amazon SQS tambahan ke antrean SQS yang ditentukan. Tubuh pesan SQS berisi bidang Status, yang diatur ke`Stopping`. Berikut ini menunjukkan contoh badan pesan SQS.

```
{
  "token": "26vcYbeWsZ",
  "pipelineExecutionArn": "arn:aws:sagemaker:us-east-2:012345678901:pipeline/callback-
pipeline/execution/7pinimwddh3a",
```

```
  "arguments": {
    "number": 5,
    "stringArg": "some-arg",
    "inputData": "s3://sagemaker-us-west-2-012345678901/abalone/abalone-dataset.csv"
  },
  "status": "Stopping"
}
```

Anda harus menambahkan logika ke konsumen pesan Amazon SQS Anda untuk mengambil tindakan apa pun yang diperlukan (misalnya, pembersihan sumber daya) setelah menerima pesan, diikuti dengan panggilan ke atau. `SendPipelineExecutionStepSuccess` `SendPipelineExecutionStepFailure`

Hanya ketika SageMaker Pipelines menerima salah satu panggilan ini barulah ia menghentikan proses pipeline.

Langkah Lambda

Anda menggunakan langkah Lambda untuk menjalankan fungsi. AWS Lambda Anda dapat menjalankan fungsi Lambda yang ada, atau SageMaker membuat dan menjalankan fungsi Lambda baru. Untuk buku catatan yang menunjukkan cara menggunakan langkah Lambda dalam SageMaker pipeline, lihat sagemaker-pipelines-lambda-step .ipynb.

> **Important**
>
> Langkah-langkah Lambda diperkenalkan di Amazon SageMaker Python SDK v2.51.0 dan Amazon Studio Classic v3.9.1. SageMaker Anda harus memperbarui Studio Classic sebelum menggunakan langkah Lambda atau DAG pipeline tidak ditampilkan. Untuk memperbarui Studio Classic, lihatMatikan dan Perbarui SageMaker Studio Classic.

SageMaker menyediakan kelas SageMaker.lambda_helper.Lambda untuk membuat, memperbarui, memanggil, dan menghapus fungsi Lambda. `Lambda`memiliki tanda tangan berikut.

```
Lambda(
    function_arn,       # Only required argument to invoke an existing Lambda function

    # The following arguments are required to create a Lambda function:
    function_name,
    execution_role_arn,
    zipped_code_dir,    # Specify either zipped_code_dir and s3_bucket, OR script
```

```
    s3_bucket,          # S3 bucket where zipped_code_dir is uploaded
    script,             # Path of Lambda function script
    handler,            # Lambda handler specified as "lambda_script.lambda_handler"
    timeout,            # Maximum time the Lambda function can run before the lambda
 step fails
    ...
)
```

Sagemaker.workflow.lambda_step. LambdaStepkelas memiliki `lambda_func` argumen tipe`Lambda`. Untuk menjalankan fungsi Lambda yang ada, satu-satunya persyaratan adalah menyediakan Amazon Resource Name (ARN) dari fungsi tersebut. `function_arn` Jika Anda tidak memberikan nilai untuk`function_arn`, Anda harus menentukan `handler` dan salah satu dari berikut:

- `zipped_code_dir`— Jalur fungsi Lambda yang di-zip

  `s3_bucket`- Bucket Amazon S3 tempat akan `zipped_code_dir` diunggah
- `script`— Jalur file skrip fungsi Lambda

Contoh berikut menunjukkan cara membuat definisi `Lambda` langkah yang memanggil fungsi Lambda yang ada.

```
from sagemaker.workflow.lambda_step import LambdaStep
from sagemaker.lambda_helper import Lambda

step_lambda = LambdaStep(
    name="ProcessingLambda",
    lambda_func=Lambda(
        function_arn="arn:aws:lambda:us-west-2:012345678910:function:split-dataset-
lambda"
    ),
    inputs={
        s3_bucket = s3_bucket,
        data_file = data_file
    },
    outputs=[
        "train_file", "test_file"
    ]
)
```

Contoh berikut menunjukkan cara membuat definisi `Lambda` langkah yang membuat dan memanggil fungsi Lambda menggunakan skrip fungsi Lambda.

```
from sagemaker.workflow.lambda_step import LambdaStep
from sagemaker.lambda_helper import Lambda

step_lambda = LambdaStep(
    name="ProcessingLambda",
    lambda_func=Lambda(
      function_name="split-dataset-lambda",
      execution_role_arn=execution_role_arn,
      script="lambda_script.py",
      handler="lambda_script.lambda_handler",
      ...
    ),
    inputs={
        s3_bucket = s3_bucket,
        data_file = data_file
    },
    outputs=[
        "train_file", "test_file"
    ]
)
```

Input dan output

Jika `Lambda` fungsi Anda memiliki input atau output, ini juga harus didefinisikan dalam langkah Anda`Lambda`.

> **Note**
>
> Parameter input dan output tidak boleh bersarang. Misalnya, jika Anda menggunakan kamus bersarang sebagai parameter keluaran Anda, maka kamus diperlakukan sebagai string tunggal (mis. `{"output1": "{\"nested_output1\":\"my-output\"}"}`). Jika Anda memberikan nilai bersarang dan mencoba merujuknya nanti, kesalahan klien yang tidak dapat dicoba ulang akan dilemparkan.

Saat mendefinisikan `Lambda` langkah, `inputs` harus kamus pasangan kunci-nilai. Setiap nilai `inputs` kamus harus berupa tipe primitif (string, integer, atau float). Objek bersarang tidak didukung. Jika dibiarkan tidak terdefinisi, `inputs` nilai defaultnya. `None`

`outputs`Nilai harus berupa daftar kunci. Kunci-kunci ini mengacu pada kamus yang didefinisikan dalam output `Lambda` fungsi. Seperti`inputs`, kunci ini harus tipe primitif, dan objek bersarang tidak didukung.

Timeout dan menghentikan perilaku

`Lambda`Kelas memiliki `timeout` argumen yang menentukan waktu maksimum bahwa fungsi Lambda dapat berjalan. Nilai default adalah 120 detik dengan nilai maksimum 10 menit. Jika fungsi Lambda berjalan saat batas waktu terpenuhi, langkah Lambda gagal; Namun, fungsi Lambda terus berjalan.

Proses pipeline tidak dapat dihentikan saat langkah Lambda berjalan karena fungsi Lambda yang dipanggil oleh langkah Lambda tidak dapat dihentikan. Jika Anda mencoba menghentikan proses saat fungsi Lambda berjalan, pipeline menunggu fungsi Lambda selesai atau sampai batas waktu tercapai, mana yang terjadi lebih dulu, lalu berhenti. Jika fungsi Lambda selesai, status proses pipeline adalah. `Stopped` Jika batas waktu tercapai, status proses pipeline adalah`Failed`.

ClarifyCheck Langkah

Anda dapat menggunakan `ClarifyCheck` langkah untuk melakukan pemeriksaan penyimpangan dasar terhadap garis dasar sebelumnya untuk analisis bias dan penjelasan model. Anda kemudian dapat membuat dan mendaftarkan baseline Anda dengan `model.register()` metode dan meneruskan output dari metode itu untuk Contoh digunakan. `step_args` Garis dasar untuk pemeriksaan drift ini dapat digunakan oleh Amazon SageMaker Model Monitor untuk titik akhir model Anda sehingga Anda tidak perlu melakukan saran dasar secara terpisah. `ClarifyCheck`Langkah ini juga dapat menarik garis dasar untuk pemeriksaan drift dari registri model. `ClarifyCheck`Langkah ini memanfaatkan container bawaan Amazon SageMaker Clarify yang menyediakan berbagai kemampuan pemantauan model, termasuk saran kendala dan validasi kendala terhadap baseline tertentu. Untuk informasi lebih lanjut, lihat Memulai dengan Container SageMaker Clarify.

Mengkonfigurasi langkahnya ClarifyCheck

Anda dapat mengonfigurasi `ClarifyCheck` langkah untuk melakukan hanya satu dari jenis pemeriksaan berikut setiap kali digunakan dalam pipeline.

- Pemeriksaan bias data
- Default-nya
- Pemeriksaan penjelasan model

Anda melakukan ini dengan mengatur `clarify_check_config` parameter dengan salah satu nilai jenis cek berikut:

- `DataBiasCheckConfig`
- `ModelBiasCheckConfig`
- `ModelExplainabilityCheckConfig`

`ClarifyCheck`Langkah ini meluncurkan pekerjaan pemrosesan yang menjalankan kontainer bawaan SageMaker Clarify dan memerlukan konfigurasi khusus untuk pemeriksaan dan pekerjaan pemrosesan. `ClarifyCheckConfig`dan `CheckJobConfig` merupakan fungsi pembantu untuk konfigurasi ini yang selaras dengan bagaimana pekerjaan pemrosesan SageMaker Clarify menghitung untuk memeriksa bias model, bias data, atau penjelasan model. Untuk informasi selengkapnya, lihat Jalankan Pekerjaan Pemrosesan SageMaker Klarifikasi untuk Analisis Bias dan Penjelasan.

Mengontrol perilaku langkah untuk pemeriksaan drift

`ClarifyCheck`Langkah ini membutuhkan dua flag boolean berikut untuk mengontrol perilakunya:

- `skip_check`: Parameter ini menunjukkan apakah pemeriksaan drift terhadap baseline sebelumnya dilewati atau tidak. Jika disetel ke`False`, baseline sebelumnya dari jenis pemeriksaan yang dikonfigurasi harus tersedia.
- `register_new_baseline`: Parameter ini menunjukkan apakah baseline yang baru dihitung dapat diakses melalui properti langkah. `BaselineUsedForDriftCheckConstraints` Jika disetel ke`False`, baseline sebelumnya dari jenis pemeriksaan yang dikonfigurasi juga harus tersedia. Informasi ini dapat diakses melalui `BaselineUsedForDriftCheckConstraints` properti.

Untuk informasi selengkapnya, lihat Perhitungan dasar, deteksi drift, dan siklus hidup dengan serta ClarifyCheck langkah-langkah QualityCheck di Amazon Model Building Pipelines SageMaker .

Bekerja dengan baseline

Anda dapat secara opsional menentukan `model_package_group_name` untuk menemukan baseline yang ada dan `ClarifyCheck` langkah menarik paket model terbaru yang disetujui dalam grup paket model. `DriftCheckBaselines` Atau, Anda dapat memberikan baseline sebelumnya melalui parameter. `supplied_baseline_constraints` Jika Anda menentukan

kedua `model_package_group_name` dan`supplied_baseline_constraints`, `ClarifyCheck` langkah menggunakan garis dasar yang ditentukan oleh parameter. `supplied_baseline_constraints`

Untuk informasi selengkapnya tentang penggunaan persyaratan `ClarifyCheck` langkah, lihat sagemaker.workflow.steps. ClarifyCheckStepdi Amazon SageMaker SageMaker SDK untuk Python. Untuk notebook Amazon SageMaker Studio Classic yang menunjukkan cara menggunakan `ClarifyCheck` step di SageMaker Pipelines, lihat sagemaker-pipeline-model-monitor-clarify-steps.ipynb.

Example Buat **`ClarifyCheck`** langkah untuk pemeriksaan bias data

```
from sagemaker.workflow.check_job_config import CheckJobConfig
from sagemaker.workflow.clarify_check_step import DataBiasCheckConfig, ClarifyCheckStep
from sagemaker.workflow.execution_variables import ExecutionVariables

check_job_config = CheckJobConfig(
    role=role,
    instance_count=1,
    instance_type="ml.c5.xlarge",
    volume_size_in_gb=120,
    sagemaker_session=sagemaker_session,
)

data_bias_data_config = DataConfig(

 s3_data_input_path=step_process.properties.ProcessingOutputConfig.Outputs["train"].S3Output.S3
    s3_output_path=Join(on='/', values=['s3:/', your_bucket, base_job_prefix,
 ExecutionVariables.PIPELINE_EXECUTION_ID, 'databiascheckstep']),
    label=0,
    dataset_type="text/csv",
    s3_analysis_config_output_path=data_bias_analysis_cfg_output_path,
)

data_bias_config = BiasConfig(
    label_values_or_threshold=[15.0], facet_name=[8], facet_values_or_threshold=[[0.5]]

)

data_bias_check_config = DataBiasCheckConfig(
    data_config=data_bias_data_config,
    data_bias_config=data_bias_config,
)h
```

```
data_bias_check_step = ClarifyCheckStep(
    name="DataBiasCheckStep",
    clarify_check_config=data_bias_check_config,
    check_job_config=check_job_config,
    skip_check=False,
    register_new_baseline=False
   supplied_baseline_constraints="s3://sagemaker-us-west-2-111122223333/baseline/
analysis.json",
    model_package_group_name="MyModelPackageGroup"
)
```

QualityCheck Langkah

Anda dapat menggunakan `QualityCheck` langkah ini untuk melakukan saran dasar dan pemeriksaan drift terhadap baseline sebelumnya untuk kualitas data atau kualitas model dalam pipeline. Anda kemudian dapat membuat dan mendaftarkan baseline Anda dengan `model.register()` metode dan meneruskan output dari metode itu untuk Contoh digunakan. `step_args` Model Monitor dapat menggunakan garis dasar ini untuk pemeriksaan drift untuk titik akhir model Anda sehingga Anda tidak perlu melakukan saran dasar secara terpisah. `QualityCheck`Langkah ini juga dapat menarik garis dasar untuk pemeriksaan drift dari registri model. `QualityCheck`Langkah ini memanfaatkan wadah prebuilt Amazon SageMaker Model Monitor, yang memiliki berbagai kemampuan pemantauan model termasuk saran kendala, pembuatan statistik, dan validasi kendala terhadap baseline. Untuk informasi selengkapnya, lihat Amazon SageMaker Model Monitor wadah bawaan.

Mengkonfigurasi langkahnya QualityCheck

Anda dapat mengonfigurasi `QualityCheck` langkah untuk melakukan hanya satu dari jenis pemeriksaan berikut setiap kali digunakan dalam pipeline.

- Pemeriksaan kualitas data
- Pemeriksaan kualitas model

Anda melakukan ini dengan mengatur `quality_check_config` parameter dengan salah satu nilai jenis cek berikut:

- `DataQualityCheckConfig`
- `ModelQualityCheckConfig`

`QualityCheck`Langkah ini meluncurkan pekerjaan pemrosesan yang menjalankan wadah bawaan Model Monitor dan memerlukan konfigurasi khusus untuk pemeriksaan dan pekerjaan pemrosesan. Fungsi `QualityCheckConfig` dan `CheckJobConfig` pembantu untuk konfigurasi ini yang selaras dengan cara Model Monitor membuat garis dasar untuk kualitas model atau pemantauan kualitas data. Untuk informasi selengkapnya tentang saran dasar Model Monitor, lihat Membuat dasar dan. Buat Model Quality Baseline

Mengontrol perilaku langkah untuk pemeriksaan drift

`QualityCheck`Langkah ini membutuhkan dua flag Boolean berikut untuk mengontrol perilakunya:

- `skip_check`: Parameter ini menunjukkan apakah pemeriksaan drift terhadap baseline sebelumnya dilewati atau tidak. Jika disetel ke`False`, baseline sebelumnya dari jenis pemeriksaan yang dikonfigurasi harus tersedia.
- `register_new_baseline`: Parameter ini menunjukkan apakah baseline yang baru dihitung dapat diakses melalui properti `BaselineUsedForDriftCheckConstraints` langkah dan. `BaselineUsedForDriftCheckStatistics` Jika disetel ke`False`, baseline sebelumnya dari jenis pemeriksaan yang dikonfigurasi juga harus tersedia. Bidang ini dapat diakses melalui `BaselineUsedForDriftCheckConstraints` dan `BaselineUsedForDriftCheckStatistics` properti.

Untuk informasi selengkapnya, lihat Perhitungan dasar, deteksi drift, dan siklus hidup dengan serta ClarifyCheck langkah-langkah QualityCheck di Amazon Model Building Pipelines SageMaker .

Bekerja dengan baseline

Anda dapat menentukan garis dasar sebelumnya secara langsung melalui `supplied_baseline_constraints` parameter `supplied_baseline_statistics` dan, atau Anda cukup menentukan `model_package_group_name` dan `QualityCheck` langkah menarik paket model terbaru yang disetujui dalam grup paket model. `DriftCheckBaselines` Saat Anda menentukan`model_package_group_name`, `QualityCheck` langkah`supplied_baseline_constraints`, dan`supplied_baseline_statistics`, menggunakan garis dasar yang ditentukan oleh `supplied_baseline_constraints` dan `supplied_baseline_statistics` pada jenis pemeriksaan dari `QualityCheck` langkah yang Anda jalankan.

Untuk informasi selengkapnya tentang penggunaan persyaratan `QualityCheck` langkah, lihat sagemaker.workflow.steps. QualityCheckStepdi Amazon SageMaker SageMaker SDK untuk

Python. Untuk notebook Amazon SageMaker Studio Classic yang menunjukkan cara menggunakan `QualityCheck` step di SageMaker Pipelines, lihat sagemaker-pipeline-model-monitor-clarify-steps.ipynb.

Example Buat **`QualityCheck`** langkah untuk pemeriksaan kualitas data

```
from sagemaker.workflow.check_job_config import CheckJobConfig
from sagemaker.workflow.quality_check_step import DataQualityCheckConfig,
 QualityCheckStep
from sagemaker.workflow.execution_variables import ExecutionVariables

check_job_config = CheckJobConfig(
    role=role,
    instance_count=1,
    instance_type="ml.c5.xlarge",
    volume_size_in_gb=120,
    sagemaker_session=sagemaker_session,
)

data_quality_check_config = DataQualityCheckConfig(

 baseline_dataset=step_process.properties.ProcessingOutputConfig.Outputs["train"].S3Output.S3Ur
    dataset_format=DatasetFormat.csv(header=False, output_columns_position="START"),
    output_s3_uri=Join(on='/', values=['s3:/', your_bucket, base_job_prefix,
 ExecutionVariables.PIPELINE_EXECUTION_ID, 'dataqualitycheckstep'])
)

data_quality_check_step = QualityCheckStep(
    name="DataQualityCheckStep",
    skip_check=False,
    register_new_baseline=False,
    quality_check_config=data_quality_check_config,
    check_job_config=check_job_config,
    supplied_baseline_statistics="s3://sagemaker-us-west-2-555555555555/baseline/
statistics.json",
    supplied_baseline_constraints="s3://sagemaker-us-west-2-555555555555/baseline/
constraints.json",
    model_package_group_name="MyModelPackageGroup"
)
```

Langkah EMR

Anda dapat menggunakan langkah EMR Amazon SageMaker Model Building Pipelines untuk memproses langkah-langkah EMR Amazon pada klaster EMR Amazon yang sedang berjalan atau meminta pipeline membuat dan mengelola klaster EMR Amazon untuk Anda. Untuk informasi selengkapnya tentang Amazon EMR, lihat Memulai dengan Amazon EMR.

Langkah EMR mengharuskan `EMRStepConfig` menyertakan lokasi file JAR yang akan digunakan oleh cluster EMR Amazon dan argumen apa pun yang akan diteruskan. Anda juga memberikan ID klaster EMR Amazon jika Anda ingin menjalankan langkah pada klaster EMR yang sedang berjalan, atau konfigurasi cluster jika Anda ingin langkah EMR berjalan di klaster yang dibuat, dikelola, dan dihentikan untuk Anda. Bagian berikut mencakup contoh dan tautan ke buku catatan sampel yang menunjukkan kedua metode.

Note

- Langkah-langkah EMR mengharuskan peran yang diteruskan ke pipeline Anda memiliki izin tambahan. Anda harus melampirkan kebijakan AWS terkelola: `AmazonSageMakerPipelinesIntegrations` ke peran pipeline Anda, atau memastikan bahwa peran tersebut menyertakan izin dalam kebijakan tersebut.
- Langkah EMR tidak didukung pada EMR tanpa server, atau di Amazon EMR di EKS.
- Jika Anda memproses langkah EMR pada cluster yang sedang berjalan, Anda hanya dapat menggunakan cluster yang berada di salah satu status berikut:`STARTING`,, `BOOTSTRAPPING``RUNNING`, atau. `WAITING`
- Jika Anda memproses langkah-langkah EMR pada cluster yang sedang berjalan, Anda dapat memiliki paling banyak 256 langkah EMR dalam keadaan di cluster EMR. `PENDING` Langkah-langkah EMR yang diajukan di luar batas ini mengakibatkan kegagalan eksekusi pipeline. Anda dapat mempertimbangkan untuk menggunakanCoba lagi Kebijakan untuk Langkah-langkah Pipa.
- Anda dapat menentukan baik ID klaster atau konfigurasi klaster, bukan keduanya.
- Langkah EMR bergantung pada Amazon EventBridge untuk memantau perubahan dalam langkah EMR atau status cluster. Jika Anda memproses pekerjaan EMR Amazon di klaster yang sedang berjalan, langkah EMR menggunakan aturan `SageMakerPipelineExecutionEMRStepStatusUpdateRule` untuk memantau status langkah EMR. Jika Anda memproses pekerjaan Anda di klaster yang dibuat langkah EMR untuk Anda, langkah tersebut menggunakan

`SageMakerPipelineExecutionEMRClusterStatusRule` aturan untuk memantau perubahan status klaster. Jika Anda melihat salah satu dari EventBridge aturan ini di AWS akun Anda, jangan menghapusnya atau langkah EMR Anda mungkin tidak selesai.

Luncurkan pekerjaan baru di klaster EMR Amazon yang sedang berjalan

Jika Anda ingin meluncurkan pekerjaan baru di klaster EMR Amazon yang sedang berjalan, Anda meneruskan ID cluster sebagai string ke `cluster_id` argumen. `EMRStep` Contoh berikut menunjukkan prosedur ini.

```
from sagemaker.workflow.emr_step import EMRStep, EMRStepConfig

emr_config = EMRStepConfig(
    jar="jar-location", # required, path to jar file used
    args=["--verbose", "--force"], # optional list of arguments to pass to the jar
    main_class="com.my.Main1", # optional main class, this can be omitted if jar above
 has a manifest
    properties=[ # optional list of Java properties that are set when the step runs
    {
        "key": "mapred.tasktracker.map.tasks.maximum",
        "value": "2"
    },
    {
        "key": "mapreduce.map.sort.spill.percent",
        "value": "0.90"
   },
   {
       "key": "mapreduce.tasktracker.reduce.tasks.maximum",
       "value": "5"
    }
  ]
)

step_emr = EMRStep (
    name="EMRSampleStep", # required
    cluster_id="j-1ABCDEFG2HIJK", # include cluster_id to use a running cluster
    step_config=emr_config, # required
    display_name="My EMR Step",
    description="Pipeline step to execute EMR job"
)
```

Untuk contoh buku catatan yang memandu Anda melalui contoh lengkap, lihat SageMaker Pipelines EMR Step With Running EMR Cluster.

Luncurkan pekerjaan baru di cluster EMR Amazon baru

Jika Anda ingin meluncurkan pekerjaan baru di cluster baru yang `EMRStep` membuat untuk Anda, berikan konfigurasi cluster Anda sebagai kamus dengan struktur yang sama sebagai RunJobFlowpermintaan. Namun, jangan sertakan bidang berikut dalam konfigurasi klaster Anda:

- `[Name]`
- `[Steps]`
- `[AutoTerminationPolicy]`
- `[Instances][KeepJobFlowAliveWhenNoSteps]`
- `[Instances][TerminationProtected]`

Semua `RunJobFlow` argumen lain tersedia untuk digunakan dalam konfigurasi cluster Anda. Untuk detail tentang sintaks permintaan, lihat RunJobFlow.

Contoh berikut meneruskan konfigurasi cluster ke definisi langkah EMR, yang meminta langkah untuk meluncurkan pekerjaan baru pada cluster EMR baru. Konfigurasi cluster EMR dalam contoh ini mencakup spesifikasi untuk node cluster EMR primer dan inti. Untuk informasi selengkapnya tentang jenis node EMR Amazon, lihat Memahami tipe node: node primer, inti, dan tugas.

```
from sagemaker.workflow.emr_step import EMRStep, EMRStepConfig

emr_step_config = EMRStepConfig(
    jar="jar-location", # required, path to jar file used
    args=["--verbose", "--force"], # optional list of arguments to pass to the jar
    main_class="com.my.Main1", # optional main class, this can be omitted if jar above
 has a manifest
    properties=[ # optional list of Java properties that are set when the step runs
    {
        "key": "mapred.tasktracker.map.tasks.maximum",
        "value": "2"
    },
    {
        "key": "mapreduce.map.sort.spill.percent",
        "value": "0.90"
   },
   {
```

```
        "key": "mapreduce.tasktracker.reduce.tasks.maximum",
        "value": "5"
    }
  ]
)

# include your cluster configuration as a dictionary
emr_cluster_config = {
    "Applications": [
        {
            "Name": "Spark",
        }
    ],
    "Instances":{
        "InstanceGroups":[
            {
                "InstanceRole": "MASTER",
                "InstanceCount": 1,
                "InstanceType": "m5.2xlarge"
            },
            {
                "InstanceRole": "CORE",
                "InstanceCount": 2,
                "InstanceType": "m5.2xlarge"
            }
        ]
    },
    "BootstrapActions":[],
    "ReleaseLabel": "emr-6.6.0",
    "JobFlowRole": "job-flow-role",
    "ServiceRole": "service-role"
}

emr_step = EMRStep(
    name="emr-step",
    cluster_id=None,
    display_name="emr_step",
    description="MyEMRStepDescription",
    step_config=emr_step_config,
    cluster_config=emr_cluster_config
)
```

Untuk contoh buku catatan yang memandu Anda melalui contoh lengkap, lihat SageMaker Pipelines EMR Step With Cluster Lifecycle Management.

Langkah Job Notebook

Gunakan a `NotebookJobStep` untuk menjalankan Job SageMaker Notebook Anda secara non-interaktif sebagai langkah pipeline. Untuk informasi selengkapnya tentang Pekerjaan SageMaker Notebook, lihatSageMaker Lowongan Notebook.

A `NotebookJobStep` membutuhkan minimal notebook input, URI gambar dan nama kernel. Untuk informasi selengkapnya tentang persyaratan langkah Job Notebook dan parameter lain yang dapat Anda atur untuk menyesuaikan langkah Anda, lihat sagemaker.workflow.steps. NotebookJobStep.

Contoh berikut menggunakan argumen minimum untuk mendefinisikan a`NotebookJobStep`.

```
from sagemaker.workflow.notebook_job_step import NotebookJobStep


notebook_job_step = NotebookJobStep(
    input_notebook=input_notebook,
    image_uri=image_uri,
    kernel_name=kernel_name
)
```

Langkah `NotebookJobStep` pipeline Anda diperlakukan sebagai pekerjaan SageMaker notebook, sehingga Anda dapat melacak status eksekusi di dasbor pekerjaan notebook UI Studio Classic jika menyertakan tag tertentu dengan `tags` argumen. Untuk detail selengkapnya tentang tag yang akan disertakan, lihatMelihat pekerjaan notebook Anda di dasbor Studio UI.

Selain itu, jika Anda menjadwalkan pekerjaan notebook menggunakan SageMaker Python SDK, Anda hanya dapat menentukan gambar tertentu untuk menjalankan pekerjaan notebook Anda. Untuk informasi selengkapnya, lihat Kendala gambar untuk pekerjaan notebook Python SageMaker SDK.

Langkah Gagal

Anda menggunakan a `FailStep` untuk menghentikan eksekusi Amazon SageMaker Model Building Pipelines ketika kondisi atau status yang diinginkan tidak tercapai dan untuk menandai eksekusi pipeline tersebut sebagai gagal. Ini `FailStep` juga memungkinkan Anda memasukkan pesan kesalahan khusus, yang menunjukkan penyebab kegagalan eksekusi pipeline.

> **Note**
>
> Ketika sebuah `FailStep` dan langkah-langkah pipeline lainnya dijalankan secara bersamaan, pipeline tidak berakhir sampai semua langkah bersamaan selesai.

Batasan untuk menggunakan **`FailStep`**

- Anda tidak dapat menambahkan `FailStep` ke `DependsOn` daftar langkah lain. Untuk informasi selengkapnya, lihat Ketergantungan Kustom Antar Langkah.
- Langkah-langkah lain tidak dapat mereferensikan`FailStep`. Itu selalu merupakan langkah terakhir dalam eksekusi pipeline.
- Anda tidak dapat mencoba lagi eksekusi pipeline yang diakhiri dengan a`FailStep`.

Anda dapat membuat `FailStep` `ErrorMessage` dalam bentuk string teks statis. Atau, Anda juga dapat menggunakan Parameter Pipeline operasi Gabung, atau properti langkah lainnya untuk membuat pesan kesalahan yang lebih informatif.

Example

Contoh cuplikan kode berikut menggunakan dengan `ErrorMessage` dikonfigurasi `FailStep` dengan Parameter Pipeline dan operasi. `Join`

```
from sagemaker.workflow.fail_step import FailStep
from sagemaker.workflow.functions import Join
from sagemaker.workflow.parameters import ParameterInteger

mse_threshold_param = ParameterInteger(name="MseThreshold", default_value=5)
step_fail = FailStep(
    name="AbaloneMSEFail",
    error_message=Join(
        on=" ", values=["Execution failed due to MSE >", mse_threshold_param]
    ),
)
```

Lab

`properties`Atribut digunakan untuk menambahkan dependensi data antara langkah-langkah dalam pipeline. Dependensi data ini kemudian digunakan oleh SageMaker Pipelines untuk membangun

DAG dari definisi pipeline. Properti ini dapat direferensikan sebagai nilai placeholder dan diselesaikan saat runtime.

`properties`Atribut langkah SageMaker Pipelines cocok dengan objek yang dikembalikan oleh `Describe` panggilan untuk jenis SageMaker pekerjaan yang sesuai. Untuk setiap jenis pekerjaan, `Describe` panggilan mengembalikan objek respons berikut:

- `ProcessingStep` – DescribeProcessingJob
- `TrainingStep` – DescribeTrainingJob
- `TransformStep` – DescribeTransformJob

Untuk memeriksa properti mana yang dapat direferensikan untuk setiap tipe langkah selama pembuatan dependensi data, lihat Ketergantungan Data - Referensi Properti di Amazon Python SageMaker SDK.

Paralelisme Langkah

Ketika sebuah langkah tidak bergantung pada langkah lain, itu dijalankan segera setelah eksekusi pipeline. Namun, mengeksekusi terlalu banyak langkah pipeline secara paralel dapat dengan cepat menghabiskan sumber daya yang tersedia. Kontrol jumlah langkah bersamaan untuk eksekusi pipeline dengan`ParallelismConfiguration`.

Contoh berikut digunakan `ParallelismConfiguration` untuk mengatur batas langkah bersamaan ke lima.

```
pipeline.create(
    parallelism_config=ParallelismConfiguration(5),
)
```

Ketergantungan Data Antar Langkah

Anda menentukan struktur DAG Anda dengan menentukan hubungan data antar langkah. Untuk membuat dependensi data antar langkah, berikan properti dari satu langkah sebagai input ke langkah lain dalam pipeline. Langkah menerima input tidak dimulai sampai setelah langkah menyediakan input selesai berjalan.

Ketergantungan data menggunakan JsonPath notasi dalam format berikut. Format ini melintasi file properti JSON, yang berarti Anda dapat menambahkan sebanyak mungkin <property>instance yang

diperlukan untuk mencapai properti bersarang yang diinginkan dalam file. Untuk informasi lebih lanjut tentang JsonPath notasi, lihat JsonPath repo.

```
<step_name>.properties.<property>.<property>
```

Berikut ini menunjukkan cara menentukan bucket Amazon S3 menggunakan `ProcessingOutputConfig` properti langkah pemrosesan.

```
step_process.properties.ProcessingOutputConfig.Outputs["train_data"].S3Output.S3Uri
```

Untuk membuat ketergantungan data, teruskan bucket ke langkah pelatihan sebagai berikut.

```
from sagemaker.workflow.pipeline_context import PipelineSession

sklearn_train = SKLearn(..., sagemaker_session=PipelineSession())

step_train = TrainingStep(
    name="CensusTrain",
    step_args=sklearn_train.fit(inputs=TrainingInput(
        s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
            "train_data"].S3Output.S3Uri
    ))
)
```

Untuk memeriksa properti mana yang dapat direferensikan untuk setiap tipe langkah selama pembuatan dependensi data, lihat Ketergantungan Data - Referensi Properti di Amazon Python SageMaker SDK.

Ketergantungan Kustom Antar Langkah

Saat Anda menentukan ketergantungan data, SageMaker Pipelines menyediakan koneksi data di antara langkah-langkah tersebut. Atau, satu langkah dapat mengakses data dari langkah sebelumnya tanpa langsung menggunakan SageMaker Pipelines. Dalam hal ini, Anda dapat membuat dependensi khusus yang memberi tahu SageMaker Pipelines untuk tidak memulai langkah sampai setelah langkah lain selesai berjalan. Anda membuat dependensi kustom dengan menentukan atribut step. `DependsOn`

Sebagai contoh, berikut ini mendefinisikan langkah `C` yang dimulai hanya setelah kedua langkah `A` dan langkah `B` selesai berjalan.

```
{
  'Steps': [
    {'Name':'A', ...},
    {'Name':'B', ...},
    {'Name':'C', 'DependsOn': ['A', 'B']}
  ]
}
```

SageMaker Pipelines melempar pengecualian validasi jika ketergantungan akan membuat ketergantungan siklik.

Contoh berikut membuat langkah pelatihan yang dimulai setelah langkah pemrosesan selesai berjalan.

```
processing_step = ProcessingStep(...)
training_step = TrainingStep(...)

training_step.add_depends_on([processing_step])
```

Contoh berikut membuat langkah pelatihan yang tidak dimulai sampai dua langkah pemrosesan yang berbeda selesai berjalan.

```
processing_step_1 = ProcessingStep(...)
processing_step_2 = ProcessingStep(...)

training_step = TrainingStep(...)

training_step.add_depends_on([processing_step_1, processing_step_2])
```

Berikut ini menyediakan cara alternatif untuk membuat ketergantungan kustom.

```
training_step.add_depends_on([processing_step_1])
training_step.add_depends_on([processing_step_2])
```

Contoh berikut membuat langkah pelatihan yang menerima masukan dari satu langkah pemrosesan dan menunggu langkah pemrosesan yang berbeda untuk selesai berjalan.

```
processing_step_1 = ProcessingStep(...)
processing_step_2 = ProcessingStep(...)
```

```
training_step = TrainingStep(
    ...,
    inputs=TrainingInput(
        s3_data=processing_step_1.properties.ProcessingOutputConfig.Outputs[
            "train_data"
        ].S3Output.S3Uri
    )

training_step.add_depends_on([processing_step_2])
```

Contoh berikut menunjukkan cara mengambil daftar string dependensi kustom langkah.

```
custom_dependencies = training_step.depends_on
```

Gunakan Gambar Kustom dalam Satu Langkah

Anda dapat menggunakan salah satu gambar SageMaker Deep Learning Container yang tersedia saat membuat langkah dalam pipeline.

Anda juga dapat menggunakan kontainer Anda sendiri dengan langkah-langkah pipa. Karena Anda tidak dapat membuat gambar dari dalam Amazon SageMaker Studio Classic, Anda harus membuat gambar menggunakan metode lain sebelum menggunakannya dengan Amazon SageMaker Model Building Pipelines.

Untuk menggunakan penampung Anda sendiri saat membuat langkah-langkah untuk pipeline Anda, sertakan URI gambar dalam definisi estimator. Untuk informasi selengkapnya tentang menggunakan kontainer Anda sendiri SageMaker, lihat Menggunakan Kontainer Docker dengan SageMaker.

L Kode ift-and-shift Python dengan dekorator @step

`@step`Dekorator adalah fitur yang mengubah kode machine learning (ML) lokal Anda menjadi satu atau beberapa langkah pipeline. Anda dapat menulis fungsi ML Anda seperti yang Anda lakukan untuk setiap proyek ML. Setelah diuji secara lokal atau sebagai pekerjaan pelatihan menggunakan `@remote` dekorator, Anda dapat mengubah fungsi menjadi langkah SageMaker pipa dengan menambahkan dekorator. `@step` Anda kemudian dapat meneruskan output dari panggilan fungsi `@step` -decorated sebagai langkah ke SageMaker Pipelines untuk membuat dan menjalankan pipeline. Anda dapat menghubungkan serangkaian fungsi dengan `@step` dekorator untuk membuat pipa grafik asiklik terarah (DAG) multi-langkah juga.

Pengaturan untuk menggunakan `@step` dekorator sama dengan pengaturan untuk menggunakan `@remote` dekorator. Anda dapat merujuk ke dokumentasi fungsi jarak jauh untuk detail tentang cara

mengatur lingkungan dan menggunakan file konfigurasi untuk mengatur default. Untuk informasi selengkapnya tentang `@step` dekorator, lihat sagemaker.workflow.function_step.step.

Untuk melihat contoh buku catatan yang menunjukkan penggunaan `@step` dekorator, lihat contoh notebook @step decorator.

Bagian berikut menjelaskan bagaimana Anda dapat membuat anotasi kode HTML lokal Anda dengan `@step` dekorator untuk membuat langkah, membuat dan menjalankan pipeline menggunakan langkah, dan menyesuaikan pengalaman untuk kasus penggunaan Anda.

Topik

- Buat pipa dengan fungsi yang @step didekorasi
- Jalankan pipa
- Konfigurasikan alur Anda
- Praktik Terbaik
- Batasan

Buat pipa dengan fungsi yang **`@step`** didekorasi

Anda dapat membuat pipeline dengan mengubah fungsi Python menjadi langkah-langkah pipeline menggunakan dekorator, membuat dependensi `@step` di antara fungsi-fungsi tersebut untuk membuat grafik pipeline (atau grafik asiklik terarah (DAG)), dan meneruskan simpul daun grafik itu sebagai daftar langkah ke pipeline. Bagian berikut menjelaskan prosedur ini secara rinci dengan contoh.

Topik

- Ubah fungsi menjadi langkah
- Buat dependensi di antara langkah-langkah
- Gunakan ConditionStep dengan langkah-langkah @step yang didekorasi
- Tentukan pipeline menggunakan DelayedReturn output dari langkah-langkah
- Membuat Alur

Ubah fungsi menjadi langkah

Untuk membuat langkah menggunakan `@step` dekorator, beri anotasi fungsi dengan. `@step` Contoh berikut menunjukkan fungsi `@step` -decorated yang memproses data.

```
from sagemaker.workflow.function_step import step

@step
def preprocess(raw_data):
    df = pandas.read_csv(raw_data)
    ...
    return procesed_dataframe

step_process_result = preprocess(raw_data)
```

Saat Anda memanggil fungsi `@step` -decorated, SageMaker mengembalikan `DelayedReturn` instance alih-alih menjalankan fungsi. Sebuah `DelayedReturn` instance adalah proxy untuk pengembalian aktual dari fungsi itu. `DelayedReturn`Instance dapat diteruskan ke fungsi lain sebagai argumen atau langsung ke instance pipeline sebagai langkah. Untuk informasi tentang `DelayedReturn` kelas, lihat sagemaker.workflow.function_step. DelayedReturn.

Buat dependensi di antara langkah-langkah

Saat Anda membuat dependensi di antara dua langkah, Anda membuat koneksi antara langkah-langkah dalam grafik pipeline Anda. Bagian berikut memperkenalkan beberapa cara Anda dapat membuat dependensi di antara langkah-langkah pipeline Anda.

Dependensi data melalui argumen masukan

Melewati `DelayedReturn` output dari satu fungsi sebagai input ke fungsi lain secara otomatis menciptakan ketergantungan data dalam pipeline DAG. Dalam contoh berikut, meneruskan `DelayedReturn` output fungsi ke `preprocess` `train` fungsi menciptakan ketergantungan antara `preprocess` dan`train`.

```
from sagemaker.workflow.function_step import step

@step
def preprocess(raw_data):
    df = pandas.read_csv(raw_data)
    ...
    return procesed_dataframe

@step
def train(training_data):
    ...
    return trained_model
```

```
step_process_result = preprocess(raw_data)
step_train_result = train(step_process_result)
```

Contoh sebelumnya mendefinisikan fungsi pelatihan yang dihiasi dengan`@step`. Ketika fungsi ini dipanggil, ia menerima `DelayedReturn` output dari langkah pipa preprocessing sebagai input. Memanggil fungsi pelatihan mengembalikan `DelayedReturn` instance lain. Instance ini menyimpan informasi tentang semua langkah sebelumnya yang didefinisikan dalam fungsi itu (yaitu, `preprocess` langkah dalam contoh ini) yang membentuk saluran DAG.

Pada contoh sebelumnya, `preprocess` fungsi mengembalikan nilai tunggal. Untuk jenis pengembalian yang lebih kompleks seperti daftar atau tupel, lihat. Batasan

Menetapkan dependensi kustom

Pada contoh sebelumnya, `train` fungsi menerima `DelayedReturn` output `preprocess` dan menciptakan ketergantungan. Jika Anda ingin mendefinisikan ketergantungan secara eksplisit tanpa melewatkan output langkah sebelumnya, gunakan `add_depends_on` fungsi dengan langkah tersebut. Anda dapat menggunakan `get_step()` fungsi untuk mengambil langkah yang mendasari dari `DelayedReturn` instance-nya, dan kemudian memanggil `add_depends_on` _on dengan ketergantungan sebagai input. Untuk melihat definisi `get_step()` fungsi, lihat sagemaker.workflow.step_outputs.get_step. Contoh berikut menunjukkan cara membuat ketergantungan antara `preprocess` dan `train` menggunakan `get_step()` dan`add_depends_on()`.

```
from sagemaker.workflow.step_outputs import get_step

@step
def preprocess(raw_data):
    df = pandas.read_csv(raw_data)
    ...
    processed_data = ..
    return s3.upload(processed_data)

@step
def train():
    training_data = s3.download(....)
    ...
    return trained_model

step_process_result = preprocess(raw_data)
```

```
step_train_result = train()

get_step(step_train_result).add_depends_on(step_process_result)
```

Meneruskan data ke dan dari fungsi **`@step`** yang didekorasi ke langkah pipa tradisional

Anda dapat membuat pipa yang mencakup langkah yang `@step` didekorasi dan langkah pipa tradisional dan meneruskan data di antara mereka. Misalnya, Anda dapat menggunakan `ProcessingStep` untuk memproses data dan meneruskan hasilnya ke fungsi pelatihan `@step` yang didekorasi. Dalam contoh berikut, langkah pelatihan yang `@step` didekorasi mereferensikan output dari langkah pemrosesan.

```
# Define processing step

from sagemaker.sklearn.processing import SKLearnProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker.workflow.steps import ProcessingStep

sklearn_processor = SKLearnProcessor(
    framework_version='1.2-1',
    role='arn:aws:iam::123456789012:role/SagemakerExecutionRole',
    instance_type='ml.m5.large',
    instance_count='1',
)

inputs = [
    ProcessingInput(source=input_data, destination="/opt/ml/processing/input"),
]
outputs = [
    ProcessingOutput(output_name="train", source="/opt/ml/processing/train"),
    ProcessingOutput(output_name="validation", source="/opt/ml/processing/validation"),
    ProcessingOutput(output_name="test", source="/opt/ml/processing/test")
]

process_step = ProcessingStep(
    name="MyProcessStep",
    step_args=sklearn_processor.run(inputs=inputs,
 outputs=outputs,code='preprocessing.py'),
)
```

```
# Define a @step-decorated train step which references the
# output of a processing step
```

```
@step
def train(train_data_path, test_data_path):
    ...
    return trained_model

step_train_result = train(
   process_step.properties.ProcessingOutputConfig.Outputs["train"].S3Output.S3Uri,
   process_step.properties.ProcessingOutputConfig.Outputs["test"].S3Output.S3Uri,
)
```

Gunakan **`ConditionStep`** dengan langkah-langkah **`@step`** yang didekorasi

SageMaker Pipelines mendukung `ConditionStep` kelas yang mengevaluasi hasil dari langkah-langkah sebelumnya untuk memutuskan tindakan apa yang harus diambil dalam pipeline. Anda dapat menggunakan `ConditionStep` dengan langkah `@step` yang didekorasi juga. Untuk menggunakan output dari setiap langkah `@step` yang didekorasi dengan`ConditionStep`, masukkan output dari langkah itu sebagai argumen untuk`ConditionStep`. Dalam contoh berikut, langkah kondisi menerima output dari langkah evaluasi model `@step` yang didekorasi.

```
# Define steps

@step(name="evaluate")
def evaluate_model():
    # code to evaluate the model
    return {
        "rmse":rmse_value
    }

@step(name="register")
def register_model():
    # code to register the model
    ...
```

```
# Define ConditionStep

from sagemaker.workflow.condition_step import ConditionStep
from sagemaker.workflow.conditions import ConditionGreaterThanOrEqualTo
from sagemaker.workflow.fail_step import FailStep

conditionally_register = ConditionStep(
    name="conditional_register",
```

```
    conditions=[
        ConditionGreaterThanOrEqualTo(
            # Output of the evaluate step must be json serializable
            left=evaluate_model()["rmse"],  #
            right=5,
        )
    ],
    if_steps=[FailStep(name="Fail", error_message="Model performance is not good
 enough")],
    else_steps=[register_model()],
)
```

Tentukan pipeline menggunakan **`DelayedReturn`** output dari langkah-langkah

Anda mendefinisikan alur dengan cara yang sama apakah Anda menggunakan `@step` dekorator atau tidak. Ketika Anda meneruskan `DelayedReturn` instance ke pipeline Anda, Anda tidak perlu melewati daftar lengkap langkah-langkah untuk membangun pipeline. SDK secara otomatis menyimpulkan langkah-langkah sebelumnya berdasarkan dependensi yang Anda tentukan. Semua langkah sebelumnya dari `Step` objek yang Anda lewatkan ke pipa atau `DelayedReturn` objek termasuk dalam grafik pipa. Dalam contoh berikut, pipa menerima `DelayedReturn` objek untuk `train` fungsi tersebut. SageMaker menambahkan `preprocess` langkah, sebagai langkah sebelumnya`train`, ke grafik pipeline.

```
from sagemaker.workflow.pipeline import Pipeline

pipeline = Pipeline(
    name="<pipeline-name>",
    steps=[step_train_result],
    sagemaker_session=<sagemaker-session>,
)
```

Jika tidak ada data atau dependensi khusus di antara langkah-langkah dan Anda menjalankan beberapa langkah secara paralel, grafik pipeline memiliki lebih dari satu simpul daun. Berikan semua node daun ini dalam daftar ke `steps` argumen dalam definisi pipeline Anda, seperti yang ditunjukkan pada contoh berikut:

```
@step
def process1():
    ...
    return data
```

```
@step
def process2():
   ...
   return data

step_process1_result = process1()
step_process2_result = process2()

pipeline = Pipeline(
    name="<pipeline-name>",
    steps=[step_process1_result, step_process2_result],
    sagemaker_session=sagemaker-session,
)
```

Saat pipeline berjalan, kedua langkah berjalan secara paralel.

Anda hanya meneruskan simpul daun grafik ke pipeline karena simpul daun berisi informasi tentang semua langkah sebelumnya yang ditentukan melalui data atau dependensi khusus. Ketika mengkompilasi pipa, SageMaker juga menyimpulkan semua langkah selanjutnya yang membentuk grafik pipa dan menambahkan masing-masing sebagai langkah terpisah ke pipa.

Membuat Alur

Buat alur dengan memanggil`pipeline.create()`, seperti yang ditunjukkan pada potongan berikut. Untuk detailnya`create()`, lihat SageMaker.Workflow.Pipeline.Pipeline.Create.

```
role = "pipeline-role"
pipeline.create(role)
```

Saat Anda memanggil`pipeline.create()`, SageMaker mengkompilasi semua langkah yang didefinisikan sebagai bagian dari instance pipeline. SageMaker mengunggah fungsi serial, argumen, dan semua artefak terkait langkah lainnya ke Amazon S3.

Data berada di bucket S3 sesuai dengan struktur berikut:

```
s3_root_uri/
    pipeline_name/
        sm_rf_user_ws/
            workspace.zip  # archive of the current working directory (workdir)
        step_name/
            timestamp/
                arguments/                # serialized function arguments
```

```
                function/                 # serialized function
                pre_train_dependencies/   # any dependencies and pre_execution scripts
provided for the step
       execution_id/
           step_name/
                results     # returned output from the serialized function including
the model
```

`s3_root_uri`didefinisikan dalam file SageMaker konfigurasi dan berlaku untuk seluruh pipeline. Jika tidak terdefinisi, SageMaker bucket default digunakan.

> **Note**
>
> Setiap kali SageMaker mengkompilasi pipeline, SageMaker menyimpan fungsi, argumen, dan dependensi serial langkah-langkah dalam folder yang diberi cap waktu dengan waktu saat ini. Ini terjadi setiap kali Anda berlari`pipeline.create()`,`pipeline.update()`, `pipeline.upsert()` atau`pipeline.definition()`.

Jalankan pipa

Mulai proses pipeline baru dengan `pipeline.start()` fungsi seperti yang Anda lakukan untuk menjalankan SageMaker pipeline tradisional. Untuk informasi tentang `start()` fungsi, lihat SageMaker.workflow.pipeline.pipeline.pipeline.start.

> **Note**
>
> Langkah yang didefinisikan menggunakan `@step` dekorator berjalan sebagai pekerjaan pelatihan. Karena itu, perhatikan batas-batas berikut:
>
> - Batas instans dan batas pekerjaan pelatihan di akun Anda. Perbarui batas Anda untuk menghindari masalah pembatasan atau batas sumber daya.
> - Biaya moneter yang terkait dengan setiap langkah pelatihan dalam pipa. Untuk detail selengkapnya, lihat SageMaker Harga Amazon.

Mengambil hasil dari pipeline yang dijalankan secara lokal

Untuk melihat hasil dari setiap langkah dari pipeline run, gunakan execution.result (), seperti yang ditunjukkan pada cuplikan berikut:

```
execution = pipeline.start()
execution.result(step_name="train")
```

> **Note**
>
> SageMaker Pipelines tidak mendukung `execution.result()` dalam mode lokal.

Anda hanya dapat mengambil hasil untuk satu langkah dalam satu waktu. Jika nama langkah dihasilkan oleh SageMaker, Anda dapat mengambil nama langkah dengan memanggil `list_steps` sebagai berikut:

```
execution.list_step()
```

Jalankan pipeline secara lokal

Anda dapat menjalankan pipa dengan langkah-langkah yang `@step` didekorasi secara lokal seperti yang Anda lakukan untuk langkah-langkah pipa tradisional. Untuk detail tentang proses pipeline mode lokal, lihatMode. Untuk menggunakan mode lokal, berikan definisi pipeline sebagai `LocalPipelineSession` pengganti a`SageMakerSession`, seperti yang ditunjukkan pada contoh berikut:

```
from sagemaker.workflow.function_step import step
from sagemaker.workflow.pipeline import Pipeline
from sagemaker.workflow.pipeline_context import LocalPipelineSession

@step
def train():
    training_data = s3.download(....)
    ...
    return trained_model

step_train_result = train()

local_pipeline_session = LocalPipelineSession()

local_pipeline = Pipeline(
    name="<pipeline-name>",
    steps=[step_train_result],
    sagemaker_session=local_pipeline_session # needed for local mode
```

```
)

local_pipeline.create(role_arn="role_arn")

# pipeline runs locally
execution = local_pipeline.start()
```

Konfigurasikan alur Anda

Anda disarankan untuk menggunakan file SageMaker konfigurasi untuk mengatur default untuk pipeline. Untuk informasi tentang file SageMaker konfigurasi, lihat Mengkonfigurasi dan menggunakan default dengan Python SDK. SageMaker Konfigurasi apa pun yang ditambahkan ke file konfigurasi berlaku untuk semua langkah dalam pipeline. Jika Anda ingin mengganti opsi untuk salah satu langkah, berikan nilai baru dalam argumen `@step` dekorator.

Konfigurasi `@step` dekorator dalam file konfigurasi identik dengan konfigurasi `@remote` dekorator. Untuk menyiapkan ARN peran pipeline dan tag pipeline di file konfigurasi, gunakan `Pipeline` bagian yang ditunjukkan dalam cuplikan berikut:

```
SchemaVersion: '1.0'
SageMaker:
  Pipeline:
    RoleArn: 'arn:aws:iam::555555555555:role/IMRole'
    Tags:
    - Key: 'tag_key'
      Value: 'tag_value'
```

Untuk sebagian besar default yang dapat Anda atur dalam file konfigurasi, Anda juga dapat mengganti dengan meneruskan nilai baru ke dekorator. `@step` Misalnya, Anda dapat mengganti tipe instans yang disetel di file konfigurasi untuk langkah preprocessing Anda, seperti yang ditunjukkan pada contoh berikut:

```
@step(instance_type="ml.m5.large")
def preprocess(raw_data):
    df = pandas.read_csv(raw_data)
    ...
    return procesed_dataframe
```

Beberapa argumen bukan bagian dari daftar parameter `@step` dekorator — ini dapat dikonfigurasi untuk seluruh pipeline hanya melalui file konfigurasi. SageMaker Mereka terdaftar sebagai berikut:

- `sagemaker_session(sagemaker.session.Session)`: SageMaker Sesi yang mendasari panggilan layanan SageMaker delegasi. Jika tidak ditentukan, sesi dibuat menggunakan konfigurasi default sebagai berikut:

```
SageMaker:
  PythonSDK:
    Modules:
      Session:
        DefaultS3Bucket: 'default_s3_bucket'
        DefaultS3ObjectKeyPrefix: 'key_prefix'
```

- `custom_file_filter(CustomFileFilter)`: `CustomFileFilter` Objek yang menentukan direktori dan file lokal untuk disertakan dalam langkah pipeline. Jika tidak ditentukan, nilai ini default ke. `None` `custom_file_filter`Agar berlaku, Anda harus mengatur `IncludeLocalWorkdir` ke`True`. Contoh berikut menunjukkan konfigurasi yang mengabaikan semua file notebook, dan file dan direktori bernama. `data`

```
SchemaVersion: '1.0'
SageMaker:
  PythonSDK:
    Modules:
      RemoteFunction:
        IncludeLocalWorkDir: true
        CustomFileFilter:
          IgnoreNamePatterns: # files or directories to ignore
          - "*.ipynb" # all notebook files
          - "data" # folder or file named "data"
```

  Untuk detail selengkapnya tentang cara menggunakannya `IncludeLocalWorkdirCustomFileFilter`, lihat[Menggunakan kode modular dengan dekorator @remote](#).

- `s3_root_uri (str)`: Folder root Amazon S3 yang SageMaker mengunggah arsip kode dan data. Jika tidak ditentukan, SageMaker bucket default digunakan.
- `s3_kms_key (str)`: Kunci yang digunakan untuk mengenkripsi data input dan output. Anda hanya dapat mengonfigurasi argumen ini di file SageMaker konfigurasi dan argumen berlaku untuk semua langkah yang ditentukan dalam pipeline. Jika tidak ditentukan, nilai default ke. `None` Lihat cuplikan berikut untuk contoh konfigurasi kunci S3 KMS:

```
SchemaVersion: '1.0'
```

```
SageMaker:
  PythonSDK:
    Modules:
      RemoteFunction:
        S3KmsKeyId: 's3kmskeyid'
        S3RootUri: 's3://my-bucket/my-project
```

Praktik Terbaik

Bagian berikut menyarankan praktik terbaik untuk diikuti saat Anda menggunakan `@step` dekorator untuk langkah-langkah pipa Anda.

Gunakan kolam hangat

Untuk langkah pipa yang lebih cepat berjalan, gunakan fungsionalitas penyatuan hangat yang disediakan untuk pekerjaan pelatihan. Anda dapat mengaktifkan fungsionalitas kolam hangat dengan memberikan `keep_alive_period_in_seconds` argumen kepada `@step` dekorator seperti yang ditunjukkan dalam cuplikan berikut:

```
@step(
   keep_alive_period_in_seconds=900
)
```

Untuk informasi selengkapnya tentang kolam hangat, lihatMelatih Menggunakan Kolam Hangat yang SageMaker Dikelola.

Susun direktori Anda

Anda disarankan untuk menggunakan modul kode saat menggunakan `@step` dekorator. Letakkan `pipeline.py` modul, di mana Anda menjalankan fungsi langkah dan menentukan pipeline, di root ruang kerja. Struktur yang direkomendasikan ditampilkan sebagai berikut:

```
.
### config.yaml # the configuration file that define the infra settings
### requirements.txt # dependencies
### pipeline.py  # invoke @step-decorated functions and define the pipeline here
### steps/
| ### processing.py
| ### train.py
### data/
### test/
```

Batasan

Perhatikan keterbatasan berikut saat Anda menggunakan `@step` dekorator untuk langkah alur Anda.

Keterbatasan argumen fungsi

Saat Anda meneruskan argumen input ke fungsi `@step` -decorated, batasan berikut berlaku:

- Anda dapat meneruskan`DelayedReturn`, `Properties` (langkah dari jenis lain),`Parameter`, dan `ExecutionVariable` objek ke fungsi `@step` -decorated sebagai argumen. Tapi fungsi `@step` yang didekorasi tidak mendukung `JsonGet` dan `Join` objek sebagai argumen.
- Anda tidak dapat langsung mengakses variabel pipeline dari suatu `@step` fungsi. Contoh berikut menghasilkan kesalahan:

```
param = ParameterInteger(name="<parameter-name>", default_value=10)

@step
def func():
    print(param)

func() # this raises a SerializationError
```

- Anda tidak dapat membuat sarang variabel pipeline di objek lain dan meneruskannya ke `@step` fungsi. Contoh berikut menghasilkan kesalahan:

```
param = ParameterInteger(name="<parameter-name>", default_value=10)

@step
def func(arg):
    print(arg)

func(arg=(param,)) # this raises a SerializationError because param is nested in a
 tuple
```

- Karena input dan output dari suatu fungsi diserialisasi, ada batasan pada jenis data yang dapat diteruskan sebagai input atau output dari suatu fungsi. Lihat bagian Serialisasi dan deserialisasi data Memanggil fungsi untuk detail selengkapnya. Pembatasan yang sama berlaku untuk fungsi `@step` yang didekorasi.
- Objek apa pun yang memiliki klien boto tidak dapat diserialisasi, oleh karena itu Anda tidak dapat meneruskan objek seperti input ke atau output dari fungsi `@step` -decorated. Misalnya, kelas klien SDK SageMaker Python sepertiEstimator,Predictor, dan tidak `Processor` dapat diserialisasi.

Impor fungsi

Anda harus mengimpor pustaka yang diperlukan oleh langkah di dalam daripada di luar fungsi. Jika Anda mengimpornya di lingkup global, Anda berisiko mengalami tabrakan impor saat membuat serial fungsi. Misalnya, `sklearn.pipeline.Pipeline` bisa diganti dengan. `sagemaker.workflow.pipeline.Pipeline`

Mereferensikan anggota anak dari nilai pengembalian fungsi

Jika Anda mereferensikan anggota turunan dari nilai pengembalian fungsi `@step` -decorated, batasan berikut berlaku:

- Anda dapat mereferensikan anggota anak dengan `[]` jika `DelayedReturn` objek mewakili Tuple, list atau dict, seperti yang ditunjukkan pada contoh berikut:

```
delayed_return[0]
delayed_return["a_key"]
delayed_return[1]["a_key"]
```

- Anda tidak dapat membongkar keluaran tuple atau daftar karena panjang yang tepat dari tupel atau daftar yang mendasarinya tidak dapat diketahui saat Anda menjalankan fungsi tersebut. Contoh berikut menghasilkan kesalahan:

```
a, b, c = func() # this raises ValueError
```

- Anda tidak dapat mengulangi `DelayedReturn` objek. Contoh berikut menimbulkan kesalahan:

```
for item in func(): # this raises a NotImplementedError
```

- Anda tidak dapat mereferensikan anggota anak yang sewenang-wenang dengan '`.`'. Contoh berikut menghasilkan kesalahan:

```
delayed_return.a_child # raises AttributeError
```

Fitur pipeline yang ada yang tidak didukung

Anda tidak dapat menggunakan `@step` dekorator dengan fitur pipeline berikut:

- Caching langkah pipa
- Eksekusi selektif

- File properti

Lulus Data Antar Langkah

Ketika Anda perlu mengambil informasi dari output dari langkah pipa, Anda dapat menggunakannya`JsonGet`. `JsonGet`membantu Anda mengekstrak informasi dari Amazon S3 atau file properti. Bagian berikut menjelaskan metode yang dapat Anda gunakan untuk mengekstrak keluaran langkah dengan`JsonGet`.

Lulus data antar langkah dengan Amazon S3

Anda dapat menggunakan `JsonGet` dalam `ConditionStep` untuk mengambil output JSON langsung dari Amazon S3. URI Amazon S3 dapat berupa `Std:Join` fungsi yang berisi string primitif, variabel pipeline run, atau parameter pipeline. Contoh berikut menunjukkan cara menggunakan `JsonGet` di`ConditionStep`:

```
# Example json file in s3 bucket generated by a processing_step
{
   "Output": [5, 10]
}

cond_lte = ConditionLessThanOrEqualTo(
    left=JsonGet(
        step_name="<step-name>",
        s3_uri="<s3-path-to-json>",
        json_path="Output[1]"
    ),
    right=6.0
)
```

Jika Anda menggunakan `JsonGet` jalur Amazon S3 di langkah kondisi, Anda harus secara eksplisit menambahkan ketergantungan antara langkah kondisi dan langkah yang menghasilkan output JSON. Dalam contoh berikut, langkah kondisi dibuat dengan ketergantungan pada langkah pemrosesan:

```
cond_step = ConditionStep(
        name="<step-name>",
        conditions=[cond_lte],
        if_steps=[fail_step],
        else_steps=[register_model_step],
        depends_on=[processing_step],
```

```
)
```

Lulus data antar langkah dengan file properti

Gunakan file properti untuk menyimpan informasi dari output dari langkah pemrosesan. Ini sangat berguna ketika menganalisis hasil dari langkah pemrosesan untuk memutuskan bagaimana langkah bersyarat harus dijalankan. `JsonGet`Fungsi memproses file properti dan memungkinkan Anda untuk menggunakan JsonPath notasi untuk query file JSON properti. Untuk informasi lebih lanjut tentang JsonPath notasi, lihat JsonPath repo.

Untuk menyimpan file properti untuk digunakan nanti, Anda harus terlebih dahulu membuat `PropertyFile` instance dengan format berikut. `path`Parameter adalah nama file JSON tempat file properti disimpan. Setiap `output_name` harus cocok dengan `output_name` `ProcessingOutput` yang Anda tentukan dalam langkah pemrosesan Anda. Hal ini memungkinkan file properti untuk menangkap `ProcessingOutput` dalam langkah.

```
from sagemaker.workflow.properties import PropertyFile

<property_file_instance> = PropertyFile(
    name="<property_file_name>",
    output_name="<processingoutput_output_name>",
    path="<path_to_json_file>"
)
```

Saat membuat `ProcessingStep` instance, tambahkan `property_files` parameter untuk mencantumkan semua file parameter yang harus diindeks oleh layanan Amazon SageMaker Model Building Pipelines. Sumber daya yang dimaksud akan menyimpan file properti untuk digunakan nanti.

```
property_files=[<property_file_instance>]
```

Untuk menggunakan file properti Anda dalam langkah kondisi, tambahkan `property_file` ke kondisi yang Anda teruskan ke langkah kondisi Anda seperti yang ditunjukkan pada contoh berikut untuk menanyakan file JSON untuk properti yang Anda inginkan menggunakan `json_path` parameter.

```
cond_lte = ConditionLessThanOrEqualTo(
    left=JsonGet(
        step_name=step_eval.name,
        property_file=<property_file_instance>,
        json_path="mse"
```

```
    ),
    right=6.0
)
```

Untuk contoh lebih mendalam, lihat File properti di Amazon SageMaker Python SDK.

Langkah Alur Caching

Saat Anda menggunakan caching tanda tangan langkah, SageMaker Pipelines mencoba menemukan langkah pipeline Anda yang berjalan sebelumnya dengan nilai yang sama untuk atribut tertentu. Jika ditemukan, SageMaker Pipelines menyebarkan output dari proses sebelumnya daripada menghitung ulang langkah. Atribut yang dicentang khusus untuk jenis langkah, dan tercantum dalamAtribut kunci cache default berdasarkan tipe langkah pipa.

Anda harus memilih untuk melakukan caching langkah — ini tidak aktif secara default. Saat Anda mengaktifkan caching langkah, Anda juga harus menentukan batas waktu. Batas waktu ini menentukan berapa umur lari sebelumnya untuk tetap menjadi kandidat untuk digunakan kembali.

Step caching hanya mempertimbangkan proses yang berhasil - tidak pernah menggunakan kembali proses yang gagal. Ketika beberapa proses yang berhasil ada dalam periode batas waktu, SageMaker Pipelines menggunakan hasilnya untuk menjalankan sukses terbaru. Jika tidak ada pertandingan berjalan yang berhasil dalam periode batas waktu, SageMaker Pipelines menjalankan ulang langkahnya. Jika pelaksana menemukan proses sebelumnya yang memenuhi kriteria tetapi masih dalam proses, kedua langkah terus berjalan dan memperbarui cache jika berhasil.

Step caching hanya dicakup untuk pipeline individu, jadi Anda tidak dapat menggunakan kembali langkah dari pipeline lain meskipun ada kecocokan tanda tangan langkah.

Langkah caching tersedia untuk jenis langkah berikut:

- Pemrosesan
- Pelatihan
- Penyetelan
- AutoML
- Transformasi
- `ClarifyCheck`
- `QualityCheck`
- EMR

Topik

Mengaktifkan caching langkah

Untuk mengaktifkan caching langkah, Anda harus menambahkan `CacheConfig` properti ke definisi langkah.

`CacheConfig`properti menggunakan format berikut dalam file definisi pipeline:

```
{
    "CacheConfig": {
        "Enabled": false,
        "ExpireAfter": "<time>"
    }
}
```

`Enabled`Bidang menunjukkan apakah caching dihidupkan untuk langkah tertentu. Anda dapat mengatur bidang k`e`true, yang memberitahu SageMaker untuk mencoba menemukan langkah sebelumnya dengan atribut yang sama. Atau, Anda dapat mengatur bidang k`e`false, yang memberitahu SageMaker untuk menjalankan langkah setiap kali pipeline berjalan. `ExpireAfter`adalah string dalam format durasi ISO 8601 yang menentukan periode batas waktu. `ExpireAfter`Durasi bisa berupa nilai tahun, bulan, minggu, hari, jam, jam, atau menit. Setiap nilai terdiri dari angka diikuti dengan huruf yang menunjukkan satuan durasi. Sebagai contoh:

- “30d” = 30 hari
- “5y” = 5 tahun
- “T16m” = 16 menit
- “30dT5h” = 30 hari dan 5 jam.

Diskusi berikut menjelaskan prosedur untuk mengaktifkan caching untuk pipeline baru atau yang sudah ada sebelumnya menggunakan Amazon Python SageMaker SDK.

Aktifkan caching untuk saluran pipa baru

Untuk pipeline baru, inisialisasi `CacheConfig` instance dengan `enable_caching=True` dan berikan sebagai input ke langkah pipeline Anda. Contoh berikut mengaktifkan caching dengan periode batas waktu 1 jam untuk langkah pelatihan:

```
from sagemaker.workflow.pipeline_context import PipelineSession
from sagemaker.workflow.steps import CacheConfig

cache_config = CacheConfig(enable_caching=True, expire_after="PT1H")
estimator = Estimator(..., sagemaker_session=PipelineSession())

step_train = TrainingStep(
    name="TrainAbaloneModel",
    step_args=estimator.fit(inputs=inputs),
    cache_config=cache_config
)
```

Aktifkan caching untuk pipeline yang sudah ada sebelumnya

Untuk mengaktifkan caching untuk pipeline yang sudah ada sebelumnya dan sudah ditentukan, nyalakan `enable_caching` properti untuk langkah tersebut, dan setel `expire_after` ke nilai batas waktu. Terakhir, perbarui pipeline dengan `pipeline.upsert()` atau`pipeline.update()`. Setelah Anda menjalankannya lagi, contoh kode berikut mengaktifkan caching dengan periode batas waktu 1 jam untuk langkah pelatihan:

```
from sagemaker.workflow.pipeline_context import PipelineSession
from sagemaker.workflow.steps import CacheConfig
from sagemaker.workflow.pipeline import Pipeline

cache_config = CacheConfig(enable_caching=True, expire_after="PT1H")
estimator = Estimator(..., sagemaker_session=PipelineSession())

step_train = TrainingStep(
    name="TrainAbaloneModel",
    step_args=estimator.fit(inputs=inputs),
    cache_config=cache_config
)

# define pipeline
pipeline = Pipeline(
    steps=[step_train]
)
```

```
# additional step for existing pipelines
pipeline.update()
# or, call upsert() to update the pipeline
# pipeline.upsert()
```

Atau, perbarui konfigurasi cache setelah Anda mendefinisikan pipeline (yang sudah ada sebelumnya), memungkinkan satu kode berkelanjutan dijalankan. Contoh kode berikut menunjukkan metode ini:

```
# turn on caching with timeout period of one hour
pipeline.steps[0].cache_config.enable_caching = True
pipeline.steps[0].cache_config.expire_after = "PT1H"

# additional step for existing pipelines
pipeline.update()
# or, call upsert() to update the pipeline
# pipeline.upsert()
```

Untuk contoh kode yang lebih rinci dan diskusi tentang bagaimana parameter SDK Python memengaruhi caching, lihat Konfigurasi Caching dalam dokumentasi Amazon Python SDK. SageMaker

Mematikan caching langkah

Langkah pipeline tidak dijalankan kembali jika Anda mengubah atribut apa pun yang tidak tercantum Atribut kunci cache default berdasarkan tipe langkah pipa untuk jenis langkahnya. Namun, Anda dapat memutuskan bahwa Anda tetap ingin langkah pipeline dijalankan kembali. Dalam hal ini, Anda perlu mematikan caching langkah.

Untuk menonaktifkan caching langkah, atur `Enabled` atribut di `CacheConfig` properti definisi langkah dalam definisi langkah ke`false`, seperti yang ditunjukkan pada cuplikan kode berikut:

```
{
    "CacheConfig": {
        "Enabled": false,
        "ExpireAfter": "<time>"
    }
}
```

Perhatikan bahwa `ExpireAfter` atribut diabaikan `Enabled` ketika`false`.

Untuk mematikan caching untuk langkah pipeline menggunakan Amazon SageMaker Python SDK, tentukan pipeline langkah pipeline Anda, matikan properti, dan `enable_caching` perbarui pipeline.

Setelah Anda menjalankannya lagi, contoh kode berikut mematikan caching untuk langkah pelatihan:

```
from sagemaker.workflow.pipeline_context import PipelineSession
from sagemaker.workflow.steps import CacheConfig
from sagemaker.workflow.pipeline import Pipeline

cache_config = CacheConfig(enable_caching=False, expire_after="PT1H")
estimator = Estimator(..., sagemaker_session=PipelineSession())

step_train = TrainingStep(
    name="TrainAbaloneModel",
    step_args=estimator.fit(inputs=inputs),
    cache_config=cache_config
)

# define pipeline
pipeline = Pipeline(
    steps=[step_train]
)

# update the pipeline
pipeline.update()
# or, call upsert() to update the pipeline
# pipeline.upsert()
```

Atau, matikan `enable_caching` properti setelah Anda telah menentukan pipeline, memungkinkan satu kode berkelanjutan dijalankan. Contoh kode berikut menunjukkan solusi ini:

```
# turn off caching for the training step
pipeline.steps[0].cache_config.enable_caching = False

# update the pipeline
pipeline.update()
# or, call upsert() to update the pipeline
# pipeline.upsert()
```

Untuk contoh kode yang lebih rinci dan diskusi tentang bagaimana parameter SDK Python memengaruhi caching, lihat Konfigurasi Caching dalam dokumentasi Amazon Python SDK. SageMaker

Atribut kunci cache default berdasarkan tipe langkah pipa

Saat memutuskan apakah akan menggunakan kembali langkah pipeline sebelumnya atau menjalankan kembali langkah tersebut, SageMaker Pipelines memeriksa untuk melihat apakah atribut tertentu telah berubah. Jika kumpulan atribut berbeda dari semua proses sebelumnya dalam periode batas waktu, langkah berjalan lagi. Atribut ini termasuk artefak input, spesifikasi aplikasi atau algoritma, dan variabel lingkungan.

Daftar berikut menunjukkan setiap jenis langkah pipeline dan atribut yang, jika diubah, memulai tayangan ulang langkah. Untuk informasi selengkapnya tentang parameter SDK Python yang digunakan untuk membuat atribut berikut, lihat Konfigurasi Caching di dokumentasi Amazon SageMaker Python SDK.

Langkah pemrosesan

- AppSpecification
- Environment
- ProcessingInputs. Atribut ini berisi informasi tentang skrip preprocessing.

Langkah pelatihan

- AlgorithmSpecification
- CheckpointConfig
- DebugHookConfig
- DebugRuleConfigurations
- Environment
- HyperParameters
- InputDataConfig. Atribut ini berisi informasi tentang skrip pelatihan.

Langkah penyetelan

- HyperParameterTuningJobConfig
- TrainingJobDefinition. Atribut ini terdiri dari beberapa atribut anak, tidak semuanya menyebabkan langkah dijalankan kembali. Atribut anak yang dapat menimbulkan tayangan ulang (jika diubah) adalah:

  - AlgorithmSpecification
  - HyperParameterRanges
  - InputDataConfig
  - StaticHyperParameters
  - TuningObjective
- TrainingJobDefinitions

Langkah AutoML

- AutoMlJobConfig. Atribut ini terdiri dari beberapa atribut anak, tidak semuanya menyebabkan langkah dijalankan kembali. Atribut anak yang dapat menimbulkan tayangan ulang (jika diubah) adalah:
  - CompletionCriteria
  - CandidateGenerationConfig
  - DataSplitConfig
  - Mode
- AutoML JobObjective
- InputDataConfig
- ProblemType

Langkah transformasi

- DataProcessing
- Environment
- ModelName
- TransformInput

ClarifyCheck Langkah

- ClarifyCheckConfig
- CheckJobConfig

- SkipCheck
- RegisterNewBaseline
- ModelPackageGroupName
- SuppliedBaselineConstraints

QualityCheck Langkah

- QualityCheckConfig
- CheckJobConfig
- SkipCheck
- RegisterNewBaseline
- ModelPackageGroupName
- SuppliedBaselineConstraints
- SuppliedBaselineStatistics

Langkah EMR

- ClusterId
- StepConfig

Kontrol akses data cache

Ketika SageMaker pipeline berjalan, ia menyimpan parameter dan metadata yang terkait dengan SageMaker pekerjaan yang diluncurkan oleh pipeline dan menyimpannya untuk digunakan kembali dalam proses berikutnya. Metadata ini dapat diakses melalui berbagai sumber selain langkah-langkah pipeline yang di-cache, dan mencakup jenis berikut:

- Permintaan `Describe*Job`
- CloudWatch Log
- CloudWatch Acara
- CloudWatch Metrik
- SageMaker Cari

Perhatikan bahwa akses ke setiap sumber data dalam daftar dikendalikan oleh kumpulan izin IAM sendiri. Menghapus akses peran tertentu ke satu sumber data tidak mempengaruhi tingkat akses ke yang lain. Misalnya, admin akun mungkin menghapus izin IAM untuk `Describe*Job` permintaan dari peran pemanggil. Meskipun penelepon tidak dapat lagi membuat `Describe*Job` permintaan, mereka masih dapat mengambil metadata dari pipeline yang dijalankan dengan langkah-langkah cache selama mereka memiliki izin untuk menjalankan pipeline. Jika admin akun ingin menghapus akses ke metadata dari SageMaker pekerjaan tertentu sepenuhnya, mereka perlu menghapus izin untuk setiap layanan terkait yang menyediakan akses ke data.

Coba lagi Kebijakan untuk Langkah-langkah Pipa

Kebijakan coba lagi membantu Anda mencoba ulang langkah SageMaker Pipelines secara otomatis setelah terjadi kesalahan. Setiap langkah pipeline dapat menemukan pengecualian, dan pengecualian terjadi karena berbagai alasan. Dalam beberapa kasus, coba lagi dapat menyelesaikan masalah ini. Dengan kebijakan coba lagi untuk langkah-langkah pipeline, Anda dapat memilih apakah akan mencoba lagi langkah pipeline tertentu atau tidak.

Kebijakan coba lagi hanya mendukung langkah-langkah pipeline berikut:

- Langkah Pemrosesan
- Langkah Pelatihan
- Langkah 1: Lab
- Langkah AutoMl
- CreateModel Langkah
- RegisterModel Langkah
- Pembuatan
- Langkah Job Notebook

> **Note**
>
> Pekerjaan yang berjalan di dalam langkah penyetelan dan AutoML melakukan percobaan ulang secara internal dan tidak akan mencoba lagi jenis pengecualian, meskipun kebijakan coba `SageMaker.JOB_INTERNAL_ERROR` lagi dikonfigurasi. Anda dapat memprogram Strategi Coba Ulang Anda sendiri menggunakan SageMaker API.

Jenis pengecualian yang didukung untuk kebijakan coba lagi

Kebijakan coba lagi untuk langkah-langkah pipeline mendukung jenis pengecualian berikut:

- `Step.SERVICE_FAULT`: Pengecualian ini terjadi ketika kesalahan server internal atau kesalahan sementara terjadi saat memanggil layanan hilir. SageMaker Pipelines mencoba ulang jenis kesalahan ini secara otomatis. Dengan kebijakan coba lagi, Anda dapat mengganti operasi coba ulang default untuk jenis pengecualian ini.
- `Step.THROTTLING`: Pengecualian pelambatan dapat terjadi saat memanggil layanan hilir. SageMaker Pipelines mencoba ulang jenis kesalahan ini secara otomatis. Dengan kebijakan coba lagi, Anda dapat mengganti operasi coba ulang default untuk jenis pengecualian ini.
- `SageMaker.JOB_INTERNAL_ERROR`: Pengecualian ini terjadi ketika SageMaker pekerjaan kembali`InternalServerError`. Dalam hal ini, memulai pekerjaan baru dapat memperbaiki masalah sementara.
- `SageMaker.CAPACITY_ERROR`: SageMaker Pekerjaan itu mungkin menghadapi Amazon `EC2InsufficientCapacityErrors`, yang menyebabkan kegagalan SageMaker pekerjaan. Anda dapat mencoba lagi dengan memulai SageMaker pekerjaan baru untuk menghindari masalah.
- `SageMaker.RESOURCE_LIMIT`: Anda dapat melampaui kuota batas sumber daya saat menjalankan pekerjaan. SageMaker Anda dapat menunggu dan mencoba lagi menjalankan SageMaker pekerjaan setelah periode singkat dan melihat apakah sumber daya dirilis.

Skema JSON untuk kebijakan coba lagi

Kebijakan coba lagi untuk Pipelines memiliki skema JSON berikut:

```
"RetryPolicy": {
   "ExceptionType": [String]
   "IntervalSeconds": Integer
   "BackoffRate": Double
   "MaxAttempts": Integer
   "ExpireAfterMin": Integer
}
```

- `ExceptionType`: Bidang ini membutuhkan jenis pengecualian berikut dalam format array string.
  - `Step.SERVICE_FAULT`
  - `Step.THROTTLING`

  - `SageMaker.JOB_INTERNAL_ERROR`
  - `SageMaker.CAPACITY_ERROR`
  - `SageMaker.RESOURCE_LIMIT`
- `IntervalSeconds`(opsional): Jumlah detik sebelum percobaan ulang pertama (1 secara default). `IntervalSeconds`memiliki nilai maksimum 43.200 detik (12 jam).
- `BackoffRate`(Opsional): Pengali yang mana interval coba lagi meningkat selama setiap upaya (2.0 secara default).
- `MaxAttempts`(Opsional): Sebuah bilangan bulat positif yang mewakili jumlah maksimum upaya coba lagi (5 secara default). Jika kesalahan berulang kali lebih dari yang `MaxAttempts` ditentukan, percobaan ulang berhenti dan penanganan kesalahan normal dilanjutkan. Nilai 0 menentukan bahwa kesalahan tidak pernah dicoba ulang. `MaxAttempts`memiliki nilai maksimum 20.
- `ExpireAfterMin`(opsional): Sebuah bilangan bulat positif yang mewakili rentang waktu maksimum percobaan lagi. Jika kesalahan berulang kali setelah hitungan `ExpireAfterMin` menit dari langkah dieksekusi, percobaan ulang berhenti dan penanganan kesalahan normal dilanjutkan. Nilai 0 menentukan bahwa kesalahan tidak pernah dicoba ulang. `ExpireAfterMin` memiliki nilai maksimum 14.400 menit (10 hari).

> **Note**
>
> Hanya satu dari `MaxAttempts` atau `ExpireAfterMin` dapat diberikan, tetapi tidak keduanya; jika keduanya tidak ditentukan, `MaxAttempts` menjadi default. Jika kedua properti diidentifikasi dalam satu kebijakan, maka kebijakan coba lagi menghasilkan kesalahan validasi.

Mengonfigurasi kebijakan coba lagi

Berikut ini adalah contoh langkah pelatihan dengan kebijakan coba ulang.

```
{
    "Steps": [
        {
            "Name": "MyTrainingStep",
            "Type": "Training",
            "RetryPolicies": [
                {
                    "ExceptionType": [
```

```
                    "SageMaker.JOB_INTERNAL_ERROR",
                    "SageMaker.CAPACITY_ERROR"
                ],
                "IntervalSeconds": 1,
                "BackoffRate": 2,
                "MaxAttempts": 5
            }
        ]
    }
]
}
```

Berikut ini adalah contoh cara membuat `TrainingStep` in SDK for Python (Boto3) dengan kebijakan coba lagi.

```
from sagemaker.workflow.retry import (
    StepRetryPolicy,
    StepExceptionTypeEnum,
    SageMakerJobExceptionTypeEnum,
    SageMakerJobStepRetryPolicy
)

step_train = TrainingStep(
    name="MyTrainingStep",
    xxx,
    retry_policies=[
        // override the default
        StepRetryPolicy(
            exception_types=[
                StepExceptionTypeEnum.SERVICE_FAULT,
                StepExceptionTypeEnum.THROTTLING
            ],
            expire_after_mins=5,
            interval_seconds=10,
            backoff_rate=2.0
        ),
        // retry when resource limit quota gets exceeded
        SageMakerJobStepRetryPolicy(
            exception_types=[SageMakerJobExceptionTypeEnum.RESOURCE_LIMIT],
            expire_after_mins=120,
            interval_seconds=60,
            backoff_rate=2.0
        ),
```

```
        // retry when job failed due to transient error or EC2 ICE.
        SageMakerJobStepRetryPolicy(
            failure_reason_types=[
                SageMakerJobExceptionTypeEnum.INTERNAL_ERROR,
                SageMakerJobExceptionTypeEnum.CAPACITY_ERROR,
            ],
            max_attempts=10,
            interval_seconds=30,
            backoff_rate=2.0
        )
    ]
)
```

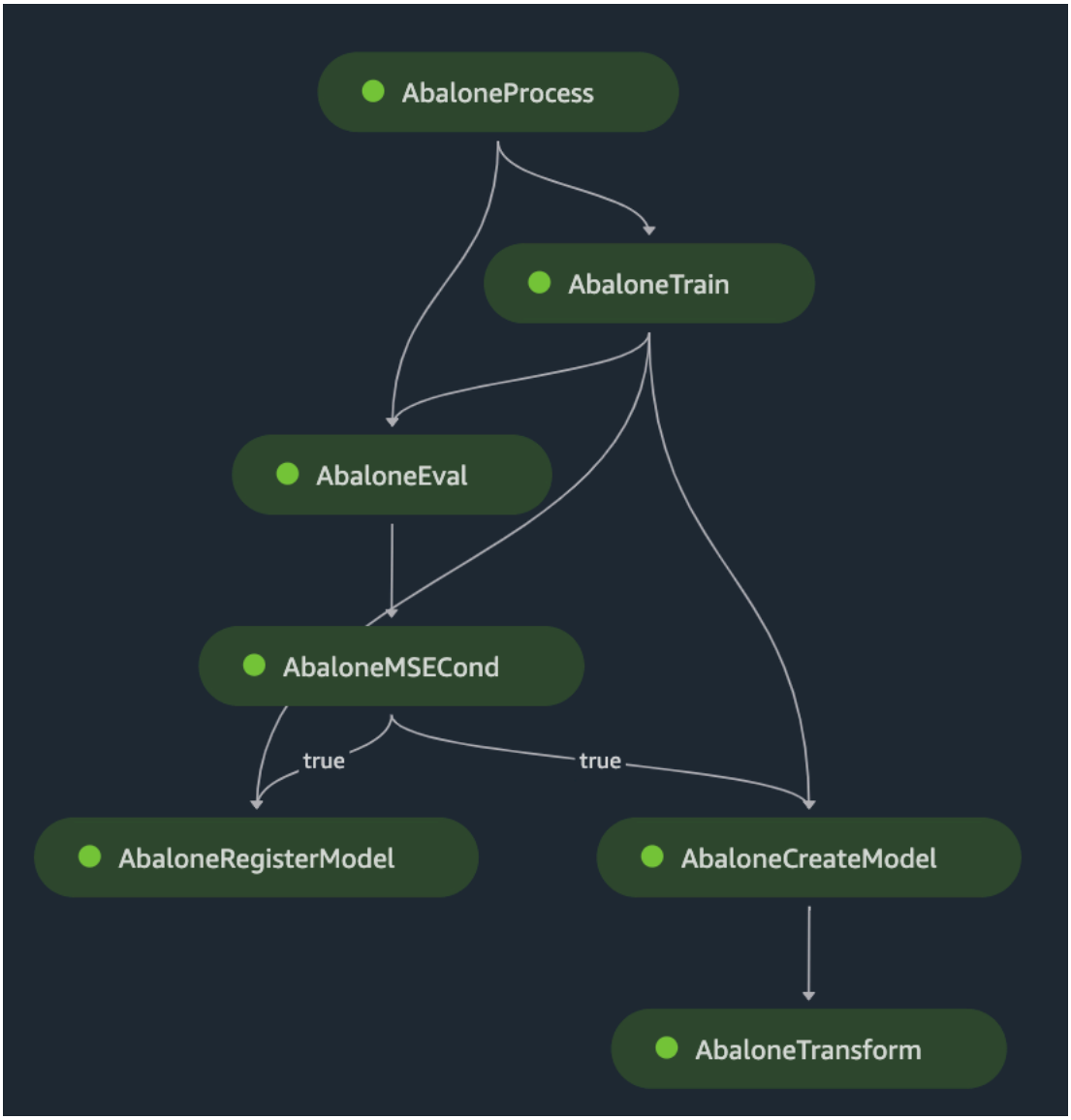
Untuk informasi selengkapnya tentang mengonfigurasi perilaku coba lagi untuk jenis langkah tertentu, lihat Amazon SageMaker Model Building Pipelines - Kebijakan Coba Lagi dalam dokumentasi Amazon Python SageMaker SDK.

Eksekusi selektif langkah-langkah pipa

Saat Anda menggunakan SageMaker Pipelines untuk membuat alur kerja dan mengatur langkah pelatihan ML Anda, Anda mungkin perlu melakukan beberapa fase eksperimen. Alih-alih menjalankan seluruh pipeline dari awal hingga akhir, Anda mungkin hanya ingin mengulangi langkah-langkah tertentu. SageMaker Pipelines mendukung pelaksanaan langkah-langkah pipeline secara selektif untuk membantu Anda mengoptimalkan pelatihan ML Anda. Eksekusi selektif berguna dalam skenario berikut:

- Anda ingin memulai ulang langkah tertentu dengan jenis instance yang diperbarui, hyperparameters, atau variabel lain sambil menjaga parameter dari langkah hulu.
- Pipeline Anda gagal dalam langkah perantara. Langkah-langkah sebelumnya dalam eksekusi, seperti persiapan data atau ekstraksi fitur, mahal untuk dijalankan kembali. Anda mungkin perlu memperkenalkan perbaikan dan menjalankan kembali langkah-langkah tertentu secara manual untuk menyelesaikan pipeline.

Dengan menggunakan eksekusi selektif, Anda dapat memilih untuk menjalankan subset langkah apa pun selama mereka terhubung dalam grafik asiklik terarah (DAG) dari pipeline Anda. DAG berikut menunjukkan contoh alur kerja pipeline:



Anda dapat memilih langkah-langkah `AbaloneTrain` dan `AbaloneEval` dalam eksekusi selektif, tetapi Anda tidak dapat memilih hanya `AbaloneTrain` dan `AbaloneMSECond` langkah-langkah untuk menjalankan eksekusi selektif karena langkah-langkah ini tidak terhubung di DAG. Untuk langkah-langkah yang tidak dipilih dalam alur kerja, eksekusi selektif menggunakan kembali output dari eksekusi pipeline referensi daripada menghitung ulang langkah-langkahnya. Juga, langkah-langkah yang tidak dipilih yang hilir dari langkah-langkah yang dipilih tidak berjalan dalam eksekusi selektif.

Jika Anda memilih untuk menjalankan subset langkah perantara di pipeline, langkah Anda mungkin memiliki dependensi pada langkah hulu. SageMaker membutuhkan eksekusi pipeline referensi untuk sumber daya dependensi ini. Misalnya, jika Anda memilih untuk menjalankan langkah-langkah `AbaloneTrain` dan`AbaloneEval`, Anda memerlukan agunan output untuk `AbaloneProcess`

langkah dari eksekusi pipeline referensi. Anda dapat memberikan ARN eksekusi referensi atau langsung SageMaker untuk menggunakan eksekusi pipeline terbaru, yang merupakan perilaku default. Jika Anda memiliki eksekusi referensi, Anda juga dapat membuat parameter runtime dari proses referensi Anda dan menyediakannya ke eksekutif selektif Anda dengan penggantian apa pun. Untuk detailnya, lihat Gunakan kembali nilai parameter runtime dari eksekusi referensi.

Secara detail, Anda menentukan konfigurasi untuk pipeline eksekusi selektif yang dijalankan. `SelectiveExecutionConfig` Jika Anda menentukan ARN untuk eksekusi pipeline referensi (dengan `source_pipeline_execution_arn` argumen), SageMaker gunakan dependensi langkah hulu dari eksekusi pipeline yang ditentukan. Jika Anda tidak menentukan ARN dan eksekusi pipeline terbaru ada, SageMaker gunakan eksekusi pipeline terbaru sebagai referensi secara default. Jika Anda tidak menentukan ARN dan tidak SageMaker ingin menggunakan eksekusi pipeline terbaru Anda, setel `reference_latest_execution` ke. `False` Eksekusi pipeline yang SageMaker pada akhirnya digunakan sebagai referensi, apakah yang terbaru atau yang ditentukan pengguna, harus dalam `Success` atau `Failed` status.

Tabel berikut merangkum bagaimana SageMaker memilih eksekusi referensi berdasarkan argumen Anda. `SelectiveExecutionConfig`

Nilai **`source_pipeline_execution_arn`** argumen	Nilai **`reference_latest_execution`** argumen	Eksekusi referensi yang digunakan
Sebuah pipa ARN	`True`atau tidak ditentukan	ARN pipa yang ditentukan
Sebuah pipa ARN	`False`	ARN pipa yang ditentukan
`null`atau tidak ditentukan	`True`atau tidak ditentukan	Eksekusi pipeline terbaru
`null`atau tidak ditentukan	`False`	Tidak ada—dalam hal ini, pilih langkah tanpa dependensi upstream

Untuk informasi selengkapnya tentang persyaratan konfigurasi eksekusi selektif, lihat sagemaker.workflow.selective_execution_config. SelectiveExecutionConfigdokumentasi.

Diskusi berikut mencakup contoh untuk kasus di mana Anda ingin menentukan eksekusi referensi pipeline, menggunakan eksekusi pipeline terbaru sebagai referensi, atau menjalankan eksekusi selektif tanpa eksekusi pipeline referensi.

Eksekusi selektif dengan referensi pipeline yang ditentukan pengguna

Contoh berikut menunjukkan penggunaan eksekusi selektif untuk menjalankan ulang `AbaloneTrain` dan `AbaloneEval` dalam pipeline yang sama dijalankan kembali menggunakan eksekusi pipeline referensi.

```
from sagemaker.workflow.selective_execution_config import SelectiveExecutionConfig

selective_execution_config = SelectiveExecutionConfig(
    source_pipeline_execution_arn="arn:aws:sagemaker:us-west-2:123123123123:pipeline/
abalone/execution/123ab12cd3ef",
    selected_steps=["AbaloneTrain", "AbaloneEval"]
)

selective_execution = pipeline.start(
    execution_display_name=f"Sample-Selective-Execution-1",
    parameters={"MaxDepth":6, "NumRound":60},
    selective_execution_config=selective_execution_config,
)
```

Eksekusi selektif dengan eksekusi pipeline terbaru sebagai referensi

Contoh berikut menunjukkan penggunaan eksekusi selektif untuk menjalankan ulang `AbaloneTrain` dan `AbaloneEval` dalam pipeline yang sama dijalankan kembali menggunakan eksekusi pipeline terbaru sebagai referensi. Karena SageMaker menggunakan eksekusi pipeline terbaru secara default, Anda dapat mengatur `reference_latest_execution` argumen secara opsional. `True`

```
# Prepare a new selective execution. Select only the first step in the pipeline without
 providing source_pipeline_execution_arn.
selective_execution_config = SelectiveExecutionConfig(
    selected_steps=["AbaloneTrain", "AbaloneEval"],
    # optional
    reference_latest_execution=True
)

# Start pipeline execution without source_pipeline_execution_arn
pipeline.start(
    execution_display_name=f"Sample-Selective-Execution-1",
```

```
    parameters={"MaxDepth":6, "NumRound":60},
    selective_execution_config=selective_execution_config,
)
```

Eksekusi selektif tanpa pipa referensi

Contoh berikut menunjukkan penggunaan eksekusi selektif untuk menjalankan ulang `AbaloneProcess` dan `AbaloneTrain` dalam pipeline yang sama dijalankan kembali tanpa memberikan referensi ARN dan melarang penggunaan pipeline terbaru yang dijalankan sebagai referensi. SageMaker memungkinkan konfigurasi ini karena subset langkah ini tidak memiliki dependensi upstream.

```
# Prepare a new selective execution. Select only the first step in the pipeline without
 providing source_pipeline_execution_arn.
selective_execution_config = SelectiveExecutionConfig(
    selected_steps=["AbaloneProcess", "AbaloneTrain"],
    reference_latest_execution=False
)

# Start pipeline execution without source_pipeline_execution_arn
pipeline.start(
    execution_display_name=f"Sample-Selective-Execution-1",
    parameters={"MaxDepth":6, "NumRound":60},
    selective_execution_config=selective_execution_config,
)
```

Gunakan kembali nilai parameter runtime dari eksekusi referensi

Anda dapat membuat parameter dari eksekusi pipeline referensi Anda menggunakan`build_parameters_from_execution`, dan memberikan hasilnya ke pipeline eksekusi selektif Anda. Anda dapat menggunakan parameter asli dari eksekusi referensi, atau menerapkan penggantian apa pun menggunakan argumen. `parameter_value_overrides`

Contoh berikut menunjukkan cara membuat parameter dari eksekusi referensi dan menerapkan override untuk `MseThreshold` parameter.

```
# Prepare a new selective execution.
selective_execution_config = SelectiveExecutionConfig(
    source_pipeline_execution_arn="arn:aws:sagemaker:us-west-2:123123123123:pipeline/
abalone/execution/123ab12cd3ef",
    selected_steps=["AbaloneTrain", "AbaloneEval", "AbaloneMSECond"],
)
```

```
# Define a new parameters list to test.
new_parameters_mse={
    "MseThreshold": 5,
}

# Build parameters from reference execution and override with new parameters to test.
new_parameters = pipeline.build_parameters_from_execution(
    pipeline_execution_arn="arn:aws:sagemaker:us-west-2:123123123123:pipeline/abalone/
execution/123ab12cd3ef",
    parameter_value_overrides=new_parameters_mse
)

# Start pipeline execution with new parameters.
execution = pipeline.start(
    selective_execution_config=selective_execution_config,
    parameters=new_parameters
)
```

Perhitungan dasar, deteksi drift, dan siklus hidup dengan serta ClarifyCheck langkah-langkah QualityCheck di Amazon Model Building Pipelines SageMaker

Topik berikut membahas bagaimana baseline dan versi model berkembang di Amazon SageMaker Model Building Pipelines saat menggunakan dan langkah-langkahnya. `ClarifyCheck` `QualityCheck`

Untuk `ClarifyCheck` langkah ini, baseline adalah file tunggal yang berada di properti langkah dengan akhiran. `constraints` Untuk `QualityCheck` langkah ini, baseline adalah kombinasi dari dua file yang berada di properti langkah: satu dengan akhiran `statistics` dan yang lainnya dengan akhiran. `constraints` Dalam topik berikut, kami membahas properti ini dengan awalan yang menjelaskan cara penggunaannya, memengaruhi perilaku dasar dan siklus hidup dalam dua langkah pipeline ini. Misalnya, `ClarifyCheck` langkah selalu menghitung dan menetapkan garis dasar baru di properti dan `QualityCheck` langkahnya melakukan hal yang sama di `CalculatedBaselineConstraints` properti dan. `CalculatedBaselineConstraints` `CalculatedBaselineStatistics`

Perhitungan dasar dan pendaftaran untuk ClarifyCheck dan langkah-langkah QualityCheck

Baik `QualityCheck` langkah `ClarifyCheck` dan langkah selalu menghitung baseline baru berdasarkan input langkah melalui pekerjaan pemrosesan yang mendasarinya. Garis dasar yang baru dihitung ini diakses melalui properti dengan awalan. `CalculatedBaseline` Anda dapat merekam properti ini sebagai paket model Anda di[Contoh](). `ModelMetrics` Paket model ini dapat

didaftarkan dengan 5 baseline yang berbeda. Anda dapat mendaftarkannya dengan satu untuk setiap jenis pemeriksaan: bias data, bias model, dan penjelasan model dari menjalankan `ClarifyCheck` langkah dan kualitas model, dan kualitas data dari menjalankan langkah. `QualityCheck` `register_new_baseline`Parameter menentukan nilai yang ditetapkan dalam properti dengan awalan `BaselineUsedForDriftCheck` setelah langkah berjalan.

Tabel kasus penggunaan potensial berikut menunjukkan perilaku berbeda yang dihasilkan dari parameter langkah yang dapat Anda tetapkan untuk `ClarifyCheck` dan `QualityCheck` langkah-langkahnya:

Kemungkinan kasus penggunaan yang dapat Anda pertimbangkan untuk memilih konfigurasi ini	**`skip_check`** / **`register_new_baseline`**	Apakah langkah melakukan pemeriksaan drift?	Nilai properti langkah **`CalculatedBaseline`**	Nilai properti langkah **`BaselineUsedForDriftCheck`**
Anda melakukan pelatihan ulang reguler dengan pemeriksaan diaktifkan untuk mendapatkan versi model baru, tetapi Anda ingin membawa baseline sebelumnya seperti `DriftCheckBaselines` dalam registri model untuk versi model baru Anda.	`False`/ `False`	Pemeriksaan drift berjalan terhadap baseline yang ada	Garis dasar baru dihitung dengan menjalankan langkah	Baseline dari model terbaru yang disetujui di Model Registry atau baseline yang disediakan sebagai parameter langkah

Kemungkinan kasus penggunaan yang dapat Anda pertimbangkan untuk memilih konfigurasi ini	**skip_check** / **register_new_baseline**	Apakah langkah melakukan pemeriksaan drift?	Nilai properti langkah **CalculatedBaseline**	Nilai properti langkah **BaselineUsedForDriftCheck**
Anda melakukan pelatihan ulang reguler dengan pemeriksaan diaktifkan untuk mendapatkan versi model baru, tetapi Anda ingin menyegarkan *DriftCheckBaselines* di registri model dengan garis dasar yang baru dihitung untuk versi model baru Anda.	`False`/ `True`	Pemeriksaan drift berjalan terhadap baseline yang ada	Garis dasar baru dihitung dengan menjalankan langkah	Garis dasar yang baru dihitung dengan menjalankan langkah (nilai properti) `CalculatedBaseline`

Kemungkinan kasus penggunaan yang dapat Anda pertimbangkan untuk memilih konfigurasi ini	**skip_check** / **register_new_baseline**	Apakah langkah melakukan pemeriksaan drift?	Nilai properti langkah **CalculatedBaseline**	Nilai properti langkah **BaselineUsedForDriftCheck**
Anda memulai pipeline untuk melatih ulang versi model baru karena ada pelanggaran yang terdeteksi oleh Amazon SageMaker Model Monitor pada titik akhir untuk jenis pemeriksaan tertentu, dan Anda ingin melewati jenis pemeriksaan ini terhadap baseline sebelumnya, tetapi bawa baseline sebelumnya seperti *DriftCheckBaselines* dalam registri model untuk	`True`/ `False`	Penghitung Galois	Garis dasar baru dihitung dengan menjalankan	Baseline dari model terbaru yang disetujui dalam registri model atau baseline yang disediakan sebagai parameter langkah

Kemungkinan kasus penggunaan yang dapat Anda pertimbangkan untuk memilih konfigurasi ini	**skip_check** / **register_new_baseline**	Apakah langkah melakukan pemeriksaan drift?	Nilai properti langkah **CalculatedBaseline**	Nilai properti langkah **BaselineUsedForDriftCheck**
versi model baru Anda.				

Kemungkinan kasus penggunaan yang dapat Anda pertimbangkan untuk memilih konfigurasi ini	**skip_check** / **register_new_baseline**	Apakah langkah melakukan pemeriksaan drift?	Nilai properti langkah **CalculatedBaseline**	Nilai properti langkah **BaselineUsedForDriftCheck**
Hal ini terjadi dalam kasus berikut: • Anda memulai proses awal pipeline, membangun versi model pertama Anda, dan menghasilkan baseline awal. • Anda memulai pipeline untuk melatih ulang versi model baru karena ada pelanggaran yang terdeteksi oleh Model Monitor pada titik akhir untuk jenis pemeriksaan tertentu. Jika Anda ingin melewati pemeriksa	`True`/ `True`	Penghitung Galois	Garis dasar baru dihitung dengan menjalankan langkah	Garis dasar yang baru dihitung dengan menjalankan langkah (nilai properti) `CalculatedBaseline`

Kemungkinan kasus penggunaan yang dapat Anda pertimbangkan untuk memilih konfigurasi ini	**skip_check / register_new_baseline**	Apakah langkah melakukan pemeriksaan drift?	Nilai properti langkah **CalculatedBaseline**	Nilai properti langkah **BaselineUsedForDriftCheck**
an terhadap baseline sebelumnya dan menyegarkan *DriftCheckBaselines* dengan baseline yang baru dihitung di registri model secara langsung.				

> **Note**
>
> Jika Anda menggunakan notasi ilmiah dalam kendala Anda, Anda perlu mengkonversi ke float. Untuk contoh skrip preprocessing tentang cara melakukannya, lihat Membuat Garis Dasar Kualitas Model.

Saat Anda mendaftarkan modelContoh, Anda dapat mendaftarkan `BaselineUsedForDriftCheck` properti sebagai`DriftCheckBaselines`. File dasar ini kemudian dapat digunakan oleh Model Monitor untuk pemeriksaan kualitas model dan data. Selain itu, baseline ini juga dapat digunakan dalam `QualityCheck` langkah ClarifyCheckStep dan untuk membandingkan model yang baru dilatih dengan model yang ada yang terdaftar dalam registri model untuk proses pipeline future.

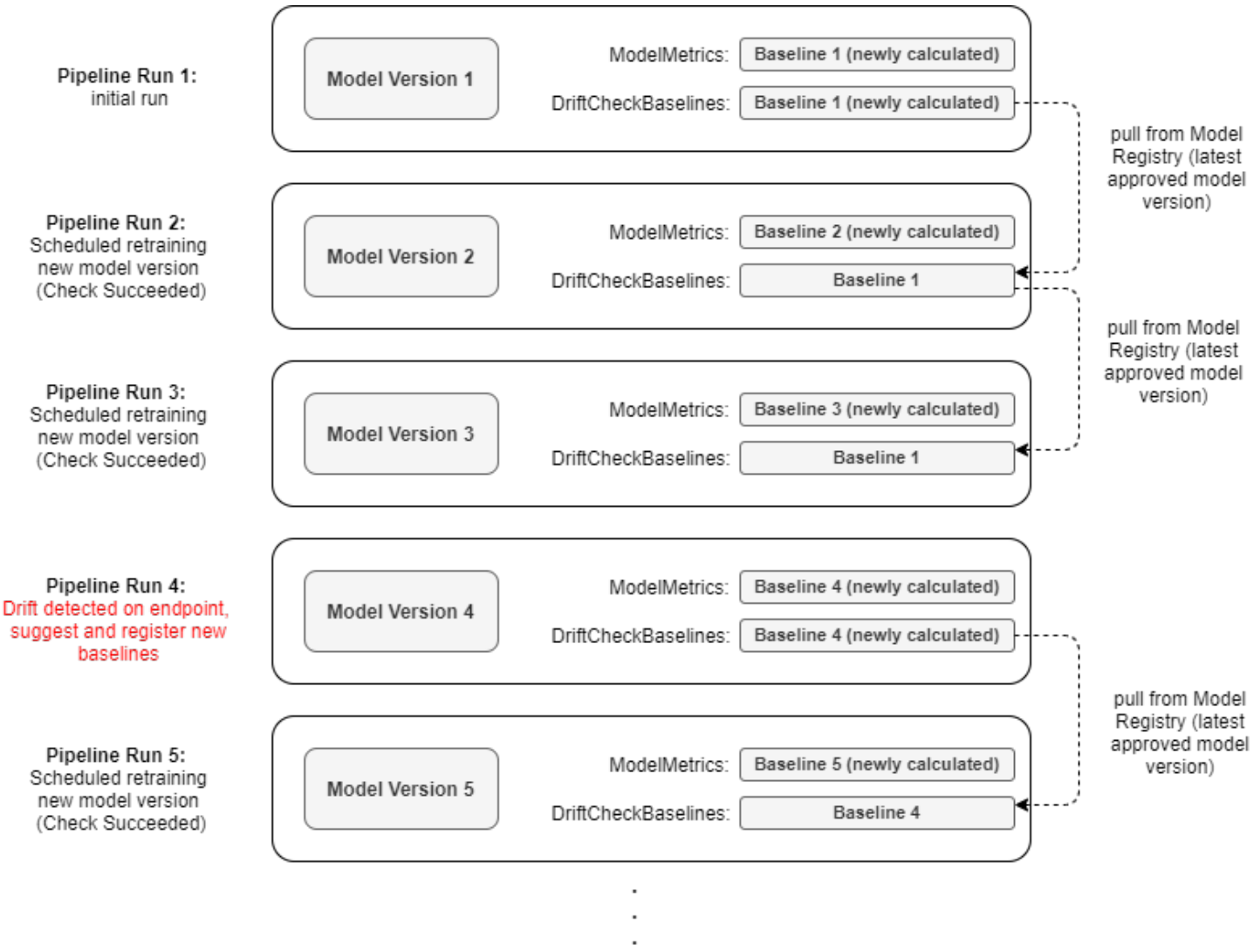
Deteksi Drift terhadap Baseline Sebelumnya di Pipa SageMaker

Dalam hal `QualityCheck` langkah, ketika Anda memulai pipeline untuk pelatihan ulang reguler untuk mendapatkan versi model baru, Anda mungkin tidak ingin menjalankan langkah pelatihan jika kualitas data dan bias data ada Skema untuk Pelanggaran (file constraint_violations.json) pada garis dasar versi model yang disetujui sebelumnya. Anda juga mungkin tidak ingin mendaftarkan versi model yang baru dilatih jika kualitas model, bias model, atau penjelasan model melanggar baseline terdaftar dari versi model yang disetujui sebelumnya saat menjalankan langkah. `ClarifyCheck` Dalam kasus ini, Anda dapat mengaktifkan pemeriksaan yang Anda inginkan dengan menyetel `skip_check` properti dari langkah pemeriksaan terkait yang disetel`False`, sehingga `QualityCheck` langkah `ClarifyCheck` dan gagal jika pelanggaran terdeteksi terhadap garis dasar sebelumnya. Proses pipa kemudian tidak dilanjutkan sehingga model yang melayang dari baseline tidak terdaftar. `ClarifyCheck`dan `QualityCheck` langkah-langkah bisa mendapatkan `DriftCheckBaselines` versi model terbaru yang disetujui dari grup paket model tertentu yang dapat dibandingkan. Garis dasar sebelumnya juga dapat diberikan langsung melalui `supplied_baseline_constraints` (selain `supplied_baseline_statistics` jika itu adalah `QualityCheck` langkah) dan selalu diprioritaskan di atas garis dasar apa pun yang ditarik dari grup paket model.

Siklus hidup dan evolusi versi dasar dan model dengan Pipelines SageMaker

Dengan menetapkan `register_new_baseline` `QualityCheck` langkah Anda `ClarifyCheck` dan ke`False`, baseline Anda sebelumnya dapat diakses melalui awalan properti langkah. `BaselineUsedForDriftCheck` Anda kemudian dapat mendaftarkan baseline ini sebagai `DriftCheckBaselines` dalam versi model baru ketika Anda mendaftarkan model dengan. Contoh Setelah Anda menyetujui versi model baru ini di registri model, versi model ini akan tersedia untuk `ClarifyCheck` dan `QualityCheck` langkah-langkah dalam proses pipeline berikutnya. `DriftCheckBaseline` Jika Anda ingin me-refresh baseline dari jenis pemeriksaan tertentu untuk versi model future, Anda dapat mengatur `register_new_baseline` `True` agar properti dengan awalan `BaselineUsedForDriftCheck` menjadi baseline yang baru dihitung. Dengan cara ini, Anda dapat mempertahankan baseline pilihan Anda untuk model yang dilatih di masa mendatang, atau menyegarkan baseline untuk pemeriksaan drift bila diperlukan, mengelola evolusi dasar dan siklus hidup Anda di seluruh iterasi pelatihan model Anda.

Diagram berikut menggambarkan model-version-centric pandangan evolusi dasar dan siklus hidup.



Jadwalkan Alur Berjalan

Anda dapat menjadwalkan eksekusi Amazon SageMaker Model Building Pipelines menggunakan Amazon. EventBridge Amazon SageMaker Model Building Pipelines didukung sebagai target di Amazon EventBridge. Ini memungkinkan Anda untuk memulai eksekusi pipa bangunan model Anda berdasarkan peristiwa apa pun di bus acara Anda. Dengan EventBridge, Anda dapat mengotomatiskan eksekusi pipeline dan merespons secara otomatis peristiwa seperti pekerjaan pelatihan atau perubahan status titik akhir. Acara mencakup file baru yang diunggah ke bucket Amazon S3, perubahan status titik akhir Amazon karena drift, dan SageMaker topik Amazon Simple Notification Service (SNS).

Tindakan SageMaker Pipelines berikut dapat dimulai secara otomatis:

- `StartPipelineExecution`

Untuk informasi selengkapnya tentang penjadwalan SageMaker pekerjaan, lihat Mengotomatisasi dengan SageMaker Amazon. EventBridge

Topik

- Jadwalkan Pipeline dengan Amazon EventBridge
- Jadwalkan pipeline dengan SageMaker Python SDK

Jadwalkan Pipeline dengan Amazon EventBridge

Untuk memulai eksekusi pipeline dengan Amazon CloudWatch Events, Anda harus membuat EventBridge aturan. Ketika Anda membuat aturan untuk peristiwa, Anda menentukan tindakan target untuk mengambil ketika EventBridge menerima peristiwa yang cocok dengan aturan. Ketika peristiwa cocok dengan aturan, EventBridge kirim peristiwa ke target yang ditentukan dan memulai tindakan yang didefinisikan dalam aturan.

Tutorial berikut menunjukkan cara menjadwalkan eksekusi pipeline dengan EventBridge menggunakan EventBridge konsol atau fileAWS CLI.

Prasyarat

- Peran yang EventBridge dapat diasumsikan dengan `SageMaker::StartPipelineExecution` izin. Peran ini dapat dibuat secara otomatis jika Anda membuat aturan dari EventBridge konsol; jika tidak, Anda perlu membuat peran ini sendiri. Untuk informasi tentang membuat SageMaker peran, lihat SageMaker Peran.
- SageMaker Pipeline Amazon untuk menjadwalkan. Untuk membuat SageMaker Pipeline Amazon, lihat Mendefinisikan Pipeline.

Buat EventBridge aturan menggunakan EventBridge konsol

Prosedur berikut menunjukkan cara membuat EventBridge aturan menggunakan EventBridge konsol.

1. Navigasikan ke konsol EventBridge tersebut.
2. Pilih Aturan di sisi kiri.
3. Pilih `Create Rule`.
4. Masukkan nama dan deskripsi untuk aturan Anda.
5. Pilih bagaimana Anda ingin memulai aturan ini. Anda memiliki pilihan berikut untuk aturan Anda:

- Pola acara: Aturan Anda dimulai ketika peristiwa yang cocok dengan pola terjadi. Anda dapat memilih pola standar yang cocok dengan jenis acara tertentu, atau Anda dapat membuat pola kustom. Jika Anda memilih pola yang telah ditentukan, Anda dapat mengedit pola untuk menyesuaikannya. Untuk informasi selengkapnya tentang pola Peristiwa, lihat Pola CloudWatch Peristiwa di Acara.
- Jadwal: Aturan Anda dimulai secara teratur pada jadwal yang ditentukan. Anda dapat menggunakan jadwal tarif tetap yang dimulai secara teratur untuk jumlah menit, jam, atau minggu tertentu. Anda juga dapat menggunakan ekspresi cron untuk membuat jadwal yang lebih halus, seperti “Senin pertama setiap bulan pukul 8 pagi.” Jadwal tidak didukung pada bus acara khusus atau mitra.

6. Pilih bus Acara yang Anda inginkan.
7. Pilih target yang akan dipanggil saat acara cocok dengan pola acara Anda atau saat jadwal dimulai. Anda dapat menambahkan hingga 5 target per aturan. Pilih `SageMaker Pipeline` dalam daftar tarik-turun target.
8. Pilih alur yang ingin Anda mulai dari daftar dropdown alur.
9. Tambahkan parameter untuk diteruskan ke eksekusi pipeline Anda menggunakan nama dan pasangan nilai. Nilai parameter dapat bersifat statis atau dinamis. Untuk informasi selengkapnya tentang parameter Amazon SageMaker Pipeline, lihat AWS::Events::Rule SagemakerPipelineParameters.
   - Nilai statis diteruskan ke eksekusi pipeline setiap kali pipeline dimulai. Misalnya, jika `{"Name": "Instance_type", "Value": "ml.4xlarge"}` ditentukan dalam daftar parameter, maka itu dilewatkan sebagai parameter di `StartPipelineExecutionRequest` setiap kali EventBridge memulai pipeline.
   - Nilai dinamis ditentukan menggunakan jalur JSON. EventBridge mem-parsing nilai dari muatan peristiwa, lalu meneruskannya ke eksekusi pipeline. Sebagai contoh: *`$.detail.param.value`*
10. Pilih peran yang akan digunakan untuk aturan ini. Anda dapat menggunakan peran yang sudah ada atau membuat peran baru.
11. (Opsional) Tambahkan tag.
12. Pilih `Create` untuk menyelesaikan aturan Anda.

Aturan Anda sekarang berlaku dan siap untuk memulai eksekusi pipeline Anda.

Buat EventBridge aturan menggunakan AWS CLI

Prosedur berikut menunjukkan cara membuat EventBridge aturan menggunakanAWS CLI.

1. Buat aturan yang akan dimulai. Saat membuat EventBridge aturan menggunakanAWS CLI, Anda memiliki dua opsi untuk bagaimana aturan Anda dimulai, pola acara dan jadwal.

   - Pola acara: Aturan Anda dimulai ketika peristiwa yang cocok dengan pola terjadi. Anda dapat memilih pola standar yang cocok dengan jenis acara tertentu, atau Anda dapat membuat pola kustom. Jika Anda memilih pola yang telah ditentukan, Anda dapat mengedit pola untuk menyesuaikannya. Anda dapat membuat aturan dengan pola acara menggunakan perintah berikut:

     ```
     aws events put-rule --name <RULE_NAME> ----event-pattern <YOUR_EVENT_PATTERN>
      --description <RULE_DESCRIPTION> --role-arn <ROLE_TO_EXECUTE_PIPELINE> --
     tags <TAGS>
     ```

   - Jadwal: Aturan Anda dimulai secara teratur pada jadwal yang ditentukan. Anda dapat menggunakan jadwal tarif tetap yang dimulai secara teratur untuk jumlah menit, jam, atau minggu tertentu. Anda juga dapat menggunakan ekspresi cron untuk membuat jadwal yang lebih halus, seperti “Senin pertama setiap bulan pukul 8 pagi.” Jadwal tidak didukung pada bus acara khusus atau mitra. Anda dapat membuat aturan dengan jadwal menggunakan perintah berikut:

     ```
     aws events put-rule --name <RULE_NAME> --schedule-
     expression <YOUR_CRON_EXPRESSION> --description <RULE_DESCRIPTION> --role-
     arn <ROLE_TO_EXECUTE_PIPELINE> --tags <TAGS>
     ```

2. Tambahkan target untuk dipanggil saat acara cocok dengan pola acara Anda atau saat jadwal dimulai. Anda dapat menambahkan hingga 5 target per aturan. Untuk setiap target, Anda harus menentukan:

   - Outputnya: Sumber daya ARN dari pipa Anda.
   - Peran ARN: ARN peran EventBridge harus diambil untuk menjalankan pipa.
   - Parameter: Parameter SageMaker pipa Amazon untuk dilewati.

3. Jalankan perintah berikut untuk meneruskan SageMaker pipeline Amazon sebagai target ke aturan Anda menggunakan put-target:

```
aws events put-targets --rule <RULE_NAME> --event-bus-name <EVENT_BUS_NAME>
 --targets "[{\"Id\": <ID>, \"Arn\": <RESOURCE_ARN>, \"RoleArn\": <ROLE_ARN>,
 \"SageMakerPipelineParameter\": { \"SageMakerParameterList\": [{\"Name\": <NAME>,
 \"Value\": <VALUE>}]} }]"]
```

Jadwalkan pipeline dengan SageMaker Python SDK

Bagian berikut menunjukkan cara mengatur izin untuk mengakses EventBridge sumber daya dan membuat jadwal pipeline Anda menggunakan SageMaker Python SDK.

Izin yang diperlukan

Anda harus memiliki izin yang diperlukan untuk menggunakan penjadwal alur. Selesaikan langkah-langkah berikut untuk menyiapkan izin Anda:

1. Lampirkan kebijakan hak istimewa minimum berikut ke peran IAM yang digunakan untuk membuat pemicu pipeline, atau gunakan kebijakan AWS terkelola. `AmazonEventBridgeSchedulerFullAccess`

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:ListSchedules",
                "scheduler:GetSchedule",
                "scheduler:CreateSchedule",
                "scheduler:UpdateSchedule",
                "scheduler:DeleteSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

2. Membangun hubungan kepercayaan EventBridge dengan menambahkan prinsip layanan `scheduler.amazonaws.com` ke kebijakan kepercayaan peran ini. Pastikan Anda melampirkan kebijakan kepercayaan berikut ke peran eksekusi jika Anda meluncurkan buku catatan di SageMaker Studio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "scheduler.amazonaws.com",
                    "sagemaker.amazonaws.com"
                ]
            },
        "Action": "sts:AssumeRole"
        }
    ]
}
```

Membuat jadwal alur

Dengan menggunakan `PipelineSchedule` konstruktor, Anda dapat menjadwalkan pipeline untuk dijalankan sekali atau pada interval yang telah ditentukan. Jadwal pipa harus dari jenis`at`,`rate`, atau`cron`. Kumpulan jenis penjadwalan ini merupakan perpanjangan dari opsi EventBridge penjadwalan. Untuk informasi selengkapnya tentang cara menggunakan `PipelineSchedule` kelas, lihat sagemaker.workflow.trigger. PipelineSchedule. Contoh berikut menunjukkan cara membuat setiap jenis penjadwalan dengan. `PipelineSchedule`

```
from sagemaker.workflow.triggers import PipelineSchedule
```

```
# schedules a pipeline run for 12/13/2023 at time 10:15:20 UTC
my_datetime_schedule = PipelineSchedule(
    name="<schedule-name>",
    at=datetime(2023, 12, 13, 10, 15, 20)
)

# schedules a pipeline run every 5 minutes
my_rate_schedule = PipelineSchedule(
    name="<schedule-name>",
    rate=(5, "minutes")
)

# schedules a pipeline run at 10:15am UTC on the last Friday of each month during the
 years 2022 to 2023
my_cron_schedule = PipelineSchedule(
    name="<schedule-name>",
    cron="15 10 ? * 6L 2022-2023"
)
```

> **Note**
>
> Jika Anda membuat jadwal satu kali dan perlu mengakses waktu saat ini, gunakan `datetime.utcnow()` sebagai gantinya. `datetime.now()` Yang terakhir tidak menyimpan konteks zona saat ini dan menghasilkan waktu yang salah EventBridge.

Pasang pelatuk ke pipa Anda

Untuk melampirkan `PipelineSchedule` ke pipeline, panggil `put_triggers` panggilan pada objek pipeline yang Anda buat dengan daftar pemicu. Jika Anda mendapatkan respons ARN, Anda berhasil membuat jadwal di akun Anda dan EventBridge mulai memanggil pipeline target pada waktu atau tarif yang ditentukan. Anda harus menentukan peran dengan izin yang benar untuk melampirkan pemicu ke pipeline induk. Jika Anda tidak menyediakannya, SageMaker Pipelines mengambil peran default yang digunakan untuk membuat pipeline dari file konfigurasi.

Contoh berikut menunjukkan cara melampirkan jadwal ke alur.

```
scheduled_pipeline = Pipeline(
    name="<pipeline-name>",
    steps=[...],
```

```
    sagemaker_session=<sagemaker-session>,
)
custom_schedule = PipelineSchedule(
    name="<schedule-name>",
    at=datetime(year=2023, month=12, date=25, hour=10, minute=30, second=30)
)
scheduled_pipeline.put_triggers(triggers=[custom_schedule], role_arn=<role>)
```

Jelaskan pemicu saat ini

Untuk mengambil informasi tentang pemicu pipeline yang dibuat, Anda dapat memanggil `describe_trigger()` API dengan nama pemicu. Perintah ini mengembalikan detail tentang ekspresi jadwal yang dibuat seperti waktu mulai, status diaktifkan, dan informasi berguna lainnya. Cuplikan berikut menunjukkan contoh pemanggilan:

```
scheduled_pipeline.describe_trigger(name="<schedule-name>")
```

Sumber daya pemicu pembersihan

Sebelum menghapus pipeline, bersihkan pemicu yang ada untuk menghindari kebocoran sumber daya di akun Anda. Anda harus menghapus pemicu sebelum menghancurkan pipeline induk. Anda dapat menghapus pemicu dengan meneruskan daftar nama pemicu ke `delete_triggers` API. Cuplikan berikut menunjukkan cara menghapus pemicu.

```
pipeline.delete_triggers(trigger_names=["<schedule-name>"])
```

> **Note**
>
> Waspadai keterbatasan berikut saat Anda menghapus pemicu:
>
> - Opsi untuk menghapus pemicu dengan menentukan nama pemicu hanya tersedia di Python SageMaker SDK. Menghapus pipeline di CLI atau panggilan API tidak menghapus `DeletePipeline` pemicu Anda. Akibatnya, pemicu menjadi yatim piatu dan SageMaker mencoba untuk memulai lari untuk pipa yang tidak ada.
> - Juga, jika Anda menggunakan sesi notebook lain atau sudah menghapus target pipeline, bersihkan jadwal yatim melalui CLI penjadwal atau konsol. EventBridge

Integrasi SageMaker Eksperimen Amazon

Amazon SageMaker Model Building Pipelines terintegrasi erat dengan Amazon SageMaker Experiments. Secara default, saat SageMaker Pipelines membuat dan mengeksekusi pipeline, entitas SageMaker Eksperimen berikut dibuat jika tidak ada:

- Eksperimen untuk pipa
- Grup run untuk setiap eksekusi pipeline
- Run yang ditambahkan ke grup run untuk setiap SageMaker pekerjaan yang dibuat dalam langkah eksekusi pipeline

Anda dapat membandingkan metrik seperti akurasi pelatihan model di beberapa eksekusi pipeline seperti halnya Anda dapat membandingkan metrik tersebut di beberapa grup lari dari eksperimen pelatihan SageMaker model.

Contoh berikut menunjukkan parameter yang relevan dari kelas Pipeline di Amazon SageMaker Python SDK.

```
Pipeline(
    name="MyPipeline",
    parameters=[...],
    pipeline_experiment_config=PipelineExperimentConfig(
      ExecutionVariables.PIPELINE_NAME,
      ExecutionVariables.PIPELINE_EXECUTION_ID
    ),
    steps=[...]
)
```

Jika Anda tidak ingin grup eksperimen dan jalankan dibuat untuk pipeline, setel `pipeline_experiment_config` ke`None`.

> **Note**
>
> Integrasi eksperimen diperkenalkan di Amazon SageMaker Python SDK v2.41.0.

Aturan penamaan berikut berlaku berdasarkan apa yang Anda tentukan untuk `ExperimentName` dan `TrialName` parameter`pipeline_experiment_config`:

- Jika Anda tidak menentukan`ExperimentName`, pipeline `name` digunakan untuk nama percobaan.

  Jika Anda menentukan`ExperimentName`, itu digunakan untuk nama percobaan. Jika ada eksperimen dengan nama itu, grup run yang dibuat pipeline akan ditambahkan ke eksperimen yang ada. Jika eksperimen dengan nama itu tidak ada, eksperimen baru dibuat.
- Jika Anda tidak menentukan`TrialName`, ID eksekusi pipeline digunakan untuk nama grup run.

  Jika Anda menentukan`TrialName`, itu digunakan untuk nama grup run. Jika grup run dengan nama itu ada, proses yang dibuat pipeline akan ditambahkan ke grup run yang ada. Jika grup run dengan nama itu tidak ada, grup run baru akan dibuat.

> **Note**
>
> Entitas eksperimen tidak dihapus saat pipeline yang membuat entitas dihapus. Anda dapat menggunakan API SageMaker Eksperimen untuk menghapus entitas. Untuk informasi selengkapnya, lihat Bersihkan Sumber Daya SageMaker Eksperimen Amazon.

Untuk informasi tentang cara melihat entitas SageMaker Eksperimen yang terkait dengan pipeline, lihatLihat Entitas Eksperimen yang Dibuat oleh SageMaker Pipelines. Untuk informasi lebih lanjut tentang SageMaker Eksperimen, lihatKelola Machine Learning dengan Amazon SageMaker Experiments.

Bagian berikut menunjukkan contoh aturan sebelumnya dan bagaimana mereka direpresentasikan dalam file definisi pipeline. Untuk informasi selengkapnya tentang file definisi pipeline, lihatSageMaker Ikhtisar Pipelines.

Topik

- Peran
- Nonaktifkan Integrasi Eksperimen
- Tentukan Nama Eksperimen Kustom
- Tentukan Nama Grup Jalankan Kustom

Peran

Buat pipeline

`pipeline_experiment_config`Itu dihilangkan. `ExperimentName`default ke pipa. `name` `TrialName`default ke ID eksekusi.

```
pipeline_name = f"MyPipeline"
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[...],
    steps=[step_train]
)
```

File definisi saluran pipa

```
{
  "Version": "2020-12-01",
  "Parameters": [
    {
      "Name": "InputDataSource"
    },
    {
      "Name": "InstanceCount",
      "Type": "Integer",
      "DefaultValue": 1
    }
  ],
  "PipelineExperimentConfig": {
    "ExperimentName": {"Get": "Execution.PipelineName"},
    "TrialName": {"Get": "Execution.PipelineExecutionId"}
  },
  "Steps": [...]
}
```

Nonaktifkan Integrasi Eksperimen

Buat pipeline

`pipeline_experiment_config` ditetapkan ke `None`.

```
pipeline_name = f"MyPipeline"
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[...],
    pipeline_experiment_config=None,
    steps=[step_train]
```

```
)
```

File definisi saluran pipa

Ini sama dengan contoh default sebelumnya, tanpa. `PipelineExperimentConfig`

Tentukan Nama Eksperimen Kustom

Nama eksperimen khusus digunakan. Nama grup run diatur ke ID eksekusi, seperti perilaku default.

Buat pipeline

```
pipeline_name = f"MyPipeline"
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[...],
    pipeline_experiment_config=PipelineExperimentConfig(
      "CustomExperimentName",
      ExecutionVariables.PIPELINE_EXECUTION_ID
    ),
    steps=[step_train]
)
```

File definisi saluran pipa

```
{
  ...,
  "PipelineExperimentConfig": {
    "ExperimentName": "CustomExperimentName",
    "TrialName": {"Get": "Execution.PipelineExecutionId"}
  },
  "Steps": [...]
}
```

Tentukan Nama Grup Jalankan Kustom

Nama grup run kustom digunakan dan ditambahkan dengan ID eksekusi. Nama eksperimen disetel ke nama pipeline, seperti perilaku default.

Buat pipeline

```
pipeline_name = f"MyPipeline"
pipeline = Pipeline(
```

```
    name=pipeline_name,
    parameters=[...],
    pipeline_experiment_config=PipelineExperimentConfig(
      ExecutionVariables.PIPELINE_NAME,
      Join(on="-", values=["CustomTrialName",
 ExecutionVariables.PIPELINE_EXECUTION_ID])
    ),
    steps=[step_train]
)
```

File definisi saluran pipa

```
{
  ...,
  "PipelineExperimentConfig": {
    "ExperimentName": {"Get": "Execution.PipelineName"},
    "TrialName": {
      "On": "-",
      "Values": [
         "CustomTrialName",
         {"Get": "Execution.PipelineExecutionId"}
       ]
    }
  },
  "Steps": [...]
}
```

Mode

SageMaker Mode lokal pipeline adalah cara mudah untuk menguji skrip pelatihan, pemrosesan, dan inferensi Anda, serta kompatibilitas runtime parameter pipeline sebelum Anda menjalankan pipeline pada layanan terkelola. SageMaker Dengan menggunakan mode lokal, Anda dapat menguji SageMaker pipeline secara lokal menggunakan kumpulan data yang lebih kecil. Hal ini memungkinkan debugging cepat dan mudah kesalahan dalam skrip pengguna dan definisi pipeline itu sendiri tanpa menimbulkan biaya menggunakan layanan terkelola.

Pipelines mode lokal memanfaatkan SageMaker pekerjaan mode lokal di bawah tenda. Ini adalah fitur dalam SDK SageMaker Python yang memungkinkan Anda menjalankan gambar SageMaker bawaan atau kustom secara lokal menggunakan wadah Docker. Pipelines mode lokal dibangun di atas modus SageMaker pekerjaan lokal. Oleh karena itu, Anda dapat mengharapkan untuk melihat hasil yang sama seolah-olah Anda menjalankan pekerjaan itu secara terpisah. Misalnya, mode lokal

masih menggunakan Amazon S3 untuk mengunggah artefak model dan output pemrosesan. Jika Anda ingin data yang dihasilkan oleh pekerjaan lokal berada di disk lokal, Anda dapat menggunakan pengaturan yang disebutkan dalam Mode Lokal.

Mode lokal pipa saat ini mendukung jenis langkah berikut:

- Langkah Pelatihan
- Langkah Pemrosesan
- Pembuatan
- Model Step (dengan argumen Create Model saja)
- Langkah Kondisi
- Langkah Gagal

Berbeda dengan layanan Pipelines terkelola yang memungkinkan beberapa langkah untuk dijalankan secara paralel menggunakan Parallelism Configuration, pelaksana pipeline lokal menjalankan langkah-langkah secara berurutan. Oleh karena itu, kinerja eksekusi keseluruhan dari pipa lokal mungkin lebih buruk daripada yang berjalan di cloud - ini sebagian besar tergantung pada ukuran dataset, algoritma, serta kekuatan komputer lokal Anda. Perhatikan juga bahwa Pipelines berjalan dalam mode lokal tidak direkam dalam SageMaker Eksperimen.

> **Note**
>
> Pipelines mode lokal tidak kompatibel dengan SageMaker algoritma seperti XGBoost. Jika Anda ingin menggunakan algoritma ini, Anda harus menggunakannya dalam mode skrip.

Untuk menjalankan pipa secara lokal, `sagemaker_session` bidang yang terkait dengan langkah-langkah pipa dan pipa itu sendiri harus bertipe`LocalPipelineSession`. Contoh berikut menunjukkan cara menentukan SageMaker alur untuk mengeksekusi secara lokal.

```
from sagemaker.workflow.pipeline_context import LocalPipelineSession
from sagemaker.pytorch import PyTorch
from sagemaker.workflow.steps import TrainingStep
from sagemaker.workflow.pipeline import Pipeline

local_pipeline_session = LocalPipelineSession()
```

```
pytorch_estimator = PyTorch(
    sagemaker_session=local_pipeline_session,
    role=sagemaker.get_execution_role(),
    instance_type="ml.c5.xlarge",
    instance_count=1,
    framework_version="1.8.0",
    py_version="py36",
    entry_point="./entry_point.py",
)

step = TrainingStep(
    name="MyTrainingStep",
    step_args=pytorch_estimator.fit(
        inputs=TrainingInput(s3_data="s3://my-bucket/my-data/train"),
    )
)

pipeline = Pipeline(
    name="MyPipeline",
    steps=[step],
    sagemaker_session=local_pipeline_session
)

pipeline.create(
    role_arn=sagemaker.get_execution_role(),
    description="local pipeline example"
)

// pipeline will execute locally
execution = pipeline.start()

steps = execution.list_steps()

training_job_name = steps['PipelineExecutionSteps'][0]['Metadata']['TrainingJob']
['Arn']

step_outputs = pipeline_session.sagemaker_client.describe_training_job(TrainingJobName
 = training_job_name)
```

Setelah Anda siap untuk menjalankan pipeline pada layanan SageMaker Pipelines terkelola, Anda dapat melakukannya dengan mengganti `LocalPipelineSession` cuplikan kode sebelumnya dengan `PipelineSession` (seperti yang ditunjukkan pada contoh kode berikut) dan menjalankan ulang kode.

```
from sagemaker.workflow.pipeline_context import PipelineSession

pipeline_session = PipelineSession()
```

Memecahkan Masalah Pipa Pembuatan SageMaker Model Amazon

Saat menggunakan Amazon SageMaker Model Building Pipelines, Anda mungkin mengalami masalah karena berbagai alasan. Topik ini memberikan informasi tentang kesalahan umum dan cara mengatasinya.

Masalah Definisi Pipeline

Definisi pipeline Anda mungkin tidak diformat dengan benar. Hal ini dapat mengakibatkan eksekusi Anda gagal atau pekerjaan Anda menjadi tidak akurat. Kesalahan ini dapat ditangkap ketika pipeline dibuat atau ketika eksekusi terjadi. Jika definisi Anda tidak memvalidasi, SageMaker Pipelines mengembalikan pesan kesalahan yang mengidentifikasi karakter di mana file JSON salah format. Untuk memperbaiki masalah ini, tinjau langkah-langkah yang dibuat menggunakan SageMaker Python SDK untuk akurasi.

Anda hanya dapat menyertakan langkah-langkah dalam definisi pipeline sekali. Karena itu, langkah-langkah tidak dapat ada sebagai bagian dari langkah kondisi dan pipa dalam pipa yang sama.

Memeriksa Log Pipa

Anda dapat melihat status langkah Anda menggunakan perintah berikut:

```
execution.list_steps()
```

Setiap langkah mencakup informasi berikut:

- ARN entitas yang diluncurkan oleh pipeline, seperti pekerjaan ARN, model SageMaker ARN, atau paket model ARN.
- Alasan kegagalan mencakup penjelasan singkat tentang kegagalan langkah.
- Jika langkah adalah langkah kondisi, itu termasuk apakah kondisi dievaluasi menjadi benar atau salah.
- Jika eksekusi menggunakan kembali eksekusi pekerjaan sebelumnya, `CacheHit` daftar eksekusi sumber.

Anda juga dapat melihat pesan kesalahan dan log di antarmuka Amazon SageMaker Studio. Untuk informasi tentang cara melihat log di Studio, lihatMelihat Eksekusi Alur.

Izin Hilang

Izin yang benar diperlukan untuk peran yang membuat eksekusi pipeline, dan langkah-langkah yang membuat setiap pekerjaan dalam eksekusi pipeline Anda. Tanpa izin ini, Anda mungkin tidak dapat mengirimkan eksekusi pipeline atau menjalankan SageMaker pekerjaan seperti yang diharapkan. Untuk memastikan bahwa izin Anda diatur dengan benar, lihatManajemen Akses IAM.

Kesalahan Eksekusi Job

Anda mungkin mengalami masalah saat menjalankan langkah-langkah Anda karena masalah dalam skrip yang menentukan fungsionalitas pekerjaan Anda SageMaker . Setiap pekerjaan memiliki satu set CloudWatch log. Untuk melihat log ini dari Studio, lihatMelihat Eksekusi Alur. Untuk informasi tentang menggunakan CloudWatch log dengan SageMaker, lihatLog SageMaker Acara Amazon dengan Amazon CloudWatch.

Kesalahan File Properti

Anda mungkin mengalami masalah saat salah menerapkan file properti dengan pipeline Anda. Untuk memastikan bahwa implementasi file properti Anda berfungsi seperti yang diharapkan, lihatLulus Data Antar Langkah.

Membuat dan Mengelola SageMaker Pipelines

Anda dapat menggunakan Amazon SageMaker Model Building Pipelines untuk membuat end-to-end alur kerja yang mengelola dan menerapkan SageMaker pekerjaan. SageMaker Pipelines dilengkapi dengan integrasi SageMaker Python SDK, sehingga Anda dapat membangun setiap langkah pipeline Anda menggunakan antarmuka berbasis Python.

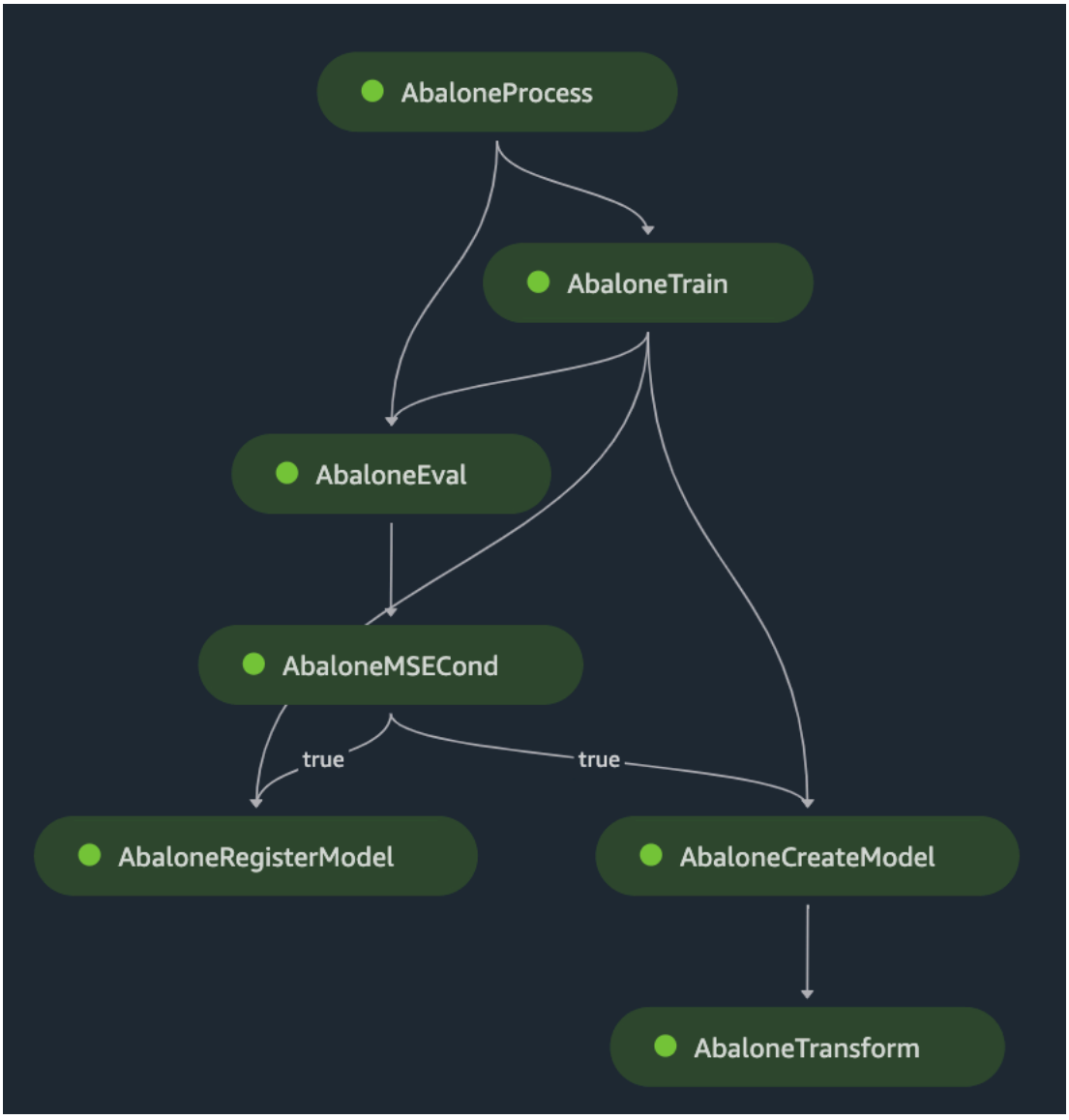
Setelah pipeline diterapkan, Anda dapat melihat grafik asiklik terarah (DAG) untuk pipeline dan mengelola eksekusi menggunakan Amazon Studio. SageMaker Menggunakan SageMaker Studio, Anda bisa mendapatkan informasi tentang pipeline Anda saat ini dan historis, membandingkan eksekusi, melihat DAG untuk eksekusi Anda, mendapatkan informasi metadata, dan banyak lagi. Untuk mempelajari cara melihat pipeline dari SageMaker Studio, lihatLihat, Lacak, dan Jalankan SageMaker Pipelines di Studio SageMaker .

Topik

- Membuat Alur
- Jalankan pipa
- Lihat, Lacak, dan Jalankan SageMaker Pipelines di Studio SageMaker

Membuat Alur

Untuk mengatur alur kerja Anda dengan SageMaker Amazon Model Building Pipelines, Anda perlu membuat grafik asiklik terarah (DAG) dalam bentuk definisi pipeline JSON. Gambar berikut adalah representasi dari pipeline DAG yang Anda buat dalam tutorial ini:



Anda dapat membuat definisi pipeline JSON Anda menggunakan SageMaker Python SDK. Tutorial berikut menunjukkan cara menghasilkan definisi pipa untuk pipa yang memecahkan masalah regresi

untuk menentukan usia abalon berdasarkan pengukuran fisiknya. Untuk notebook Jupyter yang menyertakan konten dalam tutorial ini yang dapat Anda jalankan, lihat Orchestrating Jobs with Amazon SageMaker Model Building Pipelines.

Topik

- Prasyarat
- Membuat Alur

Prasyarat

Untuk menjalankan tutorial berikut ini, Anda harus melakukan hal berikut ini:

- Siapkan instance notebook Anda seperti yang diuraikan dalam Buat instance notebook. Ini memberi izin peran Anda untuk membaca dan menulis ke Amazon S3, serta membuat pekerjaan pelatihan, transformasi batch, dan pemrosesan. SageMaker
- Berikan izin buku catatan Anda untuk mendapatkan dan meneruskan perannya sendiri seperti yang ditunjukkan dalam Memodifikasi kebijakan izin peran. Tambahkan cuplikan JSON berikut untuk melampirkan kebijakan ke peran. Ganti `<your-role-arn>` dengan ARN yang digunakan untuk membuat instance notebook Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole"
            ],
            "Resource": "<your-role-arn>"
        }
    ]
}
```

- Percayai kepala SageMaker layanan dengan mengikuti langkah-langkah dalam Memodifikasi kebijakan kepercayaan peran. Tambahkan fragmen pernyataan berikut ke hubungan kepercayaan peran Anda:

```
{
```

```
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
```

Siapkan Lingkungan Anda

Buat SageMaker sesi baru menggunakan blok kode berikut. Ini mengembalikan peran ARN untuk sesi tersebut. Peran ARN ini harus menjadi peran eksekusi ARN yang Anda tetapkan sebagai prasyarat.

```
import boto3
import sagemaker
import sagemaker.session


region = boto3.Session().region_name
sagemaker_session = sagemaker.session.Session()
role = sagemaker.get_execution_role()
default_bucket = sagemaker_session.default_bucket()
model_package_group_name = f"AbaloneModelPackageGroupName"
```

Membuat Alur

Jalankan langkah-langkah berikut dari instans SageMaker notebook Anda untuk membuat pipeline termasuk langkah-langkah untuk pra-pemrosesan, pelatihan, evaluasi, evaluasi bersyarat, dan pendaftaran model.

Langkah 1: Mengunduh Dataset

Notebook ini menggunakan Dataset Abalone Machine Learning UCI. Dataset berisi fitur berikut:

- `length`— Pengukuran cangkang terpanjang dari abalon.
- `diameter`— Diameter abalon tegak lurus dengan panjangnya.
- `height`— Ketinggian abalon dengan daging di cangkang.
- `whole_weight`— Berat seluruh abalon.

- `shucked_weight`— Berat daging dikeluarkan dari abalon.
- `viscera_weight`— Berat visera abalon setelah pendarahan.
- `shell_weight`— Berat cangkang abalon setelah pengangkatan dan pengeringan daging.
- `sex`- Jenis kelamin abalon. Salah satu dari 'M', 'F', atau 'I', di mana 'I' adalah abalon bayi.
- `rings`— Jumlah cincin di cangkang abalon.

Jumlah cincin dalam cangkang abalon adalah perkiraan yang baik untuk usianya menggunakan rumus. `age=rings + 1.5` Namun, mendapatkan nomor ini adalah tugas yang memakan waktu. Anda harus memotong cangkang melalui kerucut, menodai bagian, dan menghitung jumlah cincin melalui mikroskop. Namun, pengukuran fisik lainnya lebih mudah ditentukan. Notebook ini menggunakan dataset untuk membangun model prediktif dari cincin variabel menggunakan pengukuran fisik lainnya.

Untuk mengunduh dataset

1. Unduh kumpulan data ke dalam bucket Amazon S3 default akun Anda.

```
!mkdir -p data
local_path = "data/abalone-dataset.csv"

s3 = boto3.resource("s3")
s3.Bucket(f"sagemaker-servicecatalog-seedcode-{region}").download_file(
    "dataset/abalone-dataset.csv",
    local_path
)

base_uri = f"s3://{default_bucket}/abalone"
input_data_uri = sagemaker.s3.S3Uploader.upload(
    local_path=local_path,
    desired_s3_uri=base_uri,
)
print(input_data_uri)
```

2. Unduh kumpulan data kedua untuk transformasi batch setelah model Anda dibuat.

```
local_path = "data/abalone-dataset-batch.csv"

s3 = boto3.resource("s3")
s3.Bucket(f"sagemaker-servicecatalog-seedcode-{region}").download_file(
    "dataset/abalone-dataset-batch",
```

```
    local_path
)

base_uri = f"s3://{default_bucket}/abalone"
batch_data_uri = sagemaker.s3.S3Uploader.upload(
    local_path=local_path,
    desired_s3_uri=base_uri,
)
print(batch_data_uri)
```

Langkah 2: Tentukan Parameter Alur

Blok kode ini mendefinisikan parameter berikut untuk pipeline Anda:

- `processing_instance_count`— Jumlah instance dari pekerjaan pemrosesan.
- `input_data`— Lokasi Amazon S3 dari data input.
- `batch_data`— Lokasi input data Amazon S3 untuk transformasi batch.
- `model_approval_status`— Status persetujuan untuk mendaftarkan model terlatih untuk CI/CD. Untuk informasi selengkapnya, lihat Otomatiskan MLOP dengan Proyek SageMaker .

```
from sagemaker.workflow.parameters import (
    ParameterInteger,
    ParameterString,
)

processing_instance_count = ParameterInteger(
    name="ProcessingInstanceCount",
    default_value=1
)
model_approval_status = ParameterString(
    name="ModelApprovalStatus",
    default_value="PendingManualApproval"
)
input_data = ParameterString(
    name="InputData",
    default_value=input_data_uri,
)
batch_data = ParameterString(
    name="BatchData",
    default_value=batch_data_uri,
```

```
)
```

Langkah 3: Tentukan Langkah Pemrosesan untuk Rekayasa Fitur

Bagian ini menunjukkan cara membuat langkah pemrosesan untuk menyiapkan data dari kumpulan data untuk pelatihan.

Untuk membuat langkah pemrosesan

1. Buat direktori untuk skrip pemrosesan.

```
!mkdir -p abalone
```

2. Buat file di `/abalone` direktori bernama `preprocessing.py` dengan konten berikut. Skrip preprocessing ini diteruskan ke langkah pemrosesan untuk eksekusi pada data input. Langkah pelatihan kemudian menggunakan fitur dan label pelatihan yang telah diproses sebelumnya untuk melatih model, dan langkah evaluasi menggunakan model terlatih dan fitur dan label uji yang telah diproses sebelumnya untuk mengevaluasi model. Skrip digunakan `scikit-learn` untuk melakukan hal berikut:

   - Isi data `sex` kategoris yang hilang dan kodekan sehingga cocok untuk pelatihan.
   - Skala dan normalkan semua bidang numerik kecuali untuk `rings` dan`sex`.
   - Pisahkan data ke dalam kumpulan data pelatihan, tes, dan validasi.

```
%%writefile abalone/preprocessing.py
import argparse
import os
import requests
import tempfile
import numpy as np
import pandas as pd


from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder


# Because this is a headerless CSV file, specify the column names here.
```

```
feature_columns_names = [
    "sex",
    "length",
    "diameter",
    "height",
    "whole_weight",
    "shucked_weight",
    "viscera_weight",
    "shell_weight",
]
label_column = "rings"

feature_columns_dtype = {
    "sex": str,
    "length": np.float64,
    "diameter": np.float64,
    "height": np.float64,
    "whole_weight": np.float64,
    "shucked_weight": np.float64,
    "viscera_weight": np.float64,
    "shell_weight": np.float64
}
label_column_dtype = {"rings": np.float64}


def merge_two_dicts(x, y):
    z = x.copy()
    z.update(y)
    return z


if __name__ == "__main__":
    base_dir = "/opt/ml/processing"

    df = pd.read_csv(
        f"{base_dir}/input/abalone-dataset.csv",
        header=None,
        names=feature_columns_names + [label_column],
        dtype=merge_two_dicts(feature_columns_dtype, label_column_dtype)
    )
    numeric_features = list(feature_columns_names)
    numeric_features.remove("sex")
    numeric_transformer = Pipeline(
        steps=[
```

```
            ("imputer", SimpleImputer(strategy="median")),
            ("scaler", StandardScaler())
        ]
    )

    categorical_features = ["sex"]
    categorical_transformer = Pipeline(
        steps=[
            ("imputer", SimpleImputer(strategy="constant", fill_value="missing")),
            ("onehot", OneHotEncoder(handle_unknown="ignore"))
        ]
    )

    preprocess = ColumnTransformer(
        transformers=[
            ("num", numeric_transformer, numeric_features),
            ("cat", categorical_transformer, categorical_features)
        ]
    )

    y = df.pop("rings")
    X_pre = preprocess.fit_transform(df)
    y_pre = y.to_numpy().reshape(len(y), 1)

    X = np.concatenate((y_pre, X_pre), axis=1)

    np.random.shuffle(X)
    train, validation, test = np.split(X, [int(.7*len(X)), int(.85*len(X))])


    pd.DataFrame(train).to_csv(f"{base_dir}/train/train.csv", header=False,
index=False)
    pd.DataFrame(validation).to_csv(f"{base_dir}/validation/validation.csv",
header=False, index=False)
    pd.DataFrame(test).to_csv(f"{base_dir}/test/test.csv", header=False,
index=False)
```

3. Buat instance dari an `SKLearnProcessor` untuk diteruskan ke langkah pemrosesan.

```
from sagemaker.sklearn.processing import SKLearnProcessor


framework_version = "0.23-1"
```

```
sklearn_processor = SKLearnProcessor(
    framework_version=framework_version,
    instance_type="ml.m5.xlarge",
    instance_count=processing_instance_count,
    base_job_name="sklearn-abalone-process",
    role=role,
)
```

4. Buat langkah pemrosesan. Langkah ini mengambilSKLearnProcessor, saluran input dan output, dan preprocessing.py skrip yang Anda buat. Ini sangat mirip dengan run metode instance prosesor di SageMaker Python SDK. input_dataParameter yang diteruskan ProcessingStep adalah data input dari langkah itu sendiri. Data input ini digunakan oleh instance prosesor saat berjalan.

   Perhatikan"train,"validation, dan saluran "test" bernama yang ditentukan dalam konfigurasi output untuk pekerjaan pemrosesan. Langkah Properties seperti ini dapat digunakan dalam langkah-langkah berikutnya dan menyelesaikan nilai runtime mereka saat eksekusi.

```
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker.workflow.steps import ProcessingStep


step_process = ProcessingStep(
    name="AbaloneProcess",
    processor=sklearn_processor,
    inputs=[
      ProcessingInput(source=input_data, destination="/opt/ml/processing/input"),
    ],
    outputs=[
        ProcessingOutput(output_name="train", source="/opt/ml/processing/train"),
        ProcessingOutput(output_name="validation", source="/opt/ml/processing/
validation"),
        ProcessingOutput(output_name="test", source="/opt/ml/processing/test")
    ],
    code="abalone/preprocessing.py",
)
```

Langkah 4: Tentukan langkah Pelatihan

Bagian ini menunjukkan cara menggunakan Algoritma SageMaker XGBoost untuk melatih model pada output data pelatihan dari langkah-langkah pemrosesan.

Untuk menentukan langkah pelatihan

1. Tentukan jalur model tempat Anda ingin menyimpan model dari pelatihan.

```
model_path = f"s3://{default_bucket}/AbaloneTrain"
```

2. Konfigurasikan estimator untuk algoritma XGBoost dan dataset input. Jenis instance pelatihan diteruskan ke estimator. Skrip pelatihan tipikal memuat data dari saluran input, mengonfigurasi pelatihan dengan hiperparameter, melatih model, dan menyimpan model `model_dir` agar dapat di-host nanti. SageMaker mengunggah model ke Amazon S3 dalam bentuk `model.tar.gz` a di akhir pekerjaan pelatihan.

```
from sagemaker.estimator import Estimator


image_uri = sagemaker.image_uris.retrieve(
    framework="xgboost",
    region=region,
    version="1.0-1",
    py_version="py3",
    instance_type="ml.m5.xlarge"
)
xgb_train = Estimator(
    image_uri=image_uri,
    instance_type="ml.m5.xlarge",
    instance_count=1,
    output_path=model_path,
    role=role,
)
xgb_train.set_hyperparameters(
    objective="reg:linear",
    num_round=50,
    max_depth=5,
    eta=0.2,
    gamma=4,
    min_child_weight=6,
    subsample=0.7,
```

```
    silent=0
)
```

3. Buat `TrainingStep` menggunakan instance estimator dan properti dari. `ProcessingStep` Secara khusus, lewati saluran `"train"` dan `"validation"` output ke`TrainingStep`. `S3Uri`

```
from sagemaker.inputs import TrainingInput
from sagemaker.workflow.steps import TrainingStep


step_train = TrainingStep(
    name="AbaloneTrain",
    estimator=xgb_train,
    inputs={
        "train": TrainingInput(
            s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
                "train"
            ].S3Output.S3Uri,
            content_type="text/csv"
        ),
        "validation": TrainingInput(
            s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
                "validation"
            ].S3Output.S3Uri,
            content_type="text/csv"
        )
    },
)
```

Langkah 5: Tentukan Langkah Pemrosesan untuk Evaluasi Model

Bagian ini menunjukkan cara membuat langkah pemrosesan untuk mengevaluasi keakuratan model. Hasil evaluasi model ini digunakan dalam langkah kondisi untuk menentukan jalur eksekusi mana yang harus diambil.

Untuk menentukan langkah pemrosesan untuk evaluasi model

1. Buat file di `/abalone` direktori bernama`evaluation.py`. Skrip ini digunakan dalam langkah pemrosesan untuk melakukan evaluasi model. Dibutuhkan model terlatih dan kumpulan data pengujian sebagai input, kemudian menghasilkan file JSON yang berisi metrik evaluasi klasifikasi.

```
%%writefile abalone/evaluation.py
import json
import pathlib
import pickle
import tarfile
import joblib
import numpy as np
import pandas as pd
import xgboost


from sklearn.metrics import mean_squared_error


if __name__ == "__main__":
    model_path = f"/opt/ml/processing/model/model.tar.gz"
    with tarfile.open(model_path) as tar:
        tar.extractall(path=".")

    model = pickle.load(open("xgboost-model", "rb"))

    test_path = "/opt/ml/processing/test/test.csv"
    df = pd.read_csv(test_path, header=None)

    y_test = df.iloc[:, 0].to_numpy()
    df.drop(df.columns[0], axis=1, inplace=True)

    X_test = xgboost.DMatrix(df.values)

    predictions = model.predict(X_test)

    mse = mean_squared_error(y_test, predictions)
    std = np.std(y_test - predictions)
    report_dict = {
        "regression_metrics": {
            "mse": {
                "value": mse,
                "standard_deviation": std
            },
        },
    }

    output_dir = "/opt/ml/processing/evaluation"
```

```
    pathlib.Path(output_dir).mkdir(parents=True, exist_ok=True)

    evaluation_path = f"{output_dir}/evaluation.json"
    with open(evaluation_path, "w") as f:
        f.write(json.dumps(report_dict))
```

2. Buat sebuah instance dari sebuah `ScriptProcessor` yang digunakan untuk membuat sebuah`ProcessingStep`.

```
from sagemaker.processing import ScriptProcessor


script_eval = ScriptProcessor(
    image_uri=image_uri,
    command=["python3"],
    instance_type="ml.m5.xlarge",
    instance_count=1,
    base_job_name="script-abalone-eval",
    role=role,
)
```

3. Buat `ProcessingStep` menggunakan instance prosesor, saluran input dan output, dan `evaluation.py` skrip. Secara khusus, lewati `S3ModelArtifacts` properti dari langkah `step_train` pelatihan, serta saluran `S3Uri` `"test"` output dari langkah `step_process` pemrosesan. Ini sangat mirip dengan `run` metode instance prosesor di SageMaker Python SDK.

```
from sagemaker.workflow.properties import PropertyFile


evaluation_report = PropertyFile(
    name="EvaluationReport",
    output_name="evaluation",
    path="evaluation.json"
)
step_eval = ProcessingStep(
    name="AbaloneEval",
    processor=script_eval,
    inputs=[
        ProcessingInput(
            source=step_train.properties.ModelArtifacts.S3ModelArtifacts,
            destination="/opt/ml/processing/model"
```

```
        ),
        ProcessingInput(
            source=step_process.properties.ProcessingOutputConfig.Outputs[
                "test"
            ].S3Output.S3Uri,
            destination="/opt/ml/processing/test"
        )
    ],
    outputs=[
        ProcessingOutput(output_name="evaluation", source="/opt/ml/processing/
evaluation"),
    ],
    code="abalone/evaluation.py",
    property_files=[evaluation_report],
)
```

Langkah 6: Tentukan Transformasi Batch CreateModelStep

> **Important**
>
> Kami merekomendasikan penggunaan Contoh untuk membuat model pada v2.90.0 dari Python SDK. SageMaker `CreateModelStep`akan terus bekerja di versi SDK SageMaker Python sebelumnya, tetapi tidak lagi didukung secara aktif.

Bagian ini menunjukkan cara membuat SageMaker model dari output langkah pelatihan. Model ini digunakan untuk transformasi batch pada dataset baru. Langkah ini diteruskan ke langkah kondisi dan hanya dijalankan jika langkah kondisi dievaluasi. `true`

Untuk menentukan transformasi CreateModelStep untuk batch

1. Buat SageMaker model. Lewati `S3ModelArtifacts` properti dari langkah `step_train` pelatihan.

```
from sagemaker.model import Model


model = Model(
    image_uri=image_uri,
    model_data=step_train.properties.ModelArtifacts.S3ModelArtifacts,
    sagemaker_session=sagemaker_session,
```

```
    role=role,
)
```

2. Tentukan input model untuk SageMaker model Anda.

```
from sagemaker.inputs import CreateModelInput


inputs = CreateModelInput(
    instance_type="ml.m5.large",
    accelerator_type="ml.eia1.medium",
)
```

3. Buat contoh `CreateModelStep` menggunakan `CreateModelInput` dan SageMaker model yang Anda tentukan.

```
from sagemaker.workflow.steps import CreateModelStep


step_create_model = CreateModelStep(
    name="AbaloneCreateModel",
    model=model,
    inputs=inputs,
)
```

Langkah 7: Tentukan TransformStep untuk Melakukan Transformasi Batch

Bagian ini menunjukkan cara membuat `TransformStep` untuk melakukan transformasi batch pada kumpulan data setelah model dilatih. Langkah ini diteruskan ke langkah kondisi dan hanya dijalankan jika langkah kondisi dievaluasi. `true`

Untuk menentukan TransformStep untuk melakukan transformasi batch

1. Buat instance transformator dengan jenis instans komputasi yang sesuai, jumlah instans, dan URI bucket Amazon S3 keluaran yang diinginkan. Lewati `ModelName` properti dari `step_create_model` `CreateModel` langkah.

```
from sagemaker.transformer import Transformer


transformer = Transformer(
```

```
    model_name=step_create_model.properties.ModelName,
    instance_type="ml.m5.xlarge",
    instance_count=1,
    output_path=f"s3://{default_bucket}/AbaloneTransform"
)
```

2. Buat `TransformStep` menggunakan instance transformator yang Anda tentukan dan parameter `batch_data` pipeline.

```
from sagemaker.inputs import TransformInput
from sagemaker.workflow.steps import TransformStep


step_transform = TransformStep(
    name="AbaloneTransform",
    transformer=transformer,
    inputs=TransformInput(data=batch_data)
)
```

Langkah 8: Tentukan RegisterModel Langkah untuk Membuat Model Package

> **Important**
>
> Kami merekomendasikan penggunaan Contoh untuk mendaftarkan model pada v2.90.0 dari Python SDK. SageMaker `RegisterModel`akan terus bekerja di versi SDK SageMaker Python sebelumnya, tetapi tidak lagi didukung secara aktif.

Bagian ini menunjukkan cara membangun sebuah instans dari. `RegisterModel` Hasil eksekusi `RegisterModel` dalam pipeline adalah paket model. Paket model adalah abstraksi artefak model yang dapat digunakan kembali yang mengemas semua bahan yang diperlukan untuk inferensi. Ini terdiri dari spesifikasi inferensi yang mendefinisikan gambar inferensi untuk digunakan bersama dengan lokasi bobot model opsional. Grup paket model adalah kumpulan paket model. Anda dapat menggunakan `ModelPackageGroup` for SageMaker Pipelines untuk menambahkan versi baru dan paket model ke grup untuk setiap eksekusi pipeline. Untuk informasi lebih lanjut tentang registri model, lihatDaftarkan dan Terapkan Model dengan Registri Model.

Langkah ini diteruskan ke langkah kondisi dan hanya dijalankan jika langkah kondisi dievaluasi. `true`

Untuk menentukan RegisterModel langkah untuk membuat paket model

- Buat `RegisterModel` langkah menggunakan instance estimator yang Anda gunakan untuk langkah pelatihan. Lewati `S3ModelArtifacts` properti dari langkah `step_train` pelatihan dan tentukan a`ModelPackageGroup`. SageMaker Pipelines menciptakan ini `ModelPackageGroup` untuk Anda.

```
from sagemaker.model_metrics import MetricsSource, ModelMetrics
from sagemaker.workflow.step_collections import RegisterModel


model_metrics = ModelMetrics(
    model_statistics=MetricsSource(
        s3_uri="{}/evaluation.json".format(
            step_eval.arguments["ProcessingOutputConfig"]["Outputs"][0]["S3Output"]
["S3Uri"]
        ),
        content_type="application/json"
    )
)
step_register = RegisterModel(
    name="AbaloneRegisterModel",
    estimator=xgb_train,
    model_data=step_train.properties.ModelArtifacts.S3ModelArtifacts,
    content_types=["text/csv"],
    response_types=["text/csv"],
    inference_instances=["ml.t2.medium", "ml.m5.xlarge"],
    transform_instances=["ml.m5.xlarge"],
    model_package_group_name=model_package_group_name,
    approval_status=model_approval_status,
    model_metrics=model_metrics
)
```

Langkah 9: Tentukan Langkah Kondisi untuk Memverifikasi Akurasi Model

A `ConditionStep` memungkinkan SageMaker Pipelines untuk mendukung eksekusi bersyarat di DAG pipeline Anda berdasarkan kondisi properti langkah. Dalam hal ini, Anda hanya ingin mendaftarkan paket model jika keakuratan model tersebut, sebagaimana ditentukan oleh langkah evaluasi model, melebihi nilai yang diperlukan. Jika akurasi melebihi nilai yang diperlukan, pipeline juga membuat SageMaker Model dan menjalankan transformasi batch pada kumpulan data. Bagian ini menunjukkan cara menentukan langkah Kondisi.

Untuk menentukan langkah kondisi untuk memverifikasi akurasi model

1. Tentukan `ConditionLessThanOrEqualTo` kondisi menggunakan nilai akurasi yang ditemukan dalam output dari langkah pemrosesan evaluasi model,`step_eval`. Dapatkan output ini menggunakan file properti yang Anda indeks dalam langkah pemrosesan dan masing-masing JsonPath dari nilai kesalahan kuadrat rata-rata,. `"mse"`

```
from sagemaker.workflow.conditions import ConditionLessThanOrEqualTo
from sagemaker.workflow.condition_step import ConditionStep
from sagemaker.workflow.functions import JsonGet


cond_lte = ConditionLessThanOrEqualTo(
    left=JsonGet(
        step_name=step_eval.name,
        property_file=evaluation_report,
        json_path="regression_metrics.mse.value"
    ),
    right=6.0
)
```

2. Bangun a. `ConditionStep` Lulus `ConditionEquals` kondisi, lalu atur pendaftaran paket model dan langkah transformasi batch sebagai langkah selanjutnya jika kondisi berlalu.

```
step_cond = ConditionStep(
    name="AbaloneMSECond",
    conditions=[cond_lte],
    if_steps=[step_register, step_create_model, step_transform],
    else_steps=[],
)
```

Langkah 10: Membuat Alur

Sekarang setelah Anda membuat semua langkah, gabungkan mereka ke dalam pipeline.

Untuk membuat alur

1. Tentukan hal berikut untuk pipeline Anda:`name`,`parameters`, dan`steps`. Nama dalam `(account, region)` pasangan harus unik.

> **Note**
>
> Sebuah langkah hanya dapat muncul sekali di daftar langkah pipeline atau daftar langkah jika/lain dari langkah kondisi. Itu tidak bisa muncul di keduanya.

```
from sagemaker.workflow.pipeline import Pipeline


pipeline_name = f"AbalonePipeline"
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[
        processing_instance_count,
        model_approval_status,
        input_data,
        batch_data,
    ],
    steps=[step_process, step_train, step_eval, step_cond],
)
```

2. (Opsional) Periksa definisi pipa JSON untuk memastikan bahwa itu terbentuk dengan baik.

```
import json

json.loads(pipeline.definition())
```

Definisi pipeline ini siap untuk dikirimkan SageMaker. Dalam tutorial berikutnya, Anda mengirimkan pipeline ini ke SageMaker dan memulai eksekusi.

Langkah selanjutnya: Jalankan pipa

Jalankan pipa

Setelah Anda membuat definisi pipeline menggunakan SageMaker Python SDK, Anda dapat mengirimkannya SageMaker untuk memulai eksekusi. Tutorial berikut menunjukkan cara mengirimkan pipeline, memulai eksekusi, memeriksa hasil eksekusi itu, dan menghapus pipeline Anda.

Topik

Prasyarat

Tutorial ini membutuhkan hal berikut:

- Contoh SageMaker notebook.
- Definisi SageMaker pipa pipa. Tutorial ini mengasumsikan Anda menggunakan definisi pipeline yang dibuat dengan menyelesaikan Membuat Alur tutorial.

Langkah 1: Memulai Alur

Pertama, Anda harus memulai pipa.

Untuk memulai alur

1. Periksa definisi pipa JSON untuk memastikan bahwa itu terbentuk dengan baik.

```
import json

json.loads(pipeline.definition())
```

2. Kirimkan definisi SageMaker pipeline ke layanan Pipelines untuk membuat pipeline jika tidak ada, atau perbarui pipeline jika ada. Peran yang diteruskan digunakan oleh SageMaker Pipelines untuk menciptakan semua pekerjaan yang ditentukan dalam langkah-langkah.

```
pipeline.upsert(role_arn=role)
```

3. Mulai eksekusi pipeline.

```
execution = pipeline.start()
```

Langkah 2: Periksa Eksekusi Pipeline

Selanjutnya, Anda perlu memeriksa eksekusi pipa.

Untuk memeriksa eksekusi pipa

1. Jelaskan status eksekusi pipeline untuk memastikan bahwa itu telah dibuat dan dimulai dengan sukses.

```
execution.describe()
```

2. Tunggu eksekusi selesai.

```
execution.wait()
```

3. Buat daftar langkah-langkah eksekusi dan statusnya.

```
execution.list_steps()
```

Outputnya akan terlihat seperti berikut:

```
[{'StepName': 'AbaloneTransform',
  'StartTime': datetime.datetime(2020, 11, 21, 2, 41, 27, 870000,
 tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 11, 21, 2, 45, 50, 492000, tzinfo=tzlocal()),
  'StepStatus': 'Succeeded',
  'CacheHitResult': {'SourcePipelineExecutionArn': ''},
  'Metadata': {'TransformJob': {'Arn': 'arn:aws:sagemaker:us-
east-2:111122223333:transform-job/pipelines-cfvy1tjuxdq8-abalonetransform-
ptyjoef3jy'}}},
 {'StepName': 'AbaloneRegisterModel',
  'StartTime': datetime.datetime(2020, 11, 21, 2, 41, 26, 929000,
 tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 11, 21, 2, 41, 28, 15000, tzinfo=tzlocal()),
  'StepStatus': 'Succeeded',
  'CacheHitResult': {'SourcePipelineExecutionArn': ''},
  'Metadata': {'RegisterModel': {'Arn': 'arn:aws:sagemaker:us-
east-2:111122223333:model-package/abalonemodelpackagegroupname/1'}}},
 {'StepName': 'AbaloneCreateModel',
  'StartTime': datetime.datetime(2020, 11, 21, 2, 41, 26, 895000,
 tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 11, 21, 2, 41, 27, 708000, tzinfo=tzlocal()),
```

```
  'StepStatus': 'Succeeded',
  'CacheHitResult': {'SourcePipelineExecutionArn': ''},
  'Metadata': {'Model': {'Arn': 'arn:aws:sagemaker:us-east-2:111122223333:model/
pipelines-cfvy1tjuxdq8-abalonecreatemodel-jl94rai0ra'}}},
 {'StepName': 'AbaloneMSECond',
  'StartTime': datetime.datetime(2020, 11, 21, 2, 41, 25, 558000,
 tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 11, 21, 2, 41, 26, 329000, tzinfo=tzlocal()),
  'StepStatus': 'Succeeded',
  'CacheHitResult': {'SourcePipelineExecutionArn': ''},
  'Metadata': {'Condition': {'Outcome': 'True'}}},
 {'StepName': 'AbaloneEval',
  'StartTime': datetime.datetime(2020, 11, 21, 2, 37, 34, 767000,
 tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 11, 21, 2, 41, 18, 80000, tzinfo=tzlocal()),
  'StepStatus': 'Succeeded',
  'CacheHitResult': {'SourcePipelineExecutionArn': ''},
  'Metadata': {'ProcessingJob': {'Arn': 'arn:aws:sagemaker:us-
east-2:111122223333:processing-job/pipelines-cfvy1tjuxdq8-abaloneeval-
zfraozhmny'}}},
 {'StepName': 'AbaloneTrain',
  'StartTime': datetime.datetime(2020, 11, 21, 2, 34, 55, 867000,
 tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 11, 21, 2, 37, 34, 34000, tzinfo=tzlocal()),
  'StepStatus': 'Succeeded',
  'CacheHitResult': {'SourcePipelineExecutionArn': ''},
  'Metadata': {'TrainingJob': {'Arn': 'arn:aws:sagemaker:us-
east-2:111122223333:training-job/pipelines-cfvy1tjuxdq8-abalonetrain-
tavd6f3wdf'}}},
 {'StepName': 'AbaloneProcess',
  'StartTime': datetime.datetime(2020, 11, 21, 2, 30, 27, 160000,
 tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 11, 21, 2, 34, 48, 390000, tzinfo=tzlocal()),
  'StepStatus': 'Succeeded',
  'CacheHitResult': {'SourcePipelineExecutionArn': ''},
  'Metadata': {'ProcessingJob': {'Arn': 'arn:aws:sagemaker:us-
east-2:111122223333:processing-job/pipelines-cfvy1tjuxdq8-abaloneprocess-
mgqyfdujcj'}}}]
```

4. Setelah eksekusi pipeline Anda selesai, unduh `evaluation.json` file yang dihasilkan dari Amazon S3 untuk memeriksa laporan.

```
evaluation_json = sagemaker.s3.S3Downloader.read_file("{}/evaluation.json".format(
```

```
    step_eval.arguments["ProcessingOutputConfig"]["Outputs"][0]["S3Output"]
["S3Uri"]
))
json.loads(evaluation_json)
```

Langkah 3: Ganti Parameter Default untuk Eksekusi Pipeline

Anda dapat menjalankan eksekusi tambahan dari pipeline dengan menentukan parameter pipeline yang berbeda untuk mengganti default.

Untuk mengganti parameter default

1. Buat eksekusi pipeline. Ini memulai eksekusi pipeline lain dengan penggantian status persetujuan model disetel ke “Disetujui”. Ini berarti bahwa versi paket model yang dihasilkan oleh `RegisterModel` langkah secara otomatis siap untuk penyebaran melalui pipeline CI/CD, seperti dengan Proyek. SageMaker Untuk informasi selengkapnya, lihat Otomatiskan MLOP dengan Proyek SageMaker .

   ```
   execution = pipeline.start(
       parameters=dict(
           ModelApprovalStatus="Approved",
       )
   )
   ```

2. Tunggu eksekusi selesai.

   ```
   execution.wait()
   ```

3. Buat daftar langkah-langkah eksekusi dan statusnya.

   ```
   execution.list_steps()
   ```

4. Setelah eksekusi pipeline Anda selesai, unduh `evaluation.json` file yang dihasilkan dari Amazon S3 untuk memeriksa laporan.

   ```
   evaluation_json = sagemaker.s3.S3Downloader.read_file("{}/evaluation.json".format(
       step_eval.arguments["ProcessingOutputConfig"]["Outputs"][0]["S3Output"]
   ["S3Uri"]
   ))
   json.loads(evaluation_json)
   ```

Langkah 4: Hentikan dan Hapus Eksekusi Pipeline

Setelah Anda selesai dengan alur, Anda dapat menghentikan eksekusi yang sedang berlangsung dan menghapus alur.

Untuk menghentikan dan menghapus eksekusi pipeline

1. Hentikan eksekusi pipa.

```
execution.stop()
```

2. Hapus Alur.

```
pipeline.delete()
```

Lihat, Lacak, dan Jalankan SageMaker Pipelines di Studio SageMaker

Untuk melihat, melacak, dan menjalankan Amazon SageMaker Pipelines di Amazon SageMaker Studio, Anda harus masuk ke Studio. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio.

Topik

- Membuat Alur
- Melihat Eksekusi Alur
- Mengunduh Definisi Alur
- Lihat Entitas Eksperimen yang Dibuat oleh SageMaker Pipelines
- Mulai (dan Hentikan) Eksekusi Pipeline
- Lacak Silsilah Pipeline dari Pipeline SageMaker

Membuat Alur

Prosedur ini menunjukkan kepada Anda cara menemukan pipeline secara langsung dan melihat halaman detailnya. Anda juga dapat menemukan saluran pipa yang merupakan bagian dari proyek yang tercantum di halaman detail proyek. Untuk informasi tentang menemukan pipa yang merupakan bagian dari proyek, lihatOtomatiskan MLOP dengan Proyek SageMaker .

Untuk melihat daftar pipeline di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi sebelah kiri, pilih Pipelines.
3. (Opsional) Untuk memfilter daftar pipa berdasarkan nama, masukkan nama pipa lengkap atau sebagian di kolom pencarian.
4. Pilih nama pipeline untuk melihat detail tentang pipeline. Halaman Eksekusi pipeline membuka dan menampilkan daftar eksekusi pipeline. Gunakan ikon Kolom () untuk memilih kolom mana yang akan ditampilkan.
5. Dari halaman Eksekusi pipeline, pilih salah satu halaman berikut di menu tarik-turun Ikhtisar, Pengaturan, atau Detail (di sebelah kiri tabel eksekusi pipeline) untuk melihat detail pipeline:
   - Eksekusi — Detail tentang eksekusi.
   - Grafik — DAG untuk pipa.
   - Parameter - Termasuk status persetujuan model.
   - Informasi — Metadata yang terkait dengan pipeline, seperti saluran Amazon Resource Name (ARN) dan peran ARN. Anda juga dapat mengedit deskripsi saluran pipa dari halaman ini.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di sidebar Studio Classic, pilih ikon Beranda ().
3. Pilih Pipelines dari menu.
4. Untuk mempersempit daftar saluran pipa dengan nama, masukkan nama pipa lengkap atau sebagian di bidang pencarian.
5. Pilih nama pipeline untuk melihat detail tentang pipeline. Tab rincian pipeline membuka dan menampilkan daftar eksekusi pipeline. Anda dapat memulai eksekusi atau memilih salah satu tab lain untuk informasi lebih lanjut tentang pipeline. Gunakan ikon Property Inspector

()
untuk memilih kolom mana yang akan ditampilkan.

6. Dari halaman detail pipeline, pilih salah satu tab berikut untuk melihat detail tentang pipeline:
   - Eksekusi — Detail tentang eksekusi. Anda dapat membuat eksekusi dari tab ini atau tab Grafik.
   - Grafik — DAG untuk pipa.
   - Parameter - Termasuk status persetujuan model.
   - Pengaturan — Metadata yang terkait dengan pipa. Anda dapat mengunduh file definisi pipeline dan mengedit nama dan deskripsi pipeline dari tab ini.

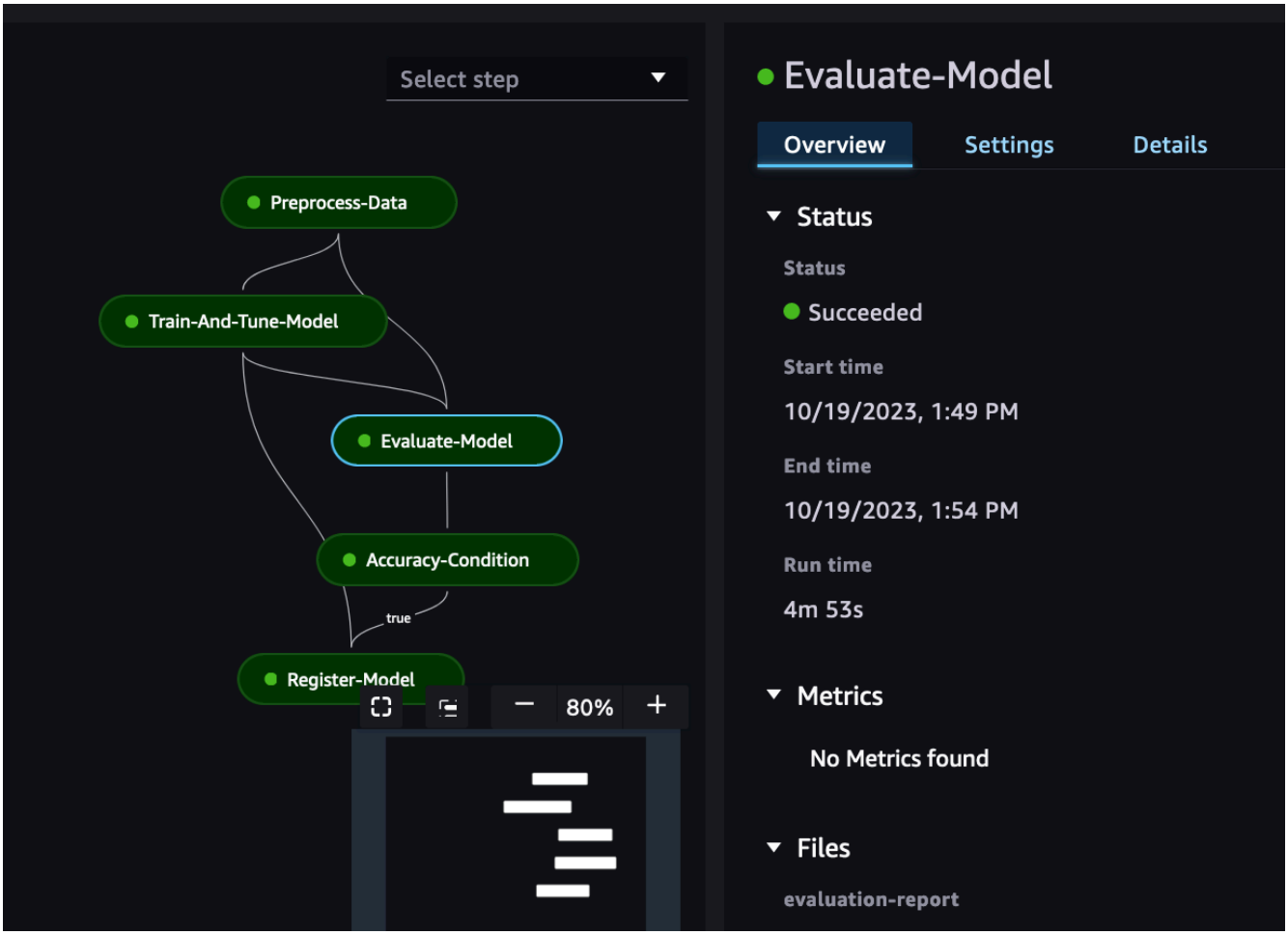
Melihat Eksekusi Alur

Prosedur ini menunjukkan cara melihat eksekusi pipeline. Untuk informasi tentang cara melihat daftar eksekusi pipeline, dan cara menggunakan SageMaker pencarian untuk mempersempit eksekusi dalam daftar, lihat. Membuat Alur

Untuk melihat eksekusi pipeline di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi sebelah kiri, pilih Pipelines.
3. (Opsional) Untuk memfilter daftar pipa berdasarkan nama, masukkan nama pipa lengkap atau sebagian di kolom pencarian.
4. Pilih nama pipeline untuk melihat detail tentang pipeline. Halaman Eksekusi pipeline membuka dan menampilkan daftar eksekusi pipeline.
5. Pilih nama eksekusi pipeline untuk dilihat. Grafik pipa eksekusi muncul.
6. (Opsional) Pilih langkah di Pilih langkah menu tarik-turun di sebelah kanan grafik untuk memusatkan grafik pada langkah yang Anda pilih. Gunakan ikon pengubahan ukuran di sisi kanan bawah grafik untuk memperbesar dan memperkecil grafik, menyesuaikan grafik ke layar, dan memperluas grafik ke layar penuh. Untuk fokus pada bagian tertentu dari grafik, Anda dapat memilih area kosong grafik dan menyeret grafik ke tengah area itu.



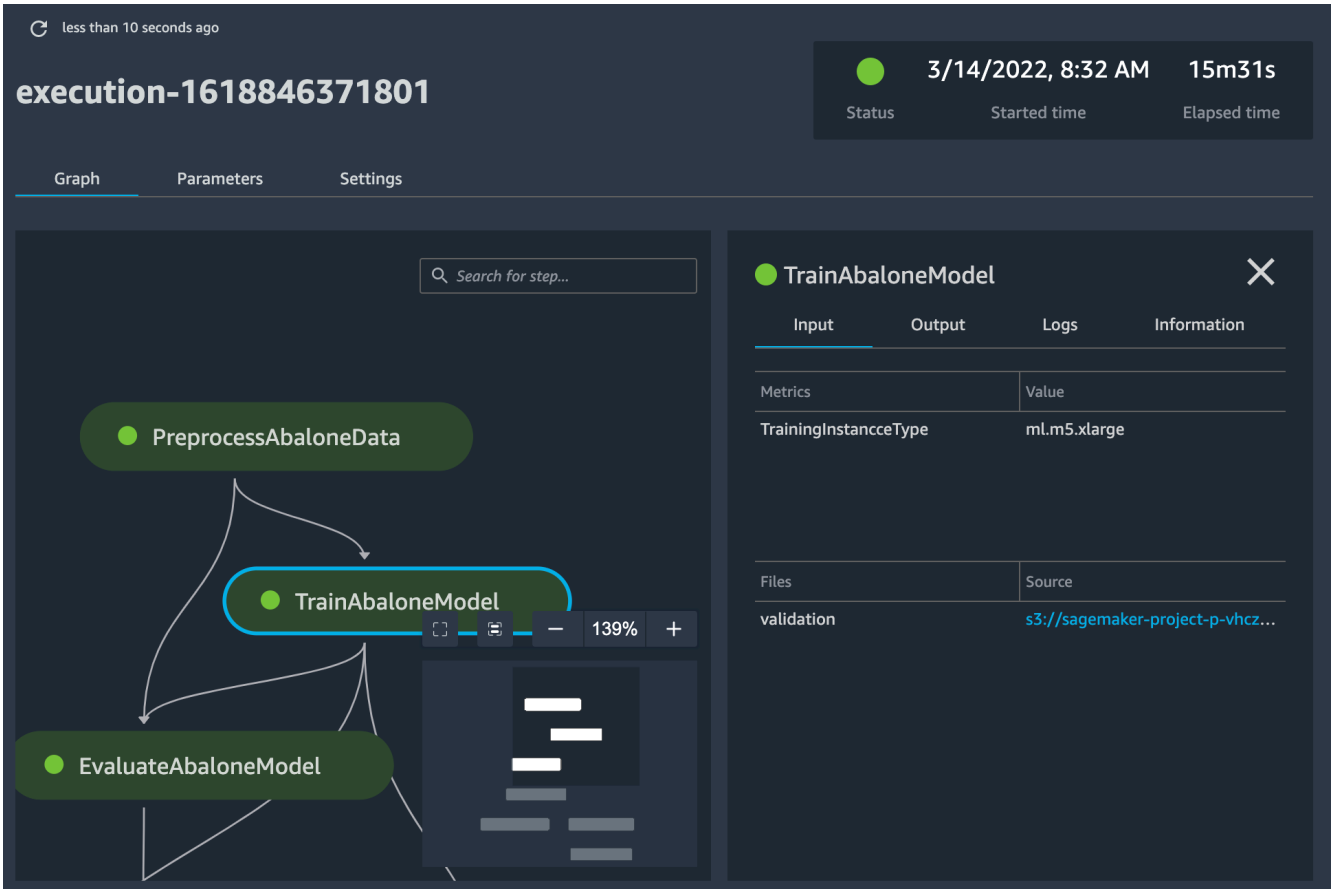
7. Pilih salah satu langkah pipeline dalam grafik untuk melihat detail tentang langkah tersebut. Anda dapat melihat detail eksekusi langkah di tab berikut:

   - Ikhtisar — Detail yang terkait dengan eksekusi langkah, termasuk status dan runtime, metrik dan bagan terkait, dan lokasi file agunan keluaran.
   - Pengaturan — Parameter dan nilai yang terkait dengan langkah pipeline Anda, seperti yang ditentukan oleh definisi JSON untuk langkah tersebut. Termasuk skrip masukan dan kumpulan data.
   - Detail — Informasi umum tentang langkah, termasuk jenis langkah (seperti pemrosesan atau pelatihan), dan lokasi file log.

Studio Classic

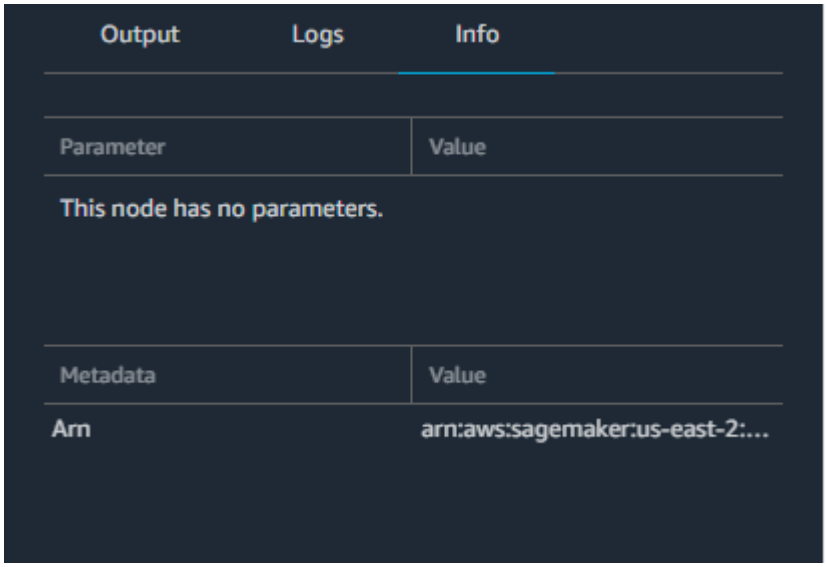
1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.

2. Di sidebar Studio Classic, pilih ikon Beranda
   ().
3. Pilih Pipelines dari menu.
4. Untuk mempersempit daftar saluran pipa dengan nama, masukkan nama pipa lengkap atau sebagian di bidang pencarian.
5. Pilih nama pipeline. Halaman Eksekusi pipeline terbuka.
6. Di halaman Eksekusi, pilih nama eksekusi untuk melihat detail tentang eksekusi. Tab detail eksekusi terbuka dan menampilkan grafik langkah-langkah dalam pipeline.
7. Untuk mencari langkah demi nama, ketik karakter yang cocok dengan nama langkah di bidang pencarian. Gunakan ikon pengubahan ukuran di sisi kanan bawah grafik untuk memperbesar dan memperkecil grafik, menyesuaikan grafik ke layar, dan memperluas grafik ke layar penuh. Untuk fokus pada bagian tertentu dari grafik, Anda dapat memilih area kosong grafik dan menyeret grafik ke tengah area itu.



8. Pilih salah satu langkah pipeline dalam grafik untuk melihat detail tentang langkah tersebut. Pada tangkapan layar sebelumnya, langkah pelatihan dipilih dan menampilkan tab berikut:

- Input — Input pelatihan. Jika sumber input berasal dari Amazon Simple Storage Service (Amazon S3), pilih tautan untuk melihat file di konsol Amazon S3.
- Output — Output pelatihan, seperti metrik, bagan, file, dan hasil evaluasi. Grafik diproduksi menggunakan API Pelacak.
- Log — CloudWatch Log Amazon diproduksi oleh langkah.
- Info — Parameter dan metadata yang terkait dengan langkah.



Mengunduh Definisi Alur

Anda dapat mengunduh definisi alur di konsol Amazon SageMaker Studio. Untuk mengunduh definisi pipeline, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi sebelah kiri, pilih Pipelines.
3. (Opsional) Untuk memfilter daftar pipa berdasarkan nama, masukkan nama pipa lengkap atau sebagian di kolom pencarian.
4. Pilih nama pipeline. Halaman Eksekusi membuka dan menampilkan daftar eksekusi pipeline.

5. Tetap di halaman Eksekusi atau pilih halaman Grafik, Informasi, atau Parameter di sebelah kiri tabel eksekusi pipeline. Anda dapat mengunduh definisi pipeline dari salah satu halaman ini.
6. Di kanan atas halaman, pilih elipsis vertikal dan pilih Download pipeline definition (JSON).

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di sidebar Studio Classic, pilih ikon Beranda ().
3. Pilih Pipelines dari menu.
4. Untuk mempersempit daftar saluran pipa dengan nama, masukkan nama pipa lengkap atau sebagian di bidang pencarian.
5. Pilih nama pipeline.
6. Pilih tab Pengaturan.
7. Pilih Unduh file definisi saluran pipa.

Lihat Entitas Eksperimen yang Dibuat oleh SageMaker Pipelines

> **Note**
>
> SageMaker Eksperimen adalah fitur yang disediakan di Studio Classic saja.

Saat Anda membuat pipeline dan menentukan pipeline_experiment_config, SageMaker Pipelines akan membuat entitas SageMaker Eksperimen berikut secara default jika tidak ada:

- Eksperimen untuk pipa
- Grup run untuk setiap eksekusi pipeline
- Jalankan untuk setiap SageMaker pekerjaan yang dibuat dalam langkah pipeline

Untuk informasi tentang bagaimana eksperimen diintegrasikan dengan jaringan pipa, lihatIntegrasi SageMaker Eksperimen Amazon. Untuk informasi lebih lanjut tentang SageMaker Eksperimen, lihatKelola Machine Learning dengan Amazon SageMaker Experiments.

Anda dapat membuka daftar proses yang terkait dengan pipeline baik dari daftar eksekusi pipeline atau daftar eksperimen.

Untuk melihat daftar run dari daftar eksekusi pipeline

1. Untuk melihat daftar eksekusi pipeline, ikuti lima langkah pertama di tab Studio Classic. Membuat Alur
2. Di bagian kanan atas layar, pilih ikon Filter
   ().
3. Pilih Eksperimen. Jika integrasi eksperimen tidak dinonaktifkan saat pipeline dibuat, nama eksperimen akan ditampilkan dalam daftar eksekusi.

   > **Note**
   >
   > Integrasi eksperimen diperkenalkan di v2.41.0 dari Amazon Python SageMaker SDK. Pipeline yang dibuat dengan versi SDK yang lebih lama tidak terintegrasi dengan eksperimen secara default.

4. Pilih eksperimen pilihan Anda untuk melihat grup yang dijalankan dan menjalankan yang terkait dengan eksperimen tersebut.

Untuk melihat daftar berjalan dari daftar eksperimen

1. Di bilah sisi kiri Studio Classic, pilih ikon Beranda
   ().
2. Pilih Eksperimen dari menu.
3. Gunakan bilah pencarian atau ikon Filter
   ()
   untuk memfilter daftar ke eksperimen yang dibuat oleh pipeline.
4. Buka nama eksperimen dan lihat daftar proses yang dibuat oleh pipeline.

Mulai (dan Hentikan) Eksekusi Pipeline

Anda dapat memulai dan menghentikan eksekusi pipeline di konsol Amazon SageMaker Studio. Untuk informasi tentang cara melihat daftar eksekusi alur, lihatMembuat Alur.

Untuk memulai dan menghentikan eksekusi pipeline di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

Untuk memulai eksekusi pipeline

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi sebelah kiri, pilih Pipelines.
3. (Opsional) Untuk memfilter daftar pipa berdasarkan nama, masukkan nama pipa lengkap atau sebagian di kolom pencarian.
4. Pilih nama pipeline. Halaman Eksekusi membuka dan menampilkan daftar eksekusi pipeline.
5. Anda dapat membuat eksekusi dari halaman Eksekusi atau Grafik. Untuk membuat eksekusi dari halaman Eksekusi, pilih Buat. Untuk membuat eksekusi dari halaman Grafik, pilih Grafik di sebelah kiri tabel eksekusi dan kemudian Buat eksekusi di kanan atas DAG.
6. Masukkan atau perbarui informasi berikut:
   - Nama — Nama unik untuk akun Anda di AWS Wilayah.
   - Deskripsi — Deskripsi opsional untuk eksekusi Anda.
   - ProcessingInstanceType- Jenis instans Amazon EC2 yang akan digunakan untuk pekerjaan pemrosesan.
   - TrainingInstanceType— Jenis instans Amazon EC2 yang akan digunakan untuk pekerjaan pelatihan
   - InputData— URI Amazon S3 ke data input.
   - PreprocessScript— URI Amazon S3 ke skrip preprocessing.
   - EvaluateScript— URI Amazon S3 ke skrip evaluasi model.
   - AccuracyConditionThreshold— Ambang akurasi model yang harus dicapai untuk mendaftarkan model ke dalam registri.
   - ModelGroup— Registri untuk mendaftarkan model.
   - MaximumParallelTrainingJobs— Jumlah maksimum pekerjaan pelatihan untuk dijalankan secara paralel.
   - MaximumTrainingJobs— Jumlah maksimum pekerjaan pelatihan untuk dijalankan.
7. Pilih Buat.

Untuk menghentikan eksekusi pipeline

1. Di panel navigasi sebelah kiri, pilih Pipelines.
2. (Opsional) Untuk memfilter daftar pipa berdasarkan nama, masukkan nama pipa lengkap atau sebagian di kolom pencarian.
3. Pilih nama pipeline. Halaman Eksekusi membuka dan menampilkan daftar eksekusi pipeline.
4. Pilih eksekusi untuk berhenti.
5. Pilih Berhenti.

Untuk melanjutkan eksekusi pipa yang dihentikan

1. Di panel navigasi sebelah kiri, pilih Pipelines.
2. (Opsional) Untuk memfilter daftar pipa berdasarkan nama, masukkan nama pipa lengkap atau sebagian di kolom pencarian.
3. Pilih nama pipeline. Halaman Eksekusi membuka dan menampilkan daftar eksekusi pipeline.
4. Pilih eksekusi untuk melanjutkan.
5. Pilih Resume.

Studio Classic

Untuk memulai, menghentikan, atau melanjutkan eksekusi pipa.

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di sidebar Studio Classic, pilih ikon Beranda ().
3. Pilih Pipelines dari menu.
4. Untuk mempersempit daftar saluran pipa dengan nama, masukkan nama pipa lengkap atau sebagian di bidang pencarian.
5. Pilih nama pipeline.
6. Dari tab Eksekusi atau Grafik dalam daftar eksekusi, pilih Buat eksekusi.
7. Masukkan atau perbarui informasi berikut:
   - Nama — Harus unik untuk akun Anda di AWS Wilayah.

- ProcessingInstanceCount— Jumlah instance yang digunakan untuk pemrosesan.
- ModelApprovalStatus- Untuk kenyamanan Anda.
- InputDataUrl— Amazon S3 dari data input.

8. Pilih Mulai.

- Untuk melihat detail eksekusi atau menghentikan eksekusi, pilih Lihat detail pada spanduk status.
  - Untuk menghentikan eksekusi, pilih Berhenti pada spanduk status.
  - Untuk melanjutkan eksekusi dari tempat itu dihentikan, pilih Lanjutkan pada spanduk status.

> **Note**
>
> Jika pipeline Anda gagal, spanduk status akan menampilkan status Gagal. Setelah memecahkan masalah langkah yang gagal, pilih Coba lagi pada spanduk status untuk melanjutkan menjalankan pipeline dari langkah itu.

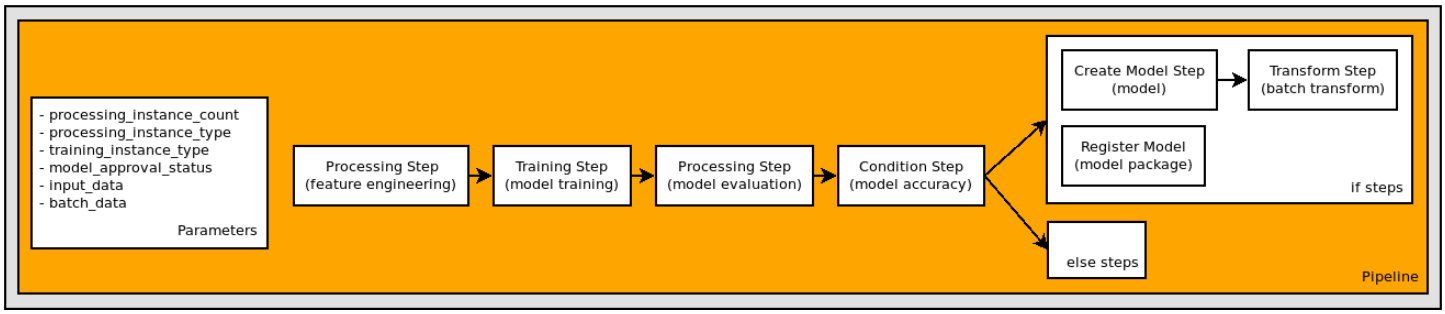
Untuk daftar model terdaftar, lihatOtomatiskan MLOP dengan Proyek SageMaker .

Lacak Silsilah Pipeline dari Pipeline SageMaker

Dalam tutorial ini, Anda menggunakan Amazon SageMaker Studio untuk melacak garis keturunan dari Amazon SageMaker MLPipeline.

Pipeline ini dibuat oleh notebook Orchestrating Jobs with Amazon SageMaker Model Building Pipelines di repositori contoh Amazon. SageMaker GitHub Untuk informasi rinci tentang bagaimana pipa dibuat, lihatMembuat Alur.

Pelacakan silsilah di Studio berpusat di sekitar grafik asiklik terarah (DAG). DAG mewakili langkah-langkah dalam pipa. Dari DAG Anda dapat melacak garis keturunan dari langkah apa pun ke langkah lain. Diagram berikut menampilkan langkah-langkah dalam pipa. Langkah-langkah ini muncul sebagai DAG di Studio.



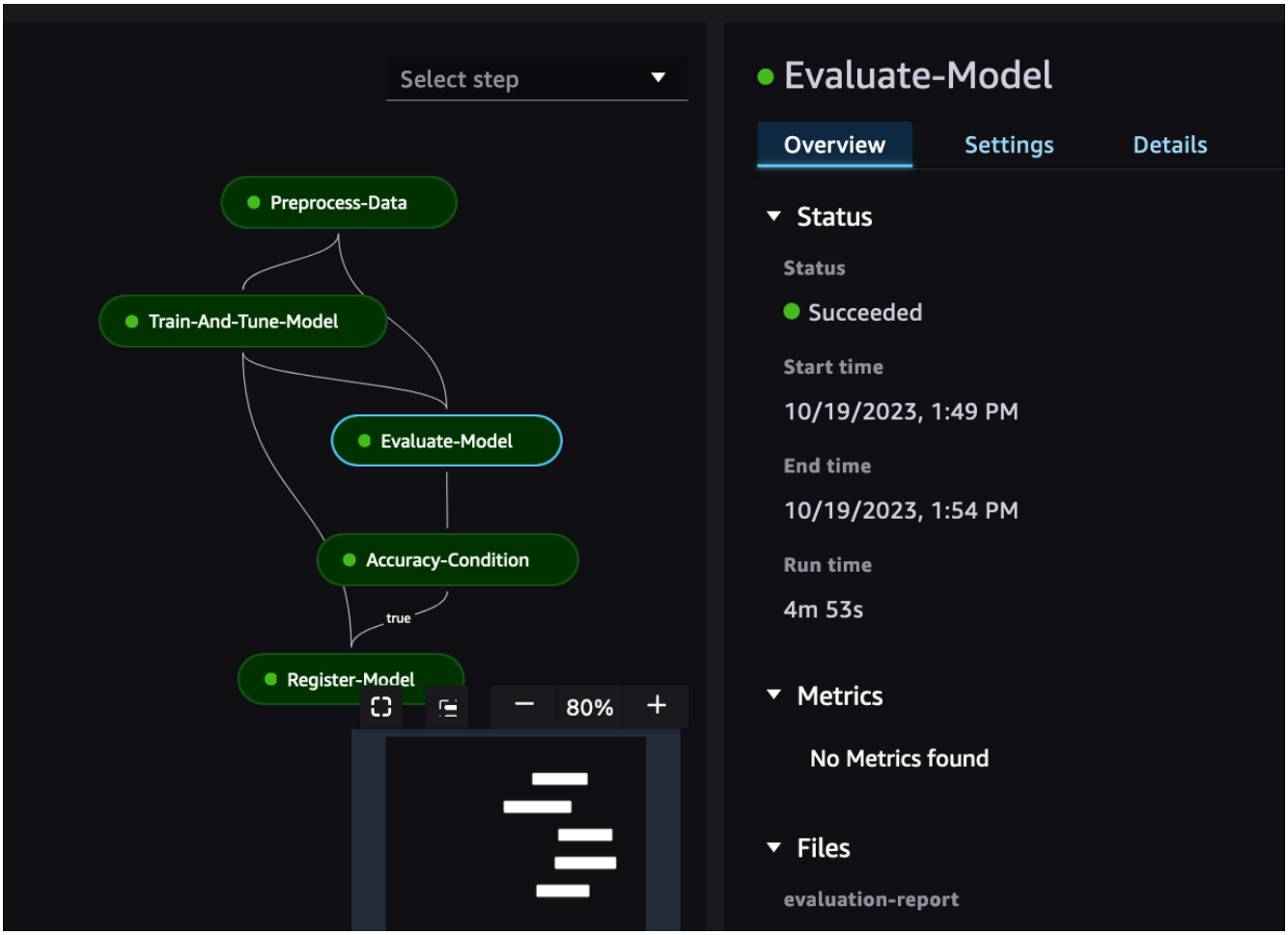
Untuk melacak garis keturunan pipeline di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

Untuk melacak garis keturunan pipa

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi sebelah kiri, pilih Pipelines.
3. (Opsional) Untuk memfilter daftar pipa berdasarkan nama, masukkan nama pipa lengkap atau sebagian di kolom pencarian.
4. Di kolom Nama, pilih nama pipeline untuk melihat detail tentang pipeline. Halaman Eksekusi pipeline membuka dan menampilkan daftar eksekusi pipeline.
5. Di kolom Nama tabel Eksekusi, pilih nama eksekusi pipeline untuk dilihat.
6. Di kanan atas halaman Eksekusi, pilih elipsis vertikal dan pilih Download pipeline definition (JSON). Anda dapat melihat file untuk melihat bagaimana grafik pipeline didefinisikan.
7. Gunakan ikon pengubahan ukuran di sisi kanan bawah grafik untuk memperbesar dan memperkecil grafik, menyesuaikan grafik ke layar, atau memperluas grafik ke layar penuh. Untuk fokus pada bagian tertentu dari grafik, Anda dapat memilih area kosong grafik dan menyeret grafik ke tengah area itu. Sisipan di sisi kanan bawah grafik menampilkan lokasi Anda dalam grafik.

   Citra berikut menunjukkan grafik alur dengan ikon inset dan mengubah ukuran. Selain itu, tab di sebelah kanan grafik berisi informasi terperinci tentang proses pipeline Anda.



8. Untuk melihat set data pelatihan, validasi, dan uji, selesaikan langkah-langkah berikut:

   a. Pilih langkah Pemrosesan dalam grafik pipeline Anda.

   b. Di tab Ikhtisar, di bagian File, temukan jalur Amazon S3 ke kumpulan data pelatihan, validasi, dan pengujian.

9. Untuk melihat artefak model Anda, selesaikan langkah-langkah berikut:

   a. Pilih langkah Pelatihan dalam grafik pipeline Anda.

   b. Di tab Ikhtisar, di bagian File, temukan jalur Amazon S3 ke artefak model.

10. Untuk menemukan paket model ARN, selesaikan langkah-langkah berikut:

    a. Pilih langkah model register (`RegisterModel`).

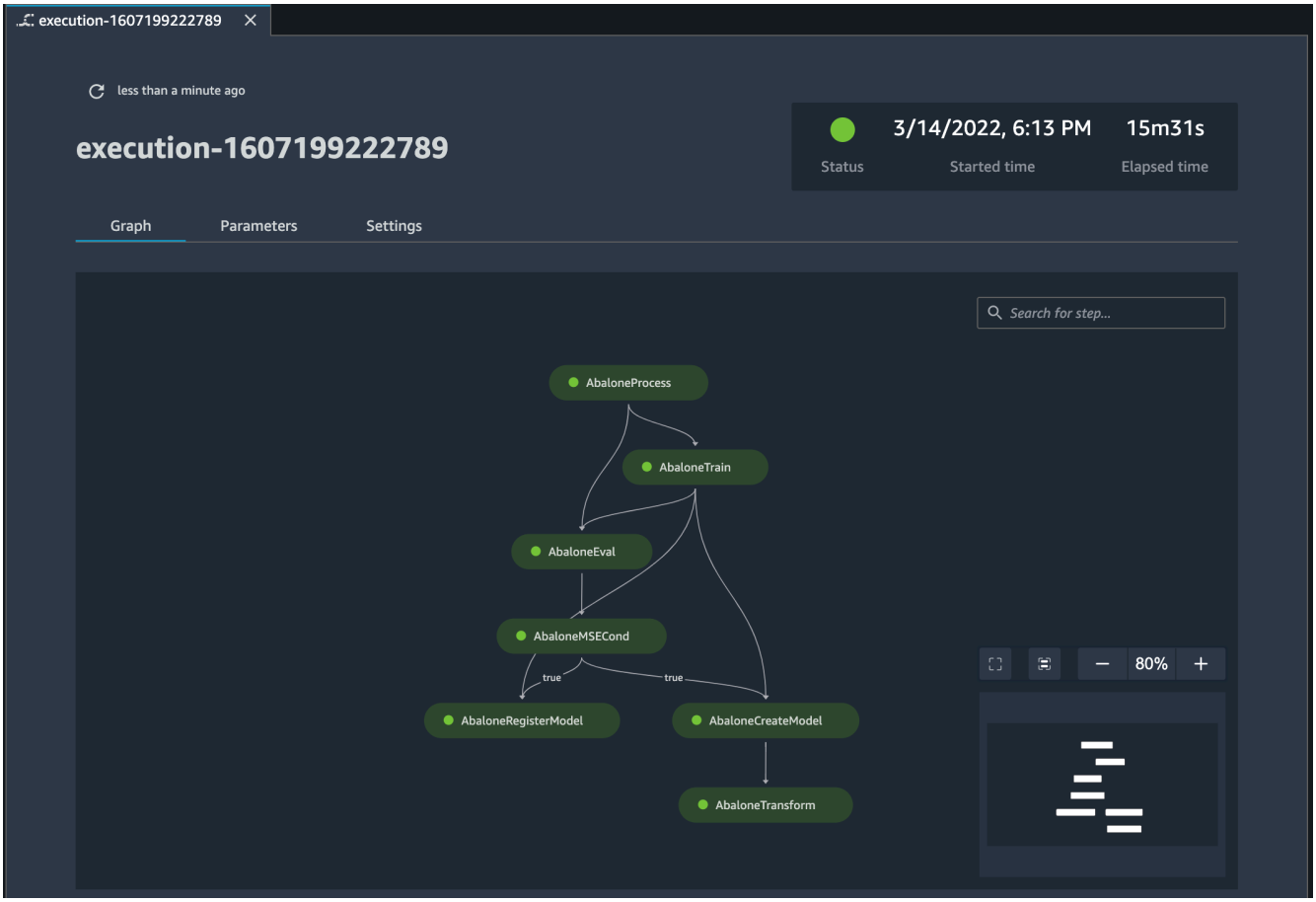
    b. Di tab Ikhtisar, di bagian File, temukan ARN dari paket model.

Studio Classic

Untuk melacak garis keturunan pipa

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di bilah sisi kiri Studio, pilih ikon Beranda ().
3. Di menu, pilih Pipelines.
4. Gunakan kotak Pencarian untuk memfilter daftar saluran pipa.
5. Pilih `AbalonePipeline` pipeline untuk melihat daftar eksekusi dan detail lainnya tentang pipeline.
6. Pilih ikon Property Inspector () di sidebar kanan untuk membuka panel TABLE PROPERTIES, di mana Anda dapat memilih properti mana yang akan dilihat.
7. Pilih tab Pengaturan dan kemudian pilih Unduh file definisi saluran pipa. Anda dapat melihat file untuk melihat bagaimana grafik pipeline didefinisikan.
8. Pada tab Eksekusi, pilih baris pertama dalam daftar eksekusi untuk melihat grafik eksekusi dan detail lainnya tentang eksekusi. Perhatikan bahwa grafik cocok dengan diagram yang ditampilkan di awal tutorial.

   Gunakan ikon pengubahan ukuran di sisi kanan bawah grafik untuk memperbesar dan memperkecil grafik, menyesuaikan grafik ke layar, atau memperluas grafik ke layar penuh. Untuk fokus pada bagian tertentu dari grafik, Anda dapat memilih area kosong grafik dan menyeret grafik ke tengah area itu. Sisipan di sisi kanan bawah grafik menampilkan lokasi Anda dalam grafik.



9. Pada tab Grafik, pilih `AbaloneProcess` langkah untuk melihat detail tentang langkah tersebut.
10. Temukan jalur Amazon S3 ke kumpulan data pelatihan, validasi, dan pengujian di tab Output, di bawah File.

> **Note**
>
> Untuk mendapatkan jalur lengkap, klik kanan jalur dan kemudian pilih Salin isi sel.

```
s3://sagemaker-eu-west-1-acct-id/sklearn-abalone-
process-2020-12-05-17-28-28-509/output/train
s3://sagemaker-eu-west-1-acct-id/sklearn-abalone-
process-2020-12-05-17-28-28-509/output/validation
s3://sagemaker-eu-west-1-acct-id/sklearn-abalone-
process-2020-12-05-17-28-28-509/output/test
```

11. Pilih `AbaloneTrain` langkahnya.

12. Temukan jalur Amazon S3 ke artefak model di tab Output, di bawah File:

```
s3://sagemaker-eu-west-1-acct-id/AbaloneTrain/pipelines-6locnsqz4bfu-
AbaloneTrain-NtfEpI0Ahu/output/model.tar.gz
```

13. Pilih `AbaloneRegisterModel` langkahnya.
14. Temukan ARN dari paket model di tab Output, di bawah File:

```
arn:aws:sagemaker:eu-west-1:acct-id:model-package/abalonemodelpackagegroupname/2
```

Orkestrasi Kubernetes

Anda dapat mengatur pekerjaan SageMaker pelatihan dan inferensi Anda dengan SageMaker Operator untuk Kubernetes dan Komponen untuk Pipelines Kubeflow. SageMaker SageMaker Operator untuk Kubernetes mempermudah pengembang dan ilmuwan data yang menggunakan Kubernetes untuk melatih, menyetel, dan menerapkan model machine learning (ML). SageMaker SageMaker Komponen untuk Pipelines Kubeflow memungkinkan Anda memindahkan pekerjaan pemrosesan dan pelatihan data dari klaster Kubernetes ke layanan terkelola yang dioptimalkan untuk SageMaker pembelajaran mesin.

Daftar Isi

SageMaker Operator Kubernetes

SageMaker Operator untuk Kubernetes mempermudah pengembang dan ilmuwan data yang menggunakan Kubernetes untuk melatih, menyetel, dan menerapkan model machine learning (ML). SageMaker Anda dapat menginstal SageMaker Operator ini di klaster Kubernetes Anda di Amazon Elastic Kubernetes Service (Amazon EKS SageMaker) untuk membuat pekerjaan secara native menggunakan API Kubernetes dan alat Kubernetes baris perintah seperti. `kubectl` Panduan ini menunjukkan cara menyiapkan dan menggunakan operator untuk menjalankan pelatihan model, tuning hyperparameter, atau inferensi (real-time dan batch) SageMaker dari cluster Kubernetes. Prosedur dan pedoman dalam Bab ini mengasumsikan bahwa Anda sudah familiar dengan Kubernetes dan perintah-perintah dasarnya.

> **Important**
>
> Kami menghentikan pengembangan dan dukungan teknis dari versi asli SageMaker Operator untuk Kubernetes.
> Jika saat ini Anda menggunakan SageMaker Operator untuk Kubernetes versi `v1.2.2` atau di bawah ini, kami sarankan untuk memigrasikan sumber daya Anda ke pengontrol layanan ACK untuk Amazon. SageMaker Pengontrol layanan ACK adalah generasi baru SageMaker Operator untuk Kubernetes berdasarkan AWSController for Kubernetes (ACK).
> Untuk informasi tentang langkah-langkah migrasi, lihatMigrasikan sumber daya ke Operator terbaru.
> Untuk jawaban atas pertanyaan umum di akhir dukungan versi asli SageMaker Operator untuk Kubernetes, lihat Mengumumkan Berakhirnya Support Versi Original SageMaker Operator untuk Kubernetes

> **Note**
>
> Tidak ada biaya tambahan untuk menggunakan operator ini. Anda dikenakan biaya untuk SageMaker sumber daya apa pun yang Anda gunakan melalui operator ini.

Apa itu operator?

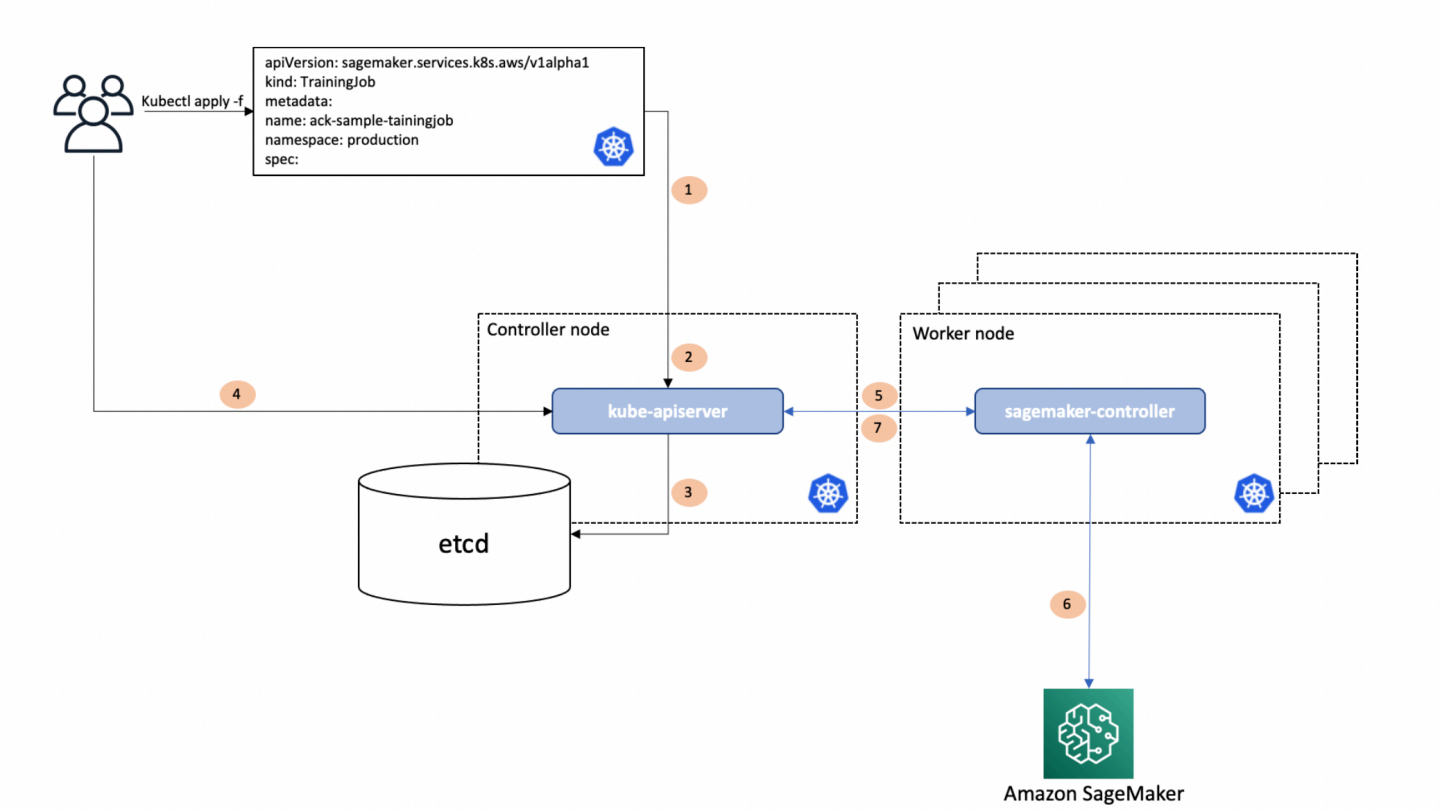
Operator Kubernetes adalah pengontrol aplikasi yang mengelola aplikasi atas nama pengguna Kubernetes. Pengontrol bidang kontrol mencakup berbagai loop kontrol yang mendengarkan manajer negara pusat (ETCD) untuk mengatur keadaan aplikasi yang mereka kendalikan. Contoh aplikasi tersebut termasuk C loud-controller-manager dan`kube-controller-manager`. Operator biasanya menyediakan abstraksi tingkat yang lebih tinggi daripada API Kubernetes mentah, sehingga memudahkan pengguna untuk menerapkan dan mengelola aplikasi. Untuk menambahkan kemampuan baru ke Kubernetes, pengembang dapat memperluas API Kubernetes dengan membuat sumber daya khusus yang berisi logika dan komponen khusus aplikasi atau domain khusus mereka. Operator di Kubernetes memungkinkan pengguna untuk memanggil sumber daya kustom ini secara native dan mengotomatiskan alur kerja terkait.

Bagaimana cara kerja AWS Controllers for Kubernetes (ACK)?

SageMaker Operator untuk Kubernetes memungkinkan Anda mengelola pekerjaan SageMaker dari klaster Kubernetes Anda. Versi terbaru SageMaker Operator untuk Kubernetes didasarkan pada

AWS Controller for Kubernetes (ACK). ACK mencakup runtime pengontrol umum, generator kode, dan satu set pengontrol AWS khusus layanan, salah satunya adalah pengontrol. SageMaker

Diagram berikut mengimagekan cara kerja ACK.



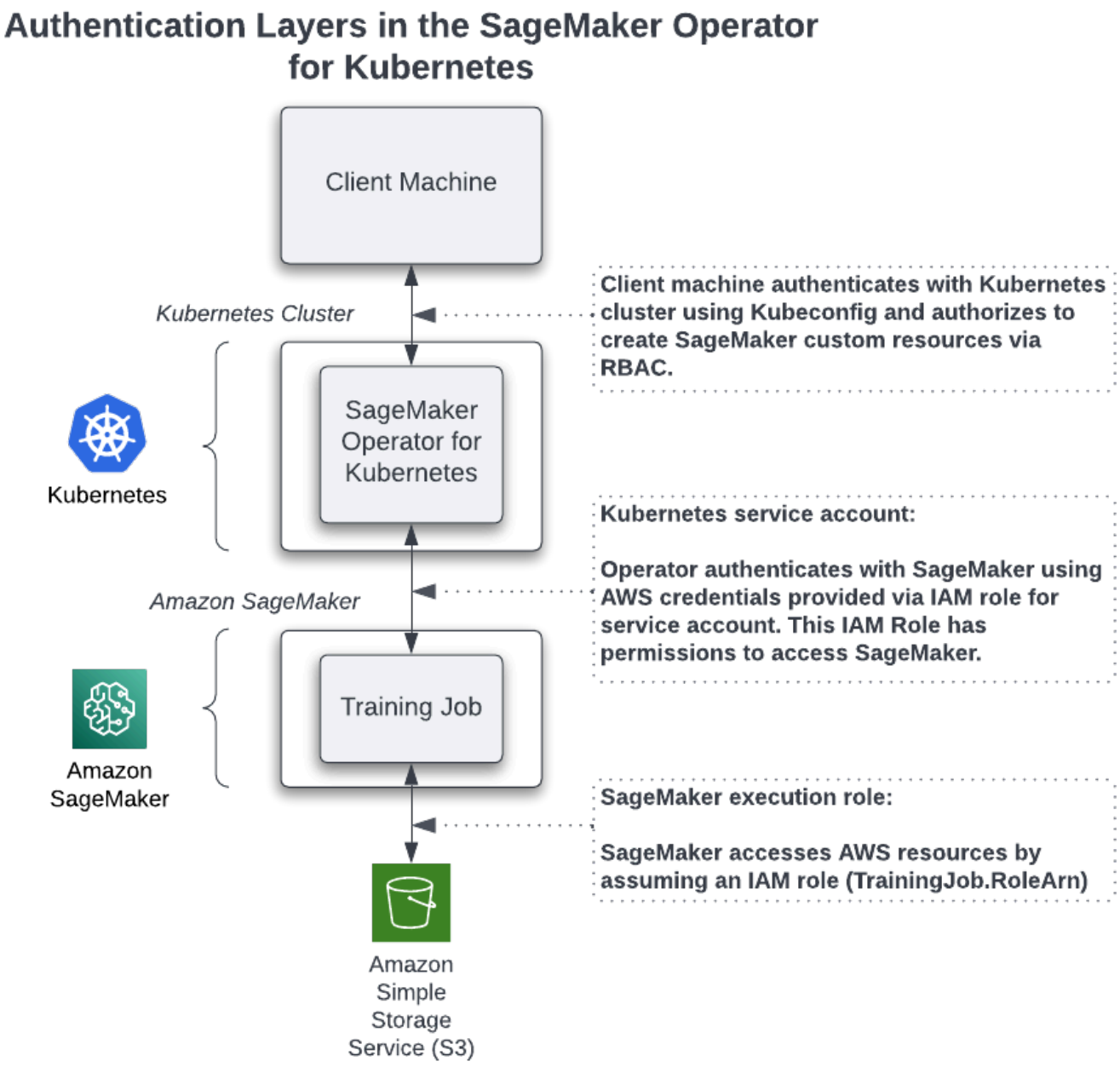
Dalam diagram ini, pengguna Kubernetes ingin menjalankan pelatihan model SageMaker dari dalam klaster Kubernetes menggunakan API Kubernetes. Pengguna mengeluarkan panggilan kekubectl apply, meneruskan file yang menjelaskan sumber daya kustom Kubernetes yang menjelaskan pekerjaan pelatihan. SageMaker kubectl applymeneruskan file ini, yang disebut manifes, ke server API Kubernetes yang berjalan di node controller Kubernetes (Langkah 1 dalam diagram alur kerja). Server API Kubernetes menerima manifes dengan spesifikasi tugas SageMaker pelatihan dan menentukan apakah pengguna memiliki izin untuk membuat jenis sumber daya kustom*sageMaker.services.k8s.aws/TrainingJob*, dan apakah sumber daya kustom diformat dengan benar (Langkah 2). Jika pengguna diotorisasi dan sumber daya kustom valid, server API Kubernetes menulis (Langkah 3) sumber daya kustom ke penyimpanan data etcd-nya dan kemudian merespons kembali (Langkah 4) kepada pengguna bahwa sumber daya kustom telah dibuat. SageMaker Controller, yang berjalan pada node pekerja Kubernetes dalam konteks Pod Kubernetes normal, diberi tahu (Langkah 5) bahwa sebuah sumber daya kustom baru telah dibuat. sageMaker.services.k8s.aws/TrainingJob SageMaker Pengontrol kemudian berkomunikasi

(Langkah 6) dengan SageMaker API, memanggil SageMaker `CreateTrainingJob` API untuk membuat pekerjaan pelatihan diAWS. Setelah berkomunikasi dengan SageMaker API, SageMaker controller memanggil server Kubernetes API untuk memperbarui (Langkah 7) status sumber daya kustom dengan informasi yang diterimanya. SageMaker Oleh karena itu, SageMaker pengontrol memberikan informasi yang sama kepada pengembang yang akan mereka terima menggunakan AWS SDK.

Gambaran umum izin

Operator mengakses SageMaker sumber daya atas nama Anda. Peran IAM yang diasumsikan operator untuk berinteraksi dengan AWS sumber daya berbeda dari kredenal yang Anda gunakan untuk mengakses klaster Kubernetes. Peran ini juga berbeda dari peran yang AWS diasumsikan saat menjalankan pekerjaan pembelajaran mesin Anda.

Gambar berikut menjelaskan berbagai lapisan otentikasi.



SageMaker Operator Terbaru untuk Kubernetes

Bagian ini didasarkan pada versi terbaru SageMaker Operator untuk Kubernetes menggunakan AWS Controllers for Kubernetes (ACK).

> **Important**
>
> Jika saat ini Anda menggunakan SageMaker Operator untuk Kubernetes versi `v1.2.2` atau di bawah ini, kami sarankan untuk memigrasikan sumber daya Anda ke pengontrol layanan ACK untuk Amazon. SageMaker Pengontrol layanan ACK adalah generasi baru SageMaker Operator untuk Kubernetes berdasarkan AWSController for Kubernetes (ACK).
> Untuk informasi tentang langkah-langkah migrasi, lihatMigrasikan sumber daya ke Operator terbaru.
> Untuk jawaban atas pertanyaan umum di akhir dukungan versi asli SageMaker Operator untuk Kubernetes, lihat Mengumumkan Berakhirnya Support Versi Original SageMaker Operator untuk Kubernetes

Versi terbaru dari SageMaker Operator untuk Kubernetes didasarkan pada AWSControllers for Kubernetes (ACK), sebuah framework untuk membangun kubernetes custom controllers dimana setiap controller berkomunikasi dengan service API. AWS Pengontrol ini memungkinkan pengguna Kubernetes untuk menyediakan AWS sumber daya seperti database atau antrian pesan menggunakan Kubernetes API.

Gunakan langkah-langkah berikut untuk menginstal dan menggunakan ACK untuk melatih, menyetel, dan menerapkan model pembelajaran mesin dengan Amazon SageMaker.

Daftar Isi

Instal SageMaker Operator untuk Kubernetes

Untuk menyiapkan versi terbaru SageMaker Operator untuk Kubernetes yang tersedia, lihat bagian Setup di Machine Learning dengan ACK Controller. SageMaker

Menggunakan SageMaker Operator untuk Kubernetes

Untuk tutorial tentang cara melatih model pembelajaran mesin dengan pengontrol layanan ACK untuk Amazon SageMaker menggunakan Amazon EKS, lihat Machine Learning with the ACK SageMaker Controller.

Untuk contoh penskalaan otomatis, lihat Menskalakan SageMaker Beban Kerja dengan Application Auto Scaling

Referensi

Lihat juga pengontrol layanan ACK untuk SageMaker GitHub repositori Amazon atau baca AWS Controllers for Kubernetes Documentation.

SageMaker Operator Lama untuk Kubernetes

Bagian ini didasarkan pada versi asli SageMaker Operator untuk Kubernetes.

> **Important**
>
> Kami menghentikan pengembangan dan dukungan teknis dari versi asli SageMaker Operator untuk Kubernetes.
> Jika saat ini Anda menggunakan SageMaker Operator untuk Kubernetes versi `v1.2.2` atau di bawah ini, kami sarankan untuk memigrasikan sumber daya Anda ke pengontrol layanan ACK untuk Amazon. SageMaker Pengontrol layanan ACK adalah generasi baru SageMaker Operator untuk Kubernetes berdasarkan AWSController for Kubernetes (ACK).
> Untuk informasi tentang langkah-langkah migrasi, lihatMigrasikan sumber daya ke Operator terbaru.
> Untuk jawaban atas pertanyaan umum di akhir dukungan versi asli SageMaker Operator untuk Kubernetes, lihat Mengumumkan Berakhirnya Support Versi Original SageMaker Operator untuk Kubernetes

Daftar Isi

Instal SageMaker Operator untuk Kubernetes

Gunakan langkah-langkah berikut untuk menginstal dan menggunakan SageMaker Operator untuk Kubernetes untuk melatih, menyetel, dan menerapkan model pembelajaran mesin dengan Amazon. SageMaker

Daftar Isi

Penyiapan berbasis peran IAM dan penyebaran operator

Bagian berikut menjelaskan langkah-langkah untuk mengatur dan menyebarkan versi asli operator.

> **Warning**
>
> Pengingat: Langkah-langkah berikut tidak menginstal versi terbaru SageMaker Operator untuk Kubernetes. Untuk menginstal SageMaker Operator berbasis ACK baru untuk Kubernetes, lihat. SageMaker Operator Terbaru untuk Kubernetes

Prasyarat

Panduan ini mengasumsikan bahwa Anda telah menyelesaikan prasyarat berikut:

- Instal alat-alat berikut pada mesin klien yang digunakan untuk mengakses klaster Kubernetes Anda:
  - `kubectl`Versi 1.13 atau yang lebih baru. Gunakan `kubectl` versi yang ada dalam satu versi minor bidang kontrol klaster Amazon EKS Anda. Sebagai contoh, `kubectl` klien 1,13 bekerja dengan klaster Kubernetes 1.13 dan 1.14. OpenID Connect (OIDC) tidak didukung dalam versi lebih awal dari 1.13.
  - `eksctl`Versi 0.7.0 atau lebih baru
  - AWSCLI Versi 1.16.232 atau yang lebih baru
  - (opsional) Helm Versi 3.0 atau lebih baru
  - aws-iam-authenticator
- Memiliki izin IAM untuk membuat peran dan melampirkan kebijakan ke peran.
- Membuat cluster Kubernetes untuk menjalankan operator. Seharusnya Kubernetes versi 1.13 atau 1.14. Untuk pembuatan klaster otomatis menggunakan`eksctl`, lihat Memulai dengan eksctl. Dibutuhkan 20-30 menit untuk menyediakan cluster.

Penerapan cakupan cluster

Sebelum Anda dapat menerapkan operator menggunakan peran IAM, kaitkan Penyedia Identitas OpenID Connect (OIDC) (iDP) dengan peran Anda untuk mengautentikasi dengan layanan IAM.

Buat penyedia OIDC untuk klaster Anda

Petunjuk berikut menunjukkan cara membuat dan mengaitkan penyedia OIDC dengan kluster Amazon EKS Anda.

1. Tetapkan variabel lokal `CLUSTER_NAME` dan `AWS_REGION` lingkungan sebagai berikut:

   ```
   # Set the Region and cluster
   export CLUSTER_NAME="<your cluster name>"
   export AWS_REGION="<your region>"
   ```

2. Gunakan perintah berikut untuk menghubungkan penyedia OIDC dengan klaster Anda. Untuk informasi selengkapnya, lihat Mengaktifkan Peran IAM untuk Akun Layanan di Cluster Anda.

   ```
   eksctl utils associate-iam-oidc-provider --cluster ${CLUSTER_NAME} \
         --region ${AWS_REGION} --approve
   ```

   Outputnya akan terlihat seperti berikut:

   ```
   [_]  eksctl version 0.10.1
     [_]  using region us-east-1
     [_]  IAM OpenID Connect provider is associated with cluster "my-cluster" in "us-
   east-1"
   ```

Sekarang klaster memiliki penyedia identitas OIDC, Anda dapat membuat peran dan memberikan ServiceAccount izin Kubernetes untuk mengambil peran tersebut.

Dapatkan ID OIDC

Untuk mengatur ServiceAccount, dapatkan URL penerbit OIDC menggunakan perintah berikut:

```
aws eks describe-cluster --name ${CLUSTER_NAME} --region ${AWS_REGION} \
      --query cluster.identity.oidc.issuer --output text
```

Perintah mengembalikan URL seperti berikut:

```
https://oidc.eks.${AWS_REGION}.amazonaws.com/id/D48675832CA65BD10A532F597OIDCID
```

Di URL ini, nilainya D48675832CA65BD10A532F597OIDCID adalah ID OIDC. ID OIDC untuk cluster Anda berbeda. Anda memerlukan nilai ID OIDC ini untuk membuat peran.

Jika output AndaNone, itu berarti versi klien Anda sudah tua. Untuk mengatasinya, jalankan perintah berikut:

```
aws eks describe-cluster --region ${AWS_REGION} --query cluster --name ${CLUSTER_NAME}
 --output text | grep OIDC
```

URL OIDC dikembalikan sebagai berikut:

```
OIDC https://oidc.eks.us-east-1.amazonaws.com/id/D48675832CA65BD10A532F597OIDCID
```

Membuat peran IAM

1. Buat file bernama `trust.json` dan masukkan blok kode hubungan kepercayaan berikut ke dalamnya. Pastikan untuk mengganti all<`OIDC ID`>,<`AWS account number`>, dan <`EKS Cluster region`> placeholder dengan nilai yang sesuai dengan cluster Anda.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
         {
           "Effect": "Allow",
           "Principal": {
             "Federated": "arn:aws:iam::<AWS account number>:oidc-provider/
   oidc.eks.<EKS Cluster region>.amazonaws.com/id/<OIDC ID>"
           },
           "Action": "sts:AssumeRoleWithWebIdentity",
           "Condition": {
             "StringEquals": {
               "oidc.eks.<EKS Cluster region>.amazonaws.com/id/<OIDC ID>:aud":
    "sts.amazonaws.com",
               "oidc.eks.<EKS Cluster region>.amazonaws.com/id/<OIDC ID>:sub":
    "system:serviceaccount:sagemaker-k8s-operator-system:sagemaker-k8s-operator-
   default"
             }
           }
         }
   ```

```
    ]
  }
```

2. Jalankan perintah berikut untuk membuat peran dengan hubungan kepercayaan yang ditentukan dalam`trust.json`. Peran ini memungkinkan kluster Amazon EKS untuk mendapatkan dan menyegarkan kredensil dari IAM.

```
aws iam create-role --region ${AWS_REGION} --role-name <role name> --assume-role-
policy-document file://trust.json --output=text
```

Outputnya akan terlihat seperti berikut:

```
ROLE    arn:aws:iam::123456789012:role/my-role 2019-11-22T21:46:10Z    /
 ABCDEFSFODNN7EXAMPLE   my-role
ASSUMEROLEPOLICYDOCUMENT        2012-10-17
STATEMENT       sts:AssumeRoleWithWebIdentity   Allow
STRINGEQUALS    sts.amazonaws.com       system:serviceaccount:sagemaker-k8s-
operator-system:sagemaker-k8s-operator-default
PRINCIPAL       arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-
east-1.amazonaws.com/id/
```

PerhatikanROLE ARN; Anda meneruskan nilai ini ke operator Anda.

Melampirkan AmazonSageMakerFullAccess kebijakan pada peran tersebut

Untuk memberikan akses peran SageMaker, lampirkan AmazonSageMakerFullAccesskebijakan. Jika Anda ingin membatasi izin ke operator, Anda dapat membuat kebijakan kustom Anda sendiri dan melampirkannya.

Untuk melampirkan`AmazonSageMakerFullAccess`, jalankan perintah berikut:

```
aws iam attach-role-policy --role-name <role name>  --policy-arn
 arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

Kubernetes ServiceAccount `sagemaker-k8s-operator-default` harus memiliki izin. `AmazonSageMakerFullAccess` Konfirmasikan ini saat Anda menginstal operator.

Terapkan operator

Saat menerapkan operator, Anda dapat menggunakan file YAMM atau bagan Helm.

Menyebarkan operator menggunakan YAMAL

Cara ini paling mudah untuk menggunakan operator Anda. Prosesnya adalah sebagai berikut:

1. Unduh skrip penginstal menggunakan perintah berikut:

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/release/rolebased/installer.yaml
```

2. Edit `installer.yaml` file yang akan digantieks.amazonaws.com/role-arn. Ganti ARN di sini dengan Amazon Resource Name (ARN) untuk peran berbasis OIDC yang Anda buat.
3. Gunakan perintah berikut untuk men-deploy klaster:

```
kubectl apply -f installer.yaml
```

Menerapkan operator menggunakan Helm Charts

Gunakan Bagan Helm yang disediakan untuk menginstal operator.

1. Kloning direktori penginstal Helm menggunakan perintah berikut:

```
git clone https://github.com/aws/amazon-sagemaker-operator-for-k8s.git
```

2. Arahkan ke `amazon-sagemaker-operator-for-k8s/hack/charts/installer` folder. Edit `rolebased/values.yaml` file, yang mencakup parameter tingkat tinggi untuk bagan. Ganti peran ARN di sini dengan Amazon Resource Name (ARN) untuk peran berbasis OIDC yang Anda buat.
3. Instal Bagan Helm menggunakan perintah berikut:

```
kubectl create namespace sagemaker-k8s-operator-system
  helm install --namespace sagemaker-k8s-operator-system sagemaker-operator
 rolebased/
```

   Jika Anda memutuskan untuk menginstal operator ke namespace selain yang ditentukan, Anda perlu menyesuaikan namespace yang ditentukan dalam file peran IAM agar sesuai. `trust.json`
4. Setelah beberapa saat, bagan diinstal dengan nama yang dihasilkan secara acak. Verifikasi bahwa penginstalan berhasil dengan menjalankan perintah berikut:

```
helm ls
```

Outputnya akan terlihat seperti berikut:

```
NAME                    NAMESPACE                       REVISION        UPDATED
                              STATUS          CHART                           APP
 VERSION
  sagemaker-operator      sagemaker-k8s-operator-system   1
  2019-11-20 23:14:59.6777082 +0000 UTC   deployed        sagemaker-k8s-
operator-0.1.0
```

Verifikasi penyebaran operator

1. Anda harus dapat melihat Definisi Sumber Daya SageMaker Kustom (CRD) untuk setiap operator yang diterapkan ke cluster Anda dengan menjalankan perintah berikut:

   ```
   kubectl get crd | grep sagemaker
   ```

   Outputnya akan terlihat seperti berikut:

   ```
   batchtransformjobs.sagemaker.aws.amazon.com         2019-11-20T17:12:34Z
   endpointconfigs.sagemaker.aws.amazon.com            2019-11-20T17:12:34Z
   hostingdeployments.sagemaker.aws.amazon.com         2019-11-20T17:12:34Z
   hyperparametertuningjobs.sagemaker.aws.amazon.com   2019-11-20T17:12:34Z
   models.sagemaker.aws.amazon.com                     2019-11-20T17:12:34Z
   trainingjobs.sagemaker.aws.amazon.com               2019-11-20T17:12:34Z
   ```

2. Pastikan pod operator berhasil berjalan. Gunakan perintah berikut untuk mencantumkan semua pod:

   ```
   kubectl -n sagemaker-k8s-operator-system get pods
   ```

   Anda akan melihat sebuah pod bernama `sagemaker-k8s-operator-controller-manager-*****` di namespace `sagemaker-k8s-operator-system` sebagai berikut:

   ```
   NAME                                                  READY   STATUS
    RESTARTS   AGE
   ```

```
sagemaker-k8s-operator-controller-manager-12345678-r8abc     2/2     Running   0
       23s
```

Penerapan dengan cakupan ruang nama

Anda memiliki opsi untuk menginstal operator Anda dalam lingkup namespace Kubernetes individual. Dalam mode ini, pengontrol hanya memantau dan merekonsiliasi sumber daya dengan SageMaker jika sumber daya dibuat dalam namespace itu. Ini memungkinkan kontrol yang lebih halus atas pengontrol mana yang mengelola sumber daya mana. Ini berguna untuk menyebarkan ke beberapa AWS akun atau mengontrol pengguna mana yang memiliki akses ke pekerjaan tertentu.

Panduan ini menguraikan cara menginstal operator ke namespace tertentu yang telah ditentukan sebelumnya. Untuk menerapkan controller ke namespace kedua, ikuti panduan dari awal hingga akhir dan ubah namespace di setiap langkah.

Buat penyedia OIDC untuk klaster Amazon EKS EKS

Petunjuk berikut menunjukkan cara membuat dan mengaitkan penyedia OIDC dengan kluster Amazon EKS Anda.

1. Tetapkan variabel lokal `CLUSTER_NAME` dan `AWS_REGION` lingkungan sebagai berikut:

   ```
   # Set the Region and cluster
   export CLUSTER_NAME="<your cluster name>"
   export AWS_REGION="<your region>"
   ```

2. Gunakan perintah berikut untuk menghubungkan penyedia OIDC dengan klaster Anda. Untuk informasi selengkapnya, lihat Mengaktifkan Peran IAM untuk Akun Layanan di Cluster Anda.

   ```
   eksctl utils associate-iam-oidc-provider --cluster ${CLUSTER_NAME} \
         --region ${AWS_REGION} --approve
   ```

   Outputnya akan terlihat seperti berikut:

   ```
   [_]  eksctl version 0.10.1
     [_]  using region us-east-1
     [_]  IAM OpenID Connect provider is associated with cluster "my-cluster" in "us-
   east-1"
   ```

Sekarang klaster memiliki penyedia identitas OIDC, buat peran dan berikan ServiceAccount izin Kubernetes untuk mengambil peran tersebut.

Dapatkan ID OIDC Anda

Untuk mengatur ServiceAccount, pertama-tama dapatkan URL penerbit OpenID Connect menggunakan perintah berikut:

```
aws eks describe-cluster --name ${CLUSTER_NAME} --region ${AWS_REGION} \
      --query cluster.identity.oidc.issuer --output text
```

Perintah mengembalikan URL seperti berikut:

```
https://oidc.eks.${AWS_REGION}.amazonaws.com/id/D48675832CA65BD10A532F597OIDCID
```

Dalam URL ini, nilai D48675832CA65BD10A532F597OIDCID adalah ID OIDC. ID OIDC untuk cluster Anda berbeda. Anda memerlukan nilai ID OIDC ini untuk membuat peran.

Jika output Anda`None`, itu berarti versi klien Anda sudah tua. Untuk mengatasinya, jalankan perintah berikut:

```
aws eks describe-cluster --region ${AWS_REGION} --query cluster --name ${CLUSTER_NAME}
 --output text | grep OIDC
```

URL OIDC dikembalikan sebagai berikut:

```
OIDC https://oidc.eks.us-east-1.amazonaws.com/id/D48675832CA65BD10A532F597OIDCID
```

Membuat peran IAM Anda

1. Buat file bernama `trust.json` dan masukkan blok kode hubungan kepercayaan berikut ke dalamnya. Pastikan untuk mengganti semua`<OIDC ID>`,, `<AWS account number>``<EKS Cluster region>`, dan `<Namespace>` placeholder dengan nilai yang sesuai dengan cluster Anda. Untuk keperluan panduan ini, `my-namespace` digunakan untuk `<Namespace>` nilai.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
         {
           "Effect": "Allow",
           "Principal": {
   ```

```
          "Federated": "arn:aws:iam::<AWS account number>:oidc-provider/
oidc.eks.<EKS Cluster region>.amazonaws.com/id/<OIDC ID>"
        },
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {
          "StringEquals": {
            "oidc.eks.<EKS Cluster region>.amazonaws.com/id/<OIDC ID>:aud":
 "sts.amazonaws.com",
            "oidc.eks.<EKS Cluster region>.amazonaws.com/id/<OIDC ID>:sub":
 "system:serviceaccount:<Namespace>:sagemaker-k8s-operator-default"
          }
        }
      }
    ]
  }
```

2. Jalankan perintah berikut untuk membuat peran dengan hubungan kepercayaan yang ditentukan dalam`trust.json`. Peran ini memungkinkan kluster Amazon EKS untuk mendapatkan dan menyegarkan kredensil dari IAM.

   ```
   aws iam create-role --region ${AWS_REGION} --role-name <role name> --assume-role-
   policy-document file://trust.json --output=text
   ```

   Outputnya akan terlihat seperti berikut:

   ```
   ROLE    arn:aws:iam::123456789012:role/my-role 2019-11-22T21:46:10Z    /
    ABCDEFSFODNN7EXAMPLE   my-role
     ASSUMEROLEPOLICYDOCUMENT        2012-10-17
     STATEMENT       sts:AssumeRoleWithWebIdentity   Allow
     STRINGEQUALS    sts.amazonaws.com       system:serviceaccount:my-
   namespace:sagemaker-k8s-operator-default
     PRINCIPAL       arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-
   east-1.amazonaws.com/id/
   ```

PerhatikanROLE ARN. Anda meneruskan nilai ini ke operator Anda.

Lampirkan AmazonSageMakerFullAccess kebijakan ke peran Anda

Untuk memberikan akses peran SageMaker, lampirkan AmazonSageMakerFullAccesskebijakan. Jika Anda ingin membatasi izin ke operator, Anda dapat membuat kebijakan kustom Anda sendiri dan melampirkannya.

Untuk melampirkan`AmazonSageMakerFullAccess`, jalankan perintah berikut:

```
aws iam attach-role-policy --role-name <role name>  --policy-arn
 arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

Kubernetes ServiceAccount `sagemaker-k8s-operator-default` harus memiliki izin. `AmazonSageMakerFullAccess` Konfirmasikan ini saat Anda menginstal operator.

Terapkan operator ke namespace Anda

Saat menerapkan operator, Anda dapat menggunakan file YAMM atau bagan Helm.

Menerapkan operator ke namespace Anda menggunakan YAMAL

Ada dua bagian untuk menerapkan operator dalam lingkup namespace. Yang pertama adalah set CRD yang diinstal pada tingkat cluster. Definisi sumber daya ini hanya perlu diinstal satu kali per klaster Kubernetes. Bagian kedua adalah izin operator dan penyebaran itu sendiri.

Jika Anda belum menginstal CRD ke dalam cluster, terapkan CRD installer YAMM menggunakan perintah berikut:

```
kubectl apply -f https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-
k8s/master/release/rolebased/namespaced/crd.yaml
```

Untuk menginstal operator ke cluster:

1. Unduh penginstal operator YAM menggunakan perintah berikut:

   ```
   wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
   master/release/rolebased/namespaced/operator.yaml
   ```

2. Perbarui penginstal YAMAL untuk menempatkan sumber daya ke dalam namespace yang Anda tentukan menggunakan perintah berikut:

   ```
   sed -i -e 's/PLACEHOLDER-NAMESPACE/<YOUR NAMESPACE>/g' operator.yaml
   ```

3. Edit `operator.yaml` file untuk menempatkan sumber daya ke dalam file Andaeks.amazonaws.com/role-arn. Ganti ARN di sini dengan Amazon Resource Name (ARN) untuk peran berbasis OIDC yang Anda buat.
4. Gunakan perintah berikut untuk men-deploy klaster:

```
kubectl apply -f operator.yaml
```

Terapkan operator ke namespace Anda menggunakan Helm Charts

Ada dua bagian yang diperlukan untuk menyebarkan operator dalam lingkup namespace. Yang pertama adalah set CRD yang diinstal pada tingkat cluster. Definisi sumber daya ini hanya perlu diinstal satu kali per klaster Kubernetes. Bagian kedua adalah izin operator dan penyebaran itu sendiri. Saat menggunakan Helm Charts, Anda harus terlebih dahulu membuat namespace menggunakan. `kubectl`

1. Kloning direktori penginstal Helm menggunakan perintah berikut:

   ```
   git clone https://github.com/aws/amazon-sagemaker-operator-for-k8s.git
   ```

2. Arahkan ke `amazon-sagemaker-operator-for-k8s/hack/charts/installer/namespaced` folder. Edit `rolebased/values.yaml` file, yang mencakup parameter tingkat tinggi untuk bagan. Ganti peran ARN di sini dengan Amazon Resource Name (ARN) untuk peran berbasis OIDC yang Anda buat.
3. Instal Bagan Helm menggunakan perintah berikut:

   ```
   helm install crds crd_chart/
   ```

4. Buat namespace yang diperlukan dan instal operator menggunakan perintah berikut:

   ```
   kubectl create namespace <namespace>
   helm install --n <namespace> op operator_chart/
   ```

5. Setelah beberapa saat, bagan dipasang dengan nama`sagemaker-operator`. Verifikasi bahwa penginstalan berhasil dengan menjalankan perintah berikut:

   ```
   helm ls
   ```

   Outputnya akan terlihat seperti berikut:

   ```
   NAME                    NAMESPACE                       REVISION        UPDATED
                                 STATUS          CHART                           APP
    VERSION
   ```

```
sagemaker-operator      my-namespace                    1               2019-11-20
 23:14:59.6777082 +0000 UTC   deployed        sagemaker-k8s-operator-0.1.0
```

Verifikasi penyebaran operator ke namespace Anda

1. Anda harus dapat melihat Definisi Sumber Daya SageMaker Kustom (CRD) untuk setiap operator yang diterapkan ke cluster Anda dengan menjalankan perintah berikut:

   ```
   kubectl get crd | grep sagemaker
   ```

   Outputnya akan terlihat seperti berikut:

   ```
   batchtransformjobs.sagemaker.aws.amazon.com         2019-11-20T17:12:34Z
   endpointconfigs.sagemaker.aws.amazon.com            2019-11-20T17:12:34Z
   hostingdeployments.sagemaker.aws.amazon.com         2019-11-20T17:12:34Z
   hyperparametertuningjobs.sagemaker.aws.amazon.com   2019-11-20T17:12:34Z
   models.sagemaker.aws.amazon.com                     2019-11-20T17:12:34Z
   trainingjobs.sagemaker.aws.amazon.com               2019-11-20T17:12:34Z
   ```

2. Pastikan pod operator berhasil berjalan. Gunakan perintah berikut untuk mencantumkan semua pod:

   ```
   kubectl -n my-namespace get pods
   ```

   Anda akan melihat sebuah pod bernama `sagemaker-k8s-operator-controller-manager-*****` di namespace `my-namespace` sebagai berikut:

   ```
   NAME                                                     READY   STATUS
    RESTARTS   AGE
   sagemaker-k8s-operator-controller-manager-12345678-r8abc     2/2     Running   0
          23s
   ```

Instal **`kubectl`** plugin SageMaker log

Sebagai bagian dari SageMaker Operator untuk Kubernetes, Anda dapat menggunakan plugin untuk. `smlogs` `kubectl` Hal ini memungkinkan SageMaker CloudWatch log untuk dialirkan dengan`kubectl`. `kubectl`harus diinstal ke PATH Anda. Perintah berikut menempatkan biner di

`sagemaker-k8s-bin` direktori di direktori home Anda, dan menambahkan direktori itu ke direktori Anda`PATH`.

```
export os="linux"

wget https://amazon-sagemaker-operator-for-k8s-us-east-1.s3.amazonaws.com/kubectl-
smlogs-plugin/v1/${os}.amd64.tar.gz
tar xvzf ${os}.amd64.tar.gz

# Move binaries to a directory in your homedir.
mkdir ~/sagemaker-k8s-bin
cp ./kubectl-smlogs.${os}.amd64/kubectl-smlogs ~/sagemaker-k8s-bin/.

# This line adds the binaries to your PATH in your .bashrc.

echo 'export PATH=$PATH:~/sagemaker-k8s-bin' >> ~/.bashrc

# Source your .bashrc to update environment variables:
source ~/.bashrc
```

Gunakan perintah berikut untuk memverifikasi bahwa `kubectl` plugin terinstal dengan benar:

```
kubectl smlogs
```

Jika `kubectl` plugin terinstal dengan benar, output Anda akan terlihat seperti berikut:

```
View SageMaker logs via Kubernetes

Usage:
  smlogs [command]

Aliases:
  smlogs, SMLogs, Smlogs

Available Commands:
  BatchTransformJob       View BatchTransformJob logs via Kubernetes
  TrainingJob             View TrainingJob logs via Kubernetes
  help                    Help about any command

Flags:
   -h, --help   help for smlogs
```

```
Use "smlogs [command] --help" for more information about a command.
```

Pembersihan sumber daya

Untuk menghapus instalasi operator dari cluster Anda, Anda harus terlebih dahulu memastikan untuk menghapus semua SageMaker sumber daya dari cluster. Kegagalan untuk melakukannya menyebabkan operasi penghapusan operator hang. Jalankan perintah berikut untuk menghentikan semua pekerjaan:

```
# Delete all SageMaker jobs from Kubernetes
kubectl delete --all --all-namespaces hyperparametertuningjob.sagemaker.aws.amazon.com
kubectl delete --all --all-namespaces trainingjobs.sagemaker.aws.amazon.com
kubectl delete --all --all-namespaces batchtransformjob.sagemaker.aws.amazon.com
kubectl delete --all --all-namespaces hostingdeployment.sagemaker.aws.amazon.com
```

Anda akan melihat output yang serupa dengan yang berikut:

```
$ kubectl delete --all --all-namespaces trainingjobs.sagemaker.aws.amazon.com
trainingjobs.sagemaker.aws.amazon.com "xgboost-mnist-from-for-s3" deleted

$ kubectl delete --all --all-namespaces
 hyperparametertuningjob.sagemaker.aws.amazon.com
hyperparametertuningjob.sagemaker.aws.amazon.com "xgboost-mnist-hpo" deleted

$ kubectl delete --all --all-namespaces batchtransformjob.sagemaker.aws.amazon.com
batchtransformjob.sagemaker.aws.amazon.com "xgboost-mnist" deleted

$ kubectl delete --all --all-namespaces hostingdeployment.sagemaker.aws.amazon.com
hostingdeployment.sagemaker.aws.amazon.com "host-xgboost" deleted
```

Setelah Anda menghapus semua SageMaker pekerjaan, lihat Hapus operator untuk menghapus operator dari klaster Anda.

Hapus operator

Hapus operator berbasis cluster

Operator diinstal menggunakan YAMAL

Untuk menghapus instalasi operator dari cluster Anda, pastikan bahwa semua SageMaker sumber daya telah dihapus dari cluster. Kegagalan untuk melakukannya menyebabkan operasi penghapusan operator hang.

> **Note**
>
> Sebelum menghapus cluster Anda, pastikan untuk menghapus semua SageMaker sumber daya dari cluster. Untuk informasi selengkapnya, lihat Pembersihan sumber daya.

Setelah Anda menghapus semua SageMaker pekerjaan, gunakan `kubectl` untuk menghapus operator dari cluster:

```
# Delete the operator and its resources
kubectl delete -f /installer.yaml
```

Anda akan melihat output yang serupa dengan yang berikut:

```
$ kubectl delete -f raw-yaml/installer.yaml
namespace "sagemaker-k8s-operator-system" deleted
customresourcedefinition.apiextensions.k8s.io
 "batchtransformjobs.sagemaker.aws.amazon.com" deleted
customresourcedefinition.apiextensions.k8s.io
 "endpointconfigs.sagemaker.aws.amazon.com" deleted
customresourcedefinition.apiextensions.k8s.io
 "hostingdeployments.sagemaker.aws.amazon.com" deleted
customresourcedefinition.apiextensions.k8s.io
 "hyperparametertuningjobs.sagemaker.aws.amazon.com" deleted
customresourcedefinition.apiextensions.k8s.io "models.sagemaker.aws.amazon.com" deleted
customresourcedefinition.apiextensions.k8s.io "trainingjobs.sagemaker.aws.amazon.com"
 deleted
role.rbac.authorization.k8s.io "sagemaker-k8s-operator-leader-election-role" deleted
clusterrole.rbac.authorization.k8s.io "sagemaker-k8s-operator-manager-role" deleted
clusterrole.rbac.authorization.k8s.io "sagemaker-k8s-operator-proxy-role" deleted
rolebinding.rbac.authorization.k8s.io "sagemaker-k8s-operator-leader-election-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "sagemaker-k8s-operator-manager-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "sagemaker-k8s-operator-proxy-rolebinding"
 deleted
service "sagemaker-k8s-operator-controller-manager-metrics-service" deleted
deployment.apps "sagemaker-k8s-operator-controller-manager" deleted
secrets "sagemaker-k8s-operator-abcde" deleted
```

Operator diinstal menggunakan Helm Charts

Untuk menghapus CRD operator, pertama-tama hapus semua pekerjaan yang sedang berjalan. Kemudian hapus Bagan Helm yang digunakan untuk menyebarkan operator menggunakan perintah berikut:

```
# get the helm charts
helm ls

# delete the charts
helm delete <chart_name>
```

Hapus operator berbasis namespace

Operator diinstal dengan YAMM

Untuk menghapus instalan operator dari cluster Anda, pertama-tama pastikan bahwa semua SageMaker sumber daya telah dihapus dari cluster. Kegagalan untuk melakukannya menyebabkan operasi penghapusan operator hang.

> **Note**
>
> Sebelum menghapus cluster Anda, pastikan untuk menghapus semua SageMaker sumber daya dari cluster. Untuk informasi selengkapnya, lihat Pembersihan sumber daya.

Setelah Anda menghapus semua SageMaker pekerjaan, gunakan `kubectl` untuk terlebih dahulu menghapus operator dari namespace dan kemudian CRD dari cluster. Jalankan perintah berikut untuk menghapus operator dari cluster:

```
# Delete the operator using the same yaml file that was used to install the operator
kubectl delete -f operator.yaml

# Now delete the CRDs using the CRD installer yaml
kubectl delete -f https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/master/release/rolebased/namespaced/crd.yaml

# Now you can delete the namespace if you want
kubectl delete namespace <namespace>
```

Operator diinstal dengan Helm Charts

Untuk menghapus CRD operator, pertama-tama hapus semua pekerjaan yang sedang berjalan. Kemudian hapus Bagan Helm yang digunakan untuk menyebarkan operator menggunakan perintah berikut:

```
# Delete the operator
helm delete <chart_name>

# delete the crds
helm delete crds

# optionally delete the namespace
kubectl delete namespace <namespace>
```

Memecahkan masalah

Mendebug pekerjaan yang gagal

Gunakan langkah-langkah ini untuk men-debug pekerjaan yang gagal.

- Periksa status pekerjaan dengan menjalankan yang berikut:

  ```
  kubectl get <CRD Type> <job name>
  ```

- Jika pekerjaan dibuat di SageMaker, Anda dapat menggunakan perintah berikut untuk melihat `STATUS` dan`SageMaker Job Name`:

  ```
  kubectl get <crd type> <job name>
  ```

- Anda dapat menggunakan `smlogs` untuk menemukan penyebab masalah menggunakan perintah berikut:

  ```
  kubectl smlogs <crd type> <job name>
  ```

- Anda juga dapat menggunakan `describe` perintah untuk mendapatkan detail lebih lanjut tentang pekerjaan menggunakan perintah berikut. Output memiliki `additional` bidang yang memiliki informasi lebih lanjut tentang status pekerjaan.

  ```
  kubectl describe <crd type> <job name>
  ```

- Jika pekerjaan tidak dibuat SageMaker, gunakan log pod operator untuk menemukan penyebab masalah sebagai berikut:

```
$ kubectl get pods -A | grep sagemaker
# Output:
sagemaker-k8s-operator-system   sagemaker-k8s-operator-controller-manager-5cd7df4d74-
wh22z   2/2     Running   0          3h33m

$ kubectl logs -p <pod name> -c manager -n sagemaker-k8s-operator-system
```

Menghapus CRD operator

Jika menghapus pekerjaan tidak berfungsi, periksa apakah operator sedang berjalan. Jika operator tidak berjalan, maka Anda harus menghapus finalizer menggunakan langkah-langkah berikut:

1. Di terminal baru, buka pekerjaan di editor menggunakan `kubectl edit` sebagai berikut:

   ```
   kubectl edit <crd type> <job name>
   ```

2. Edit pekerjaan untuk menghapus finalizer dengan menghapus dua baris berikut dari file. Simpan file dan pekerjaan akan dihapus.

   ```
   finalizers:
     - sagemaker-operator-finalizer
   ```

Gambar dan SMlog di setiap Wilayah

Tabel berikut mencantumkan gambar operator dan SMLogs yang tersedia di setiap Wilayah.

Wila	Gambar Pengontrol	Contoh
us-east-	`957583890962.dkr.ecr.us-east-1.amazonaws.com/amazon-sagemaker-operator-for-k8s:v1`	https://s3.us-east-1.amazonaws.com/ amazon-sagemaker-operator-for -k8 -1/ s-us-east /v1/linux .amd64.tar.gz kubectl-smlogs-plugin

Wila	Gambar Pengontrol	Contoh
us-east-	`922499468684.dkr.ecr.us-east-2.amazonaws.com/amazon-sagemaker-operator-for-k8s:v1`	https://s3.us-east-2.amazonaws.com/ amazon-sagemaker-operator-for -k8 -2/ s-us-east /v1/linux .amd64.tar.gz kubectl-smlogs-plugin
us-west	`640106867763.dkr.ecr.us-west-2.amazonaws.com/amazon-sagemaker-operator-for-k8s:v1`	https://s3.us-west-2.amazonaws.com/ amazon-sagemaker-operator-for -k8 -2/ s-us-west /v1/linux .amd64.tar.gz kubectl-smlogs-plugin
eu-west	`613661167059.dkr.ecr.eu-west-1.amazonaws.com/amazon-sagemaker-operator-for-k8s:v1`	https://s3.eu-west-1.amazonaws.com/ amazon-sagemaker-operator-for -k8 -1/ s-eu-west /v1/linux .amd64.tar.gz kubectl-smlogs-plugin

Gunakan SageMaker Pekerjaan Amazon

Bagian ini didasarkan pada versi asli SageMaker Operator untuk Kubernetes.

> **Important**
>
> Kami menghentikan pengembangan dan dukungan teknis dari versi asli SageMaker Operator untuk Kubernetes.
> Jika saat ini Anda menggunakan SageMaker Operator untuk Kubernetes versi `v1.2.2` atau di bawah ini, kami sarankan untuk memigrasikan sumber daya Anda ke pengontrol layanan ACK untuk Amazon. SageMaker Pengontrol layanan ACK adalah generasi baru SageMaker Operator untuk Kubernetes berdasarkan AWSController for Kubernetes (ACK).
> Untuk informasi tentang langkah-langkah migrasi, lihatMigrasikan sumber daya ke Operator terbaru.
> Untuk jawaban atas pertanyaan umum di akhir dukungan versi asli SageMaker Operator untuk Kubernetes, lihat Mengumumkan Berakhirnya Support Versi Original SageMaker Operator untuk Kubernetes

Untuk menjalankan SageMaker pekerjaan Amazon menggunakan Operator untuk Kubernetes, Anda dapat menerapkan file YAMM atau menggunakan Bagan Helm yang disediakan.

Semua pekerjaan operator sampel dalam tutorial berikut menggunakan data sampel yang diambil dari kumpulan data MNIST publik. Untuk menjalankan sampel ini, unduh kumpulan data ke bucket Amazon S3 Anda. Anda dapat menemukan kumpulan data di Unduh Dataset MNIST.

Daftar Isi

TrainingJob Operator

Operator pekerjaan pelatihan merekonsiliasi spesifikasi pekerjaan pelatihan yang Anda tentukan SageMaker dengan meluncurkannya untuk Anda di. SageMaker Anda dapat mempelajari lebih lanjut tentang pekerjaan SageMaker pelatihan dalam dokumentasi SageMaker CreateTrainingJob API.

Topik

Membuat file TrainingJob menggunakan YAMAL

1. Unduh file YAMAL contoh untuk pelatihan menggunakan perintah berikut:

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/samples/xgboost-mnist-trainingjob.yaml
```

2. Edit `xgboost-mnist-trainingjob.yaml` file untuk mengganti `roleArn` parameter dengan Anda`<sagemaker-execution-role>`, dan `outputPath` dengan bucket Amazon S3 Anda yang peran SageMaker eksekusi memiliki akses tulis. `roleArn`Harus memiliki izin sehingga SageMaker dapat mengakses Amazon S3, CloudWatch Amazon, dan layanan lainnya atas nama Anda. Untuk informasi selengkapnya tentang membuat SageMaker ExecutionRole, lihat SageMaker Peran. Terapkan file YAMAL menggunakan perintah berikut:

   ```
   kubectl apply -f xgboost-mnist-trainingjob.yaml
   ```

Membuat TrainingJob Menggunakan Bagan Helm

Anda dapat menggunakan Helm Charts untuk menjalankan TrainingJobs.

1. Kloning GitHub repositori untuk mendapatkan sumber menggunakan perintah berikut:

   ```
   git clone https://github.com/aws/amazon-sagemaker-operator-for-k8s.git
   ```

2. Arahkan ke `amazon-sagemaker-operator-for-k8s/hack/charts/training-jobs/` folder dan edit `values.yaml` file untuk mengganti nilai seperti `rolearn` dan `outputpath` dengan nilai yang sesuai dengan akun Anda. RoLearn harus memiliki izin sehingga SageMaker dapat mengakses Amazon S3, Amazon CloudWatch, dan layanan lainnya atas nama Anda. Untuk informasi selengkapnya tentang membuat SageMaker ExecutionRole, lihat SageMaker Peran.

Buat TrainingJob

Dengan peran dan bucket Amazon S3 diganti dengan nilai yang sesuai di`values.yaml`, Anda dapat membuat pekerjaan pelatihan menggunakan perintah berikut:

```
helm install . --generate-name
```

Outputnya akan terlihat seperti berikut:

```
NAME: chart-12345678
LAST DEPLOYED: Wed Nov 20 23:35:49 2019
NAMESPACE: default
STATUS: deployed
REVISION: 1
```

```
TEST SUITE: None
NOTES:
Thanks for installing the sagemaker-k8s-trainingjob.
```

Verifikasi Bagan Helm pelatihan Anda

Untuk memverifikasi bahwa Bagan Helm berhasil dibuat, jalankan:

```
helm ls
```

Outputnya akan terlihat seperti berikut:

```
NAME                    NAMESPACE       REVISION        UPDATED
         STATUS          CHART                           APP VERSION
chart-12345678        default         1               2019-11-20 23:35:49.9136092 +0000
 UTC   deployed        sagemaker-k8s-trainingjob-0.1.0
rolebased-12345678    default         1               2019-11-20 23:14:59.6777082 +0000
 UTC   deployed        sagemaker-k8s-operator-0.1.0
```

`helm install`membuat sumber daya `TrainingJob` Kubernetes. Operator meluncurkan pekerjaan pelatihan yang sebenarnya SageMaker dan memperbarui sumber daya `TrainingJob` Kubernetes untuk mencerminkan status pekerjaan di. SageMaker Anda dikenakan biaya untuk SageMaker sumber daya yang digunakan selama durasi pekerjaan Anda. Anda tidak dikenakan biaya apa pun setelah pekerjaan Anda selesai atau berhenti.

Catatan: SageMaker tidak memungkinkan Anda memperbarui pekerjaan pelatihan yang sedang berjalan. Anda tidak dapat mengedit parameter apa pun dan menerapkan kembali file konfigurasi. Baik mengubah nama metadata atau menghapus pekerjaan yang sudah ada dan membuat pekerjaan baru. Mirip dengan operator pekerjaan pelatihan yang ada seperti TFJob di Kubeflow, `update` tidak didukung.

Daftar TrainingJobs

Gunakan perintah berikut untuk mencantumkan semua pekerjaan yang dibuat menggunakan operator Kubernetes:

```
kubectl get TrainingJob
```

Outputnya daftar semua pekerjaan akan terlihat seperti berikut:

```
kubectl get trainingjobs
NAME                          STATUS       SECONDARY-STATUS   CREATION-TIME
 SAGEMAKER-JOB-NAME
xgboost-mnist-from-for-s3   InProgress   Starting           2019-11-20T23:42:35Z
 xgboost-mnist-from-for-s3-examplef11eab94e0ed4671d5a8f
```

Pekerjaan pelatihan terus terdaftar setelah pekerjaan selesai atau gagal. Anda dapat menghapus `TrainingJob` pekerjaan dari daftar dengan mengikuti Hapus TrainingJobs langkah-langkahnya. Pekerjaan yang telah selesai atau dihentikan tidak dikenakan biaya apa pun untuk SageMaker sumber daya.

TrainingJob nilai status

`STATUS`Bidang dapat berupa salah satu dari nilai berikut:

- `Completed`
- `InProgress`
- `Failed`
- `Stopped`
- `Stopping`

Status ini datang langsung dari dokumentasi API SageMaker resmi.

Selain SageMaker status resmi, dimungkinkan `STATUS` untuk menjadi`SynchronizingK8sJobWithSageMaker`. Ini berarti bahwa operator belum memproses pekerjaan.

Nilai status sekunder

Status sekunder datang langsung dari dokumentasi API SageMaker resmi. Bidang ini berisi informasi yang lebih terperinci tentang status pekerjaan.

Jelaskan TrainingJob

Anda bisa mendapatkan detail lebih lanjut tentang pekerjaan pelatihan dengan menggunakan `describe` `kubectl` perintah. Ini biasanya digunakan untuk men-debug masalah atau memeriksa parameter pekerjaan pelatihan. Untuk mendapatkan informasi tentang pekerjaan pelatihan Anda, gunakan perintah berikut:

```
kubectl describe trainingjob xgboost-mnist-from-for-s3
```

Outputnya untuk pekerjaan pelatihan Anda akan terlihat seperti berikut:

```
Name:         xgboost-mnist-from-for-s3
Namespace:    default
Labels:       <none>
Annotations:  <none>
API Version:  sagemaker.aws.amazon.com/v1
Kind:         TrainingJob
Metadata:
  Creation Timestamp:  2019-11-20T23:42:35Z
  Finalizers:
    sagemaker-operator-finalizer
  Generation:        2
  Resource Version:  23119
  Self Link:         /apis/sagemaker.aws.amazon.com/v1/namespaces/default/trainingjobs/
xgboost-mnist-from-for-s3
  UID:               6d7uiui-0bef-11ea-b94e-0ed467example
Spec:
  Algorithm Specification:
    Training Image:       8256416981234.dkr.ecr.us-east-2.amazonaws.com/xgboost:1
    Training Input Mode:  File
  Hyper Parameters:
    Name:   eta
    Value:  0.2
    Name:   gamma
    Value:  4
    Name:   max_depth
    Value:  5
    Name:   min_child_weight
    Value:  6
    Name:   num_class
    Value:  10
    Name:   num_round
    Value:  10
    Name:   objective
    Value:  multi:softmax
    Name:   silent
    Value:  0
  Input Data Config:
    Channel Name:      train
    Compression Type:  None
```

```
    Content Type:      text/csv
    Data Source:
      S 3 Data Source:
        S 3 Data Distribution Type:  FullyReplicated
        S 3 Data Type:               S3Prefix
        S 3 Uri:                     https://s3-us-east-2.amazonaws.com/my-bucket/
sagemaker/xgboost-mnist/train/
    Channel Name:                    validation
    Compression Type:                None
    Content Type:                    text/csv
    Data Source:
      S 3 Data Source:
        S 3 Data Distribution Type:  FullyReplicated
        S 3 Data Type:               S3Prefix
        S 3 Uri:                     https://s3-us-east-2.amazonaws.com/my-bucket/
sagemaker/xgboost-mnist/validation/
  Output Data Config:
    S 3 Output Path:  s3://my-bucket/sagemaker/xgboost-mnist/xgboost/
  Region:             us-east-2
  Resource Config:
    Instance Count:     1
    Instance Type:      ml.m4.xlarge
    Volume Size In GB:  5
  Role Arn:             arn:aws:iam::12345678910:role/service-role/AmazonSageMaker-
ExecutionRole
  Stopping Condition:
    Max Runtime In Seconds:  86400
  Training Job Name:         xgboost-mnist-from-for-s3-6d7fa0af0bef11eab94e0example
Status:
  Cloud Watch Log URL:           https://us-east-2.console.aws.amazon.com/
cloudwatch/home?region=us-east-2#logStream:group=/aws/sagemaker/
TrainingJobs;prefix=<example>;streamFilter=typeLogStreamPrefix
  Last Check Time:               2019-11-20T23:44:29Z
  Sage Maker Training Job Name:  xgboost-mnist-from-for-s3-6d7fa0af0bef11eab94eexample
  Secondary Status:              Downloading
  Training Job Status:           InProgress
Events:                          <none>
```

Lihat log dari TrainingJobs

Gunakan perintah berikut untuk melihat log dari pekerjaan `kmeans-mnist` pelatihan:

```
kubectl smlogs trainingjob xgboost-mnist-from-for-s3
```

Outputnya semestinya mirip dengan yang berikut. Log dari instance diurutkan secara kronologis.

```
"xgboost-mnist-from-for-s3" has SageMaker TrainingJobName "xgboost-mnist-from-
for-s3-123456789" in region "us-east-2", status "InProgress" and secondary status
 "Starting"
xgboost-mnist-from-for-s3-6d7fa0af0bef11eab94e0ed46example/algo-1-1574293123 2019-11-20
 23:45:24.7 +0000 UTC Arguments: train
xgboost-mnist-from-for-s3-6d7fa0af0bef11eab94e0ed46example/algo-1-1574293123 2019-11-20
 23:45:24.7 +0000 UTC [2019-11-20:23:45:22:INFO] Running standalone xgboost training.
xgboost-mnist-from-for-s3-6d7fa0af0bef11eab94e0ed46example/algo-1-1574293123 2019-11-20
 23:45:24.7 +0000 UTC [2019-11-20:23:45:22:INFO] File size need to be processed in the
 node: 1122.95mb. Available memory size in the node: 8586.0mb
xgboost-mnist-from-for-s3-6d7fa0af0bef11eab94e0ed46example/algo-1-1574293123 2019-11-20
 23:45:24.7 +0000 UTC [2019-11-20:23:45:22:INFO] Determined delimiter of CSV input is
 ','
xgboost-mnist-from-for-s3-6d7fa0af0bef11eab94e0ed46example/algo-1-1574293123 2019-11-20
 23:45:24.7 +0000 UTC [23:45:22] S3DistributionType set as FullyReplicated
```

Hapus TrainingJobs

Gunakan perintah berikut untuk menghentikan pekerjaan pelatihan di Amazon SageMaker:

```
kubectl delete trainingjob xgboost-mnist-from-for-s3
```

Perintah ini menghapus tugas SageMaker pelatihan dari Kubernetes. Perintah ini mengembalikan output berikut:

```
trainingjob.sagemaker.aws.amazon.com "xgboost-mnist-from-for-s3" deleted
```

Jika pekerjaan masih berlangsung SageMaker, pekerjaan berhenti. Anda tidak dikenakan biaya apa pun untuk SageMaker sumber daya setelah pekerjaan Anda berhenti atau selesai.

Catatan: SageMaker tidak menghapus pekerjaan pelatihan. Pekerjaan yang dihentikan terus ditampilkan di SageMaker konsol. `delete`Perintah membutuhkan waktu sekitar 2 menit untuk membersihkan sumber daya dari SageMaker.

HyperParameterTuningJobOperator

Operator pekerjaan tuning hyperparameter merekonsiliasi spesifikasi pekerjaan tuning hyperparameter yang Anda tentukan dengan meluncurkannya di. SageMaker SageMaker Anda

dapat mempelajari lebih lanjut tentang tugas penyetelan SageMaker hyperparameter di dokumentasi SageMaker CreateHyperParameterTuningJob API.

Topik

Membuat file HyperparameterTuningJob menggunakan YAMAL

1. Unduh file YAMM sampel untuk tugas tuning hyperparameter menggunakan perintah berikut:

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/samples/xgboost-mnist-hpo.yaml
```

2. Edit `xgboost-mnist-hpo.yaml` file untuk mengganti `roleArn` parameter dengan file Anda`sagemaker-execution-role`. Agar pekerjaan tuning hyperparameter berhasil, Anda juga harus mengubah `s3InputPath` dan `s3OutputPath` ke nilai yang sesuai dengan akun Anda. Terapkan pembaruan file YAM menggunakan perintah berikut:

```
kubectl apply -f xgboost-mnist-hpo.yaml
```

Buat HyperparameterTuningJob menggunakan Bagan Helm

Anda dapat menggunakan Helm Charts untuk menjalankan pekerjaan tuning hyperparameter.

1. Kloning GitHub repositori untuk mendapatkan sumber menggunakan perintah berikut:

```
git clone https://github.com/aws/amazon-sagemaker-operator-for-k8s.git
```

2. Arahkan ke `amazon-sagemaker-operator-for-k8s/hack/charts/hyperparameter-tuning-jobs/` folder.

3. Edit `values.yaml` file untuk mengganti `roleArn` parameter dengan file Andasagemaker-execution-role. Agar pekerjaan tuning hyperparameter berhasil, Anda juga harus mengubah `s3InputPath` dan `s3OutputPath` ke nilai yang sesuai dengan akun Anda.

Buat HyperparameterTuningJob

Dengan peran dan jalur Amazon S3 diganti dengan nilai yang sesuai di`values.yaml`, Anda dapat membuat tugas penyetelan hyperparameter menggunakan perintah berikut:

```
helm install . --generate-name
```

Outputnya semestinya mirip dengan yang berikut:

```
NAME: chart-1574292948
LAST DEPLOYED: Wed Nov 20 23:35:49 2019
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thanks for installing the sagemaker-k8s-hyperparametertuningjob.
```

Verifikasi instalasi bagan

Untuk memverifikasi bahwa Bagan Helm berhasil dibuat, jalankan perintah berikut:

```
helm ls
```

Outputnya akan terlihat seperti berikut:

```
NAME                    NAMESPACE       REVISION        UPDATED
chart-1474292948        default         1               2019-11-20 23:35:49.9136092
 +0000 UTC   deployed        sagemaker-k8s-hyperparametertuningjob-0.1.0
                STATUS          CHART                           APP VERSION
chart-1574292948        default         1               2019-11-20 23:35:49.9136092
 +0000 UTC   deployed        sagemaker-k8s-trainingjob-0.1.0
rolebased-1574291698    default         1               2019-11-20 23:14:59.6777082
 +0000 UTC   deployed        sagemaker-k8s-operator-0.1.0
```

`helm install`membuat sumber daya `HyperParameterTuningJob` Kubernetes. Operator meluncurkan pekerjaan optimasi hyperparameter yang sebenarnya SageMaker dan memperbarui sumber daya `HyperParameterTuningJob` Kubernetes untuk mencerminkan status pekerjaan di. SageMaker Anda dikenakan biaya untuk SageMaker sumber daya yang digunakan selama durasi pekerjaan Anda. Anda tidak dikenakan biaya apa pun setelah pekerjaan Anda selesai atau berhenti.

Catatan: SageMaker tidak mengizinkan Anda memperbarui pekerjaan tuning hyperparameter yang sedang berjalan. Anda tidak dapat mengedit parameter apa pun dan menerapkan kembali file konfigurasi. Anda harus mengubah nama metadata atau menghapus pekerjaan yang ada dan membuat pekerjaan yang baru. Mirip dengan operator pekerjaan pelatihan yang ada seperti `TFJob` di Kubeflow, tidak `update` didukung.

Daftar HyperparameterTuningJobs

Gunakan perintah berikut untuk mencantumkan semua pekerjaan yang dibuat menggunakan operator Kubernetes:

```
kubectl get hyperparametertuningjob
```

Outputnya akan terlihat seperti berikut:

```
NAME         STATUS      CREATION-TIME          COMPLETED   INPROGRESS   ERRORS
 STOPPED   BEST-TRAINING-JOB                                SAGEMAKER-JOB-NAME
xgboost-mnist-hpo   Completed   2019-10-17T01:15:52Z   10          0
        0        0         xgboostha92f5e3cf07b11e9bf6c06d6-009-4c7a123
 xgboostha92f5e3cf07b11e9bf6c123
```

Pekerjaan tuning hyperparameter terus terdaftar setelah pekerjaan selesai atau gagal. Anda dapat menghapus a `hyperparametertuningjob` dari daftar dengan mengikuti langkah-langkah di[Menghapus HyperparameterTuningJob](). Pekerjaan yang telah selesai atau dihentikan tidak dikenakan biaya apa pun untuk SageMaker sumber daya.

Nilai status pekerjaan tuning hyperparameter

`STATUS`Bidang dapat berupa salah satu dari nilai berikut:

- `Completed`
- `InProgress`
- `Failed`
- `Stopped`

- `Stopping`

Status ini datang langsung dari dokumentasi API SageMaker resmi.

Selain SageMaker status resmi, dimungkinkan `STATUS` untuk menjadi`SynchronizingK8sJobWithSageMaker`. Ini berarti bahwa operator belum memproses pekerjaan.

Penghitung Galois

Outputnya memiliki beberapa penghitung, seperti `COMPLETED` dan`INPROGRESS`. Ini mewakili berapa banyak pekerjaan pelatihan yang telah diselesaikan dan sedang berlangsung, masing-masing. Untuk informasi selengkapnya tentang cara penentuannya, lihat TrainingJobStatusCountersdi dokumentasi SageMaker API.

Terbaik TrainingJob

Kolom ini berisi nama `TrainingJob` yang paling dioptimalkan metrik yang dipilih.

Untuk melihat ringkasan hiperparameter yang disetel, jalankan:

```
kubectl describe hyperparametertuningjob xgboost-mnist-hpo
```

Untuk melihat informasi rinci tentang`TrainingJob`, jalankan:

```
kubectl describe trainingjobs <job name>
```

Melahirkan TrainingJobs

Anda juga dapat melacak semua 10 pekerjaan pelatihan di Kubernetes yang diluncurkan `HyperparameterTuningJob` dengan menjalankan perintah berikut:

```
kubectl get trainingjobs
```

Jelaskan HyperparameterTuningJob

Anda dapat memperoleh detail debugging menggunakan `describe` `kubectl` perintah.

```
kubectl describe hyperparametertuningjob xgboost-mnist-hpo
```

Selain informasi tentang pekerjaan tuning, SageMaker Operator untuk Kubernetes juga memaparkan pekerjaan pelatihan terbaik yang ditemukan oleh pekerjaan tuning hyperparameter dalam output sebagai berikut: `describe`

```
Name:         xgboost-mnist-hpo
Namespace:    default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"sagemaker.aws.amazon.com/
v1","kind":"HyperparameterTuningJob","metadata":{"annotations":{},"name":"xgboost-
mnist-hpo","namespace":...
API Version:  sagemaker.aws.amazon.com/v1
Kind:         HyperparameterTuningJob
Metadata:
  Creation Timestamp:  2019-10-17T01:15:52Z
  Finalizers:
    sagemaker-operator-finalizer
  Generation:        2
  Resource Version:  8167
  Self Link:         /apis/sagemaker.aws.amazon.com/v1/namespaces/default/
hyperparametertuningjobs/xgboost-mnist-hpo
  UID:               a92f5e3c-f07b-11e9-bf6c-06d6f303uidu
Spec:
  Hyper Parameter Tuning Job Config:
    Hyper Parameter Tuning Job Objective:
      Metric Name:  validation:error
      Type:         Minimize
    Parameter Ranges:
      Integer Parameter Ranges:
        Max Value:     20
        Min Value:     10
        Name:          num_round
        Scaling Type:  Linear
    Resource Limits:
      Max Number Of Training Jobs:     10
      Max Parallel Training Jobs:      10
    Strategy:                          Bayesian
    Training Job Early Stopping Type:  Off
  Hyper Parameter Tuning Job Name:     xgboostha92f5e3cf07b11e9bf6c06d6
  Region:                              us-east-2
  Training Job Definition:
    Algorithm Specification:
      Training Image:       12345678910.dkr.ecr.us-east-2.amazonaws.com/xgboost:1
```

```
      Training Input Mode:  File
    Input Data Config:
      Channel Name:  train
      Content Type:  text/csv
      Data Source:
        s3DataSource:
          s3DataDistributionType:  FullyReplicated
          s3DataType:              S3Prefix
          s3Uri:                   https://s3-us-east-2.amazonaws.com/my-bucket/
sagemaker/xgboost-mnist/train/
      Channel Name:                validation
      Content Type:                text/csv
      Data Source:
        s3DataSource:
          s3DataDistributionType:  FullyReplicated
          s3DataType:              S3Prefix
          s3Uri:                   https://s3-us-east-2.amazonaws.com/my-bucket/
sagemaker/xgboost-mnist/validation/
    Output Data Config:
      s3OutputPath:  https://s3-us-east-2.amazonaws.com/my-bucket/sagemaker/xgboost-
mnist/xgboost
    Resource Config:
      Instance Count:     1
      Instance Type:      ml.m4.xlarge
      Volume Size In GB:  5
    Role Arn:             arn:aws:iam::123456789012:role/service-role/AmazonSageMaker-
ExecutionRole
    Static Hyper Parameters:
      Name:   base_score
      Value:  0.5
      Name:   booster
      Value:  gbtree
      Name:   csv_weights
      Value:  0
      Name:   dsplit
      Value:  row
      Name:   grow_policy
      Value:  depthwise
      Name:   lambda_bias
      Value:  0.0
      Name:   max_bin
      Value:  256
      Name:   max_leaves
      Value:  0
```

```
      Name:   normalize_type
      Value:  tree
      Name:   objective
      Value:  reg:linear
      Name:   one_drop
      Value:  0
      Name:   prob_buffer_row
      Value:  1.0
      Name:   process_type
      Value:  default
      Name:   rate_drop
      Value:  0.0
      Name:   refresh_leaf
      Value:  1
      Name:   sample_type
      Value:  uniform
      Name:   scale_pos_weight
      Value:  1.0
      Name:   silent
      Value:  0
      Name:   sketch_eps
      Value:  0.03
      Name:   skip_drop
      Value:  0.0
      Name:   tree_method
      Value:  auto
      Name:   tweedie_variance_power
      Value:  1.5
    Stopping Condition:
      Max Runtime In Seconds:  86400
Status:
  Best Training Job:
    Creation Time:  2019-10-17T01:16:14Z
    Final Hyper Parameter Tuning Job Objective Metric:
      Metric Name:        validation:error
      Value:
    Objective Status:     Succeeded
    Training End Time:    2019-10-17T01:20:24Z
    Training Job Arn:     arn:aws:sagemaker:us-east-2:123456789012:training-job/
xgboostha92f5e3cf07b11e9bf6c06d6-009-4sample
    Training Job Name:    xgboostha92f5e3cf07b11e9bf6c06d6-009-4c7a3059
    Training Job Status:  Completed
    Training Start Time:  2019-10-17T01:18:35Z
    Tuned Hyper Parameters:
```

```
      Name:                                     num_round
      Value:                                    18
  Hyper Parameter Tuning Job Status:            Completed
  Last Check Time:                              2019-10-17T01:21:01Z
  Sage Maker Hyper Parameter Tuning Job Name:   xgboostha92f5e3cf07b11e9bf6c06d6
  Training Job Status Counters:
    Completed:            10
    In Progress:          0
    Non Retryable Error:  0
    Retryable Error:      0
    Stopped:              0
    Total Error:          0
Events:                   <none>
```

Lihat log dari HyperparameterTuningJobs

Pekerjaan tuning hyperparameter tidak memiliki log, tetapi semua pekerjaan pelatihan yang diluncurkan oleh mereka memang memiliki log. Log ini dapat diakses seolah-olah itu adalah pekerjaan pelatihan normal. Untuk informasi selengkapnya, lihat Lihat log dari TrainingJobs.

Menghapus HyperparameterTuningJob

Gunakan perintah berikut untuk menghentikan pekerjaan hyperparameter di SageMaker.

```
kubectl delete hyperparametertuningjob xgboost-mnist-hpo
```

Perintah ini menghapus tugas tuning hyperparameter dan pekerjaan pelatihan terkait dari klaster Kubernetes Anda dan menghentikannya. SageMaker Pekerjaan yang telah berhenti atau selesai tidak dikenakan biaya apa pun untuk SageMaker sumber daya. SageMaker tidak menghapus pekerjaan tuning hyperparameter. Pekerjaan yang dihentikan terus ditampilkan di SageMaker konsol.

Outputnya akan terlihat seperti berikut:

```
hyperparametertuningjob.sagemaker.aws.amazon.com "xgboost-mnist-hpo" deleted
```

Catatan: Perintah delete membutuhkan waktu sekitar 2 menit untuk membersihkan sumber daya dari SageMaker.

BatchTransformJob Operator

Operator pekerjaan transformasi Batch merekonsiliasi spesifikasi pekerjaan transformasi batch yang Anda tentukan SageMaker dengan meluncurkannya. SageMaker Anda dapat mempelajari

lebih lanjut tentang pekerjaan transformasi SageMaker batch dalam dokumentasi SageMaker CreateTransformJob API.

Topik

- Membuat BatchTransformJob menggunakan File YAMM
- Buat BatchTransformJob menggunakan Bagan Helm
- Daftar BatchTransformJobs
- Jelaskan BatchTransformJob
- Lihat log dari BatchTransformJobs
- Menghapus BatchTransformJob

Membuat BatchTransformJob menggunakan File YAMM

1. Unduh file YAMM sampel untuk tugas transformasi batch menggunakan perintah berikut:

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/samples/xgboost-mnist-batchtransform.yaml
```

2. Edit file `xgboost-mnist-batchtransform.yaml` untuk mengubah parameter yang diperlukan untuk mengganti `inputdataconfig` dengan data input Anda dan `s3OutputPath` dengan bucket Amazon S3 Anda yang peran SageMaker eksekusi memiliki akses tulis.
3. Terapkan file YAMAL menggunakan perintah berikut:

```
kubectl apply -f xgboost-mnist-batchtransform.yaml
```

Buat BatchTransformJob menggunakan Bagan Helm

Anda dapat menggunakan Helm Charts untuk menjalankan pekerjaan transformasi batch.

Dapatkan direktori installer Helm

Kloning GitHub repositori untuk mendapatkan sumber menggunakan perintah berikut:

```
git clone https://github.com/aws/amazon-sagemaker-operator-for-k8s.git
```

Konfigurasikan Bagan Helm

Arahkan ke `amazon-sagemaker-operator-for-k8s/hack/charts/batch-transform-jobs/` folder.

Edit `values.yaml` file untuk mengganti `inputdataconfig` dengan data input dan outputPath Anda dengan bucket S3 Anda yang peran SageMaker eksekusi memiliki akses tulis.

Buat BatchTransformJob

1. Gunakan perintah berikut untuk membuat pekerjaan transformasi batch:

```
helm install . --generate-name
```

   Outputnya akan terlihat seperti berikut:

```
NAME: chart-1574292948
LAST DEPLOYED: Wed Nov 20 23:35:49 2019
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thanks for installing the sagemaker-k8s-batch-transform-job.
```

2. Untuk memverifikasi bahwa Bagan Helm berhasil dibuat, jalankan perintah berikut:

```
helm ls
NAME                    NAMESPACE       REVISION        UPDATED
             STATUS          CHART                           APP VERSION
chart-1474292948        default         1               2019-11-20 23:35:49.9136092
 +0000 UTC   deployed        sagemaker-k8s-batchtransformjob-0.1.0
chart-1474292948        default         1               2019-11-20 23:35:49.9136092
 +0000 UTC   deployed        sagemaker-k8s-hyperparametertuningjob-0.1.0
chart-1574292948        default         1               2019-11-20 23:35:49.9136092
 +0000 UTC   deployed        sagemaker-k8s-trainingjob-0.1.0
rolebased-1574291698    default         1               2019-11-20 23:14:59.6777082
 +0000 UTC   deployed        sagemaker-k8s-operator-0.1.0
```

   Perintah ini membuat sumber daya `BatchTransformJob` Kubernetes. Operator meluncurkan pekerjaan transformasi yang sebenarnya SageMaker dan memperbarui sumber daya `BatchTransformJob` Kubernetes untuk mencerminkan status pekerjaan di. SageMaker Anda

dikenakan biaya untuk SageMaker sumber daya yang digunakan selama durasi pekerjaan Anda. Anda tidak dikenakan biaya apa pun setelah pekerjaan Anda selesai atau berhenti.

Catatan: SageMaker tidak mengizinkan Anda memperbarui pekerjaan transformasi batch yang sedang berjalan. Anda tidak dapat mengedit parameter apa pun dan menerapkan kembali file konfigurasi. Anda harus mengubah nama metadata atau menghapus pekerjaan yang ada dan membuat pekerjaan yang baru. Mirip dengan operator pekerjaan pelatihan yang ada seperti `TFJob` di Kubeflow, tidak `update` didukung.

Daftar BatchTransformJobs

Gunakan perintah berikut untuk mencantumkan semua pekerjaan yang dibuat menggunakan operator Kubernetes:

```
kubectl get batchtransformjob
```

Outputnya akan terlihat seperti berikut:

```
NAME                              STATUS      CREATION-TIME          SAGEMAKER-JOB-
NAME
xgboost-mnist-batch-transform     Completed   2019-11-18T03:44:00Z   xgboost-mnist-
a88fb19809b511eaac440aa8axgboost
```

Pekerjaan transformasi batch terus terdaftar setelah pekerjaan selesai atau gagal. Anda dapat menghapus a `hyperparametertuningjob` dari daftar dengan mengikuti Menghapus BatchTransformJob langkah-langkahnya. Pekerjaan yang telah selesai atau dihentikan tidak dikenakan biaya apa pun untuk SageMaker sumber daya.

Nilai status transformasi Batch

`STATUS`Bidang dapat berupa salah satu dari nilai berikut:

- `Completed`
- `InProgress`
- `Failed`
- `Stopped`
- `Stopping`

Status ini datang langsung dari dokumentasi API SageMaker resmi.

Selain SageMaker status resmi, dimungkinkan `STATUS` untuk menjadi`SynchronizingK8sJobWithSageMaker`. Ini berarti bahwa operator belum memproses pekerjaan.

Jelaskan BatchTransformJob

Anda dapat memperoleh detail debugging menggunakan `describe` `kubectl` perintah.

```
kubectl describe batchtransformjob xgboost-mnist-batch-transform
```

Outputnya akan terlihat seperti berikut:

```
Name:         xgboost-mnist-batch-transform
Namespace:    default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"sagemaker.aws.amazon.com/
v1","kind":"BatchTransformJob","metadata":{"annotations":{},"name":"xgboost-
mnist","namespace"...
API Version:  sagemaker.aws.amazon.com/v1
Kind:         BatchTransformJob
Metadata:
  Creation Timestamp:  2019-11-18T03:44:00Z
  Finalizers:
    sagemaker-operator-finalizer
  Generation:        2
  Resource Version:  21990924
  Self Link:         /apis/sagemaker.aws.amazon.com/v1/namespaces/default/
batchtransformjobs/xgboost-mnist
  UID:               a88fb198-09b5-11ea-ac44-0aa8a9UIDNUM
Spec:
  Model Name:  TrainingJob-20190814SMJOb-IKEB
  Region:      us-east-1
  Transform Input:
    Content Type:  text/csv
    Data Source:
      S 3 Data Source:
        S 3 Data Type:  S3Prefix
        S 3 Uri:        s3://my-bucket/mnist_kmeans_example/input
  Transform Job Name:   xgboost-mnist-a88fb19809b511eaac440aa8a9SMJOB
  Transform Output:
    S 3 Output Path:  s3://my-bucket/mnist_kmeans_example/output
  Transform Resources:
```

```
    Instance Count:  1
    Instance Type:   ml.m4.xlarge
Status:
  Last Check Time:                2019-11-19T22:50:40Z
  Sage Maker Transform Job Name:  xgboost-mnist-a88fb19809b511eaac440aaSMJOB
  Transform Job Status:           Completed
Events:                           <none>
```

Lihat log dari BatchTransformJobs

Gunakan perintah berikut untuk melihat log dari pekerjaan transformasi `xgboost-mnist` batch:

```
kubectl smlogs batchtransformjob xgboost-mnist-batch-transform
```

Menghapus BatchTransformJob

Gunakan perintah berikut untuk menghentikan pekerjaan transformasi batch di SageMaker.

```
kubectl delete batchTransformJob xgboost-mnist-batch-transform
```

Outputnya akan terlihat seperti berikut:

```
batchtransformjob.sagemaker.aws.amazon.com "xgboost-mnist" deleted
```

Perintah ini menghapus tugas transformasi batch dari cluster Kubernetes Anda, serta menghentikannya. SageMaker Pekerjaan yang telah berhenti atau selesai tidak dikenakan biaya apa pun untuk SageMaker sumber daya. Hapus membutuhkan waktu sekitar 2 menit untuk membersihkan sumber daya dari SageMaker.

Catatan: SageMaker tidak menghapus pekerjaan transformasi batch. Pekerjaan yang dihentikan terus ditampilkan di SageMaker konsol.

HostingDeployment Operator

HostingDeployment operator mendukung pembuatan dan penghapusan titik akhir, serta memperbarui titik akhir yang ada, untuk inferensi waktu nyata. Operator penerapan hosting merekonsiliasi spesifikasi pekerjaan penerapan hosting yang Anda tentukan SageMaker dengan membuat model, konfigurasi titik akhir, dan titik akhir di. SageMaker Anda dapat mempelajari lebih lanjut tentang SageMaker inferensi dalam dokumentasi SageMaker CreateEndpoint API.

Topik

Konfigurasikan HostingDeployment sumber daya

Unduh file YAMM sampel untuk pekerjaan penerapan hosting menggunakan perintah berikut:

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/master/
samples/xgboost-mnist-hostingdeployment.yaml
```

`xgboost-mnist-hostingdeployment.yaml`File memiliki komponen berikut yang dapat diedit sesuai kebutuhan:

- ProductionVariants. Varian produksi adalah serangkaian contoh yang melayani satu model. SageMaker load-balances antara semua varian produksi sesuai dengan bobot yang ditetapkan.
- Model. Model adalah wadah dan peran eksekusi ARN yang diperlukan untuk melayani model. Ini membutuhkan setidaknya satu wadah.
- Wadah. Sebuah wadah menentukan dataset dan melayani gambar. Jika Anda menggunakan algoritme kustom Anda sendiri alih-alih algoritme yang disediakan oleh SageMaker, kode inferensi harus memenuhi SageMaker persyaratan. Untuk informasi selengkapnya, lihat Menggunakan Algoritme Anda Sendiri dengan SageMaker.

Buat HostingDeployment

Untuk membuat HostingDeployment, gunakan `kubectl` untuk menerapkan file `hosting.yaml` dengan perintah berikut:

```
kubectl apply -f hosting.yaml
```

SageMaker membuat titik akhir dengan konfigurasi yang ditentukan. Anda dikenakan biaya untuk SageMaker sumber daya yang digunakan selama masa akhir Anda. Anda tidak dikenakan biaya apa pun setelah titik akhir Anda dihapus.

Proses pembuatan membutuhkan waktu sekitar 10 menit.

Daftar HostingDeployments

Untuk memverifikasi bahwa HostingDeployment itu dibuat, gunakan perintah berikut:

```
kubectl get hostingdeployments
```

Outputnya akan terlihat seperti berikut:

```
NAME           STATUS     SAGEMAKER-ENDPOINT-NAME
host-xgboost   Creating   host-xgboost-def0e83e0d5f11eaaa450aSMLOGS
```

HostingDeployment nilai status

Kolom status dapat berupa salah satu dari beberapa nilai:

- `SynchronizingK8sJobWithSageMaker`: Operator sedang mempersiapkan untuk membuat titik akhir.
- `ReconcilingEndpoint`: Operator membuat, memperbarui, atau menghapus sumber daya titik akhir. Jika HostingDeployment sisa-sisa dalam keadaan ini, gunakan `kubectl describe` untuk melihat alasannya di `Additional` lapangan.
- `OutOfService`: Titik akhir tidak tersedia untuk menerima permintaan yang masuk.
- `Creating`: CreateEndpointsedang berjalan.
- `Updating`: UpdateEndpointatau UpdateEndpointWeightsAndCapacitiessedang berjalan.
- `SystemUpdating`: Titik akhir sedang menjalani pemeliharaan dan tidak dapat diperbarui atau dihapus atau diskalakan ulang sampai selesai. Operasi pemeliharaan ini tidak mengubah nilai yang ditentukan pelanggan seperti konfigurasi VPC, AWS KMS enkripsi, model, jenis instans, atau jumlah instance.
- `RollingBack`: Titik akhir gagal menaikkan atau menurunkan atau mengubah bobot variannya dan sedang dalam proses memutar kembali ke konfigurasi sebelumnya. Setelah rollback selesai, titik akhir kembali ke status. `InService` Status transisi ini hanya berlaku untuk titik akhir yang mengaktifkan penskalaan otomatis dan sedang mengalami perubahan bobot atau kapasitas varian sebagai bagian dari UpdateEndpointWeightsAndCapacitiespanggilan atau ketika operasi dipanggil secara eksplisit. UpdateEndpointWeightsAndCapacities

- `InService`: Titik akhir tersedia untuk memproses permintaan yang masuk.
- `Deleting`: DeleteEndpointsedang berjalan.
- `Failed`: Titik akhir tidak dapat dibuat, diperbarui, atau diskalakan ulang. Gunakan DescribeEndpoint: FailureReason untuk informasi tentang kegagalan. DeleteEndpointadalah satu-satunya operasi yang dapat dilakukan pada titik akhir yang gagal.

Jelaskan HostingDeployment

Anda dapat memperoleh detail debugging menggunakan `describe` `kubectl` perintah.

```
kubectl describe hostingdeployment
```

Outputnya akan terlihat seperti berikut:

```
Name:         host-xgboost
Namespace:    default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"sagemaker.aws.amazon.com/
v1","kind":"HostingDeployment","metadata":{"annotations":{},"name":"host-
xgboost","namespace":"def..."
API Version:  sagemaker.aws.amazon.com/v1
Kind:         HostingDeployment
Metadata:
  Creation Timestamp:  2019-11-22T19:40:00Z
  Finalizers:
    sagemaker-operator-finalizer
  Generation:        1
  Resource Version:  4258134
  Self Link:         /apis/sagemaker.aws.amazon.com/v1/namespaces/default/
hostingdeployments/host-xgboost
  UID:               def0e83e-0d5f-11ea-aa45-0a3507uiduid
Spec:
  Containers:
    Container Hostname:  xgboost
    Image:               123456789012.dkr.ecr.us-east-2.amazonaws.com/xgboost:latest
    Model Data URL:      s3://my-bucket/inference/xgboost-mnist/model.tar.gz
  Models:
    Containers:
      xgboost
    Execution Role Arn:  arn:aws:iam::123456789012:role/service-role/AmazonSageMaker-
ExecutionRole
```

```
    Name:                xgboost-model
    Primary Container:   xgboost
  Production Variants:
    Initial Instance Count:  1
    Instance Type:           ml.c5.large
    Model Name:              xgboost-model
    Variant Name:            all-traffic
  Region:                    us-east-2
Status:
  Creation Time:         2019-11-22T19:40:04Z
  Endpoint Arn:          arn:aws:sagemaker:us-east-2:123456789012:endpoint/host-
xgboost-def0e83e0d5f11eaaaexample
  Endpoint Config Name:  host-xgboost-1-def0e83e0d5f11e-e08f6c510d5f11eaaa450aexample
  Endpoint Name:         host-xgboost-def0e83e0d5f11eaaa450a350733ba06
  Endpoint Status:       Creating
  Endpoint URL:          https://runtime.sagemaker.us-east-2.amazonaws.com/endpoints/
host-xgboost-def0e83e0d5f11eaaaexample/invocations
  Last Check Time:       2019-11-22T19:43:57Z
  Last Modified Time:    2019-11-22T19:40:04Z
  Model Names:
    Name:   xgboost-model
    Value:  xgboost-model-1-def0e83e0d5f11-df5cc9fd0d5f11eaaa450aexample
Events:     <none>
```

Bidang status memberikan informasi lebih lanjut menggunakan bidang berikut:

- `Additional`: Informasi tambahan tentang status penyebaran hosting. Bidang ini opsional dan hanya diisi jika terjadi kesalahan.
- `Creation Time`: Ketika titik akhir dibuat di SageMaker.
- `Endpoint ARN`: SageMaker Titik akhir ARN.
- `Endpoint Config Name`: SageMaker Nama konfigurasi endpoint.
- `Endpoint Name`: SageMaker Nama titik akhir.
- `Endpoint Status`: Status titik akhir.
- `Endpoint URL`URL HTTPS yang dapat digunakan untuk mengakses endpoint. Untuk informasi selengkapnya, lihat Menerapkan Model pada SageMaker Layanan Hosting.
- `FailureReason`: Jika perintah membuat, memperbarui, atau menghapus gagal, penyebabnya ditampilkan di sini.
- `Last Check Time`: Terakhir kali operator memeriksa status titik akhir.
- `Last Modified Time`: Terakhir kali titik akhir dimodifikasi.
- `Model Names`: Sepasang nilai kunci dari nama HostingDeployment model untuk nama SageMaker model.

Memanggil titik akhir

Setelah status endpoint`InService`, Anda dapat memanggil endpoint dengan dua cara: menggunakan AWS CLI, yang melakukan otentikasi dan penandatanganan permintaan URL, atau menggunakan klien HTTP seperti cURL. Jika Anda menggunakan klien Anda sendiri, Anda perlu melakukan penandatanganan dan otentikasi URL AWS v4 sendiri.

Untuk memanggil titik akhir menggunakan AWS CLI, jalankan perintah berikut. Pastikan untuk mengganti nama Region dan endpoint dengan nama Region dan SageMaker endpoint endpoint Anda. Informasi ini dapat diperoleh dari output`kubectl describe`.

```
# Invoke the endpoint with mock input data.
aws sagemaker-runtime invoke-endpoint \
  --region us-east-2 \
  --endpoint-name <endpoint name> \
  --body $(seq 784 | xargs echo | sed 's/ /,/g') \
  >(cat) \
  --content-type text/csv > /dev/null
```

Misalnya, jika Region Anda `us-east-2` dan nama konfigurasi titik akhir Anda adalah`host-xgboost-f56b6b280d7511ea824b129926example`, maka perintah berikut akan memanggil titik akhir:

```
aws sagemaker-runtime invoke-endpoint \
  --region us-east-2 \
  --endpoint-name host-xgboost-f56b6b280d7511ea824b1299example \
  --body $(seq 784 | xargs echo | sed 's/ /,/g') \
  >(cat) \
  --content-type text/csv > /dev/null
4.95847082138
```

Di sini, `4.95847082138` adalah prediksi dari model untuk data tiruan.

Perbarui HostingDeployment

1. Setelah HostingDeployment memiliki status`InService`, itu dapat diperbarui. Mungkin butuh waktu sekitar 10 menit HostingDeployment untuk bisa beroperasi. Untuk memverifikasi bahwa statusnya`InService`, gunakan perintah berikut:

   ```
   kubectl get hostingdeployments
   ```

2. HostingDeployment Dapat diperbarui sebelum statusnya`InService`. Operator menunggu hingga SageMaker titik akhir `InService` sebelum menerapkan pembaruan.

   Untuk menerapkan pembaruan, ubah `hosting.yaml` file. Misalnya, ubah `initialInstanceCount` bidang dari 1 menjadi 2 sebagai berikut:

```
apiVersion: sagemaker.aws.amazon.com/v1
kind: HostingDeployment
metadata:
  name: host-xgboost
spec:
    region: us-east-2
    productionVariants:
        - variantName: all-traffic
          modelName: xgboost-model
          initialInstanceCount: 2
          instanceType: ml.c5.large
    models:
        - name: xgboost-model
          executionRoleArn: arn:aws:iam::123456789012:role/service-role/
AmazonSageMaker-ExecutionRole
          primaryContainer: xgboost
          containers:
            - xgboost
    containers:
        - containerHostname: xgboost
          modelDataUrl: s3://my-bucket/inference/xgboost-mnist/model.tar.gz
          image: 123456789012.dkr.ecr.us-east-2.amazonaws.com/xgboost:latest
```

3. Simpan file, lalu gunakan `kubectl` untuk menerapkan pembaruan Anda sebagai berikut. Anda akan melihat perubahan status dari `InService` ke`ReconcilingEndpoint`, kemudian`Updating`.

```
$ kubectl apply -f hosting.yaml
hostingdeployment.sagemaker.aws.amazon.com/host-xgboost configured

$ kubectl get hostingdeployments
NAME           STATUS                SAGEMAKER-ENDPOINT-NAME
host-xgboost   ReconcilingEndpoint   host-xgboost-def0e83e0d5f11eaaa450a350abcdef

$ kubectl get hostingdeployments
NAME           STATUS     SAGEMAKER-ENDPOINT-NAME
```

```
host-xgboost   Updating   host-xgboost-def0e83e0d5f11eaaa450a3507abcdef
```

SageMaker menyebarkan serangkaian instance baru dengan model Anda, mengalihkan lalu lintas untuk menggunakan instance baru, dan menguras instance lama. Begitu proses ini dimulai, statusnya menjadi`Updating`. Setelah pembaruan selesai, titik akhir Anda menjadi`InService`. Proses ini memakan waktu sekitar 10 menit.

Hapus HostingDeployment

1. Gunakan `kubectl` untuk menghapus HostingDeployment dengan perintah berikut:

   ```
   kubectl delete hostingdeployments host-xgboost
   ```

   Outputnya akan terlihat seperti berikut:

   ```
   hostingdeployment.sagemaker.aws.amazon.com "host-xgboost" deleted
   ```

2. Untuk memverifikasi bahwa penyebaran hosting telah dihapus, gunakan perintah berikut:

   ```
   kubectl get hostingdeployments
   No resources found.
   ```

Titik akhir yang telah dihapus tidak dikenakan biaya apa pun untuk SageMaker sumber daya.

ProcessingJob Operator

ProcessingJob operator digunakan untuk meluncurkan pekerjaan SageMaker pemrosesan Amazon. Untuk informasi lebih lanjut tentang pekerjaan SageMaker pemrosesan, lihat CreateProcessingJob.

Topik

- Membuat file ProcessingJob menggunakan YAMAL
- Daftar ProcessingJobs
- Jelaskan ProcessingJob
- Menghapus ProcessingJob

Membuat file ProcessingJob menggunakan YAMAL

Ikuti langkah-langkah berikut untuk membuat pekerjaan SageMaker pemrosesan Amazon dengan menggunakan file YAMM:

1. Unduh skrip `kmeans_preprocessing.py` pra-pemrosesan.

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/samples/kmeans_preprocessing.py
```

2. Di salah satu bucket Amazon Simple Storage Service (Amazon S3) Anda, buat `mnist_kmeans_example/processing_code` folder dan unggah skrip ke folder.
3. Mengunduh `kmeans-mnist-processingjob.yaml` file.

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/samples/kmeans-mnist-processingjob.yaml
```

4. Edit file YAMAL untuk menentukan `sagemaker-execution-role` dan mengganti semua instance `my-bucket` dengan bucket S3 Anda.

```
...
metadata:
  name: kmeans-mnist-processing
...
  roleArn: arn:aws:iam::<acct-id>:role/service-role/<sagemaker-execution-role>
  ...
  processingOutputConfig:
    outputs:
      ...
          s3Output:
            s3Uri: s3://<my-bucket>/mnist_kmeans_example/output/
  ...
  processingInputs:
    ...
        s3Input:
          s3Uri: s3://<my-bucket>/mnist_kmeans_example/processing_code/
kmeans_preprocessing.py
```

`sagemaker-execution-role`Harus memiliki izin sehingga SageMaker dapat mengakses bucket S3 CloudWatch, Amazon, dan layanan lainnya atas nama Anda. Untuk informasi selengkapnya tentang membuat peran eksekusi, lihat SageMakerPeran.

5. Terapkan file YAM menggunakan salah satu perintah berikut.

   Untuk instalasi dengan cakupan cluster:

   ```
   kubectl apply -f kmeans-mnist-processingjob.yaml
   ```

   Untuk instalasi dengan cakupan ruang nama:

   ```
   kubectl apply -f kmeans-mnist-processingjob.yaml -n <NAMESPACE>
   ```

Daftar ProcessingJobs

Gunakan salah satu perintah berikut untuk mencantumkan semua pekerjaan yang dibuat menggunakan ProcessingJob operator. `SAGEMAKER-JOB-NAME` berasal dari `metadata` bagian file YAMG.

Untuk instalasi dengan cakupan cluster:

```
kubectl get ProcessingJob kmeans-mnist-processing
```

Untuk instalasi dengan cakupan ruang nama:

```
kubectl get ProcessingJob -n <NAMESPACE> kmeans-mnist-processing
```

Outputnya semestinya mirip dengan yang berikut:

```
NAME                    STATUS     CREATION-TIME        SAGEMAKER-JOB-NAME
kmeans-mnist-processing InProgress 2020-09-22T21:13:25Z kmeans-mnist-
processing-7410ed52fd1811eab19a165ae9f9e385
```

Output mencantumkan semua pekerjaan terlepas dari statusnya. Untuk menghapus pekerjaan dari daftar, lihat Menghapus Pekerjaan Pemrosesan.

ProcessingJob Status

- `SynchronizingK8sJobWithSageMaker`— Pekerjaan pertama kali diserahkan ke cluster. Operator telah menerima permintaan dan sedang bersiap untuk membuat pekerjaan pemrosesan.

- `Reconciling`— Operator menginisialisasi atau memulihkan dari kesalahan sementara, bersama dengan yang lain. Jika pekerjaan pemrosesan tetap dalam keadaan ini, gunakan `kubectl describe` perintah untuk melihat alasannya di `Additional` lapangan.
- `InProgress | Completed | Failed | Stopping | Stopped`— Status pekerjaan SageMaker pemrosesan. Untuk informasi selengkapnya, lihat DescribeProcessingJob.
- `Error`— Operator tidak dapat memulihkan dengan rekonsiliasi.

Pekerjaan yang telah selesai, berhenti, atau gagal tidak dikenakan biaya lebih lanjut untuk SageMaker sumber daya.

Jelaskan ProcessingJob

Gunakan salah satu perintah berikut untuk mendapatkan detail selengkapnya tentang pekerjaan pemrosesan. Perintah ini biasanya digunakan untuk men-debug masalah atau memeriksa parameter pekerjaan pemrosesan.

Untuk instalasi dengan cakupan cluster:

```
kubectl describe processingjob kmeans-mnist-processing
```

Untuk instalasi dengan cakupan ruang nama:

```
kubectl describe processingjob kmeans-mnist-processing -n <NAMESPACE>
```

Output untuk pekerjaan pemrosesan Anda akan terlihat serupa dengan yang berikut ini.

```
$ kubectl describe ProcessingJob kmeans-mnist-processing
Name:         kmeans-mnist-processing
Namespace:    default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"sagemaker.aws.amazon.com/
v1","kind":"ProcessingJob","metadata":{"annotations":{},"name":"kmeans-mnist-
processing",...
API Version:  sagemaker.aws.amazon.com/v1
Kind:         ProcessingJob
Metadata:
  Creation Timestamp:  2020-09-22T21:13:25Z
  Finalizers:
    sagemaker-operator-finalizer
```

```
  Generation:        2
  Resource Version:  21746658
  Self Link:         /apis/sagemaker.aws.amazon.com/v1/namespaces/default/
processingjobs/kmeans-mnist-processing
  UID:               7410ed52-fd18-11ea-b19a-165ae9f9e385
Spec:
  App Specification:
    Container Entrypoint:
      python
      /opt/ml/processing/code/kmeans_preprocessing.py
    Image Uri:  763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-training:1.5.0-
cpu-py36-ubuntu16.04
  Environment:
    Name:   MYVAR
    Value:  my_value
    Name:   MYVAR2
    Value:  my_value2
  Network Config:
  Processing Inputs:
    Input Name:  mnist_tar
    s3Input:
      Local Path:   /opt/ml/processing/input
      s3DataType:   S3Prefix
      s3InputMode:  File
      s3Uri:        s3://<s3bucket>-us-west-2/algorithms/kmeans/mnist/mnist.pkl.gz
    Input Name:     source_code
    s3Input:
      Local Path:   /opt/ml/processing/code
      s3DataType:   S3Prefix
      s3InputMode:  File
      s3Uri:        s3://<s3bucket>/mnist_kmeans_example/processing_code/
kmeans_preprocessing.py
  Processing Output Config:
    Outputs:
      Output Name:  train_data
      s3Output:
        Local Path:    /opt/ml/processing/output_train/
        s3UploadMode:  EndOfJob
        s3Uri:         s3://<s3bucket>/mnist_kmeans_example/output/
      Output Name:     test_data
      s3Output:
        Local Path:    /opt/ml/processing/output_test/
        s3UploadMode:  EndOfJob
        s3Uri:         s3://<s3bucket>/mnist_kmeans_example/output/
```

```
      Output Name:     valid_data
      s3Output:
        Local Path:    /opt/ml/processing/output_valid/
        s3UploadMode:  EndOfJob
        s3Uri:         s3://<s3bucket>/mnist_kmeans_example/output/
  Processing Resources:
    Cluster Config:
      Instance Count:     1
      Instance Type:      ml.m5.xlarge
      Volume Size In GB:  20
  Region:                 us-west-2
  Role Arn:               arn:aws:iam::<acct-id>:role/m-sagemaker-role
  Stopping Condition:
    Max Runtime In Seconds:  1800
  Tags:
    Key:    tagKey
    Value:  tagValue
Status:
  Cloud Watch Log URL:             https://us-west-2.console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logStream:group=/aws/sagemaker/ProcessingJobs;prefix=kmeans-
mnist-processing-7410ed52fd1811eab19a165ae9f9e385;streamFilter=typeLogStreamPrefix
  Last Check Time:                 2020-09-22T21:14:29Z
  Processing Job Status:           InProgress
  Sage Maker Processing Job Name:  kmeans-mnist-
processing-7410ed52fd1811eab19a165ae9f9e385
Events:                            <none>
```

Menghapus ProcessingJob

Saat Anda menghapus pekerjaan pemrosesan, pekerjaan SageMaker pemrosesan akan dihapus dari Kubernetes tetapi pekerjaan tersebut tidak dihapus. SageMaker Jika status pekerjaan di SageMaker `InProgress` adalah pekerjaan dihentikan. Memproses pekerjaan yang dihentikan tidak dikenakan biaya apa pun untuk SageMaker sumber daya. Gunakan salah satu perintah berikut untuk menghapus pekerjaan pemrosesan.

Untuk instalasi dengan cakupan cluster:

```
kubectl delete processingjob kmeans-mnist-processing
```

Untuk instalasi dengan cakupan ruang nama:

```
kubectl delete processingjob kmeans-mnist-processing -n <NAMESPACE>
```

Output untuk pekerjaan pemrosesan Anda akan terlihat serupa dengan yang berikut ini.

```
processingjob.sagemaker.aws.amazon.com "kmeans-mnist-processing" deleted
```

> **Note**
>
> SageMaker tidak menghapus pekerjaan pemrosesan. Pekerjaan yang dihentikan terus ditampilkan di SageMaker konsol. `delete`Perintah membutuhkan beberapa menit untuk membersihkan sumber daya dari SageMaker.

HostingAutoscalingPolicy (HAP) Operator

Operator HostingAutoscalingPolicy (HAP) mengambil daftar ID sumber daya sebagai input dan menerapkan kebijakan yang sama untuk masing-masingnya. Setiap ID sumber daya adalah kombinasi dari nama endpoint dan nama varian. Operator HAP melakukan dua langkah: ini mendaftarkan ID sumber daya dan kemudian menerapkan kebijakan penskalaan ke setiap ID sumber daya. `Delete`membatalkan kedua tindakan tersebut. Anda dapat menerapkan HAP ke SageMaker titik akhir yang ada atau Anda dapat membuat SageMaker titik akhir baru menggunakan operator. HostingDeployment Anda dapat membaca lebih lanjut tentang penskalaan otomatis dalam dokumentasi Kebijakan SageMaker Autoscaling Aplikasi.

> **Note**
>
> Dalam `kubectl` perintah Anda, Anda dapat menggunakan formulir pendek`hap`,, sebagai penggantihostingautoscalingpolicy.

Topik

- Membuat file HostingAutoscalingPolicy menggunakan YAMAL
- Daftar HostingAutoscalingPolicies
- Jelaskan HostingAutoscalingPolicy
- Perbarui HostingAutoscalingPolicy
- Menghapus HostingAutoscalingPolicy
- Memperbarui atau menghapus titik akhir dengan HostingAutoscalingPolicy

Membuat file HostingAutoscalingPolicy menggunakan YAMAL

Gunakan file YAMM untuk membuat HostingAutoscalingPolicy (HAP) yang menerapkan metrik yang telah ditentukan atau kustom ke satu atau beberapa titik akhir. SageMaker

Amazon SageMaker memerlukan nilai tertentu untuk menerapkan penskalaan otomatis ke varian Anda. Jika nilai-nilai ini tidak ditentukan dalam spesifikasi YAMAL, operator HAP menerapkan nilai default berikut.

```
# Do not change
Namespace                   = "sagemaker"
# Do not change
ScalableDimension           = "sagemaker:variant:DesiredInstanceCount"
# Only one supported
PolicyType                  = "TargetTrackingScaling"
# This is the default policy name but can be changed to apply a custom policy
DefaultAutoscalingPolicyName = "SageMakerEndpointInvocationScalingPolicy"
```

Gunakan sampel berikut untuk membuat HAP yang menerapkan metrik yang telah ditentukan atau kustom ke satu atau beberapa titik akhir.

Contoh 1: Menerapkan metrik yang telah ditentukan ke varian titik akhir tunggal

1. Unduh file YAMAL sampel untuk metrik yang telah ditentukan menggunakan perintah berikut:

   ```
   wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
   master/samples/hap-predefined-metric.yaml
   ```

2. Edit file YAMG untuk menentukan file Anda`endpointName`,`variantName`, dan`Region`.
3. Gunakan salah satu perintah berikut untuk menerapkan metrik yang telah ditentukan ke ID sumber daya tunggal (nama titik akhir dan kombinasi nama varian).

   Untuk instalasi dengan cakupan cluster:

   ```
   kubectl apply -f hap-predefined-metric.yaml
   ```

   Untuk instalasi dengan cakupan ruang nama:

   ```
   kubectl apply -f hap-predefined-metric.yaml -n <NAMESPACE>
   ```

Contoh 2: Menerapkan metrik khusus ke varian titik akhir tunggal

1. Unduh contoh file YAMAL untuk metrik kustom menggunakan perintah berikut:

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/samples/hap-custom-metric.yaml
```

2. Edit file YAMG untuk menentukan file Andaendpoint`Name`,`variantName`, dan`Region`.
3. Gunakan salah satu perintah berikut untuk menerapkan metrik kustom ke ID sumber daya tunggal (nama titik akhir dan kombinasi nama varian) sebagai pengganti yang direkomendasikan`SageMakerVariantInvocationsPerInstance`.

> **Note**
>
> Amazon SageMaker tidak memeriksa validitas spesifikasi YAMAL Anda.

Untuk instalasi dengan cakupan cluster:

```
kubectl apply -f hap-custom-metric.yaml
```

Untuk instalasi dengan cakupan ruang nama:

```
kubectl apply -f hap-custom-metric.yaml -n <NAMESPACE>
```

Contoh 3: Menerapkan kebijakan penskalaan ke beberapa titik akhir dan varian

Anda dapat menggunakan operator HAP untuk menerapkan kebijakan penskalaan yang sama ke beberapa ID sumber daya. `scaling_policy`Permintaan terpisah dibuat untuk setiap ID sumber daya (nama titik akhir dan kombinasi nama varian).

1. Unduh file YAMAL sampel untuk metrik yang telah ditentukan menggunakan perintah berikut:

```
wget https://raw.githubusercontent.com/aws/amazon-sagemaker-operator-for-k8s/
master/samples/hap-predefined-metric.yaml
```

2. Edit file YAMM untuk menentukan Anda `Region` `endpointName` dan beberapa serta `variantName` nilai.

3. Gunakan salah satu perintah berikut untuk menerapkan metrik yang telah ditentukan ke beberapa ID sumber daya (nama titik akhir dan kombinasi nama varian).

   Untuk instalasi dengan cakupan cluster:

   ```
   kubectl apply -f hap-predefined-metric.yaml
   ```

   Untuk instalasi dengan cakupan ruang nama:

   ```
   kubectl apply -f hap-predefined-metric.yaml -n <NAMESPACE>
   ```

Pertimbangan HostingAutoscalingPolicies untuk beberapa titik akhir dan varian

Pertimbangan berikut berlaku saat Anda menggunakan beberapa ID sumber daya:

- Jika Anda menerapkan satu kebijakan di beberapa ID sumber daya, satu PolicYarn dibuat per ID sumber daya. Lima titik akhir memiliki lima PolicyArns. Saat Anda menjalankan `describe` perintah pada kebijakan, respons akan muncul sebagai satu pekerjaan dan menyertakan satu status pekerjaan.
- Jika Anda menerapkan metrik kustom ke beberapa ID sumber daya, dimensi atau nilai yang sama akan digunakan untuk semua nilai ID sumber daya (varian). Misalnya, jika Anda menerapkan metrik pelanggan untuk instance 1-5, dan dimensi varian titik akhir dipetakan ke varian 1, ketika varian 1 melebihi metrik, semua titik akhir diskalakan ke atas atau ke bawah.
- Operator HAP mendukung pembaruan daftar ID sumber daya. Jika Anda mengubah, menambah, atau menghapus ID sumber daya ke spesifikasi, kebijakan penskalaan otomatis akan dihapus dari daftar varian sebelumnya dan diterapkan ke kombinasi ID sumber daya yang baru ditentukan. Gunakan `describe`perintah untuk mencantumkan ID sumber daya yang saat ini diterapkan kebijakan.

Daftar HostingAutoscalingPolicies

Gunakan salah satu perintah berikut untuk mencantumkan semua HostingAutoscalingPolicies (HAP) yang dibuat menggunakan operator HAP.

Untuk instalasi dengan cakupan cluster:

```
kubectl get hap
```

Untuk instalasi dengan cakupan ruang nama:

```
kubectl get hap -n <NAMESPACE>
```

Outputnya semestinya mirip dengan yang berikut:

```
NAME             STATUS   CREATION-TIME
hap-predefined   Created  2021-07-13T21:32:21Z
```

Gunakan perintah berikut untuk memeriksa status HostingAutoscalingPolicy (HAP) Anda.

```
kubectl get hap <job-name>
```

Salah satu nilai berikut ditampilkan:

- `Reconciling`— Jenis kesalahan tertentu menunjukkan status sebagai `Reconciling` penggantiError. Beberapa contoh adalah kesalahan sisi server dan titik akhir di negara bagian atau. `Creating` Updating Periksa `Additional` bidang di log status atau operator untuk detail lebih lanjut.
- `Created`
- `Error`

Untuk melihat titik akhir penskalaan otomatis tempat Anda menerapkan kebijakan

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel sisi kiri, perluas Inferensi.
3. Pilih Titik akhir.
4. Nama dari titik akhir yang diinginkan.
5. Gulir ke bagian Pengaturan runtime Endpoint.

Jelaskan HostingAutoscalingPolicy

Gunakan perintah berikut untuk mendapatkan detail lebih lanjut tentang HostingAutoscalingPolicy (HAP). Perintah ini biasanya digunakan untuk men-debug masalah atau memeriksa ID sumber daya (nama endpoint dan kombinasi nama varian) dari HAP.

```
kubectl describe hap <job-name>
```

Perbarui HostingAutoscalingPolicy

Operator HostingAutoscalingPolicy (HAP) mendukung pembaruan. Anda dapat mengedit spesifikasi YAMAL untuk mengubah nilai, lalu menerapkan kembali kebijakan tersebut. Operator HAP menghapus kebijakan yang ada dan menerapkan kebijakan baru.

Menghapus HostingAutoscalingPolicy

Gunakan salah satu perintah berikut untuk menghapus kebijakan HostingAutoscalingPolicy (HAP).

Untuk instalasi dengan cakupan cluster:

```
kubectl delete hap hap-predefined
```

Untuk instalasi dengan cakupan ruang nama:

```
kubectl delete hap hap-predefined -n <NAMESPACE>
```

Perintah ini menghapus kebijakan penskalaan dan membatalkan pendaftaran target penskalaan dari Kubernetes. Perintah ini mengembalikan output berikut:

```
hostingautoscalingpolicies.sagemaker.aws.amazon.com "hap-predefined" deleted
```

Memperbarui atau menghapus titik akhir dengan HostingAutoscalingPolicy

Untuk memperbarui titik akhir yang memiliki HostingAutoscalingPolicy (HAP), gunakan `kubectl delete` perintah untuk menghapus HAP, memperbarui titik akhir, dan kemudian menerapkan kembali HAP.

Untuk menghapus titik akhir yang memiliki HAP, gunakan `kubectl delete` perintah untuk menghapus HAP sebelum Anda menghapus titik akhir.

Migrasikan sumber daya ke Operator terbaru

Kami menghentikan pengembangan dan dukungan teknis dari versi asli SageMaker Operator untuk Kubernetes.

Jika saat ini Anda menggunakan SageMaker Operator untuk Kubernetes versi `v1.2.2` atau di bawah ini, kami sarankan untuk memigrasikan sumber daya Anda ke pengontrol layanan ACK untuk

Amazon. SageMaker Pengontrol layanan ACK adalah generasi baru SageMaker Operator untuk Kubernetes berdasarkan AWSController for Kubernetes (ACK).

Untuk jawaban atas pertanyaan umum di akhir dukungan versi asli SageMaker Operator untuk Kubernetes, lihat Mengumumkan Berakhirnya Support Versi Original SageMaker Operator untuk Kubernetes

Gunakan langkah-langkah berikut untuk memigrasikan sumber daya Anda dan gunakan ACK untuk melatih, menyetel, dan menerapkan model pembelajaran mesin dengan Amazon. SageMaker

> **Note**
>
> SageMaker Operator terbaru untuk Kubernetes tidak kompatibel ke belakang.

Daftar Isi

Prasyarat

Untuk melakukan migrasi sumber daya ke SageMaker Operator terbaru untuk Kubernetes, Anda harus melakukan hal berikut:

1. Instal SageMaker Operator terbaru untuk Kubernetes. Lihat Pengaturan di Machine Learning dengan ACK SageMaker Controller untuk step-by-step instruksi.
2. Jika Anda menggunakanSumber daya HostingAutoscalingPolicy, instal Application Auto Scaling Operator yang baru. Lihat Pengaturan di SageMaker Beban Kerja Skala dengan Application Auto Scaling untuk petunjuk. step-by-step Langkah ini opsional jika Anda tidak menggunakan HostingAutoScalingPolicy sumber daya.

Jika izin dikonfigurasi dengan benar, maka pengontrol SageMaker layanan ACK dapat menentukan spesifikasi dan status AWS sumber daya dan merekonsiliasi sumber daya seolah-olah pengontrol ACK awalnya membuatnya.

Mengadopsi sumber daya

SageMaker Operator baru untuk Kubernetes menyediakan kemampuan untuk mengadopsi sumber daya yang awalnya tidak dibuat oleh ACK Service Controller. Untuk informasi selengkapnya, lihat Mengadopsi Sumber AWS Daya yang Ada di dokumentasi ACK.

Langkah-langkah berikut menunjukkan bagaimana SageMaker Operator baru untuk Kubernetes dapat mengadopsi endpoint yang sudah ada. SageMaker Simpan sampel berikut ini ke file dengan nama`adopt-endpoint-sample.yaml`.

```
apiVersion: services.k8s.aws/v1alpha1
kind: AdoptedResource
metadata:
  name: adopt-endpoint-sample
spec:
  aws:
    # resource to adopt, not created by ACK
    nameOrID: xgboost-endpoint
  kubernetes:
    group: sagemaker.services.k8s.aws
    kind: Endpoint
    metadata:
      # target K8s CR name
      name: xgboost-endpoint
```

Kirim sumber daya kustom (CR) menggunakan`kubectl apply`:

```
kubectl apply -f adopt-endpoint-sample.yaml
```

Gunakan `kubectl describe` untuk memeriksa kondisi status sumber daya yang Anda adopsi.

```
kubectl describe adoptedresource adopt-endpoint-sample
```

Verifikasi bahwa `ACK.Adopted` kondisinya`True`. Outputnya akan terlihat serupa dengan contoh berikut:

```
---
kind: AdoptedResource
metadata:
  annotations:
```

```
    kubectl.kubernetes.io/last-applied-configuration: '{"apiVersion":"services.k8s.aws/
v1alpha1","kind":"AdoptedResource","metadata":{"annotations":{},"name":"xgboost-
endpoint","namespace":"default"},"spec":{"aws":{"nameOrID":"xgboost-
endpoint"},"kubernetes":
{"group":"sagemaker.services.k8s.aws","kind":"Endpoint","metadata":{"name":"xgboost-
endpoint"}}}}'
  creationTimestamp: '2021-04-27T02:49:14Z'
  finalizers:
  - finalizers.services.k8s.aws/AdoptedResource
  generation: 1
  name: adopt-endpoint-sample
  namespace: default
  resourceVersion: '12669876'
  selfLink: "/apis/services.k8s.aws/v1alpha1/namespaces/default/adoptedresources/adopt-
endpoint-sample"
  uid: 35f8fa92-29dd-4040-9d0d-0b07bbd7ca0b
spec:
  aws:
    nameOrID: xgboost-endpoint
  kubernetes:
    group: sagemaker.services.k8s.aws
    kind: Endpoint
    metadata:
      name: xgboost-endpoint
status:
  conditions:
  - status: 'True'
    type: ACK.Adopted
```

Periksa apakah sumber daya Anda ada di klaster Anda:

```
kubectl describe endpoints.sagemaker xgboost-endpoint
```

Sumber daya HostingAutoscalingPolicy

Sumber daya `HostingAutoscalingPolicy` (HAP) terdiri dari beberapa sumber daya Application Auto Scaling: dan. `ScalableTarget` `ScalingPolicy` Saat mengadopsi sumber daya HAP dengan ACK, pertama-tama instal pengontrol Application Auto Scaling. Untuk mengadopsi sumber daya HAP, Anda perlu mengadopsi keduanya `ScalableTarget` dan `ScalingPolicy` sumber daya. Anda dapat menemukan pengindentifikasi sumber daya untuk sumber daya ini dalam status `HostingAutoscalingPolicy` resource ()`status.ResourceIDList`.

HostingDeployment sumber daya

Sumber `HostingDeployment` daya terdiri dari beberapa SageMaker sumber daya:`Endpoint`,`EndpointConfig`, dan masing-masing`Model`. Jika Anda mengadopsi SageMaker titik akhir di ACK, Anda perlu mengadopsi`Endpoint`,`EndpointConfig`, dan masing-masing `Model` secara terpisah. `Model`Nama`Endpoint`,`EndpointConfig`, dan dapat ditemukan dalam status `HostingDeployment` sumber daya (`status.endpointName`,`status.endpointConfigName`, dan`status.modelNames`).

Untuk daftar semua sumber SageMaker daya yang didukung, lihat Referensi ACK API.

Bersihkan sumber daya lama

Setelah SageMaker Operator baru untuk Kubernetes mengadopsi sumber daya Anda, Anda dapat menghapus instalan operator lama dan membersihkan sumber daya lama.

Langkah 1: Copot pemasangan operator lama

Untuk menghapus instalan operator lama, lihatHapus operator.

> **Warning**
>
> Copot pemasangan operator lama sebelum menghapus sumber daya lama.

Langkah 2: Hapus finalizer dan hapus sumber daya lama

> **Warning**
>
> Sebelum menghapus sumber daya lama, pastikan Anda telah menghapus instalan operator lama.

Setelah menghapus instalan operator lama, Anda harus secara eksplisit menghapus finalizer untuk menghapus sumber daya operator lama. Contoh skrip berikut menunjukkan cara menghapus semua pekerjaan pelatihan yang dikelola oleh operator lama di namespace tertentu. Anda dapat menggunakan pola serupa untuk menghapus sumber daya tambahan setelah diadopsi oleh operator baru.

> **Note**
>
> Anda harus menggunakan nama sumber daya lengkap untuk mendapatkan sumber daya. Misalnya, gunakan `kubectl get trainingjobs.sagemaker.aws.amazon.com` sebagai ganti dari `kubectl get trainingjob`.

```
namespace=sagemaker_namespace
training_jobs=$(kubectl get trainingjobs.sagemaker.aws.amazon.com -n $namespace -ojson
 | jq -r '.items | .[] | .metadata.name')

for job in $training_jobs
do
    echo "Deleting $job resource in $namespace namespace"
    kubectl patch trainingjobs.sagemaker.aws.amazon.com $job -n $namespace -p
 '{"metadata":{"finalizers":null}}' --type=merge
    kubectl delete trainingjobs.sagemaker.aws.amazon.com $job -n $namespace
done
```

Gunakan SageMaker Operator baru untuk Kubernetes

Untuk panduan mendalam tentang penggunaan SageMaker Operator baru untuk Kubernetes, lihat Menggunakan SageMaker Operator untuk Kubernetes

Mengumumkan Berakhirnya Support Versi Original SageMaker Operator untuk Kubernetes

Halaman ini mengumumkan akhir dukungan untuk versi asli SageMaker Operator untuk Kubernetes dan memberikan jawaban atas pertanyaan umum serta informasi migrasi tentang pengontrol layanan ACK untuk Amazon SageMaker, generasi baru Operator Kubernetes yang didukung SageMaker penuh. Untuk informasi umum tentang SageMaker Operator baru untuk Kubernetes, lihat. SageMaker Operator Terbaru untuk Kubernetes

Pertanyaan yang Sering Diajukan di Akhir Dukungan

Daftar Isi

- Bagaimana saya bisa memigrasikan beban kerja saya ke SageMaker Operator baru untuk Kubernetes untuk pelatihan dan inferensi?
- Versi ACK mana yang harus saya migrasi?
- Apakah SageMaker Operator awal untuk Kubernetes dan Operator baru (ACK service controller for Amazon SageMaker) secara fungsional setara?

Mengapa kami mengakhiri dukungan untuk versi asli SageMaker Operator untuk Kubernetes?

Pengguna sekarang dapat memanfaatkan pengontrol layanan ACK untuk Amazon SageMaker. Pengontrol layanan ACK adalah generasi baru SageMaker Operator untuk Kubernetes berdasarkan AWSControllers for Kubernetes (ACK), sebuah proyek berbasis komunitas yang dioptimalkan untuk produksi, menstandarisasi cara mengekspos layanan melalui operator Kubernetes. AWS Oleh karena itu kami mengumumkan akhir dukungan (EOS) untuk versi asli (bukan berbasis ACK) dari SageMaker Operator untuk Kubernetes. Dukungan berakhir pada 15 Februari 2023 bersama dengan Amazon Elastic Kubernetes Service Kubernetes 1.21.

Untuk informasi lebih lanjut tentang ACK, lihat sejarah dan prinsip ACK.

Di mana saya dapat menemukan informasi lebih lanjut tentang SageMaker Operator baru untuk Kubernetes dan ACK?

- Untuk informasi selengkapnya tentang SageMaker Operator baru untuk Kubernetes, lihat ACK service controller for Amazon SageMaker GitHub repository atau baca AWS Controllers for Kubernetes Documentation.
- Untuk tutorial tentang cara melatih model pembelajaran mesin dengan pengontrol layanan ACK untuk Amazon SageMaker menggunakan Amazon EKS, lihat SageMaker contoh ini.

  Untuk contoh penskalaan otomatis, lihat Menskalakan SageMaker Beban Kerja dengan Application Auto Scaling.
- Untuk informasi tentang AWS Controller for Kubernetes (ACK), lihat dokumentasi AWSControllers for Kubernetes (ACK).
- Untuk daftar SageMaker sumber daya yang didukung, lihat Referensi ACK API.

Apa arti akhir dukungan (EOS)?

Meskipun pengguna dapat terus menggunakan operator mereka saat ini, kami tidak lagi mengembangkan fitur baru untuk operator, kami juga tidak akan merilis patch atau pembaruan

keamanan untuk masalah apa pun yang ditemukan. `v1.2.2`adalah rilis terakhir dari SageMaker Operator untuk Kubernetes. Pengguna harus memigrasikan beban kerja mereka untuk menggunakan pengontrol layanan ACK untuk Amazon. SageMaker

Bagaimana saya bisa memigrasikan beban kerja saya ke SageMaker Operator baru untuk Kubernetes untuk pelatihan dan inferensi?

Untuk informasi tentang migrasi sumber daya dari yang lama ke SageMaker Operator baru untuk Kubernetes, ikuti. Migrasikan sumber daya ke Operator terbaru

Versi ACK mana yang harus saya migrasi?

Pengguna harus bermigrasi ke versi terbaru dari pengontrol layanan ACK untuk Amazon SageMaker.

Apakah SageMaker Operator awal untuk Kubernetes dan Operator baru (ACK service controller for Amazon SageMaker) secara fungsional setara?

Ya, mereka berada pada paritas fitur.

Beberapa sorotan dari perbedaan penting utama antara kedua versi meliputi:

- Definisi Sumber Daya Kustom (CRD) yang digunakan oleh SageMaker Operator berbasis ACK untuk Kubernetes mengikuti definisi AWS API sehingga tidak kompatibel dengan spesifikasi sumber daya kustom dari SageMaker Operator untuk Kubernetes dalam versi aslinya. Lihat CRD di pengontrol baru atau gunakan panduan migrasi untuk mengadopsi sumber daya dan menggunakan pengontrol baru.
- `Hosting Autoscaling`Kebijakan ini tidak lagi menjadi bagian dari SageMaker Operator baru untuk Kubernetes dan telah dimigrasikan ke Application autoscaling ACK controller. Untuk mempelajari cara menggunakan pengontrol penskalaan otomatis aplikasi untuk mengonfigurasi penskalaan otomatis pada SageMaker Endpoint, ikuti contoh penskalaan otomatis ini.
- `HostingDeployment`Sumber daya digunakan untuk membuat Model, Konfigurasi Titik Akhir, dan Titik Akhir dalam satu CRD. SageMaker Operator baru untuk Kubernetes memiliki CRD terpisah untuk masing-masing sumber daya ini.

SageMaker Komponen untuk Pipa Kubeflow

Dokumen ini menguraikan cara menggunakan SageMaker Komponen untuk Pipelines Kubeflow. Dengan komponen pipeline ini, Anda dapat membuat dan memantau pekerjaan SageMaker pelatihan, penyetelan, penerapan titik akhir, dan transformasi batch native dari Pipelines Kubeflow Anda. Dengan menjalankan lowongan Kubeflow Pipeline SageMaker, Anda memindahkan pekerjaan

pemrosesan dan pelatihan data dari klaster Kubernetes ke layanan terkelola yang dioptimalkan untuk pembelajaran mesin SageMaker. Dokumen ini mengasumsikan pengetahuan sebelumnya tentang Kubernetes dan Kubeflow.

Daftar Isi

Apa itu Pipelines Kubeflow?

Kubeflow Pipelines (KFP) adalah platform untuk membangun dan menerapkan alur kerja machine learning (ML) portabel yang dapat diskalakan berdasarkan kontainer Docker. Platform Kubeflow Pipelines terdiri atas hal berikut ini:

- Antarmuka pengguna (UI) untuk mengelola dan melacak eksperimen, pekerjaan, dan proses.
- Mesin (Argo) untuk penjadwalan alur kerja ML-langkah.
- SDK untuk mendefinisikan dan memanipulasi jaringan pipa dan komponen.
- Notebook untuk berinteraksi dengan sistem menggunakan SDK.

Pipeline adalah deskripsi alur kerja ML yang dinyatakan sebagai grafik asiklik terarah. Setiap langkah dalam alur kerja dinyatakan sebagai komponen Pipeline Kubeflow, yang merupakan modul. AWS SDK for Python (Boto3)

Untuk informasi selengkapnya tentang Pipelines Kubeflow, lihat dokumentasi Pipelines Kubeflow.

Apa saja komponen Pipeline Kubeflow?

Komponen Pipeline Kubeflow adalah sekumpulan kode yang digunakan untuk mengeksekusi satu langkah dari pipeline Kubeflow. Komponen diwakili oleh modul Python yang dibangun ke dalam

gambar Docker. Ketika pipeline berjalan, container komponen akan dipakai pada salah satu node worker di klaster Kubernetes yang menjalankan Kubeflow, dan logika Anda dieksekusi. Komponen pipeline dapat membaca output dari komponen sebelumnya dan membuat output yang dapat dikonsumsi oleh komponen berikutnya dalam pipeline. Komponen-komponen ini membuatnya cepat dan mudah untuk menulis pipeline untuk lingkungan eksperimen dan produksi tanpa harus berinteraksi dengan infrastruktur Kubernetes yang mendasarinya.

Anda dapat menggunakan SageMaker Komponen di pipeline Kubeflow Anda. Daripada merangkum logika Anda dalam wadah khusus, Anda cukup memuat komponen dan mendeskripsikan pipeline Anda menggunakan Kubeflow Pipelines SDK. Saat pipeline berjalan, instruksi Anda diterjemahkan ke dalam SageMaker pekerjaan atau penerapan. Beban kerja kemudian berjalan pada infrastruktur yang dikelola sepenuhnya. SageMaker

Mengapa menggunakan SageMaker Komponen untuk Pipa Kubeflow?

SageMaker Komponen untuk Pipelines Kubeflow menawarkan alternatif untuk meluncurkan pekerjaan intensif komputasi Anda dari. SageMaker Komponen terintegrasi SageMaker dengan portabilitas dan orkestrasi Pipelines Kubeflow. Menggunakan SageMaker Components for Kubeflow Pipelines, Anda dapat membuat dan memantau SageMaker sumber daya Anda sebagai bagian dari alur kerja Pipelines Kubeflow. Setiap pekerjaan di pipeline Anda berjalan, SageMaker bukan klaster Kubernetes lokal yang memungkinkan Anda memanfaatkan SageMaker fitur-fitur utama seperti pelabelan data, penyetelan hiperparameter skala besar dan pekerjaan pelatihan terdistribusi, atau penerapan model yang aman dan dapat diskalakan dengan sekali klik. Parameter pekerjaan, status, log, dan output dari SageMaker masih dapat diakses dari UI Pipelines Kubeflow.

SageMaker Komponen mengintegrasikan SageMaker fitur-fitur utama ke dalam alur kerja ML Anda mulai dari menyiapkan data, membangun, melatih, dan menerapkan model ML. Anda dapat membuat Pipeline Kubeflow yang dibangun seluruhnya menggunakan komponen-komponen ini, atau mengintegrasikan komponen individual ke dalam alur kerja Anda sesuai kebutuhan. Komponen tersedia dalam satu atau dua versi. Setiap versi komponen memanfaatkan backend yang berbeda. Untuk informasi selengkapnya tentang versi tersebut, lihatSageMaker Komponen untuk versi Pipelines Kubeflow.

Tidak ada biaya tambahan jika menggunakan SageMaker Komponen untuk Pipeline Kubeflow. Anda dikenakan biaya untuk SageMaker sumber daya apa pun yang Anda gunakan melalui komponen ini.

SageMaker Komponen untuk versi Pipelines Kubeflow

SageMaker Komponen untuk Pipelines Kubeflow hadir dalam dua versi. Setiap versi memanfaatkan backend yang berbeda untuk membuat dan mengelola sumber daya. SageMaker

- SageMaker Komponen untuk Pipelines Kubeflow versi 1 (v1.x atau di bawah) menggunakan Boto3 () sebagai backend. AWS SDK for Python (Boto3)
- Versi 2 (v2.0.0-alpha2 dan yang lebih baru) dari SageMaker Components for Kubeflow Pipelines menggunakan Operator for Kubernetes (ACK). SageMaker

  AWSmemperkenalkan ACK untuk memfasilitasi cara asli Kubernetes dalam mengelola sumber daya Cloud. AWS ACK mencakup satu set pengontrol AWS khusus layanan, salah satunya adalah pengontrol. SageMaker SageMaker Controller memudahkan pengembang machine learning dan data scientist menggunakan Kubernetes sebagai control plane mereka untuk melatih, menyetel, dan menerapkan model machine learning (ML). SageMaker Untuk informasi selengkapnya, lihat SageMaker Operator untuk Kubernetes

Kedua versi SageMaker Komponen untuk Pipelines Kubeflow didukung. Namun, versi 2 memberikan beberapa keuntungan tambahan. Secara khusus, ia menawarkan:

1. Pengalaman yang konsisten untuk mengelola SageMaker sumber daya Anda dari aplikasi apa pun; apakah Anda menggunakan pipeline Kubeflow, atau Kubernetes CLI (`kubectl`) atau aplikasi Kubeflow lainnya seperti Notebook.
2. Fleksibilitas untuk mengelola dan memantau SageMaker sumber daya Anda di luar alur kerja pipeline Kubeflow.
3. Tidak ada waktu penyiapan untuk menggunakan SageMaker komponen jika Anda menerapkan Kubeflow penuh saat AWS rilis karena SageMaker Operator adalah bagian dari penerapannya.

Daftar SageMaker Komponen untuk Pipeline Kubeflow

Berikut ini adalah daftar semua SageMaker Komponen untuk Pipelines Kubeflow dan versinya yang tersedia. Atau, Anda dapat menemukan semua SageMaker Komponen untuk Pipelines Kubeflow di. GitHub

> **Note**
>
> Kami mendorong pengguna untuk menggunakan Versi 2 dari SageMaker komponen di mana pun itu tersedia.

Komponen Ground Truth

- Ground Truth

  Komponen Ground Truth memungkinkan Anda mengirimkan pekerjaan pelabelan SageMaker Ground Truth langsung dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
SageMaker Komponen Ground Truth Kubeflow Pipelines versi 1	X

- Tim kerja

  Komponen Workteam memungkinkan Anda untuk membuat pekerjaan tim kerja SageMaker pribadi langsung dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
SageMaker buat komponen tim kerja pribadi Kubeflow Pipelines versi 1	X

Komponen pemrosesan data

- Pengolahan

  Komponen Processing memungkinkan Anda mengirimkan pekerjaan pemrosesan SageMaker langsung dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
SageMaker Memproses komponen Pipeline Kubeflow versi 1	X

Komponen pelatihan

- Pelatihan

Komponen Pelatihan memungkinkan Anda mengirimkan pekerjaan SageMaker Pelatihan langsung dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
SageMaker Pelatihan komponen Pipelines Kubeflow versi 1	SageMaker Pelatihan komponen Pipelines Kubeflow versi 2

- Optimasi Hyperparameter

  Komponen Hyperparameter Optimization memungkinkan Anda mengirimkan tugas tuning hyperparameter SageMaker langsung dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
SageMaker optimasi hyperparameter komponen Pipeline Kubeflow versi 1	X

Komponen inferensi

- Menyebarkan Hosting

  Komponen Hosting memungkinkan Anda untuk menerapkan model menggunakan layanan SageMaker hosting dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
SageMaker Layanan Hosting - Buat komponen Endpoint Kubeflow Pipeline versi 1.	Versi 2 dari komponen Hosting terdiri dari tiga sub-komponen yang diperlukan untuk membuat penyebaran hosting. SageMaker • Komponen SageMaker Model Kubeflow Pipelines versi 2 yang bertanggung jawab atas artefak model dan jalur registri gambar model yang berisi kode inferensi. • Komponen SageMaker Endpoint Configuration Kubeflow Pipelines versi 2 yang

Versi 1 komponen	Versi 2 komponen
	bertanggung jawab untuk mendefinisikan konfigurasi titik akhir seperti tipe instance, model, jumlah instance, dan opsi inferensi tanpa server. • Komponen SageMaker Endpoint Kubeflow Pipelines versi 2 yang bertanggung jawab untuk membuat atau memperbarui titik akhir SageMaker seperti yang ditentukan dalam konfigurasi titik akhir.

- Transform Batch

  Komponen Batch Transform memungkinkan Anda menjalankan tugas inferensi untuk seluruh kumpulan data SageMaker dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
SageMaker Komponen Batch Transform Kubeflow Pipeline versi 1	X

- Model Monitor

  Komponen Model Monitor memungkinkan Anda memantau kualitas model pembelajaran SageMaker mesin dalam produksi dari alur kerja Pipelines Kubeflow.

Versi 1 komponen	Versi 2 komponen
X	Komponen Model Monitor terdiri dari empat sub-komponen untuk memantau penyimpan gan dalam suatu model. • Komponen SageMaker Data Quality Job Definition Kubeflow Pipelines versi 2 yang bertanggung jawab untuk memantau penyimpangan kualitas data.

Versi 1 komponen	Versi 2 komponen
	• Komponen SageMaker Model Quality Job Definition Kubeflow Pipelines versi 2 yang bertanggung jawab untuk memantau penyimpangan dalam metrik kualitas model. • Sebuah SageMaker Model Bias Job Definition Kubeflow Pipelines komponen versi 2 bertanggung jawab untuk memantau bias dalam prediksi model. • Sebuah SageMaker Model Explainability Job Definition Kubeflow Pipelines komponen versi 2 yang bertanggung jawab untuk memantau penyimpangan dalam atribusi fitur. Selain itu, untuk pemantauan sesuai jadwal pada frekuensi tertentu, komponen kelima, Komponen SageMaker Monitoring Schedule Kubeflow Pipelines versi 2, bertanggung jawab untuk memantau data yang dikumpulkan dari titik akhir real-time sesuai jadwal. Untuk informasi selengkapnya tentang Amazon SageMaker Model Monitor, lihatMemantau data dan kualitas model.

Izin IAM

Menyebarkan Pipelines Kubeflow dengan SageMaker komponen memerlukan tiga lapisan otentikasi berikut:

- Peran IAM yang memberikan akses node gateway Anda (yang bisa berupa mesin lokal atau instans jarak jauh) ke cluster Amazon Elastic Kubernetes Service (Amazon EKS).

  Pengguna yang mengakses node gateway mengasumsikan peran ini untuk:
  - Buat klaster Amazon EKS dan instal KFP

- Membuat Alur
- Buat bucket Amazon S3 untuk data masukan sampel Anda

Peran tersebut membutuhkan izin berikut:

- CloudWatchLogsFullAccess
- AWSCloudFormationFullAccess
- IAM FullAccess
- AmazonS3 FullAccess
- AmazonEC2 FullAccess
- AdminPolicy Amazoneks (Buat kebijakan ini menggunakan skema dari Contoh Kebijakan Berbasis Identitas Amazon EKS)

- Peran eksekusi IAM Kubernetes yang diasumsikan oleh Kubernetes pipeline pods (kfp-example-pod-role) atau Pod SageMaker pengendali Operator untuk Kubernetes untuk mengaksesnya. SageMaker Peran ini digunakan untuk membuat dan memantau SageMaker pekerjaan dari Kubernetes.

  Peran tersebut membutuhkan izin berikut:

  - AmazonSageMakerFullAccess

  Anda dapat membatasi izin ke KFP dan pod pengontrol dengan membuat dan melampirkan kebijakan kustom Anda sendiri.

- Peran eksekusi SageMaker IAM diasumsikan oleh SageMaker pekerjaan untuk mengakses AWS sumber daya seperti Amazon S3 atau Amazon ECR kfp-example-sagemaker-execution (-role).

  SageMaker pekerjaan menggunakan peran ini untuk:

  - Akses SageMaker sumber daya
  - Masukan Data dari Amazon S3
  - Simpan model keluaran Anda ke Amazon S3

  Peran tersebut membutuhkan izin berikut:

  - AmazonSageMakerFullAccess
  - AmazonS3 FullAccess

Mengonversi saluran pipa untuk digunakan SageMaker

Anda dapat mengonversi pipeline yang ada untuk digunakan SageMaker dengan mem-porting kontainer pemrosesan Python generik dan wadah pelatihan Anda. Jika Anda menggunakan

SageMaker untuk inferensi, Anda juga perlu melampirkan izin IAM ke cluster Anda dan mengonversi artefak menjadi model.

Instal Alur Kubeflow

Kubeflow Pipelines (KFP) adalah komponen orkestrasi pipa dari Kubeflow.

Anda dapat menerapkan Pipelines Kubeflow (KFP) di Amazon Elastic Kubernetes Service (Amazon EKS) yang sudah ada atau membuat cluster Amazon EKS baru. Gunakan node gateway untuk berinteraksi dengan cluster Anda. Node gateway dapat berupa mesin lokal Anda atau contoh Amazon EC2.

Bagian berikut memandu Anda melalui langkah-langkah untuk menyiapkan dan mengonfigurasi sumber daya ini.

Topik

- Pilih opsi instalasi
- Konfigurasikan izin pipeline Anda untuk mengakses SageMaker
- Akses UI KFP (Dasbor Kubeflow)

Pilih opsi instalasi

Pipelines Kubeflow tersedia sebagai komponen inti dari distribusi penuh Kubeflow pada AWS atau sebagai instalasi mandiri.

Pilih opsi yang berlaku untuk kasus penggunaan Anda:

1. Kubeflow Penuh pada Deployment AWS

   Untuk menggunakan komponen Kubeflow lainnya selain Pipelines Kubeflow, pilih distribusi penuh AWS penerapan Kubeflow.

2. Penyebaran Pipa Kubeflow Mandiri

   Untuk menggunakan Pipelines Kubeflow tanpa komponen lain dari Kubeflow, instal pipeline Kubeflow secara mandiri.

Kubeflow Penuh pada Deployment AWS

Untuk menginstal rilis lengkap Kubeflow aktifAWS, pilih opsi penyebaran vanilla dari Kubeflow pada panduan penerapan atau opsi AWS penerapan lainnya yang mendukung integrasi dengan berbagai layanan (AWSAmazon S3, Amazon RDS, Amazon Cognito).

Penyebaran Pipa Kubeflow Mandiri

Bagian ini mengasumsikan bahwa pengguna Anda memiliki izin untuk membuat peran dan menentukan kebijakan untuk peran tersebut.

Siapkan node gateway

Anda dapat menggunakan mesin lokal atau instans Amazon EC2 sebagai node gateway Anda. Node gateway digunakan untuk membuat klaster Amazon EKS dan mengakses UI Kubeflow Pipelines.

Selesaikan langkah-langkah berikut untuk menyiapkan node Anda.

1. Membuat node gateway.

   Anda dapat menggunakan instans Amazon EC2 yang sudah ada atau membuat instance baru dengan versi DLAMI Ubuntu 18.04 terbaru menggunakan langkah-langkah dalam Meluncurkan dan Mengonfigurasi DLAMI.
2. Buat peran IAM untuk memberikan akses node gateway Anda ke AWS sumber daya.

   Buat peran IAM dengan izin ke sumber daya berikut: CloudWatch,, IAMAWS CloudFormation, Amazon EC2, Amazon S3, Amazon EKS.

   Lampirkan kebijakan berikut ke peran IAM:

   - CloudWatchLogsFullAccess
   - AWSCloudFormationFullAccess
   - IAM FullAccess
   - AmazonS3 FullAccess
   - AmazonEC2 FullAccess
   - AdminPolicy Amazoneks (Buat kebijakan ini menggunakan skema dari Contoh Kebijakan Berbasis Identitas Amazon EKS)

   Untuk informasi tentang menambahkan izin IAM ke peran IAM, lihat Menambahkan dan menghapus izin identitas IAM.

3. Instal alat dan klien berikut

   Instal dan konfigurasikan alat dan sumber daya berikut di node gateway Anda untuk mengakses kluster Amazon EKS dan Antarmuka Pengguna KFP (UI).

   - AWS CLI: Alat baris perintah untuk bekerja dengan AWS layanan. Untuk informasi AWS CLI konfigurasi, lihat Mengonfigurasi. AWS CLI
   - aws-iam-authenticatorversi 0.1.31 dan di atasnya: Sebuah alat untuk menggunakan kredensional AWS IAM untuk mengautentikasi klaster Kubernetes.
   - `eksctl`versi di atas 0,15: Alat baris perintah untuk bekerja dengan cluster Amazon EKS.
   - `kubectl`: Alat baris perintah untuk bekerja dengan klaster Kubernetes. Versi harus cocok dengan versi Kubernetes Anda dalam satu versi minor.
   - `AWS SDK for Python (Boto3)`.

     ```
     pip install boto3
     ```

Atur klaster Amazon EKS

1. Jika Anda tidak memiliki kluster Amazon EKS yang ada, jalankan langkah-langkah berikut dari baris perintah node gateway Anda, lewati langkah ini jika tidak.

   a. Jalankan perintah berikut untuk membuat cluster Amazon EKS dengan versi 1.17 atau lebih tinggi. Ganti `<clustername>` dengan nama apa pun untuk cluster Anda.

      ```
      eksctl create cluster --name <clustername> --region us-east-1 --auto-kubeconfig
       --timeout=50m --managed --nodes=1
      ```

   b. Saat pembuatan klaster selesai, pastikan Anda memiliki akses ke klaster Anda dengan mencantumkan node cluster.

      ```
      kubectl get nodes
      ```

2. Pastikan bahwa `kubectl` konteks saat ini menunjuk ke klaster Anda dengan perintah berikut. Konteks saat ini ditandai dengan tanda bintang (*) di output.

   ```
   kubectl config get-contexts

   CURRENT NAME     CLUSTER
   ```

```
*   <username>@<clustername>.us-east-1.eksctl.io   <clustername>.us-
east-1.eksctl.io
```

3. Jika cluster yang diinginkan tidak dikonfigurasi sebagai default Anda saat ini, perbarui default dengan perintah berikut.

```
aws eks update-kubeconfig --name <clustername> --region us-east-1
```

Instal Alur Kubeflow

Jalankan langkah-langkah berikut dari terminal node gateway Anda untuk menginstal Pipelines Kubeflow di cluster Anda.

1. Instal semua komponen cert-manager.

```
kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/
v1.9.1/cert-manager.yaml
```

2. Instal Pipelines Kubeflow.

```
export PIPELINE_VERSION=2.0.0-alpha.5
kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/env/cert-
manager/cluster-scoped-resources?ref=$KFP_VERSION"
kubectl wait --for condition=established --timeout=60s crd/applications.app.k8s.io
kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/env/cert-
manager/dev?ref=$KFP_VERSION"
```

3. Pastikan layanan Pipelines Kubeflow dan sumber daya terkait lainnya berjalan.

```
kubectl -n kubeflow get all | grep pipeline
```

Outputnya akan terlihat seperti berikut.

```
pod/ml-pipeline-6b88c67994-kdtjv                     1/1     Running            0
         2d
pod/ml-pipeline-persistenceagent-64d74dfdbf-66stk    1/1     Running            0
         2d
pod/ml-pipeline-scheduledworkflow-65bdf46db7-5x9qj   1/1     Running            0
         2d
```

```
pod/ml-pipeline-ui-66cc4cffb6-cmsdb                   1/1     Running            0
         2d
pod/ml-pipeline-viewer-crd-6db65ccc4-wqlzj            1/1     Running            0
         2d
pod/ml-pipeline-visualizationserver-9c47576f4-bqmx4   1/1     Running            0
         2d
service/ml-pipeline                         ClusterIP   10.100.170.170   <none>
  8888/TCP,8887/TCP   2d
service/ml-pipeline-ui                      ClusterIP   10.100.38.71     <none>
  80/TCP              2d
service/ml-pipeline-visualizationserver     ClusterIP   10.100.61.47     <none>
  8888/TCP            2d
deployment.apps/ml-pipeline                         1/1     1            1
 2d
deployment.apps/ml-pipeline-persistenceagent        1/1     1            1
 2d
deployment.apps/ml-pipeline-scheduledworkflow       1/1     1            1
 2d
deployment.apps/ml-pipeline-ui                      1/1     1            1
 2d
deployment.apps/ml-pipeline-viewer-crd              1/1     1            1
 2d
deployment.apps/ml-pipeline-visualizationserver     1/1     1            1
 2d
replicaset.apps/ml-pipeline-6b88c67994                      1         1         1
     2d
replicaset.apps/ml-pipeline-persistenceagent-64d74dfdbf     1         1         1
     2d
replicaset.apps/ml-pipeline-scheduledworkflow-65bdf46db7    1         1         1
     2d
replicaset.apps/ml-pipeline-ui-66cc4cffb6                   1         1         1
     2d
replicaset.apps/ml-pipeline-viewer-crd-6db65ccc4            1         1         1
     2d
replicaset.apps/ml-pipeline-visualizationserver-9c47576f4   1         1         1
     2d
```

Konfigurasikan izin pipeline Anda untuk mengakses SageMaker

Di bagian ini, Anda membuat peran eksekusi IAM yang memberikan akses pod Kubeflow Pipeline ke layanan. SageMaker

Konfigurasi untuk SageMaker komponen versi 2

Untuk menjalankan SageMaker Components versi 2 untuk Pipelines Kubeflow, Anda perlu menginstal SageMaker Operator untuk Kubernetes dan mengkonfigurasi Role-Based Access Control (RBAC) yang memungkinkan pod Kubeflow Pipelines untuk membuat resource kustom di klaster Kubernetes Anda. SageMaker

> **Important**
>
> Ikuti bagian ini jika Anda menggunakan penerapan mandiri pipeline Kubeflow. Jika Anda menggunakan AWS distribusi Kubeflow versi 1.6.0-aws-b1.0.0 atau yang lebih baru, komponen versi 2 sudah disiapkan. SageMaker

1. Instal SageMaker Operator untuk Kubernetes untuk menggunakan SageMaker komponen versi 2.

   Ikuti bagian Setup dari Machine Learning with ACK SageMaker Controller tutorial.

2. Konfigurasikan izin RBAC untuk peran eksekusi (akun layanan) yang digunakan oleh pod Kubeflow Pipelines. Dalam penerapan mandiri Kubeflow Pipelines, pipeline run dijalankan di `kubeflow` namespace menggunakan akun layanan. `pipeline-runner`

   a. Buat RoleBindingyang memberikan izin akun layanan untuk mengelola sumber daya SageMaker khusus.

```
cat > manage_sagemaker_cr.yaml <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
name: manage-sagemaker-cr
namespace: kubeflow
subjects:
- kind: ServiceAccount
name: pipeline-runner
namespace: kubeflow
roleRef:
kind: ClusterRole
name: ack-sagemaker-controller
apiGroup: rbac.authorization.k8s.io
EOF
```

```
kubectl apply -f manage_sagemaker_cr.yaml
```

b. Pastikan rolebinding dibuat dengan menjalankan:

```
kubectl get rolebinding manage-sagemaker-cr -n kubeflow -o yaml
```

Konfigurasi untuk SageMaker komponen versi 1

Untuk menjalankan SageMaker Components versi 1 untuk Pipelines Kubeflow, pod Pipeline Kubeflow memerlukan akses ke. SageMaker

> **Important**
>
> Ikuti bagian ini apakah Anda menggunakan Kubeflow penuh saat AWS penerapan atau Kubeflow Pilepines mandiri.

Untuk membuat peran eksekusi IAM yang memberikan akses ke Pod pipeline Kubeflow SageMaker, ikuti langkah-langkah berikut:

1. Ekspor nama cluster Anda (misalnya, my-cluster-name) dan wilayah cluster (misalnya, us-east-1).

```
export CLUSTER_NAME=my-cluster-name
export CLUSTER_REGION=us-east-1
```

2. Ekspor namespace dan nama akun layanan sesuai dengan instalasi Anda.

   - Untuk Kubeflow lengkap saat AWS instalasi, ekspor profil Anda `namespace` (mis., kubeflow-user-example-com) dan default-editor sebagai akun layanan.

```
export NAMESPACE=kubeflow-user-example-com
export KUBEFLOW_PIPELINE_POD_SERVICE_ACCOUNT=default-editor
```

   - Untuk penerapan Pipelines mandiri, ekspor kubeflow sebagai *`namespace`* dan pipeline-runner sebagai akun layanan.

```
export NAMESPACE=kubeflow
export KUBEFLOW_PIPELINE_POD_SERVICE_ACCOUNT=pipeline-runner
```

3. Buat penyedia IAM OIDC untuk klaster Amazon EKS dengan perintah berikut.

```
eksctl utils associate-iam-oidc-provider --cluster ${CLUSTER_NAME} \
            --region ${CLUSTER_REGION} --approve
```

4. Buat peran eksekusi IAM untuk pod KFP untuk mengakses AWS layanan (SageMaker,). CloudWatch

```
eksctl create iamserviceaccount \
--name ${KUBEFLOW_PIPELINE_POD_SERVICE_ACCOUNT} \
--namespace ${NAMESPACE} --cluster ${CLUSTER_NAME} \
--region ${CLUSTER_REGION} \
--attach-policy-arn arn:aws:iam::aws:policy/AmazonSageMakerFullAccess \
--attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
--override-existing-serviceaccounts \
--approve
```

Setelah izin pipeline Anda dikonfigurasi untuk mengakses SageMaker Komponen versi 1, ikuti panduan SageMaker komponen untuk pipeline Kubeflow pada dokumentasi Kubeflow. AWS

Akses UI KFP (Dasbor Kubeflow)

Kubeflow Pipelines UI digunakan untuk mengelola dan melacak eksperimen, pekerjaan, dan berjalan di klaster Anda. Untuk petunjuk tentang cara mengakses UI Pipelines Kubeflow dari node gateway Anda, ikuti langkah-langkah yang berlaku untuk opsi penerapan Anda di bagian ini.

Kubeflow Penuh pada Deployment AWS

Ikuti petunjuk di AWSsitus web Kubeflow untuk terhubung ke dasbor Kubeflow dan menavigasi ke tab pipelines.

Penyebaran Pipa Kubeflow Mandiri

Gunakan port forwarding untuk mengakses Kubeflow Pipelines UI dari node gateway Anda dengan mengikuti langkah-langkah tersebut.

Mengatur penerusan port ke layanan KFP UI

Jalankan perintah berikut dari baris perintah node gateway Anda.

1. Verifikasi bahwa layanan UI KFP sedang berjalan menggunakan perintah berikut.

```
kubectl -n kubeflow get service ml-pipeline-ui

NAME             TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
ml-pipeline-ui   ClusterIP   10.100.38.71   <none>        80/TCP    2d22h
```

2. Jalankan perintah berikut untuk mengatur penerusan port ke layanan KFP UI. Ini meneruskan UI KFP ke port 8080 pada node gateway Anda dan memungkinkan Anda untuk mengakses UI KFP dari browser Anda.

```
kubectl port-forward -n kubeflow service/ml-pipeline-ui 8080:80
```

   Port maju dari mesin jarak jauh Anda turun jika tidak ada aktivitas. Jalankan perintah ini lagi jika dasbor Anda tidak bisa mendapatkan log atau pembaruan. Jika perintah mengembalikan kesalahan, pastikan tidak ada proses yang sudah berjalan pada port yang Anda coba gunakan.

Akses layanan UI KFP

Metode Anda untuk mengakses UI KFP tergantung pada jenis node gateway Anda.

- Mesin lokal sebagai node gateway:

  1. Akses dasbor di browser Anda sebagai berikut:

```
http://localhost:8080
```

  2. Pilih Pipelines untuk mengakses UI pipelines.

- Instans Amazon EC2 sebagai node gateway:

  1. Anda perlu menyiapkan terowongan SSH di instans Amazon EC2 Anda untuk mengakses dasbor Kubeflow dari browser mesin lokal Anda.

     Dari sesi terminal baru di mesin lokal Anda, jalankan yang berikut ini. Ganti `<public-DNS-of-gateway-node>` dengan alamat IP instans Anda yang ditemukan di konsol Amazon EC2. Anda juga dapat menggunakan DNS publik. Ganti `<path_to_key>` dengan jalur ke kunci pem yang digunakan untuk mengakses node gateway.

```
public_DNS_address=<public-DNS-of-gateway-node>
key=<path_to_key>
```

```
on Ubuntu:
ssh -i ${key} -L 9000:localhost:8080 ubuntu@${public_DNS_address}

or on Amazon Linux:
ssh -i ${key} -L 9000:localhost:8080 ec2-user@${public_DNS_address}
```

2. Akses dasbor di peramban Anda.

```
http://localhost:9000
```

3. Pilih Pipelines untuk mengakses UI KFP.

(Opsional) Berikan instans SageMaker notebook akses ke Amazon EKS, dan jalankan pipeline KFP dari notebook Anda.

Instans SageMaker notebook adalah instans komputasi Amazon EC2 yang dikelola sepenuhnya yang menjalankan Aplikasi Notebook Jupyter. Anda dapat menggunakan instance notebook untuk membuat dan mengelola notebook Jupyter lalu menentukan, mengkompilasi, menyebarkan, dan menjalankan pipeline KFP Anda menggunakan atau KFP CLI. AWS SDK for Python (Boto3)

1. Ikuti langkah-langkah dalam Membuat Instance SageMaker Notebook untuk membuat instance notebook, lalu lampirkan `S3FullAccess` kebijakan tersebut ke peran eksekusi IAM-nya.
2. Dari baris perintah node gateway Anda, jalankan perintah berikut untuk mengambil ARN peran IAM dari instance notebook yang Anda buat. Ganti `<instance-name>` dengan nama instans Anda.

```
aws sagemaker describe-notebook-instance --notebook-instance-name <instance-name>
 --region <region> --output text --query 'RoleArn'
```

   Perintah ini menampilkan peran IAM ARN dalam format. `arn:aws:iam::<account-id>:role/<role-name>` Perhatikan ARN ini.
3. Jalankan perintah ini untuk melampirkan kebijakan berikut (AmazonSageMakerFullAccess, AmazonEks, WorkerNodePolicy FullAccess AmazonS3) ke peran IAM ini. Ganti `<role-name>` dengan `<role-name>` di ARN Anda.

```
aws iam attach-role-policy --role-name <role-name> --policy-arn
 arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

```
aws iam attach-role-policy --role-name <role-name> --policy-arn
 arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
aws iam attach-role-policy --role-name <role-name> --policy-arn
 arn:aws:iam::aws:policy/AmazonS3FullAccess
```

4. Cluster Amazon EKS menggunakan peran IAM untuk mengontrol akses ke cluster. Aturan diimplementasikan dalam peta konfigurasi bernama`aws-auth`. `eksctl`menyediakan perintah untuk membaca dan mengedit peta `aws-auth` konfigurasi. Hanya pengguna yang memiliki akses ke cluster yang dapat mengedit peta konfigurasi ini.

   `system:masters`adalah salah satu grup pengguna default dengan izin pengguna super ke cluster. Tambahkan pengguna Anda ke grup ini atau buat grup dengan izin yang lebih ketat.

5. Ikat peran ke klaster Anda dengan menjalankan perintah berikut. Ganti `<IAM-Role-arn>` dengan ARN dari IAM role. `<your_username>`dapat berupa nama pengguna yang unik.

```
eksctl create iamidentitymapping \
--cluster <cluster-name> \
--arn <IAM-Role-arn> \
--group system:masters \
--username <your-username> \
--region <region>
```

6. Buka notebook Jupyter pada SageMaker instance Anda dan jalankan perintah berikut untuk memastikan bahwa ia memiliki akses ke cluster.

```
aws eks --region <region> update-kubeconfig --name <cluster-name>
kubectl -n kubeflow get all | grep pipeline
```

Gunakan SageMaker komponen

Dalam tutorial ini, Anda menjalankan pipeline menggunakan SageMaker Components for Kubeflow Pipelines untuk melatih model klasifikasi menggunakan Kmeans dengan dataset MNIST aktif. SageMaker Alur kerja menggunakan Pipelines Kubeflow sebagai orkestrator dan SageMaker untuk mengeksekusi setiap langkah alur kerja. Contoh diambil dari contoh yang ada SageMaker dan dimodifikasi untuk bekerja dengan SageMaker Components for Kubeflow Pipelines.

Anda dapat menentukan pipeline Anda dengan Python menggunakan AWS SDK for Python (Boto3) kemudian menggunakan dasbor KFP, KFP CLI, atau Boto3 untuk mengkompilasi, menyebarkan,

dan menjalankan alur kerja Anda. Kode lengkap untuk contoh pipeline klasifikasi MNIST tersedia di repositori Kubeflow Github. Untuk menggunakannya, kloning file Python ke node gateway Anda.

Anda dapat menemukan contoh Pipelines SageMaker Kubeflow tambahan di. GitHub Untuk informasi tentang komponen yang digunakan, lihat GitHub repositori KubeFlow Pipelines.

Untuk menjalankan contoh pipeline klasifikasi, buat peran eksekusi SageMaker IAM yang memberikan izin kepada tugas pelatihan Anda untuk mengakses AWS sumber daya, lalu lanjutkan dengan langkah-langkah yang sesuai dengan opsi penerapan Anda.

Untuk membuat peran SageMaker eksekusi

Peran `kfp-example-sagemaker-execution-role` IAM adalah peran runtime yang diasumsikan oleh SageMaker pekerjaan untuk mengakses AWS sumber daya. Dalam perintah berikut, Anda membuat peran eksekusi IAM bernama`kfp-example-sagemaker-execution-role`, melampirkan dua kebijakan terkelola (AmazonSageMakerFullAccess, AmazonS3FullAccess), dan membuat hubungan kepercayaan dengan SageMaker untuk memberikan SageMaker lowongan akses ke sumber daya tersebut. AWS

Anda memberikan peran ini sebagai parameter input saat menjalankan pipeline.

Jalankan perintah berikut untuk membuat peran. Catat ARN yang dikembalikan dalam output Anda.

```
SAGEMAKER_EXECUTION_ROLE_NAME=kfp-example-sagemaker-execution-role

TRUST="{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow
\", \"Principal\": { \"Service\": \"sagemaker.amazonaws.com\" }, \"Action\":
 \"sts:AssumeRole\" } ] }"
aws iam create-role --role-name ${SAGEMAKER_EXECUTION_ROLE_NAME} --assume-role-policy-
document "$TRUST"
aws iam attach-role-policy --role-name ${SAGEMAKER_EXECUTION_ROLE_NAME} --policy-arn
 arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
aws iam attach-role-policy --role-name ${SAGEMAKER_EXECUTION_ROLE_NAME} --policy-arn
 arn:aws:iam::aws:policy/AmazonS3FullAccess

aws iam get-role --role-name ${SAGEMAKER_EXECUTION_ROLE_NAME} --output text --query
 'Role.Arn'
```

Kubeflow Penuh pada Deployment AWS

Ikuti instruksi tutorial Pipeline SageMaker Pelatihan untuk Klasifikasi MNIST dengan K-Means.

Penyebaran Pipa Kubeflow Mandiri

Membuat Alur

Untuk menjalankan pipeline, Anda perlu mengunggah skrip pra-pemrosesan ekstraksi data ke bucket Amazon S3. Bucket ini dan semua sumber daya untuk contoh ini harus berada di `us-east-1` wilayah tersebut. Untuk informasi tentang membuat bucket, lihat Membuat bucket.

Dari `mnist-kmeans-sagemaker` folder repositori Kubeflow yang Anda kloning pada node gateway Anda, jalankan perintah berikut untuk mengunggah file ke bucket Amazon S3 `kmeans_preprocessing.py` Anda. Ubah `<bucket-name>` ke nama bucket Amazon S3.

```
aws s3 cp mnist-kmeans-sagemaker/kmeans_preprocessing.py s3://<bucket-name>/
mnist_kmeans_example/processing_code/kmeans_preprocessing.py
```

Kompilasi dan terapkan pipeline Anda

Setelah mendefinisikan pipeline, Anda harus mengompilasinya ke representasi perantara sebelum Anda mengirimkannya ke layanan Pipelines Kubeflow di klaster Anda. Representasi perantara adalah spesifikasi alur kerja dalam bentuk file YAMAL yang dikompresi menjadi file tar.gz. Anda memerlukan SDK KFP untuk mengkompilasi pipeline Anda.

Instal

Jalankan berikut ini dari baris perintah node gateway Anda:

1. Instal SDK KFP mengikuti petunjuk dalam dokumentasi pipeline Kubeflow.
2. Lakukan verifikasi apakah KFP SDK diinstal dengan perintah berikut:

   ```
   pip show kfp
   ```

3. Verifikasi bahwa `dsl-compile` telah diinstal dengan benar sebagai berikut:

   ```
   which dsl-compile
   ```

Kompilasi alur Anda

Anda memiliki tiga opsi untuk berinteraksi dengan Pipelines Kubeflow: KFP UI, KFP CLI, atau KFP SDK. Bagian berikut menggambarkan alur kerja menggunakan UI KFP dan CLI.

Selesaikan langkah-langkah berikut dari simpul gateway Anda.

1. Ubah file Python Anda dengan nama bucket Amazon S3 dan ARN peran IAM.
2. Gunakan `dsl-compile` perintah dari baris perintah untuk mengkompilasi pipeline Anda sebagai berikut. Ganti `<path-to-python-file>` dengan jalur ke pipeline Anda dan `<path-to-output>` dengan lokasi di mana Anda ingin file tar.gz Anda berada.

```
dsl-compile --py <path-to-python-file> --output <path-to-output>
```

Unggah dan jalankan pipeline menggunakan KFP CLI

Selesaikan langkah-langkah berikut dari baris perintah node gateway Anda. KFP mengatur proses pipa Anda sebagai eksperimen. Anda memiliki opsi untuk menentukan nama eksperimen. Jika Anda tidak menentukannya, proses akan terdaftar di bawah Eksperimen default.

1. Unggah pipeline Anda sebagai berikut:

```
kfp pipeline upload --pipeline-name <pipeline-name> <path-to-output-tar.gz>
```

Outputnya akan terlihat seperti berikut. Perhatikan pipa tersebut`ID`.

```
Pipeline 29c3ff21-49f5-4dfe-94f6-618c0e2420fe has been submitted

Pipeline Details
------------------
ID           29c3ff21-49f5-4dfe-94f6-618c0e2420fe
Name         sm-pipeline
Description
Uploaded at  2020-04-30T20:22:39+00:00
...
...
```

2. Buat proses menggunakan perintah berikut. Perintah KFP CLI run saat ini tidak mendukung menentukan parameter input saat membuat run. Anda perlu memperbarui parameter Anda di file AWS SDK for Python (Boto3) pipeline sebelum dikompilasi. Ganti `<experiment-name>` dan `<job-name>` dengan nama apa pun. Ganti `<pipeline-id>` dengan ID alur yang Anda kirimkan. Ganti `<your-role-arn>` dengan ARN dari. `kfp-example-pod-role` Ganti `<your-bucket-name>` dengan nama bucket Amazon S3 yang Anda buat.

```
kfp run submit --experiment-name <experiment-name> --run-name <job-name> --
pipeline-id <pipeline-id> role_arn="<your-role-arn>" bucket_name="<your-bucket-
name>"
```

Anda juga dapat langsung mengirimkan run menggunakan paket pipeline yang dikompilasi yang dibuat sebagai output dari `dsl-compile` perintah.

```
kfp run submit --experiment-name <experiment-name> --run-name <job-name> --package-
file <path-to-output> role_arn="<your-role-arn>" bucket_name="<your-bucket-name>"
```

Outputnya akan terlihat seperti berikut:

```
Creating experiment aws.
Run 95084a2c-f18d-4b77-a9da-eba00bf01e63 is submitted
+--------------------------------------+--------+----------
+--------------------------+
| run id                               | name   | status   | created at
    |
+======================================+========+==========
+==========================+
| 95084a2c-f18d-4b77-a9da-eba00bf01e63 | sm-job |          |
 2020-04-30T20:36:41+00:00 |
+--------------------------------------+--------+----------
+--------------------------+
```

3. Arahkan ke UI untuk memeriksa kemajuan pekerjaan.

Unggah dan jalankan pipeline menggunakan UI KFP

1. Di panel kiri, pilih tab Pipelines.
2. Di sudut kanan atas, pilih +. UploadPipeline
3. Masukkan nama pipa dan deskripsi.
4. Pilih Unggah file dan masukkan path ke file tar.gz yang Anda buat menggunakan CLI atau dengan. AWS SDK for Python (Boto3)
5. Di panel kiri, pilih tab Pipelines.
6. Temukan pipeline yang Anda buat.
7. Pilih + CreateRun.

8. Masukkan parameter input Anda.
9. Pilih Jalankan.

Jalankan prediksi

Setelah pipeline klasifikasi diterapkan, Anda dapat menjalankan prediksi klasifikasi terhadap titik akhir yang dibuat oleh komponen Deploy. Gunakan UI KFP untuk memeriksa artefak keluaran. `sagemaker-deploy-model-endpoint_name` Unduh file.tgz untuk mengekstrak nama titik akhir atau periksa SageMaker konsol di wilayah yang Anda gunakan.

Konfigurasikan izin untuk menjalankan prediksi

Jika Anda ingin menjalankan prediksi dari node gateway Anda, lewati bagian ini.

Untuk menggunakan mesin lain untuk menjalankan prediksi, tetapkan **`sagemaker:InvokeEndpoint`** izin ke peran IAM yang digunakan oleh mesin klien.

1. Di node gateway Anda, jalankan yang berikut ini untuk membuat file kebijakan IAM:

```
cat <<EoF > ./sagemaker-invoke.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:InvokeEndpoint"
            ],
            "Resource": "*"
        }
    ]
}
EoF
```

2. Lampirkan kebijakan ke peran IAM simpul klien.

   Jalankan perintah berikut. Ganti <your-instance-IAM-role> dengan nama IAM role. Ganti <path-to-sagemaker-invoke-json> dengan jalur ke file kebijakan yang Anda buat.

```
aws iam put-role-policy --role-name <your-instance-IAM-role> --policy-name
 sagemaker-invoke-for-worker --policy-document file://<path-to-sagemaker-invoke-
json>
```

Jalankan prediksi

1. Buat AWS SDK for Python (Boto3) file dari mesin klien Anda bernama `mnist-predictions.py` dengan konten berikut. Ganti `ENDPOINT_NAME` variabelnya. Skrip memuat dataset MNIST, membuat CSV dari digit tersebut, lalu mengirimkan CSV ke titik akhir untuk prediksi dan mencetak hasilnya.

```
import boto3
import gzip
import io
import json
import numpy
import pickle

ENDPOINT_NAME='<endpoint-name>'
region = boto3.Session().region_name

# S3 bucket where the original mnist data is downloaded and stored
downloaded_data_bucket = f"jumpstart-cache-prod-{region}"
downloaded_data_prefix = "1p-notebooks-datasets/mnist"

# Download the dataset
s3 = boto3.client("s3")
s3.download_file(downloaded_data_bucket, f"{downloaded_data_prefix}/mnist.pkl.gz",
 "mnist.pkl.gz")

# Load the dataset
with gzip.open('mnist.pkl.gz', 'rb') as f:
    train_set, valid_set, test_set = pickle.load(f, encoding='latin1')

# Simple function to create a csv from our numpy array
def np2csv(arr):
    csv = io.BytesIO()
    numpy.savetxt(csv, arr, delimiter=',', fmt='%g')
    return csv.getvalue().decode().rstrip()
```

```
runtime = boto3.Session(region).client('sagemaker-runtime')

payload = np2csv(train_set[0][30:31])

response = runtime.invoke_endpoint(EndpointName=ENDPOINT_NAME,
                                   ContentType='text/csv',
                                   Body=payload)
result = json.loads(response['Body'].read().decode())
print(result)
```

2. Jalankan AWS SDK for Python (Boto3) file sebagai berikut:

```
python mnist-predictions.py
```

Lihat hasil dan log

Saat pipeline berjalan, Anda dapat memilih komponen apa pun untuk memeriksa detail eksekusi, seperti input dan output. Ini mencantumkan nama-nama sumber daya yang dibuat.

Jika permintaan KFP berhasil diproses dan SageMaker pekerjaan dibuat, log komponen di UI KFP menyediakan tautan ke pekerjaan yang dibuat di. SageMaker CloudWatch Log juga disediakan jika pekerjaan berhasil dibuat.

Jika Anda menjalankan terlalu banyak pekerjaan pipeline pada klaster yang sama, Anda mungkin akan melihat pesan kesalahan yang menunjukkan bahwa Anda tidak memiliki cukup Pod yang tersedia. Untuk memperbaikinya, masuk ke node gateway Anda dan hapus pod yang dibuat oleh pipeline yang tidak Anda gunakan:

```
kubectl get pods -n kubeflow
kubectl delete pods -n kubeflow <name-of-pipeline-pod>
```

Pembersihan

Setelah Anda selesai dengan alur, Anda perlu membersihkan sumber daya Anda.

1. Dari dasbor KFP, hentikan proses pipeline Anda jika tidak keluar dengan benar dengan memilih Terminate.
2. Jika opsi Terminate tidak berfungsi, masuk ke node gateway Anda dan hentikan secara manual semua pod yang dibuat oleh pipeline Anda yang dijalankan sebagai berikut:

```
kubectl get pods -n kubeflow
kubectl delete pods -n kubeflow <name-of-pipeline-pod>
```

3. Menggunakan AWS akun Anda, masuk ke SageMaker layanan. Hentikan semua pelatihan, transformasi batch, dan pekerjaan HPO secara manual. Hapus model, bucket data, dan titik akhir untuk menghindari biaya tambahan. Mengakhiri jalur pipa tidak menghentikan pekerjaan di SageMaker.

SageMaker Lowongan Notebook

Anda dapat menggunakan Amazon SageMaker untuk membuat, melatih, dan menerapkan model pembelajaran mesin secara interaktif dari notebook Jupyter di lingkungan apa pun. JupyterLab Namun, ada berbagai skenario di mana Anda mungkin ingin menjalankan notebook Anda sebagai pekerjaan terjadwal yang tidak interaktif. Misalnya, Anda mungkin ingin membuat laporan audit reguler yang menganalisis semua pekerjaan pelatihan yang dijalankan selama jangka waktu tertentu dan menganalisis nilai bisnis dari penerapan model tersebut ke dalam produksi. Atau Anda mungkin ingin meningkatkan pekerjaan rekayasa fitur setelah menguji logika transformasi data pada sebagian kecil data. Kasus penggunaan umum lainnya meliputi:

- Penjadwalan pekerjaan untuk pemantauan drift model
- Menjelajahi ruang parameter untuk model yang lebih baik

Dalam skenario ini, Anda dapat menggunakan Pekerjaan SageMaker Notebook untuk membuat pekerjaan noninteraktif (yang SageMaker berjalan sebagai pekerjaan pelatihan dasar) untuk berjalan sesuai permintaan atau sesuai jadwal. SageMaker Pekerjaan Notebook menyediakan antarmuka pengguna yang intuitif sehingga Anda dapat menjadwalkan pekerjaan langsung JupyterLab dengan memilih widget Pekerjaan Notebook ( ) di buku catatan Anda. Anda juga dapat menjadwalkan pekerjaan Anda menggunakan SageMaker Python SDK, yang menawarkan fleksibilitas penjadwalan beberapa pekerjaan notebook dalam alur kerja pipeline. Anda dapat menjalankan beberapa notebook secara paralel, dan membuat parameter sel di notebook Anda untuk menyesuaikan parameter input.

Fitur ini memanfaatkan layanan Amazon EventBridge, SageMaker Training, dan SageMaker Pipelines dan tersedia untuk digunakan di notebook Jupyter Anda di salah satu lingkungan berikut:

- Studio, Studio Lab, Studio Klasik, atau Instans Notebook

- Penyiapan lokal, seperti mesin lokal Anda, tempat Anda menjalankan JupyterLab

Prasyarat

Untuk menjadwalkan pekerjaan notebook, pastikan Anda memenuhi kriteria berikut:

- Pastikan notebook Jupyter Anda dan skrip inisialisasi atau startup apa pun bersifat mandiri sehubungan dengan kode dan paket perangkat lunak. Jika tidak, pekerjaan noninteraktif Anda dapat menimbulkan kesalahan.
- Tinjau Kendala dan pertimbangan untuk memastikan Anda mengonfigurasi notebook Jupyter, pengaturan jaringan, dan pengaturan wadah dengan benar.
- Pastikan notebook Anda dapat mengakses sumber daya eksternal yang diperlukan, seperti klaster EMR Amazon.
- Jika Anda menyiapkan Pekerjaan Notebook di notebook Jupyter lokal, selesaikan penginstalan. Untuk petunjuk, lihat Panduan Instalasi.
- Jika Anda terhubung ke klaster EMR Amazon di notebook dan ingin membuat parameter perintah koneksi EMR Amazon, Anda harus menerapkan solusi menggunakan variabel lingkungan untuk meneruskan parameter. Untuk detailnya, lihat Connect ke klaster Amazon EMR dari notebook Anda.
- Jika Anda terhubung ke kluster EMR Amazon menggunakan otentikasi Kerberos, LDAP, atau HTTP Basic Auth, Anda harus menggunakan untuk meneruskan kredensyal keamanan Anda AWS Secrets Manager ke perintah koneksi Amazon EMR Anda. Untuk detailnya, lihat Connect ke klaster Amazon EMR dari notebook Anda.
- (opsional) Jika Anda ingin UI memuat skrip untuk dijalankan saat startup notebook, admin Anda harus menginstalnya dengan Lifecycle Configuration (LCC). Untuk informasi tentang cara menggunakan skrip LCC, lihat Menyesuaikan Instans Notebook Menggunakan Skrip Konfigurasi Siklus Hidup.

Panduan Instalasi

Diskusi berikut mencakup petunjuk terperinci tentang instalasi tambahan yang perlu Anda lakukan sehingga Anda dapat menggunakan Pekerjaan Notebook di JupyterLab lingkungan Anda.

Untuk Amazon SageMaker Studio dan Amazon SageMaker Studio Lab

Jika notebook Anda ada di Amazon SageMaker Studio atau Amazon SageMaker Studio Lab, Anda tidak perlu melakukan instalasi tambahan— Pekerjaan SageMaker Notebook dibangun ke dalam

platform. Untuk menyiapkan izin yang diperlukan untuk Studio, lihatMenginstal kebijakan dan izin untuk Studio.

Untuk notebook Jupyter lokal

Jika Anda ingin menggunakan Pekerjaan SageMaker Notebook untuk JupyterLab lingkungan lokal Anda, Anda perlu melakukan instalasi tambahan.

Untuk menginstal Pekerjaan SageMaker Notebook, selesaikan langkah-langkah berikut:

1. Menginstal Untuk detailnya, lihat Menginstal Paket Python 3 dan Python.
2. Instal JupyterLab versi 3 atau lebih tinggi. Untuk detailnya, lihat dokumentasi JupyterLab SDK.
3. Instal AWS CLI. Untuk detailnya, lihat Menginstal atau memperbarui versi terbaru AWS CLI.
4. Instal dua set izin. Pengguna IAM memerlukan izin untuk mengirimkan pekerjaan ke SageMaker, dan setelah dikirimkan, pekerjaan notebook itu sendiri mengasumsikan peran IAM yang memerlukan izin untuk mengakses sumber daya tergantung pada tugas pekerjaan.
    a. Jika Anda belum membuat pengguna IAM, lihat Membuat pengguna IAM di akun Anda AWS.
    b. Jika Anda belum membuat peran pekerjaan buku catatan, lihat Membuat peran untuk mendelegasikan izin ke pengguna IAM.
    c. Lampirkan izin dan kebijakan kepercayaan yang diperlukan untuk dilampirkan ke pengguna dan peran Anda. Untuk step-by-step instruksi dan detail izin, lihatMenginstal kebijakan dan izin untuk lingkungan Jupyter lokal.
5. Hasilkan AWS kredensyal untuk pengguna IAM Anda yang baru dibuat dan simpan di file kredensyal (~/.aws/credentials) lingkungan Anda. JupyterLab Anda dapat melakukannya dengan perintah CLI. `aws configure` Untuk petunjuknya, lihat bagian Mengatur dan melihat pengaturan konfigurasi menggunakan perintah di Konfigurasi dan pengaturan file kredensyal.
6. (opsional) Secara default, ekstensi penjadwal menggunakan gambar SageMaker Docker yang sudah dibuat sebelumnya dengan Python 2.0. Kernel non-default apa pun yang digunakan dalam notebook harus dipasang di wadah. Jika Anda ingin menjalankan buku catatan Anda dalam kontainer atau gambar Docker, Anda perlu membuat image Amazon Elastic Container Registry (Amazon ECR). Untuk informasi tentang cara mendorong image Docker ke Amazon ECR, lihat Mendorong Gambar Docker.
7. Tambahkan JupyterLab ekstensi untuk Pekerjaan SageMaker Notebook. Anda dapat menambahkannya ke JupyterLab lingkungan Anda dengan perintah:`pip install`

`amazon_sagemaker_jupyter_scheduler`. Anda mungkin perlu me-restart server Jupyter Anda dengan perintah:. `sudo systemctl restart jupyter-server`

8. Mulailah JupyterLab dengan perintah:`jupyter lab`.
9. Verifikasi bahwa widget
   ( )
   Pekerjaan Notebook muncul di bilah tugas notebook Jupyter Anda.

Menginstal kebijakan dan izin untuk Studio

Sebelum menjadwalkan proses notebook pertama Anda, pastikan Anda menginstal kebijakan dan izin yang tepat. Petunjuk berikut menunjukkan cara mengonfigurasi izin berikut:

- Hubungan kepercayaan peran eksekusi pekerjaan
- Izin IAM tambahan yang dilampirkan pada peran eksekusi pekerjaan
- (opsional) Kebijakan AWS KMS izin untuk menggunakan kunci KMS kustom

> **Important**
>
> Jika AWS akun Anda milik organisasi dengan kebijakan kontrol layanan (SCP), izin efektif Anda adalah persimpangan logis antara apa yang diizinkan oleh SCP dan apa yang diizinkan oleh IAM Anda dan kebijakan pengguna. Misalnya, jika SCP organisasi Anda menetapkan bahwa Anda hanya dapat mengakses sumber daya di `us-east-1` dan`us-west-1`, dan kebijakan Anda hanya memungkinkan Anda untuk mengakses sumber daya di `us-west-1` dan`us-west-2`, pada akhirnya Anda hanya dapat mengakses sumber daya di. `us-west-1` Jika Anda ingin menggunakan semua izin yang diizinkan dalam kebijakan peran dan pengguna, SCP organisasi Anda harus memberikan kumpulan izin yang sama dengan kebijakan pengguna dan peran IAM Anda sendiri. Untuk detail tentang cara menentukan permintaan yang diizinkan, lihat Menentukan apakah permintaan diizinkan atau ditolak dalam akun.

Hubungan kepercayaan

Untuk mengubah hubungan kepercayaan, selesaikan langkah-langkah berikut:

1. Buka konsol IAM.

2. Pilih Peran di panel kiri.
3. Temukan peran eksekusi pekerjaan untuk pekerjaan notebook Anda dan pilih nama peran.
4. Pilih tab Trust relationship.
5. Pilih Edit kebijakan kepercayaan.
6. Salin dan tempel kebijakan berikut ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "events.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

7. Pilih Kebijakan Perbarui.

Izin IAM Tambahan

Anda mungkin perlu menyertakan izin IAM tambahan dalam situasi berikut:

- Eksekusi Studio dan peran pekerjaan notebook Anda berbeda
- Anda perlu mengakses sumber daya Amazon S3 melalui titik akhir VPC S3
- Anda ingin menggunakan kunci KMS kustom untuk mengenkripsi bucket Amazon S3 Anda

Diskusi berikut memberikan kebijakan yang Anda butuhkan untuk setiap kasus.

Izin diperlukan jika eksekusi Studio dan peran pekerjaan notebook Anda berbeda

Cuplikan JSON berikut adalah contoh kebijakan yang harus ditambahkan ke eksekusi Studio dan peran pekerjaan notebook jika Anda tidak menggunakan peran eksekusi Studio sebagai peran pekerjaan notebook. Tinjau dan ubah kebijakan ini jika Anda perlu membatasi hak istimewa lebih lanjut.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"iam:PassRole",
         "Resource":"arn:aws:iam::*:role/*",
         "Condition":{
            "StringLike":{
               "iam:PassedToService":[
                  "sagemaker.amazonaws.com",
                  "events.amazonaws.com"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "events:TagResource",
            "events:DeleteRule",
            "events:PutTargets",
            "events:DescribeRule",
            "events:PutRule",
            "events:RemoveTargets",
            "events:DisableRule",
            "events:EnableRule"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/sagemaker:is-scheduling-notebook-job":"true"
            }
         }
      },
      {
```

```
         "Effect":"Allow",
         "Action":[
            "s3:CreateBucket",
            "s3:PutBucketVersioning",
            "s3:PutEncryptionConfiguration"
         ],
         "Resource":"arn:aws:s3:::sagemaker-automated-execution-*"
      },
      {
            "Sid": "S3DriverAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::sagemakerheadlessexecution-*"
            ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sagemaker:ListTags"
         ],
         "Resource":[
            "arn:aws:sagemaker:*:*:user-profile/*",
            "arn:aws:sagemaker:*:*:space/*",
            "arn:aws:sagemaker:*:*:training-job/*",
            "arn:aws:sagemaker:*:*:pipeline/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sagemaker:AddTags"
         ],
         "Resource":[
            "arn:aws:sagemaker:*:*:training-job/*",
            "arn:aws:sagemaker:*:*:pipeline/*"
         ]
      },
      {
         "Effect":"Allow",
```

```
         "Action":[
            "ec2:CreateNetworkInterface",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:CreateVpcEndpoint",
            "ec2:DeleteNetworkInterface",
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeRouteTables",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcEndpoints",
            "ec2:DescribeVpcs",
            "ecr:BatchCheckLayerAvailability",
            "ecr:BatchGetImage",
            "ecr:GetDownloadUrlForLayer",
            "ecr:GetAuthorizationToken",
            "s3:ListBucket",
            "s3:GetBucketLocation",
            "s3:GetEncryptionConfiguration",
            "s3:PutObject",
            "s3:DeleteObject",
            "s3:GetObject",
            "sagemaker:DescribeDomain",
            "sagemaker:DescribeUserProfile",
            "sagemaker:DescribeSpace",
            "sagemaker:DescribeStudioLifecycleConfig",
            "sagemaker:DescribeImageVersion",
            "sagemaker:DescribeAppImageConfig",
            "sagemaker:CreateTrainingJob",
            "sagemaker:DescribeTrainingJob",
            "sagemaker:StopTrainingJob",
            "sagemaker:Search",
            "sagemaker:CreatePipeline",
            "sagemaker:DescribePipeline",
            "sagemaker:DeletePipeline",
            "sagemaker:StartPipelineExecution"
         ],
         "Resource":"*"
      }
   ]
}
```

Izin yang diperlukan untuk mengakses sumber daya Amazon S3 melalui titik akhir VPC S3

Jika Anda menjalankan SageMaker Studio dalam mode VPC pribadi dan mengakses S3 melalui titik akhir VPC S3, Anda dapat menambahkan izin ke kebijakan titik akhir VPC untuk mengontrol sumber daya S3 mana yang dapat diakses melalui titik akhir VPC. Tambahkan izin berikut ke kebijakan titik akhir VPC Anda. Anda dapat mengubah kebijakan jika perlu membatasi izin lebih lanjut—misalnya, Anda dapat memberikan spesifikasi yang lebih sempit untuk bidang tersebut. `Principal`

```
{
    "Sid": "S3DriverAccess",
    "Effect": "Allow",
    "Principal": "*",
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::sagemakerheadlessexecution-*"
}
```

Untuk detail tentang cara menyiapkan kebijakan titik akhir VPC S3, lihat Mengedit kebijakan titik akhir VPC.

Izin yang diperlukan untuk menggunakan kunci KMS kustom (opsional)

Secara default, bucket input dan output Amazon S3 dienkripsi menggunakan enkripsi sisi server, tetapi Anda dapat menentukan kunci KMS khusus untuk mengenkripsi data Anda di bucket Amazon S3 keluaran dan volume penyimpanan yang dilampirkan ke pekerjaan notebook.

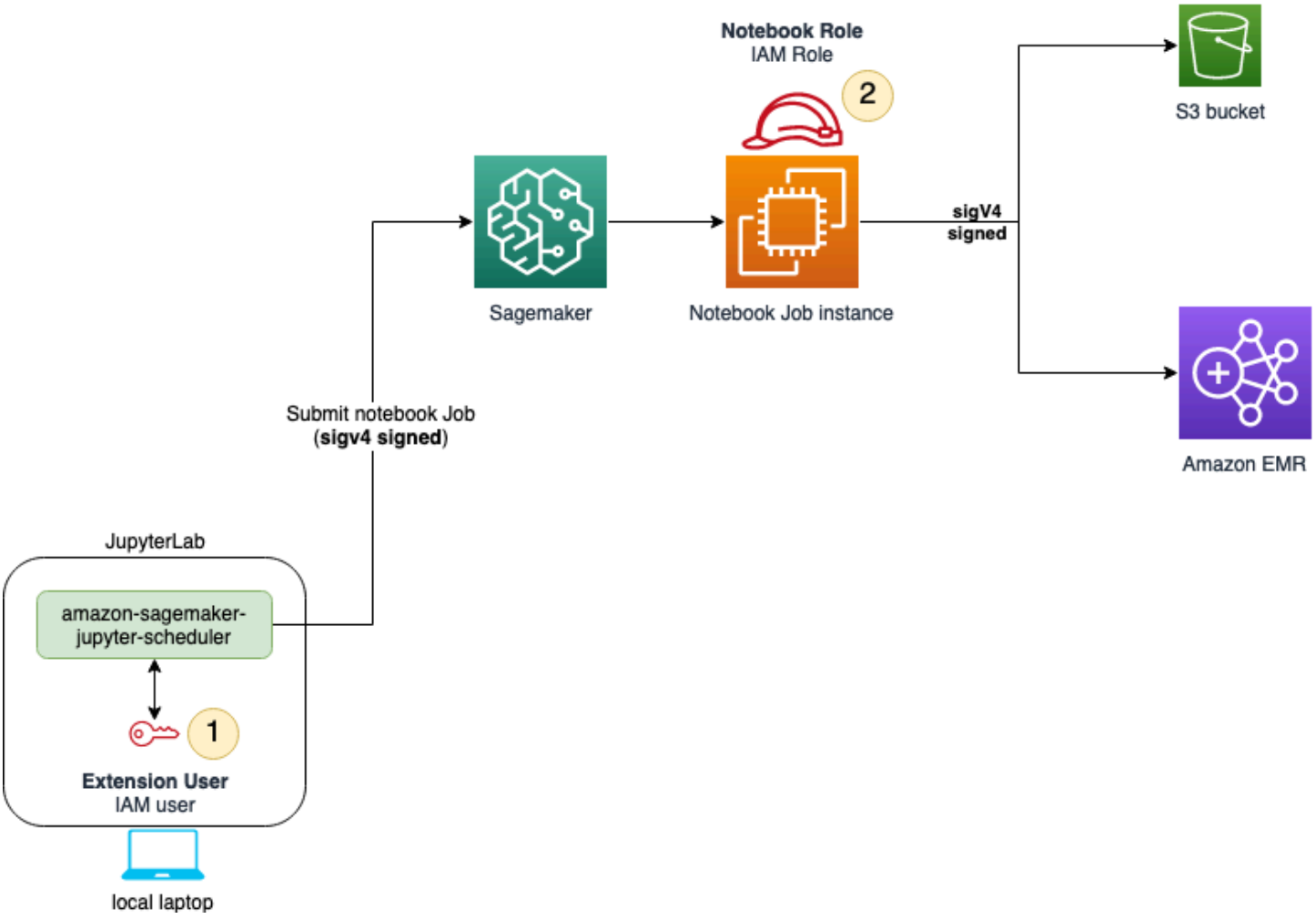
Jika Anda ingin menggunakan kunci KMS kustom, lampirkan kebijakan berikut dan berikan ARN kunci KMS Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
         "Effect":"Allow",
         "Action":[
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
```

```
            "kms:GenerateDataKey*",
            "kms:DescribeKey",
            "kms:CreateGrant"
         ],
         "Resource":"your_KMS_key_ARN"
      }
   ]
}
```

Menginstal kebijakan dan izin untuk lingkungan Jupyter lokal

Seperti yang dinyatakan sebelumnya, Anda menginstal dua set izin—izin untuk pengguna IAM dan untuk peran IAM yang diasumsikan oleh pekerjaan notebook. Seperti yang ditunjukkan pada diagram berikut, pengguna IAM perlu mengatur izin IAM untuk mengirimkan pekerjaan ke. SageMaker Setelah pengguna mengirimkan pekerjaan notebook, pekerjaan itu sendiri mengasumsikan peran IAM yang memiliki izin untuk mengakses sumber daya tergantung pada tugas pekerjaan.



Bagian berikut membantu Anda menginstal kebijakan dan izin yang diperlukan untuk pengguna IAM dan peran eksekusi pekerjaan.

Izin pengguna IAM

Izin untuk mengirimkan pekerjaan ke SageMaker

Untuk menambahkan izin untuk mengirimkan lowongan, selesaikan langkah-langkah berikut:

1. Buka konsol IAM.
2. Pilih Pengguna di panel kiri.
3. Temukan pengguna IAM untuk pekerjaan notebook Anda dan pilih nama pengguna.
4. Pilih Tambahkan Izin, dan pilih Buat kebijakan sebaris dari menu tarik-turun.
5. Pilih tab JSON.
6. Salin dan tempel kebijakan berikut ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EventBridgeSchedule",
            "Effect": "Allow",
            "Action": [
                "events:TagResource",
                "events:DeleteRule",
                "events:PutTargets",
                "events:DescribeRule",
                "events:EnableRule",
                "events:PutRule",
                "events:RemoveTargets",
                "events:DisableRule"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/sagemaker:is-scheduling-notebook-job": "true"
                }
            }
        },
        {
            "Sid": "IAMPassrole",
            "Effect": "Allow",
```

```
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "sagemaker.amazonaws.com",
                        "events.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "IAMListRoles",
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "*"
        },
        {
            "Sid": "S3ArtifactsAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutEncryptionConfiguration",
                "s3:CreateBucket",
                "s3:PutBucketVersioning",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetEncryptionConfiguration",
                "s3:DeleteObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::sagemaker-automated-execution-*"
            ]
        },
        {
            "Sid": "S3DriverAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::sagemakerheadlessexecution-*"
            ]
        },
        {
            "Sid": "SagemakerJobs",
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeTrainingJob",
                "sagemaker:StopTrainingJob",
                "sagemaker:DescribePipeline",
                "sagemaker:CreateTrainingJob",
                "sagemaker:DeletePipeline",
                "sagemaker:CreatePipeline"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/sagemaker:is-scheduling-notebook-job": "true"
                }
            }
        },
        {
            "Sid": "AllowSearch",
            "Effect": "Allow",
            "Action": "sagemaker:Search",
            "Resource": "*"
        },
        {
            "Sid": "SagemakerTags",
            "Effect": "Allow",
            "Action": [
                "sagemaker:ListTags",
                "sagemaker:AddTags"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:pipeline/*",
                "arn:aws:sagemaker:*:*:space/*",
                "arn:aws:sagemaker:*:*:training-job/*",
                "arn:aws:sagemaker:*:*:user-profile/*"
            ]
        },
        {
            "Sid": "ECRImage",
            "Effect": "Allow",
```

```
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchGetImage"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS KMSkebijakan izin (opsional)

Secara default, bucket input dan output Amazon S3 dienkripsi menggunakan enkripsi sisi server, tetapi Anda dapat menentukan kunci KMS khusus untuk mengenkripsi data Anda di bucket Amazon S3 keluaran dan volume penyimpanan yang dilampirkan ke pekerjaan notebook.

Jika Anda ingin menggunakan kunci KMS kustom, ulangi instruksi sebelumnya, lampirkan kebijakan berikut, dan berikan ARN kunci KMS Anda sendiri.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
         "Effect":"Allow",
         "Action":[
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey",
            "kms:CreateGrant"
         ],
         "Resource":"your_KMS_key_ARN"
      }
   ]
}
```

Izin peran eksekusi tugas

Hubungan kepercayaan

Untuk mengubah hubungan kepercayaan peran eksekusi pekerjaan, selesaikan langkah-langkah berikut:

1. Buka konsol IAM.
2. Pilih Peran di panel kiri.
3. Temukan peran eksekusi pekerjaan untuk pekerjaan notebook Anda dan pilih nama peran.
4. Pilih tab Trust relationship.
5. Pilih Edit kebijakan kepercayaan.
6. Salin dan tempel kebijakan berikut ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com",
                    "events.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Izin tambahan

Setelah dikirimkan, pekerjaan notebook membutuhkan izin untuk mengakses sumber daya. Petunjuk berikut menunjukkan cara menambahkan seperangkat izin minimal. Jika perlu, tambahkan lebih banyak izin berdasarkan kebutuhan pekerjaan notebook Anda. Untuk menambahkan izin ke peran eksekusi pekerjaan Anda, selesaikan langkah-langkah berikut:

1. Buka konsol IAM.
2. Pilih Peran di panel kiri.
3. Temukan peran eksekusi pekerjaan untuk pekerjaan notebook Anda dan pilih nama peran.
4. Pilih Tambahkan Izin, dan pilih Buat kebijakan sebaris dari menu tarik-turun.
5. Pilih tab JSON.
6. Salin dan tempel kebijakan berikut ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PassroleForJobCreation",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        },
        {
            "Sid": "S3ForStoringArtifacts",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::sagemaker-automated-execution-*"
        },
        {
            "Sid": "S3DriverAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::sagemakerheadlessexecution-*"
            ]
        },
        {
            "Sid": "SagemakerJobs",
            "Effect": "Allow",
            "Action": [
                "sagemaker:StartPipelineExecution",
                "sagemaker:CreateTrainingJob"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "ECRImage",
            "Effect": "Allow",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "*"
        }
    ]
}
```

7. Tambahkan izin ke sumber daya lain yang diakses pekerjaan buku catatan Anda.
8. Pilih Tinjau kebijakan.
9. Masukkan nama untuk kebijakan Anda.
10. Pilih Buat kebijakan.

Membuat Tugas Notebook

Jika ingin membuat pekerjaan notebook, Anda memiliki beberapa opsi. Anda dapat membuat pekerjaan di JupyterLab buku catatan di UI Studio, atau Anda dapat membuat pekerjaan secara terprogram dengan Python SageMaker SDK.

Jika Anda membuat pekerjaan notebook di UI Studio, Anda memberikan detail tentang gambar dan kernel, konfigurasi keamanan, dan variabel atau skrip kustom apa pun, dan pekerjaan Anda dijadwalkan. Untuk detail tentang cara menjadwalkan pekerjaan Anda menggunakan Pekerjaan SageMaker Notebook, lihatBuat pekerjaan notebook di Studio.

Untuk membuat pekerjaan notebook dengan SageMaker Python SDK, Anda membuat pipeline dengan langkah Pekerjaan Notebook dan memulai proses sesuai permintaan atau secara opsional menggunakan fitur penjadwalan pipeline untuk menjadwalkan proses di masa mendatang. SageMaker SDK memberi Anda fleksibilitas untuk menyesuaikan pipeline Anda—Anda dapat memperluas pipeline ke alur kerja dengan beberapa langkah pekerjaan notebook. Karena Anda membuat langkah Job SageMaker Notebook dan pipeline, Anda dapat melacak status eksekusi pipeline di dasbor pekerjaan Pekerjaan SageMaker Notebook dan juga melihat grafik pipeline di

Studio. Untuk detail tentang cara menjadwalkan pekerjaan Anda dengan SageMaker Python SDK dan tautan ke notebook contoh, lihat. Buat pekerjaan notebook dengan SageMaker Python SDK

Buat pekerjaan notebook dengan SageMaker Python SDK

Untuk menjalankan notebook mandiri menggunakan SageMaker Python SDK, Anda perlu membuat langkah Job Notebook, melampirkannya ke pipeline, dan menggunakan utilitas yang disediakan SageMaker oleh Pipelines untuk menjalankan pekerjaan sesuai permintaan atau secara opsional menjadwalkan satu atau beberapa pekerjaan masa depan.

Bagian berikut menjelaskan langkah-langkah dasar untuk membuat pekerjaan notebook sesuai permintaan atau terjadwal dan melacak jalannya. Selain itu, lihat diskusi berikut jika Anda perlu meneruskan parameter ke pekerjaan notebook Anda atau terhubung ke Amazon EMR di buku catatan Anda—persiapan tambahan notebook Jupyter Anda diperlukan dalam kasus ini. Anda juga dapat menerapkan default untuk subset argumen `NotebookJobStep` sehingga Anda tidak perlu menentukannya setiap kali Anda membuat langkah Job Notebook.

Untuk melihat contoh buku catatan yang menunjukkan cara menjadwalkan pekerjaan notebook dengan SageMaker Python SDK, lihat buku catatan contoh pekerjaan notebook.

Topik

- Langkah-langkah untuk membuat pekerjaan notebook
- Melihat pekerjaan notebook Anda di dasbor Studio UI
- Lihat grafik pipeline Anda di Studio
- Meneruskan parameter ke notebook Anda
- Menghubungkan ke klaster Amazon EMR di notebook input Anda
- Siapkan opsi default

Langkah-langkah untuk membuat pekerjaan notebook

Anda dapat membuat pekerjaan notebook yang berjalan segera atau sesuai jadwal. Instruksi berikut menjelaskan kedua metode.

Untuk menjadwalkan pekerjaan notebook, selesaikan langkah-langkah dasar berikut:

1. Buat sebuah `NotebookJobStep` instance. Untuk detail tentang `NotebookJobStep` parameter, lihat sagemaker.workflow.steps. NotebookJobStep. Minimal, Anda dapat memberikan argumen berikut seperti yang ditunjukkan dalam potongan kode berikut:

> **Important**
>
> Jika Anda menjadwalkan pekerjaan notebook menggunakan SageMaker Python SDK, Anda hanya dapat menentukan gambar tertentu untuk menjalankan pekerjaan notebook Anda. Untuk informasi selengkapnya, lihat Kendala gambar untuk pekerjaan notebook Python SageMaker SDK.

```
notebook_job_step = NotebookJobStep(
    input_notebook=input-notebook,
    image_uri=image-uri,
    kernel_name=kernel-name
)
```

2. Buat alur dengan langkah Anda`NotebookJobStep`, seperti yang ditunjukkan dalam potongan berikut:

```
pipeline = Pipeline(
    name=pipeline-name,
    steps=[notebook_job_step],
    sagemaker_session=sagemaker-session,
)
```

3. Jalankan pipeline sesuai permintaan atau secara opsional menjadwalkan pipeline future run. Untuk memulai proses langsung, gunakan perintah berikut:

```
execution = pipeline.start(
    parameters={...}
)
```

Secara opsional, Anda dapat menjadwalkan single future pipeline run atau beberapa run pada interval yang telah ditentukan. Anda menentukan jadwal Anda `PipelineSchedule` dan kemudian meneruskan objek jadwal ke pipeline Anda`put_triggers`. Untuk informasi lebih lanjut tentang penjadwalan alur, lihatJadwalkan pipeline dengan SageMaker Python SDK.

Contoh berikut menjadwalkan pipeline Anda untuk berjalan sekali pada 12 Desember 2023 pukul 10:31:32 UTC.

```
my_schedule = PipelineSchedule(
```

```
    name="my-schedule“,
    at=datetime(year=2023, month=12, date=25, hour=10, minute=31, second=32)
)
pipeline.put_triggers(triggers=[my_schedule])
```

Contoh berikut menjadwalkan pipeline Anda untuk berjalan pada pukul 10:15 UTC pada hari Jumat terakhir setiap bulan selama tahun 2022 hingga 2023. Untuk detail tentang penjadwalan berbasis cron, lihat Jadwal berbasis cron.

```
my_schedule = PipelineSchedule(
    name="my-schedule“,
    cron="15 10 ? * 6L 2022-2023"
)
pipeline.put_triggers(triggers=[my_schedule])
```

4. (Opsional) Lihat pekerjaan notebook Anda di dasbor Pekerjaan SageMaker Notebook. Nilai yang Anda berikan untuk `tags` argumen langkah Job Notebook mengontrol cara UI Studio menangkap dan menampilkan pekerjaan. Untuk informasi selengkapnya, lihat Melihat pekerjaan notebook Anda di dasbor Studio UI.

Melihat pekerjaan notebook Anda di dasbor Studio UI

Pekerjaan buku catatan yang Anda buat sebagai langkah pipeline akan muncul di dasbor Pekerjaan Notebook Studio jika Anda menentukan tag tertentu.

> **Note**
>
> Hanya pekerjaan notebook yang dibuat di Studio atau JupyterLab lingkungan lokal yang membuat definisi pekerjaan. Oleh karena itu, jika Anda membuat pekerjaan notebook dengan SageMaker Python SDK, Anda tidak akan melihat definisi pekerjaan di dasbor Pekerjaan Notebook. Namun, Anda dapat melihat pekerjaan notebook Anda seperti yang dijelaskan dalamMelihat tugas Anda.

Anda dapat mengontrol anggota tim mana yang dapat melihat pekerjaan buku catatan Anda dengan tag berikut:

- Untuk menampilkan buku catatan ke semua profil pengguna atau spasi di domain, tambahkan tag domain dengan nama domain Anda. Contoh ditampilkan sebagai berikut:

  - kunci:`sagemaker:domain-name`, nilai: `d-abcdefghij5k`
- Untuk menampilkan pekerjaan notebook ke profil pengguna tertentu di domain, tambahkan profil pengguna dan tag domain. Contoh tag profil pengguna ditampilkan sebagai berikut:
  - kunci:`sagemaker:user-profile-name`, nilai: `studio-user`
- Untuk menampilkan pekerjaan notebook ke spasi, tambahkan spasi dan tag domain. Contoh tag spasi ditampilkan sebagai berikut:
  - kunci:`sagemaker:shared-space-name`, nilai: `my-space-name`
- Jika Anda tidak melampirkan domain atau profil pengguna atau tag spasi apa pun, UI Studio tidak menampilkan tugas buku catatan yang dibuat berdasarkan langkah pipeline. Dalam hal ini, Anda dapat melihat pekerjaan pelatihan yang mendasarinya di konsol pekerjaan pelatihan atau Anda dapat melihat status dalam daftar eksekusi pipeline.

Setelah Anda menyiapkan tag yang diperlukan untuk melihat pekerjaan Anda di dasbor, lihat Melihat tugas Anda petunjuk tentang cara melihat pekerjaan Anda dan mengunduh output.

Lihat grafik pipeline Anda di Studio

Karena langkah pekerjaan notebook Anda adalah bagian dari pipeline, Anda dapat melihat grafik pipeline (DAG) di Studio. Dalam grafik pipeline, Anda dapat melihat status jalur pipa dan melacak garis keturunan. Untuk detailnya, lihat Melihat Eksekusi Alur.

Meneruskan parameter ke notebook Anda

Jika Anda ingin meneruskan parameter ke pekerjaan notebook Anda (menggunakan `parameters` argumen`NotebookJobStep`), Anda perlu menyiapkan notebook input Anda untuk menerima parameter.

Pelaksana pekerjaan notebook berbasis Papermill mencari sel Jupyter yang ditandai dengan `parameters` tag dan menerapkan parameter baru atau penggantian parameter segera setelah sel ini. Untuk detailnya, lihat Parameterisasi notebook Anda.

Setelah Anda melakukan langkah ini, berikan parameter Anda ke Anda`NotebookJobStep`, seperti yang ditunjukkan pada contoh berikut:

```
notebook_job_parameters = {
    "company": "Amazon"
}

notebook_job_step = NotebookJobStep(
```

```
    image_uri=image-uri,
    kernel_name=kernel-name,
    role=role-name,
    input_notebook=input-notebook,
    parameters=notebook_job_parameters,
    ...
)
```

Menghubungkan ke klaster Amazon EMR di notebook input Anda

Jika Anda terhubung ke klaster EMR Amazon dari notebook Jupyter di Studio, Anda mungkin perlu memodifikasi notebook Jupyter lebih lanjut. Lihat Connect ke klaster Amazon EMR dari notebook Anda apakah Anda perlu melakukan salah satu tugas berikut di buku catatan Anda:

- Teruskan parameter ke perintah koneksi EMR Amazon Anda. Studio menggunakan Papermill untuk menjalankan notebook. Dalam SparkMagic kernel, parameter yang Anda berikan ke perintah koneksi EMR Amazon Anda mungkin tidak berfungsi seperti yang diharapkan karena cara Papermill meneruskan informasi. SparkMagic
- Meneruskan kredensyal pengguna ke cluster EMR Amazon yang diautentikasi oleh Kerberos, LDAP, atau HTTP Basic Auth-autentikasi. Anda harus meneruskan kredensi pengguna melalui file. AWS Secrets Manager

Siapkan opsi default

SageMaker SDK memberi Anda opsi untuk mengatur default untuk subset parameter sehingga Anda tidak perlu menentukan parameter ini setiap kali Anda membuat instance. `NotebookJobStep` Parameter ini adalah`role`,`s3_root_uri`,`s3_kms_key`,`volume_kms_key`,`subnets`, dan`security_group_ids`. Gunakan file SageMaker konfigurasi untuk mengatur default untuk langkah tersebut. Untuk informasi tentang file SageMaker konfigurasi, lihat Mengkonfigurasi dan menggunakan default dengan Python SDK. SageMaker .

Untuk menyiapkan default pekerjaan notebook, terapkan default baru Anda ke bagian pekerjaan buku catatan dari file konfigurasi seperti yang ditunjukkan pada cuplikan berikut:

```
SageMaker:
  PythonSDK:
    Modules:
      NotebookJob:
        RoleArn: 'arn:aws:iam::555555555555:role/IMRole'
        S3RootUri: 's3://my-bucket/my-project'
```

```
        S3KmsKeyId: 's3kmskeyid'
        VolumeKmsKeyId: 'volumekmskeyid1'
        VpcConfig:
          SecurityGroupIds:
            - 'sg123'
          Subnets:
            - 'subnet-1234'
```

Buat pekerjaan notebook di Studio

> **Note**
>
> Penjadwal notebook dibuat dari layanan Amazon EventBridge, SageMaker Training, dan SageMaker Pipelines. Jika pekerjaan notebook Anda gagal, Anda mungkin melihat kesalahan yang terkait dengan layanan ini.

SageMaker Pekerjaan Notebook memberi Anda alat untuk membuat dan mengelola pekerjaan notebook noninteraktif Anda menggunakan widget Pekerjaan Notebook. Anda dapat membuat pekerjaan, melihat pekerjaan yang Anda buat, dan menjeda, menghentikan, atau melanjutkan pekerjaan yang ada. Anda juga dapat mengubah jadwal buku catatan.

Saat Anda membuat pekerjaan buku catatan terjadwal dengan widget, penjadwal mencoba menyimpulkan pilihan opsi default dan secara otomatis mengisi formulir untuk membantu Anda memulai dengan cepat. Jika Anda menggunakan Studio, setidaknya Anda dapat mengirimkan pekerjaan sesuai permintaan tanpa menetapkan opsi apa pun. Anda juga dapat mengirimkan definisi pekerjaan buku catatan (terjadwal) yang hanya menyediakan informasi jadwal khusus waktu. Namun, Anda dapat menyesuaikan bidang lain jika pekerjaan terjadwal Anda memerlukan pengaturan khusus. Jika Anda menjalankan buku catatan Jupyter lokal, ekstensi penjadwal menyediakan fitur bagi Anda untuk menentukan default Anda sendiri (untuk subset opsi) sehingga Anda tidak perlu memasukkan nilai yang sama secara manual setiap saat.

Untuk menjadwalkan pekerjaan notebook, selesaikan langkah-langkah berikut:

1. Buka formulir Create Job.

   Di JupyterLab lingkungan lokal, pilih ikon Create a notebook job
   ()
   di taskbar. Jika Anda tidak melihat ikon, ikuti petunjuk Panduan Instalasi untuk menginstalnya.

Di Studio, buka formulir dengan salah satu dari dua cara berikut:

- Menggunakan File Browser

  1. Di File Browser di panel kiri, klik kanan pada notebook yang ingin Anda jalankan sebagai pekerjaan terjadwal.
  2. Pilih Create Notebook Job.

- Dalam notebook Studio

  - Di dalam buku catatan Studio yang ingin Anda jalankan sebagai pekerjaan terjadwal, pilih ikon Buat pekerjaan notebook ( ) di toolbar Studio.

2. Selesaikan formulir popup. Formulir menampilkan bidang-bidang berikut:
   - Nama Job: Nama deskriptif yang Anda tentukan untuk pekerjaan Anda.
   - File input: Nama buku catatan yang Anda jadwalkan untuk dijalankan dalam mode noninteraktif.
   - Tipe komputasi: Tipe instans Amazon EC2 tempat Anda ingin menjalankan notebook Anda.
   - Parameter: Parameter khusus yang dapat Anda tentukan secara opsional sebagai input ke buku catatan Anda. Untuk menggunakan fitur ini, Anda mungkin secara opsional ingin menandai sel tertentu di buku catatan Jupyter Anda dengan **`parameters`** tag untuk mengontrol di mana parameter Anda diterapkan. Untuk detail selengkapnya, lihat Parameterisasi notebook Anda.
   - Opsi Tambahan: Anda dapat menentukan penyesuaian tambahan untuk pekerjaan Anda. Misalnya, Anda dapat menentukan gambar atau kernel, folder input dan output, opsi coba ulang pekerjaan dan batas waktu, detail enkripsi, dan skrip inisialisasi kustom. Untuk daftar lengkap penyesuaian yang dapat Anda terapkan, lihat. Pilihan yang tersedia
3. Jadwalkan pekerjaan Anda. Anda dapat menjalankan buku catatan Anda sesuai permintaan atau berdasarkan jadwal tetap.
   - Untuk menjalankan notebook sesuai permintaan, selesaikan langkah-langkah berikut:
     - Pilih Jalankan Sekarang.
     - Pilih Buat.
     - Tab Pekerjaan Notebook akan muncul. Pilih Muat Ulang untuk memuat pekerjaan Anda ke dasbor.
   - Untuk menjalankan notebook pada jadwal tetap, lakukan langkah-langkah berikut:
     - Pilih Jalankan sesuai jadwal.

- Pilih daftar dropdown Interval dan pilih interval. Intervalnya berkisar dari setiap menit hingga bulanan. Anda juga dapat memilih Jadwal khusus.
- Berdasarkan interval yang Anda pilih, bidang tambahan muncul untuk membantu Anda menentukan lebih lanjut hari dan waktu lari yang Anda inginkan. Misalnya, jika Anda memilih Hari untuk menjalankan harian, bidang tambahan akan muncul bagi Anda untuk menentukan waktu yang diinginkan. Perhatikan bahwa setiap kali Anda menentukan dalam format UTC. Perhatikan juga bahwa jika Anda memilih interval kecil, seperti satu menit, pekerjaan Anda tumpang tindih jika pekerjaan sebelumnya tidak selesai ketika pekerjaan berikutnya dimulai.

  Jika Anda memilih jadwal kustom, Anda menggunakan sintaks cron di kotak ekspresi untuk menentukan tanggal dan waktu berjalan yang tepat. Sintaks cron adalah daftar digit yang dipisahkan ruang, yang masing-masing mewakili satuan waktu dari detik ke tahun. Untuk bantuan dengan sintaks cron, Anda dapat memilih Dapatkan bantuan dengan sintaks cron di bawah kotak ekspresi.
- Pilih Buat.
- Tab Notebook Job Definitions muncul. Pilih Muat Ulang untuk memuat definisi pekerjaan Anda ke dasbor.

Menyiapkan opsi default untuk buku catatan lokal

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Jika Anda harus mengetik (atau menempelkan) nilai kustom secara manual di formulir Buat Job, Anda dapat menyimpan nilai default baru dan ekstensi penjadwal menyisipkan nilai baru Anda setiap kali Anda membuat definisi pekerjaan baru. Fitur ini tersedia untuk opsi berikut:

- Peran ARN
- Folder Masukan S3
- Folder Keluaran S3
- Kunci KMS enkripsi keluaran (jika Anda mengaktifkan Configure Job Encryption)

- Kunci KMS enkripsi volume instance Job (jika Anda mengaktifkan Configure Job Encryption)

Fitur ini menghemat waktu Anda jika Anda memasukkan nilai yang berbeda dari default yang disediakan dan terus menggunakan nilai-nilai tersebut untuk menjalankan pekerjaan di masa depan. Pengaturan pengguna yang Anda pilih disimpan di mesin yang menjalankan JupyterLab server Anda dan diambil dengan bantuan API asli. Jika Anda memberikan nilai default baru untuk satu atau lebih tetapi tidak semua lima opsi, default sebelumnya diambil untuk yang tidak Anda sesuaikan.

Petunjuk berikut menunjukkan cara melihat pratinjau nilai default yang ada, menetapkan nilai default baru, dan mengatur ulang nilai default untuk pekerjaan buku catatan Anda.

Untuk melihat nilai default yang ada untuk pekerjaan buku catatan Anda, selesaikan langkah berikut:

1. Buka konsol Amazon SageMaker Studio Classic dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio Classic.
2. Di File Browser di panel kiri, klik kanan pada notebook yang ingin Anda jalankan sebagai pekerjaan terjadwal.
3. Pilih Create Notebook Job.
4. Pilih Opsi tambahan untuk memperluas tab pengaturan pekerjaan notebook. Anda dapat melihat pengaturan default di sini.

Untuk menetapkan nilai default baru untuk tugas notebook Anda di masa mendatang, selesaikan langkah berikut:

1. Buka konsol Amazon SageMaker Studio Classic dengan mengikuti petunjuk diLuncurkan Amazon SageMaker Studio Classic.
2. Dari menu atas di Studio Classic, pilih Settings, lalu pilih Advanced Settings Editor.
3. Pilih Amazon SageMaker Scheduler dari daftar di bawah Pengaturan. Ini mungkin sudah terbuka secara default.
4. Anda dapat memperbarui pengaturan default langsung di halaman UI ini atau dengan menggunakan editor JSON.

    - Di UI Anda dapat menyisipkan nilai baru untuk ARN Peran, Folder Input S3, Folder Output S3, kunci KMS enkripsi Output, atau kunci KMS enkripsi volume Job instance. Jika Anda mengubah nilai-nilai ini, Anda akan melihat default baru untuk bidang ini saat Anda membuat pekerjaan buku catatan berikutnya di bawah Opsi tambahan.

- (Opsional) Untuk memperbarui default pengguna menggunakan JSON Settings Editor, selesaikan langkah-langkah berikut:

  1. Di pojok kanan atas, pilih JSON Settings Editor.
  2. Di bilah sisi kiri Pengaturan, pilih SageMaker Penjadwal Amazon. Ini mungkin sudah terbuka secara default.

     Anda dapat melihat nilai default Anda saat ini di panel Preferensi Pengguna.

     Anda dapat melihat nilai default sistem di panel System Defaults.
  3. Untuk memperbarui nilai default Anda, salin dan tempel cuplikan JSON dari panel Default Sistem ke panel Preferensi Pengguna, dan perbarui bidangnya.
  4. Jika Anda memperbarui nilai default, pilih ikon Simpan Pengaturan Pengguna
     ()
     di sudut kanan atas. Menutup editor tidak menyimpan perubahan.

Jika sebelumnya Anda berubah dan sekarang ingin mengatur ulang nilai default yang ditentukan pengguna, selesaikan langkah-langkah berikut:

1. Dari menu atas di Studio Classic, pilih Settings, lalu pilih Advanced Settings Editor.
2. Pilih Amazon SageMaker Scheduler dari daftar di bawah Pengaturan. Ini mungkin sudah terbuka secara default.
3. Anda dapat mengembalikan default dengan langsung menggunakan halaman UI ini atau menggunakan editor JSON.

   - Di UI Anda dapat memilih Restore to Defaults di pojok kanan atas. Default Anda dikembalikan ke string kosong. Anda hanya melihat opsi ini jika sebelumnya Anda mengubah nilai default Anda.
   - (Opsional) Untuk memulai ulang pengaturan default menggunakan Editor Pengaturan JSON, selesaikan langkah-langkah berikut:

     1. Di pojok kanan atas, pilih JSON Settings Editor.
     2. Di bilah sisi kiri Pengaturan, pilih SageMaker Penjadwal Amazon. Ini mungkin sudah terbuka secara default.

        Anda dapat melihat nilai default Anda saat ini di panel Preferensi Pengguna.

Anda dapat melihat nilai default sistem di panel System Defaults.

3. Untuk memulihkan pengaturan default Anda saat ini, salin konten dari panel Default Sistem ke panel Preferensi Pengguna.
4. Pilih ikon Simpan Pengaturan Pengguna
   ()
   di sudut kanan atas. Menutup editor tidak menyimpan perubahan.

Untuk membuat alur kerja notebook

Karena pekerjaan notebook menjalankan kode kustom, Anda dapat membuat pipeline yang menyertakan satu atau beberapa langkah pekerjaan notebook. Alur kerja ML sering berisi beberapa langkah, seperti langkah pemrosesan untuk memproses data sebelumnya, langkah pelatihan untuk membangun model Anda, dan langkah evaluasi model, antara lain. Salah satu kemungkinan penggunaan pekerjaan notebook adalah untuk menangani praproses—Anda mungkin memiliki buku catatan yang melakukan transformasi atau konsumsi data, langkah EMR yang melakukan pembersihan data, dan pekerjaan notebook lain yang melakukan featurisasi input Anda sebelum memulai langkah pelatihan. Pekerjaan notebook mungkin memerlukan informasi dari langkah sebelumnya dalam pipeline atau dari kustomisasi yang ditentukan pengguna sebagai parameter di notebook input. Untuk contoh yang menunjukkan cara meneruskan variabel dan parameter lingkungan ke buku catatan Anda dan mengambil informasi dari langkah sebelumnya, lihatBerikan informasi ke dan dari langkah buku catatan Anda.

Dalam kasus penggunaan lain, salah satu pekerjaan notebook Anda mungkin memanggil buku catatan lain untuk melakukan beberapa tugas selama menjalankan notebook Anda—dalam skenario ini Anda perlu menentukan buku catatan bersumber ini sebagai dependensi dengan langkah pekerjaan notebook Anda. Untuk informasi tentang cara menelepon buku catatan lain, lihatPanggil buku catatan lain di pekerjaan notebook Anda.

Untuk melihat contoh buku catatan yang menunjukkan cara menjadwalkan pekerjaan notebook dengan SageMaker Python SDK, lihat buku catatan contoh pekerjaan notebook.

Berikan informasi ke dan dari langkah buku catatan Anda

Bagian berikut menjelaskan cara untuk meneruskan informasi ke buku catatan Anda sebagai variabel dan parameter lingkungan.

Lulus variabel lingkungan

Lulus variabel lingkungan sebagai kamus ke `environment_variable` argumen Anda`NotebookJobStep`, seperti yang ditunjukkan pada contoh berikut:

```
environment_variables = {"RATE": 0.0001, "BATCH_SIZE": 1000}

notebook_job_step = NotebookJobStep(
    ...
    environment_variables=environment_variables,
    ...
)
```

Anda dapat menggunakan variabel lingkungan di buku catatan menggunakan`os.getenv()`, seperti yang ditunjukkan pada contoh berikut:

```
# inside your notebook
import os
print(f"ParentNotebook: env_key={os.getenv('env_key')}")
```

Parameter lulus

Saat meneruskan parameter ke langkah Job Notebook pertama dalam `NotebookJobStep` instans, Anda mungkin ingin menandai sel di buku catatan Jupyter untuk menunjukkan tempat menerapkan parameter baru atau penggantian parameter. Untuk petunjuk tentang cara menandai sel di buku catatan Jupyter Anda, lihat. Parameterisasi notebook Anda

Anda meneruskan parameter melalui parameter langkah Notebook Job, seperti yang ditampilkan di cuplikan berikut: `parameters`

```
notebook_job_parameters = {
    "company": "Amazon",
}

notebook_job_step = NotebookJobStep(
    ...
    parameters=notebook_job_parameters,
    ...
)
```

Di dalam buku catatan input Anda, parameter Anda diterapkan setelah sel ditandai dengan `parameters` atau di awal buku catatan jika Anda tidak memiliki sel yang ditandai.

```
# this cell is in your input notebook and is tagged with 'parameters'
# your parameters and parameter overrides are applied after this cell
company='default'
```

```
# in this cell, your parameters are applied
# prints "company is Amazon"
print(f'company is {company}')
```

Mengambil informasi dari langkah sebelumnya

Diskusi berikut menjelaskan bagaimana Anda dapat mengekstrak data dari langkah sebelumnya untuk diteruskan ke langkah Job Notebook Anda.

Gunakan **`properties`** atribut

Anda dapat menggunakan properti berikut dengan `properties` atribut langkah sebelumnya:

- `ComputingJobName`—Nama pekerjaan pelatihan
- `ComputingJobStatus`—Status pekerjaan pelatihan
- `NotebookJobInputLocation`Masukan lokasi Amazon S3
- `NotebookJobOutputLocationPrefix`—Jalan menuju output pekerjaan pelatihan Anda, lebih {*`NotebookJobOutputLocationPrefix`*}/{*`training-job-name`*}`/output/output.tar.gz` spesifik. berisi output
- `InputNotebookName`—Nama file notebook masukan
- `OutputNotebookName`—Nama file notebook keluaran (yang mungkin tidak ada di folder keluaran pekerjaan pelatihan jika pekerjaan gagal)

Potongan kode berikut menunjukkan cara mengekstrak parameter dari atribut properti.

```
notebook_job_step2 = NotebookJobStep(
    ....
    parameters={
        "step1_JobName": notebook_job_step1.properties.ComputingJobName,
        "step1_JobStatus": notebook_job_step1.properties.ComputingJobStatus,
        "step1_NotebookJobInput":
 notebook_job_step1.properties.NotebookJobInputLocation,
```

```
        "step1_NotebookJobOutput":
notebook_job_step1.properties.NotebookJobOutputLocationPrefix,
    }
```

Gunakan JsonGet

Jika Anda ingin meneruskan parameter selain yang disebutkan sebelumnya dan output JSON dari langkah Anda sebelumnya berada di Amazon S3, gunakan. `JsonGet` `JsonGet`adalah mekanisme umum yang dapat langsung mengekstrak data dari file JSON di Amazon S3.

Untuk mengekstrak file JSON di Amazon S3, `JsonGet` selesaikan langkah-langkah berikut:

1. Unggah file JSON Anda ke Amazon S3. Jika data Anda sudah di-upload ke Amazon S3, lewati langkah ini. Contoh berikut menunjukkan mengunggah file JSON ke Amazon S3.

   ```
   import json
   from sagemaker.s3 import S3Uploader

   output = {
       "key1": "value1",
       "key2": [0,5,10]
   }

   json_output = json.dumps(output)

   with open("notebook_job_params.json", "w") as file:
       file.write(json_output)

   S3Uploader.upload(
       local_path="notebook_job_params.json",
       desired_s3_uri="s3://path/to/bucket"
   )
   ```

2. Berikan URI S3 Anda dan jalur JSON ke nilai yang ingin Anda ekstrak. Dalam contoh berikut, `JsonGet` mengembalikan sebuah objek yang mewakili indeks 2 dari nilai yang terkait dengan key `key2` (`10`).

   ```
   NotebookJobStep(
       ....
       parameters={
           # the key job_key1 returns an object representing the value 10
           "job_key1": JsonGet(
   ```

```
            s3_uri=Join(on="/", values=["s3:/", ..]),
            json_path="key2[2]" # value to reference in that json file
        ),
        "job_key2": "Amazon"
    }
)
```

Panggil buku catatan lain di pekerjaan notebook Anda

Diskusi berikut menyiapkan contoh pipeline dengan langkah Notebook Job di mana notebook memanggil dua notebook lainnya. Notebook input berisi baris berikut:

```
%run 'subfolder/notebook_to_call_in_subfolder.ipynb'
%run 'notebook_to_call.ipynb'
```

Masukkan buku catatan ini ke `NotebookJobStep` instans Anda`additional_dependencies`, seperti yang ditunjukkan pada cuplikan berikut. Perhatikan bahwa jalur yang disediakan untuk buku catatan `additional_dependencies` disediakan dari lokasi root. Untuk informasi tentang cara SageMaker mengunggah file dan folder dependen Anda ke Amazon S3 sehingga Anda dapat memberikan jalur ke dependensi dengan benar, lihat deskripsi untuk di. `additional_dependencies` NotebookJobStep

```
input_notebook = "inputs/input_notebook.ipynb"
simple_notebook_path = "inputs/notebook_to_call.ipynb"
folder_with_sub_notebook = "inputs/subfolder"

notebook_job_step = NotebookJobStep(
    image_uri=image-uri,
    kernel_name=kernel-name,
    role=role-name,
    input_notebook=input_notebook,
    additional_dependencies=[simple_notebook_path, folder_with_sub_notebook],
    tags=tags,
)
```

Pilihan yang tersedia

Tabel berikut menampilkan semua opsi yang tersedia yang dapat Anda gunakan untuk menyesuaikan pekerjaan notebook, baik Anda menjalankan Job Notebook di Studio, lingkungan Jupyter lokal, atau menggunakan Python SageMaker SDK. Tabel ini mencakup jenis opsi kustom,

deskripsi, pedoman tambahan tentang cara menggunakan opsi, nama bidang untuk opsi di Studio (jika tersedia) dan nama parameter untuk langkah pekerjaan notebook di SDK SageMaker Python (jika tersedia).

Untuk beberapa opsi, Anda juga dapat mengatur nilai default kustom sehingga Anda tidak perlu menentukannya setiap kali Anda menyiapkan pekerjaan buku catatan. Untuk Studio, opsi ini adalah Peran, folder Input, folder Output, dan ID Kunci KMS, dan ditentukan dalam tabel berikut. Jika Anda menetapkan default kustom untuk opsi ini, bidang ini akan diisi sebelumnya dalam formulir Buat Job saat Anda membuat pekerjaan buku catatan. Untuk detail tentang cara membuat default kustom di Studio dan lingkungan Jupyter lokal, lihat. Menyiapkan opsi default untuk buku catatan lokal

SageMaker SDK juga memberi Anda opsi untuk mengatur default cerdas sehingga Anda tidak perlu menentukan parameter ini saat membuat file. `NotebookJobStep` Parameter ini adalah `roles3_root_uri`,`s3_kms_key`,`volume_kms_key`, `subnetssecurity_group_ids`,,, dan ditentukan dalam tabel berikut. Untuk informasi tentang cara mengatur default cerdas, lihat. Siapkan opsi default

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Nama Tugas	Nama pekerjaan Anda seperti yang akan muncul di dasbor Pekerjaan Notebook.	Nama Field Job.	Default-nya adalah 1.	Parameter `notebook_job_name` . Default ke `None`.
Citra	Gambar kontainer yang digunakan untuk menjalankan notebook secara noninteraktif pada jenis komputasi yang dipilih.	Gambar Lapangan. Bidang ini default ke gambar notebook Anda saat ini. Ubah bidang ini dari default ke nilai kustom jika diperlukan. Jika Studio tidak dapat menyimpulkan nilai ini, formulir akan menampilk	Gambar Lapangan. Bidang ini memerlukan URI ECR dari image Docker yang dapat menjalankan notebook yang disediaka n pada jenis komputasi yang dipilih. Secara default, ekstensi penjadwal menggunakan SageMaker	Diperluka n. Parameter `image_uri` . Lokasi URI dari gambar

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
		an kesalahan validasi yang mengharuskan Anda untuk menentukannya. Gambar ini dapat berupa kustom, bring-your-own gambar, atau SageMaker gambar Amazon yang tersedia. Untuk daftar SageMaker gambar yang tersedia yang didukung oleh penjadwal buku catatan, lihat SageMaker Gambar Amazon yang Tersedia.	gambar Docker yang sudah dibuat sebelumnya — dasar Python 2.0. Ini adalah gambar resmi Python 3.8 dari DockerHub dengan boto3,, AWS CLI dan kernel Python 3. Anda juga dapat memberikan URI ECR apa pun yang memenuhi spesifikasi gambar kustom notebook. Untuk detailnya, lihat Spesifikasi SageMaker gambar kustom. Gambar ini harus memiliki semua kernel dan pustaka yang diperlukan untuk menjalankan notebook.	Docker di ECR. Anda dapat menggunakan Gambar SageMaker Distribusi tertentu atau gambar kustom berdasarkan gambar tersebut, atau gambar Anda sendiri yang telah diinstal sebelumnya dengan dependensi pekerjaan

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
				notebook yang memenuhi persyaratan tambahan. Untuk detailnya, lihat Kendala gambar untuk pekerjaan notebook Python SageMaker SDK.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Jenis instans	Jenis instans EC2 yang akan digunakan untuk menjalankan tugas notebook. Pekerjaan notebook menggunakan SageMaker Training Job sebagai lapisan komputasi, jadi tipe instance yang ditentukan harus berupa tipe instans yang didukung SageMaker Pelatihan.	Jenis Komputasi Bidang. Default ke `ml.m5.large` .	Default-nya adalah 1.	Parameter `instance_type` . Default ke `ml.m5.large` .

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Kernel	Kernel Jupyter digunakan untuk menjalankan pekerjaan notebook.	Kernel Lapangan. Bidang ini default ke kernel notebook Anda saat ini. Ubah bidang ini dari default ke nilai kustom jika diperlukan. Jika Studio tidak dapat menyimpulkan nilai ini, formulir akan menampilkan kesalahan validasi yang mengharuskan Anda untuk menentukannya.	Kernel Lapangan. Kernel ini harus ada dalam gambar dan mengikuti spesifikasi kernel Jupyter. Bidang ini default ke kernel Python3 yang ditemukan di gambar dasar Python 2.0. SageMaker Ubah bidang ini menjadi nilai kustom jika diperlukan.	Diperlukan. Parameter `kernel_name` . Kernel ini harus ada dalam gambar dan mengikuti spesifikasi kernel Jupyter. Untuk melihat pengidentifikasi kernel untuk gambar Anda, lihat (LINK).

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
SageMak sesi	SageMaker Sesi yang mendasari panggilan SageMaker layanan didelegas ikan.	N/A	N/A	Parameter `sagemaker _session` . Jika tidak ditentuka n, satu dibuat menggunak an rantai konfigura si default.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
ARN Peran	Amazon Resource Name (ARN) peran yang digunakan dengan pekerjaan notebook.	Peran Lapangan ARN. Bidang ini default ke peran eksekusi Studio. Ubah bidang ini menjadi nilai kustom jika diperlukan. **Note** Jika Studio tidak dapat menyimpulkan nilai ini, bidang ARN Peran kosong. Dalam hal ini, masukkan ARN yang ingin Anda gunakan.	Peran Lapangan ARN. Bidang ini default ke peran apa pun yang diawali dengan. `Sagemaker JupyterScheduler` Jika Anda memiliki beberapa peran dengan awalan, ekstensi memilih satu. Ubah bidang ini menjadi nilai kustom jika diperlukan. Untuk bidang ini, Anda dapat mengatur default pengguna Anda sendiri yang telah terisi sebelumnya setiap kali Anda membuat definisi pekerjaan baru. Untuk detailnya, lihat Menyiapkan opsi default untuk buku catatan lokal.	Parameter `role`. Default ke peran IAM SageMaker default jika SDK berjalan di SageMaker Notebook atau Buku Catatan Studio. SageMaker Kalau tidak, itu melempar a. `ValueError` Memungkinkan default cerdas.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Menginstal Notebook	Nama notebook yang Anda jadwalkan untuk dijalankan.	Diperlukan. Berkas Masukan Bidang.	Default-nya adalah 1.	Diperlukan .Parameterinput_notebook .
Mengaktifkan	Folder yang berisi input Anda. Input pekerjaan , termasuk notebook input dan skrip start-up atau inisialisasi opsional apa pun, dimasukkan ke dalam folder ini.	Folder Masukan Bidang. Jika Anda tidak menyediakan folder, penjadwal akan membuat bucket Amazon S3 default untuk input Anda.	Default-nya adalah 1. Untuk bidang ini, Anda dapat mengatur default pengguna Anda sendiri yang telah terisi sebelumnya setiap kali Anda membuat definisi pekerjaan baru. Untuk detailnya, lihat Menyiapkan opsi default untuk buku catatan lokal.	N/A. Folder input ditempatkan di dalam lokasi yang ditentukan oleh parameter s3_root_uri .

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Pembuata	Folder yang berisi output Anda. Output pekerjaan, termasuk notebook keluaran dan log, dimasukkan ke dalam folder ini.	Folder Keluaran Bidang. Jika Anda tidak menentuka n folder, penjadwal akan membuat bucket Amazon S3 default untuk output Anda.	Default-nya adalah 1. Untuk bidang ini, Anda dapat mengatur default pengguna Anda sendiri yang telah terisi sebelumnya setiap kali Anda membuat definisi pekerjaan baru. Untuk detailnya, lihat Menyiapka n opsi default untuk buku catatan lokal.	N/A. Folder output ditempatk an di dalam lokasi yang ditentuka n oleh parameter `s3_root_u ri` .
Paramete	Kamus variabel dan nilai untuk diteruskan ke pekerjaan notebook Anda.	Parameter Bidang. Anda perlu membuat parameter notebook Anda untuk menerima parameter.	Default-nya adalah 1.	Parameter `parameter s` . Anda perlu membuat parameter notebook Anda untuk menerima parameter .

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Dependensi tambahan (file atau folder)	Daftar dependensi file atau folder yang diunggah oleh pekerjaan notebook ke folder bertahap s3.	Tidak didukung.	Tidak didukung.	Parameter `additional_dependencies` . Pekerjaan notebook mengunggah dependensi ini ke folder bertahap S3 sehingga dapat dikonsumsi selama eksekusi.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
URI akar S3	Folder yang berisi input Anda. Input pekerjaan , termasuk notebook input dan skrip start-up atau inisialisasi opsional apa pun, dimasukkan ke dalam folder ini.	N/A. Gunakan Folder Input dan folder Output.	Default-nya adalah 1.	Parameter `s3_root_uri` . Default ke bucket S3 default. Memungkinkan default cerdas.
Variabel-variabel lingkungan	Setiap variabel lingkungan yang ada yang ingin Anda timpa, atau variabel lingkungan baru yang ingin Anda perkenalkan dan gunakan di buku catatan Anda.	Variabel Lingkungan Lapangan.	Default-nya adalah 1.	Parameter `environment_variables` . Default ke `None`.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Tag	Daftar tanda yang terlampir pada pekerjaan tersebut.	N/A	N/A	Parameter `tags`. Default ke `None`. Tag Anda mengontrol cara UI Studio menangkap dan menampilkan pekerjaan yang dibuat oleh pipeline. Untuk detailnya, lihat Melihat pekerjaan notebook Anda di dasbor

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
				Studio UI.
Skrip start-up	Skrip yang dimuat sebelumnya di menu startup notebook yang dapat Anda pilih untuk dijalankan sebelum menjalankan notebook.	Skrip Start-up bidang. Pilih skrip Lifecycle Configuration (LCC) yang berjalan pada gambar saat start-up. **Note** Skrip start-up berjalan di shell di luar lingkungan Studio. Oleh karena itu, skrip ini tidak dapat bergantung pada penyimpanan lokal Studio, variabel lingkungan, atau metadata aplikasi (in `/opt/ml/metadata`). Juga, jika Anda menggunakan skrip start-up dan skrip inisialisasi, skrip start-up berjalan terlebih dahulu.	Tidak didukung.	Tidak didukung.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Skrip inisialisasi	Jalur ke skrip lokal yang dapat Anda jalankan saat notebook Anda dinyalakan.	Skrip Inisialisasi Bidang. Masukkan jalur file EFS tempat skrip lokal atau skrip Lifecycle Configuration (LCC) berada. Jika Anda menggunakan skrip start-up dan skrip inisialisasi, skrip start-up berjalan terlebih dahulu. **Note** Skrip inisialisasi bersumber dari shell yang sama dengan pekerjaan notebook. Ini tidak berlaku untuk skrip start-up yang dijelaskan sebelumnya. Juga, jika Anda menggunakan skrip start-up dan skrip inisialisasi, skrip start-up berjalan terlebih dahulu.	Skrip Inisialisasi Bidang. Masukkan jalur file lokal tempat skrip lokal atau skrip Konfigurasi Siklus Hidup (LCC) berada.	Parameter `initialization_script` . Default ke `None`.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Upaya coba lagi maksimal	Berapa kali Studio mencoba menjalankan kembali pekerjaan yang gagal.	Upaya coba lagi Field Max. Default-nya adalah 1.	Default-nya adalah 1.	Parameter `max_retry_attempts` . Default-nya adalah 1.
Waktu lari maks (dalam detik)	Panjang waktu maksimum, dalam detik, bahwa pekerjaan notebook dapat berjalan sebelum dihentikan. Jika Anda mengonfig urasi upaya Max run time dan Max retry, waktu berjalan berlaku untuk setiap percobaan ulang. Jika pekerjaan tidak selesai saat ini, statusnya diatur ke`Failed`.	Field Max run time (dalam hitungan detik). Default ke `172800 seconds (2 days)`.	Mulai Visual Studio.	Parameter `max_runtime_in_seconds` . Default ke `172800 seconds (2 days)`.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Pertanyaan dan jawaban	Daftar kebijakan coba lagi, yang mengatur tindakan yang harus diambil jika terjadi kegagalan.	Tidak didukung.	Tidak didukung.	Parameter `retry_policies` . Default ke `None`.
Tambah `Step` atau `StepCollection` dependensi	Daftar `Step` atau `StepCollection` nama atau contoh di mana pekerjaan tergantung.	Tidak didukung.	Tidak didukung.	Parameter `depends_on` . Default ke `None`. Gunakan ini untuk menentukan dependensi eksplisit di antara langkah-langkah dalam grafik pipeline Anda.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Ukuran volume	Ukuran dalam GB volume penyimpanan untuk menyimpan data input dan output selama pelatihan.	Tidak didukung.	Tidak didukung.	Parameter `volume_size` . Default ke 30GB.
Enkripsi lalu lintas antar kontainer	Bendera yang menentukan apakah lalu lintas antar wadah pelatihan dienkripsi untuk pekerjaan pelatihan.	N/A. Diaktifkan secara default.	N/A. Diaktifkan secara default.	Parameter `encrypt_inter_container_traffic` . Default ke `True`.
Konfigurasi enkripsi tugas	Indikator bahwa Anda ingin mengenkripsi output pekerjaan notebook, volume instans pekerjaan, atau keduanya.	Bidang Konfigurasi enkripsi pekerjaan. Centang kotak ini untuk memilih enkripsi. Jika dibiarkan tidak dicentang, output pekerjaan dienkripsi dengan kunci KMS default akun dan volume instance pekerjaan tidak dienkripsi.	Mulai Visual Studio.	Tidak didukung.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Kunci KMS enkripsi keluaran	Kunci KMS untuk digunakan jika Anda ingin menyesuaikan kunci enkripsi yang digunakan untuk output pekerjaan notebook Anda. Bidang ini hanya berlaku jika Anda memeriksa Konfigurasi enkripsi pekerjaan.	Kunci KMS enkripsi Keluaran Bidang. Jika Anda tidak menentukan bidang ini, output pekerjaan notebook Anda dienkripsi dengan SSE-KMS menggunakan kunci Amazon S3 KMS default. Selain itu, jika Anda membuat bucket Amazon S3 sendiri dan menggunakan enkripsi, metode enkripsi Anda akan dipertahankan.	Mulai Visual Studio. Untuk bidang ini, Anda dapat mengatur default pengguna Anda sendiri yang telah terisi sebelumnya setiap kali Anda membuat definisi pekerjaan baru. Untuk detailnya, lihat Menyiapkan opsi default untuk buku catatan lokal.	Parameter `s3_kms_key` . Default ke `None`. Memungkinkan default cerdas.
Kunci KMS enkripsi volume contoh Job	Kunci KMS untuk digunakan jika Anda ingin mengenkripsi volume instance pekerjaan Anda. Bidang ini hanya berlaku jika Anda memeriksa Konfigurasi enkripsi pekerjaan.	Kunci KMS enkripsi volume instance Field Job.	Kunci KMS enkripsi volume instance Field Job. Untuk bidang ini, Anda dapat mengatur default pengguna Anda sendiri yang telah terisi sebelumnya setiap kali Anda membuat definisi pekerjaan baru. Untuk detailnya, lihat Menyiapkan opsi default untuk buku catatan lokal.	Parameter `volume_kms_key` . Default ke `None`. Memungkinkan default cerdas.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Gunakan Virtual Private Cloud untuk menjalankan pekerjaan ini (untuk pengguna VPC)	Indikator bahwa Anda ingin menjalankan tugas ini di Virtual Private Cloud (VPC). Untuk keamanan yang lebih baik, Anda disarankan untuk menggunakan VPC pribadi.	Bidang Gunakan Virtual Private Cloud untuk menjalankan pekerjaan ini. Centang kotak ini jika Anda ingin menggunakan VPC. Minimal, buat titik akhir VPC berikut untuk memungkinkan pekerjaan notebook Anda terhubung secara pribadi ke sumber daya tersebut: AWS • SageMaker: Untuk informasi tentang cara menghubungkan SageMaker melalui titik akhir antarmuka VPC, lihat. Connect ke SageMaker Dalam VPC • Amazon S3: Untuk informasi tentang cara menyambung ke Amazon S3 melalui titik akhir antarmuka VPC, lihat Titik akhir Gateway untuk Amazon S3. • Amazon EC2: Untuk informasi tentang cara menyambung ke Amazon EC2 melalui titik akhir	Mulai Visual Studio.	N/A

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
		antarmuka VPC, lihat Mengakses Amazon EC2 menggunakan titik akhir VPC antarmuka. • Amazon EventBridge: Titik akhir ini hanya diperlukan saat menyiapkan buku catatan terjadwal. Itu tidak diperlukan saat meluncurkan tugas sesuai permintaan. Untuk informasi tentang cara menyambung EventBridge melalui titik akhir antarmuka VPC, lihat Menggunakan dengan EventBridge antarmuka VPC Endpoint. Jika Anda memilih untuk menggunakan VPC, Anda perlu menentukan setidaknya satu subnet pribadi dan setidaknya satu grup keamanan dalam opsi berikut. Jika Anda tidak menggunakan subnet pribadi apa pun, Anda perlu mempertim		

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
		bangkan opsi konfigurasi lainnya. Untuk detailnya , lihat Subnet VPC Publik yang tidak didukung. Kendala dan pertimbangan		
Subnet (untuk pengguna VPC)	Subnet Anda. Bidang ini harus berisi setidakny a satu dan paling banyak lima, dan semua subnet yang Anda berikan harus bersifat pribadi. Untuk detailnya, lihat Subnet VPC Publik yang tidak didukung. Kendala dan pertimbangan	Subnet Bidang. Bidang ini default ke subnet yang terkait dengan domain Studio, tetapi Anda dapat mengubah bidang ini jika diperlukan.	Subnet Bidang. Penjadwal tidak dapat mendeteksi subnet Anda, jadi Anda harus memasukkan subnet apa pun yang Anda konfigurasi untuk VPC Anda.	Parameter `subnets`. Default ke `None`. Memungkin kan default cerdas.
Grup keamanar (untuk pengguna VPC)	Grup keamanan Anda. Bidang ini harus berisi setidaknya satu dan paling banyak 15. Untuk detailnya , lihat Subnet VPC Publik yang tidak didukung. Kendala dan pertimbangan	Kelompok Keamanan Lapangan. Bidang ini default ke grup keamanan yang terkait dengan VPC domain, tetapi Anda dapat mengubah bidang ini jika diperlukan.	Kelompok Keamanan Lapangan. Penjadwal tidak dapat mendeteksi grup keamanan Anda, jadi Anda harus memasukkan grup keamanan apa pun yang Anda konfigurasikan untuk VPC Anda.	Parameter `security_group_ids` . Default ke `None`. Memungkin kan default cerdas.

Opsi kustom	Deskripsi	Pedoman khusus studio	Pedoman lingkungan Jupyter lokal	SageMaker Pedoman Python SDK
Nama	Nama langkah tugas notebook.	N/A	N/A	Parameter `name`. Jika tidak ditentukan, itu berasal dari nama file notebook.
Nama tampilan	Nama pekerjaan Anda seperti yang akan muncul dalam daftar eksekusi pipeline Anda.	N/A	N/A	Parameter `display_name` . Default ke `None`.
Deskripsi	Deskripsi tugas Anda.	N/A	N/A	Parameter `description` .

Parameterisasi notebook Anda

Untuk meneruskan parameter baru atau penggantian parameter ke pekerjaan buku catatan terjadwal, Anda mungkin ingin memodifikasi buku catatan Jupyter jika Anda ingin nilai parameter baru diterapkan setelah sel. Saat Anda melewati parameter, pelaksana pekerjaan notebook menggunakan metodologi yang diberlakukan oleh Papermill. Pelaksana pekerjaan notebook mencari sel Jupyter yang ditandai dengan `parameters` tag dan menerapkan parameter baru atau penggantian parameter segera setelah sel ini. Jika Anda tidak memiliki sel yang ditandai`parameters`, parameter

diterapkan di awal buku catatan. Jika Anda memiliki lebih dari satu sel yang ditandai`parameters`, parameter diterapkan setelah sel pertama ditandai dengan. `parameters`

Untuk menandai sel di buku catatan Anda dengan `parameters` tag, selesaikan langkah-langkah berikut:

1. Pilih sel untuk membuat parameter.
2. Pilih ikon Property Inspector
   ()
   di sidebar kanan.
3. Ketik **`parameters`** di kotak Tambah Tag.
4. Pilih tanda +.
5. `parameters`Tag muncul di bawah Tag Sel dengan tanda centang, yang berarti tag diterapkan ke sel.

Connect ke klaster Amazon EMR dari notebook Anda

Jika Anda terhubung ke kluster EMR Amazon dari notebook Jupyter di Studio, Anda mungkin perlu melakukan penyiapan tambahan. Secara khusus, diskusi berikut membahas dua masalah:

- Meneruskan parameter ke perintah koneksi EMR Amazon Anda. Dalam SparkMagic kernel, parameter yang Anda berikan ke perintah koneksi EMR Amazon mungkin tidak berfungsi seperti yang diharapkan karena perbedaan cara Papermill melewati parameter dan SparkMagic cara menerima parameter. Solusi untuk mengatasi batasan ini adalah dengan meneruskan parameter sebagai variabel lingkungan. Untuk detail selengkapnya tentang masalah dan solusinya, lihat. Teruskan parameter ke perintah koneksi EMR Anda
- Meneruskan kredensyal pengguna ke cluster EMR Amazon yang diautentikasi oleh Kerberos, LDAP, atau HTTP Basic Auth-autentikasi. Dalam mode interaktif, Studio meminta kredensyal dalam bentuk popup tempat Anda dapat memasukkan kredensyal masuk Anda. Dalam buku catatan terjadwal noninteraktif Anda, Anda harus melewati mereka melalui. AWS Secrets Manager Untuk detail selengkapnya tentang cara menggunakan pekerjaan buku catatan terjadwal Anda, lihatTeruskan kredensyal pengguna ke cluster EMR Amazon yang diautentikasi Auth-Kerberos, LDAP, atau HTTP Basic Auth-Anda. AWS Secrets Manager

Teruskan parameter ke perintah koneksi EMR Anda

Jika Anda menggunakan gambar dengan kernel SparkMagic PySpark dan Spark dan ingin membuat parameter perintah koneksi EMR Anda, berikan parameter Anda di bidang variabel Lingkungan alih-alih bidang Parameter dalam formulir Buat Job (di menu tarik-turun Opsi Tambahan). Pastikan perintah koneksi EMR Anda di notebook Jupyter melewati parameter ini sebagai variabel lingkungan. Misalnya, Anda lulus `cluster-id` sebagai variabel lingkungan saat Anda membuat pekerjaan Anda. Perintah koneksi EMR Anda akan terlihat seperti berikut ini:

```
%%local
import os
```

```
%sm_analytics emr connect –cluster-id {os.getenv('cluster_id')} --auth-type None
```

Anda memerlukan solusi ini untuk memenuhi persyaratan oleh SparkMagic dan Papermill. Untuk konteks latar belakang, SparkMagic kernel mengharapkan bahwa perintah `%%local` ajaib menyertai variabel lokal apa pun yang Anda tentukan. Namun, Papermill tidak lulus perintah `%%local` ajaib dengan penggantian Anda. Untuk mengatasi batasan Papermill ini, Anda harus menyediakan parameter Anda sebagai variabel lingkungan di bidang variabel Lingkungan.

Teruskan kredensyal pengguna ke cluster EMR Amazon yang diautentikasi Auth-Kerberos, LDAP, atau HTTP Basic Auth-Anda

Untuk membuat koneksi aman ke kluster EMR Amazon yang menggunakan otentikasi Kerberos, LDAP, atau HTTP Basic Auth, Anda menggunakan AWS Secrets Manager untuk meneruskan kredensyal pengguna ke perintah koneksi Anda. Untuk informasi tentang cara membuat rahasia Secrets Manager, lihat Membuat AWS Secrets Manager rahasia. Rahasia Anda harus berisi nama pengguna dan kata sandi Anda. Anda meneruskan rahasia dengan `--secrets` argumen, seperti yang ditunjukkan pada contoh berikut:

```
%sm_analytics emr connect --cluster-id j_abcde12345
    --auth Kerberos
    --secret aws_secret_id_123
```

Administrator Anda dapat menyiapkan kebijakan akses fleksibel menggunakan metode attribute-based-access-control (ABAC), yang menetapkan akses berdasarkan tag khusus. Anda dapat mengatur akses fleksibel untuk membuat satu rahasia untuk semua pengguna di akun atau rahasia untuk setiap pengguna. Sampel kode berikut mendemonstrasikan skenario ini:

Buat satu rahasia untuk semua pengguna di akun

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Effect": "Allow",
            "Principal" : {"AWS" : "arn:aws:iam::AWS_ACCOUNT_ID:role/service-role/
AmazonSageMaker-ExecutionRole-20190101T012345"},

            "Action" : "secretsmanager:GetSecretValue",
            "Resource" : [  "arn:aws:secretsmanager:us-
west-2:AWS_ACCOUNT_ID:secret:aes123-1a2b3c",
                            "arn:aws:secretsmanager:us-
west-2:AWS_ACCOUNT_ID:secret:aes456-4d5e6f",
                            "arn:aws:secretsmanager:us-
west-2:AWS_ACCOUNT_ID:secret:aes789-7g8h9i" ]
        }
    ]
}
```

Buat rahasia yang berbeda untuk setiap pengguna

Anda dapat membuat rahasia yang berbeda untuk setiap pengguna menggunakan `PrincipleTag` tag, seperti yang ditunjukkan dalam contoh berikut:

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Effect": "Allow",
            "Principal" : {"AWS" : "arn:aws:iam::AWS_ACCOUNT_ID:role/service-role/
AmazonSageMaker-ExecutionRole-20190101T012345"},
            "Condition" : {
                "StringEquals" : {
                    "aws:ResourceTag/user-identity": "${aws:PrincipalTag/user-
identity}"
                }
            },
            "Action" : "secretsmanager:GetSecretValue",
            "Resource" : [  "arn:aws:secretsmanager:us-
west-2:AWS_ACCOUNT_ID:secret:aes123-1a2b3c",
```

```
                            "arn:aws:secretsmanager:us-
west-2:AWS_ACCOUNT_ID:secret:aes456-4d5e6f",
                            "arn:aws:secretsmanager:us-
west-2:AWS_ACCOUNT_ID:secret:aes789-7g8h9i" ]
        }
    ]
}
```

Lacak pekerjaan notebook dan definisi pekerjaan

SageMaker Dasbor Pekerjaan Notebook membantu mengatur definisi pekerjaan yang Anda jadwalkan, dan juga melacak pekerjaan aktual yang dijalankan dari definisi pekerjaan Anda. Ada dua konsep penting yang harus dipahami saat menjadwalkan pekerjaan notebook: definisi pekerjaan dan pekerjaan berjalan. Definisi Job adalah jadwal yang Anda tetapkan untuk menjalankan buku catatan tertentu. Misalnya, Anda dapat membuat definisi pekerjaan yang menjalankan notebook XYZ.IpyNB setiap hari Rabu. Definisi pekerjaan ini meluncurkan pekerjaan aktual yang terjadi Rabu mendatang, Rabu depan, Rabu setelah itu, dan seterusnya.

> **Note**
>
> Langkah pekerjaan notebook SDK SageMaker Python tidak membuat definisi pekerjaan. Namun, Anda dapat melihat pekerjaan Anda di dasbor Pekerjaan Notebook. Baik pekerjaan dan definisi pekerjaan tersedia jika Anda menjadwalkan pekerjaan Anda di suatu JupyterLab lingkungan.

Antarmuka menyediakan dua tab utama yang membantu Anda melacak definisi pekerjaan yang ada dan menjalankan pekerjaan:

- Tab Pekerjaan Notebook: Tab ini menampilkan daftar semua pekerjaan Anda berjalan dari pekerjaan sesuai permintaan dan definisi pekerjaan Anda. Dari tab ini, Anda dapat langsung mengakses detail untuk satu pekerjaan yang dijalankan. Misalnya, Anda dapat melihat satu pekerjaan berjalan yang terjadi dua hari Rabu lalu.
- Notebook Job Definitions tab: Tab ini menampilkan daftar semua definisi pekerjaan Anda. Dari tab ini, Anda dapat langsung mengakses detail untuk satu definisi pekerjaan. Misalnya, Anda dapat melihat jadwal yang Anda buat untuk menjalankan XYZ.ipynb setiap hari Rabu.

Untuk detail tentang tab Pekerjaan Notebook, lihatMelihat tugas Anda.

Untuk detail tentang tab Definisi Pekerjaan Notebook, lihatMelihat definisi tugas notebook.

Melihat tugas Anda

> **Note**
>
> Anda dapat melihat pekerjaan notebook secara otomatis jika menjadwalkan pekerjaan notebook dari UI Studio. Jika Anda menggunakan SageMaker Python SDK untuk menjadwalkan pekerjaan notebook Anda, Anda perlu menyediakan tag tambahan saat membuat langkah pekerjaan notebook. Untuk detailnya, lihat Melihat pekerjaan notebook Anda di dasbor Studio UI.

Tab Pekerjaan Notebook (yang Anda akses dengan memilih ikon Buat pekerjaan buku catatan ( ) di toolbar Studio) menampilkan riwayat pekerjaan sesuai permintaan Anda dan semua pekerjaan yang dijalankan dari definisi pekerjaan yang Anda buat. Tab ini terbuka setelah Anda membuat pekerjaan sesuai permintaan, atau Anda dapat melihat sendiri tab ini untuk melihat riwayat pekerjaan masa lalu dan saat ini. Jika Anda memilih nama Job untuk pekerjaan apa pun, Anda dapat melihat detail untuk satu pekerjaan di halaman Job Detail. Untuk informasi selengkapnya tentang halaman Job Detail, lihat bagian berikutMelihat tugas Anda.

Tab Pekerjaan Notebook menyertakan informasi berikut untuk setiap pekerjaan:

- File keluaran: Menampilkan ketersediaan file output. Kolom ini dapat berisi salah satu dari berikut ini:
  - Ikon unduhan ( ): Notebook keluaran dan log tersedia untuk diunduh; pilih tombol ini untuk mengunduhnya. Perhatikan bahwa pekerjaan yang gagal masih dapat menghasilkan file keluaran jika kegagalan terjadi setelah file dibuat. Dalam hal ini, akan sangat membantu untuk melihat notebook keluaran untuk mengidentifikasi titik kegagalan.
  - Tautan ke log Notebook dan Output: Notebook dan log keluaran diunduh. Pilih tautan untuk melihat isinya.
  - (blank): Pekerjaan dihentikan oleh pengguna, atau terjadi kegagalan dalam pekerjaan yang dijalankan sebelum dapat menghasilkan file keluaran. Misalnya, kegagalan jaringan dapat mencegah pekerjaan dimulai.

Notebook keluaran adalah hasil dari menjalankan semua sel di notebook, dan juga menggabungkan parameter atau variabel lingkungan baru atau utama yang Anda sertakan. Log keluaran menangkap detail pekerjaan yang dijalankan untuk membantu Anda memecahkan masalah pekerjaan yang gagal.

- Dibuat di: Waktu pekerjaan sesuai permintaan atau pekerjaan terjadwal dibuat.
- Status: Status pekerjaan saat ini, yang merupakan salah satu dari nilai berikut:
  - Sedang berlangsung: Pekerjaan sedang berjalan
  - Gagal: Pekerjaan gagal karena kesalahan konfigurasi atau logika notebook
  - Berhenti: Pekerjaan dihentikan oleh pengguna
  - Selesai: Pekerjaan selesai
- Tindakan: Kolom ini menyediakan pintasan untuk membantu Anda menghentikan atau menghapus pekerjaan apa pun secara langsung di antarmuka.

Melihat tugas Anda

Dari tab Pekerjaan Notebook, Anda dapat memilih nama pekerjaan untuk melihat halaman Detail Pekerjaan untuk pekerjaan tertentu. Halaman Detil Pekerjaan mencakup semua detail yang Anda berikan dalam formulir Buat Job. Gunakan halaman ini untuk mengonfirmasi pengaturan yang Anda tentukan saat Anda membuat definisi pekerjaan.

Selain itu, Anda dapat mengakses pintasan untuk membantu Anda melakukan tindakan berikut di halaman itu sendiri:

- Hapus Job: Hapus pekerjaan dari tab Pekerjaan Notebook.
- Stop Job: Hentikan pekerjaan Anda.

Melihat definisi tugas notebook

> **Note**
>
> Jika Anda menjadwalkan pekerjaan notebook Anda dengan SageMaker Python SDK, lewati bagian ini. Hanya pekerjaan notebook yang dibuat di Studio atau JupyterLab lingkungan lokal yang membuat definisi pekerjaan. Oleh karena itu, jika Anda membuat pekerjaan notebook dengan SageMaker Python SDK, Anda tidak akan melihat definisi pekerjaan di dasbor

Pekerjaan Notebook. Namun, Anda dapat melihat pekerjaan notebook Anda seperti yang dijelaskan dalamMelihat tugas Anda.

Saat Anda membuat ketentuan tugas, Anda membuat jadwal untuk suatu pekerjaan. Tab Notebook Job Definitions mencantumkan jadwal ini. Misalnya, Anda dapat membuat definisi pekerjaan yang menjalankan buku catatan tertentu setiap menit. Setelah definisi pekerjaan ini aktif, Anda melihat pekerjaan baru setiap menit di tab Pekerjaan Notebook.

Tab Definisi Job Notebook menampilkan dasbor dengan semua definisi pekerjaan Anda dan menyertakan buku catatan masukan, waktu pembuatan, jadwal, dan status untuk setiap definisi pekerjaan. Nilai di kolom Status adalah salah satu nilai berikut:

- Dijeda: Anda menghentikan sementara definisi pekerjaan. Studio tidak memulai pekerjaan apa pun sampai Anda melanjutkan definisi.
- Aktif: Jadwal aktif dan Studio dapat menjalankan notebook sesuai dengan jadwal yang Anda tentukan.

Selain itu, kolom Tindakan menyediakan pintasan untuk membantu Anda melakukan tugas-tugas berikut secara langsung di antarmuka:

- Jeda: Menjeda definisi pekerjaan. Studio tidak akan membuat pekerjaan apa pun sampai Anda melanjutkan definisi.
- Hapus: Menghapus definisi pekerjaan dari tab Definisi Job Notebook.
- Resume: Melanjutkan definisi pekerjaan yang dijeda sehingga dapat memulai pekerjaan.

Jika Anda membuat definisi pekerjaan tetapi tidak memulai pekerjaan, lihat Definisi Job tidak menciptakan pekerjaan di. Panduan pemecahan masalah

Melihat satu definisi tugas

Jika Anda memilih nama definisi pekerjaan di tab Definisi Pekerjaan Buku Catatan, Anda akan melihat halaman Definisi Pekerjaan tempat Anda dapat melihat detail spesifik untuk definisi pekerjaan. Gunakan halaman ini untuk mengonfirmasi pengaturan yang Anda tentukan saat Anda membuat definisi pekerjaan. Jika Anda tidak melihat pekerjaan apa pun yang dibuat dari definisi pekerjaan Anda, lihat Definisi Job tidak menciptakan pekerjaan diPanduan pemecahan masalah.

Halaman ini juga berisi bagian yang mencantumkan pekerjaan yang dijalankan dari definisi pekerjaan ini. Melihat pekerjaan Anda di halaman Definisi Pekerjaan mungkin merupakan cara yang lebih produktif untuk membantu Anda mengatur pekerjaan Anda daripada melihat pekerjaan di tab Pekerjaan Notebook, yang menggabungkan semua pekerjaan dari semua definisi pekerjaan Anda.

Selain itu, halaman ini menyediakan pintasan untuk tindakan berikut:

- Jeda/Lanjutkan: Jeda definisi pekerjaan Anda, atau lanjutkan definisi yang dijeda. Perhatikan bahwa jika pekerjaan sedang berjalan untuk definisi ini, Studio tidak menghentikannya.
- Jalankan: Jalankan satu pekerjaan sesuai permintaan dari definisi pekerjaan ini. Opsi ini juga memungkinkan Anda menentukan parameter input yang berbeda ke buku catatan Anda sebelum memulai pekerjaan.
- Edit Definisi Job: Ubah jadwal definisi pekerjaan Anda. Anda dapat memilih interval waktu yang berbeda, atau Anda dapat memilih jadwal khusus menggunakan sintaks cron.
- Hapus Definisi Pekerjaan: Hapus definisi pekerjaan dari tab Definisi Pekerjaan Notebook. Perhatikan bahwa jika pekerjaan sedang berjalan untuk definisi ini, Studio tidak menghentikannya.

Panduan pemecahan masalah

Lihat panduan pemecahan masalah ini untuk membantu Anda men-debug kegagalan yang mungkin Anda alami saat pekerjaan notebook terjadwal berjalan.

Definisi Job tidak menciptakan pekerjaan

Jika definisi pekerjaan Anda tidak memulai pekerjaan apa pun, lihat kemungkinan penyebab berikut:

Izin tidak ada

- Peran yang diberikan ke definisi pekerjaan tidak memiliki hubungan kepercayaan dengan Amazon EventBridge. Artinya, EventBridge tidak dapat mengasumsikan peran tersebut.
- Peran yang ditetapkan ke definisi pekerjaan tidak memiliki izin untuk meneleponSageMaker:StartPipelineExecution.
- Peran yang ditetapkan ke definisi pekerjaan tidak memiliki izin untuk meneleponSageMaker:CreateTrainingJob.

EventBridge kuota terlampaui

Jika Anda melihat `Put*` kesalahan seperti contoh berikut, Anda melebihi EventBridge kuota. Untuk mengatasi ini, Anda dapat membersihkan proses yang tidak terpakai EventBridge , atau meminta AWS Support untuk menambah kuota Anda.

```
LimitExceededException) when calling the PutRule operation:
The requested resource exceeds the maximum number allowed
```

Untuk informasi selengkapnya tentang EventBridge kuota, lihat EventBridge Kuota Amazon.

Batas kuota pipa terlampaui

Jika Anda melihat kesalahan seperti contoh berikut, Anda melebihi jumlah pipeline yang dapat Anda jalankan. Untuk mengatasi ini, Anda dapat membersihkan saluran pipa yang tidak terpakai di akun Anda, atau meminta AWS Support untuk menambah kuota Anda.

```
ResourceLimitExceeded: The account-level service limit
'Maximum number of pipelines allowed per account' is XXX Pipelines,
with current utilization of XXX Pipelines and a request delta of 1 Pipelines.
```

Untuk informasi selengkapnya tentang kuota pipeline, lihat SageMaker titik akhir dan kuota Amazon.

Batas pekerjaan pelatihan terlampaui

Jika Anda melihat kesalahan seperti contoh berikut, Anda melebihi jumlah pekerjaan pelatihan yang dapat Anda jalankan. Untuk mengatasi ini, kurangi jumlah pekerjaan pelatihan di akun Anda, atau minta AWS Support untuk menambah kuota Anda.

```
ResourceLimitExceeded: The account-level service limit
'ml.m5.2xlarge for training job usage' is 0 Instances, with current
utilization of 0 Instances and a request delta of 1 Instances.
Please contact AWS support to request an increase for this limit.
```

Untuk informasi selengkapnya tentang kuota pekerjaan pelatihan, lihat SageMaker titik akhir dan kuota Amazon.

Visualisasi otomatis dinonaktifkan di notebook SparkMagic

Jika notebook Anda menggunakan SparkMagic PySpark kernel dan Anda menjalankan notebook sebagai Job Notebook, Anda mungkin melihat bahwa visualisasi otomatis Anda dinonaktifkan dalam output. Mengaktifkan visualisasi otomatis menyebabkan kernel macet, sehingga pelaksana pekerjaan notebook saat ini menonaktifkan visualisasi otomatis sebagai solusinya.

Kendala dan pertimbangan

Tinjau kendala berikut untuk memastikan pekerjaan notebook Anda berhasil diselesaikan. Studio menggunakan Papermill untuk menjalankan notebook. Anda mungkin perlu memperbarui notebook Jupyter untuk menyelaraskan dengan persyaratan Papermill. Ada juga batasan pada konten skrip LCC dan detail penting untuk dipahami mengenai konfigurasi VPC.

JupyterLab versi

JupyterLab versi 3.0 dan di atas didukung.

Instalasi paket yang memerlukan kernel restart

Papermill tidak mendukung panggilan `pip install` untuk menginstal paket yang memerlukan kernel restart. Dalam situasi ini, gunakan `pip install` dalam skrip inisialisasi. Untuk instalasi paket yang tidak memerlukan kernel restart, Anda masih dapat memasukkan `pip install` dalam notebook.

Kernel dan nama bahasa yang terdaftar di Jupyter

Papermill mendaftarkan penerjemah untuk kernel dan bahasa tertentu. Jika Anda membawa instance Anda sendiri (BYOI), gunakan nama kernel standar seperti yang ditunjukkan pada cuplikan berikut:

```
papermill_translators.register("python", PythonTranslator)
papermill_translators.register("R", RTranslator)
papermill_translators.register("scala", ScalaTranslator)
papermill_translators.register("julia", JuliaTranslator)
papermill_translators.register("matlab", MatlabTranslator)
papermill_translators.register(".net-csharp", CSharpTranslator)
papermill_translators.register(".net-fsharp", FSharpTranslator)
papermill_translators.register(".net-powershell", PowershellTranslator)
papermill_translators.register("pysparkkernel", PythonTranslator)
papermill_translators.register("sparkkernel", ScalaTranslator)
papermill_translators.register("sparkrkernel", RTranslator)
papermill_translators.register("bash", BashTranslator)
```

Parameter dan batas variabel lingkungan

Parameter dan batas variabel lingkungan. Saat Anda membuat pekerjaan notebook Anda, ia menerima parameter dan variabel lingkungan yang Anda tentukan. Anda dapat meneruskan hingga 100 parameter. Setiap nama parameter dapat mencapai 256 karakter, dan nilai yang terkait dapat

mencapai 2500 karakter. Jika Anda melewati variabel lingkungan, Anda dapat melewatkan hingga 28 variabel. Nama variabel dan nilai terkait dapat memiliki panjang hingga 512 karakter. Jika Anda membutuhkan lebih dari 28 variabel lingkungan, gunakan variabel lingkungan tambahan dalam skrip inisialisasi yang tidak memiliki batasan jumlah variabel lingkungan yang dapat Anda gunakan.

Melihat pekerjaan dan definisi pekerjaan

Melihat pekerjaan dan definisi pekerjaan. Jika Anda menjadwalkan pekerjaan notebook di UI Studio di JupyterLab buku catatan, Anda dapat melihat pekerjaan buku catatan dan definisi pekerjaan notebook Anda di UI Studio. Jika Anda menjadwalkan pekerjaan notebook Anda dengan SageMaker Python SDK, Anda hanya dapat melihat pekerjaan Anda—langkah pekerjaan notebook SDK SageMaker Python tidak membuat definisi pekerjaan. Untuk melihat pekerjaan Anda, Anda juga perlu memberikan tag tambahan ke instance langkah pekerjaan notebook Anda. Untuk detailnya, lihat Melihat pekerjaan notebook Anda di dasbor Studio UI.

Citra

Anda perlu mengelola batasan gambar tergantung pada apakah Anda menjalankan pekerjaan notebook di Studio atau langkah pekerjaan notebook SDK SageMaker Python dalam pipeline.

Kendala gambar untuk Pekerjaan SageMaker Notebook (Studio)

Dukungan gambar dan kernel. Driver yang meluncurkan pekerjaan notebook Anda mengasumsikan hal berikut:

- Lingkungan runtime Python dasar dipasang di gambar Studio atau bring-your-own (BYO) dan merupakan default di shell.
- Lingkungan runtime Python dasar mencakup klien Jupyter dengan spesifikasi kernel yang dikonfigurasi dengan benar.
- Lingkungan runtime Python dasar menyertakan `pip` fungsi sehingga pekerjaan notebook dapat menginstal dependensi sistem.
- Untuk gambar dengan beberapa lingkungan, skrip inisialisasi Anda harus beralih ke lingkungan khusus kernel yang tepat sebelum menginstal paket khusus notebook. Anda harus beralih kembali ke lingkungan runtime Python default, jika berbeda dari lingkungan runtime kernel, setelah mengonfigurasi lingkungan runtime Python kernel.

Driver yang meluncurkan pekerjaan notebook Anda adalah skrip bash, dan Bash v4 harus tersedia di /bin/bash.

Hak akses root pada bring-your-own-images (BYOI). Anda harus memiliki hak akses root pada gambar Studio Anda sendiri, baik sebagai pengguna root atau melalui `sudo` akses. Jika Anda bukan pengguna root tetapi mengakses hak akses root melalui`sudo`, gunakan **`1000/100`** sebagai. `UID/GID`

Kendala gambar untuk pekerjaan notebook Python SageMaker SDK

Langkah pekerjaan notebook mendukung gambar-gambar berikut:

- SageMaker Gambar Distribusi tercantum dalam SageMaker Gambar Amazon yang Tersedia.
- Gambar kustom berdasarkan gambar SageMaker Distribusi dalam daftar sebelumnya. Gunakan gambar SageMaker Distribusi sebagai basis.
- Gambar kustom (BYOI) yang sudah diinstal sebelumnya dengan dependensi pekerjaan notebook (yaitu,. sagemaker-headless-execution-driver Gambar Anda harus memenuhi persyaratan berikut:
  - Gambar sudah diinstal sebelumnya dengan dependensi pekerjaan notebook.
  - Lingkungan runtime Python dasar diinstal dan default di lingkungan shell.
  - Lingkungan runtime Python dasar mencakup klien Jupyter dengan spesifikasi kernel yang dikonfigurasi dengan benar.
  - Anda memiliki hak akses root, baik sebagai pengguna root atau melalui `sudo` akses. Jika Anda bukan pengguna root tetapi mengakses hak akses root melalui`sudo`, gunakan **`1000/100`** sebagai. `UID/GID`

Subnet VPC yang digunakan selama penciptaan lapangan kerja

Jika Anda menggunakan VPC, Studio menggunakan subnet pribadi Anda untuk membuat pekerjaan Anda. Tentukan satu hingga lima subnet pribadi (dan 1-15 grup keamanan).

Jika Anda menggunakan VPC dengan subnet pribadi, Anda harus memilih salah satu opsi berikut untuk memastikan pekerjaan notebook dapat terhubung ke layanan atau sumber daya yang bergantung:

- Jika pekerjaan membutuhkan akses ke AWS layanan yang mendukung titik akhir VPC antarmuka, buat titik akhir untuk terhubung ke layanan. Untuk daftar layanan yang mendukung titik akhir antarmuka, lihat AWSlayanan yang terintegrasi dengannya AWS PrivateLink. Untuk informasi tentang membuat titik akhir VPC antarmuka, lihat Mengakses AWS layanan menggunakan titik akhir VPC antarmuka. Minimal, gateway VPC Amazon S3 harus disediakan.

- Jika pekerjaan notebook memerlukan akses ke AWS layanan yang tidak mendukung titik akhir VPC antarmuka atau sumber daya di luarAWS, buat gateway NAT dan konfigurasikan grup keamanan Anda untuk mengizinkan koneksi keluar. Untuk informasi tentang menyiapkan gateway NAT untuk VPC Anda, lihat VPC dengan Subnet publik dan pribadi (NAT) di Panduan Pengguna Amazon Virtual Private Cloud.

Batas layanan

Karena penjadwal pekerjaan notebook dibuat dari SageMaker Pipelines, SageMaker Training, dan EventBridge layanan Amazon, pekerjaan notebook Anda tunduk pada kuota khusus layanan mereka. Jika Anda melebihi kuota ini, Anda mungkin melihat pesan kesalahan yang terkait dengan layanan ini. Misalnya, ada batasan untuk berapa banyak pipa yang dapat Anda jalankan pada satu waktu, dan berapa banyak aturan yang dapat Anda atur untuk satu bus acara. Untuk informasi selengkapnya tentang SageMaker kuota, lihat SageMaker Titik Akhir dan Kuota Amazon. Untuk informasi selengkapnya tentang EventBridge kuota, lihat EventBridge Kuota Amazon.

Harga untuk Pekerjaan SageMaker Notebook

Saat Anda menjadwalkan pekerjaan buku catatan, buku catatan Jupyter Anda berjalan pada SageMaker instance pelatihan. Setelah Anda memilih Image dan Kernel di formulir Create Job, formulir tersebut menyediakan daftar jenis komputasi yang tersedia. Anda dikenakan biaya untuk jenis komputasi yang Anda pilih, berdasarkan durasi penggunaan gabungan untuk semua pekerjaan notebook yang dijalankan dari definisi pekerjaan. Jika Anda tidak menentukan jenis komputasi, SageMaker berikan Anda jenis instans Amazon EC2 default. `ml.m5.large` Untuk rincian SageMaker harga berdasarkan jenis komputasi, lihat SageMakerHarga Amazon.

Pelacakan SageMaker Silsilah Amazon

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

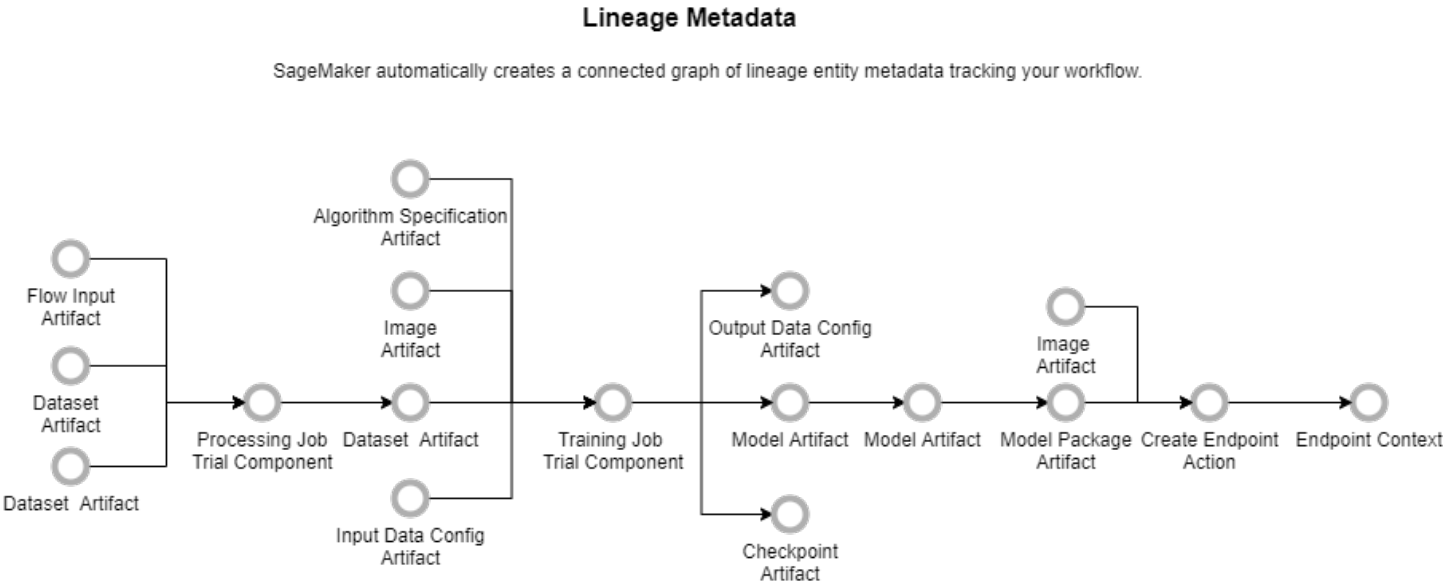
Amazon SageMaker ML Lineage Tracking membuat dan menyimpan informasi tentang langkah-langkah alur kerja machine learning (ML) mulai dari persiapan data hingga penerapan model.

Dengan informasi pelacakan, Anda dapat mereproduksi langkah alur kerja, melacak garis keturunan model dan kumpulan data, serta menetapkan tata kelola model dan standar audit.

Dengan SageMaker Lineage Tracking, para ilmuwan data dan pembuat model dapat melakukan hal berikut:

- Simpan riwayat eksperimen penemuan model.
- Menetapkan tata kelola model dengan melacak artefak garis keturunan model untuk audit dan verifikasi kepatuhan.

Diagram berikut menunjukkan contoh grafik garis keturunan yang SageMaker secara otomatis dibuat Amazon dalam pelatihan end-to-end model dan alur kerja MS penerapan.



Topik

- Entitas Pelacakan Silsilah
- Amazon SageMaker —Menciptakan Entitas Pelacakan
- Buat Entitas Pelacakan Secara Manual
- Meminta Entitas Silsilah
- Pelacakan Lintas Akun

Entitas Pelacakan Silsilah

Entitas pelacakan mempertahankan representasi semua elemen alur kerja pembelajaran end-to-end mesin Anda. Anda dapat menggunakan representasi ini untuk membuat tata kelola model, mereproduksi alur kerja Anda, dan memelihara catatan riwayat pekerjaan Anda.

Amazon SageMaker secara otomatis membuat entitas pelacakan untuk komponen uji coba dan uji coba serta eksperimen terkait saat Anda membuat SageMaker pekerjaan seperti memproses pekerjaan, pekerjaan pelatihan, dan pekerjaan transformasi batch. Selain pelacakan otomatis, Anda juga dapat Buat Entitas Pelacakan Secara Manual memodelkan langkah-langkah khusus dalam alur kerja Anda. Untuk informasi selengkapnya, lihat Kelola Machine Learning dengan Amazon SageMaker Experiments.

SageMaker juga secara otomatis membuat entitas pelacakan untuk langkah-langkah lain dalam alur kerja sehingga Anda dapat melacak alur kerja dari ujung ke ujung. Untuk informasi selengkapnya, lihat Amazon SageMaker —Menciptakan Entitas Pelacakan.

Anda dapat membuat entitas tambahan untuk melengkapi yang dibuat oleh SageMaker. Untuk informasi selengkapnya, lihat Buat Entitas Pelacakan Secara Manual.

SageMaker menggunakan kembali entitas yang ada daripada membuat yang baru. Misalnya, hanya ada satu artefak dengan unik`SourceUri`.

Konsep kunci untuk menanyakan garis keturunan

- Lineage — Metadata yang melacak hubungan antara berbagai entitas dalam alur kerja ML Anda.
- QueryLineage— Tindakan untuk memeriksa garis keturunan Anda dan menemukan hubungan antar entitas.
- Entitas garis keturunan — Elemen metadata yang menyusun garis keturunan Anda.
- Silsilah lintas akun — Alur kerja ML Anda mungkin menjangkau lebih dari satu akun. Dengan silsilah lintas akun, Anda dapat mengonfigurasi beberapa akun untuk secara otomatis membuat asosiasi garis keturunan antara sumber daya entitas bersama. QueryLineage kemudian dapat mengembalikan entitas bahkan dari akun bersama ini.

Entitas pelacakan berikut didefinisikan:

Ekstensi nama file

- Komponen percobaan — Tahap uji coba pembelajaran mesin. Termasuk pekerjaan pemrosesan, pekerjaan pelatihan, dan pekerjaan transformasi batch.
- Percobaan — Kombinasi komponen percobaan yang umumnya menghasilkan model.
- Eksperimen — Pengelompokan uji coba yang umumnya berfokus pada pemecahan kasus penggunaan tertentu.

Performa per detik

- Komponen Uji Coba - Merupakan pemrosesan, pelatihan, dan transformasi pekerjaan dalam garis keturunan. Juga bagian dari manajemen eksperimen.
- Konteks — Menyediakan pengelompokan logis dari entitas pelacakan atau eksperimen lainnya. Secara konseptual, eksperimen dan percobaan adalah konteks. Beberapa contoh adalah endpoint dan paket model.
- Tindakan - Merupakan tindakan atau aktivitas. Umumnya, tindakan melibatkan setidaknya satu artefak input atau artefak keluaran. Beberapa contoh adalah langkah alur kerja dan penerapan model.
- Artifact - Merupakan objek atau data yang dapat dialamatkan URI. Artefak umumnya berupa input atau output ke komponen percobaan atau tindakan. Beberapa contoh termasuk kumpulan data (URI bucket S3), atau gambar (jalur registri Amazon ECR).
- Asosiasi — Menghubungkan entitas pelacakan atau eksperimen lainnya, seperti hubungan antara lokasi data pelatihan dan pekerjaan pelatihan.

  Sebuah asosiasi memiliki `AssociationType` properti opsional. Nilai-nilai berikut tersedia bersama dengan penggunaan yang disarankan untuk setiap jenis. SageMaker tidak menempatkan batasan pada penggunaannya:

  - `ContributedTo`— Sumber berkontribusi pada tujuan atau memiliki bagian dalam memungkinkan tujuan. Misalnya, data pelatihan berkontribusi pada pekerjaan pelatihan.
  - `AssociatedWith`— Sumber terhubung ke tujuan. Misalnya, alur kerja persetujuan dikaitkan dengan penerapan model.
  - `DerivedFrom`- Tujuan adalah modifikasi dari sumbernya. Misalnya, output intisari dari input saluran untuk pekerjaan pemrosesan berasal dari input asli.
  - `Produced`— Sumber yang menghasilkan tujuan. Misalnya, pekerjaan pelatihan menghasilkan artefak model.

- `SameAs`— Ketika entitas silsilah yang sama digunakan dalam akun yang berbeda.

Properti umum

- Jenis properti

  Entitas aksi, artefak, dan konteks memiliki properti tipe,`ActionType`,`ArtifactType`, dan`ContextType`, masing-masing. Properti ini adalah string kustom yang dapat mengaitkan informasi yang bermakna dengan entitas dan digunakan sebagai filter dalam API Daftar.
- Properti sumber

  Entitas aksi, artefak, dan konteks memiliki `Source` properti. Properti ini menyediakan URI dasar yang diwakili entitas. Beberapa contohnya adalah:
  - `UpdateEndpoint`Tindakan di mana sumbernya adalah`EndpointArn`.
  - Artefak gambar untuk pekerjaan pemrosesan di mana sumbernya adalah. `ImageUri`
  - `Endpoint`Konteks di mana sumbernya adalah`EndpointArn`.
- Properti metadata

  Entitas aksi dan artefak memiliki `Metadata` properti opsional yang dapat memberikan informasi berikut:
  - `ProjectId`— Misalnya, ID proyek SageMaker MLOPs yang menjadi milik model.
  - `GeneratedBy`— Misalnya, eksekusi SageMaker pipeline yang mendaftarkan versi paket model.
  - `Repository`Misalnya, repositori yang berisi algoritma.
  - `CommitId`— Misalnya, ID komit dari versi algoritma.

Amazon SageMaker —Menciptakan Entitas Pelacakan

Amazon SageMaker secara otomatis membuat entitas pelacakan untuk SageMaker pekerjaan, model, paket model, dan titik akhir jika data tersedia. Tidak ada batasan jumlah entitas garis keturunan yang dibuat oleh. SageMaker

Untuk informasi tentang cara membuat entitas pelacakan secara manual, lihatBuat Entitas Pelacakan Secara Manual.

Topik

- Melacak Entitas untuk SageMaker Pekerjaan

- Melacak Entitas untuk Paket Model
- Melacak Entitas untuk Titik Akhir

Melacak Entitas untuk SageMaker Pekerjaan

SageMaker membuat komponen percobaan untuk dan terkait dengan setiap SageMaker pekerjaan. SageMaker membuat artefak untuk melacak metadata pekerjaan dan asosiasi antara setiap artefak dan pekerjaan.

Artefak dibuat untuk properti tugas berikut dan terkait dengan Amazon Resource Name (ARN) dari SageMaker tugas. Artefak `SourceUri` tercantum dalam tanda kurung.

Contoh Tugas

- Gambar yang berisi algoritma pelatihan (`TrainingImage`).
- Sumber data dari setiap saluran input (`S3Uri`).
- Lokasi untuk model (`S3OutputPath`).
- Lokasi untuk data pos pemeriksaan spot terkelola (`S3Uri`).

Melacak tugas

- Kontainer yang akan dijalankan oleh tugas pemrosesan (`ImageUri`).
- Lokasi data untuk setiap input pemrosesan dan output pemrosesan (`S3Uri`).

Melacak tugas

- Sumber data input yang akan diubah (`S3Uri`).
- Hasil transformasi (`S3OutputPath`).

> **Note**
>
> Artefak Amazon Simple Storage Service (Amazon S3) dilacak berdasarkan nilai URI Amazon S3 yang disediakan untuk Create API, misalnya, CreateTrainingJobdan bukan pada kunci Amazon S3 dan nilai hash atau etag dari setiap file.

Melacak Entitas untuk Paket Model

Entitas berikut dibuat:

Model paket

- Konteks untuk setiap grup paket model.
- Artefak untuk setiap paket model.
- Hubungan antara setiap artefak paket model dan konteks untuk setiap grup paket model yang menjadi milik paket tersebut.
- Tindakan untuk pembuatan versi paket model.
- Hubungan antara artefak paket model dan tindakan pembuatan.
- Hubungan antara artefak paket model dan setiap konteks grup paket model yang menjadi milik paket tersebut.
- Wadah inferensi
  - Artefak untuk gambar yang digunakan dalam setiap wadah didefinisikan dalam paket model.
  - Artefak untuk model yang digunakan di setiap wadah.
  - Hubungan antara setiap artefak dan artefak paket model.
- Algoritma
  - Artefak untuk setiap algoritma yang didefinisikan dalam paket model.
  - Artefak untuk model yang dibuat oleh setiap algoritma.
  - Hubungan antara setiap artefak dan artefak paket model.

Melacak Entitas untuk Titik Akhir

Entitas berikut dibuat oleh Amazon SageMaker:

Titik akhir

- Konteks untuk setiap titik akhir
- Tindakan untuk penerapan model yang membuat setiap titik akhir
- Artefak untuk setiap model yang digunakan ke titik akhir
- Artefak untuk gambar yang digunakan dalam model
- Artefak untuk paket model untuk model
- Artefak untuk setiap gambar yang disebarkan ke titik akhir

- Hubungan antara setiap artefak dan tindakan penerapan model

Buat Entitas Pelacakan Secara Manual

Anda dapat secara manual membuat entitas pelacakan untuk properti apa pun. Untuk informasi tentang entitas pelacakan yang dibuat Amazon SageMaker secara otomatis, lihatAmazon SageMaker —Menciptakan Entitas Pelacakan.

Anda dapat menambahkan tag ke semua entitas kecuali asosiasi. Tanda adalah pasangan nilai kunci arbitrer yang memberikan informasi kustom. Anda dapat memfilter atau mengurutkan daftar atau kueri pencarian berdasarkan tag. Untuk informasi selengkapnya, lihat Menandai AWS sumber daya di. Referensi Umum AWS

Untuk contoh buku catatan yang menunjukkan cara membuat entitas garis keturunan, lihat SageMakernotebook Amazon Lineage di repositori contoh Amazon. SageMaker GitHub

Topik

- Buat Entitas Secara Manual
- Melacak Alur Kerja Secara Manual
- Batas

Buat Entitas Secara Manual

Prosedur berikut menunjukkan cara membuat dan menghubungkan artefak antara tugas SageMaker pelatihan dan titik akhir. Anda harus melakukan langkah-langkah berikut ini:

Impor entitas dan asosiasi pelacak

1. Impor entitas pelacak garis keturunan.

```
import sys
!{sys.executable} -m pip install -q sagemaker

from sagemaker import get_execution_role
from sagemaker.session import Session
from sagemaker.lineage import context, artifact, association, action

import boto3
boto_session = boto3.Session(region_name=region)
```

```
sagemaker_client = boto_session.client("sagemaker")
```

2. Buat artefak input dan output.

```
code_location_arn = artifact.Artifact.create(
    artifact_name='source-code-location',
    source_uri='s3://...',
    artifact_type='code-location'
).artifact_arn

# Similar constructs for train_data_location_arn and test_data_location_arn

model_location_arn = artifact.Artifact.create(
    artifact_name='model-location',
    source_uri='s3://...',
    artifact_type='model-location'
).artifact_arn
```

3. Latih model dan dapatkan `trial_component_arn` yang mewakili pekerjaan pelatihan.
4. Kaitkan artefak input dan artefak keluaran dengan pekerjaan pelatihan (komponen percobaan).

```
input_artifacts = [code_location_arn, train_data_location_arn,
 test_data_location_arn]
for artifact_arn in input_artifacts:
    try:
        association.Association.create(
            source_arn=artifact_arn,
            destination_arn=trial_component_arn,
            association_type='ContributedTo'
        )
    except:
        logging.info('association between {} and {} already exists', artifact_arn,
 trial_component_arn)

output_artifacts = [model_location_arn]
for artifact_arn in output_artifacts:
    try:
         association.Association.create(
            source_arn=trial_component_arn,
            destination_arn=artifact_arn,
            association_type='Produced'
        )
    except:
```

```
        logging.info('association between {} and {} already exists', artifact_arn,
 trial_component_arn)
```

5. Buat titik akhir inferensi.

```
predictor = mnist_estimator.deploy(initial_instance_count=1,
                                   instance_type='ml.m4.xlarge')
```

6. Buat konteks titik akhir.

```
from sagemaker.lineage import context

endpoint = sagemaker_client.describe_endpoint(EndpointName=predictor.endpoint_name)
endpoint_arn = endpoint['EndpointArn']

endpoint_context_arn = context.Context.create(
    context_name=predictor.endpoint_name,
    context_type='Endpoint',
    source_uri=endpoint_arn
).context_arn
```

7. Kaitkan pekerjaan pelatihan (komponen percobaan) dan konteks titik akhir.

```
association.Association.create(
    source_arn=trial_component_arn,
    destination_arn=endpoint_context_arn
)
```

Melacak Alur Kerja Secara Manual

Anda dapat melacak alur kerja yang dibuat di bagian sebelumnya.

Mengingat titik akhir Amazon Resource Name (ARN) dari contoh sebelumnya, prosedur berikut menunjukkan cara melacak alur kerja kembali ke kumpulan data yang digunakan untuk melatih model yang diterapkan ke titik akhir. Anda harus melakukan langkah-langkah berikut ini:

Untuk melacak alur kerja dari titik akhir ke sumber data pelatihan

1. Impor entitas pelacakan.

```
import sys
```

```
!{sys.executable} -m pip install -q sagemaker

from sagemaker import get_execution_role
from sagemaker.session import Session
from sagemaker.lineage import context, artifact, association, action

import boto3
boto_session = boto3.Session(region_name=region)
sagemaker_client = boto_session.client("sagemaker")
```

2. Dapatkan konteks titik akhir dari ARN titik akhir.

```
endpoint_context_arn = sagemaker_client.list_contexts(
    SourceUri=endpoint_arn)['ContextSummaries'][0]['ContextArn']
```

3. Dapatkan komponen uji coba dari asosiasi antara komponen uji coba dan konteks titik akhir.

```
trial_component_arn = sagemaker_client.list_associations(
    DestinationArn=endpoint_context_arn)['AssociationSummaries'][0]['SourceArn']
```

4. Dapatkan artefak lokasi data pelatihan dari asosiasi antara komponen uji coba dan konteks titik akhir.

```
train_data_location_artifact_arn = sagemaker_client.list_associations(
    DestinationArn=trial_component_arn, SourceType='Model')['AssociationSummaries']
[0]['SourceArn']
```

5. Dapatkan lokasi data pelatihan dari artefak lokasi data pelatihan.

```
train_data_location = sagemaker_client.describe_artifact(
    ArtifactArn=train_data_location_artifact_arn)['Source']['SourceUri']
    print(train_data_location)
```

Respons:

```
s3://sagemaker-sample-data-us-east-2/mxnet/mnist/train
```

Batas

Anda dapat membuat asosiasi antara entitas, eksperimen, dan garis keturunan apa pun, kecuali yang berikut ini:

- Anda tidak dapat membuat asosiasi antara dua entitas eksperimen. Entitas eksperimen terdiri dari eksperimen, uji coba, dan komponen percobaan.
- Anda dapat membuat asosiasi dengan asosiasi lain.

Terjadi kesalahan jika Anda mencoba membuat entitas yang sudah ada.

Jumlah maksimum entitas silsilah yang dibuat secara manual

- 0,1 per detik
- Artefak: 6000
- Asosiasi: 6000
- Mengekspor:

Tidak ada batasan jumlah entitas garis keturunan yang dibuat secara otomatis oleh Amazon. SageMaker

Meminta Entitas Silsilah

Amazon SageMaker secara otomatis menghasilkan grafik entitas garis keturunan saat Anda menggunakannya. Anda dapat menanyakan data ini untuk menjawab berbagai pertanyaan. Anda dapat menanyakan entitas silsilah Anda ke:

- Ambil semua kumpulan data yang masuk ke dalam pembuatan model.
- Ambil semua pekerjaan yang masuk ke penciptaan titik akhir.
- Ambil semua model yang menggunakan kumpulan data.
- Ambil semua titik akhir yang menggunakan model.
- Ambil titik akhir mana yang berasal dari kumpulan data tertentu.
- Ambil eksekusi pipeline yang menciptakan pekerjaan pelatihan.
- Mengambil hubungan antara entitas untuk investigasi, tata kelola, dan reproduktifitas.
- Ambil semua uji coba hilir yang menggunakan artefak.

- Ambil semua uji coba hulu yang menggunakan artefak.
- Ambil daftar artefak yang menggunakan uri S3 yang disediakan.
- Ambil artefak hulu yang menggunakan artefak dataset.
- Ambil artefak hilir yang menggunakan artefak dataset.
- Ambil dataset yang menggunakan artefak gambar.
- Ambil tindakan yang menggunakan konteks.
- Ambil pekerjaan pemrosesan yang menggunakan titik akhir.
- Ambil pekerjaan transformasi yang menggunakan titik akhir.
- Ambil komponen percobaan yang menggunakan titik akhir.
- Ambil ARN untuk eksekusi pipeline yang terkait dengan grup paket model.
- Ambil semua artefak yang menggunakan tindakan.
- Ambil semua kumpulan data hulu yang menggunakan tindakan persetujuan paket model.
- Ambil paket model dari tindakan persetujuan paket model.
- Ambil konteks titik akhir hilir yang menggunakan titik akhir.
- Ambil ARN untuk eksekusi pipeline yang terkait dengan komponen uji coba.
- Ambil kumpulan data yang menggunakan komponen percobaan.
- Ambil model yang menggunakan komponen percobaan.
- Jelajahi garis keturunan Anda untuk visualisasi.

Batasan

- Kueri silsilah tidak tersedia di Wilayah berikut:
  - Afrika (Cape Town) — af-southern
  - Asia Pacific (Jakarta) — ap-southeast-3
  - Asia Pacific (Osaka) – ap-northeast-3
  - Eropa (Milan) — eu-south-1
  - Eropa (Spanyol) — eu-south-2
  - Israel (Tel Aviv) — tengah-1
- Kedalaman maksimum hubungan untuk ditemukan saat ini terbatas pada 10.
- Pemfilteran terbatas pada properti berikut: tanggal modifikasi terakhir, tanggal dibuat, jenis, dan jenis entitas garis keturunan.

Topik

- Memulai dengan Entitas Lineage

Memulai dengan Entitas Lineage

Cara termudah untuk memulai adalah melalui:

- Amazon SageMaker SDK untuk Python yang telah mendefinisikan banyak kasus penggunaan umum.
- Untuk buku catatan yang mendemonstrasikan cara menggunakan API SageMaker Lineage untuk menanyakan hubungan di seluruh grafik garis keturunan, lihat .ipynb. sagemaker-lineage-multihop-queries

Contoh berikut menunjukkan cara menggunakan `LineageFilter` API `LineageQuery` dan untuk membuat kueri guna menjawab pertanyaan tentang Grafik Lineage dan mengekstrak hubungan entitas untuk beberapa kasus penggunaan.

Example Menggunakan **`LineageQuery`** API untuk menemukan asosiasi entitas

```
from sagemaker.lineage.context import Context, EndpointContext
from sagemaker.lineage.action import Action
from sagemaker.lineage.association import Association
from sagemaker.lineage.artifact import Artifact, ModelArtifact, DatasetArtifact

from sagemaker.lineage.query import (
    LineageQuery,
    LineageFilter,
    LineageSourceEnum,
    LineageEntityEnum,
    LineageQueryDirectionEnum,
)
# Find the endpoint context and model artifact that should be used for the lineage
 queries.

contexts = Context.list(source_uri=endpoint_arn)
context_name = list(contexts)[0].context_name
endpoint_context = EndpointContext.load(context_name=context_name)
```

Example Temukan semua kumpulan data yang terkait dengan titik akhir

```
# Define the LineageFilter to look for entities of type `ARTIFACT` and the source of
 type `DATASET`.

query_filter = LineageFilter(
    entities=[LineageEntityEnum.ARTIFACT], sources=[LineageSourceEnum.DATASET]
)

# Providing this `LineageFilter` to the `LineageQuery` constructs a query that
 traverses through the given context `endpoint_context`
# and find all datasets.

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[endpoint_context.context_arn],
    query_filter=query_filter,
    direction=LineageQueryDirectionEnum.ASCENDANTS,
    include_edges=False,
)

# Parse through the query results to get the lineage objects corresponding to the
 datasets
dataset_artifacts = []
for vertex in query_result.vertices:
    dataset_artifacts.append(vertex.to_lineage_object().source.source_uri)

pp.pprint(dataset_artifacts)
```

Example Temukan model yang terkait dengan titik akhir

```
# Define the LineageFilter to look for entities of type `ARTIFACT` and the source of
 type `MODEL`.

query_filter = LineageFilter(
    entities=[LineageEntityEnum.ARTIFACT], sources=[LineageSourceEnum.MODEL]
)

# Providing this `LineageFilter` to the `LineageQuery` constructs a query that
 traverses through the given context `endpoint_context`
# and find all datasets.

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[endpoint_context.context_arn],
```

```
    query_filter=query_filter,
    direction=LineageQueryDirectionEnum.ASCENDANTS,
    include_edges=False,
)

# Parse through the query results to get the lineage objects corresponding to the model
model_artifacts = []
for vertex in query_result.vertices:
    model_artifacts.append(vertex.to_lineage_object().source.source_uri)

# The results of the `LineageQuery` API call return the ARN of the model deployed to
 the endpoint along with
# the S3 URI to the model.tar.gz file associated with the model
pp.pprint(model_artifacts)
```

Example Temukan komponen uji coba yang terkait dengan titik akhir

```
# Define the LineageFilter to look for entities of type `TRIAL_COMPONENT` and the
 source of type `TRAINING_JOB`.

query_filter = LineageFilter(
    entities=[LineageEntityEnum.TRIAL_COMPONENT],
    sources=[LineageSourceEnum.TRAINING_JOB],
)

# Providing this `LineageFilter` to the `LineageQuery` constructs a query that
 traverses through the given context `endpoint_context`
# and find all datasets.

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[endpoint_context.context_arn],
    query_filter=query_filter,
    direction=LineageQueryDirectionEnum.ASCENDANTS,
    include_edges=False,
)

# Parse through the query results to get the ARNs of the training jobs associated with
 this Endpoint
trial_components = []
for vertex in query_result.vertices:
    trial_components.append(vertex.arn)

pp.pprint(trial_components)
```

Example Mengubah titik fokus garis keturunan

`LineageQuery`Dapat dimodifikasi agar berbeda `start_arns` yang mengubah titik fokus garis keturunan. Selain itu, `LineageFilter` dapat mengambil beberapa sumber dan entitas untuk memperluas cakupan kueri.

Berikut ini kami menggunakan model sebagai titik fokus garis keturunan dan menemukan titik akhir dan kumpulan data yang terkait dengannya.

```
# Get the ModelArtifact

model_artifact_summary = list(Artifact.list(source_uri=model_package_arn))[0]
model_artifact = ModelArtifact.load(artifact_arn=model_artifact_summary.artifact_arn)
query_filter = LineageFilter(
    entities=[LineageEntityEnum.ARTIFACT],
    sources=[LineageSourceEnum.ENDPOINT, LineageSourceEnum.DATASET],
)

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[model_artifact.artifact_arn],  # Model is the starting artifact
    query_filter=query_filter,
    # Find all the entities that descend from the model, i.e. the endpoint
    direction=LineageQueryDirectionEnum.DESCENDANTS,
    include_edges=False,
)

associations = []
for vertex in query_result.vertices:
    associations.append(vertex.to_lineage_object().source.source_uri)

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[model_artifact.artifact_arn],  # Model is the starting artifact
    query_filter=query_filter,
    # Find all the entities that ascend from the model, i.e. the datasets
    direction=LineageQueryDirectionEnum.ASCENDANTS,
    include_edges=False,
)

for vertex in query_result.vertices:
    associations.append(vertex.to_lineage_object().source.source_uri)

pp.pprint(associations)
```

Example Menggunakan **`LineageQueryDirectionEnum.BOTH`** untuk menemukan hubungan naik dan turun

Ketika arah diatur keBOTH, kueri melintasi grafik untuk menemukan hubungan ascendant dan turunan. Traversal ini terjadi tidak hanya dari node awal, tetapi dari setiap node yang dikunjungi. Misalnya; jika pekerjaan pelatihan dijalankan dua kali dan kedua model yang dihasilkan oleh pekerjaan pelatihan diterapkan ke titik akhir, hasil kueri dengan arah diatur untuk BOTH menunjukkan kedua titik akhir. Ini karena gambar yang sama digunakan untuk pelatihan dan penerapan model. Karena gambar umum untuk model, `start_arn` dan kedua titik akhir, muncul di hasil kueri.

```
query_filter = LineageFilter(
    entities=[LineageEntityEnum.ARTIFACT],
    sources=[LineageSourceEnum.ENDPOINT, LineageSourceEnum.DATASET],
)

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[model_artifact.artifact_arn],  # Model is the starting artifact
    query_filter=query_filter,
    # This specifies that the query should look for associations both ascending and
 descending for the start
    direction=LineageQueryDirectionEnum.BOTH,
    include_edges=False,
)

associations = []
for vertex in query_result.vertices:
    associations.append(vertex.to_lineage_object().source.source_uri)

pp.pprint(associations)
```

Example Petunjuk dalam **`LineageQuery`** - **`ASCENDANTS`** vs. **`DESCENDANTS`**

Untuk memahami arah dalam Grafik Lineage, ambil grafik hubungan entitas berikut - Dataset -> Training Job -> Model -> Endpoint

Titik akhir adalah keturunan dari model, dan model adalah keturunan dari dataset. Demikian pula, model adalah ascendant dari endpoint. `direction`Parameter dapat digunakan untuk menentukan apakah kueri harus mengembalikan entitas yang merupakan turunan atau ascendants dari entitas di. `start_arns` Jika `start_arns` berisi model dan arahnya`DESCENDANTS`, kueri mengembalikan titik akhir. Jika arahnya`ASCENDANTS`, kueri mengembalikan dataset.

```
# In this example, we'll look at the impact of specifying the direction as ASCENDANT or
 DESCENDANT in a `LineageQuery`.

query_filter = LineageFilter(
    entities=[LineageEntityEnum.ARTIFACT],
    sources=[
        LineageSourceEnum.ENDPOINT,
        LineageSourceEnum.MODEL,
        LineageSourceEnum.DATASET,
        LineageSourceEnum.TRAINING_JOB,
    ],
)

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[model_artifact.artifact_arn],
    query_filter=query_filter,
    direction=LineageQueryDirectionEnum.ASCENDANTS,
    include_edges=False,
)

ascendant_artifacts = []

# The lineage entity returned for the Training Job is a TrialComponent which can't be
 converted to a
# lineage object using the method `to_lineage_object()` so we extract the
 TrialComponent ARN.
for vertex in query_result.vertices:
    try:
        ascendant_artifacts.append(vertex.to_lineage_object().source.source_uri)
    except:
        ascendant_artifacts.append(vertex.arn)

print("Ascendant artifacts : ")
pp.pprint(ascendant_artifacts)

query_result = LineageQuery(sagemaker_session).query(
    start_arns=[model_artifact.artifact_arn],
    query_filter=query_filter,
    direction=LineageQueryDirectionEnum.DESCENDANTS,
    include_edges=False,
)

descendant_artifacts = []
```

```
for vertex in query_result.vertices:
    try:
        descendant_artifacts.append(vertex.to_lineage_object().source.source_uri)
    except:
        # Handling TrialComponents.
        descendant_artifacts.append(vertex.arn)

print("Descendant artifacts : ")
pp.pprint(descendant_artifacts)
```

Example Fungsi pembantu SDK untuk membuat kueri silsilah lebih mudah

KelasEndpointContext,ModelArtifact, dan DatasetArtifact memiliki fungsi pembantu yang merupakan pembungkus LineageQuery API untuk membuat kueri garis keturunan tertentu lebih mudah dimanfaatkan. Contoh berikut menunjukkan cara menggunakan fungsi pembantu ini.

```
# Find all the datasets associated with this endpoint

datasets = []
dataset_artifacts = endpoint_context.dataset_artifacts()
for dataset in dataset_artifacts:
    datasets.append(dataset.source.source_uri)
print("Datasets : ", datasets)

# Find the training jobs associated with the endpoint
training_job_artifacts = endpoint_context.training_job_arns()
training_jobs = []
for training_job in training_job_artifacts:
    training_jobs.append(training_job)
print("Training Jobs : ", training_jobs)

# Get the ARN for the pipeline execution associated with this endpoint (if any)
pipeline_executions = endpoint_context.pipeline_execution_arn()
if pipeline_executions:
    for pipeline in pipelines_executions:
        print(pipeline)

# Here we use the `ModelArtifact` class to find all the datasets and endpoints
 associated with the model

dataset_artifacts = model_artifact.dataset_artifacts()
endpoint_contexts = model_artifact.endpoint_contexts()
```

```
datasets = [dataset.source.source_uri for dataset in dataset_artifacts]
endpoints = [endpoint.source.source_uri for endpoint in endpoint_contexts]

print("Datasets associated with this model : ")
pp.pprint(datasets)

print("Endpoints associated with this model : ")
pp.pprint(endpoints)

# Here we use the `DatasetArtifact` class to find all the endpoints hosting models that
 were trained with a particular dataset
# Find the artifact associated with the dataset

dataset_artifact_arn = list(Artifact.list(source_uri=training_data))[0].artifact_arn
dataset_artifact = DatasetArtifact.load(artifact_arn=dataset_artifact_arn)

# Find the endpoints that used this training dataset
endpoint_contexts = dataset_artifact.endpoint_contexts()
endpoints = [endpoint.source.source_uri for endpoint in endpoint_contexts]

print("Endpoints associated with the training dataset {}".format(training_data))
pp.pprint(endpoints)
```

Example Mendapatkan visualisasi grafik Lineage

Kelas pembantu `Visualizer` disediakan di sameple notebook visualizer.py untuk membantu memplot grafik garis keturunan. Saat respons kueri dirender, grafik dengan hubungan garis keturunan dari ditampilkan. `StartArns` Dari visualisasi menunjukkan hubungan dengan entitas garis keturunan lain yang ditampilkan dalam tindakan API. `StartArns` `query_lineage`

```
# Graph APIs
# Here we use the boto3 `query_lineage` API to generate the query response to plot.

from visualizer import Visualizer

query_response = sm_client.query_lineage(
    StartArns=[endpoint_context.context_arn], Direction="Ascendants", IncludeEdges=True
)

viz = Visualizer()
viz.render(query_response, "Endpoint")

        query_response = sm_client.query_lineage(
```

```
    StartArns=[model_artifact.artifact_arn], Direction="Ascendants", IncludeEdges=True
)
viz.render(query_response, "Model")
```

Pelacakan Lintas Akun

Amazon SageMaker mendukung melacak entitas garis keturunan dari akun yang berbedaAWS. AWSAkun lain dapat berbagi entitas silsilah mereka dengan Anda dan Anda dapat mengakses entitas silsilah ini melalui panggilan API langsung atau kueri garis keturunan. SageMaker

SageMaker digunakan AWS Resource Access Manageruntuk membantu Anda berbagi sumber daya garis keturunan Anda dengan aman. Anda dapat membagikan sumber daya Anda melalui AWS RAMkonsol.

Mengatur Lintas Akun Tracking

Anda dapat mengelompokkan dan berbagi Entitas Pelacakan Silsilah melalui grup garis keturunan di Amazon. SageMaker SageMaker hanya mendukung satu grup garis keturunan default per akun. SageMaker membuat grup silsilah default setiap kali entitas silsilah dibuat di akun Anda. Setiap entitas silsilah yang dimiliki oleh akun Anda ditetapkan ke grup silsilah default ini. Untuk berbagi entitas silsilah dengan akun lain, Anda membagikan grup silsilah default ini dengan akun tersebut.

> **Note**
>
> Anda dapat membagikan semua entitas pelacakan garis keturunan dalam grup silsilah atau tidak sama sekali.

Buat pembagian sumber daya untuk entitas silsilah Anda menggunakan AWS Resource Access Manager konsol. Untuk informasi selengkapnya, lihat Berbagi AWS sumber daya Anda di Panduan AWS Resource Access Manager Pengguna.

> **Note**
>
> Setelah pembagian sumber daya dibuat, mungkin diperlukan waktu beberapa menit hingga asosiasi sumber daya dan prinsipal selesai. Setelah asosiasi ditetapkan, akun bersama menerima undangan untuk bergabung dengan pembagian sumber daya. Akun bersama harus menerima undangan untuk mendapatkan akses ke sumber daya bersama. Untuk

informasi selengkapnya tentang menerima undangan berbagi sumber dayaAWS RAM, lihat Menggunakan AWS sumber daya bersama di Panduan Pengguna AWS Resource Access Manager.

Kebijakan sumber daya pelacakan silsilah lintas akun Anda

Amazon hanya SageMaker mendukung satu jenis kebijakan sumber daya. Kebijakan SageMaker sumber daya harus mengizinkan semua operasi berikut:

```
"sagemaker:DescribeAction"
"sagemaker:DescribeArtifact"
"sagemaker:DescribeContext"
"sagemaker:DescribeTrialComponent"
"sagemaker:AddAssociation"
"sagemaker:DeleteAssociation"
"sagemaker:QueryLineage"
```

Example Berikut ini adalah kebijakan SageMaker sumber daya yang dibuat menggunakan AWS Resource Access Manager untuk membuat pembagian sumber daya untuk grup silsilah akun.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FullLineageAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012" #account-id
      },
      "Action": [
        "sagemaker:DescribeAction",
        "sagemaker:DescribeArtifact",
        "sagemaker:DescribeContext",
        "sagemaker:DescribeTrialComponent",
        "sagemaker:AddAssociation",
        "sagemaker:DeleteAssociation",
        "sagemaker:QueryLineage"
      ],
      "Resource": "arn:aws:sagemaker:us-west-2:111111111111:lineage-group/sagemaker-
default-lineage-group" #Sample lineage group resource
```

```
    }
  ]
}
```

Melacak Entitas Garis Akun

Dengan pelacakan silsilah lintas akun, Anda dapat mengaitkan entitas silsilah di akun yang berbeda menggunakan tindakan API yang sama. `AddAssociation` Saat Anda mengaitkan dua entitas garis keturunan, SageMaker validasi jika Anda memiliki izin untuk melakukan tindakan `AddAssociation` API pada kedua entitas garis keturunan. SageMaker kemudian mendirikan asosiasi. Jika Anda tidak memiliki izin, SageMaker tidak membuat asosiasi. Setelah asosiasi lintas akun dibuat, Anda dapat mengakses entitas silsilah dari yang lain melalui tindakan API. `QueryLineage` Untuk informasi selengkapnya, lihat Meminta Entitas Silsilah.

Selain SageMaker secara otomatis membuat entitas garis keturunan, jika Anda memiliki akses lintas akun, SageMaker menghubungkan artefak yang mereferensikan objek atau data yang sama. Jika data dari satu akun digunakan dalam pelacakan garis keturunan oleh akun yang berbeda, SageMaker buat artefak di setiap akun untuk melacak data tersebut. Dengan silsilah lintas akun, setiap kali SageMaker membuat artefak baru, SageMaker periksa apakah ada artefak lain yang dibuat untuk data yang sama yang juga dibagikan dengan Anda. SageMaker kemudian membangun asosiasi antara artefak yang baru dibuat dan masing-masing artefak yang dibagikan dengan Anda dengan set ke`AssociationType`. `SameAs` Anda kemudian dapat menggunakan tindakan `QueryLineage` API untuk melintasi entitas silsilah di akun Anda sendiri ke entitas silsilah yang dibagikan dengan Anda tetapi dimiliki oleh akun lain. AWS Lihat informasi yang lebih lengkap di Meminta Entitas Silsilah

Topik

- Mengakses sumber silsilah dari akun yang berbeda
- Otorisasi untuk menanyakan entitas silsilah lintas akun

Mengakses sumber silsilah dari akun yang berbeda

Setelah akses lintas-akun untuk berbagi silsilah telah disiapkan, Anda dapat memanggil tindakan SageMaker API berikut secara langsung dengan ARN untuk menjelaskan entitas silsilah bersama dari akun lain:

- `DescribeAction`
- `DescribeArtifact`

- `DescribeContext`
- `DescribeTrialComponent`

Anda juga dapat mengelola Asosiasi untuk entitas silsilah yang dimiliki oleh akun berbeda yang dibagikan dengan Anda, menggunakan tindakan SageMaker API berikut:

- `AddAssociation`
- `DeleteAssociation`

Untuk buku catatan yang menunjukkan cara menggunakan API SageMaker Lineage untuk menanyakan garis keturunan di seluruh akun., lihat -with-ram.ipynb. sagemaker-lineage-cross-account

Otorisasi untuk menanyakan entitas silsilah lintas akun

Amazon SageMaker harus memvalidasi bahwa Anda memiliki izin untuk melakukan tindakan `QueryLineage` API pada. `StartArns` Ini diberlakukan melalui kebijakan sumber daya yang dilampirkan pada. `LineageGroup` Hasil dari tindakan ini mencakup semua entitas garis keturunan yang dapat Anda akses, apakah mereka dimiliki oleh akun Anda atau dibagikan oleh akun lain. Untuk informasi selengkapnya, lihat Meminta Entitas Silsilah.

Daftarkan dan Terapkan Model dengan Registri Model

Dengan Registry SageMaker Model Anda dapat melakukan hal berikut ini:

- Model katalog untuk produksi.
- Mengelola versi model.
- Kaitkan metadata, seperti metrik pelatihan, dengan model.
- Mengelola status persetujuan model.
- Menyebarkan model ke produksi.
- Otomatiskan penerapan model dengan CI/CD.

Katalog model dengan membuat SageMaker Model Registry Model (Package) Grup yang berisi versi model yang berbeda. Anda dapat membuat Grup Model yang melacak semua model yang Anda latih untuk memecahkan masalah tertentu. Anda kemudian dapat mendaftarkan setiap model yang Anda

latih dan Model Registry menambahkannya ke Grup Model sebagai versi model baru. Terakhir, Anda dapat membuat kategori Grup Model dengan mengaturnya lebih lanjut ke dalam Koleksi Registri SageMaker Model. Alur kerja tipikal mungkin terlihat seperti berikut ini:

- Buat Grup Model.
- Buat pipeline ML yang melatih model. Untuk informasi tentang SageMaker jaringan pipa, lihatMembuat dan Mengelola SageMaker Pipelines.
- Untuk setiap proses pipeline ML, buat versi model yang Anda daftarkan di Grup Model yang Anda buat pada langkah pertama.
- Tambahkan Grup Model Anda ke dalam satu atau beberapa Koleksi Registri Model.

Untuk detail tentang cara membuat dan bekerja dengan model, versi model, dan Grup Model, lihatModel Registry Model, Versi Model, dan Grup Model. Secara opsional, jika Anda ingin mengelompokkan Grup Model lebih lanjut ke dalam Koleksi, lihatKoleksi Registri Model.

Model Registry Model, Versi Model, dan Grup Model

Registri SageMaker Model disusun sebagai beberapa Grup Model (Package) dengan paket model di setiap grup. Grup Model ini secara opsional dapat ditambahkan ke satu atau beberapa Koleksi. Setiap paket model dalam Grup Model sesuai dengan model yang terlatih. Versi dari setiap paket model adalah nilai numerik yang dimulai pada 1 dan ditambah dengan setiap paket model baru ditambahkan ke Grup Model. Misalnya, jika 5 paket model ditambahkan ke Grup Model, versi paket model akan menjadi 1, 2, 3, 4, dan 5.

Ada dua jenis paket model di SageMaker. Satu jenis digunakan di AWS Marketplace, dan yang lainnya digunakan dalam Model Registry. Paket model yang digunakan di AWS Marketplace bukan entitas berversi dan tidak terkait dengan Grup Model di Registri Model. Untuk informasi selengkapnya tentang paket model yang digunakan di AWS Marketplace, lihatJual algoritma dan paket di AWS Marketplace.

Paket model yang digunakan dalam Registri Model berversi, dan harus dikaitkan dengan Grup Model. ARN dari jenis paket model ini memiliki struktur: `'arn:aws:sagemaker:region:account:model-package-group/version'`

Topik berikut menunjukkan cara membuat dan bekerja dengan model, versi model, dan Grup Model di Registri Model.

Topik

- Membuat Grup Model
- Menghapus Grup Model
- Daftarkan Versi Model
- Lihat Grup dan Versi Model
- Lihat Detail Versi Model
- Bandingkan Versi Model
- Lihat dan Kelola Grup Model dan Tag Versi Model
- Bagikan Model dengan Pengguna SageMaker Canvas
- Menghapus Versi Model
- Memperbarui Status Persetujuan Model
- Menyebarkan Model dari Registry
- Lihat Riwayat Penerapan Model

Membuat Grup Model

Grup Model berisi sekelompok model berversi. Buat Grup Model dengan menggunakan konsol Amazon SageMaker Studio AWS SDK for Python (Boto3) atau Amazon.

Buat Grup Model (Boto3)

Untuk membuat Grup Model menggunakan Boto3, panggil operasi `create_model_package_group` API dan tentukan nama dan deskripsi sebagai parameter. Contoh berikut menunjukkan cara membuat Grup Model. Respons dari `create_model_package_group` panggilan tersebut adalah Amazon Resource Name (ARN) dari Grup Model baru.

Pertama, impor paket yang diperlukan dan atur klien SageMaker Boto3.

```
import time
import os
from sagemaker import get_execution_role, session
import boto3

region = boto3.Session().region_name

role = get_execution_role()

sm_client = boto3.client('sagemaker', region_name=region)
```

Sekarang buat Grup Model.

```
import time
model_package_group_name = "scikit-iris-detector-" + str(round(time.time()))
model_package_group_input_dict = {
 "ModelPackageGroupName" : model_package_group_name,
 "ModelPackageGroupDescription" : "Sample model package group"
}

create_model_package_group_response =
 sm_client.create_model_package_group(**model_package_group_input_dict)
print('ModelPackageGroup Arn :
 {}'.format(create_model_package_group_response['ModelPackageGroupArn']))
```

Membuat Grup Model (konsol)

Untuk membuat Grup Model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Pilih Daftar, lalu pilih Grup model.
6. Di kotak dialog Grup model daftar, masukkan informasi berikut:
   - Nama Grup Model baru di bidang nama grup Model.
   - (Opsional) Deskripsi untuk Grup Model di bidang Deskripsi.
   - (Opsional) Setiap pasangan kunci-nilai yang ingin Anda kaitkan dengan Grup Model di bidang Tag. Untuk informasi tentang penggunaan tag, lihat Menandai AWS sumber daya di. Referensi Umum AWS
7. Pilih Daftar grup model.
8. (Opsional) Di halaman Model, pilih tab Model terdaftar, lalu pilih Grup Model. Konfirmasikan Grup Model yang baru Anda buat muncul di daftar Grup Model.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Pilih Tindakan, lalu pilih Buat grup model.
5. Di Buat grup model, masukkan informasi berikut:
   - Masukkan nama Grup Model baru di bidang Nama grup Model.
   - (Opsional) Masukkan deskripsi untuk Grup Model di kolom Deskripsi.
   - (Opsional) Masukkan pasangan kunci-nilai yang ingin Anda kaitkan dengan Grup Model di bidang Tag. Untuk informasi tentang penggunaan tag, lihat Menandai AWS sumber daya di. Referensi Umum AWS
   - (Opsional) Pilih proyek yang dapat digunakan untuk mengaitkan Grup Model di bidang Proyek. Untuk informasi tentang proyek, lihatOtomatiskan MLOP dengan Proyek SageMaker .
6. Pilih Buat grup model.

Menghapus Grup Model

Prosedur ini menunjukkan cara menghapus Grup Model di konsol Amazon SageMaker Studio.

Menghapus Grup Model (konsol)

> **Important**
>
> Anda hanya dapat menghapus grup model kosong. Sebelum Anda menghapus grup model Anda, hapus versi modelnya, jika ada.

Untuk menghapus Grup Model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari daftar grup model, pilih kotak centang di samping nama Grup Model yang ingin Anda hapus.
6. Pilih elipsis vertikal di atas sudut kanan atas daftar grup model, dan pilih Hapus.
7. Dalam kotak dialog Hapus grup model, pilih Ya, hapus grup model.
8. Pilih Hapus.
9. Konfirmasikan bahwa grup model yang dihapus tidak lagi muncul dalam daftar grup model Anda.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model. Daftar Grup Model Anda akan muncul.
4. Dari grup model, pilih nama Grup Model yang ingin Anda hapus.
5. Di pojok kanan atas, pilih Hapus.
6. Di kotak dialog konfirmasi, masukkan`REMOVE`.
7. Pilih Hapus.

Daftarkan Versi Model

Anda dapat mendaftarkan SageMaker model Amazon dengan membuat versi model yang menentukan grup model yang menjadi miliknya. Versi model harus mencakup artefak model (bobot terlatih model) dan kode inferensi untuk model tersebut.

Pipa inferensi adalah SageMaker model yang terdiri dari urutan linier dari dua hingga lima belas kontainer yang memproses permintaan inferensi. Anda mendaftarkan pipeline inferensi dengan menentukan kontainer dan variabel lingkungan terkait. Untuk informasi selengkapnya tentang saluran inferensi, lihat. Model host bersama dengan logika pra-pemrosesan sebagai pipa inferensi serial di belakang satu titik akhir

Anda dapat mendaftarkan model dengan pipeline inferensi, dengan menentukan wadah dan variabel lingkungan terkait. Untuk membuat versi model dengan pipeline inferensi dengan menggunakan konsol Amazon SageMaker Studio, atau dengan membuat langkah dalam pipeline pembuatan SageMaker model, gunakan langkah-langkah berikut. AWS SDK for Python (Boto3)

Topik

- Daftarkan Versi Model (SageMakerPipelines)
- Daftarkan Versi Model (Boto3)
- Mendaftarkan Versi Model (konsol)
- Daftarkan Versi Model dari Akun yang Berbeda

Daftarkan Versi Model (SageMakerPipelines)

Untuk mendaftarkan versi model dengan menggunakan pipa bangunan SageMaker model, buat `RegisterModel` langkah dalam pipeline Anda. Untuk informasi tentang membuat `RegisterModel` langkah sebagai bagian dari pipeline, lihatLangkah 8: Tentukan RegisterModel Langkah untuk Membuat Model Package.

Daftarkan Versi Model (Boto3)

Untuk mendaftarkan versi model dengan menggunakan Boto3, hubungi operasi `create_model_package` API.

Pertama, Anda mengatur kamus parameter untuk diteruskan ke operasi `create_model_package` API.

```
# Specify the model source
model_url = "s3://your-bucket-name/model.tar.gz"

modelpackage_inference_specification =  {
    "InferenceSpecification": {
      "Containers": [
         {
```

```
            "Image": '257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-
xgboost:1.2-1',
     "ModelDataUrl": model_url
         }
      ],
      "SupportedContentTypes": [ "text/csv" ],
      "SupportedResponseMIMETypes": [ "text/csv" ],
   }
 }

# Alternatively, you can specify the model source like this:
# modelpackage_inference_specification["InferenceSpecification"]["Containers"][0]
["ModelDataUrl"]=model_url

create_model_package_input_dict = {
    "ModelPackageGroupName" : model_package_group_name,
    "ModelPackageDescription" : "Model to detect 3 different types of irises (Setosa,
 Versicolour, and Virginica)",
    "ModelApprovalStatus" : "PendingManualApproval"
}
create_model_package_input_dict.update(modelpackage_inference_specification)
```

Kemudian Anda memanggil operasi `create_model_package` API, meneruskan kamus parameter yang baru saja Anda atur.

```
create_model_package_response =
 sm_client.create_model_package(**create_model_package_input_dict)
model_package_arn = create_model_package_response["ModelPackageArn"]
print('ModelPackage Version ARN : {}'.format(model_package_arn))
```

Mendaftarkan Versi Model (konsol)

Untuk mendaftarkan versi model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model dari menu.
3. Pilih tab Model terdaftar, jika belum dipilih.

4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Pilih Daftar, lalu pilih Versi model.
6. Dalam formulir versi model Register, masukkan informasi berikut:
   - Di dropdown nama grup Model, pilih nama grup model yang menjadi milik versi Anda.
   - (Opsional) Masukkan deskripsi untuk versi model Anda.
   - Di menu tarik-turun Status Persetujuan Model, pilih status persetujuan versi.
   - (Opsional) Di bidang Metadata khusus, pilih + Tambahkan baru dan tambahkan tag khusus sebagai pasangan nilai kunci.
7. Pilih Berikutnya.
8. Dalam formulir Spesifikasi Inferensi, masukkan informasi berikut:
   - Di Inference image location (ECR), masukkan lokasi gambar inferensi ECR Anda.
   - Di lokasi artefak Model (S3), masukkan lokasi bucket Amazon S3 dari artefak data model Anda.
   - Untuk menentukan dan memasukkan konfigurasi data atau variabel lingkungan, pilih Konfigurasi tambahan dan masukkan informasi ini.
   - Untuk menambahkan lebih banyak wadah, pilih + Tambahkan wadah.
   - Dalam tipe instance inferensi Realtime, masukkan tipe instance yang akan digunakan untuk inferensi waktu nyata.
   - Dalam jenis instance inferensi Transform, masukkan tipe instance yang akan digunakan untuk transformasi batch.
   - Di Jenis konten yang didukung, masukkan jenis MIME masukan Anda.
   - Di jenis konten respons yang didukung, masukkan jenis MIME keluaran Anda.
9. Pilih Berikutnya.
10. Dalam formulir Rekomendasi Inferensi opsional, masukkan informasi berikut:
    - Untuk masalah Bisnis, pilih aplikasi yang berlaku untuk model Anda.
    - Untuk Tugas, pilih jenis masalah yang berlaku untuk model Anda.
    - Untuk alamat bucket S3, masukkan lokasi bucket Amazon S3 dari muatan sampel Anda.
    - Untuk kontainer pertama, masukkan informasi berikut:
      - Untuk nama Model, masukkan nama model seperti yang digunakan di kebun binatang model.

- Untuk Framework, pilih kerangka kerja.
- Untuk versi Framework, masukkan versi kerangka kerja.
- Ulangi langkah sebelumnya untuk semua wadah.

11. Pilih Berikutnya.
12. Pilih kotak centang di samping satu atau beberapa metrik model yang ditampilkan.
13. Pilih Berikutnya.
14. Pastikan pengaturan yang ditampilkan sudah benar, dan pilih Daftar versi model. Jika Anda kemudian melihat jendela modal dengan pesan kesalahan, pilih Lihat (di sebelah pesan) untuk melihat sumber kesalahan.
15. Konfirmasikan versi model baru Anda muncul di halaman grup model induk.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Buka formulir Register Version. Anda dapat melakukannya dengan salah satu dari dua cara berikut:

   - Pilih Tindakan, lalu pilih Buat versi model.
   - Pilih nama grup model yang ingin Anda buat versi model, lalu pilih Buat versi model.

5. Dalam formulir versi model Register, masukkan informasi berikut:

   - Dalam dropdown nama grup paket Model, pilih nama grup model.
   - (Opsional) Masukkan deskripsi untuk versi model Anda.
   - Di menu tarik-turun Status Persetujuan Model, pilih status persetujuan versi.
   - (Opsional) Di bidang metadata kustom, tambahkan tag kustom sebagai pasangan nilai kunci.

6. Pilih Berikutnya.
7. Dalam formulir Spesifikasi Inferensi, masukkan informasi berikut:

   - Masukkan lokasi gambar inferensi Anda.

- Masukkan lokasi artefak data model Anda.
- (Opsional) Masukkan informasi tentang gambar yang akan digunakan untuk pekerjaan inferensi transformasi dan real-time, serta jenis MIME input dan output yang didukung.

8. Pilih Berikutnya.
9. (Opsional) Berikan detail untuk membantu rekomendasi titik akhir.
10. Pilih Berikutnya.
11. (Opsional) Pilih metrik model yang ingin Anda sertakan.
12. Pilih Berikutnya.
13. Pastikan pengaturan yang ditampilkan sudah benar, dan pilih Daftar versi model. Jika Anda kemudian melihat jendela modal dengan pesan kesalahan, pilih Lihat (di sebelah pesan) untuk melihat sumber kesalahan.
14. Konfirmasikan versi model baru Anda muncul di halaman grup model induk.

Daftarkan Versi Model dari Akun yang Berbeda

Untuk mendaftarkan versi model dengan Grup Model yang dibuat oleh AWS akun lain, Anda harus menambahkan kebijakan AWS Identity and Access Management sumber daya lintas akun untuk mengaktifkan akun tersebut. Misalnya, satu AWS akun di organisasi Anda bertanggung jawab atas model pelatihan, dan akun lain bertanggung jawab untuk mengelola, menerapkan, dan memperbarui model. Anda membuat kebijakan sumber daya IAM dan menerapkan kebijakan ke sumber daya akun tertentu yang ingin Anda berikan akses untuk kasus ini. Untuk informasi selengkapnya tentang kebijakan sumber daya lintas akunAWS, lihat Logika evaluasi kebijakan lintas akun di AWS Identity and Access ManagementPanduan Pengguna.

> **Note**
>
> Anda juga harus menggunakan kunci KMS untuk mengenkripsi tindakan konfigurasi data keluaran selama pelatihan untuk penerapan model lintas akun.

Untuk mengaktifkan registri model lintas akun SageMaker, Anda harus menyediakan kebijakan sumber daya lintas akun untuk Grup Model yang berisi versi model. Berikut ini adalah contoh yang membuat kebijakan lintas akun untuk Grup Model dan menerapkan kebijakan ini ke sumber daya tertentu.

Konfigurasi berikut harus diatur dalam akun sumber yang mendaftarkan model lintas akun di Grup Model. Dalam contoh ini, akun sumber adalah akun pelatihan model yang akan melatih dan kemudian mendaftarkan akun silang model ke dalam Model Registry dari akun Model Registry.

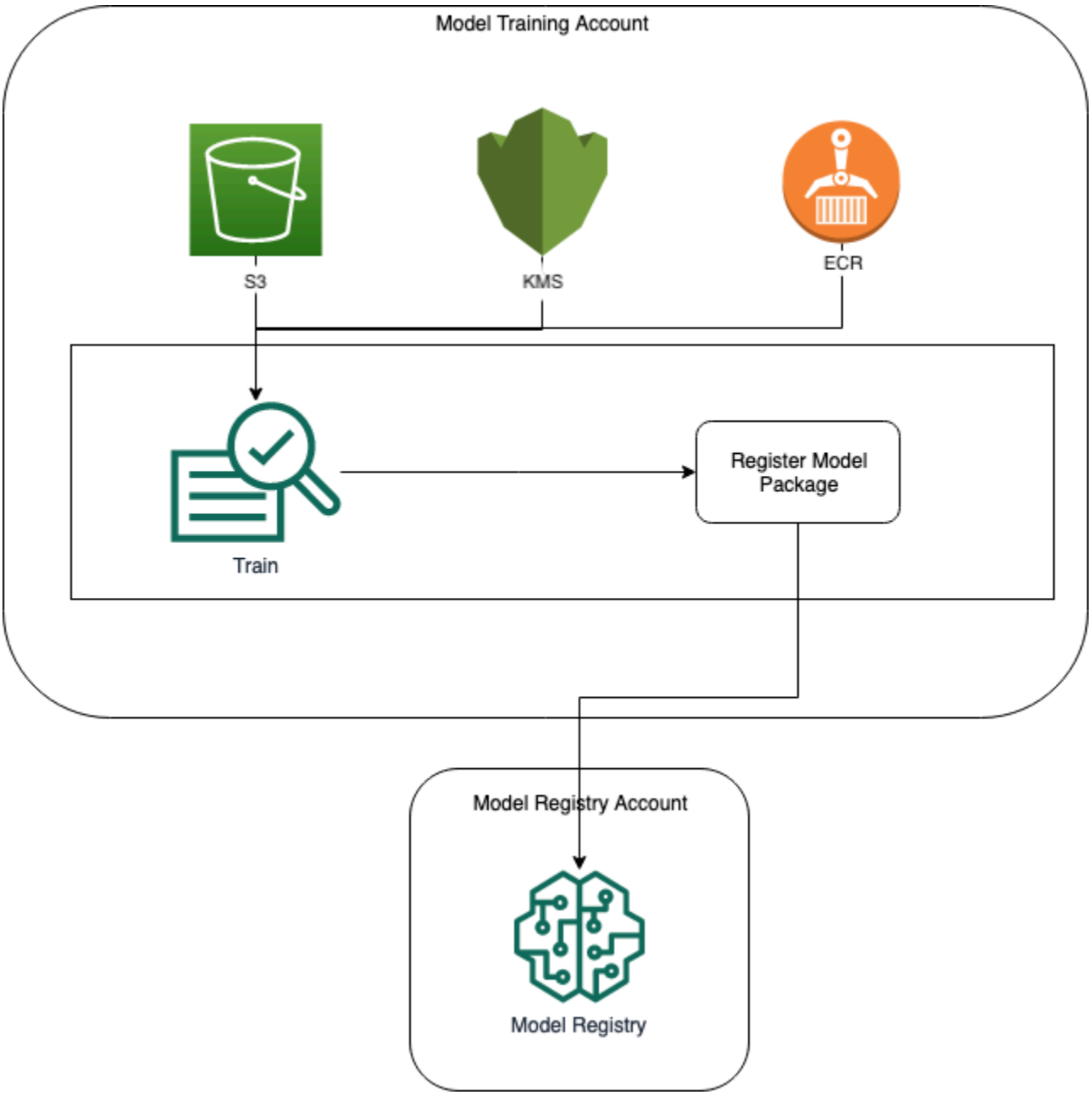
Contoh ini mengasumsikan bahwa Anda sebelumnya menetapkan variabel berikut:

- `sm_client`- Klien SageMaker Boto3.
- `model_package_group_name`— Grup Model yang ingin Anda berikan akses.
- `model_package_group_arn`— Grup Model ARN tempat Anda ingin memberikan akses akun.
- `bucket`- Bucket Amazon S3 tempat artefak pelatihan model disimpan.

Untuk dapat menerapkan model yang dibuat di akun yang berbeda, pengguna harus memiliki peran yang memiliki akses ke SageMaker tindakan, seperti peran dengan kebijakan `AmazonSageMakerFullAccess` terkelola. Untuk informasi tentang kebijakan SageMaker terkelola, lihatAWSKebijakan Terkelola untuk Amazon SageMaker.

Kebijakan sumber daya IAM yang diperlukan

Diagram berikut menangkap kebijakan yang diperlukan untuk memungkinkan pendaftaran model lintas akun. Seperti yang ditunjukkan, kebijakan ini harus aktif selama pelatihan model untuk mendaftarkan model dengan benar ke dalam akun Model Registry.



Amazon ECR, Amazon S3, AWS KMS dan kebijakan ditunjukkan dalam contoh kode berikut.

Contoh kebijakan Amazon ECR

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AddPerm",
      "Effect": "Allow",
```

```
      "Principal": {
        "AWS": "arn:aws:iam::{model_registry_account}:root"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:Describe*"
      ]
    }
  ]
}
```

Contoh kebijakan Amazon S3

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AddPerm",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{model_registry_account}:root"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetBucketAcl",
        "s3:GetObjectAcl"
      ],
      "Resource": "arn:aws:s3:::{bucket}/*"
    }
  ]
}
```

AWS KMSKebijakan sampel

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AddPerm",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{model_registry_account}:root"
      },
```

```
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
      ],
      "Resource": "*"
    }
  ]
}
```

Menerapkan kebijakan sumber daya ke akun

Konfigurasi kebijakan berikut menerapkan kebijakan yang dibahas di bagian sebelumnya dan harus dimasukkan ke dalam akun pelatihan model.

```
import json

# The Model Registry account id of the Model Group
model_registry_account = "111111111111"

# The model training account id where training happens
model_training_account = "222222222222"

# 1. Create a policy for access to the ECR repository
# in the model training account for the Model Registry account Model Group
ecr_repository_policy = {"Version": "2012-10-17",
    "Statement": [{"Sid": "AddPerm",
        "Effect": "Allow",
        "Principal": {
          "AWS": f"arn:aws:iam::{model_registry_account}:root"
        },
        "Action": [
          "ecr:BatchGetImage",
          "ecr:Describe*"
        ]
    }]
}

# Convert the ECR policy from JSON dict to string
ecr_repository_policy = json.dumps(ecr_repository_policy)

# Set the new ECR policy
ecr = boto3.client('ecr')
response = ecr.set_repository_policy(
```

```
    registryId = model_training_account,
    repositoryName = "decision-trees-sample",
    policyText = ecr_repository_policy
)

# 2. Create a policy in the model training account for access to the S3 bucket
# where the model is present in the Model Registry account Model Group
bucket_policy = {"Version": "2012-10-17",
    "Statement": [{"Sid": "AddPerm",
        "Effect": "Allow",
        "Principal": {"AWS": f"arn:aws:iam::{model_registry_account}:root"
        },
        "Action": [
          "s3:GetObject",
          "s3:GetBucketAcl",
          "s3:GetObjectAcl"
        ],
        "Resource": "arn:aws:s3:::{bucket}/*"
    }]
}

# Convert the S3 policy from JSON dict to string
bucket_policy = json.dumps(bucket_policy)

# Set the new bucket policy
s3 = boto3.client("s3")
response = s3.put_bucket_policy(
    Bucket = bucket,
    Policy = bucket_policy)

# 3. Create the KMS grant for the key used during training for encryption
# in the model training account to the Model Registry account Model Group
client = boto3.client("kms")

response = client.create_grant(
    GranteePrincipal=model_registry_account,
    KeyId=kms_key_id
    Operations=[
        "Decrypt",
        "GenerateDataKey",
    ],
)
```

Konfigurasi berikut perlu dimasukkan ke dalam akun Model Registry di mana Grup Model ada.

```
# The Model Registry account id of the Model Group
model_registry_account = "111111111111"

# 1. Create policy to allow the model training account to access the ModelPackageGroup
model_package_group_policy = {"Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AddPermModelPackageVersion",
            "Effect": "Allow",
            "Principal": {"AWS": f"arn:aws:iam::{model_training_account}:root"},
            "Action": ["sagemaker:CreateModelPackage"],
            "Resource": f"arn:aws:sagemaker:{region}:{model_registry_account}:model-
package/{model_package_group_name}/*"
        }
    ]
}

# Convert the policy from JSON dict to string
model_package_group_policy = json.dumps(model_package_group_policy)

# Set the new policy
response = sm_client.put_model_package_group_policy(
    ModelPackageGroupName = model_package_group_name,
    ResourcePolicy = model_package_group_policy)
```

Terakhir, gunakan `create_model_package` tindakan dari akun pelatihan model untuk mendaftarkan paket model di akun silang.

```
# Specify the model source
model_url = "s3://{bucket}/model.tar.gz"

#Set up the parameter dictionary to pass to the create_model_package API operation
modelpackage_inference_specification =  {
    "InferenceSpecification": {
        "Containers": [
            {
                "Image": f"{model_training_account}.dkr.ecr.us-east-2.amazonaws.com/
decision-trees-sample:latest",
```

```
                "ModelDataUrl": model_url
            }
        ],
        "SupportedContentTypes": [ "text/csv" ],
        "SupportedResponseMIMETypes": [ "text/csv" ],
    }
}

# Alternatively, you can specify the model source like this:
# modelpackage_inference_specification["InferenceSpecification"]["Containers"][0]
["ModelDataUrl"]=model_url

create_model_package_input_dict = {
    "ModelPackageGroupName" : model_package_group_arn,
    "ModelPackageDescription" : "Model to detect 3 different types of irises (Setosa,
 Versicolour, and Virginica)",
    "ModelApprovalStatus" : "PendingManualApproval"
}
create_model_package_input_dict.update(modelpackage_inference_specification)

# Create the model package in the Model Registry account
create_model_package_response =
 sm_client.create_model_package(**create_model_package_input_dict)
model_package_arn = create_model_package_response["ModelPackageArn"]
print('ModelPackage Version ARN : {}'.format(model_package_arn))
```

Lihat Grup dan Versi Model

Grup Model dan versi membantu Anda mengatur model Anda. Anda dapat melihat daftar versi model di Grup Model dengan menggunakan konsol AWS SDK for Python (Boto3) (Boto3) atau Amazon SageMaker Studio.

Melihat Daftar Versi Model dalam Grup

Anda dapat melihat semua versi model yang terkait dengan Grup Model. Jika Grup Model mewakili semua model yang Anda latih untuk mengatasi masalah ML tertentu, Anda dapat melihat semua model terkait tersebut.

Lihat Daftar Versi Model dalam Grup (Boto3)

Untuk melihat versi model yang terkait dengan Grup Model menggunakan Boto3, panggil operasi `list_model_packages` API, dan berikan nama Grup Model sebagai nilai parameter.

`ModelPackageGroupName` Kode berikut mencantumkan versi model yang terkait dengan Grup Model yang Anda buatBuat Grup Model (Boto3).

```
sm_client.list_model_packages(ModelPackageGroupName=model_package_group_name)
```

Melihat Daftar Versi Model dalam Grup (konsol)

Untuk melihat daftar versi model di Grup Model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model dari menu.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari daftar grup model, pilih braket sudut di sebelah kiri grup model yang ingin Anda lihat.
6. Daftar versi model dalam grup model muncul.
7. (Opsional) Pilih Lihat semua, jika ditampilkan, untuk melihat versi model tambahan.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Dari grup model, pilih nama Grup Model yang ingin Anda lihat.
5. Tab baru muncul dengan daftar versi model di Grup Model.

Lihat Detail Versi Model

Anda dapat melihat detail versi model tertentu dengan menggunakan konsol Amazon SageMaker Studio AWS SDK for Python (Boto3) atau Amazon.

Lihat Detail Versi Model (Boto3)

Untuk melihat detail versi model dengan menggunakan Boto3, selesaikan langkah-langkah berikut.

1. Panggil operasi `list_model_packages` API untuk melihat versi model dalam Grup Model.

```
sm_client.list_model_packages(ModelPackageGroupName="ModelGroup1")
```

   Responsnya adalah daftar ringkasan paket model. Anda bisa mendapatkan Amazon Resource Name (ARN) dari versi model dari daftar ini.

```
{'ModelPackageSummaryList': [{'ModelPackageGroupName':
 'AbaloneMPG-16039329888329896',
   'ModelPackageVersion': 1,
   'ModelPackageArn': 'arn:aws:sagemaker:us-east-2:123456789012:model-package/
ModelGroup1/1',
   'ModelPackageDescription': 'TestMe',
   'CreationTime': datetime.datetime(2020, 10, 29, 1, 27, 46, 46000,
 tzinfo=tzlocal()),
   'ModelPackageStatus': 'Completed',
   'ModelApprovalStatus': 'Approved'}],
 'ResponseMetadata': {'RequestId': '12345678-abcd-1234-abcd-aabbccddeeff',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '12345678-abcd-1234-abcd-aabbccddeeff',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '349',
   'date': 'Mon, 23 Nov 2020 04:56:50 GMT'},
  'RetryAttempts': 0}}
```

2. Hubungi `describe_model_package` untuk melihat detail versi model. Anda meneruskan ARN dari versi model yang Anda dapatkan di output panggilan ke. `list_model_packages`

```
sm_client.describe_model_package(ModelPackageName="arn:aws:sagemaker:us-
east-2:123456789012:model-package/ModelGroup1/1")
```

   Output dari panggilan ini adalah objek JSON dengan detail versi model.

```
{'ModelPackageGroupName': 'ModelGroup1',
 'ModelPackageVersion': 1,
 'ModelPackageArn': 'arn:aws:sagemaker:us-east-2:123456789012:model-package/
ModelGroup/1',
```

```
 'ModelPackageDescription': 'Test Model',
 'CreationTime': datetime.datetime(2020, 10, 29, 1, 27, 46, 46000,
 tzinfo=tzlocal()),
 'InferenceSpecification': {'Containers': [{'Image': '257758044811.dkr.ecr.us-
east-2.amazonaws.com/sagemaker-xgboost:1.0-1-cpu-py3',
    'ImageDigest':
 'sha256:99fa602cff19aee33297a5926f8497ca7bcd2a391b7d600300204eef803bca66',
    'ModelDataUrl': 's3://sagemaker-us-east-2-123456789012/ModelGroup1/
pipelines-0gdonccek7o9-AbaloneTrain-stmiylhtIR/output/model.tar.gz'}],
  'SupportedTransformInstanceTypes': ['ml.m5.xlarge'],
  'SupportedRealtimeInferenceInstanceTypes': ['ml.t2.medium', 'ml.m5.xlarge'],
  'SupportedContentTypes': ['text/csv'],
  'SupportedResponseMIMETypes': ['text/csv']},
 'ModelPackageStatus': 'Completed',
 'ModelPackageStatusDetails': {'ValidationStatuses': [],
  'ImageScanStatuses': []},
 'CertifyForMarketplace': False,
 'ModelApprovalStatus': 'PendingManualApproval',
 'LastModifiedTime': datetime.datetime(2020, 10, 29, 1, 28, 0, 438000,
 tzinfo=tzlocal()),
 'ResponseMetadata': {'RequestId': '12345678-abcd-1234-abcd-aabbccddeeff',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '212345678-abcd-1234-abcd-aabbccddeeff',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '1038',
   'date': 'Mon, 23 Nov 2020 04:59:38 GMT'},
  'RetryAttempts': 0}}
```

Lihat Detail Versi Model (konsol)

Untuk melihat detail versi model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model dari menu.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Pilih nama grup model yang berisi versi model yang akan dilihat.

6. Dalam daftar versi model, pilih versi model yang akan dilihat.
7. Untuk melihat detail yang terkait dengan pelatihan model, pilih tombol Radio pelatihan. Untuk melihat detail yang terkait dengan inferensi, pilih tombol radio Inferensi.

   Tab berikut mencakup detail yang terkait dengan pelatihan model:

   - Kinerja: Pengukuran statistik untuk menilai kinerja model, seperti kesalahan rata-rata relatif (RME).
   - Evaluasi: Bagan dan metrik untuk menggambarkan bias dan penjelasan.
   - Asosiasi: Sumber daya yang diturunkan, berasal dari, atau dikaitkan dengan versi model.
   - Aktivitas: Tindakan yang Anda lakukan dengan versi model, seperti persetujuan.
   - Tags: Tag yang termasuk dalam versi model.
   - Metadata: Informasi ARN untuk versi model dan peran Identity and Access Management (IAM) terkait.

   Tab berikut mencakup detail yang terkait dengan inferensi model:

   - Instance: Instance di mana model Anda digunakan.
   - Metadata: Kontainer yang menjalankan inferensi dengan model Anda.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda
   ().
3. Pilih Model, dan kemudian registri Model.
4. Dari grup model, pilih nama Grup Model yang ingin Anda lihat.
5. Tab baru muncul dengan daftar versi model di Grup Model.
6. Di daftar versi model, pilih nama versi model yang ingin Anda lihat detailnya.
7. Pada tab versi model yang terbuka, pilih salah satu dari berikut ini untuk melihat detail tentang versi model:

   - Aktivitas: Menampilkan peristiwa untuk versi model, seperti pembaruan status persetujuan.

- Kualitas model: Melaporkan metrik yang terkait dengan pemeriksaan kualitas model Model Monitor Anda, yang membandingkan prediksi model dengan Ground Truth. Untuk informasi selengkapnya tentang pemeriksaan kualitas model Model Monitor, lihatMonitor kualitas model.
- Keterjelasan: Melaporkan metrik yang terkait dengan pemeriksaan atribusi fitur Monitor Model Anda, yang membandingkan peringkat relatif fitur Anda dalam data pelatihan versus data langsung. Untuk informasi selengkapnya tentang pemeriksaan penjelasan Model Monitor, lihat. Monitor Fitur Atribusi Drift untuk Model dalam Produksi
- Bias: Melaporkan metrik yang terkait dengan pemeriksaan penyimpangan bias Monitor Model Anda, yang membandingkan distribusi data langsung dengan data pelatihan. Untuk informasi lebih lanjut tentang pemeriksaan penyimpangan bias Model Monitor, lihatMonitor Bias Drift untuk Model dalam Produksi.
- Rekomendasi inferensi: Memberikan rekomendasi instans awal untuk kinerja optimal berdasarkan model dan muatan sampel Anda.
- Uji beban: Menjalankan uji beban di seluruh pilihan jenis instans saat Anda memberikan persyaratan produksi spesifik, seperti batasan latensi dan throughput.
- Spesifikasi inferensi: Menampilkan jenis instans untuk inferensi real-time Anda dan mengubah pekerjaan, serta informasi tentang container Amazon ECR Anda.
- Informasi: Menampilkan informasi seperti proyek yang terkait dengan versi model, pipeline yang menghasilkan model, Grup Model, dan lokasi model di Amazon S3.

Bandingkan Versi Model

Saat Anda membuat versi model, Anda mungkin ingin membandingkan versi model dengan melihat metrik side-by-side kualitas model yang relevan. Misalnya, Anda mungkin ingin melacak akurasi dengan membandingkan nilai mean squared error (MSE), atau Anda mungkin memutuskan untuk menghapus model yang berkinerja buruk pada ukuran yang dipilih. Prosedur berikut menunjukkan cara menyiapkan perbandingan versi model di Registry Model menggunakan konsol Amazon SageMaker Studio Classic.

Bandingkan Versi Model (Amazon SageMaker Studio Classic)

> **Note**
>
> Anda hanya dapat membandingkan versi model konsol Amazon SageMaker Studio Classic.

Untuk membandingkan versi model dalam grup model, selesaikan langkah berikut:

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Dari grup model, pilih nama Grup Model yang ingin Anda lihat. Tab baru terbuka dengan daftar versi model di Grup Model.
5. Dalam daftar versi model, centang kotak di sebelah versi model yang ingin Anda bandingkan.
6. Pilih menu tarik-turun Tindakan, lalu pilih Bandingkan. Daftar metrik kualitas model muncul untuk model yang Anda pilih.

Lihat dan Kelola Grup Model dan Tag Versi Model

Model Registry membantu Anda melihat dan mengelola tag yang terkait dengan grup model Anda. Anda dapat menggunakan tag untuk mengelompokkan grup model berdasarkan tujuan, pemilik, lingkungan, atau kriteria lainnya. Petunjuk berikut menunjukkan cara melihat, menambah, menghapus, dan mengedit tag di konsol Amazon SageMaker Studio.

Melihat dan mengelola tag grup model

Studio

Untuk melihat tag grup model, lakukan langkah-langkah berikut:

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model untuk menampilkan daftar grup model Anda.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari Grup Model, pilih nama Grup Model yang ingin Anda lihat.
6. Di halaman grup model, pilih tab Tag. Lihat tag yang terkait dengan grup model.

Untuk menambahkan tag grup model, lakukan langkah-langkah berikut:

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model untuk menampilkan daftar grup model Anda.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari Grup Model, pilih nama Grup Model yang ingin Anda edit.
6. Di halaman grup model, pilih tab Tag.
7. Pilih Tambah/Edit tag.
8. Di atas+Tambahkan tag baru, masukkan kunci baru Anda di bidang Kunci kosong.
9. (Opsional) Masukkan nilai baru Anda di bidang Nilai kosong.
10. Pilih Konfirmasi perubahan.
11. Konfirmasikan tag baru Anda muncul di bagian Tag pada halaman Informasi.

Untuk menghapus tag grup model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model untuk menampilkan daftar grup model Anda.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari Grup Model, pilih nama Grup Model yang ingin Anda edit.
6. Di halaman grup model, pilih tab Tag.
7. Pilih Tambah/Edit tag.
8. Pilih ikon Sampah di samping pasangan nilai kunci yang ingin Anda hapus.
9. Pilih Konfirmasi perubahan.

Untuk mengedit tag grup model, lakukan langkah-langkah berikut:

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.

2. Di panel navigasi kiri, pilih Model untuk menampilkan daftar grup model Anda.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari Grup Model, pilih nama Grup Model yang ingin Anda edit.
6. Di halaman grup model, pilih tab Tag.
7. Pilih Tambah/Edit tag.
8. Masukkan nilai baru di bidang Nilai pasangan kunci yang ingin Anda edit.
9. Pilih Konfirmasikan perubahan.

Studio Classic

Untuk melihat tag grup model, lakukan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda
   ( ).
3. Pilih Model, dan kemudian registri Model.
4. Dari Grup Model, pilih nama Grup Model yang ingin Anda edit.
5. Pilih Informasi.
6. Lihat tag Anda di bagian Tag pada halaman Informasi.

Untuk menambahkan tag grup model, lakukan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di panel navigasi kiri, pilih ikon Beranda
   ( ).
3. Pilih Model, dan kemudian registri Model.
4. Dari Grup Model, pilih nama Grup Model yang ingin Anda edit.
5. Pilih Informasi.
6. Jika Anda tidak memiliki tag apa pun, pilih Tambahkan tag.

7. Jika Anda memiliki tag yang sudah ada sebelumnya, pilih Kelola tag di bagian Tag. Daftar tag grup model muncul sebagai pasangan nilai kunci.
8. Di atas Tambahkan tag baru, masukkan kunci baru Anda di bidang Kunci kosong.
9. (Opsional) Masukkan nilai baru Anda di bidang Nilai kosong.
10. Pilih Konfirmasikan perubahan.
11. Konfirmasikan tag baru Anda muncul di bagian Tag pada halaman Informasi.

Untuk menghapus tag grup model, selesaikan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di panel navigasi kiri, pilih ikon Beranda
   ().
3. Pilih Model, dan kemudian registri Model.
4. Dari Grup Model, pilih nama Grup Model yang ingin Anda edit.
5. Pilih Informasi.
6. Di bagian Tanda, pilih Kelola tanda. Daftar tag grup model muncul sebagai pasangan nilai kunci.
7. Pilih ikon Sampah di sebelah kanan tag yang ingin Anda hapus.
8. Pilih Konfirmasikan perubahan.
9. Konfirmasikan bahwa tag Anda yang dihapus tidak muncul di bagian Tag pada halaman Informasi.

Untuk mengedit tag grup model, lakukan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di panel navigasi kiri, pilih ikon Beranda
   ().
3. Pilih Model, dan kemudian registri Model.
4. Dari Grup Model, pilih nama Grup Model yang ingin Anda edit.
5. Pilih Informasi.

6. Di bagian Tanda, pilih Kelola tanda. Daftar tag grup model muncul sebagai pasangan nilai kunci.
7. Edit kunci atau nilai apa pun.
8. Pilih Konfirmasikan perubahan.
9. Konfirmasikan tag Anda berisi hasil edit Anda di bagian Tag pada halaman Informasi.

Kelola tag versi model

Untuk mengelola tag versi model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

Untuk menambahkan tag versi model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model dari menu.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Pilih nama grup model yang berisi versi model yang akan diperbarui.
6. Dalam daftar versi model, pilih versi model yang akan diperbarui.
7. Di bawah menu tarik-turun Tindakan, pilih Edit.
8. Jika Anda menambahkan tag pertama Anda, masukkan pasangan kunci-nilai baru Anda di bidang Kunci dan Nilai kosong.
9. (Opsional) Untuk menambahkan pasangan nilai kunci metadata kustom lainnya, pilih Tambah Baru dan masukkan pasangan nilai kunci baru Anda di kolom Nama dan Nilai kosong.
10. Pilih Simpan perubahan.

Untuk menghapus tag versi model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model dari menu.

3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Pilih nama grup model yang berisi versi model yang akan diperbarui.
6. Dalam daftar versi model, pilih versi model yang akan diperbarui.
7. Di bawah menu tarik-turun Tindakan, pilih Edit.
8. Pilih ikon Sampah di sebelah kanan tag yang ingin Anda hapus.
9. Pilih Konfirmasikan perubahan.

Untuk mengedit tag versi model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model dari menu.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Pilih nama grup model yang berisi versi model yang akan diperbarui.
6. Dalam daftar versi model, pilih versi model yang akan diperbarui.
7. Di bawah menu tarik-turun Tindakan, pilih Edit.
8. Edit kunci atau nilai apa pun.
9. Pilih Konfirmasikan perubahan.

Studio Classic

Untuk menambahkan tag versi model, selesaikan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Dari grup model, pilih nama Grup Model yang ingin Anda lihat. Tab baru terbuka dengan daftar versi model di Grup Model.

5. Dalam daftar versi model, pilih nama versi model yang ingin Anda perbarui.
6. Di bawah menu tarik-turun Tindakan, pilih Edit.
7. Jika Anda menambahkan tag pertama Anda, masukkan pasangan kunci-nilai baru Anda di bidang Kunci dan Nilai kosong.
8. (Opsional) Untuk menambahkan pasangan nilai kunci metadata kustom lainnya, pilih Tambah Baru dan masukkan pasangan nilai kunci baru Anda di kolom Nama dan Nilai kosong.
9. Pilih Simpan perubahan.
10. Konfirmasikan tag baru Anda muncul di bagian Tag pada halaman Informasi.

Untuk menghapus tag versi model, selesaikan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Dari grup model, pilih nama Grup Model yang ingin Anda lihat. Tab baru terbuka dengan daftar versi model di Grup Model.
5. Dalam daftar versi model, pilih nama versi model yang ingin Anda perbarui.
6. Di bawah menu tarik-turun Tindakan, pilih Edit.
7. Pilih ikon Sampah di sebelah kanan tag yang ingin Anda hapus.
8. Pilih Konfirmasikan perubahan.
9. Konfirmasikan bahwa tag Anda yang dihapus tidak muncul di bagian Tag pada halaman Informasi.

Untuk mengedit tag versi model, selesaikan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.

4. Dari daftar grup model, pilih nama Grup Model yang ingin Anda lihat. Tab baru terbuka dengan daftar versi model di Grup Model.
5. Dalam daftar versi model, pilih nama versi model yang ingin Anda perbarui.
6. Di bawah menu tarik-turun Tindakan, pilih Edit.
7. Edit kunci atau nilai apa pun.
8. Pilih Konfirmasikan perubahan.
9. Konfirmasikan tag Anda berisi hasil edit Anda di bagian Tag pada halaman Informasi.

Bagikan Model dengan Pengguna SageMaker Canvas

> **Note**
>
> Anda hanya dapat berbagi model dengan SageMaker Canvas di konsol Amazon SageMaker Studio Classic.

Anda mungkin memiliki model yang terdaftar di Registri Model Anda yang ingin Anda bagikan dengan pengguna di SageMaker Canvas. Anda dapat membagikan model yang telah dilatih di luar SageMaker selama itu terdaftar di Registri Model Anda. Dengan fungsi ini, pengguna SageMaker Canvas dapat mengimpor model yang telah Anda latih dan menghasilkan prediksi dengannya. Untuk informasi selengkapnya tentang cara berbagi model dengan pengguna SageMaker Canvas, lihatBawa model Anda sendiri ke SageMaker Canvas.

Menghapus Versi Model

Prosedur ini menunjukkan cara menghapus versi model di konsol Amazon SageMaker Studio.

Menghapus Versi Model (konsol)

Untuk menghapus versi model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model untuk menampilkan daftar grup model Anda.

3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari daftar grup model, pilih braket sudut di sebelah kiri grup model yang ingin Anda lihat.
6. Daftar versi model dalam grup model muncul. Jika Anda tidak melihat versi model yang ingin Anda hapus, pilih Lihat semua.
7. Pilih kotak centang di samping versi model yang ingin Anda hapus.
8. Pilih elipsis vertikal di atas sudut kanan atas tabel, dan pilih Hapus (atau Hapus versi model jika Anda berada di halaman detail grup model).
9. Di kotak dialog Hapus versi model, pilih Ya, hapus versi model.
10. Pilih Hapus.
11. Konfirmasikan bahwa versi model Anda yang dihapus tidak lagi muncul di grup model.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda
   ( ).
3. Pilih Model, dan kemudian registri Model. Daftar Grup Model Anda akan muncul.
4. Dari daftar grup model, pilih nama Grup Model dari versi model yang ingin Anda hapus.
5. Dari daftar versi model, pilih nama versi model yang ingin Anda hapus.
6. Pilih menu tarik-turun Tindakan, dan pilih Hapus.
7. Di kotak dialog konfirmasi, masukkan`REMOVE`.
8. Pilih Hapus.
9. Konfirmasikan bahwa versi model yang Anda hapus tidak muncul dalam daftar versi model grup model.

Memperbarui Status Persetujuan Model

Setelah membuat versi model, Anda biasanya ingin mengevaluasi kinerjanya sebelum menerapkannya ke titik akhir produksi. Jika sesuai dengan kebutuhan Anda, Anda dapat memperbarui status persetujuan versi model ke`Approved`. Mengatur status untuk `Approved` dapat

memulai penyebaran CI/CD untuk model. Jika versi model tidak sesuai dengan kebutuhan Anda, Anda dapat memperbarui status persetujuan ke`Rejected`.

Anda dapat memperbarui status persetujuan versi model secara manual setelah Anda mendaftarkannya, atau Anda dapat membuat langkah kondisi untuk mengevaluasi model saat Anda membuat SageMaker pipeline. Untuk informasi tentang membuat langkah kondisi dalam SageMaker pipeline, lihatMembuat Alur.

Bila Anda menggunakan salah satu templat proyek yang SageMaker disediakan dan status persetujuan versi model berubah, tindakan berikut akan terjadi. Hanya transisi yang valid yang ditampilkan.

- `PendingManualApproval`untuk `Approved` — memulai penyebaran CI/CD untuk versi model yang disetujui
- `PendingManualApproval`ke `Rejected` - Tidak ada tindakan
- `Rejected`untuk `Approved` — memulai penyebaran CI/CD untuk versi model yang disetujui
- `Approved`untuk `Rejected` — memulai CI/CD untuk menyebarkan versi model terbaru dengan status `Approved`

Anda dapat memperbarui status persetujuan versi model dengan menggunakan AWS SDK for Python (Boto3) atau dengan menggunakan konsol Amazon SageMaker Studio. Anda juga dapat memperbarui status persetujuan versi model sebagai bagian dari langkah kondisi dalam SageMaker pipeline. Untuk informasi tentang menggunakan langkah persetujuan model dalam SageMaker pipeline, lihatSageMaker Ikhtisar Pipelines.

Perbarui Status Persetujuan Model (Boto3)

Saat Anda membuat versi model diDaftarkan Versi Model, Anda menyetel `ModelApprovalStatus` ke`PendingManualApproval`. Anda memperbarui status persetujuan untuk model dengan meneleponupdate_model_package. Perhatikan bahwa Anda dapat mengotomatiskan proses ini dengan menulis kode yang, misalnya, menetapkan status persetujuan model tergantung pada hasil evaluasi beberapa ukuran kinerja model. Anda juga dapat membuat langkah dalam pipeline yang secara otomatis menerapkan versi model baru saat disetujui. Potongan kode berikut menunjukkan cara mengubah status persetujuan secara manual. `Approved`

```
model_package_update_input_dict = {
    "ModelPackageArn" : model_package_arn,
```

```
    "ModelApprovalStatus" : "Approved"
}
model_package_update_response =
 sm_client.update_model_package(**model_package_update_input_dict)
```

Memperbarui Status Persetujuan Model (konsol)

Untuk mengubah status persetujuan secara manual di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model untuk menampilkan daftar grup model Anda.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari daftar grup model, pilih braket sudut di sebelah kiri grup model yang ingin Anda lihat.
6. Daftar versi model dalam grup model muncul. Jika Anda tidak melihat versi model yang ingin Anda hapus, pilih Lihat semua untuk menampilkan daftar lengkap versi model di halaman detail grup model.
7. Pilih nama versi model yang ingin Anda perbarui.
8. Pilih elipsis vertikal di kanan atas, pilih Perbarui status, lalu status model akhir.
9. Dalam kotak dialog Perbarui status model, masukkan komentar opsional dan pilih Simpan dan perbarui.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Dari daftar grup model, pilih nama Grup Model yang ingin Anda lihat. Tab baru terbuka dengan daftar versi model di Grup Model.

5. Di daftar versi model, pilih nama versi model yang ingin Anda perbarui.
6. Di bawah menu tarik-turun Tindakan, Anda dapat memilih salah satu dari dua opsi menu yang memungkinkan untuk memperbarui status versi model.

   - Menggunakan opsi Perbarui Status

     1. Di bawah menu tarik-turun Tindakan, pilih menu tarik-turun Perbarui Status, dan pilih status versi model baru.
     2. (Opsional) Di kolom Komentar, tambahkan detail tambahan.
     3. Pilih Simpan dan Perbarui.

   - Menggunakan opsi Edit

     1. Di bawah menu tarik-turun Tindakan, pilih Edit.
     2. (Opsional) Di kolom Komentar, tambahkan detail tambahan.
     3. Pilih Simpan perubahan.
7. Konfirmasikan status versi model diperbarui ke nilai yang benar di halaman versi model.

Menyebarkan Model dari Registry

Setelah Anda mendaftarkan versi model dan menyetujuinya untuk penerapan, terapkan ke SageMaker titik akhir untuk inferensi waktu nyata. Anda dapat menerapkan model Anda dengan menggunakan SageMaker SDK atau AWS SDK for Python (Boto3) (Boto3).

Saat Anda membuat proyek operasi pembelajaran mesin (MLOPs) dan memilih templat proyek MLOPS yang menyertakan penerapan model, versi model yang disetujui dalam Registri Model secara otomatis diterapkan ke produksi. Untuk informasi tentang penggunaan proyek SageMaker MLOPs, lihat. Otomatiskan MLOP dengan Proyek SageMaker

Anda juga dapat mengaktifkan AWS akun untuk menerapkan versi model yang dibuat di akun lain dengan menambahkan kebijakan sumber daya lintas akun. Misalnya, satu tim di organisasi Anda mungkin bertanggung jawab atas model pelatihan, dan tim yang berbeda bertanggung jawab untuk menerapkan dan memperbarui model.

Topik

- Menerapkan Model dari Registry (SageMaker SDK)
- Menyebarkan Model dari Registry (Boto3)
- Menerapkan Versi Model dari Akun yang Berbeda

Menerapkan Model dari Registry (SageMaker SDK)

Untuk menerapkan versi model menggunakan Amazon SageMaker Python SDK gunakan cuplikan kode berikut:

```
from sagemaker import ModelPackage
from time import gmtime, strftime

model_package_arn = 'arn:aws:sagemaker:us-east-2:12345678901:model-package/modeltest/1'
model = ModelPackage(role=role,
                     model_package_arn=model_package_arn,
                     sagemaker_session=sagemaker_session)
model.deploy(initial_instance_count=1, instance_type='ml.m5.xlarge')
```

Menyebarkan Model dari Registry (Boto3)

Untuk menerapkan versi model menggunakanAWS SDK for Python (Boto3), selesaikan langkah-langkah berikut:

1. Cuplikan kode berikut mengasumsikan Anda sudah membuat klien SageMaker Boto3 `sm_client` dan versi model yang ARN disimpan dalam variabel. `model_version_arn`

   Buat objek model dari versi model dengan memanggil operasi API create_model. Lulus Amazon Resource Name (ARN) dari versi model sebagai bagian dari `Containers` untuk objek model:

   ```
   model_name = 'DEMO-modelregistry-model-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
   print("Model name : {}".format(model_name))
   container_list = [{'ModelPackageName': model_version_arn}]

   create_model_response = sm_client.create_model(
       ModelName = model_name,
       ExecutionRoleArn = role,
       Containers = container_list
   )
   print("Model arn : {}".format(create_model_response["ModelArn"]))
   ```

2. Buat konfigurasi titik akhir dengan memanggil`create_endpoint_config`. Konfigurasi titik akhir menentukan nomor dan tipe instans Amazon EC2 yang akan digunakan untuk titik akhir.

   ```
   endpoint_config_name = 'DEMO-modelregistry-EndpointConfig-' + strftime("%Y-%m-%d-
   %H-%M-%S", gmtime())
   print(endpoint_config_name)
   ```

```
create_endpoint_config_response = sm_client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType':'ml.m4.xlarge',
        'InitialVariantWeight':1,
        'InitialInstanceCount':1,
        'ModelName':model_name,
        'VariantName':'AllTraffic'}])
```

3. Buat titik akhir dengan meneleponcreate_endpoint.

```
endpoint_name = 'DEMO-modelregistry-endpoint-' + strftime("%Y-%m-%d-%H-%M-%S",
 gmtime())
print("EndpointName={}".format(endpoint_name))

create_endpoint_response = sm_client.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name)
print(create_endpoint_response['EndpointArn'])
```

Menerapkan Versi Model dari Akun yang Berbeda

Anda dapat mengizinkan AWS akun untuk menerapkan versi model yang dibuat di akun lain dengan menambahkan kebijakan sumber daya lintas akun. Misalnya, satu tim di organisasi Anda mungkin bertanggung jawab atas model pelatihan, dan tim yang berbeda bertanggung jawab untuk menerapkan dan memperbarui model. Saat membuat kebijakan sumber daya ini, Anda menerapkan kebijakan tersebut ke sumber daya tertentu yang ingin Anda berikan aksesnya. Untuk informasi selengkapnya tentang kebijakan sumber daya lintas akunAWS, lihat Logika evaluasi kebijakan lintas akun di AWS Identity and Access ManagementPanduan Pengguna.

> **Note**
>
> Anda harus menggunakan kunci KMS untuk mengenkripsi tindakan konfigurasi data keluaran selama pelatihan untuk penerapan model lintas akun.

Untuk mengaktifkan penerapan model lintas akun SageMaker, Anda harus menyediakan kebijakan sumber daya lintas akun untuk Grup Model yang berisi versi model yang ingin Anda terapkan, repositori Amazon ECR tempat gambar inferensi untuk Grup Model berada, dan bucket Amazon S3 tempat versi model disimpan.

Untuk dapat menerapkan model yang dibuat di akun lain, Anda harus memiliki peran yang memiliki akses ke SageMaker tindakan, seperti peran dengan kebijakan `AmazonSageMakerFullAccess` terkelola. Untuk informasi tentang kebijakan SageMaker terkelola, lihatAWSKebijakan Terkelola untuk Amazon SageMaker.

Contoh berikut membuat kebijakan lintas akun untuk ketiga sumber daya ini, dan menerapkan kebijakan ke sumber daya. Contoh ini juga mengasumsikan bahwa Anda sebelumnya mendefinisikan variabel berikut:

- `bucket`- Bucket Amazon S3 tempat versi model disimpan.
- `kms_key_id`— Kunci KMS yang digunakan untuk mengenkripsi output pelatihan.
- `sm_client`- Klien SageMaker Boto3.
- `model_package_group_name`— Grup Model tempat Anda ingin memberikan akses akun.
- `model_package_group_arn`— Grup Model ARN tempat Anda ingin memberikan akses akun.

```
import json

# The cross-account id to grant access to
cross_account_id = "123456789012"

# Create the policy for access to the ECR repository
ecr_repository_policy = {
    'Version': '2012-10-17',
    'Statement': [{
        'Sid': 'AddPerm',
        'Effect': 'Allow',
        'Principal': {
            'AWS': f'arn:aws:iam::{cross_account_id}:root'
        },
        'Action': ['ecr:*']
    }]
}

# Convert the ECR policy from JSON dict to string
ecr_repository_policy = json.dumps(ecr_repository_policy)

# Set the new ECR policy
ecr = boto3.client('ecr')
response = ecr.set_repository_policy(
    registryId = account,
```

```
    repositoryName = 'decision-trees-sample',
    policyText = ecr_repository_policy
)

# Create a policy for accessing the S3 bucket
bucket_policy = {
    'Version': '2012-10-17',
    'Statement': [{
        'Sid': 'AddPerm',
        'Effect': 'Allow',
        'Principal': {
            'AWS': f'arn:aws:iam::{cross_account_id}:root'
        },
        'Action': 's3:*',
        'Resource': f'arn:aws:s3:::{bucket}/*'
    }]
}

# Convert the policy from JSON dict to string
bucket_policy = json.dumps(bucket_policy)

# Set the new policy
s3 = boto3.client('s3')
response = s3.put_bucket_policy(
    Bucket = bucket,
    Policy = bucket_policy)

# Create the KMS grant for encryption in the source account to the
# Model Registry account Model Group
client = boto3.client('kms')

response = client.create_grant(
    GranteePrincipal=cross_account_id,
    KeyId=kms_key_id
    Operations=[
        'Decrypt',
        'GenerateDataKey',
    ],
)

# 3. Create a policy for access to the Model Group.
model_package_group_policy = {
    'Version': '2012-10-17',
    'Statement': [{
```

```
        'Sid': 'AddPermModelPackageGroup',
        'Effect': 'Allow',
        'Principal': {
            'AWS': f'arn:aws:iam::{cross_account_id}:root'
        },
        'Action': ['sagemaker:DescribeModelPackageGroup'],
        'Resource': f'arn:aws:sagemaker:{region}:{account}:model-package-group/
{model_package_group_name}'
    },{
        'Sid': 'AddPermModelPackageVersion',
        'Effect': 'Allow',
        'Principal': {
            'AWS': f'arn:aws:iam::{cross_account_id}:root'
        },
        'Action': ["sagemaker:DescribeModelPackage",
                   "sagemaker:ListModelPackages",
                   "sagemaker:UpdateModelPackage",
                   "sagemaker:CreateModel"],
        'Resource': f'arn:aws:sagemaker:{region}:{account}:model-package/
{model_package_group_name}/*'
    }]
}

# Convert the policy from JSON dict to string
model_package_group_policy = json.dumps(model_package_group_policy)

# Set the policy to the Model Group
response = sm_client.put_model_package_group_policy(
    ModelPackageGroupName = model_package_group_name,
    ResourcePolicy = model_package_group_policy)

print('ModelPackageGroupArn :
 {}'.format(create_model_package_group_response['ModelPackageGroupArn']))
print("First Versioned ModelPackageArn: " + model_package_arn)
print("Second Versioned ModelPackageArn: " + model_package_arn2)

print("Success! You are all set to proceed for cross-account deployment.")
```

Lihat Riwayat Penerapan Model

Untuk melihat penerapan versi model di konsol Amazon SageMaker Studio, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

Melihat sejarah deployment untuk versi model

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model untuk menampilkan daftar grup model Anda.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Dari daftar grup model, pilih braket sudut di sebelah kiri grup model yang ingin Anda lihat.
6. Daftar versi model dalam grup model muncul. Jika Anda tidak melihat versi model yang ingin Anda hapus, pilih Lihat semua.
7. Pilih nama versi model yang ingin Anda lihat.
8. Pilih tab Aktivitas. Penerapan untuk versi model muncul sebagai peristiwa dalam daftar aktivitas dengan jenis Peristiwa. ModelDeployment

Studio Classic

Melihat sejarah deployment untuk versi model

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda
   ().
3. Pilih Model, dan kemudian registri Model.
4. Dari daftar grup model, pilih nama Grup Model yang ingin Anda lihat.
5. Tab baru muncul dengan daftar versi model di Grup Model.
6. Di daftar versi model, pilih nama versi model yang ingin Anda lihat detailnya.
7. Pada tab versi model yang terbuka, pilih Aktivitas. Penerapan untuk versi model muncul sebagai peristiwa dalam daftar aktivitas dengan jenis Peristiwa. ModelDeployment

Koleksi Registri Model

Anda dapat menggunakan Koleksi untuk mengelompokkan model terdaftar yang terkait satu sama lain dan mengaturnya dalam hierarki untuk meningkatkan kemampuan penemuan model dalam skala besar. Dengan Koleksi, Anda dapat mengatur model terdaftar yang terkait satu sama lain. Misalnya, Anda dapat mengkategorikan model Anda berdasarkan domain masalah yang mereka selesaikan sebagai Koleksi berjudul NLP-model, CV-model, atau S. peech-recognition-models Untuk mengatur model terdaftar Anda dalam struktur pohon, Anda dapat membuat sarang Koleksi satu sama lain. Operasi apa pun yang Anda lakukan pada Koleksi, seperti membuat, membaca, memperbarui, atau menghapus, tidak akan mengubah model terdaftar Anda. Anda dapat menggunakan Amazon SageMaker Studio UI atau Python SDK untuk mengelola Koleksi.

Tab Koleksi di Registri Model menampilkan daftar semua Koleksi di akun Anda. Bagian berikut menjelaskan bagaimana Anda dapat menggunakan opsi di tab Koleksi untuk melakukan hal berikut:

- Buat Koleksi
- Menambahkan Grup Model ke Koleksi
- Pindahkan Grup Model antar Koleksi
- Hapus Grup Model atau Koleksi dari Koleksi lain

Operasi apa pun yang Anda lakukan pada Koleksi tidak memengaruhi integritas masing-masing Grup Model yang dikandungnya—artefak Grup Model yang mendasari di Amazon S3 dan Amazon ECR tidak dimodifikasi.

Sementara Koleksi memberikan fleksibilitas yang lebih besar dalam mengatur model Anda, representasi internal memaksakan beberapa kendala pada ukuran hierarki Anda. Untuk ringkasan kendala ini, lihat. Batasan

Topik berikut menunjukkan kepada Anda cara membuat dan bekerja dengan Koleksi di Registri Model.

Topik

- Melihat Koleksi Induk Grup Model
- Batasan

Prasyarat

Buat kebijakan khusus yang mencakup tindakan Resource Groups yang diperlukan berikut ini:

- `resource-groups:CreateGroup`
- `resource-groups:DeleteGroup`
- `resource-groups:GetGroupQuery`
- `resource-groups:ListGroupResources`
- `resource-groups:Tag`
- `tag:GetResources`

Untuk petunjuk tentang cara menambahkan kebijakan sebaris, lihat Menambahkan izin identitas IAM (konsol). Bila Anda memilih format kebijakan, pilih format JSON dan tambahkan kebijakan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:ListGroupResources"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:GetGroupQuery"
            ],
            "Resource": "arn:aws:resource-groups:*:*:group/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:CreateGroup",
                "resource-groups:Tag"
```

```
            ],
            "Resource": "arn:aws:resource-groups:*:*:group/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "sagemaker:collection"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "resource-groups:DeleteGroup",
            "Resource": "arn:aws:resource-groups:*:*:group/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/sagemaker:collection": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Untuk membuat koleksi

Anda dapat membuat Collection di konsol SMMonarchLong;. Untuk membuat Koleksi, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Koleksi.
5. (Opsional) Untuk membuat Koleksi di dalam Koleksi lain, arahkan ke hierarki tempat Anda ingin menambahkan Koleksi. Jika tidak, Koleksi Anda dibuat di tingkat root.
6. Di menu tarik-turun Tindakan di kanan atas, pilih Buat koleksi baru.

7. Masukkan nama untuk Koleksi Anda di bidang Nama pada kotak dialog.

> **Note**
>
> Jika Anda berencana untuk membuat beberapa hierarki dalam Koleksi ini, pertahankan agar nama Koleksi Anda tetap pendek. Jalur absolut, yang merupakan string yang mewakili lokasi Koleksi Anda dari tingkat root, harus 256 karakter atau kurang. Untuk detail tambahan, lihat Koleksi dan penandaan Grup Model.

8. (Opsional) Untuk menambahkan Grup Model ke Koleksi Anda, selesaikan langkah-langkah berikut:

   a. Pilih Pilih grup model.

   b. Pilih Grup Model yang ingin Anda tambahkan. Anda dapat memilih hingga 10 menit.

9. Pilih Buat.

10. Periksa untuk memastikan Koleksi Anda dibuat dalam hierarki saat ini. Jika Anda tidak segera melihat Koleksi baru Anda, pilih Refresh.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.

2. Di panel navigasi kiri, pilih ikon Beranda ().

3. Pilih Model, dan kemudian registri Model.

4. Pilih tab Koleksi.

5. (Opsional) Untuk membuat Koleksi di dalam Koleksi lain, arahkan ke hierarki tempat Anda ingin menambahkan Koleksi. Jika tidak, Koleksi Anda dibuat di tingkat root.

6. Di menu tarik-turun Tindakan di kanan atas, pilih Buat koleksi baru.

7. Masukkan nama untuk Koleksi Anda di bidang Nama pada kotak dialog.

> **Note**
>
> Jika Anda berencana untuk membuat beberapa hierarki dalam Koleksi ini, pertahankan agar nama Koleksi Anda tetap pendek. Jalur absolut, yang merupakan

string yang mewakili lokasi Koleksi Anda dari tingkat root, harus 256 karakter atau kurang. Untuk detail tambahan, lihat Koleksi dan penandaan Grup Model.

8. (Opsional) Untuk menambahkan Grup Model ke Koleksi Anda, selesaikan langkah-langkah berikut:
   a. Pilih Pilih grup model.
   b. Pilih Grup Model yang ingin Anda tambahkan. Anda dapat memilih hingga 10 menit.
9. Pilih Buat.
10. Periksa untuk memastikan Koleksi Anda dibuat dalam hierarki saat ini. Jika Anda tidak segera melihat Koleksi baru Anda, pilih Refresh.

Menambahkan Grup Model ke Koleksi

Anda dapat menambahkan grup model ke Koleksi di konsol Amazon SageMaker Studio. Untuk menambahkan Grup Model ke Koleksi, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Model, jika belum dipilih.
5. Pilih kotak centang di samping grup model yang ingin Anda tambahkan. Anda dapat memilih hingga 10 Grup Model. Jika Anda memilih lebih dari 10, opsi UI untuk menambahkan Grup Model Anda ke Koleksi tidak aktif.
6. Pilih elipsis vertikal di sebelah Buat, dan pilih Tambahkan ke koleksi.
7. Pilih tombol radio untuk koleksi yang ingin Anda tambahkan Grup Model yang Anda pilih.
8. Pilih Tambahkan ke koleksi.
9. Periksa untuk memastikan Grup Model Anda ditambahkan ke koleksi. Di kolom Koleksi Grup Model yang Anda pilih, Anda akan melihat nama koleksi tempat Anda menambahkan Grup Model.

Studio Classic

Anda dapat menambahkan Grup Model ke Koleksi dari tab Grup Model atau Koleksi.

Untuk menambahkan satu atau beberapa Grup Model ke Koleksi dari tab Koleksi, selesaikan langkah-langkah berikut:

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Pilih tab Koleksi.
5. Pilih Koleksi yang ingin Anda tambahkan Grup Model. Jika Koleksi yang diinginkan tidak pada tingkat root, navigasikan ke hierarki tempat Anda ingin menambahkan Grup Model Anda.
6. Di menu tarik-turun Tindakan di kanan atas, pilih Tambahkan grup model.
7. Pilih Grup Model yang ingin Anda tambahkan. Anda dapat memilih hingga 10 Grup Model. Jika Anda memilih lebih dari 10, opsi UI untuk menambahkan Grup Model Anda ke Koleksi tidak aktif.
8. Pilih Tambahkan ke koleksi.
9. Periksa untuk memastikan Grup Model Anda ditambahkan dalam hierarki saat ini. Jika Anda tidak segera melihat Grup Model baru Anda, pilih Refresh.

Untuk menambahkan satu atau beberapa Grup Model ke Koleksi dari tab Grup Model, selesaikan langkah-langkah berikut:

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Pilih tab Grup Model.
5. Pilih Grup Model yang ingin Anda tambahkan. Anda dapat memilih hingga 10 menit. Jika Anda memilih lebih dari 10, opsi UI untuk menambahkan Grup Model Anda ke Koleksi tidak aktif.

6. Di menu tarik-turun Tindakan di kanan atas, pilih Tambahkan ke koleksi.
7. Dalam dialog pop-up, pilih lokasi `Collections` jalur root. Tautan ke lokasi root ini muncul di atas tabel.
8. Arahkan ke hierarki yang berisi Koleksi tujuan Anda, atau di mana Anda ingin membuat Koleksi baru yang Anda tambahkan model Anda.
9. (Opsional) Untuk menambahkan Grup Model Anda ke Koleksi yang ada, selesaikan langkah-langkah berikut:
    a. Pilih Koleksi tujuan.
    b. Pilih Tambahkan ke koleksi.
10. (Opsional) Untuk menambahkan Grup Model Anda ke Koleksi baru, selesaikan langkah-langkah berikut:
    a. Pilih Koleksi baru.
    b. Masukkan nama untuk Koleksi baru Anda.
    c. Pilih Buat.

Menghapus Grup Model atau Koleksi dari Koleksi

Saat Anda menghapus Grup Model atau Koleksi dari Koleksi, Anda menghapusnya dari pengelompokan tertentu dan bukan dari Registri Model. Anda dapat menghapus Grup Model dari Koleksi di konsol Amazon SageMaker Studio.

Untuk menghapus satu atau beberapa Grup Model atau Koleksi dari Koleksi, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Koleksi.
5. Arahkan ke Koleksi yang berisi Grup Model atau Koleksi yang ingin Anda hapus.

6. Pilih Grup Model atau Koleksi yang ingin Anda hapus. Anda dapat memilih hingga 10 menit. Jika Anda memilih lebih dari 10 Grup Model atau Koleksi, opsi UI untuk menghapusnya tidak aktif.

> **Important**
>
> Anda tidak dapat secara bersamaan memilih Grup Model dan Koleksi untuk dihapus. Untuk menghapus Grup Model dan Koleksi, pertama-tama hapus Grup Model, lalu hapus Koleksi.

> **Important**
>
> Anda tidak dapat menghapus Koleksi yang tidak kosong. Untuk menghapus Koleksi yang tidak kosong, pertama-tama hapus isinya.

7. Di menu tarik-turun Tindakan di kanan atas, pilih Hapus item X dari koleksi (di mana X adalah jumlah Grup Model yang Anda pilih).
8. Konfirmasi bahwa Anda ingin menghapus Grup Model yang dipilih.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Pilih tab Koleksi.
5. Arahkan ke Koleksi yang berisi Grup Model atau Koleksi yang ingin Anda hapus.
6. Pilih Grup Model atau Koleksi yang ingin Anda hapus. Anda dapat memilih hingga 10 menit. Jika Anda memilih lebih dari 10 Grup Model atau Koleksi, opsi UI untuk menghapusnya tidak aktif.

> **Important**
>
> Anda tidak dapat secara bersamaan memilih Grup Model dan Koleksi untuk dihapus. Untuk menghapus Grup Model dan Koleksi, pertama-tama hapus Grup Model, lalu hapus Koleksi.

> **Important**
>
> Anda tidak dapat menghapus Koleksi yang tidak kosong. Untuk menghapus Koleksi yang tidak kosong, pertama-tama hapus isinya.

7. Di menu tarik-turun Tindakan di kanan atas, pilih Hapus item X dari koleksi (di mana X adalah jumlah Grup Model yang Anda pilih).
8. Konfirmasi bahwa Anda ingin menghapus Grup Model yang dipilih.

Memindahkan Grup Model Antar Koleksi

Anda dapat memindahkan satu atau beberapa Grup Model dari satu Koleksi ke Koleksi lainnya di konsol Amazon SageMaker Studio.

Untuk memindahkan Grup Model, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.
2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Koleksi.
5. Navigasi ke Koleksi yang berisi Grup Model yang ingin Anda pindahkan.
6. Pilih Grup Model yang ingin Anda pindahkan. Anda dapat memilih hingga 10 menit. Jika Anda memilih lebih dari 10, opsi UI untuk memindahkan Grup Model Anda tidak aktif.
7. Di menu tarik-turun Tindakan di kanan atas, pilih Pindah ke.

8. Di kotak dialog, pilih lokasi jalur root`Collections`. Tautan ke lokasi root ini muncul di atas tabel.
9. Arahkan ke hierarki yang berisi Koleksi tujuan Anda.
10. Pilih Koleksi tujuan Anda di tabel.
11. Pilih Pindah di sini.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Pilih tab Koleksi.
5. Navigasi ke Koleksi yang berisi Grup Model yang ingin Anda pindahkan.
6. Pilih Grup Model yang ingin Anda pindahkan. Anda dapat memilih hingga 10 menit. Jika Anda memilih lebih dari 10, opsi UI untuk memindahkan Grup Model Anda tidak aktif.
7. Di menu tarik-turun Tindakan di kanan atas, pilih Pindah ke.
8. Di kotak dialog, pilih lokasi jalur root`Collections`. Tautan ke lokasi root ini muncul di atas tabel.
9. Arahkan ke hierarki yang berisi Koleksi tujuan Anda.
10. Pilih Koleksi tujuan Anda di tabel.
11. Pilih Pindah di sini.

Melihat Koleksi Induk Grup Model

Anda dapat melihat Koleksi yang berisi Grup Model tertentu di konsol Amazon SageMaker Studio.

Untuk melihat Koleksi yang berisi Grup Model tertentu, selesaikan langkah-langkah berikut berdasarkan apakah Anda menggunakan Studio atau Studio Classic.

Studio

1. Buka konsol SageMaker Studio dengan mengikuti petunjuk di Luncurkan Amazon SageMaker Studio.

2. Di panel navigasi kiri, pilih Model.
3. Pilih tab Model terdaftar, jika belum dipilih.
4. Tepat di bawah label tab Model terdaftar, pilih Grup Model, jika belum dipilih.
5. Lihat kolom Koleksi untuk Grup Model Anda, yang menampilkan nama Koleksi yang berisi Grup Model ini. Jika beberapa Koleksi berisi Grup Model ini, pilih entri kolom Koleksi untuk menampilkan daftar pop-up Koleksi yang berisi Grup Model ini.

Studio Classic

1. Masuk ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Meluncurkan Amazon SageMaker Studio Classic.
2. Di panel navigasi kiri, pilih ikon Beranda ().
3. Pilih Model, dan kemudian registri Model.
4. Pilih tab Grup Model.
5. Temukan Grup Model Anda di tabel.
6. Lihat kolom Koleksi untuk Grup Model Anda, yang menampilkan nama Koleksi yang berisi Grup Model ini. Jika beberapa Koleksi berisi Grup Model ini, pilih entri kolom Koleksi untuk menampilkan daftar pop-up Koleksi yang berisi Grup Model ini.

Batasan

Saat menggunakan Koleksi, Anda mungkin menghadapi masalah yang terkait dengan batasan panjang tag atau batas tarif untuk operasi Koleksi. Tinjau daftar peringatan berikut sehingga Anda dapat menghindari masalah yang terkait dengan batasan ini saat Anda bekerja dengan Koleksi Anda.

Kendala VPC

- Koleksi tidak didukung dalam mode VPC.

Kendala operasi pengumpulan

- Anda dapat menambahkan maksimum 10 Grup Model ke Koleksi sekaligus.
- Anda dapat menghapus maksimal 10 Grup Model dari Koleksi sekaligus.

- Anda dapat memindahkan maksimal 10 Grup Model dari satu Koleksi ke Koleksi lainnya sekaligus.
- Anda tidak dapat menghapus Koleksi kecuali kosong.
- Grup Model dapat menjadi milik beberapa Koleksi, tetapi Koleksi hanya dapat dimiliki oleh satu Koleksi.

Kendala terkait tag

- Grup Model dapat tergabung dalam jumlah maksimal sebanyak 48 Koleksi. Untuk detail selengkapnya, lihat bagian Koleksi dan penandaan Grup Model berikut:
- Jalur absolut A Collection dapat memiliki panjang maksimal 256 karakter. Karena nama Collection ditentukan pengguna, Anda dapat mengontrol panjang jalur. Untuk detail selengkapnya, lihat bagian Koleksi dan penandaan Grup Model berikut:

Koleksi dan penandaan Grup Model

Registri SageMaker Model menggunakan aturan tag dan tag untuk secara internal mewakili pengelompokan dan hierarki Koleksi Anda. Anda dapat mengakses elemen tag ini diAWS Resource Access Manager, SageMaker SDK, danAWS CLI, tetapi penting bahwa Anda tidak mengubah atau menghapusnya.

> **Important**
>
> Jangan menghapus atau mengubah aturan tag atau tag apa pun yang termasuk dalam Koleksi atau Grup Model Anda. Melakukannya mencegah Anda melakukan operasi Koleksi!

Aturan tag adalah pasangan kunci-nilai yang SageMaker digunakan untuk mengidentifikasi lokasi Collection dalam hierarki. Singkatnya, kuncinya adalah kunci dari Koleksi induk, dan nilainya adalah jalur Koleksi dalam hierarki. SageMaker memungkinkan nilai tag menjadi 256 karakter atau kurang, jadi jika Anda memiliki beberapa hierarki bersarang, Anda disarankan untuk menjaga nama Koleksi tetap pendek.

> **Important**
>
> Jaga agar nama Koleksi Anda tetap pendek. Jalur absolut ke Koleksi apa pun harus memiliki panjang 256 karakter atau kurang.

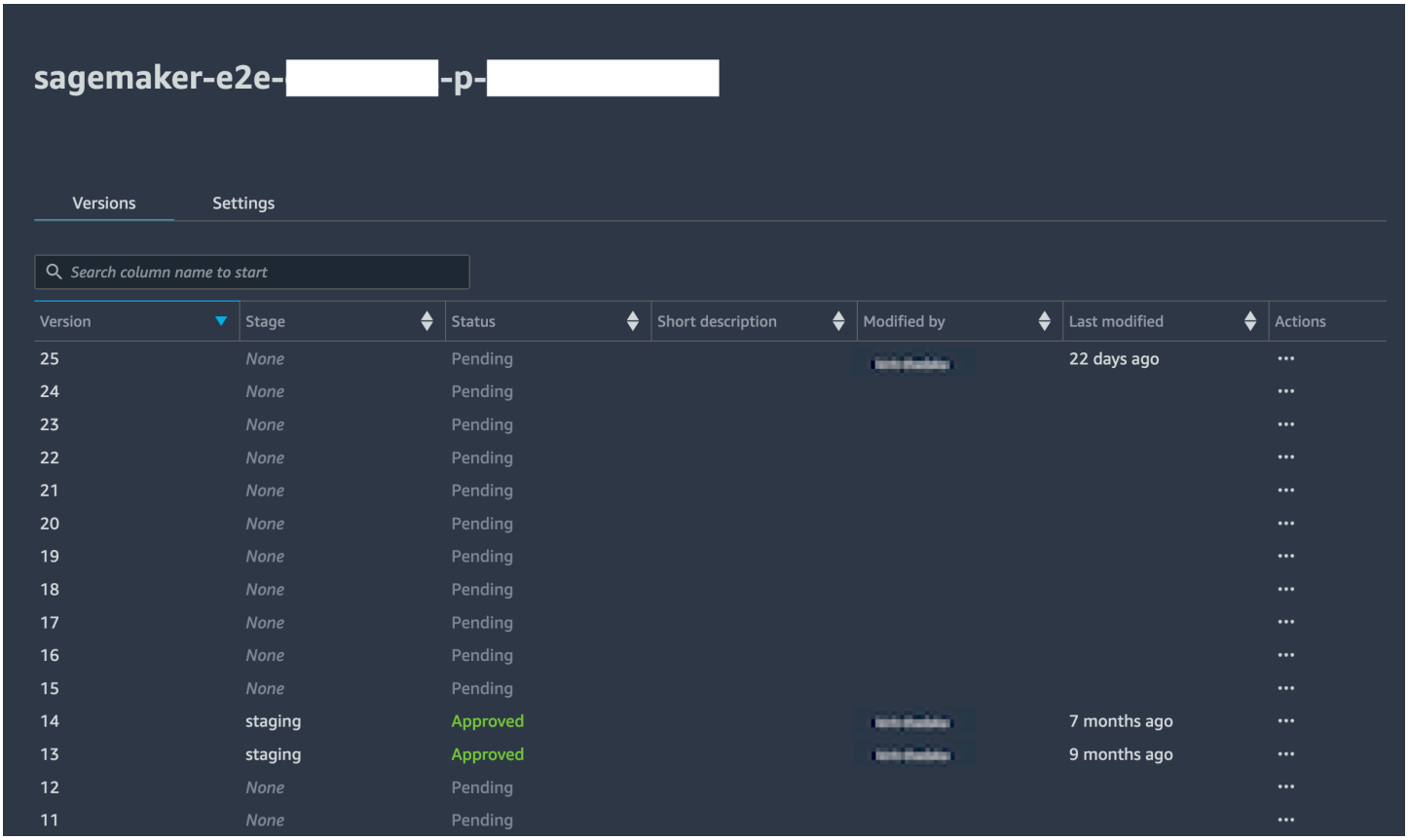
Grup Model, di sisi lain, tidak memiliki aturan tag tetapi menggunakan tag. Tag Grup Model menyertakan aturan tag untuk semua Koleksi yang berisi Grup Model. Misalnya, jika empat Koleksi berisi model-group-1, model-group-1 memiliki empat tag. SageMaker memungkinkan satu AWS sumber daya memiliki maksimum 50 tag. Karena dua pra-dialokasikan untuk tujuan umum, Grup Model dapat memiliki maksimum 48 tag. Sebagai kesimpulan, Grup Model dapat tergabung dalam maksimum 48 Koleksi.

FAQ Registri SageMaker Model Amazon

Gunakan item FAQ berikut untuk menemukan jawaban atas pertanyaan umum tentang SageMaker Model Registry.

T. Bagaimana saya harus mengatur model saya ke dalam Grup Model dan paket model di Registri SageMaker Model?

Paket model adalah model aktual yang terdaftar ke dalam Registry Model sebagai entitas berversi. Harap dicatat bahwa ada dua cara Anda dapat menggunakan paket model SageMaker. Salah satunya adalah dengan SageMakerMarketplace — paket model ini tidak berversi. Yang lainnya adalah dengan SageMaker Model Registry, di mana paket model harus berversi. Registri Model menerima setiap model baru yang Anda latih ulang, memberikannya versi, dan menetapkannya ke Grup Model di dalam Registri Model. Gambar berikut menunjukkan contoh Grup Model dengan 25 model berversi berturut-turut.



T. Bagaimana SageMaker Model Registry berbeda dari Amazon Elastic Container Registry (Amazon ECR) Registry?

SageMaker Model Registry adalah toko metadata untuk model pembelajaran mesin Anda. Amazon Elastic Container Registry adalah repositori yang menyimpan semua kontainer Anda. Dalam Registri Model, model diversikan dan terdaftar sebagai paket model dalam Grup Model. Setiap paket model berisi URI Amazon S3 ke file model yang terkait dengan model terlatih dan URI ECR Amazon yang menunjuk ke wadah yang digunakan saat menyajikan model.

T. Bagaimana cara menandai paket model di Registry SageMaker Model?

Paket model dalam Registri SageMaker Model tidak mendukung tag—ini adalah paket model berversi. Sebagai gantinya, Anda dapat menambahkan pasangan nilai kunci menggunakan`CustomerMetadataProperties`. Grup paket model dalam penandaan dukungan registri model.

Penerapan Model di SageMaker

Setelah Anda melatih dan menyetujui model untuk produksi, gunakan untuk menyebarkan model Anda SageMaker ke titik akhir untuk inferensi waktu nyata. SageMaker menyediakan beberapa opsi inferensi sehingga Anda dapat memilih opsi yang paling sesuai dengan beban kerja Anda. Anda juga mengonfigurasi titik akhir dengan memilih jenis instans dan jumlah instans yang Anda perlukan untuk kinerja optimal. Untuk detail tentang penerapan model, lihatMenyebarkan model untuk inferensi.

Setelah Anda menerapkan model Anda ke produksi, Anda mungkin ingin mencari cara untuk lebih mengoptimalkan kinerja model sambil mempertahankan ketersediaan model Anda saat ini. Misalnya, Anda dapat menyiapkan tes bayangan untuk mencoba model atau model yang berbeda yang melayani infrastruktur sebelum melakukan perubahan. SageMaker menerapkan model, wadah, atau instance baru dalam mode bayangan dan merutekan salinan permintaan inferensi secara real time dalam titik akhir yang sama. Anda dapat mencatat respons varian bayangan untuk perbandingan. Untuk detail tentang pengujian bayangan, lihatTes Bayangan. Jika Anda memutuskan untuk melanjutkan dan mengubah model Anda, pagar pembatas penerapan membantu Anda mengontrol sakelar dari model saat ini ke model baru. Anda dapat memilih metode seperti pengujian biru/hijau atau kenari dari proses pergeseran lalu lintas untuk mempertahankan kontrol granular selama pembaruan. Untuk informasi tentang pagar pembatas penerapan, lihat. Perbarui model dalam produksi

SageMaker Model Monitor

Setelah model diproduksi, Anda dapat memantau kinerjanya secara real time dengan Amazon SageMaker Model Monitor. Model Monitor membantu Anda mempertahankan kualitas model dengan mendeteksi pelanggaran ambang batas yang ditentukan pengguna untuk kualitas data, kualitas model, penyimpangan bias, dan penyimpangan atribusi fitur. Selain itu, Anda dapat mengonfigurasi peringatan sehingga Anda dapat memecahkan masalah pelanggaran saat muncul dan segera memulai pelatihan ulang. Model Monitor terintegrasi dengan SageMaker Clarify untuk meningkatkan visibilitas ke bias potensial.

Untuk mempelajari tentang SageMaker Model Monitor, lihatMemantau data dan kualitas model.

Otomatiskan MLOP dengan Proyek SageMaker

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan aplikasi Studio Classic. Untuk informasi tentang menggunakan pengalaman Studio yang diperbarui, lihat SageMaker Studio Amazon.

Buat solusi end-to-end ML dengan CI/CD dengan menggunakan SageMaker proyek.

Gunakan SageMaker proyek untuk membuat solusi MLOP untuk mengatur dan mengelola:

- Membangun gambar khusus untuk pemrosesan, pelatihan, dan inferensi
- Persiapan data dan rekayasa fitur
- Model pelatihan
- Mengevaluasi model
- Menyebarkan model
- Memantau dan memperbarui model

Topik

Apa itu SageMaker Proyek?

SageMaker Proyek membantu organisasi mengatur dan menstandarisasi lingkungan pengembang untuk ilmuwan data dan sistem CI/CD untuk insinyur MLOP. Proyek juga membantu organisasi mengatur manajemen ketergantungan, manajemen repositori kode, membangun reproduktifitas, dan berbagi artefak.

Anda dapat menyediakan SageMaker Projects dari AWS Service Catalog menggunakan template kustom atau SageMaker -provided. Untuk informasi tentang AWS Service Catalog, lihat Apa itu AWS Service Catalog. Dengan SageMaker Proyek, insinyur MLOP dan admin organisasi dapat menentukan templat mereka sendiri atau menggunakan templat yang disediakan. SageMaker Template yang SageMaker disediakan mem-bootstrap alur kerja HTML dengan kontrol versi sumber, saluran pipa HTML otomatis, dan satu set kode untuk dengan cepat mulai mengulangi kasus penggunaan ML.

Kapan Anda Harus Menggunakan SageMaker Proyek?

Meskipun notebook sangat membantu untuk pembuatan model dan eksperimen, tim ilmuwan data dan insinyur ML yang berbagi kode membutuhkan cara yang lebih skalabel untuk mempertahankan konsistensi kode dan kontrol versi yang ketat.

Setiap organisasi memiliki standar dan praktik tersendiri yang memberikan keamanan dan tata kelola bagi AWS lingkungannya. SageMaker menyediakan satu set template pihak pertama untuk organisasi yang ingin cepat memulai dengan alur kerja ML dan CI/CD. Template mencakup proyek yang menggunakan layanan AWS -native untuk CI/CD, sepertiAWS CodeBuild,, dan. AWS CodePipeline AWS CodeCommit Template juga menawarkan opsi untuk membuat proyek yang menggunakan alat pihak ketiga, seperti Jenkins dan GitHub. Untuk daftar templat proyek yang SageMaker menyediakan, lihatGunakan SageMaker Template Proyek yang Disediakan.

Organizations sering membutuhkan kontrol ketat atas sumber daya MLOP yang mereka sediakan dan kelola. Tanggung jawab tersebut mengasumsikan tugas-tugas tertentu, termasuk mengonfigurasi peran dan kebijakan IAM, menegakkan tag sumber daya, menegakkan enkripsi, dan memisahkan sumber daya di beberapa akun. SageMaker Proyek dapat mendukung semua tugas ini melalui penawaran template kustom di mana organisasi menggunakan AWS CloudFormation template untuk menentukan sumber daya yang diperlukan untuk alur kerja ML. Ilmuwan Data dapat memilih template untuk bootstrap dan pra-konfigurasi alur kerja ML mereka. Template kustom ini dibuat sebagai produk Service Catalog dan Anda dapat menyediakannya di UI Studio Classic di bawah Template Organisasi. Service Catalog adalah layanan yang membantu organisasi membuat dan mengelola

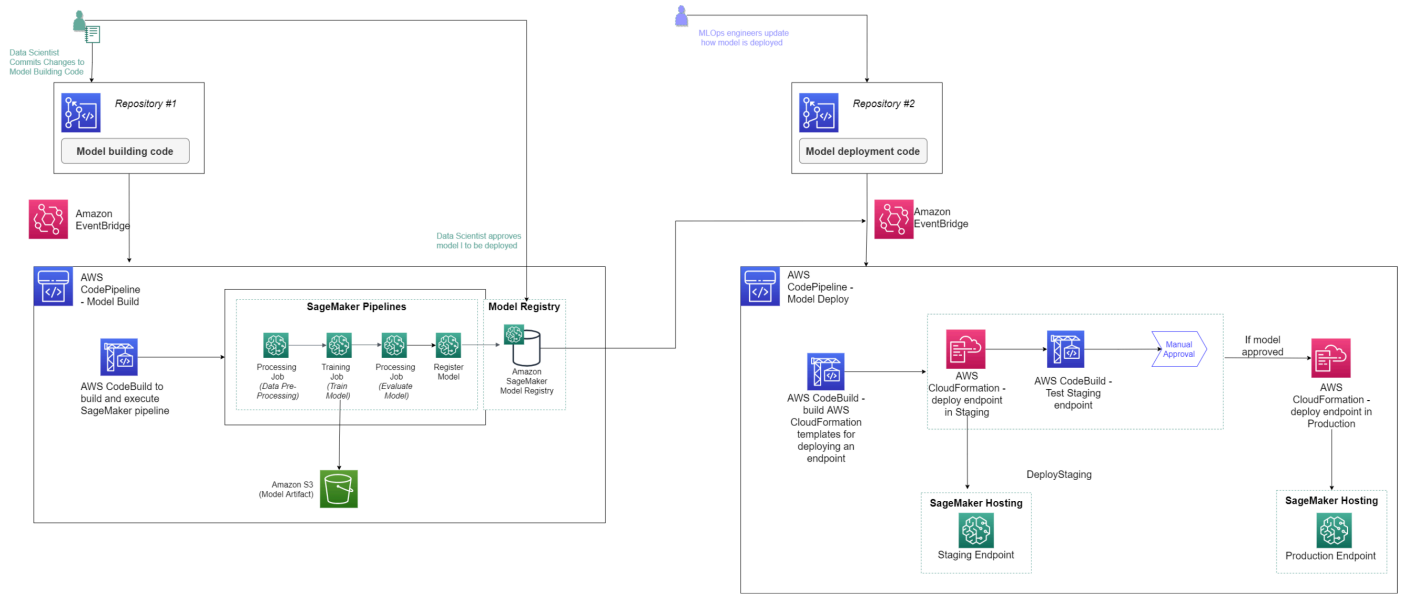
katalog produk yang disetujui untuk digunakan pada. AWS Untuk informasi selengkapnya tentang membuat templat kustom, lihat Membangun Template SageMaker Proyek Kustom — Praktik Terbaik.

SageMaker Proyek dapat membantu Anda mengelola repositori Git Anda sehingga Anda dapat berkolaborasi lebih efisien di seluruh tim, memastikan konsistensi kode, dan mendukung CI/CD. SageMaker Proyek dapat membantu Anda dengan tugas-tugas berikut:

- Atur semua entitas siklus hidup ML di bawah satu proyek.
- Menetapkan pendekatan satu klik untuk menyiapkan infrastruktur standar ML untuk pelatihan model dan penerapan yang menggabungkan praktik terbaik.
- Membuat dan berbagi template untuk infrastruktur ML untuk melayani beberapa kasus penggunaan.
- Manfaatkan templat pra-bangun yang SageMaker disediakan untuk mulai berfokus pada pembuatan model dengan cepat, atau membuat templat khusus dengan sumber daya dan pedoman khusus organisasi.
- Integrasikan dengan alat pilihan Anda dengan memperluas templat proyek. Sebagai contoh, lihat Membuat SageMaker Proyek untuk diintegrasikan dengan GitLab dan GitLab Pipelines.
- Atur semua entitas siklus hidup ML di bawah satu proyek.

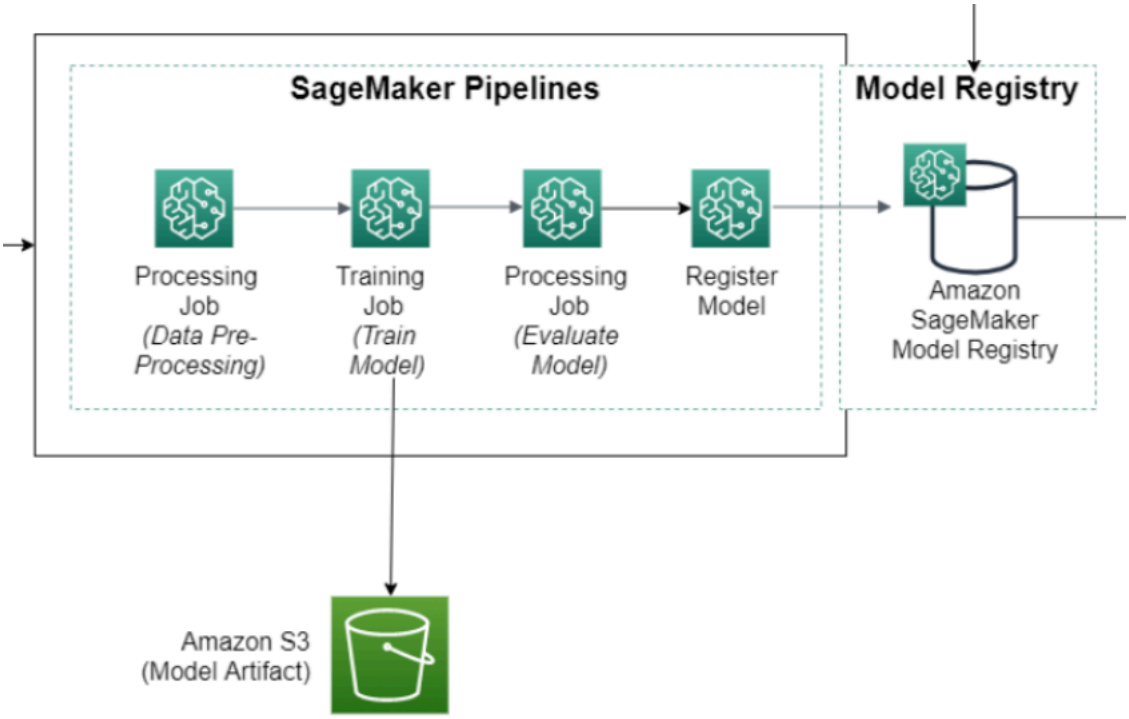
Apa yang ada dalam sebuah SageMaker proyek?

Pelanggan memiliki fleksibilitas untuk menyiapkan proyek mereka dengan sumber daya yang paling sesuai dengan kasus penggunaan mereka. Contoh di bawah ini menampilkan penyiapan MLOP untuk alur kerja ML, termasuk pelatihan model dan penerapan.



Sebuah proyek tipikal dengan template SageMaker -provided mungkin termasuk yang berikut:

- Satu atau lebih repositori dengan kode sampel untuk membangun dan menerapkan solusi ML. Ini adalah contoh kerja yang dapat Anda kloning secara lokal dan memodifikasi untuk kebutuhan Anda. Anda memiliki kode ini dan dapat memanfaatkan repositori yang dikendalikan versi untuk tugas Anda.
- SageMaker Pipeline yang mendefinisikan langkah-langkah untuk persiapan data, pelatihan, evaluasi model, dan penyebaran model, seperti yang ditunjukkan pada diagram berikut.



- Pipeline CodePipeline atau Jenkins yang menjalankan SageMaker pipeline Anda setiap kali Anda memeriksa versi kode yang baru. Untuk informasi tentang CodePipeline, lihat Apa itu AWS CodePipeline. Untuk informasi tentang Jenkins, lihat Dokumentasi Pengguna Jenkins.
- Grup model yang berisi versi model. Setiap kali Anda menyetujui versi model yang dihasilkan dari SageMaker proses pipeline, Anda dapat menerapkannya ke titik akhir. SageMaker

Setiap SageMaker proyek memiliki nama dan ID unik yang diterapkan sebagai tag untuk semua SageMaker dan AWS sumber daya yang dibuat dalam proyek. Dengan nama dan ID, Anda dapat melihat semua entitas yang terkait dengan proyek Anda. Ini termasuk:

- Alur
- Model Model
- Model yang diterapkan (titik akhir)
- Set Data
- Produk Service Catalog
- CodePipeline dan jaringan pipa Jenkins
- CodeCommit dan repositori Git pihak ketiga

Apakah Saya Perlu Membuat Proyek untuk Menggunakan SageMaker Pipelines?

Tidak. SageMaker pipeline adalah entitas mandiri seperti pekerjaan pelatihan, pekerjaan pemrosesan, dan pekerjaan lainnya SageMaker . Anda dapat membuat, memperbarui, dan menjalankan pipeline langsung di dalam notebook dengan menggunakan SageMaker Python SDK tanpa menggunakan proyek. SageMaker

Proyek menyediakan lapisan tambahan untuk membantu Anda mengatur kode Anda dan mengadopsi praktik terbaik operasional yang Anda butuhkan untuk sistem kualitas produksi.

SageMaker Izin Studio yang Diperlukan untuk Menggunakan Proyek

Pengguna dapat melihat templat proyek yang SageMaker disediakan dan membuat proyek dengan templat tersebut saat Anda memberikan izin Proyek untuk pengguna. Anda dapat memberikan izin ini saat Anda melakukan onboard atau memperbarui Amazon SageMaker Studio Classic. Ada dua izin untuk diberikan.

1. Berikan izin Proyek untuk administrator Studio Classic untuk mengizinkan administrator Studio Classic melihat templat SageMaker yang disediakan di konsol Service Catalog. Administrator dapat melihat apa yang dibuat pengguna Studio Classic lainnya jika Anda memberi mereka izin untuk menggunakan SageMaker proyek. Administrator juga dapat melihat AWS CloudFormation template yang ditentukan oleh template proyek SageMaker yang disediakan di konsol Service Catalog. Untuk informasi tentang penggunaan konsol Service Catalog, lihat Apa itu Service Catalog di Panduan Pengguna Service Catalog.
2. Izinkan pengguna Studio Classic yang dikonfigurasi untuk menggunakan peran eksekusi yang sama dengan domain untuk membuat proyek. Ini memberi pengguna Studio Classic izin untuk menggunakan templat proyek SageMaker yang disediakan untuk membuat proyek dari dalam Studio Classic.

> **Important**
>
> Jangan membuat peran Anda secara manual. Selalu buat peran melalui Pengaturan Studio Classic menggunakan langkah-langkah yang dijelaskan dalam prosedur berikut.

Untuk pengguna yang menggunakan peran apa pun selain peran eksekusi domain untuk melihat dan menggunakan templat proyek SageMaker yang disediakan, Anda harus memberikan izin Proyek ke masing-masing profil pengguna.

Prosedur berikut menunjukkan cara memberikan izin Proyek setelah Anda onboard ke Studio Classic. Untuk informasi selengkapnya tentang orientasi ke Studio Classic, lihatIkhtisar SageMaker Domain Amazon.

Untuk memberikan izin Proyek bagi administrator dan pengguna peran eksekusi domain

1. Buka konsol SageMaker .
2. Di panel navigasi kiri, pilih Konfigurasi admin.
3. Di bawah konfigurasi Admin, pilih Domain.
4. Pilih Create domain (Buat domain).
5. Jika Anda memilih Penyiapan cepat untuk menyiapkan SageMaker Domain, Anda memiliki izin untuk menggunakan templat proyek secara default.
6. Jika Anda memilih Pengaturan standar untuk mengatur SageMaker Domain, pastikan Anda mengaktifkan opsi berikut saat mengonfigurasi setelan Studio Classic:
   - Aktifkan templat SageMaker proyek Amazon dan Amazon SageMaker JumpStart untuk akun ini
   - Aktifkan templat SageMaker proyek Amazon dan Amazon SageMaker JumpStart untuk pengguna Studio Classic
7. Untuk mengonfirmasi bahwa SageMaker Domain Anda memiliki izin template proyek yang aktif:
   a. Buka konsol SageMaker .
   b. Di panel navigasi kiri, pilih Konfigurasi admin.
   c. Di bawah konfigurasi Admin, pilih Domain.
   d. Pilih domain Anda.
   e. Pilih tab Pengaturan Domain.
   f. Di bawah SageMaker Proyek dan JumpStart, pastikan opsi berikut diaktifkan:
      - Aktifkan templat SageMaker proyek Amazon dan Amazon SageMaker JumpStart untuk akun ini
      - Aktifkan templat SageMaker proyek Amazon dan Amazon SageMaker JumpStart untuk pengguna Studio Classic
8. Untuk menampilkan daftar peran Anda:
   a. Buka konsol SageMaker .

b. Di panel navigasi kiri, pilih Konfigurasi admin.

c. Di bawah konfigurasi Admin, pilih Domain.

d. Pilih domain Anda.

e. Pilih tab Pengaturan Domain.

f. Daftar peran Anda muncul di `Apps` kartu di bawah tab Studio.

> **Important**
>
> Mulai 25 Juli, kami memerlukan peran tambahan untuk menggunakan templat proyek. Berikut adalah daftar lengkap peran yang harus Anda lihat di bawah`Projects`:
> `AmazonSageMakerServiceCatalogProductsLaunchRole`
> `AmazonSageMakerServiceCatalogProductsUseRole`
> `AmazonSageMakerServiceCatalogProductsApiGatewayRole`
> `AmazonSageMakerServiceCatalogProductsCloudformationRole`
> `AmazonSageMakerServiceCatalogProductsCodeBuildRole`
> `AmazonSageMakerServiceCatalogProductsCodePipelineRole`
> `AmazonSageMakerServiceCatalogProductsEventsRole`
> `AmazonSageMakerServiceCatalogProductsFirehoseRole`
> `AmazonSageMakerServiceCatalogProductsGlueRole`
> `AmazonSageMakerServiceCatalogProductsLambdaRole`
> `AmazonSageMakerServiceCatalogProductsExecutionRole`
> Untuk deskripsi peran ini, lihatAWSKebijakan Terkelola untuk SageMaker proyek dan JumpStart.

Membuat Proyek MLOPS menggunakan Amazon Studio Classic SageMaker

Prosedur ini menunjukkan cara membuat proyek MLOPS menggunakan Amazon SageMaker Studio Classic.

Prasyarat

- Akun IAM atau Pusat Identitas IAM untuk masuk ke Studio Classic. Untuk informasi, lihat Ikhtisar SageMaker Domain Amazon.

- Izin untuk menggunakan templat proyek SageMaker yang disediakan. Untuk informasi, lihat SageMaker Izin Studio yang Diperlukan untuk Menggunakan Proyek.
- Keakraban dasar dengan antarmuka pengguna Studio Classic. Untuk informasi, lihat Ikhtisar UI Amazon SageMaker Studio Classic.

Untuk membuat proyek di Studio Classic

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di sidebar Studio Classic, pilih ikon Beranda ().
3. Pilih Deployment dari menu, lalu pilih Projects.
4. Pilih Buat proyek.

   Tab Create project terbuka menampilkan daftar template yang tersedia.
5. Jika belum dipilih, pilih SageMaker template. Untuk informasi selengkapnya tentang templat proyek, lihatTemplat Proyek MLOPS.
6. Pilih template MLOPs untuk pembuatan model, pelatihan, dan penerapan.
7. Pilih Pilih template proyek.

   Tab Create project berubah untuk menampilkan detail Project.
8. Masukkan informasi berikut:

   - Untuk detail proyek, masukkan nama dan deskripsi untuk proyek Anda.
   - Opsional, tambahkan tag, yang merupakan pasangan nilai kunci yang dapat digunakan untuk melacak proyek Anda.
9. Pilih Buat proyek dan tunggu proyek muncul di daftar Proyek.

Templat Proyek MLOPS

Template SageMaker proyek Amazon mengotomatiskan penyiapan dan implementasi MLOP untuk proyek Anda. Template SageMaker proyek adalah produk Service Catalog yang SageMaker tersedia untuk pengguna Amazon SageMaker Studio Classic. Produk Service Catalog ini akan terlihat di konsol Service Catalog setelah mengaktifkan izin saat Anda melakukan onboard atau memperbarui Amazon SageMaker Studio Classic. Untuk informasi tentang mengaktifkan izin untuk menggunakan

templat SageMaker proyek, lihat. SageMaker Izin Studio yang Diperlukan untuk Menggunakan Proyek Gunakan template SageMaker proyek untuk membuat proyek yang merupakan solusi end-to-end MLOPs.

Jika Anda seorang administrator, Anda dapat membuat templat proyek khusus dari awal atau memodifikasi salah satu templat proyek yang disediakan oleh SageMaker. Pengguna Studio Classic di organisasi Anda dapat menggunakan templat proyek khusus ini untuk membuat proyek mereka.

Topik

- Gunakan SageMaker Template Proyek yang Disediakan
- Buat Template Proyek Kustom

Gunakan SageMaker Template Proyek yang Disediakan

Amazon SageMaker menyediakan templat proyek yang membuat infrastruktur yang Anda butuhkan untuk membuat solusi MLOP untuk integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD) dari model ML. Gunakan templat ini untuk memproses data, mengekstrak fitur, melatih dan menguji model, mendaftarkan model dalam registri SageMaker model, dan menerapkan model untuk inferensi. Anda dapat menyesuaikan kode benih dan file konfigurasi agar sesuai dengan kebutuhan Anda.

> **Important**
>
> Mulai 25 Juli 2022, kami memerlukan peran tambahan untuk menggunakan templat proyek. Untuk daftar lengkap peran dan instruksi yang diperlukan tentang cara membuatnya, lihatSageMaker Izin Studio yang Diperlukan untuk Menggunakan Proyek. Jika Anda tidak memiliki peran baru, Anda akan mendapatkan pesan CodePipeline kesalahan tidak diizinkan untuk dilakukan AssumeRole pada peran arn:aws:iam: :xxx:role/service-role/AmazonSageMakerServiceCatalogProductsCode PipelineRole ketika Anda mencoba membuat proyek baru dan tidak dapat melanjutkan.

SageMaker templat proyek menawarkan pilihan repositori kode, alat otomatisasi alur kerja, dan tahapan pipeline berikut:

- Repositori kode: AWS CodeCommit atau repositori Git pihak ketiga seperti dan Bitbucket GitHub
- Otomatisasi alur kerja CI/CD: atau Jenkins AWS CodePipeline

- Tahapan pipa: Pembuatan model dan pelatihan, penerapan model, atau keduanya

Diskusi berikut memberikan gambaran umum dari setiap template yang dapat Anda pilih saat Anda membuat SageMaker proyek Anda. Anda juga dapat melihat template yang tersedia di Studio Classic dengan mengikuti Langkah 1: Buat panduan Proyek Proyek.

Untuk step-by-step petunjuk tentang cara membuat proyek nyata, Anda dapat mengikuti salah satu panduan proyek:

- Jika Anda ingin menggunakan templateTemplate MLOPs untuk pembuatan model, pelatihan, dan penyebaran, lihatSageMaker Panduan Proyek MLOPs.
- Jika Anda ingin menggunakan templateTemplate MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga menggunakan CodePipeline, lihatSageMaker Panduan Proyek MLOPS Menggunakan Repos Git Pihak Ketiga.
- Jika Anda ingin menggunakan templateTemplate MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga menggunakan Jenkins, lihat Membuat SageMaker proyek Amazon menggunakan kontrol sumber pihak ketiga dan Jenkins.

Topik

- Template MLOPs untuk pembuatan model, pelatihan, dan penyebaran
- Template MLOPs untuk pembuatan model, pelatihan, penyebaran, dan Amazon Model Monitor SageMaker
- Template MLOPs untuk pembuatan gambar, pembuatan model, dan penerapan model
- Template MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga menggunakan CodePipeline
- Template MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga menggunakan Jenkins
- Penyebaran model untuk Salesforce
- Perbarui SageMaker Proyek untuk Menggunakan Repositori Git Pihak Ketiga

Template MLOPs untuk pembuatan model, pelatihan, dan penyebaran

Template ini adalah kombinasi dari dua template berikut, yang masing-masing dapat digunakan secara independen, dan berisi semua sumber daya yang disediakan dalam template tersebut.

- Repositori kode: AWS CodeCommit
- Otomatisasi alur kerja CI/CD: AWS CodePipeline

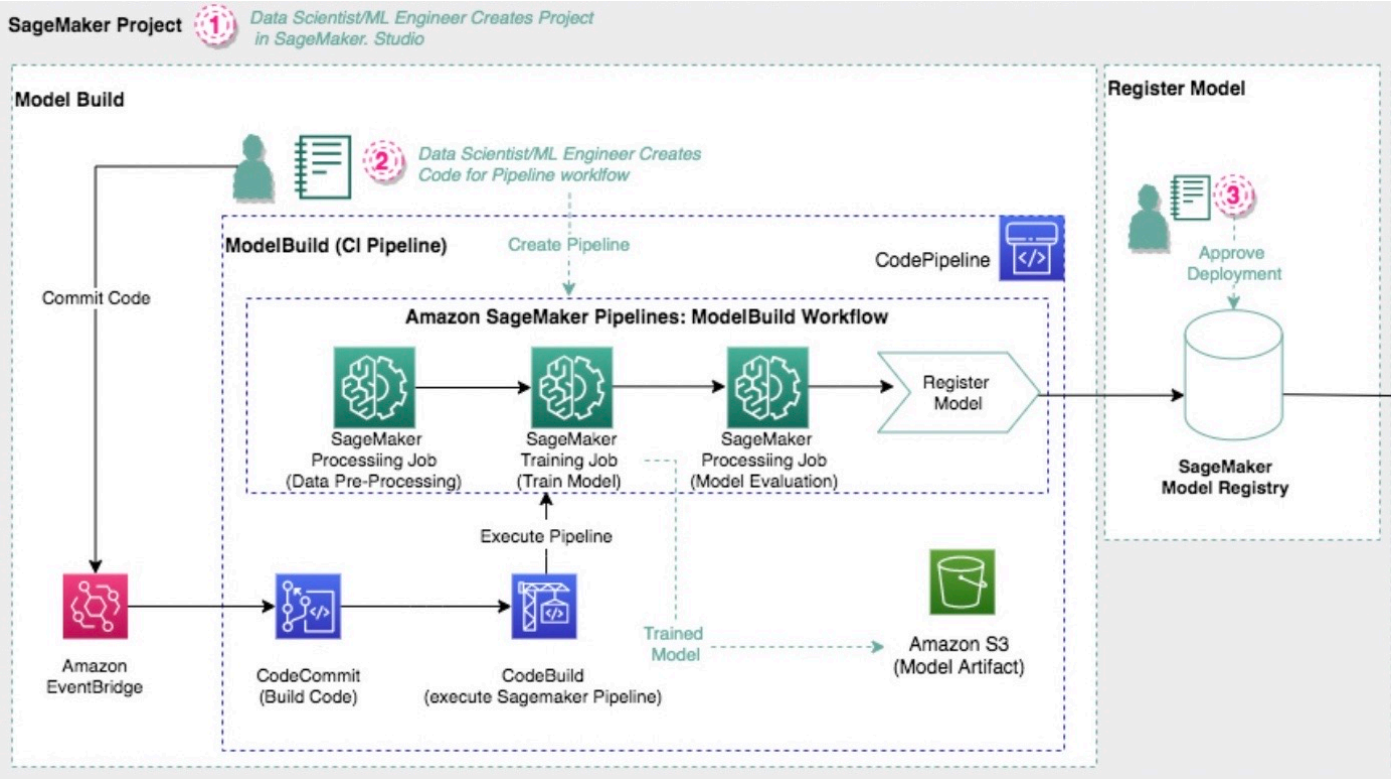
Template MLOPs untuk pembuatan model dan pelatihan

Gunakan template ini saat Anda menginginkan solusi MLOPS untuk memproses data, mengekstrak fitur, melatih dan menguji model, dan mendaftarkan model dalam registri SageMaker model.

Templat ini menyediakan sumber daya berikut:

- AWS CodeCommitRepositori yang berisi kode sampel yang membuat SageMaker pipeline Amazon dalam kode Python dan menunjukkan cara membuat dan memperbarui pipeline. SageMaker Repositori ini juga memiliki contoh notebook Python yang dapat Anda buka dan jalankan di Studio Classic.
- AWS CodePipelinePipeline yang memiliki sumber dan langkah build. Langkah sumber menunjuk ke CodeCommit repositori. Langkah build mendapatkan kode dari repositori itu, membuat dan memperbarui SageMaker pipeline, memulai eksekusi pipeline, dan menunggu eksekusi pipeline selesai.
- Bucket Amazon S3 untuk menyimpan artefak, termasuk CodePipeline dan artefak, dan CodeBuild artefak apa pun yang dihasilkan dari pipa berjalan. SageMaker

Diagram berikut menggambarkan alur kerja dan AWS sumber daya yang digunakan oleh template ini untuk membantu Anda membangun dan melatih model Anda.



Template MLOPs untuk penerapan model

Gunakan template ini untuk mengotomatiskan penerapan model dalam registri SageMaker model ke SageMaker titik akhir untuk inferensi real-time. Template ini mengenali perubahan dalam registri model. Ketika versi model baru terdaftar dan disetujui, secara otomatis memulai penerapan.

Template menyediakan CodeCommit repositori dengan file konfigurasi untuk menentukan langkah penerapan model, AWS CloudFormation template untuk mendefinisikan titik akhir sebagai infrastruktur, dan kode benih untuk menguji titik akhir.

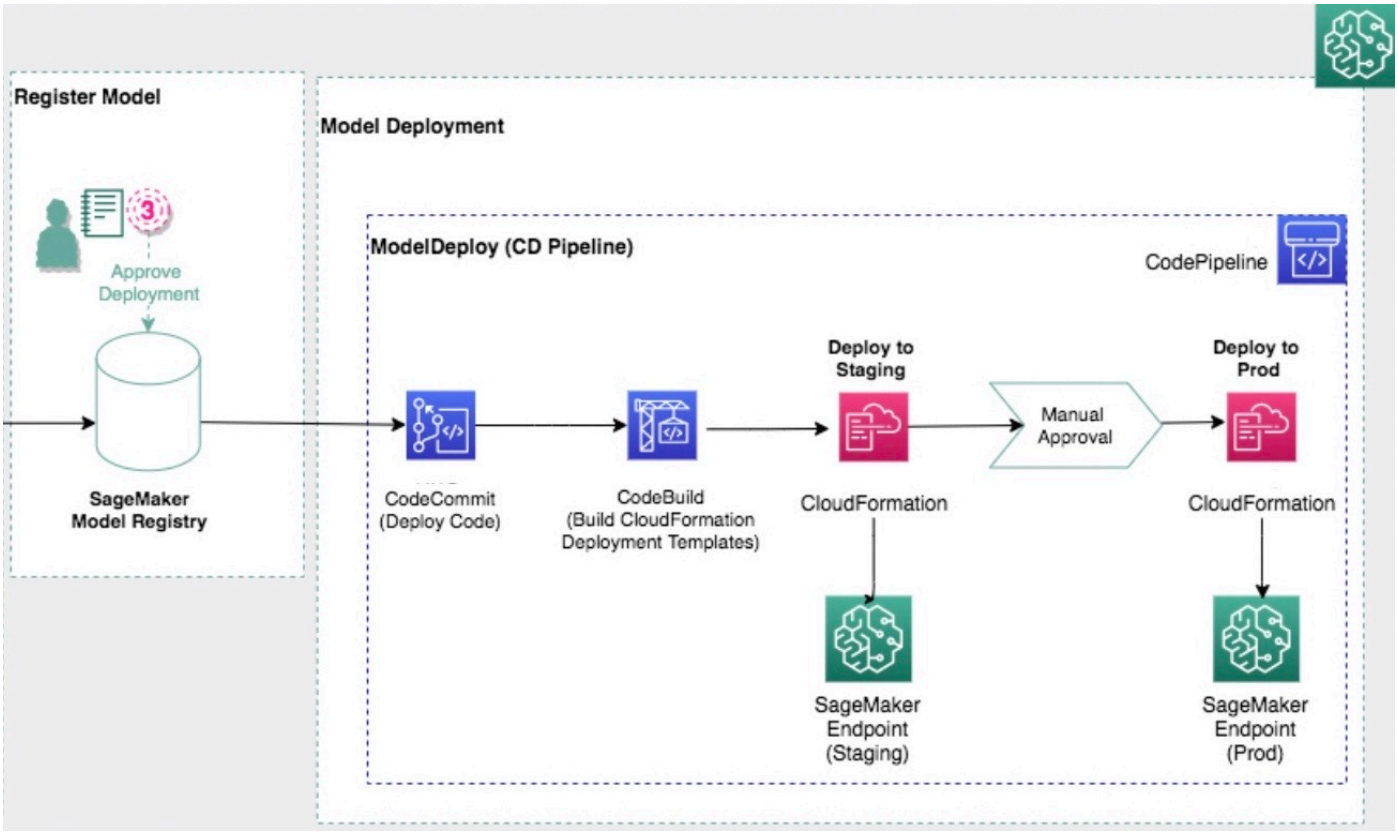
Templat ini menyediakan sumber daya berikut:

- AWS CodeCommitRepositori yang berisi kode sampel yang menyebarkan model ke titik akhir di lingkungan pementasan dan produksi.
- AWS CodePipelinePipeline yang memiliki sumber, build deploy-to-staging, dan deploy-to-production langkah-langkah. Langkah sumber menunjuk ke CodeCommit repositori, dan langkah build mendapatkan kode dari repositori itu dan menghasilkan CloudFormation tumpukan untuk diterapkan. deploy-to-production Langkah-langkah deploy-to-staging dan menyebarkan CloudFormation tumpukan ke lingkungan masing-masing. Ada langkah persetujuan manual antara langkah pementasan dan pembuatan produksi, sehingga insinyur MLOP harus menyetujui model sebelum dikerahkan ke produksi.

Ada juga langkah persetujuan terprogram dengan tes placeholder dalam kode contoh di repositori. CodeCommit Anda dapat menambahkan tes tambahan untuk mengganti tes placeholder.

- Bucket Amazon S3 untuk menyimpan artefak, termasuk CodePipeline dan artefak, dan CodeBuild artefak apa pun yang dihasilkan dari pipa berjalan. SageMaker
- CloudWatch Peristiwa untuk memulai pipeline saat versi paket model disetujui atau ditolak.

Diagram berikut mengilustrasikan alur kerja dan AWS sumber daya yang digunakan oleh template ini untuk membantu Anda menerapkan model Anda.



Seperti disebutkan sebelumnya, lihat Project Walkthrough untuk demonstrasi yang menggunakan template ini untuk membuat proyek nyata.

Template MLOPs untuk pembuatan model, pelatihan, penyebaran, dan Amazon Model Monitor SageMaker

Template ini merupakan perpanjangan dari template MLOPs untuk pembuatan model, pelatihan, dan penyebaran. Ini mencakup komponen pembuatan model, pelatihan, dan penerapan template, dan template Amazon SageMaker Model Monitor tambahan yang menyediakan jenis pemantauan berikut:

- Kualitas Data - Memantau penyimpangan dalam kualitas data.
- Kualitas Model — Pantau penyimpangan dalam metrik kualitas model, seperti akurasi.
- Bias Drift untuk Model dalam Produksi — Pantau bias dalam prediksi model.

- Repositori kode: AWS CodeCommit
- Otomatisasi alur kerja CI/CD: AWS CodePipeline

Template MLOPs untuk Amazon Model Monitor SageMaker

Anda dapat menggunakan template ini untuk solusi MLOPs guna menerapkan satu atau beberapa monitor kualitas SageMaker data Amazon, kualitas model, bias model, dan kemampuan penjelasan model untuk memantau model yang diterapkan pada titik akhir inferensi. SageMaker

Templat ini menyediakan sumber daya berikut:

- AWS CodeCommit Repositori yang berisi contoh kode Python yang mendapatkan garis dasar yang digunakan oleh monitor dari Registry SageMaker Model, dan memperbarui parameter template untuk lingkungan pementasan dan produksi. Ini juga berisi AWS CloudFormation template untuk membuat Monitor SageMaker Model Amazon.
- AWS CodePipeline Pipeline yang memiliki langkah sumber, pembuatan, dan penerapan. Langkah sumber menunjuk ke CodePipeline repositori. Langkah build mendapatkan kode dari repositori itu, mendapatkan baseline dari Model Registry, dan memperbarui parameter template untuk lingkungan pementasan dan produksi. Langkah-langkah penerapan menyebarkan monitor yang dikonfigurasi ke dalam lingkungan pementasan dan produksi. Langkah persetujuan manual, dalam `DeployStaging` tahap, mengharuskan Anda untuk memverifikasi bahwa SageMaker titik akhir produksi `InService` sebelum menyetujui dan pindah ke tahap. `DeployProd`
- Template menggunakan bucket S3 yang sama yang dibuat oleh template MLOPs untuk pembuatan model, pelatihan, dan penerapan untuk menyimpan output monitor.
- Dua aturan EventBridge peristiwa Amazon memulai Monitor SageMaker Model Amazon AWS CodePipeline setiap kali SageMaker titik akhir pementasan diperbarui, atau perubahan kode dilakukan ke repositori. CodePipeline

Template MLOPs untuk pembuatan gambar, pembuatan model, dan penerapan model

Template ini merupakan perpanjangan dariTemplate MLOPs untuk pembuatan model, pelatihan, dan penyebaran. Ini mencakup komponen pembuatan model, pelatihan, dan penerapan templat itu dan opsi berikut:

- Sertakan pemrosesan gambar—membangun pipa
- Sertakan gambar pelatihan — membangun pipa
- Sertakan gambar inferensi — membangun pipa

Untuk setiap komponen yang dipilih selama pembuatan proyek, berikut ini dibuat dengan menggunakan template:

- Repositori Amazon ECR
- Sebuah SageMaker Gambar
- CodeCommit Repositori yang berisi Dockerfile yang dapat Anda sesuaikan
- A CodePipeline yang diprakarsai oleh perubahan pada repositori CodePipeline
- CodeBuild Proyek yang membangun image Docker dan mendaftarkannya di repositori Amazon ECR
- EventBridge Aturan yang memulai CodePipeline jadwal

Ketika dimulai, ia membangun wadah Docker baru dan mendaftarkannya dengan repositori Amazon ECR. CodePipeline Ketika wadah baru terdaftar dengan repositori Amazon ECR, yang baru `ImageVersion` ditambahkan ke gambar. SageMaker Ini memulai pipa pembangunan model, yang pada gilirannya memulai pipa penyebaran.

Gambar yang baru dibuat digunakan di bagian pembuatan model, pelatihan, dan penerapan alur kerja jika berlaku.

Template MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga menggunakan CodePipeline

- Repositori kode: Git Pihak Ketiga. Buat AWS CodeStar koneksi dari AWS akun Anda ke GitHub pengguna atau organisasi Anda. Tambahkan tag dengan kunci `sagemaker` dan nilai `true` ke AWS CodeStar koneksi ini.
- Otomatisasi alur kerja CI/CD: AWS CodePipeline

Templat ini menyediakan sumber daya berikut:

- Asosiasi dengan satu atau lebih repositori Git yang ditentukan pelanggan.
- AWS CodePipelinePipeline yang memiliki sumber, build deploy-to-staging, dan deploy-to-production langkah-langkah. Langkah sumber menunjuk ke repositori Git pihak ketiga dan langkah build mendapatkan kode dari repositori itu dan menghasilkan CloudFormation tumpukan untuk diterapkan. deploy-to-production Langkah-langkah deploy-to-staging dan menyebarkan CloudFormation tumpukan ke lingkungan masing-masing. Ada langkah persetujuan manual antara langkah pementasan dan pembuatan produksi, sehingga insinyur MLOP harus menyetujui model sebelum dikerahkan ke produksi.
- Sebuah AWS CodeBuild proyek untuk mengisi repositori Git dengan informasi kode benih. Ini memerlukan AWS CodeStar koneksi dari AWS akun Anda ke akun Anda di host repositori Git.
- Bucket Amazon S3 untuk menyimpan artefak, termasuk CodePipeline dan artefak, dan CodeBuild artefak apa pun yang dihasilkan dari pipa berjalan. SageMaker

Seperti disebutkan sebelumnya, lihat Panduan Proyek Menggunakan Repo Git Pihak Ketiga untuk demonstrasi yang menggunakan template ini untuk membuat proyek nyata.

Template MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga menggunakan Jenkins

- Repositori kode: Git Pihak Ketiga. Buat AWS CodeStar koneksi dari AWS akun Anda ke GitHub pengguna atau organisasi Anda. Tambahkan tag dengan kunci `sagemaker` dan nilai `true` ke AWS CodeStar koneksi ini.
- Otomatisasi alur kerja CI/CD: Jenkins

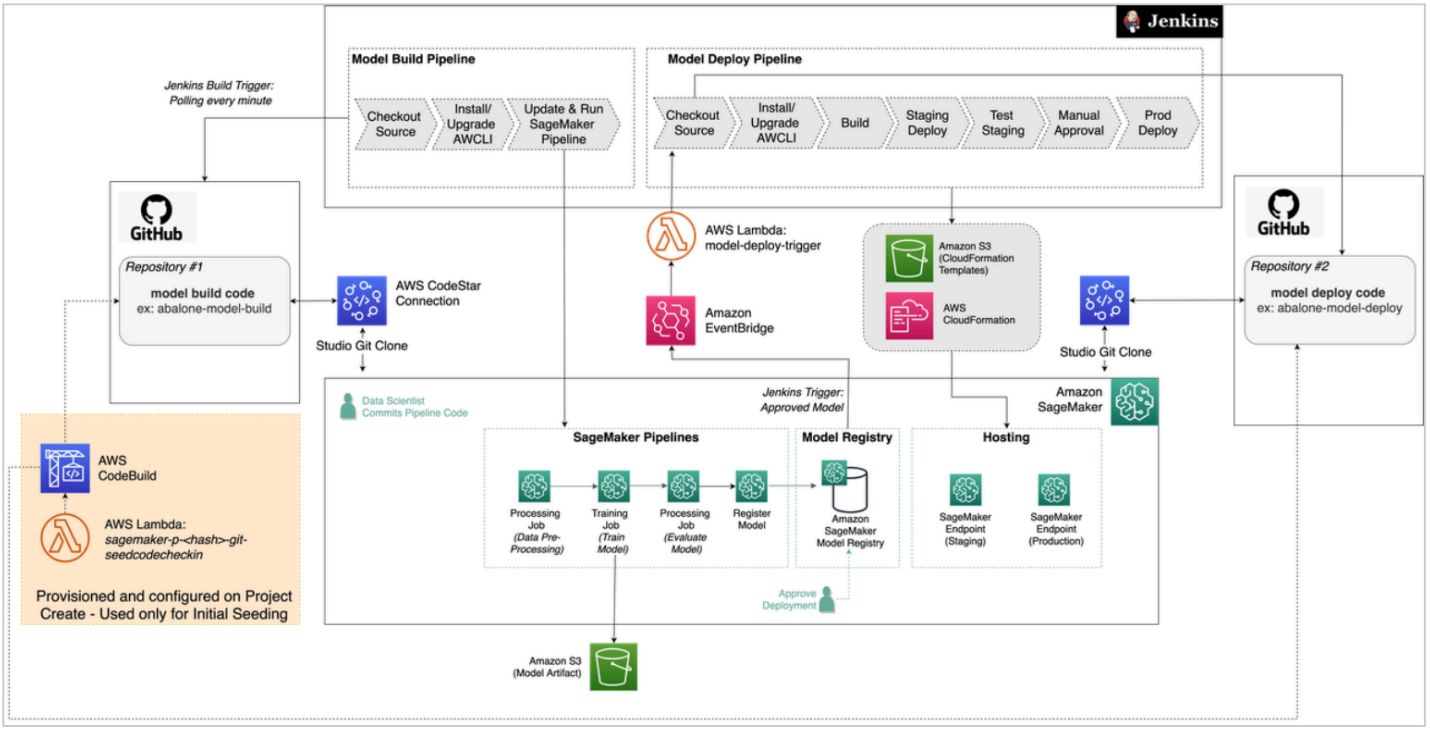
Templat ini menyediakan sumber daya berikut:

- Asosiasi dengan satu atau lebih repositori Git yang ditentukan pelanggan.
- Seed code untuk menghasilkan pipeline Jenkins yang memiliki sumber, build deploy-to-staging, dan deploy-to-production langkah-langkah. Langkah sumber menunjuk ke repositori Git yang ditentukan pelanggan. Langkah build mendapatkan kode dari repositori itu dan menghasilkan dua CloudFormation tumpukan. Langkah-langkah penerapan menyebarkan CloudFormation tumpukan ke lingkungan masing-masing. Ada langkah persetujuan antara langkah pementasan dan langkah produksi.

- Sebuah AWS CodeBuild proyek untuk mengisi repositori Git dengan informasi kode benih. Ini memerlukan AWS CodeStar koneksi dari AWS akun Anda ke akun Anda di host repositori Git.
- Bucket Amazon S3 untuk menyimpan artefak SageMaker proyek dan pipa. SageMaker

Template menciptakan hubungan antara proyek Anda dan repositori kontrol sumber, tetapi Anda perlu melakukan langkah-langkah manual tambahan untuk membangun komunikasi antara AWS akun Anda dan Jenkins. Untuk langkah-langkah mendetail, lihat Membuat SageMaker proyek Amazon menggunakan kontrol sumber pihak ketiga dan Jenkins.

Instruksi membantu Anda membangun arsitektur yang ditunjukkan dalam diagram berikut, dengan GitHub sebagai repositori kontrol sumber dalam contoh ini. Seperti yang ditunjukkan, Anda melampirkan repositori Git Anda ke proyek untuk memeriksa dan mengelola versi kode. Jenkins memulai pipeline build model ketika mendeteksi perubahan pada kode build model di repositori Git. Anda juga menghubungkan proyek ke Jenkins untuk mengatur langkah-langkah penerapan model Anda, yang dimulai saat Anda menyetujui model yang terdaftar di registri model, atau ketika Jenkins mendeteksi perubahan pada kode penerapan model.



Singkatnya, langkah-langkah memandu Anda melalui tugas-tugas berikut:

1. Buat hubungan antara akun Anda AWS dan GitHub akun.
2. Buat akun Jenkins dan impor plugin yang diperlukan.

3. Buat kebijakan pengguna dan izin Jenkins IAM.
4. Tetapkan AWS kredensyal untuk pengguna Jenkins IAM di server Jenkins Anda.
5. Buat token API untuk komunikasi dengan server Jenkins Anda.
6. Gunakan CloudFormation templat untuk menyiapkan EventBridge aturan guna memantau registri model untuk model yang baru disetujui.
7. Buat SageMaker proyek, yang menyemai GitHub repositori Anda dengan model build dan deploy code.
8. Buat pipeline build model Jenkins Anda dengan kode benih model build.
9. Buat pipeline penerapan model Jenkins Anda dengan kode benih penerapan model.

Penyebaran model untuk Salesforce

- Repositori kode: AWS CodeCommit
- Otomatisasi alur kerja CI/CD: AWS CodePipeline

Templat ini menyediakan sumber daya berikut:

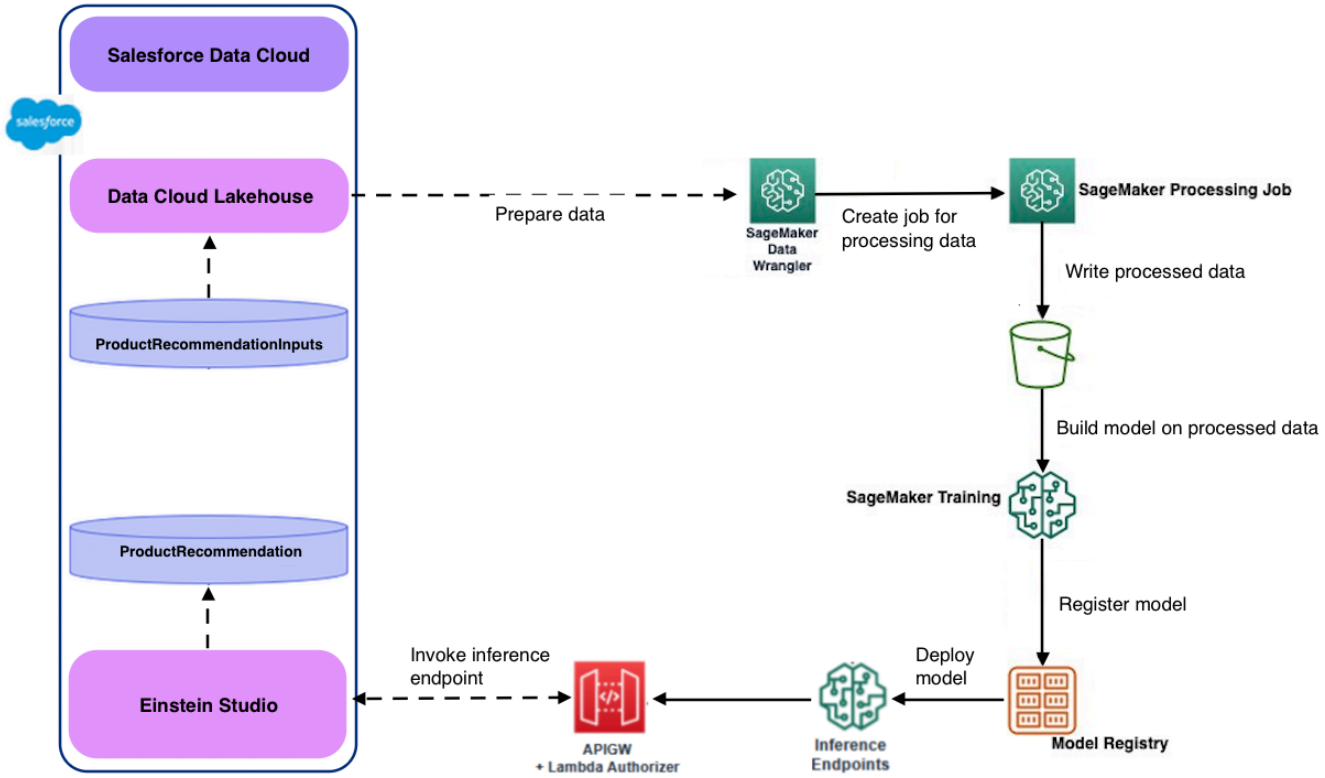
- AWS CodeCommitRepositori yang berisi kode contoh yang membuat SageMaker pipeline Amazon dalam kode Python dan menunjukkan cara membuat dan memperbarui pipeline. Repositori ini juga memiliki Notebook Python Jupyter yang dapat Anda buka dan jalankan di Studio Classic.
- AWS CodePipelinePipeline yang memiliki sumber dan langkah build. Langkah sumber menunjuk ke CodeCommit repositori. Langkah build mendapatkan kode dari repositori, membuat dan memperbarui SageMaker pipeline, memulai proses pipeline, dan menunggu pipeline berjalan selesai.
- Bucket Amazon S3 untuk menyimpan artefak, termasuk CodePipeline dan artefak, dan CodeBuild artefak apa pun yang dihasilkan dari pipa berjalan. SageMaker

Admin Anda mungkin perlu melakukan penyiapan tambahan untuk mengaktifkan akses data dari Salesforce Data Cloud ke SageMaker Studio untuk membangun model AI/ML. Lihat ikhtisar solusi di posting blog Gunakan integrasi Amazon SageMaker dan Salesforce Data Cloud untuk memberi daya pada aplikasi Salesforce Anda dengan AI/HTML untuk informasi dan instruksi terperinci.

Diagram berikut menggambarkan alur kerja tingkat tinggi yang digunakan oleh template ini untuk membantu Anda membangun dan melatih model Anda. Setelah menyiapkan sambungan antara Salesforce Data Cloud ke Data Wrangler dan pra-proses data Anda, gunakan templat proyek

penerapan Model untuk Salesforce untuk mengotomatiskan pelatihan dan penerapan model. Template menyediakan kode penerapan model yang dapat disesuaikan dan notebook contoh AWS CodePipeline untuk melatih model Anda dan mendaftarkannya ke registri model. SageMaker Setelah Anda menyetujui model, titik akhir akan diekspos ke Salesforce sebagai API melalui API Gateway, dan pelanggan dapat mulai membuat prediksi dengan model yang diterapkan dari dalam Salesforce.

> **Note**
>
> Template ini mengizinkan kebijakan Transport Layer Security (TLS) versi 1.0 dan 1.1 untuk penyiapan API Gateway. Anda dapat membuat konfigurasi ini lebih aman dengan nama domain khusus. Untuk detailnya, lihat Menyiapkan nama domain khusus untuk REST API.

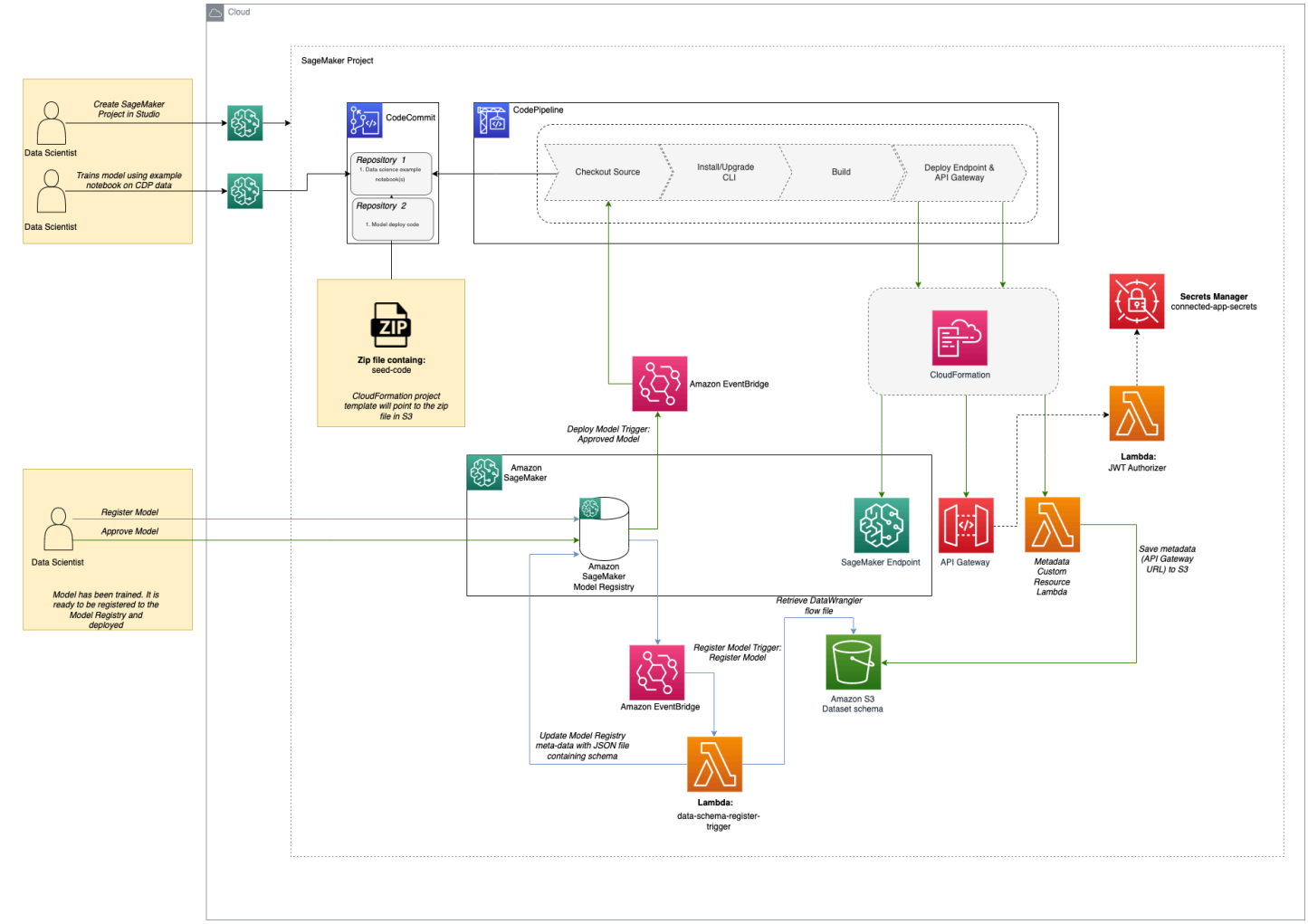


Posting blog Gunakan integrasi Amazon SageMaker dan Salesforce Data Cloud untuk memberi daya pada aplikasi Salesforce Anda dengan AI/ML memberikan petunjuk terperinci untuk memandu Anda melalui langkah-langkah berikut:

1. Pilih templat proyek Penyebaran model untuk Salesforce, dan berikan nama manajer rahasia.

2. Kloning repositori untuk menggunakan notebook sampel yang SageMaker disediakan dan kode penerapan model yang dapat disesuaikan.
3. Pra-proses data Anda dengan Data Wrangler.
    a. Buat koneksi ke Salesforce Data Cloud dan impor data ke Data Wrangler.
    b. Gunakan Data Wrangler untuk menyiapkan data dengan beberapa contoh transformasi.
    c. Memulai pekerjaan pemrosesan untuk memproses data menggunakan konfigurasi Data Wrangler Anda.
4. Latih modelnya.
5. Daftarkan model Anda di registri model.
6. Menyetujui model Anda di registri model.
7. Lihat titik akhir Anda di SageMaker konsol.
8. Panggil URL API dari Salesforce Einstein Studio untuk mendaftar dan menggunakan kesimpulan model di Einstein Studio.

Diagram berikut menunjukkan secara lebih rinci alur kerja dan AWS sumber daya yang digunakan oleh template SageMaker proyek dengan Salesforce Data Cloud Integration.



Perbarui SageMaker Proyek untuk Menggunakan Repositori Git Pihak Ketiga

Kebijakan terkelola yang dilampirkan pada `AmazonSageMakerServiceCatalogProductsUseRole` peran diperbarui pada 27 Juli 2021 untuk digunakan dengan templat Git pihak ketiga. Pengguna yang onboard ke Amazon SageMaker Studio Classic setelah tanggal ini dan mengaktifkan templat proyek menggunakan kebijakan baru. Pengguna yang melakukan onboard sebelum tanggal ini harus memperbarui kebijakan untuk menggunakan templat ini. Gunakan salah satu opsi berikut untuk memperbarui kebijakan:

- Hapus peran dan alihkan setelan Studio Classic

  1. Pada konsol IAM, hapus`AmazonSageMakerServiceCatalogProductsUseRole`.
  2. Di panel kontrol Studio Classic, pilih Edit Pengaturan.
  3. Alihkan kedua pengaturan lalu pilih Kirim.

- Di konsol IAM, tambahkan izin berikut ke: `AmazonSageMakerServiceCatalogProductsUseRole`

```
{
      "Effect": "Allow",
      "Action": [
          "codestar-connections:UseConnection"
      ],
      "Resource": "arn:aws:codestar-connections:*:*:connection/*",
      "Condition": {
          "StringEqualsIgnoreCase": {
              "aws:ResourceTag/sagemaker": "true"
          }
      }
  },
  {
      "Effect": "Allow",
      "Action": [
          "s3:PutObjectAcl"
      ],
      "Resource": [
          "arn:aws:s3:::sagemaker-*"
      ]
  }
```

Buat Template Proyek Kustom

Jika templat SageMaker yang disediakan tidak memenuhi kebutuhan Anda (misalnya, Anda ingin memiliki orkestrasi yang lebih kompleks CodePipeline dengan beberapa tahap atau langkah persetujuan khusus), buat templat Anda sendiri.

Sebaiknya mulai dengan menggunakan templat SageMaker yang disediakan untuk memahami cara mengatur kode dan sumber daya Anda dan membangun di atasnya. Untuk melakukan ini, setelah Anda mengaktifkan akses administrator ke SageMaker template, masuk ke https://console.aws.amazon.com/servicecatalog/, pilih Portofolio, lalu pilih Diimpor. Untuk informasi tentang Service Catalog, lihat Ringkasan Service Catalog di Panduan Pengguna Service Catalog.

Buat template proyek Anda sendiri untuk menyesuaikan proyek MLOPs Anda. SageMaker templat proyek adalah Katalog Layanan — produk yang disediakan untuk menyediakan sumber daya untuk proyek MLOP Anda.

Untuk membuat templat proyek kustom, selesaikan langkah berikut.

1. Buat portofolio. Untuk selengkapnya, lihat Langkah 3: Buat Portofolio Service Catalog.
2. Untuk membuat produk Produk adalah CloudFormation template. Anda dapat membuat beberapa versi produk. Untuk informasi, lihat Langkah 4: Membuat Produk Service Catalog.

   Agar produk dapat bekerja dengan SageMaker proyek, tambahkan parameter berikut ke templat produk Anda.

```
SageMakerProjectName:
Type: String
Description: Name of the project

SageMakerProjectId:
Type: String
Description: Service generated Id of the project.
```

   Important

   Kami menyarankan Anda membungkus CodeCommit repositori ke dalam repositori SageMaker kode agar repositori proyek terlihat dalam mode VPC. Template sampel dan penambahan yang diperlukan ditunjukkan dalam contoh kode berikut.
   Template asli (sampel):

```
ModelBuildCodeCommitRepository:
    Type: AWS::CodeCommit::Repository
    Properties:
      # Max allowed length: 100 chars
      RepositoryName: !Sub sagemaker-${SageMakerProjectName}-
${SageMakerProjectId}-modelbuild # max: 10+33+15+10=68
      RepositoryDescription: !Sub SageMaker Model building workflow
 infrastructure as code for the Project ${SageMakerProjectName}
      Code:
        S3:
          Bucket: SEEDCODE_BUCKETNAME
          Key: toolchain/model-building-workflow-v1.0.zip
        BranchName: main
```

   Konten tambahan untuk ditambahkan dalam mode VPC:

```
SageMakerRepository:
    Type: AWS::SageMaker::CodeRepository
    Properties:
        GitConfig:
            RepositoryUrl: !GetAtt
 ModelBuildCodeCommitRepository.CloneUrlHttp
            Branch: main
```

3. Tambahkan batasan peluncuran. Batasan peluncuran menentukan IAM role yang diasumsikan oleh Service Catalog ketika pengguna meluncurkan produk. Untuk informasi selengkapnya, lihat Langkah 6: Tambahkan Batasan Peluncuran untuk Menetapkan Role IAM.
4. Menyediakan produk di https://console.aws.amazon.com/servicecatalog/ untuk menguji template. Jika Anda puas dengan template Anda, lanjutkan ke langkah berikutnya untuk membuat template tersedia di Studio Classic.
5. Berikan akses ke portofolio Service Catalog yang Anda buat pada langkah 1 ke peran eksekusi Studio Classic Anda. Gunakan peran eksekusi domain Studio Classic atau peran pengguna yang memiliki akses Studio Classic. Untuk informasi tentang menambahkan peran ke portofolio, lihat Langkah 7: Memberikan Pengguna Akhir Akses ke Portofolio.
6. Agar template proyek Anda tersedia di daftar templat Organisasi di Studio Classic, buat tag dengan kunci dan nilai berikut ke produk Service Catalog yang Anda buat di langkah 2.
   - kunci: `sagemaker:studio-visibility`
   - nilai: `true`

Setelah Anda menyelesaikan langkah-langkah ini, pengguna Studio Classic di organisasi Anda dapat membuat proyek dengan template yang Anda buat dengan mengikuti langkah-langkah Membuat Proyek MLOPS menggunakan Amazon Studio Classic SageMaker dan memilih templat Organisasi saat Anda memilih templat.

Lihat Sumber Daya Proyek

Setelah Anda membuat proyek, lihat sumber daya yang terkait dengan proyek di Amazon SageMaker Studio Classic.

Untuk membuat proyek di Studio Classic

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di sidebar Studio Classic, pilih ikon Beranda
   ( ).
3. Pilih Deployment dari menu, lalu pilih Projects.
4. Pilih nama proyek yang ingin Anda lihat detailnya.

   Tab dengan detail proyek muncul.

Pada tab detail proyek, Anda dapat melihat entitas berikut yang terkait dengan proyek.

- Repositori: Repositori kode (repo) yang terkait dengan proyek ini. Jika Anda menggunakan template SageMaker -provided saat Anda membuat proyek Anda, itu akan membuat repo atau AWS CodeCommit repo Git pihak ketiga. Untuk informasi lebih lanjut tentang CodeCommit, lihat Apa itu AWS CodeCommit.
- Pipelines: saluran pipa SageMaker ML yang menentukan langkah-langkah untuk menyiapkan data, melatih, dan menyebarkan model. Untuk informasi tentang saluran pipa SageMaker ML, lihatMembuat dan Mengelola SageMaker Pipelines.
- Eksperimen: Satu atau lebih eksperimen Amazon SageMaker Autopilot yang terkait dengan proyek tersebut. Untuk informasi tentang Autopilot, lihat. SageMaker Autopilot
- Grup model: Grup versi model yang dibuat oleh eksekusi pipa dalam proyek. Untuk informasi tentang grup model, lihatMembuat Grup Model.
- Titik akhir: SageMaker titik akhir yang menjadi tuan rumah model yang diterapkan untuk inferensi waktu nyata. Ketika versi model disetujui, itu diterapkan ke titik akhir.
- Pengaturan: Pengaturan untuk proyek. Ini termasuk nama dan deskripsi proyek, informasi tentang template proyek dan`SourceModelPackageGroupName`, dan metadata tentang proyek.

Memperbarui Proyek MLOPS di Amazon Studio Classic SageMaker

Prosedur ini menunjukkan cara memperbarui proyek MLOPS di Amazon SageMaker Studio Classic. Anda dapat memperbarui deskripsi, versi template, dan parameter template.

Prasyarat

- Akun IAM atau Pusat Identitas IAM untuk masuk ke Studio Classic. Untuk informasi, lihat Ikhtisar SageMaker Domain Amazon.
- Keakraban dasar dengan antarmuka pengguna Studio Classic. Untuk informasi, lihat Ikhtisar UI Amazon SageMaker Studio Classic.
- Tambahkan kebijakan inline kustom berikut ke peran yang ditentukan:

  Memiliki peran yang dibuat pengguna `AmazonSageMakerFullAccess`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:CreateProvisionedProductPlan",
                "servicecatalog:DescribeProvisionedProductPlan",
                "servicecatalog:DeleteProvisionedProductPlan"
            ],
            "Resource": "*"
        }
    ]
}
```

  `AmazonSageMakerServiceCatalogProductsLaunchRole`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:DescribeChangeSet"
            ],
            "Resource": "arn:aws:cloudformation:*:*:stack/SC-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "codecommit:PutRepositoryTriggers"
```

```
            ],
            "Resource": "arn:aws:codecommit:*:*:sagemaker-*"
        }
    ]
}
```

Untuk memperbarui proyek di Studio Classic

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di sidebar Studio Classic, pilih ikon Beranda
   ( ).
3. Pilih Deployment dari menu, lalu pilih Projects. Daftar proyek Anda muncul.
4. Pilih nama proyek yang ingin Anda perbarui dalam daftar proyek.
5. Pilih Perbarui dari menu Tindakan di sudut kanan atas tab proyek.
6. Dalam kotak dialog Perbarui proyek, Anda dapat mengedit Deskripsi dan parameter template yang terdaftar.
7. Pilih Lihat perbedaan.

   Kotak dialog menampilkan pengaturan proyek asli dan diperbarui Anda. Setiap perubahan dalam pengaturan proyek Anda dapat mengubah atau menghapus sumber daya dalam proyek saat ini. Kotak dialog menampilkan perubahan ini juga.
8. Anda mungkin perlu menunggu beberapa menit agar tombol Perbarui menjadi aktif. Pilih Perbarui.
9. Pembaruan proyek mungkin memerlukan waktu beberapa menit untuk diselesaikan. Pilih Pengaturan di tab proyek dan pastikan parameter telah diperbarui dengan benar.

Menghapus Proyek MLOPS menggunakan Amazon Studio Classic SageMaker

Prosedur ini menunjukkan cara menghapus proyek MLOPS menggunakan Amazon SageMaker Studio Classic.

Prasyarat

> **Note**
>
> Anda hanya dapat menghapus proyek di Studio Classic yang telah Anda buat. Kondisi ini merupakan bagian dari izin katalog layanan `servicecatalog:TerminateProvisionedProduct` dalam `AmazonSageMakerFullAccess` kebijakan. Jika perlu, Anda dapat memperbarui kebijakan ini untuk menghapus kondisi ini.

- Akun IAM atau Pusat Identitas IAM untuk masuk ke Studio Classic. Untuk informasi, lihat Ikhtisar SageMaker Domain Amazon.
- Keakraban dasar dengan antarmuka pengguna Studio Classic. Untuk informasi, lihat Ikhtisar UI Amazon SageMaker Studio Classic.

Untuk menghapus proyek di Amazon SageMaker Studio Classic

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di sidebar Studio Classic, pilih ikon Beranda ().
3. Pilih Deployment dari menu, lalu pilih Projects.
4. Pilih proyek target dari daftar dropdown. Jika Anda tidak melihat proyek Anda, ketikkan nama proyek dan terapkan filter untuk menemukan proyek Anda.
5. Setelah Anda menemukan proyek Anda, pilih nama proyek untuk melihat detailnya.
6. Pilih Hapus dari menu Tindakan.
7. Konfirmasikan pilihan Anda dengan memilih Hapus dari jendela Hapus Proyek.

SageMaker Panduan Proyek MLOPs

Panduan ini menggunakan template Template MLOPs untuk pembuatan model, pelatihan, dan penyebaran untuk mendemonstrasikan menggunakan proyek MLOP untuk membuat sistem CI/CD untuk membangun, melatih, dan menyebarkan model.

Prasyarat

Untuk menyelesaikan panduan ini, Anda memerlukan:

- Akun IAM atau Pusat Identitas IAM untuk masuk ke Studio Classic. Untuk informasi, lihat Ikhtisar SageMaker Domain Amazon.
- Izin untuk menggunakan templat proyek SageMaker yang disediakan. Untuk informasi, lihat SageMaker Izin Studio yang Diperlukan untuk Menggunakan Proyek.
- Keakraban dasar dengan antarmuka pengguna Studio Classic. Untuk informasi, lihat Ikhtisar UI Amazon SageMaker Studio Classic.

Topik

Langkah 1: Buat Proyek

Pada langkah ini, Anda membuat proyek SageMaker MLOPs dengan menggunakan template proyek SageMaker yang disediakan untuk membangun, melatih, dan menyebarkan model.

Untuk membuat Proyek SageMaker MLOPs

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di sidebar Studio Classic, pilih ikon Beranda
   ( ).
3. Pilih Deployment dari menu, lalu pilih Projects.
4. Pilih Buat proyek.

   Tab Create project akan muncul.
5. Jika belum dipilih, pilih SageMaker template, lalu pilih template MLOPs untuk pembuatan model, pelatihan, dan penerapan.
6. Untuk detail proyek, masukkan nama dan deskripsi untuk proyek Anda.

Ketika proyek muncul di daftar Proyek dengan Status Buat selesai, lanjutkan ke langkah berikutnya.

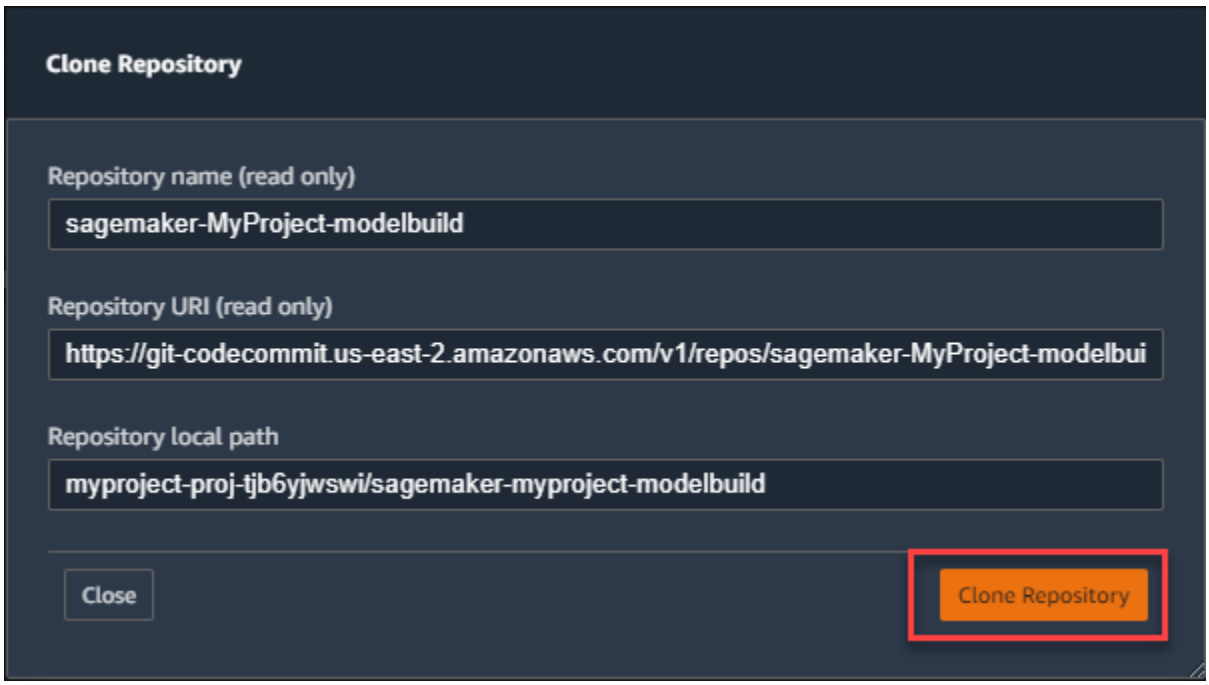
> **Important**
>
> Mulai 25 Juli 2022, kami memerlukan peran tambahan untuk menggunakan templat proyek. Jika Anda melihat pesan CodePipeline kesalahan tidak diizinkan untuk dilakukan AssumeRole pada peran arn:aws:iam: :xxx:role/service-role/AmazonSageMakerServiceCatalogProductsCode PipelineRole, lihat Langkah 5-6 untuk daftar lengkap peran dan instruksi yang diperlukan tentang cara membuatnya. SageMaker Izin Studio yang Diperlukan untuk Menggunakan Proyek

Langkah 2: Kloning Repositori Kode

Setelah Anda membuat proyek, dua CodeCommit repositori dibuat dalam proyek. Salah satu repositori berisi kode untuk membangun dan melatih model, dan satu berisi kode untuk menyebarkan model. Pada langkah ini, Anda mengkloning repositori ke SageMaker proyek lokal Anda yang berisi kode untuk membangun dan melatih model ke lingkungan Studio Classic lokal sehingga Anda dapat bekerja dengan kode.

Untuk mengkloning repositori kode

1. Di sidebar Studio Classic, pilih ikon Beranda ().
2. Pilih Deployment dari menu, lalu pilih Projects.
3. Pilih proyek yang Anda buat pada langkah sebelumnya untuk membuka tab proyek untuk proyek Anda.
4. Di tab project, pilih Repositori, dan di kolom Local path untuk repositori yang diakhiri dengan modelbuild, pilih clone repo... .
5. Di kotak dialog klon yang muncul, terima defaultnya dan memilih Repositori klon.



Ketika klon repositori selesai, jalur lokal muncul di kolom Local path. Pilih jalur untuk membuka folder lokal yang berisi kode repositori di Studio Classic.

Langkah 3: Buat Perubahan pada Kode

Sekarang buat perubahan pada kode pipeline yang membangun model dan periksa perubahan untuk memulai proses pipeline baru. Pipeline run mendaftarkan versi model baru.

Untuk membuat perubahan kode

1. Di Studio Classic, pilih ikon browser file
   ( ),
   dan navigasikan ke `pipelines/abalone` folder. Klik dua kali `pipeline.py` untuk membuka file kode.
2. Dalam `pipeline.py` file, temukan baris yang menetapkan jenis instance pelatihan.

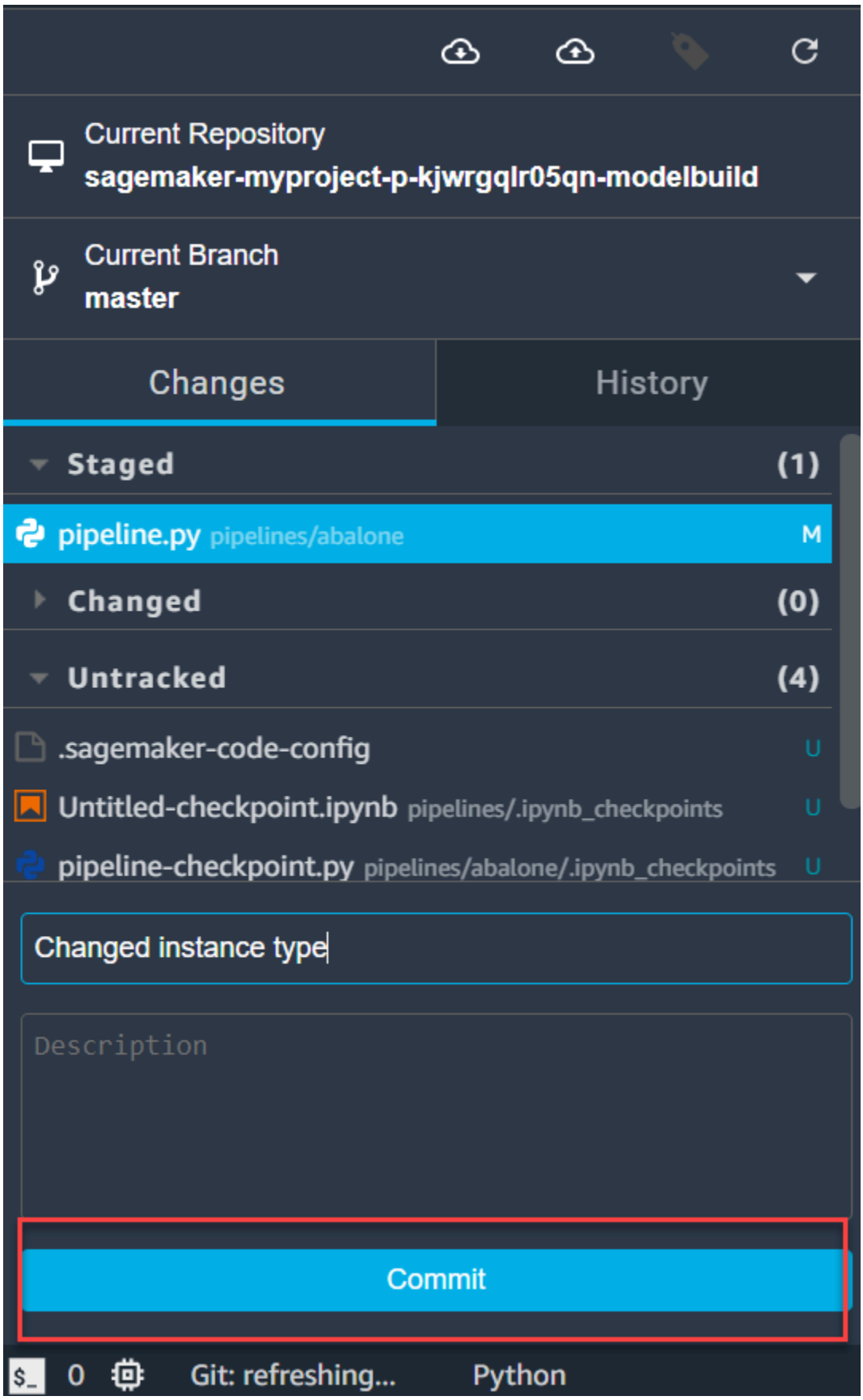
   ```
   training_instance_type = ParameterString(
          name="TrainingInstanceType", default_value="ml.m5.xlarge"
   ```

   Ubah `ml.m5.xlarge` ke`ml.m5.large`, lalu ketik `Ctrl+S` untuk menyimpan perubahan.

3. Pilih ikon Git

   ( ).

   Panggung, komit, dan dorong perubahan`pipeline.py`. Juga, masukkan ringkasan di bidang Ringkasan dan deskripsi opsional di bidang Deskripsi. Untuk informasi tentang penggunaan Git di Studio Classic, lihatMengkloning Repositori Git di Studio Classic SageMaker .


Current Repository
sagemaker-myproject-p-kjwrgqlr05qn-modelbuild
Current Branch
master
Changes
History
Staged (1)
pipeline.py pipelines/abalone M
Changed (0)
Untracked (4)
.sagemaker-code-config U
Untitled-checkpoint.ipynb pipelines/.ipynb_checkpoints U
pipeline-checkpoint.py pipelines/abalone/.ipynb_checkpoints U
Changed instance type
Description
Commit
0
Git: refreshing...
Python

Setelah mendorong perubahan kode Anda, sistem MLOPs memulai menjalankan pipeline yang membuat versi model baru. Pada langkah berikutnya, Anda menyetujui versi model baru untuk menerapkannya ke produksi.

Langkah 4: Menyetujui Model

Sekarang Anda menyetujui versi model baru yang dibuat pada langkah sebelumnya untuk memulai penerapan versi model ke titik akhir. SageMaker

Untuk menyetujui versi model

1. Di sidebar Studio Classic, pilih ikon Beranda
   ().
2. Pilih Deployment dari menu, lalu pilih Projects.
3. Pilih nama proyek yang Anda buat pada langkah pertama untuk membuka tab proyek untuk proyek Anda.
4. Di tab proyek, pilih Grup model, lalu klik dua kali nama grup model yang muncul.

   Tab grup model muncul.
5. Di tab grup model, klik dua kali Versi 1. Tab Versi 1 terbuka. Pilih Perbarui status.
6. Dalam kotak dialog Perbarui status versi model model, di daftar tarik-turun Status, pilih Menyetujui, lalu pilih Perbarui status.

   Menyetujui versi model menyebabkan sistem MLOPs menyebarkan model ke pementasan. Untuk melihat titik akhir, pilih tab Endpoints pada tab project.

(Opsional) Langkah 5: Deploy Versi Model ke Produksi

Sekarang Anda dapat menerapkan versi model ke lingkungan produksi.

> **Note**
>
> Untuk menyelesaikan langkah ini, Anda harus menjadi administrator di domain Studio Classic Anda. Jika Anda bukan administrator, lewati langkah ini.

Untuk menyebarkan versi model ke lingkungan produksi

1. Masuk ke CodePipeline konsol di https://console.aws.amazon.com/codepipeline/
2. *Pilih **Pipelines**, lalu pilih pipeline dengan nama **sagemaker- projectname - projectid -modeldeploy, di mana projectname adalah nama proyek Anda, dan projectid** adalah ID proyek Anda.*
3. Di DeployStagingpanggung, pilih Review.
4. Di kotak dialog Tinjau, pilih Menyetujui.

   Menyetujui DeployStagingtahapan menyebabkan sistem MLOPs menyebarkan model ke produksi. Untuk melihat titik akhir, pilih tab Endpoints pada tab project di Studio Classic.

Langkah 6: Bersihkan Sumber Daya

Untuk menghentikan biaya, bersihkan sumber daya yang dibuat dalam panduan ini. Untuk melakukan ini, selesaikan langkah-langkah berikut.

> **Note**
>
> Untuk menghapus AWS CloudFormation tumpukan dan bucket Amazon S3, Anda harus menjadi administrator di Studio Classic. Jika Anda bukan administrator, mintalah administrator Anda untuk menyelesaikan langkah-langkah tersebut.

1. Di sidebar Studio Classic, pilih ikon Beranda ().
2. Pilih Deployment dari menu, lalu pilih Projects.
3. Pilih proyek target dari daftar dropdown. Jika Anda tidak melihat proyek Anda, ketikkan nama proyek dan terapkan filter untuk menemukan proyek Anda.
4. Anda dapat menghapus proyek Studio Classic dengan salah satu cara berikut:

   a. Anda dapat menghapus proyek dari daftar proyek.

      Klik kanan proyek target dan pilih Hapus dari daftar dropdown.

> **Note**
>
> Fungsionalitas ini didukung di Studio Classic versi v3.17.1 atau lebih tinggi. Untuk informasi selengkapnya, lihat Matikan dan Perbarui SageMaker Studio Classic.

b. Anda dapat menghapus proyek dari bagian Detail proyek.

   i. Ketika Anda telah menemukan proyek Anda, klik dua kali untuk melihat detailnya di panel utama.

   ii. Pilih Hapus dari menu Tindakan.

5. Konfirmasikan pilihan Anda dengan memilih Hapus dari jendela Hapus Proyek.

   Ini menghapus produk yang disediakan oleh Service Catalog yang dibuat oleh proyek. Ini termasuk CodeCommit, CodePipeline, dan CodeBuild sumber daya yang dibuat untuk proyek.

6. Hapus AWS CloudFormation tumpukan yang dibuat proyek. Ada dua tumpukan, satu untuk pementasan dan satu untuk produksi. *Nama tumpukan adalah* ***sagemaker- projectname - project-id -deploy-staging dan sagemaker- projectname - project-id -deploy-prod, di mana projectname adalah nama proyek Anda, dan project-id*** *adalah ID proyek Anda.*

   Untuk selengkapnya tentang cara menghapus AWS CloudFormation tumpukan, lihat Menghapus tumpukan di AWS CloudFormation konsol di Panduan AWS CloudFormation Pengguna.

7. Hapus bucket Amazon S3 yang dibuat proyek. Nama bucket adalah sagemaker-project-***project-id, di mana project-id*** adalah ID proyek Anda.

SageMaker Panduan Proyek MLOPS Menggunakan Repos Git Pihak Ketiga

Panduan ini menggunakan template Template MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga menggunakan CodePipeline untuk mendemonstrasikan cara menggunakan proyek MLOP untuk membuat sistem CI/CD untuk membangun, melatih, dan menyebarkan model.

Prasyarat

Untuk menyelesaikan panduan ini, Anda memerlukan:

- Akun IAM atau IAM Identity Center untuk masuk ke Studio Classic. Untuk informasi, lihat Ikhtisar SageMaker Domain Amazon.
- Izin untuk menggunakan templat proyek SageMaker yang disediakan. Untuk informasi, lihat SageMaker Izin Studio yang Diperlukan untuk Menggunakan Proyek.
- Keakraban dasar dengan antarmuka pengguna Studio Classic. Untuk informasi, lihat Ikhtisar UI Amazon SageMaker Studio Classic.
- Dua GitHub repositori diinisialisasi dengan README. Anda memasukkan repositori ini ke dalam template proyek, yang akan menyemai repo ini dengan model build dan deploy code.

Topik

Langkah 1: Siapkan GitHub koneksi

Pada langkah ini, Anda terhubung ke GitHub repositori Anda menggunakan koneksiAWS CodeStar. SageMaker Proyek ini menggunakan koneksi ini untuk mengakses repositori kode sumber Anda.

Untuk mengatur GitHub koneksi:

1. Masuk ke CodePipeline konsol di https://console.aws.amazon.com/codepipeline/
2. Di bawah Pengaturan di panel navigasi, pilih Koneksi.
3. Pilih Buat koneksi.
4. Untuk Pilih penyedia, pilih GitHub.
5. Untuk nama Koneksi, masukkan nama.
6. Pilih Connect to GitHub.
7. Jika GitHub aplikasi AWS Connector sebelumnya tidak diinstal, pilih Instal aplikasi baru.

   Ini menampilkan daftar semua akun GitHub pribadi dan organisasi tempat Anda memiliki akses.

8. Pilih akun tempat Anda ingin membangun konektivitas untuk digunakan dengan SageMaker proyek dan GitHub repositori.
9. Pilih Konfigurasikan
10. Anda dapat memilih repositori spesifik Anda atau memilih Semua repositori.
11. Pilih Simpan. Saat aplikasi diinstal, Anda akan diarahkan ke GitHub halaman Connect to dan ID penginstalan diisi secara otomatis.
12. Pilih Hubungkan.
13. Tambahkan tag dengan kunci `sagemaker` dan nilai `true` ke AWS CodeStar koneksi ini.
14. Salin koneksi ARN untuk disimpan nanti. Anda menggunakan ARN sebagai parameter dalam langkah pembuatan proyek.

Langkah 2: Buat Proyek

Pada langkah ini, Anda membuat proyek SageMaker MLOPs dengan menggunakan template proyek SageMaker yang disediakan untuk membangun, melatih, dan menyebarkan model.

Untuk membuat Proyek SageMaker MLOPs

1. Masuk ke Studio Classic. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di sidebar Studio Classic, pilih ikon Beranda
   ( ).
3. Pilih Deployment dari menu, lalu pilih Projects.
4. Pilih Buat proyek.

   Tab Create project akan muncul.
5. Untuk template SageMaker proyek, pilih template MLOPs untuk pembuatan model, pelatihan, dan penerapan dengan repositori Git pihak ketiga.
6. Pilih Pilih template proyek.
7. Di bawah ModelBuild CodeRepository Info, berikan parameter berikut:

   - Untuk URL, masukkan URL repositori Git Anda untuk kode build model dalam format https://*`git-url`* .git.
   - Untuk Branch, masukkan cabang yang akan digunakan dari repositori Git Anda untuk aktivitas pipeline.

- *Untuk Nama **Repositori Lengkap, masukkan nama** repositori Git dalam format nama pengguna/nama repositori atau nama organisasi/repositori.*
- Untuk ARN Koneksi Codestar, masukkan ARN koneksi yang Anda buat AWS CodeStar di Langkah 1.
- Sakelar sakelar Kode Sampel memungkinkan Anda memilih apakah akan mengisi repositori dengan kode benih build model. Kita bisa membiarkannya untuk demo ini.

8. Di bawah ModelDeploy CodeRepository Info, berikan parameter berikut:

   - *Untuk **URL**, masukkan URL repositori Git Anda untuk kode penerapan model dalam format https://git-url .git.*
   - Untuk Branch, masukkan cabang yang akan digunakan dari repositori Git Anda untuk aktivitas pipeline.
   - *Untuk Nama **Repositori Lengkap, masukkan nama** repositori Git dalam format nama pengguna/nama repositori atau nama organisasi/repositori.*
   - Untuk ARN Koneksi Codestar, masukkan ARN koneksi yang Anda buat AWS CodeStar di Langkah 1.
   - Sakelar sakelar Kode Sampel memungkinkan Anda memilih apakah akan mengisi repositori dengan kode benih penerapan model. Kita bisa membiarkannya untuk demo ini.

9. Pilih Buat Proyek.

Proyek ini muncul di daftar Proyek dengan Status Dibuat.

Langkah 3: Buat Perubahan pada Kode

Sekarang buat perubahan pada kode pipeline yang membangun model dan lakukan perubahan untuk memulai proses pipeline baru. Pipeline run mendaftarkan versi model baru.

Untuk membuat perubahan kode

1. Di GitHub repo build model Anda, navigasikan ke `pipelines/abalone` folder. Klik dua kali `pipeline.py` untuk membuka file kode.
2. Dalam `pipeline.py` file, temukan baris yang menetapkan jenis instance pelatihan.

```
training_instance_type = ParameterString(
        name="TrainingInstanceType", default_value="ml.m5.xlarge"
```

Buka file untuk diedit, ubah `ml.m5.xlarge` ke`ml.m5.large`, lalu komit.

Setelah Anda melakukan perubahan kode, sistem MLOPs memulai proses pipeline yang membuat versi model baru. Pada langkah berikutnya, Anda menyetujui versi model baru untuk menerapkannya ke produksi.

Langkah 4: Menyetujui Model

Sekarang Anda menyetujui versi model baru yang dibuat pada langkah sebelumnya untuk memulai penerapan versi model ke titik akhir. SageMaker

Untuk menyetujui versi model

1. Di sidebar Studio Classic, pilih ikon Beranda
   ( ).
2. Pilih Deployment dari menu, lalu pilih Projects.
3. Temukan nama proyek yang Anda buat pada langkah pertama dan klik dua kali untuk membuka tab proyek untuk proyek Anda.
4. Di tab proyek, pilih Grup model, lalu klik dua kali nama grup model yang muncul.

   Tab grup model muncul.
5. Di tab grup model, klik dua kali Versi 1. Tab Versi 1 terbuka. Pilih Perbarui status.
6. Dalam kotak dialog Perbarui status versi model model, di daftar tarik-turun Status, pilih Menyetujui lalu pilih Perbarui status.

   Menyetujui versi model menyebabkan sistem MLOPs menyebarkan model ke pementasan. Untuk melihat titik akhir, pilih tab Endpoints pada tab project.

(Opsional) Langkah 5: Deploy Versi Model ke Produksi

Sekarang Anda dapat menerapkan versi model ke lingkungan produksi.

> **Note**
>
> Untuk menyelesaikan langkah ini, Anda harus menjadi administrator di domain Studio Classic Anda. Jika Anda bukan administrator, lewati langkah ini.

Untuk menyebarkan versi model ke lingkungan produksi

1. Masuk ke CodePipeline konsol di https://console.aws.amazon.com/codepipeline/
2. *Pilih **Pipelines**, lalu pilih pipeline dengan nama **sagemaker- projectname - projectid -modeldeploy**, di mana **projectname** adalah nama proyek Anda, dan **projectid** adalah ID proyek Anda.*
3. Di DeployStagingpanggung, pilih Review.
4. Di kotak dialog Tinjau, pilih Menyetujui.

   Menyetujui DeployStagingtahapan menyebabkan sistem MLOPs menyebarkan model ke produksi. Untuk melihat titik akhir, pilih tab Endpoints pada tab project di Studio Classic.

Langkah 6: Bersihkan Sumber Daya

Untuk menghentikan biaya, bersihkan sumber daya yang dibuat dalam panduan ini.

> **Note**
>
> Untuk menghapus AWS CloudFormation tumpukan dan bucket Amazon S3, Anda harus menjadi administrator di Studio Classic. Jika Anda bukan administrator, mintalah administrator Anda untuk menyelesaikan langkah-langkah tersebut.

1. Di sidebar Studio Classic, pilih ikon Beranda ().
2. Pilih Deployment dari menu, lalu pilih Projects.
3. Pilih proyek target dari daftar dropdown. Jika Anda tidak melihat proyek Anda, ketikkan nama proyek dan terapkan filter untuk menemukan proyek Anda.
4. Pilih proyek Anda untuk melihat detailnya di panel utama.
5. Pilih Hapus dari menu Tindakan.
6. Konfirmasikan pilihan Anda dengan memilih Hapus dari jendela Hapus Proyek.

   Ini menghapus produk yang disediakan oleh Service Catalog yang dibuat oleh proyek. Ini termasuk CodeCommit, CodePipeline, dan CodeBuild sumber daya yang dibuat untuk proyek.
7. Hapus AWS CloudFormation tumpukan yang dibuat proyek. Ada dua tumpukan, satu untuk pementasan dan satu untuk produksi. *Nama tumpukan adalah **sagemaker- projectname***

***- project-id -deploy-staging dan sagemaker- projectname - project-id - deploy-prod, di mana projectname adalah nama proyek Anda, dan project-id** adalah ID proyek Anda.*

Untuk selengkapnya tentang cara menghapus AWS CloudFormation tumpukan, lihat Menghapus tumpukan di AWS CloudFormation konsol di Panduan AWS CloudFormation Pengguna.

8. Hapus bucket Amazon S3 yang dibuat proyek. Nama bucket adalah sagemaker-project-***project-id, di mana project-id*** adalah ID proyek Anda.

FAQ Amazon SageMaker MLOPs

Gunakan item FAQ berikut untuk menemukan jawaban atas pertanyaan umum tentang MLOP di. SageMaker

T. Apakah saya perlu menggunakan SDK SageMaker Python untuk membuat pipeline? SageMaker

Tidak, SageMaker Python SDK tidak diperlukan untuk membuat pipeline. SageMaker Anda juga dapat menggunakan boto3 atau. AWS CloudFormation Membuat pipa memerlukan definisi pipa, yang merupakan objek JSON yang mendefinisikan setiap langkah pipa. SageMaker SDK menawarkan cara sederhana untuk membuat definisi pipeline, yang dapat Anda gunakan dengan salah satu API yang disebutkan sebelumnya untuk membuat pipeline itu sendiri. Tanpa menggunakan SDK, pengguna harus menulis definisi JSON mentah untuk membuat pipeline tanpa pemeriksaan kesalahan yang disediakan oleh Python SageMaker SDK. Untuk melihat skema definisi JSON pipeline, lihat SageMaker Pipeline Definition JSON Schema. Contoh kode berikut menunjukkan contoh objek JSON definisi SageMaker pipa:

```
{'Version': '2020-12-01',
 'Metadata': {},
 'Parameters': [{'Name': 'ProcessingInstanceType',
   'Type': 'String',
   'DefaultValue': 'ml.m5.xlarge'},
  {'Name': 'ProcessingInstanceCount', 'Type': 'Integer', 'DefaultValue': 1},
  {'Name': 'TrainingInstanceType',
   'Type': 'String',
   'DefaultValue': 'ml.m5.xlarge'},
  {'Name': 'ModelApprovalStatus',
   'Type': 'String',
   'DefaultValue': 'PendingManualApproval'},
```

```
  {'Name': 'ProcessedData',
   'Type': 'String',
   'DefaultValue': 'S3_URL',
{'Name': 'InputDataUrl',
   'Type': 'String',
   'DefaultValue': 'S3_URL',
 'PipelineExperimentConfig': {'ExperimentName': {'Get': 'Execution.PipelineName'},
  'TrialName': {'Get': 'Execution.PipelineExecutionId'}},
 'Steps': [{'Name': 'ReadTrainDataFromFS',
   'Type': 'Processing',
   'Arguments': {'ProcessingResources': {'ClusterConfig': {'InstanceType':
 'ml.m5.4xlarge',
      'InstanceCount': 2,
      'VolumeSizeInGB': 30}},
    'AppSpecification': {'ImageUri': 'IMAGE_URI',
     'ContainerArguments': [....]},
    'RoleArn': 'ROLE',
      'ProcessingInputs': [...],
    'ProcessingOutputConfig': {'Outputs': [.....]},
    'StoppingCondition': {'MaxRuntimeInSeconds': 86400}},
   'CacheConfig': {'Enabled': True, 'ExpireAfter': '30d'}},
   ...
   ...
   ...
  }
```

T. Mengapa saya melihat langkah repack di pipeline saya SageMaker ?

Pengemasan ulang model terjadi ketika pipeline perlu menyertakan skrip khusus dalam file model terkompresi (model.tar.gz) untuk diunggah ke Amazon S3 dan digunakan untuk menerapkan model ke titik akhir. SageMaker Ketika SageMaker pipeline melatih model dan mendaftarkannya ke registri model, ia memperkenalkan langkah pengemasan ulang jika keluaran model terlatih dari pekerjaan pelatihan perlu menyertakan skrip inferensi khusus. Langkah repack membuka kompres model, menambahkan skrip baru, dan mengompresi ulang model. Menjalankan pipeline menambahkan langkah repack sebagai pekerjaan pelatihan.

T. Dapatkah saya menggunakan SageMaker Eksperimen dengan SageMaker Pipelines?

Ya. SageMaker Pipelines secara native terintegrasi dengan SageMaker Eksperimen. Anda dapat menggunakan `PipelineExperimentConfig` saat membuat pipeline dan mengatur nama

SageMaker Eksperimen Anda sendiri. Setiap proses pipa membuat uji coba, dan setiap langkah dalam pipa sesuai dengan a `TrialComponent` dalam uji coba. Jika tidak ada nama percobaan yang ditentukan dalam konfigurasi eksperimen, ID eksekusi pipeline digunakan sebagai nama percobaan.

```
pipeline = Pipeline(
   name=pipeline_name,
    parameters=[...],
    steps=[...],
    sagemaker_session=sagemaker_session,
    pipeline_experiment_config=PipelineExperimentConfig(
        ExecutionVariables.PIPELINE_NAME,
        ExecutionVariables.PIPELINE_EXECUTION_ID
    )
)
```

Q. Template SageMaker proyek memiliki repositori penerapan model yang menggunakan CloudFormation (CFN) untuk membuat titik akhir. Apakah ada cara untuk menerapkan model tanpa menggunakan? CloudFormation

Anda dapat menyesuaikan repositori penerapan di template proyek untuk menyebarkan model dari registri model sesuka Anda. Template digunakan CloudFormation untuk membuat titik akhir real-time, sebagai contoh. Anda dapat memperbarui penerapan untuk menggunakan SageMaker SDK, boto3, atau API lainnya yang dapat membuat titik akhir, bukan CFN. Jika Anda perlu memperbarui CodeBuild langkah-langkah sebagai bagian dari pipeline penerapan, Anda dapat membuat template khusus.

T. Bagaimana kita meneruskan file model URL Amazon S3 dari langkah kereta ke langkah register model dalam pipeline pada SageMaker waktu berjalan?

Anda dapat mereferensikan lokasi model sebagai properti dari langkah pelatihan, seperti yang ditunjukkan pada end-to-end contoh CustomerChurn pipeline di Github.

T. Jika saya memperluas wadah bawaan untuk melatih estimator atau untuk **`ProcessingStep`** on SageMaker Pipelines, apakah perlu menyalin skrip ke wadah di Dockerfile?

Tidak, Anda dapat menyalin skrip ke wadah atau meneruskannya melalui `entry_point` argumen (entitas estimator Anda) atau `code` argumen (entitas prosesor Anda), seperti yang ditunjukkan dalam contoh kode berikut.

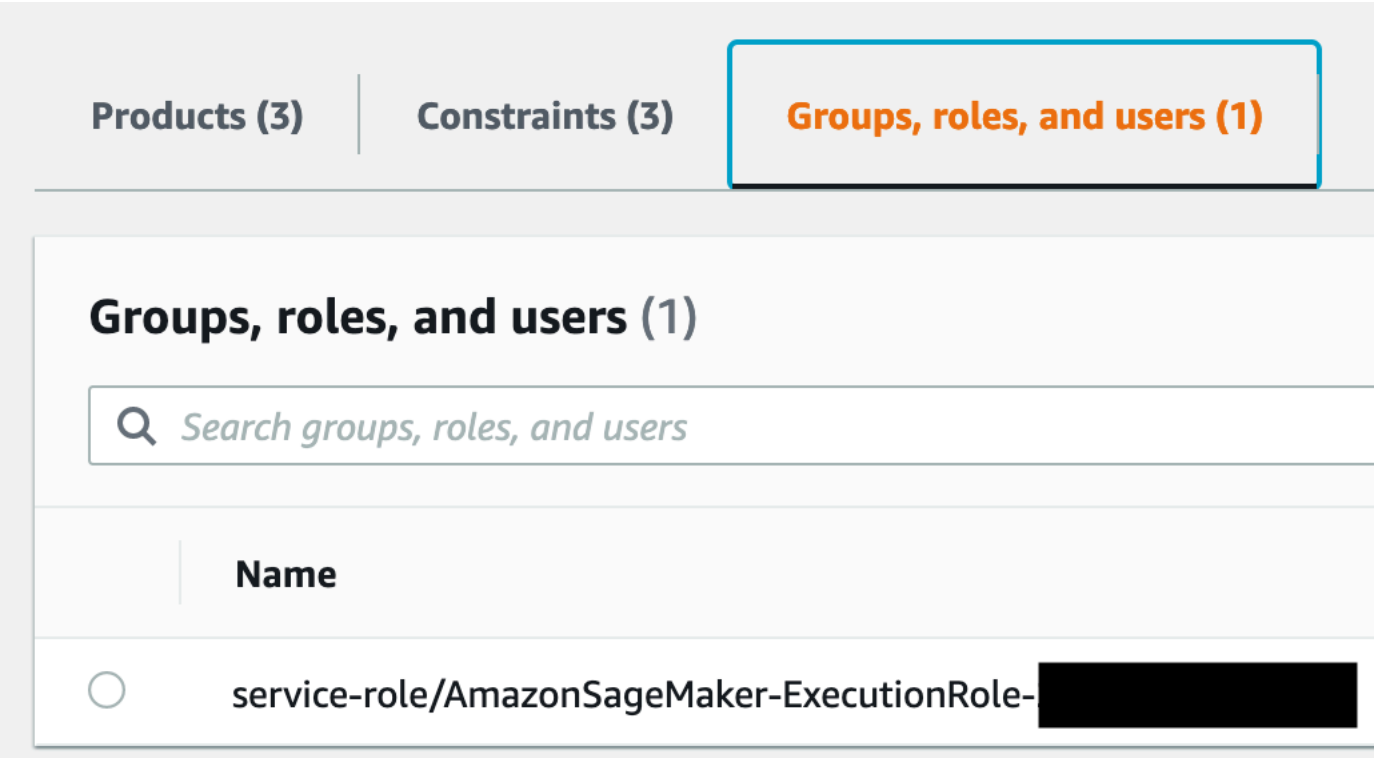
```
step_process = ProcessingStep(
    name="PreprocessAbaloneData",
    processor=sklearn_processor,
    inputs = [
        ProcessingInput(
            input_name='dataset',
            source=...,
            destination="/opt/ml/processing/code",
        )
    ],
    outputs=[
        ProcessingOutput(output_name="train", source="/opt/ml/processing/train",
 destination = processed_data_path),
        ProcessingOutput(output_name="validation", source="/opt/ml/processing/
validation", destination = processed_data_path),
        ProcessingOutput(output_name="test", source="/opt/ml/processing/test",
 destination = processed_data_path),
    ],
    code=os.path.join(BASE_DIR, "process.py"), ## Code is passed through an argument
    cache_config = cache_config,
    job_arguments = ['--input', 'arg1']
)

sklearn_estimator = SKLearn(
    entry_point=os.path.join(BASE_DIR, "train.py"), ## Code is passed through the
 entry_point
    framework_version="0.23-1",
    instance_type=training_instance_type,
    role=role,
    output_path=model_path, # New
    sagemaker_session=sagemaker_session, # New
    instance_count=1, # New
    base_job_name=f"{base_job_prefix}/pilot-train",
    metric_definitions=[
        {'Name': 'train:accuracy', 'Regex': 'accuracy_train=(.*?);'},
        {'Name': 'validation:accuracy', 'Regex': 'accuracy_validation=(.*?);'}
    ],
)
```

T. Apa cara yang disarankan untuk mengelola dependensi untuk berbagai SageMaker langkah Pipelines?

Anda dapat menggunakan template SageMaker Projects untuk mengimplementasikan CI/CD pembuatan gambar. Dengan template ini, Anda dapat mengotomatiskan CI/CD gambar yang dibangun dan didorong ke Amazon ECR. Perubahan pada file kontainer di repositori kontrol sumber proyek Anda akan memulai pipeline HTML dan menerapkan versi terbaru untuk kontainer Anda. Untuk informasi selengkapnya, lihat blog Membuat SageMaker proyek Amazon dengan pipeline CI/CD yang membangun gambar.

T. Bagaimana cara menyediakan akses SageMaker Project ke profil pengguna tertentu di Amazon SageMaker Studio Classic?

Karena SageMaker Projects didukung oleh Service Catalog, Anda harus menambahkan setiap peran yang memerlukan akses ke SageMaker Projects ke Amazon SageMaker Solutions dan Portofolio produk ML Ops di katalog layanan. Anda dapat melakukannya di Grup, Role, dan tab pengguna, seperti yang ditunjukkan pada gambar berikut. Jika setiap profil pengguna di Studio Classic memiliki peran yang berbeda, Anda harus menambahkan masing-masing peran tersebut ke katalog layanan. Anda juga dapat melakukan ini sambil membuat profil pengguna di Studio Classic.



T. Di mana saya dapat melihat properti yang terkait dengan setiap langkah SageMaker pipeline sehingga saya dapat menggunakannya pada langkah selanjutnya?

Setiap langkah dalam pipeline menggunakan SageMaker API yang mendasari untuk pekerjaan terkait. Misalnya, `TrainingStep` memanggil `CreateTrainingJob` API dan properti langkah sesuai dengan respons dari`DescribeTrainingJob`. Output respons dapat ditemukan di tautan referensi API untuk DescribeTrainingJob. Anda dapat mengikuti prosedur yang sama untuk mendapatkan properti untuk TransformStep, ProcessingStep, TuningStep, dan CreateModelStep. Untuk informasi selengkapnya tentang langkah alur, lihat Langkah Pipelkan.

T. Di SageMaker Pipelines, dapatkah saya menentukan jalur keluaran unik untuk langkah pipeline sehingga data outputnya tidak akan diganti oleh future run?

Ya, Anda dapat menggunakan ExecutionVariablesdan fungsi Gabung untuk menentukan lokasi output Anda. `ExecutionVariables`diselesaikan saat runtime. Misalnya, `ExecutionVariables.PIPELINE_EXECUTION_ID` diselesaikan ke ID eksekusi saat ini, yang dapat digunakan sebagai pengidentifikasi unik di berbagai proses.

```
from sagemaker.workflow.execution_variables import ExecutionVariables

processor_run_args = sklearn_processor.run(
    outputs=[
        ProcessingOutput(
            output_name="train",
            source="/opt/ml/processing/train",
            destination=Join(
                on="/",
                values=[
                    "s3:/",
                    default_bucket,
                    base_job_prefix,
                    ExecutionVariables.PIPELINE_EXECUTION_ID,
                    "PreprocessData",
                ],
            ),
        ),
        ProcessingOutput(
            output_name="validation",
            source="/opt/ml/processing/validation",
            destination=Join(
                on="/",
```

```
                values=[
                    "s3:/",
                    default_bucket,
                    base_job_prefix,
                    ExecutionVariables.PIPELINE_EXECUTION_ID,
                    "PreprocessData",
                ],
            ),
        ),
        ProcessingOutput(
            output_name="test",
            source="/opt/ml/processing/test",
            destination=Join(
                on="/",
                values=[
                    "s3:/",
                    default_bucket,
                    base_job_prefix,
                    ExecutionVariables.PIPELINE_EXECUTION_ID,
                    "PreprocessData",
                ],
            ),
        ),
    ],
    code="code/preprocess.py",
    arguments=["--input-data", input_data],
)

step_process = ProcessingStep(
    name="MyPreprocessingStep",
    step_args=processor_run_args,
)
```

T. Apa cara terbaik untuk mereproduksi model saya? SageMaker

SageMakerLayanan Lineage Tracking bekerja di backend untuk melacak semua metadata yang terkait dengan pelatihan model dan alur kerja penerapan Anda. Ini termasuk pekerjaan pelatihan Anda, kumpulan data yang digunakan, saluran pipa, titik akhir, dan model sebenarnya. Anda dapat menanyakan layanan garis keturunan kapan saja untuk menemukan artefak yang tepat yang digunakan untuk melatih model. Dengan menggunakan artefak tersebut, Anda dapat membuat ulang alur kerja ML yang sama untuk mereproduksi model selama Anda memiliki akses ke kumpulan data yang tepat yang digunakan. Komponen percobaan melacak pekerjaan pelatihan. Komponen uji coba

ini memiliki semua parameter yang digunakan sebagai bagian dari pekerjaan pelatihan. Jika Anda tidak perlu menjalankan kembali seluruh alur kerja, Anda dapat mereproduksi pekerjaan pelatihan untuk mendapatkan model yang sama.

T. Jika saya mencoba menghapus SageMaker proyek yang dibuat dari SageMaker template dan menerima kesalahan karena bucket Amazon S3 yang tidak kosong atau repositori Amazon ECR, bagaimana cara menghapus proyek?

Jika Anda mencoba menghapus SageMaker proyek Anda dan mendapatkan salah satu pesan kesalahan berikut:

```
The bucket you tried to delete is not empty
```

```
The repository with name 'repository-name' in registry
        with id 'id' cannot be deleted because it still contains images
```

maka Anda memiliki ember Amazon S3 yang tidak kosong atau repositori Amazon ECR yang perlu Anda hapus secara manual sebelum Anda menghapus proyek. SageMaker AWS CloudFormationtidak secara otomatis menghapus bucket Amazon S3 yang tidak kosong atau repositori Amazon ECR untuk Anda.

Memantau data dan kualitas model

Amazon SageMaker Model Monitor memantau kualitas model pembelajaran SageMaker mesin Amazon dalam produksi. Anda dapat mengatur pemantauan berkelanjutan dengan titik akhir real-time (atau pekerjaan transformasi batch yang berjalan secara teratur), atau pemantauan sesuai jadwal untuk pekerjaan transformasi batch asinkron. Dengan Model Monitor, Anda dapat mengatur peringatan yang memberi tahu Anda ketika ada penyimpangan dalam kualitas model. Deteksi dini dan proaktif dari penyimpangan ini memungkinkan Anda untuk mengambil tindakan korektif, seperti model pelatihan ulang, mengaudit sistem hulu, atau memperbaiki masalah kualitas tanpa harus memantau model secara manual atau membuat perkakas tambahan. Anda dapat menggunakan kemampuan pemantauan bawaan Model Monitor yang tidak memerlukan pengkodean. Anda juga memiliki fleksibilitas untuk memantau model dengan pengkodean untuk memberikan analisis khusus.

Model Monitor menyediakan jenis pemantauan berikut:

- Pantau kualitas data- Memantau penyimpangan dalam kualitas data.
- Monitor kualitas model- Memantau penyimpangan dalam metrik kualitas model, seperti akurasi.
- Monitor Bias Drift untuk Model dalam Produksi- Pantau bias dalam prediksi model Anda.
- Monitor Fitur Atribusi Drift untuk Model dalam Produksi- Memantau penyimpangan dalam atribusi fitur.

Topik

- Topik lanjutan
- FAQ Apple

Memantau Model dalam Produksi

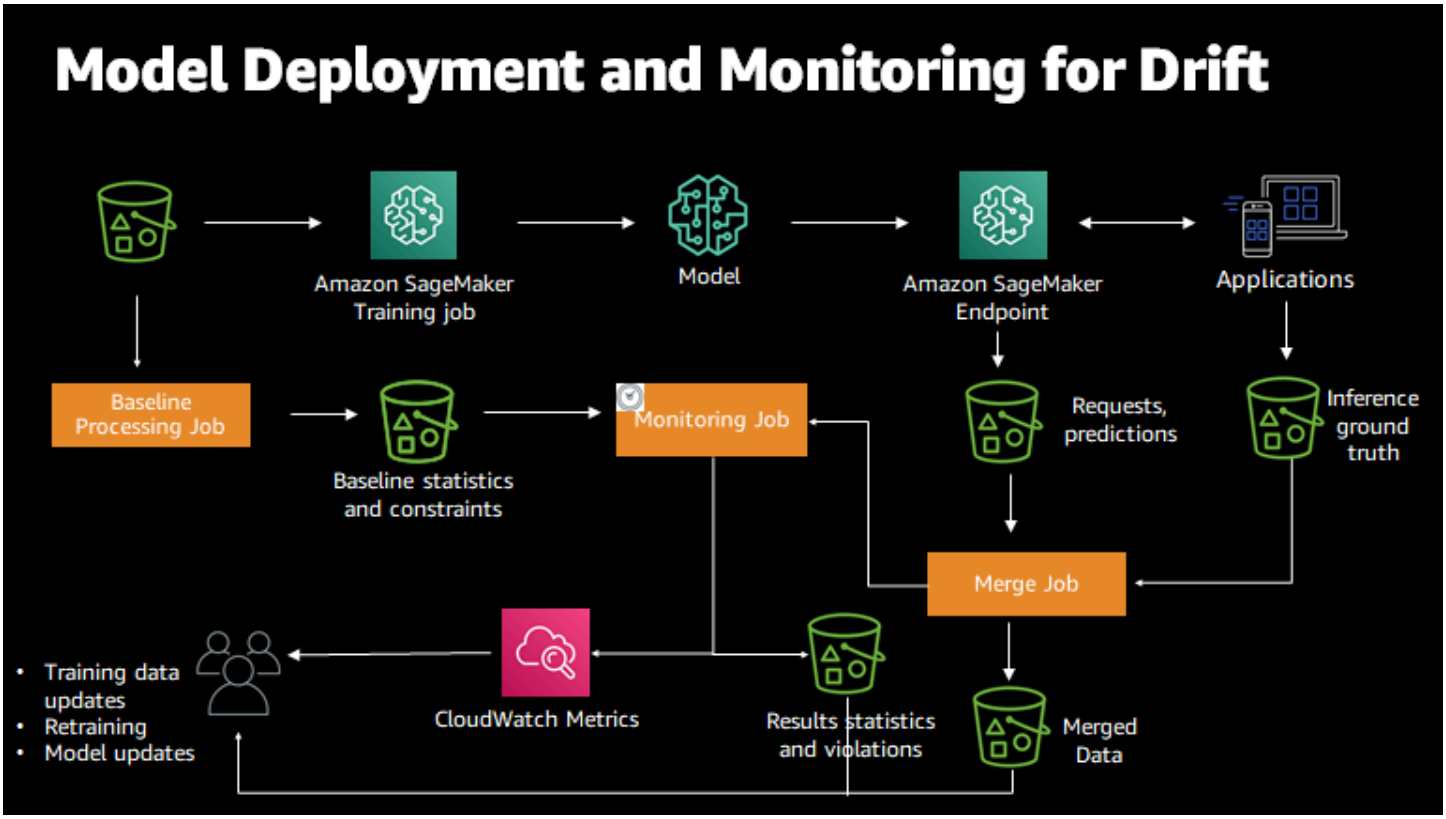
Setelah Anda menerapkan model ke lingkungan produksi Anda, gunakan monitor SageMaker model Amazon untuk terus memantau kualitas model pembelajaran mesin Anda secara real time. Monitor SageMaker model Amazon memungkinkan Anda menyiapkan sistem pemicu peringatan otomatis ketika ada penyimpangan dalam kualitas model, seperti penyimpangan data dan anomali. Amazon CloudWatch Logs mengumpulkan file log untuk memantau status model dan memberi tahu saat kualitas model mencapai ambang batas tertentu yang telah Anda tetapkan sebelumnya. CloudWatch menyimpan file log ke bucket Amazon S3 yang Anda tentukan. Deteksi penyimpangan model secara dini dan proaktif melalui produk monitor AWS model memungkinkan Anda mengambil tindakan segera untuk mempertahankan dan meningkatkan kualitas model yang Anda gunakan.

Untuk informasi selengkapnya tentang produk pemantauan SageMaker model, lihatMemantau data dan kualitas model.

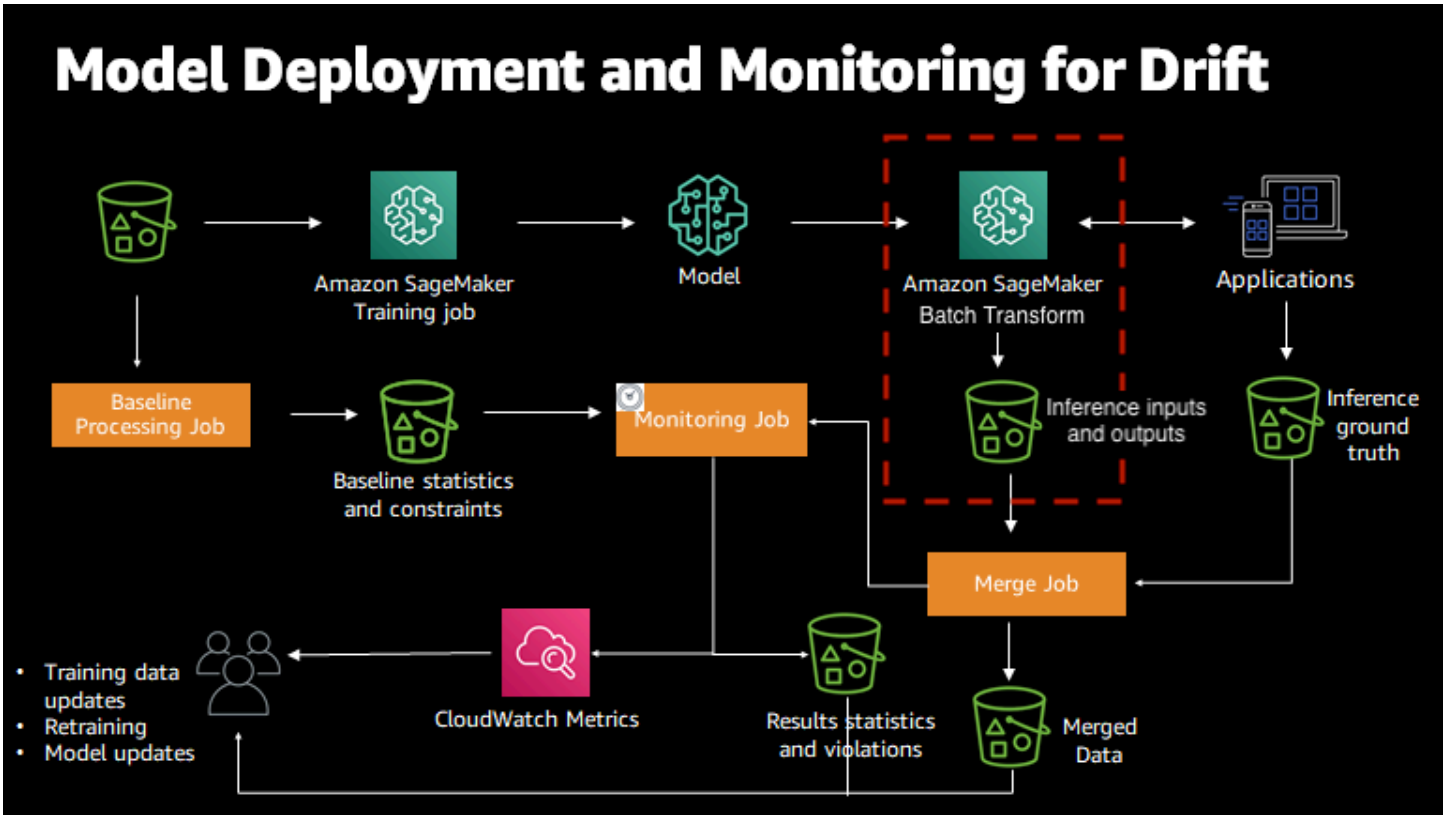
Untuk memulai perjalanan pembelajaran mesin Anda SageMaker, daftar AWS akun di Pengaturan SageMaker.

Bagaimana Model Monitor Bekerja

Amazon SageMaker Model Monitor secara otomatis memonitor model machine learning (ML) dalam produksi dan memberi tahu Anda saat masalah kualitas muncul. Model Monitor menggunakan aturan untuk mendeteksi penyimpangan dalam model Anda dan memberi tahu Anda ketika itu terjadi. Gambar berikut menunjukkan cara kerja proses ini jika model Anda diterapkan ke titik akhir waktu nyata.



Anda juga dapat menggunakan Model Monitor untuk memantau pekerjaan transformasi batch alih-alih titik akhir waktu nyata. Dalam hal ini, alih-alih menerima permintaan ke titik akhir dan melacak prediksi, Model Monitor akan memantau input dan output inferensi. Gambar berikut menggambarkan proses pemantauan pekerjaan transformasi batch.



Untuk mengaktifkan pemantauan model, Anda mengambil langkah-langkah berikut, yang mengikuti jalur data melalui berbagai proses pengumpulan, pemantauan, dan analisis data:

- Untuk titik akhir real-time, aktifkan titik akhir untuk menangkap data dari permintaan yang masuk ke model ML terlatih dan prediksi model yang dihasilkan.
- Untuk pekerjaan transformasi batch, aktifkan pengambilan data dari input dan output transformasi batch.
- Buat baseline dari dataset yang digunakan untuk melatih model. Garis dasar menghitung metrik dan menyarankan batasan untuk metrik. Prediksi real-time atau batch dari model Anda dibandingkan dengan batasan, dan dilaporkan sebagai pelanggaran jika berada di luar nilai yang dibatasi.
- Buat jadwal pemantauan yang menentukan data apa yang akan dikumpulkan, seberapa sering mengumpulkannya, bagaimana menganalisisnya, dan laporan mana yang akan dihasilkan.
- Periksa laporan, yang membandingkan data terbaru dengan baseline, dan perhatikan setiap pelanggaran yang dilaporkan serta untuk metrik dan pemberitahuan dari Amazon. CloudWatch

Catatan

- Model Monitor menghitung metrik model dan statistik hanya pada data tabular. Misalnya, model klasifikasi gambar yang mengambil gambar sebagai input dan mengeluarkan label berdasarkan gambar itu masih dapat dipantau. Model Monitor akan dapat menghitung metrik dan statistik untuk output, bukan input.
- Model Monitor saat ini hanya mendukung titik akhir yang menampung satu model dan tidak mendukung pemantauan titik akhir multi-model. Untuk informasi tentang penggunaan titik akhir multi-model, lihat. Host beberapa model dalam satu wadah di belakang satu titik akhir
- Model Monitor mendukung pemantauan saluran inferensi, tetapi menangkap dan menganalisis data dilakukan untuk seluruh pipa, bukan untuk kontainer individu dalam pipa.
- Untuk mencegah dampak pada permintaan inferensi, Data Capture berhenti menangkap permintaan pada tingkat penggunaan disk yang tinggi. Disarankan agar penggunaan disk Anda tetap di bawah 75% untuk memastikan pengambilan data terus menangkap permintaan.
- Jika Anda meluncurkan SageMaker Studio di VPC Amazon khusus, Anda perlu membuat titik akhir VPC untuk mengaktifkan Model Monitor untuk berkomunikasi dengan Amazon S3 dan. CloudWatch Untuk informasi tentang titik akhir VPC, lihat titik akhir VPC di Panduan Pengguna Amazon Virtual Private Cloud. Untuk informasi tentang meluncurkan SageMaker Studio di VPC kustom, lihat. Connect SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal

Model Monitor Contoh Notebook

Untuk contoh buku catatan yang membawa Anda melalui end-to-end alur kerja lengkap menggunakan Model Monitor dengan titik akhir real-time, lihat Pengantar Monitor SageMaker Model Amazon.

Untuk contoh buku catatan yang memvisualisasikan file statistics.json untuk eksekusi yang dipilih dalam jadwal pemantauan, lihat Visualisasi Monitor Model.

Untuk petunjuk yang menunjukkan cara membuat dan mengakses instance notebook Jupyter yang dapat Anda gunakan untuk menjalankan contoh, lihat. SageMaker Instans SageMaker Notebook Amazon Setelah Anda membuat instance notebook dan membukanya, pilih tab SageMaker Contoh

untuk melihat daftar semua SageMaker sampel. Untuk membuka buku catatan, pilih tab Gunakan buku catatan dan pilih Buat salinan.

Tangkapan Data

Untuk mencatat input ke titik akhir dan output inferensi dari model yang diterapkan ke Amazon S3, Anda dapat mengaktifkan fitur yang disebut Pengambilan Data. Data Capture biasanya digunakan untuk merekam informasi yang dapat digunakan untuk pelatihan, debugging, dan pemantauan. Amazon SageMaker Model Monitor secara otomatis mem-parsing data yang diambil ini dan membandingkan metrik dari data ini dengan baseline yang Anda buat untuk model tersebut. Untuk informasi selengkapnya tentang Model Monitor lihatMemantau data dan kualitas model.

Anda dapat menerapkan Pengambilan Data untuk mode monitor model real-time dan batch menggunakan AWS SDK for Python (Boto) atau Python SageMaker SDK. Untuk titik akhir real-time, Anda akan menentukan konfigurasi Pengambilan Data saat membuat titik akhir. Karena sifat titik akhir real-time Anda yang persisten, Anda dapat mengonfigurasi opsi tambahan untuk mengaktifkan atau menonaktifkan pengambilan data pada waktu-waktu tertentu, atau mengubah frekuensi pengambilan sampel. Anda juga dapat memilih untuk mengenkripsi data inferensi Anda.

Untuk pekerjaan transformasi batch, Anda dapat mengaktifkan Pengambilan Data jika Anda ingin menjalankan pemantauan model sesuai jadwal atau pemantauan model berkelanjutan untuk pekerjaan transformasi batch reguler dan berkala. Anda akan menentukan konfigurasi Data Capture Anda ketika Anda membuat pekerjaan transformasi batch Anda. Dalam konfigurasi ini, Anda memiliki opsi untuk mengaktifkan enkripsi atau menghasilkan ID inferensi dengan output Anda, yang membantu Anda mencocokkan data yang diambil dengan data Ground Truth.

Menangkap data dari titik akhir waktu nyata

> **Note**
>
> Untuk mencegah dampak pada permintaan inferensi, Data Capture berhenti menangkap permintaan pada tingkat penggunaan disk yang tinggi. Disarankan agar penggunaan disk Anda tetap di bawah 75% untuk memastikan pengambilan data terus menangkap permintaan.

Untuk menangkap data untuk titik akhir real-time Anda, Anda harus menerapkan model menggunakan SageMaker layanan hosting. Ini mengharuskan Anda membuat SageMaker model, menentukan konfigurasi titik akhir, dan membuat titik akhir HTTPS.

Langkah-langkah yang diperlukan untuk mengaktifkan pengambilan data serupa apakah Anda menggunakan AWS SDK for Python (Boto) atau SageMaker Python SDK. Jika Anda menggunakan AWS SDK, tentukan DataCaptureConfigkamus, bersama dengan bidang wajib, dalam CreateEndpointConfigmetode untuk mengaktifkan pengambilan data. Jika Anda menggunakan SageMaker Python SDK, impor DataCaptureConfigKelas dan inisialisasi instance dari kelas ini. Kemudian, berikan objek ini ke `DataCaptureConfig` parameter dalam `sagemaker.model.Model.deploy()` metode.

Untuk menggunakan cuplikan kode yang sedang berjalan, ganti *`teks placeholder yang dicetak miring`* dalam kode contoh dengan informasi Anda sendiri.

Cara mengaktifkan pengambilan data

Tentukan konfigurasi pengambilan data. Anda dapat menangkap payload permintaan, payload respons, atau keduanya dengan konfigurasi ini. Cuplikan kode yang sedang berlangsung menunjukkan cara mengaktifkan pengambilan data menggunakan dan Python AWS SDK for Python (Boto) SDK. SageMaker

> **Note**
>
> Anda tidak perlu menggunakan Model Monitor untuk menangkap permintaan atau muatan respons.

AWS SDK for Python (Boto)

Konfigurasikan data yang ingin Anda tangkap dengan DataCaptureConfigkamus saat Anda membuat titik akhir menggunakan `CreateEndpointConfig` metode ini. Setel `EnableCapture` ke nilai boolean True. Selain itu, berikan parameter wajib berikut:

- `EndpointConfigName`: nama konfigurasi titik akhir Anda. Anda akan menggunakan nama ini saat mengajukan `CreateEndpoint` permintaan.
- `ProductionVariants`: daftar model yang ingin Anda host di titik akhir ini. Tentukan tipe data kamus untuk setiap model.

- `DataCaptureConfig`: tipe data kamus tempat Anda menentukan nilai integer yang sesuai dengan persentase awal data untuk sampel (`InitialSamplingPercentage`), URI Amazon S3 tempat Anda ingin menyimpan data yang diambil, dan daftar opsi pengambilan `CaptureOptions` (). Tentukan salah satu `Input` atau `Output` untuk `CaptureMode` dalam `CaptureOptions` daftar.

Anda dapat secara opsional menentukan bagaimana SageMaker seharusnya menyandikan data yang diambil dengan meneruskan argumen pasangan nilai kunci ke kamus. `CaptureContentTypeHeader`

```
# Create an endpoint config name.
endpoint_config_name = '<endpoint-config-name>'

# The name of the production variant.
variant_name = '<name-of-production-variant>'

# The name of the model that you want to host.
# This is the name that you specified when creating the model.
model_name = '<The_name_of_your_model>'

instance_type = '<instance-type>'
#instance_type='ml.m5.xlarge' # Example

# Number of instances to launch initially.
initial_instance_count = <integer>

# Sampling percentage. Choose an integer value between 0 and 100
initial_sampling_percentage = <integer>




# The S3 URI containing the captured data
s3_capture_upload_path = 's3://<bucket-name>/<data_capture_s3_key>'

# Specify either Input, Output, or both
capture_modes = [ "Input",  "Output" ]
#capture_mode = [ "Input"] # Example - If you want to capture input only

endpoint_config_response = sagemaker_client.create_endpoint_config(
    EndpointConfigName=endpoint_config_name,
```

```
    # List of ProductionVariant objects, one for each model that you want to host at
 this endpoint.
    ProductionVariants=[
        {
            "VariantName": variant_name,
            "ModelName": model_name,
            "InstanceType": instance_type, # Specify the compute instance type.
            "InitialInstanceCount": initial_instance_count # Number of instances to
 launch initially.
        }
    ],
    DataCaptureConfig= {
        'EnableCapture': True, # Whether data should be captured or not.
        'InitialSamplingPercentage' : initial_sampling_percentage,
        'DestinationS3Uri': s3_capture_upload_path,
        'CaptureOptions': [{"CaptureMode" : capture_mode} for capture_mode in
 capture_modes] # Example - Use list comprehension to capture both Input and Output
    }
)
```

Untuk informasi selengkapnya tentang opsi konfigurasi titik akhir lainnya, lihat CreateEndpointConfigAPI di Panduan Referensi API SageMaker Layanan Amazon.

SageMaker Python SDK

Impor `DataCaptureConfig` Kelas dari modul sagemaker.model_monitor. Aktifkan pengambilan data dengan menyetel `EnableCapture` ke nilai `True` boolean.

Secara opsional memberikan argumen untuk parameter berikut:

- `SamplingPercentage`: nilai integer yang sesuai dengan persentase data untuk sampel. Jika Anda tidak memberikan persentase sampling, SageMaker akan mengambil sampel default 20 (20%) dari data Anda.
- `DestinationS3Uri`: URI Amazon S3 SageMaker akan digunakan untuk menyimpan data yang diambil. Jika Anda tidak menyediakannya, SageMaker akan menyimpan data yang ditangkap di`"s3://<default-session-bucket>/ model-monitor/data-capture"`.

```
from sagemaker.model_monitor import DataCaptureConfig

# Set to True to enable data capture
```

```
enable_capture = True

# Optional - Sampling percentage. Choose an integer value between 0 and 100
sampling_percentage = <int>
# sampling_percentage = 30 # Example 30%

# Optional - The S3 URI of stored captured-data location
s3_capture_upload_path = 's3://<bucket-name>/<data_capture_s3_key>'

# Specify either Input, Output or both.
capture_modes = ['REQUEST','RESPONSE'] # In this example, we specify both
# capture_mode = ['REQUEST'] # Example - If you want to only capture input.

# Configuration object passed in when deploying Models to SM endpoints
data_capture_config = DataCaptureConfig(
    enable_capture = enable_capture,
    sampling_percentage = sampling_percentage, # Optional
    destination_s3_uri = s3_capture_upload_path, # Optional
    capture_options = ["REQUEST", "RESPONSE"],
)
```

Deploy aplikasi Anda

Terapkan model Anda dan buat titik akhir HTTPS dengan `DataCapture` diaktifkan.

AWS SDK for Python (Boto3)

Berikan konfigurasi titik akhir ke SageMaker. Layanan meluncurkan instance komputasi ML dan menerapkan model atau model seperti yang ditentukan dalam konfigurasi.

Setelah Anda memiliki konfigurasi model dan titik akhir, gunakan `CreateEndpoint`API untuk membuat titik akhir Anda. Nama titik akhir harus unik dalam AWS Wilayah di AWS akun Anda.

Berikut ini membuat endpoint menggunakan konfigurasi endpoint yang ditentukan dalam permintaan. Amazon SageMaker menggunakan titik akhir untuk menyediakan sumber daya dan men-deploy model.

```
# The name of the endpoint. The name must be unique within an AWS Region in your AWS
 account.
endpoint_name = '<endpoint-name>'

# The name of the endpoint configuration associated with this endpoint.
```

```
endpoint_config_name='<endpoint-config-name>'

create_endpoint_response = sagemaker_client.create_endpoint(
                                                EndpointName=endpoint_name,

 EndpointConfigName=endpoint_config_name)
```

Untuk informasi selengkapnya, lihat CreateEndpointAPI.

SageMaker Python SDK

Tentukan nama untuk titik akhir Anda. Langkah ini bersifat opsional. Jika Anda tidak memberikan satu, SageMaker akan membuat nama yang unik untuk Anda:

```
from datetime import datetime

endpoint_name = f"DEMO-{datetime.utcnow():%Y-%m-%d-%H%M}"
print("EndpointName =", endpoint_name)
```

Terapkan model Anda ke titik akhir HTTPS real-time dengan metode bawaan `deploy()` objek Model. Berikan nama jenis instans Amazon EC2 untuk menerapkan model ini di `instance_type` bidang bersama dengan jumlah awal instance untuk menjalankan titik akhir pada bidang: `initial_instance_count`

```
initial_instance_count=<integer>
# initial_instance_count=1 # Example

instance_type='<instance-type>'
# instance_type='ml.m4.xlarge' # Example

# Uncomment if you did not define this variable in the previous step
#data_capture_config = <name-of-data-capture-configuration>

model.deploy(
    initial_instance_count=initial_instance_count,
    instance_type=instance_type,
    endpoint_name=endpoint_name,
    data_capture_config=data_capture_config
)
```

Melihat data

Buat objek prediktor dari SageMaker Python SDK Predictor Class. Anda akan menggunakan objek yang dikembalikan oleh `Predictor` Kelas untuk memanggil endpoint Anda di langkah masa depan. Berikan nama titik akhir Anda (didefinisikan sebelumnya sebagai`endpoint_name`), bersama dengan objek serializer dan deserializer untuk serializer dan deserializer, masing-masing. Untuk informasi tentang jenis serializer, lihat Kelas Serializers di Python SDK. SageMaker

```
from sagemaker.predictor import Predictor
from sagemaker.serializers import <Serializer>
from sagemaker.deserializers import <Deserializers>

predictor = Predictor(endpoint_name=endpoint_name,
                      serializer = <Serializer_Class>,
                      deserializer = <Deserializer_Class>)

# Example
#from sagemaker.predictor import Predictor
#from sagemaker.serializers import CSVSerializer
#from sagemaker.deserializers import JSONDeserializer

#predictor = Predictor(endpoint_name=endpoint_name,
#                      serializer=CSVSerializer(),
#                      deserializer=JSONDeserializer())
```

Dalam skenario contoh kode lanjutan, kami memanggil titik akhir dengan data validasi sampel yang telah kami simpan secara lokal dalam file CSV bernama. `validation_with_predictions` Set validasi sampel kami berisi label untuk setiap masukan.

Beberapa baris pertama dari pernyataan with pertama membuka file CSV set validasi, kemudian membagi setiap baris dalam file dengan karakter koma`","`, dan kemudian menyimpan dua objek yang dikembalikan ke dalam label dan variabel input_cols. Untuk setiap baris, input (`input_cols`) diteruskan ke metode `Predictor.predict()` bawaan objek variabel prediktor (`predictor`).

Misalkan model mengembalikan probabilitas. Probabilitas berkisar antara nilai integer 0 dan 1,0. Jika probabilitas yang dikembalikan oleh model lebih besar dari 80% (0,8) kami menetapkan prediksi label nilai integer 1. Jika tidak, kita menetapkan prediksi label nilai integer dari 0.

```
from time import sleep
```

```
validate_dataset = "validation_with_predictions.csv"

# Cut off threshold of 80%
cutoff = 0.8

limit = 200  # Need at least 200 samples to compute standard deviations
i = 0
with open(f"test_data/{validate_dataset}", "w") as validation_file:
    validation_file.write("probability,prediction,label\n")  # CSV header
    with open("test_data/validation.csv", "r") as f:
        for row in f:
            (label, input_cols) = row.split(",", 1)
            probability = float(predictor.predict(input_cols))
            prediction = "1" if probability > cutoff else "0"
            baseline_file.write(f"{probability},{prediction},{label}\n")
            i += 1
            if i > limit:
                break
            print(".", end="", flush=True)
            sleep(0.5)
print()
print("Done!")
```

Karena Anda mengaktifkan pengambilan data pada langkah sebelumnya, payload permintaan dan respons, bersama dengan beberapa data meta tambahan, disimpan di lokasi Amazon S3 yang Anda tentukan. `DataCaptureConfig` Pengiriman data pengambilan ke Amazon S3 dapat memerlukan beberapa menit.

Lihat data yang diambil dengan mencantumkan file pengambilan data yang disimpan di Amazon S3. Format jalur Amazon S3 adalah:. `s3:///`*`{endpoint-name}/{variant-name}/yyyy/mm/dd/hh/filename`*`.jsonl`

Berharap untuk melihat file yang berbeda dari periode waktu yang berbeda, diatur berdasarkan jam ketika pemanggilan terjadi. Jalankan yang berikut ini untuk mencetak konten file tangkapan tunggal:

```
print("\n".join(capture_file[-3:-1]))
```

Ini akan mengembalikan file berformat JSON-line SageMaker tertentu. Berikut ini adalah contoh respons yang diambil dari titik akhir real-time yang kami panggil menggunakan `csv/text` data:

```
{"captureData":{"endpointInput":{"observedContentType":"text/csv","mode":"INPUT",
```

```
"data":"69,0,153.7,109,194.0,105,256.1,114,14.1,6,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
"encoding":"CSV"},"endpointOutput":{"observedContentType":"text/csv;
 charset=utf-8","mode":"OUTPUT","data":"0.0254181120544672","encoding":"CSV"}},
"eventMetadata":{"eventId":"aaaaaaaa-bbbb-cccc-dddd-
eeeeeeeeeeee","inferenceTime":"2022-02-14T17:25:49Z"},"eventVersion":"0"}
{"captureData":{"endpointInput":{"observedContentType":"text/csv","mode":"INPUT",
"data":"94,23,197.1,125,214.5,136,282.2,103,9.5,5,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
"encoding":"CSV"},"endpointOutput":{"observedContentType":"text/csv;
 charset=utf-8","mode":"OUTPUT","data":"0.07675473392009735","encoding":"CSV"}},
"eventMetadata":{"eventId":"aaaaaaaa-bbbb-cccc-dddd-
eeeeeeeeeeee","inferenceTime":"2022-02-14T17:25:49Z"},"eventVersion":"0"}
```

Dalam contoh proses, `capture_file` objek adalah tipe daftar. Indeks elemen pertama dari daftar untuk melihat permintaan inferensi tunggal.

```
# The capture_file object is a list. Index the first element to view a single inference
 request
print(json.dumps(json.loads(capture_file[0]), indent=2))
```

Ini akan menghasilkan respons yang serupa dengan yang berikut ini. Nilai yang dikembalikan akan berbeda berdasarkan konfigurasi titik akhir, SageMaker model, dan data yang diambil:

```
{
  "captureData": {
    "endpointInput": {
      "observedContentType": "text/csv", # data MIME type
      "mode": "INPUT",
      "data":
 "50,0,188.9,94,203.9,104,151.8,124,11.6,8,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      "encoding": "CSV"
    },
    "endpointOutput": {
      "observedContentType": "text/csv; charset=character-encoding",
      "mode": "OUTPUT",
      "data": "0.023190177977085114",
      "encoding": "CSV"
    }
  },
  "eventMetadata": {
    "eventId": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "inferenceTime": "2022-02-14T17:25:06Z"
  },
```

```
  "eventVersion": "0"
}
```

Menangkap data dari pekerjaan transformasi batch

Langkah-langkah yang diperlukan untuk mengaktifkan pengambilan data untuk pekerjaan transformasi batch Anda serupa apakah Anda menggunakan SDK Python AWS SDK for Python (Boto) atau SageMaker Python. Jika Anda menggunakan AWS SDK, tentukan DataCaptureConfigkamus, bersama dengan bidang wajib, dalam `CreateTransformJob` metode untuk mengaktifkan pengambilan data. Jika Anda menggunakan SageMaker Python SDK, impor `BatchDataCaptureConfig` kelas dan inisialisasi instance dari kelas ini. Kemudian, berikan objek ini ke `batch_data_capture_config` parameter instance pekerjaan transformasi Anda.

Untuk menggunakan cuplikan kode berikut, ganti *`teks placeholder yang dicetak miring`* dalam kode contoh dengan informasi Anda sendiri.

Cara mengaktifkan pengambilan data

Tentukan konfigurasi pengambilan data saat Anda meluncurkan pekerjaan transformasi. Apakah Anda menggunakan AWS SDK for Python (Boto3) atau SageMaker Python SDK, Anda harus memberikan `DestinationS3Uri` argumen, yang merupakan direktori tempat Anda ingin pekerjaan transformasi mencatat data yang diambil. Opsional, Anda juga dapat mengatur parameter berikut:

- `KmsKeyId`: AWS KMS Kunci yang digunakan untuk mengenkripsi data yang diambil.
- `GenerateInferenceId`: Bendera Boolean yang, saat menangkap data, menunjukkan apakah Anda ingin pekerjaan transformasi menambahkan ID inferensi dan waktu ke output Anda. Ini berguna untuk pemantauan kualitas model, di mana Anda perlu menelan data Ground Truth. ID inferensi dan waktu membantu mencocokkan data yang diambil dengan data Ground Truth Anda.

AWS SDK for Python (Boto3)

Konfigurasikan data yang ingin Anda tangkap dengan DataCaptureConfigkamus saat Anda membuat pekerjaan transformasi menggunakan `CreateTransformJob` metode ini.

```
input_data_s3_uri = "s3://input_S3_uri"
output_data_s3_uri = "s3://output_S3_uri"
data_capture_destination = "s3://captured_data_S3_uri"
```

```
model_name = "model_name"

sm_client.create_transform_job(
    TransformJobName="transform_job_name",
    MaxConcurrentTransforms=2,
    ModelName=model_name,
    TransformInput={
        "DataSource": {
            "S3DataSource": {
                "S3DataType": "S3Prefix",
                "S3Uri": input_data_s3_uri,
            }
        },
        "ContentType": "text/csv",
        "CompressionType": "None",
        "SplitType": "Line",
    },
    TransformOutput={
        "S3OutputPath": output_data_s3_uri,
        "Accept": "text/csv",
        "AssembleWith": "Line",
    },
    TransformResources={
        "InstanceType": "ml.m4.xlarge",
        "InstanceCount": 1,
    },
    DataCaptureConfig={
       "DestinationS3Uri": data_capture_destination,
       "KmsKeyId": "kms_key",
       "GenerateInferenceId": True,
    }
    )
```

SageMaker Python SDK

Impor `BatchDataCaptureConfig` kelas dari sagemaker.model_monitor.

```
from sagemaker.transformer import Transformer
from sagemaker.inputs import BatchDataCaptureConfig

# Optional - The S3 URI of where to store captured data in S3
data_capture_destination = "s3://captured_data_S3_uri"

model_name = "model_name"
```

```
transformer = Transformer(model_name=model_name, ...)
transform_arg = transformer.transform(
    batch_data_capture_config=BatchDataCaptureConfig(
        destination_s3_uri=data_capture_destination,
        kms_key_id="kms_key",
        generate_inference_id=True,
    ),
    ...
)
```

Cara melihat data yang ditangkap

Setelah tugas transformasi selesai, data yang diambil akan dicatat di bawah yang `DestinationS3Uri` Anda berikan dengan konfigurasi pengambilan data. Ada dua subdirektori di bawah`DestinationS3Uri`, `/input` dan. `/output` Jika `DestinationS3Uri` yas`3://my-data-capture`, maka pekerjaan transformasi membuat direktori berikut:

- `s3://my-data-capture/input`: Data input yang diambil untuk pekerjaan transformasi.
- `s3://my-data-capture/output`: Data keluaran yang diambil untuk pekerjaan transformasi.

Untuk menghindari duplikasi data, data yang diambil di bawah dua direktori sebelumnya adalah manifes. Setiap manifes adalah file JSONL yang berisi lokasi Amazon S3 dari objek sumber. File manifes mungkin terlihat seperti contoh berikut:

```
# under "/input" directory
[
    {"prefix":"s3://input_S3_uri/"},
    "dummy_0.csv",
    "dummy_1.csv",
    "dummy_2.csv",
    ...
]

# under "/output" directory
[
    {"prefix":"s3://output_S3_uri/"},
    "dummy_0.csv.out",
    "dummy_1.csv.out",
    "dummy_2.csv.out",
```

```
    ...
]
```

Pekerjaan transformasi mengatur dan memberi label manifes ini dengan awalan *yyyy/mm/dd/hh* S3 untuk menunjukkan kapan mereka ditangkap. Ini membantu monitor model menentukan bagian data yang tepat untuk dianalisis. Misalnya, jika Anda memulai pekerjaan transformasi pada 2022-8-26 13PM UTC, maka data yang diambil diberi label dengan string awalan. `2022/08/26/13/`

InferenceId Generasi

Saat Anda mengonfigurasi `DataCaptureConfig` untuk pekerjaan transformasi, Anda dapat mengaktifkan bendera `GenerateInferenceId` Boolean. Ini sangat berguna ketika Anda perlu menjalankan kualitas model dan pekerjaan pemantauan bias model, di mana Anda memerlukan data Ground Truth yang dicerna pengguna. Monitor model mengandalkan ID inferensi untuk mencocokkan data yang diambil dan data Ground Truth. Untuk detail tambahan tentang konsumsi Ground Truth, lihat. Menelan Label Ground Truth dan Menggabungkannya Dengan Prediksi Saat `GenerateInferenceId` aktif, output transformasi menambahkan ID inferensi (UUID acak) serta waktu mulai pekerjaan transformasi di UTC untuk setiap catatan. Anda memerlukan dua nilai ini untuk menjalankan kualitas model dan pemantauan bias model. Saat Anda membuat data Ground Truth, Anda perlu memberikan ID inferensi yang sama agar sesuai dengan data keluaran. Saat ini, fitur ini mendukung output transformasi dalam format CSV, JSON, dan JSONL.

Jika output transformasi Anda dalam format CSV, file output terlihat seperti contoh berikut:

```
0, 1f1d57b1-2e6f-488c-8c30-db4e6d757861,2022-08-30T00:49:15Z
1, 22445434-0c67-45e9-bb4d-bd1bf26561e6,2022-08-30T00:49:15Z
...
```

Dua kolom terakhir adalah ID inferensi dan waktu mulai pekerjaan transformasi. Jangan memodifikasi mereka. Kolom yang tersisa adalah output pekerjaan transformasi Anda.

Jika output transformasi Anda dalam format JSON atau JSONL, file output terlihat seperti contoh berikut:

```
{"output": 0, "SageMakerInferenceId": "1f1d57b1-2e6f-488c-8c30-db4e6d757861",
 "SageMakerInferenceTime": "2022-08-30T00:49:15Z"}
{"output": 1, "SageMakerInferenceId": "22445434-0c67-45e9-bb4d-bd1bf26561e6",
 "SageMakerInferenceTime": "2022-08-30T00:49:15Z"}
...
```

Ada dua bidang tambahan yang dicadangkan, `SageMakerInferenceId` dan`SageMakerInferenceTime`. Jangan memodifikasi bidang ini jika Anda perlu menjalankan kualitas model atau pemantauan bias model — Anda memerlukannya untuk pekerjaan gabungan.

Pantau kualitas data

Pemantauan kualitas data secara otomatis memantau model pembelajaran mesin (ML) dalam produksi dan memberi tahu Anda saat masalah kualitas data muncul. Model ML dalam produksi harus membuat prediksi pada data kehidupan nyata yang tidak dikuratori dengan hati-hati seperti kebanyakan kumpulan data pelatihan. Jika sifat statistik dari data yang diterima model Anda saat dalam produksi menjauh dari sifat data dasar yang dilatihnya, model mulai kehilangan akurasi dalam prediksinya. Amazon SageMaker Model Monitor menggunakan aturan untuk mendeteksi penyimpangan data dan memberi tahu Anda ketika itu terjadi. Untuk memantau kualitas data, ikuti langkah-langkah ini:

- Aktifkan tangkapan data. Ini menangkap input dan output inferensi dari titik akhir inferensi waktu nyata atau pekerjaan transformasi batch dan menyimpan data di Amazon S3. Untuk informasi selengkapnya, lihat Tangkapan Data.
- Membuat dasar Pada langkah ini, Anda menjalankan pekerjaan dasar yang menganalisis kumpulan data input yang Anda berikan. Baseline menghitung batasan skema dasar dan statistik untuk setiap fitur menggunakan Deequ, pustaka open source yang dibangun di atas Apache Spark, yang digunakan untuk mengukur kualitas data dalam kumpulan data besar. Untuk informasi selengkapnya, lihat Membuat dasar.
- Menentukan dan menjadwalkan pekerjaan pemantauan kualitas data. Untuk informasi spesifik dan contoh kode pekerjaan pemantauan kualitas data, lihatJadwalkan pekerjaan pemantauan kualitas data. Untuk informasi umum tentang cara memantau pekerjaan, lihatJadwalkan pekerjaan pemantauan.
  - Secara opsional gunakan skrip preprocessing dan postprocessing untuk mengubah data yang keluar dari analisis kualitas data Anda. Untuk informasi selengkapnya, lihat Pracemrosesan dan Pascapemrosesan.
- Lihat metrik kualitas data. Untuk informasi selengkapnya, lihat Skema untuk Statistik (file statistik.json).
- Integrasikan pemantauan kualitas data dengan Amazon CloudWatch. Untuk informasi selengkapnya, lihat CloudWatch Metrik.

- Menafsirkan hasil tugas pemantauan. Untuk informasi selengkapnya, lihat Menafsirkan hasil.
- Gunakan SageMaker Studio untuk mengaktifkan pemantauan kualitas data dan memvisualisasikan hasil jika Anda menggunakan titik akhir real-time. Untuk informasi selengkapnya, lihat Visualisasikan hasil untuk titik akhir real-time di Amazon Studio SageMaker .

> **Note**
>
> Model Monitor menghitung metrik model dan statistik hanya pada data tabular. Misalnya, model klasifikasi gambar yang mengambil gambar sebagai input dan mengeluarkan label berdasarkan gambar itu masih dapat dipantau. Model Monitor akan dapat menghitung metrik dan statistik untuk output, bukan input.

Topik

Membuat dasar

Perhitungan dasar statistik dan kendala diperlukan sebagai standar di mana penyimpangan data dan masalah kualitas data lainnya dapat dideteksi. Model Monitor menyediakan wadah bawaan yang menyediakan kemampuan untuk menyarankan kendala secara otomatis untuk input CSV dan JSON datar. sagemaker-model-monitor-analyzerContainer ini juga memberi Anda berbagai kemampuan pemantauan model, termasuk validasi kendala terhadap baseline, dan memancarkan metrik Amazon. CloudWatch Wadah ini didasarkan pada Spark versi 3.3.0 dan dibangun dengan Deequ versi 2.0.2. Semua nama kolom dalam dataset dasar Anda harus sesuai dengan Spark. Untuk nama kolom, gunakan hanya karakter huruf kecil, dan _ sebagai satu-satunya karakter khusus.

Dataset pelatihan yang Anda gunakan untuk melatih model biasanya merupakan kumpulan data dasar yang baik. Skema data kumpulan data pelatihan dan skema kumpulan data inferensi harus sama persis (jumlah dan urutan fitur). Perhatikan bahwa kolom prediksi/output diasumsikan sebagai kolom pertama dalam kumpulan data pelatihan. Dari kumpulan data pelatihan, Anda dapat meminta

SageMaker untuk menyarankan serangkaian batasan dasar dan menghasilkan statistik deskriptif untuk menjelajahi data. Untuk contoh ini, unggah kumpulan data pelatihan yang digunakan untuk melatih model yang telah dilatih sebelumnya yang termasuk dalam contoh ini. Jika Anda sudah menyimpan kumpulan data pelatihan di Amazon S3, Anda dapat mengarahkannya secara langsung.

Membuat baseline dari dataset pelatihan

Saat data pelatihan Anda siap dan disimpan di Amazon S3, mulailah pekerjaan pemrosesan dasar dengan menggunakan `DefaultModelMonitor.suggest_baseline(..)` Amazon Python SDK. SageMaker Ini menggunakan Amazon SageMaker Model Monitor wadah bawaan yang menghasilkan statistik dasar dan menyarankan batasan dasar untuk kumpulan data dan menuliskannya ke lokasi yang Anda tentukan. `output_s3_uri`

```
from sagemaker.model_monitor import DefaultModelMonitor
from sagemaker.model_monitor.dataset_format import DatasetFormat

my_default_monitor = DefaultModelMonitor(
    role=role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
    volume_size_in_gb=20,
    max_runtime_in_seconds=3600,
)

my_default_monitor.suggest_baseline(
    baseline_dataset=baseline_data_uri+'/training-dataset-with-header.csv',
    dataset_format=DatasetFormat.csv(header=True),
    output_s3_uri=baseline_results_uri,
    wait=True
)
```

> **Note**
>
> Jika Anda memberikan nama fitur/kolom dalam kumpulan data pelatihan sebagai baris pertama dan mengatur `header=True` opsi seperti yang ditunjukkan pada contoh kode sebelumnya, SageMaker gunakan nama fitur dalam file batasan dan statistik.

Statistik dasar untuk kumpulan data terkandung dalam file statistics.json dan batasan dasar yang disarankan terkandung dalam file constraints.json di lokasi yang Anda tentukan. `output_s3_uri`

File Output untuk Statistik dan Kendala Set Data Tabular

Format nama file	Deskripsi
`statistics.json`	File ini diharapkan memiliki statistik kolumnar untuk setiap fitur dalam kumpulan data yang dianalisis. Untuk informasi selengkapnya tentang skema file ini, lihatSkema untuk Statistik (file statistik.json).
`constraints.json`	File ini diharapkan memiliki kendala pada fitur yang diamati. Untuk informasi selengkap nya tentang skema file ini, lihatSkema untuk Kendala (file kendala json).

Amazon SageMaker Python SDK menyediakan fungsi kenyamanan yang dijelaskan untuk menghasilkan statistik dan batasan dasar. Tetapi jika Anda ingin memanggil pekerjaan pemrosesan secara langsung untuk tujuan ini, Anda perlu mengatur `Environment` peta seperti yang ditunjukkan pada contoh berikut:

```
"Environment": {
    "dataset_format": "{\"csv\”: { \”header\”: true}",
    "dataset_source": "/opt/ml/processing/sm_input",
    "output_path": "/opt/ml/processing/sm_output",
    "publish_cloudwatch_metrics": "Disabled",
}
```

Jadwalkan pekerjaan pemantauan kualitas data

Setelah Anda membuat baseline Anda, Anda dapat memanggil `create_monitoring_schedule()` metode instance `DefaultModelMonitor` kelas Anda untuk menjadwalkan monitor kualitas data per jam. Bagian berikut menunjukkan cara membuat monitor kualitas data untuk model yang diterapkan ke titik akhir real-time serta untuk pekerjaan transformasi batch.

> **Important**
>
> Anda dapat menentukan input transformasi batch atau input titik akhir, tetapi tidak keduanya, saat Anda membuat jadwal pemantauan.

Pemantauan kualitas data untuk model yang digunakan ke titik akhir waktu nyata

Untuk menjadwalkan monitor kualitas data untuk titik akhir real-time, teruskan `EndpointInput` instance Anda ke `endpoint_input` argumen `DefaultModelMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

data_quality_model_monitor = DefaultModelMonitor(
   role=sagemaker.get_execution_role(),
   ...
)

schedule = data_quality_model_monitor.create_monitoring_schedule(
   monitor_schedule_name=schedule_name,
   post_analytics_processor_script=s3_code_postprocessor_uri,
   output_s3_uri=s3_report_path,
   schedule_cron_expression=CronExpressionGenerator.hourly(),
   statistics=data_quality_model_monitor.baseline_statistics(),
   constraints=data_quality_model_monitor.suggested_constraints(),
   schedule_cron_expression=CronExpressionGenerator.hourly(),
   enable_cloudwatch_metrics=True,
   endpoint_input=EndpointInput(
        endpoint_name=endpoint_name,
        destination="/opt/ml/processing/input/endpoint",
   )
)
```

Pemantauan kualitas data untuk pekerjaan transformasi batch

Untuk menjadwalkan monitor kualitas data untuk pekerjaan transformasi batch, teruskan `BatchTransformInput` instance Anda ke `batch_transform_input` argumen `DefaultModelMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

data_quality_model_monitor = DefaultModelMonitor(
   role=sagemaker.get_execution_role(),
   ...
)

schedule = data_quality_model_monitor.create_monitoring_schedule(
```

```
    monitor_schedule_name=mon_schedule_name,
    batch_transform_input=BatchTransformInput(
        data_captured_destination_s3_uri=s3_capture_upload_path,
        destination="/opt/ml/processing/input",
        dataset_format=MonitoringDatasetFormat.csv(header=False),
    ),
    output_s3_uri=s3_report_path,
    statistics= statistics_path,
    constraints = constraints_path,
    schedule_cron_expression=CronExpressionGenerator.hourly(),
    enable_cloudwatch_metrics=True,
)
```

Skema untuk Statistik (file statistik.json)

Amazon SageMaker Model Monitor kontainer bawaan menghitung per kolom/statistik fitur. Statistik dihitung untuk dataset dasar dan juga untuk dataset saat ini yang sedang dianalisis.

```
{
    "version": 0,
    # dataset level stats
    "dataset": {
        "item_count": number
    },
    # feature level stats
    "features": [
        {
            "name": "feature-name",
            "inferred_type": "Fractional" | "Integral",
            "numerical_statistics": {
                "common": {
                    "num_present": number,
                    "num_missing": number
                },
                "mean": number,
                "sum": number,
                "std_dev": number,
                "min": number,
                "max": number,
                "distribution": {
                    "kll": {
                        "buckets": [
                            {
```

```
                            "lower_bound": number,
                            "upper_bound": number,
                            "count": number
                        }
                    ],
                    "sketch": {
                        "parameters": {
                            "c": number,
                            "k": number
                        },
                        "data": [
                            [
                                num,
                                num,
                                num,
                                num
                            ],
                            [
                                num,
                                num
                            ][
                                num,
                                num
                            ]
                        ]
                    }#sketch
                }#KLL
            }#distribution
        }#num_stats
    },
    {
        "name": "feature-name",
        "inferred_type": "String",
        "string_statistics": {
            "common": {
                "num_present": number,
                "num_missing": number
            },
            "distinct_count": number,
            "distribution": {
                "categorical": {
                     "buckets": [
                            {
                                "value": "string",
```

```
                                    "count": number
                                }
                            ]
                        }
                    }
                },
                #provision for custom stats
            }
        ]
}
```

Perhatikan hal berikut:

- Kontainer prebuilt menghitung sketsa KLL, yang merupakan sketsa kuantil kompak.
- Secara default, kami mewujudkan distribusi dalam 10 ember. Saat ini tidak dapat dikonfigurasi.

CloudWatch Metrik

Anda dapat menggunakan wadah Monitor SageMaker Model Amazon bawaan untuk CloudWatch metrik. Ketika `emit_metrics` opsi ada `Enabled` di file batasan dasar, SageMaker memancarkan metrik ini untuk setiap fitur/kolom yang diamati dalam kumpulan data di namespace berikut:

- `For real-time endpoints: /aws/sagemaker/Endpoints/data-metric`namespace dengan `EndpointName` dan `ScheduleName` dimensi.
- `For batch transform jobs: /aws/sagemaker/ModelMonitoring/data-metric`namespace dengan `MonitoringSchedule` dimensi.

Untuk bidang numerik, wadah bawaan memancarkan metrik berikut: CloudWatch

- Metrik: Maks → kueri untuk `MetricName: feature_data_{feature_name}, Stat: Max`
- Metrik: Min → kueri untuk `MetricName: feature_data_{feature_name}, Stat: Min`
- Metrik: Jumlah → kueri untuk `MetricName: feature_data_{feature_name}, Stat: Sum`
- Metrik: SampleCount → kueri untuk `MetricName: feature_data_{feature_name}, Stat: SampleCount`
- Metrik: Rata-rata → kueri untuk `MetricName: feature_data_{feature_name}, Stat: Average`

Untuk bidang numerik dan string, wadah bawaan memancarkan metrik berikut: CloudWatch

- Metrik: Kelengkapan → kueri untuk `MetricName: feature_non_null_{feature_name}, Stat: Sum`
- Metrik: Baseline Drift → kueri untuk `MetricName: feature_baseline_drift_{feature_name}, Stat: Sum`

Skema untuk Pelanggaran (file constraint_violations.json)

File pelanggaran dihasilkan sebagai output dari a`MonitoringExecution`, yang mencantumkan hasil evaluasi kendala (ditentukan dalam file constraints.json) terhadap kumpulan data saat ini yang dianalisis. Container prebuilt Amazon SageMaker Model Monitor menyediakan pemeriksaan pelanggaran berikut.

```
{
    "violations": [{
      "feature_name" : "string",
      "constraint_check_type" :
              "data_type_check",
            | "completeness_check",
            | "baseline_drift_check",
            | "missing_column_check",
            | "extra_column_check",
            | "categorical_values_check"
      "description" : "string"
    }]
}
```

Tipe Pelanggaran yang Dipantau

Jenis Pemeriksaan Pelanggaran	Deskripsi
`data_type_check`	Jika tipe data dalam eksekusi saat ini tidak sama dengan pada dataset dasar, pelanggaran ini ditandai. Selama langkah dasar, kendala yang dihasilkan menyarankan tipe data yang disimpulkan untuk setiap kolom. `monitoring_config.datatype_check_threshold` Parameter

Jenis Pemeriksaan Pelanggaran	Deskripsi
	dapat disetel untuk menyesuaikan ambang batas saat ditandai sebagai pelanggaran.
`completeness_check`	Jika kelengkapan (% item non-null) yang diamati dalam eksekusi saat ini melebihi ambang batas yang ditentukan dalam ambang kelengkapan yang ditentukan per fitur, pelanggaran ini ditandai. Selama langkah dasar, kendala yang dihasilkan menunjukkan nilai kelengkapan.
`baseline_drift_check`	Jika jarak distribusi yang dihitung antara kumpulan data saat ini dan baseline lebih dari ambang batas yang ditentukan dalam`monitoring_config.comparison_threshold` , pelanggaran ini ditandai.
`missing_column_check`	Jika jumlah kolom dalam kumpulan data saat ini kurang dari jumlah dalam kumpulan data dasar, pelanggaran ini ditandai.
`extra_column_check`	Jika jumlah kolom dalam kumpulan data saat ini lebih dari jumlah di baseline, pelanggaran ini ditandai.
`categorical_values_check`	Jika ada lebih banyak nilai yang tidak diketahui dalam kumpulan data saat ini daripada di kumpulan data dasar, pelanggaran ini ditandai. Nilai ini ditentukan oleh ambang batas di`monitoring_config.domain_content_threshold` .

Monitor kualitas model

Pekerjaan pemantauan kualitas model memantau kinerja model dengan membandingkan prediksi yang dibuat model dengan label Ground Truth aktual yang coba diprediksi oleh model. Untuk melakukannya, pemantauan kualitas model menggabungkan data yang diambil dari inferensi real-time atau batch dengan label aktual yang Anda simpan di bucket Amazon S3, lalu membandingkan prediksi dengan label sebenarnya.

Untuk mengukur kualitas model, monitor model menggunakan metrik yang bergantung pada jenis masalah ML. Misalnya, jika model Anda untuk masalah regresi, salah satu metrik yang dievaluasi adalah mean square error (mse). Untuk informasi tentang semua metrik yang digunakan untuk jenis masalah ML yang berbeda, lihatMetrik Kualitas Model.

Pemantauan kualitas model mengikuti langkah yang sama seperti pemantauan kualitas data, tetapi menambahkan langkah tambahan untuk menggabungkan label aktual dari Amazon S3 dengan prediksi yang diambil dari titik akhir inferensi waktu nyata atau pekerjaan transformasi batch. Untuk memantau kualitas model, ikuti langkah-langkah ini:

- Aktifkan tangkapan data. Ini menangkap input dan output inferensi dari titik akhir inferensi waktu nyata atau pekerjaan transformasi batch dan menyimpan data di Amazon S3. Untuk informasi selengkapnya, lihat Tangkapan Data.
- Membuat dasar Pada langkah ini, Anda menjalankan pekerjaan dasar yang membandingkan prediksi dari model dengan label Ground Truth dalam dataset dasar. Pekerjaan dasar secara otomatis membuat aturan statistik dasar dan kendala yang menentukan ambang batas yang digunakan untuk mengevaluasi kinerja model. Untuk informasi selengkapnya, lihat Buat Model Quality Baseline.
- Menentukan dan menjadwalkan tugas pemantauan kualitas model. Untuk informasi spesifik dan contoh kode pekerjaan pemantauan kualitas model, lihat. Jadwal Model Pekerjaan Pemantauan Kualitas Untuk informasi umum tentang pekerjaan pemantauan, lihatJadwalkan pekerjaan pemantauan.
- Label Ground Truth yang memantau model digabungkan dengan data prediksi yang diambil dari titik akhir inferensi waktu nyata atau pekerjaan transformasi batch. Untuk informasi selengkapnya, lihat Menelan Label Ground Truth dan Menggabungkannya Dengan Prediksi.
- Integrasikan pemantauan kualitas model dengan Amazon CloudWatch. Untuk informasi selengkapnya, lihat CloudWatch Metrik Kualitas Model.
- Menafsirkan hasil tugas pemantauan. Untuk informasi selengkapnya, lihat Menafsirkan hasil.

- Gunakan SageMaker Studio untuk mengaktifkan pemantauan kualitas model dan memvisualisasikan hasil. Untuk informasi selengkapnya, lihat Visualisasikan hasil untuk titik akhir real-time di Amazon Studio SageMaker .

Topik

Buat Model Quality Baseline

Buat pekerjaan dasar yang membandingkan prediksi model Anda dengan label kebenaran dasar dalam kumpulan data dasar yang telah Anda simpan di Amazon S3. Biasanya, Anda menggunakan dataset pelatihan sebagai dataset dasar. Pekerjaan dasar menghitung metrik untuk model dan menyarankan kendala untuk digunakan untuk memantau penyimpangan kualitas model.

Untuk membuat pekerjaan dasar, Anda harus memiliki kumpulan data yang berisi prediksi dari model Anda bersama dengan label yang mewakili Ground Truth untuk data Anda.

Untuk membuat pekerjaan dasar, gunakan `ModelQualityMonitor` kelas yang disediakan oleh SageMaker Python SDK, dan selesaikan langkah-langkah berikut.

Untuk membuat pekerjaan dasar kualitas model

1. Pertama, buat instans `ModelQualityMonitor` kelas. Cuplikan kode berikut menunjukkan cara melakukannya.

```
from sagemaker import get_execution_role, session, Session
from sagemaker.model_monitor import ModelQualityMonitor

role = get_execution_role()
session = Session()

model_quality_monitor = ModelQualityMonitor(
    role=role,
    instance_count=1,
```

```
    instance_type='ml.m5.xlarge',
    volume_size_in_gb=20,
    max_runtime_in_seconds=1800,
    sagemaker_session=session
)
```

2. Sekarang panggil `suggest_baseline` metode `ModelQualityMonitor` objek untuk menjalankan pekerjaan dasar. Cuplikan kode berikut mengasumsikan bahwa Anda memiliki kumpulan data dasar yang berisi prediksi dan label yang disimpan di Amazon S3.

```
baseline_job_name = "MyBaseLineJob"
job = model_quality_monitor.suggest_baseline(
    job_name=baseline_job_name,
    baseline_dataset=baseline_dataset_uri, # The S3 location of the validation
 dataset.
    dataset_format=DatasetFormat.csv(header=True),
    output_s3_uri = baseline_results_uri, # The S3 location to store the results.
    problem_type='BinaryClassification',
    inference_attribute= "prediction", # The column in the dataset that contains
 predictions.
    probability_attribute= "probability", # The column in the dataset that contains
 probabilities.
    ground_truth_attribute= "label" # The column in the dataset that contains
 ground truth labels.
)
job.wait(logs=False)
```

3. Setelah pekerjaan dasar selesai, Anda dapat melihat batasan yang dihasilkan oleh pekerjaan. Pertama, dapatkan hasil pekerjaan dasar dengan memanggil `latest_baselining_job` metode objek. `ModelQualityMonitor`

```
baseline_job = model_quality_monitor.latest_baselining_job
```

4. Pekerjaan dasar menunjukkan kendala, yang merupakan ambang batas untuk metrik yang memodelkan ukuran monitor. Jika metrik melampaui ambang batas yang disarankan, Model Monitor melaporkan pelanggaran. Untuk melihat kendala yang dihasilkan oleh pekerjaan dasar, panggil `suggested_constraints` metode pekerjaan dasar. Cuplikan kode berikut memuat batasan untuk model klasifikasi biner ke dalam kerangka data Pandas.

```
import pandas as pd
pd.DataFrame(baseline_job.suggested_constraints().body_dict["binary_classification_constrai
```

Kami menyarankan Anda melihat batasan yang dihasilkan dan memodifikasinya seperlunya sebelum menggunakannya untuk pemantauan. Misalnya, jika kendala terlalu agresif, Anda mungkin mendapatkan lebih banyak peringatan untuk pelanggaran daripada yang Anda inginkan.

Jika kendala Anda berisi angka yang dinyatakan dalam notasi ilmiah, Anda perlu mengubahnya menjadi float. Contoh skrip preprocessing python berikut menunjukkan cara mengonversi angka dalam notasi ilmiah menjadi float.

```
import csv

def fix_scientific_notation(col):
    try:
        return format(float(col), "f")
    except:
        return col

def preprocess_handler(csv_line):
    reader = csv.reader([csv_line])
    csv_record = next(reader)
    #skip baseline header, change HEADER_NAME to the first column's name
    if csv_record[0] == “HEADER_NAME”:
       return []
    return { str(i).zfill(20) : fix_scientific_notation(d) for i, d in
 enumerate(csv_record)}
```

Anda dapat menambahkan skrip pra-pemrosesan ke baseline atau jadwal pemantauan sebagai`record_preprocessor_script`, seperti yang didefinisikan dalam dokumentasi Model Monitor.

5. Ketika Anda puas dengan kendala, berikan mereka sebagai `constraints` parameter saat Anda membuat jadwal pemantauan. Untuk informasi selengkapnya, lihat Jadwal Model Pekerjaan Pemantauan Kualitas.

Batasan dasar yang disarankan terkandung dalam file constraints.json di lokasi yang Anda tentukan. `output_s3_uri` Untuk informasi tentang skema untuk file ini di. Skema untuk Kendala (file kendala json)

Jadwal Model Pekerjaan Pemantauan Kualitas

Setelah Anda membuat baseline Anda, Anda dapat memanggil `create_monitoring_schedule()` metode instance `ModelQualityMonitor` kelas Anda untuk menjadwalkan monitor kualitas model per jam. Bagian berikut menunjukkan cara membuat monitor kualitas model untuk model yang diterapkan ke titik akhir real-time serta untuk pekerjaan transformasi batch.

> **Important**
>
> Anda dapat menentukan input transformasi batch atau input titik akhir, tetapi tidak keduanya, saat Anda membuat jadwal pemantauan.

Tidak seperti pemantauan kualitas data, Anda perlu menyediakan label Ground Truth jika Anda ingin memantau kualitas model. Namun, label Ground Truth bisa ditunda. Untuk mengatasinya, tentukan offset saat Anda membuat jadwal pemantauan.

Offset monitor model

Pekerjaan kualitas model meliputi `StartTimeOffset` dan`EndTimeOffset`, yang merupakan bidang `ModelQualityJobInput` parameter `create_model_quality_job_definition` metode yang berfungsi sebagai berikut:

- `StartTimeOffset`- Jika ditentukan, pekerjaan mengurangi waktu ini dari waktu mulai.
- `EndTimeOffset`- Jika ditentukan, pekerjaan mengurangi waktu ini dari waktu berakhir.

Format offset adalah, misalnya, -PT7H, di mana 7H adalah 7 jam. Anda dapat menggunakan -PT #H atau -P #D, di mana H = jam, D = hari, dan m = menit, dan # adalah nomornya. Selain itu, offset harus dalam format durasi ISO 8601.

Misalnya, jika Ground Truth Anda mulai masuk setelah 1 hari, tetapi tidak selesai selama seminggu, atur `StartTimeOffset` ke `-P8D` dan `EndTimeOffset` ke`-P1D`. Kemudian, jika Anda menjadwalkan pekerjaan untuk dijalankan`2020-01-09T13:00`, itu menganalisis data dari antara `2020-01-01T13:00` dan`2020-01-08T13:00`.

> **Important**
>
> Irama jadwal harus sedemikian rupa sehingga satu eksekusi selesai sebelum eksekusi berikutnya dimulai, yang memungkinkan pekerjaan penggabungan Ground Truth dan

pekerjaan pemantauan dari eksekusi selesai. Runtime maksimum eksekusi dibagi antara dua pekerjaan, jadi untuk pekerjaan pemantauan kualitas model per jam, nilai yang `MaxRuntimeInSeconds` ditentukan sebagai bagian dari `StoppingCondition` harus tidak lebih dari 1800.

Pemantauan kualitas model untuk model yang digunakan ke titik akhir waktu nyata

Untuk menjadwalkan monitor kualitas model untuk titik akhir real-time, teruskan `EndpointInput` instance Anda ke `endpoint_input` argumen `ModelQualityMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

model_quality_model_monitor = ModelQualityMonitor(
   role=sagemaker.get_execution_role(),
   ...
)

schedule = model_quality_model_monitor.create_monitoring_schedule(
   monitor_schedule_name=schedule_name,
   post_analytics_processor_script=s3_code_postprocessor_uri,
   output_s3_uri=s3_report_path,
   schedule_cron_expression=CronExpressionGenerator.hourly(),
   statistics=model_quality_model_monitor.baseline_statistics(),
   constraints=model_quality_model_monitor.suggested_constraints(),
   schedule_cron_expression=CronExpressionGenerator.hourly(),
   enable_cloudwatch_metrics=True,
   endpoint_input=EndpointInput(
        endpoint_name=endpoint_name,
        destination="/opt/ml/processing/input/endpoint",
        start_time_offset="-PT2D",
        end_time_offset="-PT1D",
    )
)
```

Pemantauan kualitas model untuk pekerjaan transformasi batch

Untuk menjadwalkan monitor kualitas model untuk pekerjaan transformasi batch, teruskan `BatchTransformInput` instance Anda ke `batch_transform_input` argumen `ModelQualityMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

model_quality_model_monitor = ModelQualityMonitor(
   role=sagemaker.get_execution_role(),
   ...
)

schedule = model_quality_model_monitor.create_monitoring_schedule(
    monitor_schedule_name=mon_schedule_name,
    batch_transform_input=BatchTransformInput(
        data_captured_destination_s3_uri=s3_capture_upload_path,
        destination="/opt/ml/processing/input",
        dataset_format=MonitoringDatasetFormat.csv(header=False),
        # the column index of the output representing the inference probablity
        probability_attribute="0",
        # the threshold to classify the inference probablity to class 0 or 1 in
        # binary classification problem
        probability_threshold_attribute=0.5,
        # look back 6 hour for transform job outputs.
        start_time_offset="-PT6H",
        end_time_offset="-PT0H"
    ),
    ground_truth_input=gt_s3_uri,
    output_s3_uri=s3_report_path,
    problem_type="BinaryClassification",
    constraints = constraints_path,
    schedule_cron_expression=CronExpressionGenerator.hourly(),
    enable_cloudwatch_metrics=True,
)
```

Menelan Label Ground Truth dan Menggabungkannya Dengan Prediksi

Pemantauan kualitas model membandingkan prediksi yang dibuat model Anda dengan label kebenaran dasar untuk mengukur kualitas model. Agar ini berfungsi, Anda secara berkala memberi label data yang diambil oleh pekerjaan endpoint atau batch transform Anda dan mengunggahnya ke Amazon S3.

Untuk mencocokkan label Ground Truth dengan data prediksi yang diambil, harus ada pengidentifikasi unik untuk setiap rekaman dalam kumpulan data. Struktur setiap catatan untuk data kebenaran dasar adalah sebagai berikut:

```
{
```

```
  "groundTruthData": {
    "data": "1",
    "encoding": "CSV" # only CSV supported at launch, we assume "data" only consists of
 label
  },
  "eventMetadata": {
    "eventId": "aaaa-bbbb-cccc"
  },
  "eventVersion": "0"
}
```

Dalam `groundTruthData` struktur, `eventId` dapat menjadi salah satu dari berikut ini:

- `eventId`— ID ini secara otomatis dihasilkan ketika pengguna memanggil titik akhir.
- `inferenceId`— Penelepon memasok ID ini ketika mereka memanggil titik akhir.

Jika `inferenceId` ada dalam rekaman data yang diambil, Model Monitor menggunakannya untuk menggabungkan data yang diambil dengan catatan Ground Truth. Anda bertanggung jawab untuk memastikan bahwa catatan `inferenceId` in the Ground Truth cocok dengan catatan yang ditangkap. `inferenceId` Jika tidak `inferenceId` ada dalam data yang diambil, monitor model menggunakan `eventId` dari catatan data yang diambil untuk mencocokkannya dengan catatan Ground Truth.

Anda harus mengunggah data Ground Truth ke bucket Amazon S3 yang memiliki format jalur yang sama dengan data yang diambil, yaitu dalam bentuk berikut:

```
s3://bucket/prefix/yyyy/mm/dd/hh
```

Tanggal di jalur ini adalah tanggal ketika label Ground Truth dikumpulkan, dan tidak harus cocok dengan tanggal ketika inferensi dihasilkan.

Setelah Anda membuat dan mengunggah label Ground Truth, sertakan lokasi label sebagai parameter saat Anda membuat pekerjaan pemantauan. Jika Anda menggunakanAWS SDK for Python (Boto3), lakukan ini dengan menentukan lokasi label Ground Truth sebagai `S3Uri` bidang `GroundTruthS3Input` parameter dalam panggilan ke `create_model_quality_job_definition` metode. Jika Anda menggunakan SageMaker Python SDK, tentukan lokasi label Ground Truth sebagai `ground_truth_input` parameter dalam panggilan ke `create_monitoring_schedule` objek. `ModelQualityMonitor`

Metrik Kualitas Model

Pekerjaan pemantauan kualitas model menghitung metrik yang berbeda tergantung pada jenis masalah ML. Bagian berikut mencantumkan metrik yang dianalisis untuk setiap jenis masalah ML.

> **Note**
>
> Standar deviasi untuk metrik disediakan hanya jika setidaknya 200 sampel tersedia. Model Monitor menghitung standar deviasi dengan mengambil sampel secara acak 80% data 5 kali, menghitung metrik, dan mengambil standar deviasi untuk hasil tersebut.

Metrik Regresi

Berikut ini menunjukkan contoh metrik yang dihitung oleh monitor kualitas model untuk masalah regresi.

```
"regression_metrics" : {
    "mae" : {
      "value" : 0.3711832061068702,
      "standard_deviation" : 0.0037566388129940394
    },
    "mse" : {
      "value" : 0.3711832061068702,
      "standard_deviation" : 0.0037566388129940524
    },
    "rmse" : {
      "value" : 0.609248066149471,
      "standard_deviation" : 0.003079253267651125
    },
    "r2" : {
      "value" : -1.3766111872212665,
      "standard_deviation" : 0.022653980022771227
    }
  }
```

Metrik Klasifikasi Biner

Berikut ini menunjukkan contoh metrik yang dihitung oleh monitor kualitas model untuk masalah klasifikasi biner.

```
"binary_classification_metrics" : {
    "confusion_matrix" : {
      "0" : {
        "0" : 1,
        "1" : 2
      },
      "1" : {
        "0" : 0,
        "1" : 1
      }
    },
    "recall" : {
      "value" : 1.0,
      "standard_deviation" : "NaN"
    },
    "precision" : {
      "value" : 0.3333333333333333,
      "standard_deviation" : "NaN"
    },
    "accuracy" : {
      "value" : 0.5,
      "standard_deviation" : "NaN"
    },
    "recall_best_constant_classifier" : {
      "value" : 1.0,
      "standard_deviation" : "NaN"
    },
    "precision_best_constant_classifier" : {
      "value" : 0.25,
      "standard_deviation" : "NaN"
    },
    "accuracy_best_constant_classifier" : {
      "value" : 0.25,
      "standard_deviation" : "NaN"
    },
    "true_positive_rate" : {
      "value" : 1.0,
      "standard_deviation" : "NaN"
    },
    "true_negative_rate" : {
      "value" : 0.3333333333333337,
      "standard_deviation" : "NaN"
    },
```

```
    "false_positive_rate" : {
      "value" : 0.6666666666666666,
      "standard_deviation" : "NaN"
    },
    "false_negative_rate" : {
      "value" : 0.0,
      "standard_deviation" : "NaN"
    },
    "receiver_operating_characteristic_curve" : {
      "false_positive_rates" : [ 0.0, 0.0, 0.0, 0.0, 0.0, 1.0 ],
      "true_positive_rates" : [ 0.0, 0.25, 0.5, 0.75, 1.0, 1.0 ]
    },
    "precision_recall_curve" : {
      "precisions" : [ 1.0, 1.0, 1.0, 1.0, 1.0 ],
      "recalls" : [ 0.0, 0.25, 0.5, 0.75, 1.0 ]
    },
    "auc" : {
      "value" : 1.0,
      "standard_deviation" : "NaN"
    },
    "f0_5" : {
      "value" : 0.3846153846153846,
      "standard_deviation" : "NaN"
    },
    "f1" : {
      "value" : 0.5,
      "standard_deviation" : "NaN"
    },
    "f2" : {
      "value" : 0.7142857142857143,
      "standard_deviation" : "NaN"
    },
    "f0_5_best_constant_classifier" : {
      "value" : 0.29411764705882354,
      "standard_deviation" : "NaN"
    },
    "f1_best_constant_classifier" : {
      "value" : 0.4,
      "standard_deviation" : "NaN"
    },
    "f2_best_constant_classifier" : {
      "value" : 0.625,
      "standard_deviation" : "NaN"
    }
```

```
  }
```

Metrik Multiclass

Berikut ini menunjukkan contoh metrik yang dihitung oleh monitor kualitas model untuk masalah klasifikasi multiclass.

```
"multiclass_classification_metrics" : {
    "confusion_matrix" : {
      "0" : {
        "0" : 1180,
        "1" : 510
      },
      "1" : {
        "0" : 268,
        "1" : 138
      }
    },
    "accuracy" : {
      "value" : 0.6288167938931297,
      "standard_deviation" : 0.00375663881299405
    },
    "weighted_recall" : {
      "value" : 0.6288167938931297,
      "standard_deviation" : 0.003756638812994008
    },
    "weighted_precision" : {
      "value" : 0.6983172269629505,
      "standard_deviation" : 0.006195912915307507
    },
    "weighted_f0_5" : {
      "value" : 0.6803947317178771,
      "standard_deviation" : 0.005328406973561699
    },
    "weighted_f1" : {
      "value" : 0.6571162346664904,
      "standard_deviation" : 0.004385008075019733
    },
    "weighted_f2" : {
      "value" : 0.6384024354394601,
      "standard_deviation" : 0.003867109755267757
    },
    "accuracy_best_constant_classifier" : {
```

```
      "value" : 0.19370229007633588,
      "standard_deviation" : 0.0032049848450732355
    },
    "weighted_recall_best_constant_classifier" : {
      "value" : 0.19370229007633588,
      "standard_deviation" : 0.0032049848450732355
    },
    "weighted_precision_best_constant_classifier" : {
      "value" : 0.03752057718081697,
      "standard_deviation" : 0.001241536088657851
    },
    "weighted_f0_5_best_constant_classifier" : {
      "value" : 0.04473443104152011,
      "standard_deviation" : 0.0014460485504284792
    },
    "weighted_f1_best_constant_classifier" : {
      "value" : 0.06286421244683643,
      "standard_deviation" : 0.0019113576884608862
    },
    "weighted_f2_best_constant_classifier" : {
      "value" : 0.10570313141262414,
      "standard_deviation" : 0.002734216826748117
    }
  }
```

CloudWatch Metrik Kualitas Model

Jika Anda menetapkan nilai `True` saat Anda membuat jadwal pemantauan, pekerjaan pemantauan kualitas model mengirimkan semua metrik ke Amazon CloudWatch. `enable_cloudwatch_metrics`

Metrik kualitas model muncul di namespace berikut:

- `For real-time endpoints: aws/sagemaker/Endpoints/model-metrics`
- `For batch transform jobs: aws/sagemaker/ModelMonitoring/model-metrics`

Untuk daftar metrik yang dipancarkan, lihat. Metrik Kualitas Model

Anda dapat menggunakan CloudWatch metrik untuk membuat alarm ketika metrik tertentu tidak memenuhi ambang batas yang Anda tentukan. Untuk petunjuk tentang cara membuat CloudWatch

alarm, lihat Membuat CloudWatch Alarm Berdasarkan Ambang Statis di Panduan CloudWatch Pengguna Amazon.

Monitor Bias Drift untuk Model dalam Produksi

Pemantauan bias Amazon SageMaker Clarify membantu ilmuwan data dan insinyur ML memantau prediksi bias secara teratur. Saat model dipantau, pelanggan dapat melihat laporan dan grafik yang dapat diekspor yang merinci bias di SageMaker Studio dan mengonfigurasi peringatan di Amazon CloudWatch untuk menerima pemberitahuan jika bias di luar ambang batas tertentu terdeteksi. Bias dapat diperkenalkan atau diperburuk dalam model ML yang diterapkan ketika data pelatihan berbeda dari data yang dilihat model selama penerapan (yaitu, data langsung). Jenis perubahan dalam distribusi data langsung ini mungkin bersifat sementara (misalnya, karena beberapa peristiwa dunia nyata yang berumur pendek) atau permanen. Dalam kedua kasus tersebut, mungkin penting untuk mendeteksi perubahan ini. Misalnya, output model untuk memprediksi harga rumah dapat menjadi bias jika tingkat hipotek yang digunakan untuk melatih model berbeda dari tingkat hipotek dunia nyata saat ini. Dengan kemampuan deteksi bias di Model Monitor, ketika SageMaker mendeteksi bias di luar ambang batas tertentu, secara otomatis menghasilkan metrik yang dapat Anda lihat di SageMaker Studio dan melalui peringatan Amazon CloudWatch.

Secara umum, mengukur bias hanya selama train-and-deploy fase mungkin tidak cukup. Ada kemungkinan bahwa setelah model digunakan, distribusi data yang dilihat oleh model yang diterapkan (yaitu, data langsung) berbeda dari distribusi data dalam kumpulan data pelatihan. Perubahan ini mungkin menimbulkan bias dalam model dari waktu ke waktu. Perubahan dalam distribusi data langsung mungkin bersifat sementara (misalnya, karena beberapa perilaku berumur pendek seperti musim liburan) atau permanen. Dalam kedua kasus, mungkin penting untuk mendeteksi perubahan ini dan mengambil langkah-langkah untuk mengurangi bias bila perlu.

Untuk mendeteksi perubahan ini, SageMaker Clarify menyediakan fungsionalitas untuk memantau metrik bias model yang diterapkan secara terus menerus dan meningkatkan peringatan otomatis jika metrik melebihi ambang batas. Misalnya, pertimbangkan metrik bias DPPL. Tentukan rentang nilai yang diizinkan A= (a_{min}, a_{max}), misalnya interval (-0.1, 0.1), yang harus dimiliki DPPL selama penerapan. Setiap penyimpangan dari kisaran ini harus meningkatkan peringatan bias yang terdeteksi. Dengan SageMaker Clarify, Anda dapat melakukan pemeriksaan ini secara berkala.

Misalnya, Anda dapat mengatur frekuensi cek menjadi 2 hari. Ini berarti SageMaker Clarify menghitung metrik DPPL pada data yang dikumpulkan selama jendela 2 hari. Dalam contoh ini, D_{win} adalah data yang diproses model selama jendela 2 hari terakhir. Peringatan dikeluarkan jika nilai DPPL b yang $_{win}$ dihitung pada D_{win} berada di luar rentang yang diizinkan A. Pendekatan ini

untuk memeriksa apakah b_{win} berada di luar A bisa agak bising. D_{win} mungkin terdiri dari sangat sedikit sampel dan mungkin tidak mewakili distribusi data langsung. Ukuran sampel yang kecil berarti bahwa nilai bias b yang $_{win}$ dihitung di atas D_{win} mungkin bukan perkiraan yang sangat kuat. Faktanya, nilai b yang sangat tinggi (atau rendah) $_{win}$ dapat diamati murni karena kebetulan. Untuk memastikan bahwa kesimpulan yang diambil dari data D_{win} yang diamati signifikan secara statistik, SageMaker Clarify menggunakan interval kepercayaan. Secara khusus, ia menggunakan metode Interval Bootstrap Normal untuk membangun interval C= (c_{min}, c_{max}) sedemikian rupa sehingga SageMaker Clarify yakin bahwa nilai bias sebenarnya yang dihitung melalui data langsung lengkap terkandung dalam C dengan probabilitas tinggi. Sekarang, jika interval kepercayaan C tumpang tindih dengan rentang A yang diizinkan, SageMaker Clarify menafsirkannya sebagai “kemungkinan nilai metrik bias dari distribusi data langsung berada dalam kisaran yang diizinkan”. Jika C dan A terputus-putus, SageMaker Clarify yakin bahwa metrik bias tidak terletak pada A dan menimbulkan peringatan.

Model Monitor Contoh Notebook

Amazon SageMaker Clarify menyediakan contoh buku catatan berikut yang menunjukkan cara menangkap data inferensi untuk titik akhir real-time, membuat garis dasar untuk memantau bias yang berkembang, dan memeriksa hasilnya:

- Memantau penyimpangan bias dan penyimpangan atribusi fitur Amazon SageMaker Clarify — Gunakan Monitor Model SageMaker Amazon untuk memantau penyimpangan bias dan fitur penyimpangan atribusi dari waktu ke waktu.

Notebook ini telah diverifikasi untuk berjalan di Amazon SageMaker Studio saja. Jika Anda memerlukan petunjuk tentang cara membuka notebook di Amazon SageMaker Studio, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic. Jika Anda diminta untuk memilih kernel, pilih Python 3 (Ilmu Data). Topik berikut berisi sorotan dari dua langkah terakhir, dan berisi contoh kode dari contoh buku catatan.

Topik

- Buat Bias Drift Baseline
- Pelanggaran Bias Drift
- Konfigurasikan Parameter untuk Memantau Drift Bias
- Jadwal Pekerjaan Bias Drift Monitoring
- Periksa Laporan untuk Data Bias Drift
- CloudWatch Metrik untuk Analisis Bias Drift

Buat Bias Drift Baseline

Setelah Anda mengonfigurasi aplikasi untuk menangkap data inferensi transformasi real-time atau batch, tugas pertama untuk memantau penyimpangan bias adalah membuat baseline. Ini melibatkan konfigurasi input data, grup mana yang sensitif, bagaimana prediksi ditangkap, dan model serta metrik bias pasca-pelatihannya. Maka Anda perlu memulai pekerjaan baselining.

Monitor bias model dapat mendeteksi penyimpangan bias model ML secara teratur. Mirip dengan jenis pemantauan lainnya, prosedur standar untuk membuat monitor bias model adalah baselining pertama dan kemudian menetapkan jadwal pemantauan.

```
model_bias_monitor = ModelBiasMonitor(
    role=role,
    sagemaker_session=sagemaker_session,
    max_runtime_in_seconds=1800,
)
```

`DataConfig`menyimpan informasi tentang dataset yang akan dianalisis (misalnya, file dataset), formatnya (yaitu, CSV atau JSON Lines), header (jika ada) dan label.

```
model_bias_baselining_job_result_uri = f"{baseline_results_uri}/model_bias"
model_bias_data_config = DataConfig(
    s3_data_input_path=validation_dataset,
    s3_output_path=model_bias_baselining_job_result_uri,
    label=label_header,
    headers=all_headers,
    dataset_type=dataset_type,
)
```

`BiasConfig`adalah konfigurasi grup sensitif dalam kumpulan data. Biasanya, bias diukur dengan menghitung metrik dan membandingkannya di seluruh kelompok. Kelompok yang diminati disebut facet. Untuk bias pasca-pelatihan, Anda juga harus mempertimbangkan label positif.

```
model_bias_config = BiasConfig(
    label_values_or_threshold=[1],
    facet_name="Account Length",
    facet_values_or_threshold=[100],
)
```

`ModelPredictedLabelConfig`menentukan cara mengekstrak label yang diprediksi dari output model. Dalam contoh ini, cutoff 0.8 telah dipilih untuk mengantisipasi bahwa pelanggan akan sering berbalik. Untuk output yang lebih rumit, ada beberapa opsi lagi, seperti “label” adalah indeks, nama, atau JMESPath untuk menemukan label yang diprediksi di payload respons titik akhir.

```
model_predicted_label_config = ModelPredictedLabelConfig(
    probability_threshold=0.8,
)
```

`ModelConfig`adalah konfigurasi yang terkait dengan model yang akan digunakan untuk inferensi. Untuk menghitung metrik bias pasca-pelatihan, perhitungan perlu mendapatkan kesimpulan untuk nama model yang diberikan. Untuk mencapai ini, pekerjaan pemrosesan menggunakan model untuk membuat titik akhir singkat (juga dikenal sebagai titik akhir bayangan). Pekerjaan pemrosesan menghapus titik akhir bayangan setelah perhitungan selesai. Konfigurasi ini juga digunakan oleh monitor penjelasan.

```
model_config = ModelConfig(
    model_name=model_name,
    instance_count=endpoint_instance_count,
    instance_type=endpoint_instance_type,
    content_type=dataset_type,
    accept_type=dataset_type,
)
```

Sekarang Anda dapat memulai tugas baselining.

```
model_bias_monitor.suggest_baseline(
    model_config=model_config,
    data_config=model_bias_data_config,
    bias_config=model_bias_config,
    model_predicted_label_config=model_predicted_label_config,
)
print(f"ModelBiasMonitor baselining job:
 {model_bias_monitor.latest_baselining_job_name}")
```

Monitor terjadwal secara otomatis mengambil nama pekerjaan dasar dan menunggunya sebelum pemantauan dimulai.

Pelanggaran Bias Drift

Pekerjaan penyimpangan bias mengevaluasi kendala dasar yang disediakan oleh konfigurasi dasar terhadap hasil analisis arus. `MonitoringExecution` Jika pelanggaran terdeteksi, pekerjaan mencantumkannya ke file constraint_violations.json di lokasi keluaran eksekusi, dan menandai status eksekusi sebagai. Menafsirkan hasil

Berikut adalah skema file pelanggaran penyimpangan bias.

- `facet`— Nama aspek, disediakan oleh aspek `name_or_index` konfigurasi analisis pekerjaan pemantauan.
- `facet_value`— Nilai aspek, yang disediakan oleh aspek `value_or_threshold` konfigurasi analisis pekerjaan pemantauan.
- `metric_name`— Nama singkat dari metrik bias. Misalnya, “CI” untuk ketidakseimbangan kelas. Lihat Ukur Bias Pra-pelatihan nama pendek dari masing-masing metrik bias pra-pelatihan dan Ukur Data Pasca-pelatihan dan Bias Model untuk nama pendek dari masing-masing metrik bias pasca-pelatihan.
- `constraint_check_type`— Jenis pelanggaran yang dipantau. Saat ini hanya mendukung `bias_drift_check`.
- `description`— Pesan deskriptif untuk menjelaskan pelanggaran.

```
{
    "version": "1.0",
    "violations": [{
        "facet": "string",
        "facet_value": "string",
        "metric_name": "string",
        "constraint_check_type": "string",
        "description": "string"
    }]
}
```

Metrik bias digunakan untuk mengukur tingkat kesetaraan dalam suatu distribusi. Nilai mendekati nol menunjukkan bahwa distribusi lebih seimbang. Jika nilai metrik bias dalam file hasil analisis pekerjaan (analysis.json) lebih buruk daripada nilai yang sesuai dalam file batasan dasar, pelanggaran dicatat. Sebagai contoh, jika batasan dasar untuk metrik bias DPPL adalah`0.2`, dan hasil analisisnya`0.1`, tidak ada pelanggaran yang dicatat karena lebih dekat dengan daripada. `0.1` `0`

`0.2` Namun, jika hasil analisisnya`-0.3`, pelanggaran dicatat karena `0` lebih jauh dari batasan dasar. `0.2`

```
{
    "version": "1.0",
    "violations": [{
        "facet": "Age",
        "facet_value": "40",
        "metric_name": "CI",
        "constraint_check_type": "bias_drift_check",
        "description": "Value 0.0751544567666083 does not meet the constraint
 requirement"
    }, {
        "facet": "Age",
        "facet_value": "40",
        "metric_name": "DPPL",
        "constraint_check_type": "bias_drift_check",
        "description": "Value -0.0791244970125596 does not meet the constraint
 requirement"
    }]
}
```

Konfigurasikan Parameter untuk Memantau Drift Bias

Amazon SageMaker Clarify bias monitoring menggunakan kembali subset parameter yang digunakan dalam konfigurasi analisis. Konfigurasikan Analisis Setelah menjelaskan parameter konfigurasi, topik ini memberikan contoh file JSON. File-file ini digunakan untuk mengonfigurasi kumpulan data CSV dan JSON Lines untuk memantau penyimpangan bias saat model pembelajaran mesin sedang diproduksi.

Parameter berikut harus disediakan dalam file JSON. Jalur ke file JSON ini harus disediakan dalam `ConfigUri` parameter `ModelBiasAppSpecification`API.

- **`"version"`**— (Opsional) Versi skema dari file konfigurasi. Jika tidak disediakan, versi terbaru yang didukung digunakan.
- **`"headers"`**— (Opsional) Daftar nama kolom dalam dataset. Jika `dataset_type` `"label"` adalah `"application/jsonlines"` dan ditentukan, maka header terakhir menjadi header kolom label.

- **"label"**— (Opsional) Atribut target untuk model yang akan digunakan untuk metrik bias. Ditentukan baik sebagai nama kolom, atau indeks (jika format dataset CSV), atau sebagai JMESPath (jika format dataset adalah JSON Lines).
- **"label_values_or_threshold"**— (Opsional) Daftar nilai label atau ambang batas. Menunjukkan hasil positif yang digunakan untuk metrik bias.
- **"facet"**— (Opsional) Daftar fitur yang merupakan atribut sensitif, disebut sebagai aspek. Segi digunakan untuk metrik bias dalam bentuk pasangan, dan meliputi yang berikut:
  - "name_or_index"— Nama atau indeks kolom facet.
  - "value_or_threshold"— (Opsional) Daftar nilai atau ambang batas yang dapat diambil oleh kolom facet. Menunjukkan kelompok sensitif, seperti kelompok yang digunakan untuk mengukur bias terhadap. Jika tidak disediakan, metrik bias dihitung sebagai satu grup untuk setiap nilai unik (bukan semua nilai). Jika kolom faset adalah numerik, nilai ambang ini diterapkan sebagai batas bawah untuk memilih grup sensitif.
- **"group_variable"**— (Opsional) Nama kolom atau indeks untuk menunjukkan variabel grup yang akan digunakan untuk metrik bias Disparitas Demografis Bersyarat.

Parameter lain harus disediakan di `EndpointInput` (untuk titik akhir waktu nyata) atau `BatchTransformInput` (untuk pekerjaan transformasi batch) dari `ModelBiasJobInput`API.

- `FeaturesAttribute`- Parameter ini diperlukan jika format data input titik akhir. `"application/jsonlines"` Ini adalah JMESPath yang digunakan untuk menemukan kolom fitur jika format dataset adalah JSON Lines.
- `InferenceAttribute`— Lokasi indeks atau JMESPath dalam keluaran model untuk atribut target yang akan digunakan untuk dipantau bias menggunakan metrik bias. Jika tidak disediakan dalam `accept_type` kasus CSV, maka diasumsikan bahwa output model adalah nilai numerik tunggal yang sesuai dengan skor atau probabilitas.
- `ProbabilityAttribute`— Indeks atau lokasi JMESPath dalam output model untuk probabilitas. Jika keluaran model adalah JSON Lines dengan daftar label dan probabilitas, misalnya, maka label yang sesuai dengan probabilitas maksimum dipilih untuk perhitungan bias.
- `ProbabilityThresholdAttribute`— (Opsional) Nilai float untuk menunjukkan ambang batas untuk memilih label biner, dalam kasus klasifikasi biner. Nilai default adalah 0,5.

Contoh File Konfigurasi JSON untuk Kumpulan Data Garis CSV dan JSON

Berikut adalah contoh file JSON yang digunakan untuk mengonfigurasi kumpulan data CSV dan JSON Lines untuk memantaunya untuk penyimpangan bias.

Topik

Set data CSV

Pertimbangkan kumpulan data yang memiliki empat kolom fitur dan satu kolom label, di mana fitur pertama dan labelnya adalah biner, seperti pada contoh berikut.

```
0, 0.5814568701544718, 0.6651538910132964, 0.3138080342665499, 0
1, 0.6711642728531724, 0.7466687034026017, 0.1215477472819713, 1
0, 0.0453256543003371, 0.6377430803264152, 0.3558625219713576, 1
1, 0.4785191813363956, 0.0265841045263860, 0.0376935084990697, 1
```

Asumsikan bahwa keluaran model memiliki dua kolom, di mana yang pertama adalah label yang diprediksi dan yang kedua adalah probabilitas, seperti pada contoh berikut.

```
1, 0.5385257417814224
```

Kemudian file konfigurasi JSON berikut menunjukkan contoh bagaimana kumpulan data CSV ini dapat dikonfigurasi.

```
{
    "headers": [
        "feature_0",
        "feature_1",
        "feature_2",
        "feature_3",
        "target"
    ],
    "label": "target",
    "label_values_or_threshold": [1],
    "facet": [{
        "name_or_index": "feature_1",
```

```
        "value_or_threshold": [1]
    }]
}
```

Label yang diprediksi dipilih oleh `"InferenceAttribute"` parameter. Penomoran berbasis nol digunakan, jadi 0 menunjukkan kolom pertama dari output model,

```
"EndpointInput": {
    ...
    "InferenceAttribute": 0
    ...
}
```

Atau, Anda dapat menggunakan parameter yang berbeda untuk mengubah nilai probabilitas menjadi label prediksi biner. Penomoran berbasis nol digunakan: 1 menunjukkan kolom kedua; `ProbabilityThresholdAttribute` nilai 0,6 menunjukkan bahwa probabilitas lebih besar dari 0,6 memprediksi label biner sebagai 1.

```
"EndpointInput": {
    ...
    "ProbabilityAttribute": 1,
    "ProbabilityThresholdAttribute": 0.6
    ...
}
```

Set Data JSON Lines

Pertimbangkan kumpulan data yang memiliki empat kolom fitur dan satu kolom label, di mana fitur pertama dan labelnya adalah biner, seperti pada contoh berikut.

```
{"features":[0, 0.5814568701544718, 0.6651538910132964, 0.3138080342665499], "label":0}
{"features":[1, 0.6711642728531724, 0.7466687034026017, 0.1215477472819713], "label":1}
{"features":[0, 0.0453256543003371, 0.6377430803264152, 0.3558625219713576], "label":1}
{"features":[1, 0.4785191813363956, 0.0265841045263860, 0.0376935084990697], "label":1}
```

Asumsikan bahwa output model memiliki dua kolom, di mana yang pertama adalah label yang diprediksi dan yang kedua adalah probabilitas.

```
{"predicted_label":1, "probability":0.5385257417814224}
```

File konfigurasi JSON berikut menunjukkan contoh bagaimana dataset JSON Lines ini dapat dikonfigurasi.

```
{
    "headers": [
        "feature_0",
        "feature_1",
        "feature_2",
        "feature_3",
        "target"
    ],
    "label": "label",
    "label_values_or_threshold": [1],
    "facet": [{
        "name_or_index": "feature_1",
        "value_or_threshold": [1]
    }]
}
```

Kemudian, nilai `"features"` parameter di `EndpointInput` (untuk titik akhir waktu nyata) atau `BatchTransformInput` (untuk pekerjaan transformasi batch) digunakan untuk menemukan fitur dalam kumpulan data, dan nilai `"predicted_label"` parameter memilih label yang diprediksi dari keluaran model.

```
"EndpointInput": {
    ...
    "FeaturesAttribute": "features",
    "InferenceAttribute": "predicted_label"
    ...
}
```

Atau, Anda dapat mengonversi nilai probabilitas menjadi label biner yang diprediksi menggunakan nilai `ProbabilityThresholdAttribute` parameter. Nilai 0,6, misalnya, menunjukkan bahwa probabilitas lebih besar dari 0,6 memprediksi label biner sebagai 1.

```
"EndpointInput": {
    ...
    "FeaturesAttribute": "features",
    "ProbabilityAttribute": "probability",
    "ProbabilityThresholdAttribute": 0.6
    ...
```

```
}
```

Jadwal Pekerjaan Bias Drift Monitoring

Setelah Anda membuat baseline Anda, Anda dapat memanggil `create_monitoring_schedule()` metode instance `ModelBiasModelMonitor` kelas Anda untuk menjadwalkan monitor drift bias per jam. Bagian berikut menunjukkan kepada Anda cara membuat monitor penyimpangan bias untuk model yang diterapkan ke titik akhir waktu nyata serta untuk pekerjaan transformasi batch.

> **Important**
>
> Anda dapat menentukan input transformasi batch atau input titik akhir, tetapi tidak keduanya, saat Anda membuat jadwal pemantauan.

Tidak seperti pemantauan kualitas data, Anda perlu menyediakan label Ground Truth jika Anda ingin memantau kualitas model. Namun, label Ground Truth bisa ditunda. Untuk mengatasinya, tentukan offset saat Anda membuat jadwal pemantauan. Untuk detail tentang cara membuat offset waktu, lihatOffset monitor model.

Jika Anda telah mengirimkan pekerjaan baselining, monitor secara otomatis mengambil konfigurasi analisis dari pekerjaan baselining. Jika Anda melewati langkah dasar atau kumpulan data pengambilan memiliki sifat yang berbeda dari kumpulan data pelatihan, Anda harus memberikan konfigurasi analisis.

Pemantauan penyimpangan bias untuk model yang digunakan ke titik akhir waktu nyata

Untuk menjadwalkan monitor penyimpangan bias untuk titik akhir waktu nyata, teruskan `EndpointInput` instance Anda ke `endpoint_input` argumen `ModelBiasModelMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

model_bias_monitor = ModelBiasModelMonitor(
    role=sagemaker.get_execution_role(),
    ...
)

model_bias_analysis_config = None
```

```
if not model_bias_monitor.latest_baselining_job:
    model_bias_analysis_config = BiasAnalysisConfig(
        model_bias_config,
        headers=all_headers,
        label=label_header,
    )

model_bias_monitor.create_monitoring_schedule(
    monitor_schedule_name=schedule_name,
    post_analytics_processor_script=s3_code_postprocessor_uri,
    output_s3_uri=s3_report_path,
    statistics=model_bias_monitor.baseline_statistics(),
    constraints=model_bias_monitor.suggested_constraints(),
    schedule_cron_expression=CronExpressionGenerator.hourly(),
    enable_cloudwatch_metrics=True,
    analysis_config=model_bias_analysis_config,
    endpoint_input=EndpointInput(
        endpoint_name=endpoint_name,
        destination="/opt/ml/processing/input/endpoint",
        start_time_offset="-PT1H",
        end_time_offset="-PT0H",
        probability_threshold_attribute=0.8,
    ),
)
```

Pemantauan penyimpangan bias untuk pekerjaan transformasi batch

Untuk menjadwalkan monitor penyimpangan bias untuk pekerjaan transformasi batch, teruskan `BatchTransformInput` instance Anda ke `batch_transform_input` argumen `ModelBiasModelMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

model_bias_monitor = ModelBiasModelMonitor(
    role=sagemaker.get_execution_role(),
    ...
)

model_bias_analysis_config = None
if not model_bias_monitor.latest_baselining_job:
    model_bias_analysis_config = BiasAnalysisConfig(
        model_bias_config,
        headers=all_headers,
```

```
        label=label_header,
    )

schedule = model_bias_monitor.create_monitoring_schedule(
   monitor_schedule_name=schedule_name,
   post_analytics_processor_script=s3_code_postprocessor_uri,
   output_s3_uri=s3_report_path,
   statistics=model_bias_monitor.baseline_statistics(),
   constraints=model_bias_monitor.suggested_constraints(),
   schedule_cron_expression=CronExpressionGenerator.hourly(),
   enable_cloudwatch_metrics=True,
   analysis_config=model_bias_analysis_config,
   batch_transform_input=BatchTransformInput(
        destination="opt/ml/processing/input",
        data_captured_destination_s3_uri=s3_capture_path,
        start_time_offset="-PT1H",
        end_time_offset="-PT0H",
        probability_threshold_attribute=0.8
   ),
)
```

Periksa Laporan untuk Data Bias Drift

Jika Anda tidak dapat memeriksa hasil pemantauan dalam laporan yang dihasilkan di SageMaker Studio, Anda dapat mencetaknya sebagai berikut:

```
schedule_desc = model_bias_monitor.describe_schedule()
execution_summary = schedule_desc.get("LastMonitoringExecutionSummary")
if execution_summary and execution_summary["MonitoringExecutionStatus"] in
 ["Completed", "CompletedWithViolations"]:
    last_model_bias_monitor_execution = model_bias_monitor.list_executions()[-1]
    last_model_bias_monitor_execution_report_uri =
 last_model_bias_monitor_execution.output.destination
    print(f'Report URI: {last_model_bias_monitor_execution_report_uri}')
    last_model_bias_monitor_execution_report_files =
 sorted(S3Downloader.list(last_model_bias_monitor_execution_report_uri))
    print("Found Report Files:")
    print("\n ".join(last_model_bias_monitor_execution_report_files))
else:
    last_model_bias_monitor_execution = None
    print("====STOP==== \n No completed executions to inspect further. Please wait till
 an execution completes or investigate previously reported failures.")
```

Jika ada pelanggaran dibandingkan dengan baseline, mereka tercantum di sini:

```
if last_model_bias_monitor_execution:
    model_bias_violations = last_model_bias_monitor_execution.constraint_violations()
    if model_bias_violations:
        print(model_bias_violations.body_dict)
```

Jika model Anda diterapkan ke titik akhir real-time, Anda dapat melihat visualisasi di SageMaker Studio hasil analisis dan CloudWatch metrik dengan memilih tab Endpoints, lalu mengklik dua kali titik akhir.

CloudWatch Metrik untuk Analisis Bias Drift

Panduan ini menunjukkan CloudWatch metrik dan propertinya yang dapat Anda gunakan untuk analisis penyimpangan bias di SageMaker Clarify. Pekerjaan pemantauan penyimpangan bias menghitung metrik bias pra-pelatihan dan metrik bias pasca-pelatihan, dan mempublikasikannya ke namespace berikut: CloudWatch

- Untuk titik akhir waktu nyata: `aws/sagemaker/Endpoints/bias-metrics`
- Untuk pekerjaan transformasi batch: `aws/sagemaker/ModelMonitoring/bias-metrics`

Nama CloudWatch metrik menambahkan nama pendek metrik ke`bias_metric`.

Misalnya, `bias_metric_CI` adalah metrik bias untuk ketidakseimbangan kelas (CI).

> **Note**
>
> `+/- infinity`diterbitkan sebagai nomor floating point`+/- 2.348543e108`, dan kesalahan termasuk nilai nol tidak dipublikasikan.

Setiap metrik memiliki properti berikut:

- `Endpoint`: Nama titik akhir yang dipantau, jika berlaku.
- `MonitoringSchedule`Nama jadwal untuk tugas pemantauan.
- `BiasStage`: Nama tahap pekerjaan pemantauan penyimpangan bias. Pilih salah satu `Pre-training` atau`Post-Training`.

- `Label`: Nama fitur target, disediakan oleh konfigurasi analisis pekerjaan pemantauan`label`.
- `LabelValue`: Nilai fitur target, disediakan oleh konfigurasi analisis pekerjaan pemantauan`label_values_or_threshold`.
- `Facet`: Nama aspek, disediakan oleh aspek `name_of_index` konfigurasi analisis pekerjaan pemantauan.
- `FacetValue`: Nilai aspek, disediakan oleh aspek `nvalue_or_threshold` konfigurasi analisis pekerjaan pemantauan.

Untuk menghentikan pekerjaan pemantauan dari penerbitan metrik, atur `publish_cloudwatch_metrics` ke `Disabled` dalam `Environment` peta definisi pekerjaan bias model.

Monitor Fitur Atribusi Drift untuk Model dalam Produksi

Penyimpangan dalam distribusi data langsung untuk model dalam produksi dapat menghasilkan penyimpangan yang sesuai dalam nilai atribusi fitur, seperti halnya dapat menyebabkan penyimpangan bias saat memantau metrik bias. Pemantauan atribusi fitur Amazon SageMaker Clarify membantu ilmuwan data dan teknisi ML memantau prediksi penyimpangan atribusi fitur secara teratur. Saat model dipantau, pelanggan dapat melihat laporan dan grafik yang dapat diekspor yang merinci atribusi fitur di SageMaker Studio dan mengonfigurasi peringatan di Amazon CloudWatch untuk menerima pemberitahuan jika terdeteksi bahwa nilai atribusi melayang melampaui ambang batas tertentu.

Untuk mengilustrasikan hal ini dengan situasi tertentu, pertimbangkan skenario hipotetis untuk penerimaan perguruan tinggi. Asumsikan bahwa kami mengamati nilai atribusi fitur (agregat) berikut dalam data pelatihan dan dalam data langsung:

Skenario Hipotesis Penerimaan Perguruan Tinggi

Fitur	Atribusi dalam data pelatihan	Atribusi dalam data langsung
Skor SAT	0,70	0,10
IPK	0,50	0,20
Peringkat kelas	0,05	0,70

Perubahan dari data pelatihan ke data langsung tampak signifikan. Peringkat fitur telah sepenuhnya terbalik. Mirip dengan penyimpangan bias, penyimpangan atribusi fitur mungkin disebabkan oleh perubahan dalam distribusi data langsung dan memerlukan pandangan lebih dekat ke perilaku model pada data langsung. Sekali lagi, langkah pertama dalam skenario ini adalah menaikkan alarm bahwa penyimpangan telah terjadi.

Kami dapat mendeteksi penyimpangan dengan membandingkan bagaimana peringkat fitur individu berubah dari data pelatihan menjadi data langsung. Selain peka terhadap perubahan urutan peringkat, kami juga ingin peka terhadap skor atribusi mentah fitur. Misalnya, mengingat dua fitur yang termasuk dalam peringkat dengan jumlah posisi yang sama mulai dari pelatihan ke data langsung, kami ingin lebih sensitif terhadap fitur yang memiliki skor atribusi lebih tinggi dalam data pelatihan. Dengan mempertimbangkan properti ini, kami menggunakan skor Normalized Discounted Cumulative Gain (NDCG) untuk membandingkan peringkat atribusi fitur pelatihan dan data langsung.

Secara khusus, asumsikan bahwa kami memiliki yang berikut:

- $F= [f_1,..., f_m]$ adalah daftar fitur yang diurutkan sehubungan dengan skor atribusi mereka dalam data pelatihan di mana m adalah jumlah total fitur. Misalnya, dalam kasus kami, F = [Skor SAT, IPK, Peringkat Kelas].
- a (f) adalah fungsi yang mengembalikan skor atribusi fitur pada data pelatihan yang diberikan fitur f. Misalnya, a (Skor SAT) = 0,70.
- $F′= [f′_1,..., f′_m]$ adalah daftar fitur yang diurutkan sehubungan dengan skor atribusi mereka dalam data langsung. Misalnya, F ′= [Peringkat Kelas, IPK, Skor SAT].

Kemudian, kita dapat menghitung NDCG sebagai:

$$ndcg=dcg/idcg$$

dengan

- $DCG = \Sigma_1^m a(f'_i) / \log(i+1_2)$
- $idCG = \Sigma_1^m a(f_i) / \log_2(i+1)$

DCG kuantitas mengukur apakah fitur dengan atribusi tinggi dalam data pelatihan juga diberi peringkat lebih tinggi dalam atribusi fitur yang dihitung pada data langsung. Kuantitas idCG mengukur skor ideal dan itu hanya faktor normalisasi untuk memastikan bahwa kuantitas akhir berada dalam kisaran [0, 1], dengan 1 menjadi nilai terbaik. Nilai NDCG 1 berarti bahwa peringkat atribusi fitur

dalam data langsung sama dengan yang ada di data pelatihan. Dalam contoh khusus ini, karena peringkat berubah sedikit, nilai NDCG adalah 0,69.

Di SageMaker Clarify, jika nilai NDCG di bawah 0,90, kami secara otomatis menaikkan peringatan.

Model Monitor Contoh Notebook

SageMaker Clarify memberikan contoh buku catatan berikut yang menunjukkan cara menangkap data inferensi untuk titik akhir waktu nyata, membuat garis dasar untuk memantau bias yang berkembang, dan memeriksa hasilnya:

- Memantau penyimpangan bias dan penyimpangan atribusi fitur Amazon SageMaker Clarify — Gunakan Monitor Model SageMaker Amazon untuk memantau penyimpangan bias dan fitur penyimpangan atribusi dari waktu ke waktu.

Notebook ini telah diverifikasi untuk berjalan di SageMaker Studio saja. Jika Anda memerlukan petunjuk tentang cara membuka notebook di SageMaker Studio, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic. Jika Anda diminta untuk memilih kernel, pilih Python 3 (Ilmu Data). Topik berikut berisi sorotan dari dua langkah terakhir, dan berisi contoh kode dari contoh buku catatan.

Topik

- Buat Baseline SHAP untuk Model dalam Produksi
- Pelanggaran Drift Atribusi Fitur Model
- Mengonfigurasi Parameter untuk Memonfigurasi Atribusi Drift
- Jadwal Fitur Atribut Pekerjaan Pemantauan Drift
- Periksa Laporan untuk Penyimpangan Atribut Fitur dalam Model Produksi
- CloudWatch Metrik untuk Analisis Drift Fitur

Buat Baseline SHAP untuk Model dalam Produksi

Penjelasan biasanya kontras, yaitu, mereka menjelaskan penyimpangan dari garis dasar. Untuk informasi tentang garis dasar penjelasan, lihat. Garis Dasar SHAP untuk Penjelasan

Selain memberikan penjelasan untuk inferensi per instance, SageMaker Clarify juga mendukung penjelasan global untuk model ML yang membantu Anda memahami perilaku model secara keseluruhan dalam hal fitur-fiturnya. SageMaker Clarify menghasilkan penjelasan global dari

model ML dengan menggabungkan nilai-nilai Shapley pada beberapa contoh. SageMaker Clarify mendukung berbagai cara agregasi berikut, yang dapat Anda gunakan untuk menentukan garis dasar:

- `mean_abs`— Rata-rata nilai SHAP absolut untuk semua contoh.
- `median`— Median nilai SHAP untuk semua instance.
- `mean_sq`— Rata-rata nilai SHAP kuadrat untuk semua instance.

Setelah Anda mengonfigurasi aplikasi untuk menangkap data inferensi transformasi real-time atau batch, tugas pertama untuk memantau drift dalam atribusi fitur adalah membuat baseline untuk dibandingkan. Ini melibatkan konfigurasi input data, grup mana yang sensitif, bagaimana prediksi ditangkap, dan model serta metrik bias pasca-pelatihannya. Maka Anda perlu memulai pekerjaan baselining. Monitor penjelasan model dapat menjelaskan prediksi model yang diterapkan yang menghasilkan kesimpulan dan mendeteksi penyimpangan atribusi fitur secara teratur.

```
model_explainability_monitor = ModelExplainabilityMonitor(
    role=role,
    sagemaker_session=sagemaker_session,
    max_runtime_in_seconds=1800,
)
```

Dalam contoh ini, pekerjaan baselining yang dapat dijelaskan membagikan kumpulan data pengujian dengan pekerjaan dasar bias, sehingga menggunakan pekerjaan yang sama`DataConfig`, dan satu-satunya perbedaan adalah URI keluaran pekerjaan.

```
model_explainability_baselining_job_result_uri = f"{baseline_results_uri}/
model_explainability"
model_explainability_data_config = DataConfig(
    s3_data_input_path=validation_dataset,
    s3_output_path=model_explainability_baselining_job_result_uri,
    label=label_header,
    headers=all_headers,
    dataset_type=dataset_type,
)
```

Saat ini penjelasan SageMaker Clarify menawarkan implementasi SHAP yang skalabel dan efisien, sehingga konfigurasi yang dapat dijelaskan adalah ShapConfig, termasuk yang berikut:

- `baseline`— Daftar baris (setidaknya satu) atau URI objek S3 yang akan digunakan sebagai dataset dasar dalam algoritma Kernel SHAP. Formatnya harus sama dengan format dataset. Setiap baris harus berisi hanya kolom/nilai fitur dan menghilangkan kolom/nilai label.
- `num_samples`— Jumlah sampel yang akan digunakan dalam algoritma Kernel SHAP. Angka ini menentukan ukuran kumpulan data sintetis yang dihasilkan untuk menghitung nilai SHAP.
- agg_method — Metode agregasi untuk nilai SHAP global. Berikut adalah nilai yang valid:
  - `mean_abs`— Rata-rata nilai SHAP absolut untuk semua contoh.
  - `median`— Median nilai SHAP untuk semua instance.
  - `mean_sq`— Rata-rata nilai SHAP kuadrat untuk semua instance.
- `use_logit`— Indikator apakah fungsi logit akan diterapkan pada prediksi model. Default-nya adalah `False`. Jika `use_logit` ya`True`, nilai SHAP akan memiliki unit log-odds.
- `save_local_shap_values`(bool) — Indikator apakah akan menyimpan nilai SHAP lokal di lokasi output. Default-nya adalah `False`.

```
# Here use the mean value of test dataset as SHAP baseline
test_dataframe = pd.read_csv(test_dataset, header=None)
shap_baseline = [list(test_dataframe.mean())]

shap_config = SHAPConfig(
    baseline=shap_baseline,
    num_samples=100,
    agg_method="mean_abs",
    save_local_shap_values=False,
)
```

Mulai pekerjaan dasar. `model_config`Hal yang sama diperlukan karena pekerjaan dasar penjelasan perlu membuat titik akhir bayangan untuk mendapatkan prediksi untuk kumpulan data sintetis yang dihasilkan.

```
model_explainability_monitor.suggest_baseline(
    data_config=model_explainability_data_config,
    model_config=model_config,
    explainability_config=shap_config,
)
print(f"ModelExplainabilityMonitor baselining job:
 {model_explainability_monitor.latest_baselining_job_name}")
```

Pelanggaran Drift Atribusi Fitur Model

Pekerjaan drift atribusi fitur mengevaluasi kendala dasar yang disediakan oleh konfigurasi dasar terhadap hasil analisis saat ini. `MonitoringExecution` Jika pelanggaran terdeteksi, pekerjaan mencantumkannya ke file constraint_violations.json di lokasi keluaran eksekusi, dan menandai status eksekusi sebagai. Menafsirkan hasil

Berikut adalah skema file pelanggaran drift atribusi fitur.

- `label`— Nama label, konfigurasi analisis pekerjaan `label_headers` atau placeholder seperti. `"label0"`
- `metric_name`— Nama metode analisis keterjelasan. Saat ini hanya mendukung `shap`.
- `constraint_check_type`— Jenis pelanggaran yang dipantau. Saat ini hanya mendukung `feature_attribution_drift_check`.
- `description`— Pesan deskriptif untuk menjelaskan pelanggaran.

```
{
    "version": "1.0",
    "violations": [{
        "label": "string",
        "metric_name": "string",
        "constraint_check_type": "string",
        "description": "string"
    }]
}
```

Untuk setiap label di *`explanations`* bagian tersebut, pekerjaan pemantauan menghitung skor NdCG dari nilai SHAP globalnya di file batasan dasar dan dalam file hasil analisis pekerjaan (analysis.json). Jika skor kurang dari 0,9, maka pelanggaran dicatat. Nilai SHAP global gabungan dievaluasi, sehingga tidak ada *“*`feature`*”* bidang dalam entri pelanggaran. Output berikut memberikan contoh beberapa pelanggaran yang dicatat.

```
{
    "version": "1.0",
    "violations": [{
        "label": "label0",
        "metric_name": "shap",
        "constraint_check_type": "feature_attribution_drift_check",
```

```
        "description": "Feature attribution drift 0.7639720923277322 exceeds threshold
 0.9"
    }, {
        "label": "label1",
        "metric_name": "shap",
        "constraint_check_type": "feature_attribution_drift_check",
        "description": "Feature attribution drift 0.7323763972092327 exceeds threshold
 0.9"
    }]
}
```

Mengonfigurasi Parameter untuk Memonfigurasi Atribusi Drift

Amazon SageMaker Clarify Explainability Monitor menggunakan kembali subset parameter yang digunakan dalam konfigurasi analisis. Konfigurasikan Analisis Parameter berikut harus disediakan dalam file JSON dan jalur harus disediakan dalam `ConfigUri` parameter. `ModelExplainabilityAppSpecification`

- **"version"**— (Opsional) Versi skema dari file konfigurasi. Jika tidak disediakan, versi terbaru yang didukung digunakan.
- **"headers"**— (Opsional) Daftar nama fitur dalam dataset. Analisis penjelasan tidak memerlukan label.
- **"methods"**— Daftar metode dan parameternya untuk analisis dan laporan. Jika ada bagian yang dihilangkan, maka itu tidak dihitung.
  - `"shap"`- (Opsional) Bagian tentang perhitungan nilai SHAP.
    - `"baseline"`— (Opsional) Daftar baris (setidaknya satu), atau URI objek Amazon Simple Storage Service Amazon S3. Untuk digunakan sebagai dataset dasar (juga dikenal sebagai dataset latar belakang) dalam algoritma Kernel SHAP. Formatnya harus sama dengan format dataset. Setiap baris harus berisi hanya kolom fitur (atau nilai). Sebelum Anda mengirim setiap baris ke model, hilangkan kolom apa pun yang harus dikecualikan.
    - `"num_samples"`— Jumlah sampel yang akan digunakan dalam algoritma Kernel SHAP. Angka ini menentukan ukuran kumpulan data sintetis yang dihasilkan untuk menghitung nilai SHAP. Jika tidak disediakan, maka pekerjaan SageMaker Clarify memilih nilai berdasarkan jumlah fitur.
    - `"agg_method"`— Metode agregasi untuk nilai SHAP global. Nilai yang valid adalah sebagai berikut:
      - `"mean_abs"`— Rata-rata nilai SHAP absolut untuk semua contoh.

    - `"median"`— Median nilai SHAP untuk semua instance.
    - `"mean_sq"`— Rata-rata nilai SHAP kuadrat untuk semua instance.
  - `"use_logit"`— (Opsional) Nilai Boolean untuk menunjukkan apakah fungsi logit akan diterapkan pada prediksi model. Jika `"use_logit"` ya`true`, maka nilai SHAP memiliki unit log-odds. Nilai default-nya adalah `false`.
  - `"save_local_shap_values"`— (Opsional) Nilai Boolean untuk menunjukkan apakah nilai SHAP lokal akan disimpan di lokasi output. Gunakan `true` untuk menyelamatkan mereka. Gunakan `false` untuk tidak menyelamatkan mereka. Default-nya adalah `false`.
- "**`predictor`**"— (Opsional untuk titik akhir real-time, diperlukan untuk transformasi batch) Bagian tentang parameter model, diperlukan jika `"shap"` dan `"post_training_bias"` bagian ada.
  - `"model_name"`— Nama model yang dibuat oleh `CreateModel` API, dengan mode kontainer sebagai`SingleModel`.
  - `"instance_type"`— Jenis instance untuk titik akhir bayangan.
  - `"initial_instance_count"`— Jumlah instans untuk titik akhir bayangan.
  - `"content_type"`— (Opsional) Format input model yang akan digunakan untuk mendapatkan kesimpulan dengan titik akhir bayangan. Nilai yang valid adalah `"text/csv"` untuk CSV, `"application/jsonlines"` untuk JSON Lines, untuk Apache Parquet, dan `application/x-parquet` `application/x-image` untuk memungkinkan penjelasan Computer Vision. Nilai defaultnya sama dengan `dataset_type` formatnya.
  - `"accept_type"`— (Opsional) Format keluaran model yang akan digunakan untuk mendapatkan kesimpulan dengan titik akhir bayangan. Nilai yang valid adalah `"text/csv"` untuk CSV, `"application/jsonlines"` untuk JSON Lines. Jika dihilangkan, SageMaker Clarify menggunakan tipe data respons dari data yang diambil.
  - `"content_template"`— (Opsional) String template yang digunakan untuk membangun input model dari instance dataset. Ini hanya digunakan ketika `"content_type"` ada`"application/jsonlines"`. Template seharusnya hanya memiliki satu placeholder`$features`, yang digantikan oleh daftar fitur saat runtime. Misalnya, diberikan`"content_template":"{\"myfeatures\":$features}"`, jika sebuah instance (tidak ada label)`1,2,3`, maka input model menjadi JSON Lines`'{"myfeatures":[1,2,3]}'`.
  - `"label_headers"`— (Opsional) Daftar nilai yang `"label"` diambil dalam dataset. Mengaitkan skor yang dikembalikan oleh titik akhir model atau pekerjaan transformasi batch dengan nilai label yang sesuai. Jika disediakan, maka laporan analisis menggunakan header alih-alih placeholder seperti. `“label0”`

Parameter lain harus disediakan di `EndpointInput` (untuk titik akhir waktu nyata) atau `BatchTransformInput` (untuk pekerjaan transformasi batch) dari `ModelExplainabilityJobInput`API.

- `FeaturesAttribute`— Parameter ini diperlukan jika format data input pekerjaan endpoint atau batch. `"application/jsonlines"` Ini adalah JMESPath yang digunakan untuk menemukan kolom fitur jika format dataset adalah JSON Lines.
- `ProbabilityAttribute`— Indeks atau lokasi JMESPath dalam output model untuk probabilitas. Jika keluaran model adalah JSON Lines dengan daftar label dan probabilitas, misalnya, maka label yang sesuai dengan probabilitas maksimum dipilih untuk perhitungan bias.

Contoh File Konfigurasi JSON untuk Kumpulan Data Garis CSV dan JSON

Berikut adalah contoh file JSON yang digunakan untuk mengonfigurasi kumpulan data CSV dan JSON Lines untuk memantaunya untuk penyimpangan atribusi fitur.

Topik

- Set data CSV
- Set Data JSON Lines

Set data CSV

Pertimbangkan kumpulan data yang memiliki tiga kolom fitur numerik, seperti pada contoh berikut.

```
0.5814568701544718, 0.6651538910132964, 0.3138080342665499
0.6711642728531724, 0.7466687034026017, 0.1215477472819713
0.0453256543003371, 0.6377430803264152, 0.3558625219713576
0.4785191813363956, 0.0265841045263860, 0.0376935084990697
```

Asumsikan bahwa keluaran model memiliki dua kolom, di mana yang pertama adalah label yang diprediksi dan yang kedua adalah probabilitas, seperti pada contoh berikut.

```
1, 0.5385257417814224
```

Contoh file konfigurasi JSON berikut menunjukkan bagaimana kumpulan data CSV ini dapat dikonfigurasi.

```
{
```

```
    "headers": [
        "feature_1",
        "feature_2",
        "feature_3"
    ],
    "methods": {
        "shap": {
            "baseline": [
                [0.4441164946610942, 0.5190374448171748, 0.20722795300473712]
            ],
            "num_samples": 100,
            "agg_method": "mean_abs"
        }
    },
    "predictor": {
        "model_name": "my_model",
        "instance_type": "ml.m5.xlarge",
        "initial_instance_count": 1
    }
}
```

Label yang diprediksi dipilih oleh "`ProbabilityAttribute`" parameter. Penomoran berbasis nol digunakan, jadi 1 menunjukkan kolom kedua dari output model.

```
"EndpointInput": {
    ...
    "ProbabilityAttribute": 1
    ...
}
```

Set Data JSON Lines

Pertimbangkan kumpulan data yang memiliki empat kolom fitur dan satu kolom label, di mana fitur pertama dan labelnya adalah biner, seperti pada contoh berikut.

```
{"features":[0, 0.5814568701544718, 0.6651538910132964, 0.3138080342665499], "label":0}
{"features":[1, 0.6711642728531724, 0.7466687034026017, 0.1215477472819713], "label":1}
{"features":[0, 0.0453256543003371, 0.6377430803264152, 0.3558625219713576], "label":1}
{"features":[1, 0.4785191813363956, 0.0265841045263860, 0.0376935084990697], "label":1}
```

Masukan model sama dengan format dataset, dan output modelnya adalah JSON Lines, seperti pada contoh berikut.

```
{"predicted_label":1, "probability":0.5385257417814224}
```

Dalam contoh berikut, file konfigurasi JSON menunjukkan bagaimana dataset JSON Lines ini dapat dikonfigurasi.

```
{
    "headers": [
        "feature_1",
        "feature_2",
        "feature_3"
    ],
    "methods": {
        "shap": {
            "baseline": [
                {"features":[0.4441164946610942, 0.5190374448171748,
 0.20722795300473712]}
            ],
            "num_samples": 100,
            "agg_method": "mean_abs"
        }
    },
    "predictor": {
        "model_name": "my_model",
        "instance_type": "ml.m5.xlarge",
        "initial_instance_count": 1,
        "content_template":"{\"features\":$features}"
    }
}
```

Kemudian nilai `"features"` parameter di `EndpointInput` (untuk titik akhir waktu nyata) atau `BatchTransformInput` (untuk pekerjaan transformasi batch) digunakan untuk menemukan fitur dalam kumpulan data, dan nilai `"probability"` parameter memilih nilai probabilitas dari keluaran model.

```
"EndpointInput": {
    ...
    "FeaturesAttribute": "features",
    "ProbabilityAttribute": "probability",
```

```
    ...
}
```

Jadwal Fitur Atribut Pekerjaan Pemantauan Drift

Setelah Anda membuat baseline SHAP Anda, Anda dapat memanggil `create_monitoring_schedule()` metode instance `ModelExplainabilityMonitor` kelas Anda untuk menjadwalkan monitor penjelasan model per jam. Bagian berikut menunjukkan cara membuat monitor penjelasan model untuk model yang diterapkan ke titik akhir waktu nyata serta untuk pekerjaan transformasi batch.

> **Important**
>
> Anda dapat menentukan input transformasi batch atau input titik akhir, tetapi tidak keduanya, saat Anda membuat jadwal pemantauan.

Jika pekerjaan baselining telah dikirimkan, monitor secara otomatis mengambil konfigurasi analisis dari pekerjaan baselining. Namun, jika Anda melewati langkah dasar atau kumpulan data pengambilan memiliki sifat yang berbeda dari kumpulan data pelatihan, Anda harus menyediakan konfigurasi analisis. `ModelConfig`diperlukan oleh karena `ExplainabilityAnalysisConfig` alasan yang sama yang diperlukan untuk pekerjaan baselining. Perhatikan bahwa hanya fitur yang diperlukan untuk menghitung atribusi fitur, jadi Anda harus mengecualikan pelabelan Ground Truth.

Pemantauan drift atribusi fitur untuk model yang digunakan ke titik akhir waktu nyata

Untuk menjadwalkan monitor penjelasan model untuk titik akhir real-time, teruskan `EndpointInput` instance Anda ke `endpoint_input` argumen `ModelExplainabilityMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

model_exp_model_monitor = ModelExplainabilityMonitor(
   role=sagemaker.get_execution_role(),
   ...
)

schedule = model_exp_model_monitor.create_monitoring_schedule(
   monitor_schedule_name=schedule_name,
```

```
   post_analytics_processor_script=s3_code_postprocessor_uri,
   output_s3_uri=s3_report_path,
   statistics=model_exp_model_monitor.baseline_statistics(),
   constraints=model_exp_model_monitor.suggested_constraints(),
   schedule_cron_expression=CronExpressionGenerator.hourly(),
   enable_cloudwatch_metrics=True,
   endpoint_input=EndpointInput(
        endpoint_name=endpoint_name,
        destination="/opt/ml/processing/input/endpoint",
    )
)
```

Fitur pemantauan drift atribusi untuk pekerjaan transformasi batch

Untuk menjadwalkan monitor penjelasan model untuk pekerjaan transformasi batch, teruskan `BatchTransformInput` instance Anda ke `batch_transform_input` argumen `ModelExplainabilityMonitor` instance Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
from sagemaker.model_monitor import CronExpressionGenerator

model_exp_model_monitor = ModelExplainabilityMonitor(
   role=sagemaker.get_execution_role(),
   ...
)

schedule = model_exp_model_monitor.create_monitoring_schedule(
   monitor_schedule_name=schedule_name,
   post_analytics_processor_script=s3_code_postprocessor_uri,
   output_s3_uri=s3_report_path,
   statistics=model_exp_model_monitor.baseline_statistics(),
   constraints=model_exp_model_monitor.suggested_constraints(),
   schedule_cron_expression=CronExpressionGenerator.hourly(),
   enable_cloudwatch_metrics=True,
   batch_transform_input=BatchTransformInput(
        destination="opt/ml/processing/data",
        model_name="batch-fraud-detection-model",
        input_manifests_s3_uri="s3://my-bucket/batch-fraud-detection/on-schedule-
monitoring/in/",
        excludeFeatures="0",
   )
)
```

Periksa Laporan untuk Penyimpangan Atribut Fitur dalam Model Produksi

Setelah jadwal yang Anda atur dimulai secara default, Anda harus menunggu eksekusi pertamanya dimulai, dan kemudian menghentikan jadwal untuk menghindari biaya yang dikenakan.

Untuk memeriksa laporan, gunakan kode berikut:

```
schedule_desc = model_explainability_monitor.describe_schedule()
execution_summary = schedule_desc.get("LastMonitoringExecutionSummary")
if execution_summary and execution_summary["MonitoringExecutionStatus"] in
 ["Completed", "CompletedWithViolations"]:
    last_model_explainability_monitor_execution =
 model_explainability_monitor.list_executions()[-1]
    last_model_explainability_monitor_execution_report_uri =
 last_model_explainability_monitor_execution.output.destination
    print(f'Report URI: {last_model_explainability_monitor_execution_report_uri}')
    last_model_explainability_monitor_execution_report_files =
 sorted(S3Downloader.list(last_model_explainability_monitor_execution_report_uri))
    print("Found Report Files:")
    print("\n ".join(last_model_explainability_monitor_execution_report_files))
else:
    last_model_explainability_monitor_execution = None
    print("====STOP==== \n No completed executions to inspect further. Please wait till
 an execution completes or investigate previously reported failures.")
```

Jika ada pelanggaran dibandingkan dengan baseline, mereka tercantum di sini:

```
if last_model_explainability_monitor_execution:
    model_explainability_violations =
 last_model_explainability_monitor_execution.constraint_violations()
    if model_explainability_violations:
        print(model_explainability_violations.body_dict)
```

Jika model Anda diterapkan ke titik akhir real-time, Anda dapat melihat visualisasi di SageMaker Studio hasil analisis dan CloudWatch metrik dengan memilih tab Endpoints, lalu mengklik dua kali titik akhir.

CloudWatch Metrik untuk Analisis Drift Fitur

Panduan ini menunjukkan CloudWatch metrik dan propertinya yang dapat Anda gunakan untuk analisis drift atribut fitur di SageMaker Clarify. Pekerjaan pemantauan drift atribut fitur menghitung dan menerbitkan dua jenis metrik:

- Nilai SHAP global dari setiap fitur.

> **Note**
>
> Nama metrik ini menambahkan nama fitur yang disediakan oleh konfigurasi analisis pekerjaan kefeature_. Misalnya, feature_X adalah nilai SHAP global untuk fiturX.

- ExpectedValueDari metrik.

Metrik ini dipublikasikan ke CloudWatch namespace berikut:

- Untuk titik akhir waktu nyata: aws/sagemaker/Endpoints/explainability-metrics
- Untuk pekerjaan transformasi batch: aws/sagemaker/ModelMonitoring/explainability-metrics

Setiap metrik memiliki properti berikut:

- Endpoint: Nama titik akhir yang dipantau, jika berlaku.
- MonitoringSchedule: Nama jadwal untuk tugas pemantauan.
- ExplainabilityMethod: Metode yang digunakan untuk menghitung nilai-nilai Shapley. Pilih KernelShap.
- Label: Nama yang disediakan oleh konfigurasi analisis pekerjaanlabel_headers, atau placeholder seperti. label0
- ValueType: Jenis nilai yang dikembalikan oleh metrik. Pilih salah satu GlobalShapValues atauExpectedValue.

Untuk menghentikan pekerjaan pemantauan dari penerbitan metrik, atur publish_cloudwatch_metrics ke Disabled dalam Environment peta definisi pekerjaan penjelasan model.

Jadwalkan pekerjaan pemantauan

Amazon SageMaker Model Monitor memberi Anda kemampuan untuk memantau data yang dikumpulkan dari titik akhir waktu nyata Anda. Anda dapat memantau data Anda pada jadwal berulang, atau Anda dapat memantaunya satu kali, segera. Anda dapat membuat jadwal pemantauan dengan CreateMonitoringScheduleAPI.

Dengan jadwal pemantauan, SageMaker dapat mulai memproses pekerjaan untuk menganalisis data yang dikumpulkan selama periode tertentu. Dalam pekerjaan pemrosesan, SageMaker bandingkan kumpulan data untuk analisis saat ini dengan statistik dasar dan kendala yang Anda berikan. Kemudian, SageMaker buat laporan pelanggaran. Selain itu, CloudWatch metrik dipancarkan untuk setiap fitur yang dianalisis.

SageMaker menyediakan wadah bawaan untuk melakukan analisis pada kumpulan data tabel. Atau, Anda dapat memilih untuk membawa wadah Anda sendiri seperti yang diuraikan dalam Bawa Kontainer Anda Sendiri topik.

Anda dapat membuat jadwal pemantauan model untuk titik akhir real-time atau pekerjaan transformasi batch Anda. Gunakan sumber daya dasar (kendala dan statistik) untuk membandingkan dengan lalu lintas real-time atau input pekerjaan batch.

Example tugas dasar

Dalam contoh berikut, kumpulan data pelatihan yang digunakan untuk melatih model diunggah ke Amazon S3. Jika Anda sudah memilikinya di Amazon S3, Anda dapat mengarahkannya secara langsung.

```
# copy over the training dataset to Amazon S3 (if you already have it in Amazon S3, you
 could reuse it)
baseline_prefix = prefix + '/baselining'
baseline_data_prefix = baseline_prefix + '/data'
baseline_results_prefix = baseline_prefix + '/results'

baseline_data_uri = 's3://{}/{}'.format(bucket,baseline_data_prefix)
baseline_results_uri = 's3://{}/{}'.format(bucket, baseline_results_prefix)
print('Baseline data uri: {}'.format(baseline_data_uri))
print('Baseline results uri: {}'.format(baseline_results_uri))
```

```
training_data_file = open("test_data/training-dataset-with-header.csv", 'rb')
s3_key = os.path.join(baseline_prefix, 'data', 'training-dataset-with-header.csv')
boto3.Session().resource('s3').Bucket(bucket).Object(s3_key).upload_fileobj(training_data_file)
```

Example jadwal untuk analisis berulang

Jika Anda menjadwalkan monitor model untuk titik akhir real-time, gunakan batasan dasar dan statistik untuk membandingkan dengan lalu lintas waktu nyata. Cuplikan kode berikut menunjukkan format umum yang Anda gunakan untuk menjadwalkan monitor model untuk titik akhir real-time. Contoh ini menjadwalkan monitor model untuk berjalan setiap jam.

```
from sagemaker.model_monitor import CronExpressionGenerator
from time import gmtime, strftime

mon_schedule_name = 'my-model-monitor-schedule-' + strftime("%Y-%m-%d-%H-%M-%S",
 gmtime())
my_default_monitor.create_monitoring_schedule(
    monitor_schedule_name=mon_schedule_name,
    endpoint_input=EndpointInput(
        endpoint_name=endpoint_name,
        destination="/opt/ml/processing/input/endpoint"
    ),
    post_analytics_processor_script=s3_code_postprocessor_uri,
    output_s3_uri=s3_report_path,
    statistics=my_default_monitor.baseline_statistics(),
    constraints=my_default_monitor.suggested_constraints(),
    schedule_cron_expression=CronExpressionGenerator.hourly(),
    enable_cloudwatch_metrics=True,
)
```

Example jadwal untuk analisis satu kali

Anda juga dapat menjadwalkan analisis untuk dijalankan sekali tanpa berulang dengan meneruskan argumen seperti berikut ke `create_monitoring_schedule` metode:

```
    schedule_cron_expression=CronExpressionGenerator.now(),
    data_analysis_start_time="-PT1H",
    data_analysis_end_time="-PT0H",
```

Dalam argumen ini, `schedule_cron_expression` parameter menjadwalkan analisis untuk dijalankan sekali, segera, dengan nilainya`CronExpressionGenerator.now()`. Untuk jadwal apa pun dengan pengaturan ini, `data_analysis_end_time` parameter `data_analysis_start_time` dan diperlukan. Parameter ini mengatur waktu mulai dan waktu akhir jendela analisis. Tentukan waktu ini sebagai offset yang relatif terhadap waktu saat ini, dan gunakan format durasi ISO 8601. Dalam contoh ini, waktu `-PT1H` dan `-PT0H` tentukan jendela antara satu jam di masa lalu dan waktu saat ini. Dengan jadwal ini, analisis hanya mengevaluasi data yang dikumpulkan selama jendela yang ditentukan.

Example jadwal untuk pekerjaan transformasi batch

Cuplikan kode berikut menunjukkan format umum yang Anda gunakan untuk menjadwalkan monitor model untuk pekerjaan transformasi batch.

```
from sagemaker.model_monitor import (
    CronExpressionGenerator,
    BatchTransformInput,
    MonitoringDatasetFormat,
)
from time import gmtime, strftime

mon_schedule_name = 'my-model-monitor-schedule-' + strftime("%Y-%m-%d-%H-%M-%S",
 gmtime())
my_default_monitor.create_monitoring_schedule(
    monitor_schedule_name=mon_schedule_name,
    batch_transform_input=BatchTransformInput(
        destination="opt/ml/processing/input",
        data_captured_destination_s3_uri=s3_capture_upload_path,
        dataset_format=MonitoringDatasetFormat.csv(header=False),
    ),
    post_analytics_processor_script=s3_code_postprocessor_uri,
    output_s3_uri=s3_report_path,
    statistics=my_default_monitor.baseline_statistics(),
    constraints=my_default_monitor.suggested_constraints(),
    schedule_cron_expression=CronExpressionGenerator.hourly(),
    enable_cloudwatch_metrics=True,
)
```

```
desc_schedule_result = my_default_monitor.describe_schedule()
print('Schedule status: {}'.format(desc_schedule_result['MonitoringScheduleStatus']))
```

Ekspresi cron untuk jadwal pemantauan

Untuk memberikan rincian untuk jadwal pemantauan, gunakan `ScheduleConfig`, yang merupakan `cron` ungkapan yang menjelaskan rincian tentang jadwal pemantauan.

Amazon SageMaker Model Monitor mendukung `cron` ekspresi berikut:

- Untuk mengatur tugas dimulai setiap jam, gunakan yang berikut ini:

  `Hourly: cron(0 * ? * * *)`
- Untuk menjalankan pekerjaan setiap hari, gunakan yang berikut ini:

  `cron(0 [00-23] ? * * *)`
- Jalankan pekerjaan satu kali, segera, gunakan kata kunci berikut:

NOW

Misalnya, berikut ini adalah `cron` ekspresi yang valid:

- Setiap hari pukul 12 PM UTC: `cron(0 12 ? * * *)`
- Setiap hari pukul 12 pagi UTC: `cron(0 0 ? * * *)`

Untuk mendukung menjalankan setiap 6, 12 jam, Model Monitor mendukung ekspresi berikut:

`cron(0 [00-23]/[01-24] ? * * *)`

Misalnya, berikut ini adalah `cron` ekspresi yang valid:

- Setiap 12 jam, mulai pukul 5 sore UTC: `cron(0 17/12 ? * * *)`
- Setiap dua jam, mulai pukul 12 pagi UTC: `cron(0 0/2 ? * * *)`

> **Catatan**
>
> - Meskipun `cron` ekspresi diatur untuk dimulai pada 17.00 UTC, perhatikan bahwa mungkin ada penundaan 0-20 menit dari waktu aktual yang diminta untuk menjalankan eksekusi.
> - Jika Anda ingin menjalankan jadwal harian, jangan berikan parameter ini. SageMakermemilih waktu untuk berlari setiap hari.
> - Saat ini, SageMaker hanya mendukung tingkat integer per jam antara 1 jam dan 24 jam.

Mengkonfigurasi kebijakan kontrol layanan untuk jadwal pemantauan

Anda harus menentukan parameter pekerjaan pemantauan saat Anda membuat atau memperbarui jadwal untuk itu dengan CreateMonitoringScheduleAPI atau UpdateMonitoringScheduleAPI, masing-masing. Tergantung pada kasus penggunaan Anda, Anda dapat melakukannya melalui salah satu cara berikut:

- Anda dapat menentukan MonitoringJobDefinitionbidang MonitoringScheduleConfig, ketika Anda memanggil `CreateMonitoringSchedule` atau`UpdateMonitoringSchedule`. Anda dapat menggunakan ini hanya untuk membuat atau memperbarui jadwal untuk pekerjaan pemantauan kualitas data.

- Anda dapat menentukan nama definisi pekerjaan pemantauan, yang telah Anda buat, untuk `MonitoringJobDefinitionName` bidang`MonitoringScheduleConfig`, saat Anda memanggil `CreateMonitoringSchedule` atau`UpdateMonitoringSchedule`. Anda dapat menggunakan ini untuk definisi pekerjaan apa pun yang Anda buat dengan salah satu API berikut:
  - CreateDataQualityJobDefinition
  - CreateModelQualityJobDefinition
  - CreateModelBiasJobDefinition
  - CreateModelExplainabilityJobDefinition

  Jika Anda ingin menggunakan SageMaker Python SDK untuk membuat atau memperbarui jadwal, maka Anda harus menggunakan proses ini.

Proses yang disebutkan di atas saling eksklusif, yaitu, Anda dapat menentukan `MonitoringJobDefinition` bidang atau `MonitoringJobDefinitionName` bidang saat membuat atau memperbarui jadwal pemantauan.

Saat Anda membuat definisi pekerjaan pemantauan, atau menentukannya di `MonitoringJobDefinition` bidang, Anda dapat mengatur parameter keamanan, seperti `NetworkConfig` dan`VolumeKmsKeyId`. Sebagai administrator, Anda mungkin ingin agar parameter ini selalu disetel ke nilai tertentu, sehingga pekerjaan pemantauan selalu berjalan di lingkungan yang aman. Untuk memastikan hal ini, siapkan kebijakan kontrol Layanan (SCP) yang sesuai. SCP adalah jenis kebijakan organisasi yang dapat digunakan untuk mengelola izin di organisasi Anda.

Contoh berikut menunjukkan SCP yang dapat Anda gunakan untuk memastikan bahwa parameter infrastruktur diatur dengan benar saat membuat atau memperbarui jadwal untuk pekerjaan pemantauan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "sagemaker:CreateDataQualityJobDefinition",
                "sagemaker:CreateModelBiasJobDefinition",
                "sagemaker:CreateModelExplainabilityJobDefinition",
                "sagemaker:CreateModelQualityJobDefinition"
            ],
            "Resource": "arn:*:sagemaker:*:*:*",
```

```
            "Condition": {
                "Null": {
                    "sagemaker:VolumeKmsKey":"true",
                    "sagemaker:VpcSubnets": "true",
                    "sagemaker:VpcSecurityGroupIds": "true"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "sagemaker:CreateDataQualityJobDefinition",
                "sagemaker:CreateModelBiasJobDefinition",
                "sagemaker:CreateModelExplainabilityJobDefinition",
                "sagemaker:CreateModelQualityJobDefinition"
            ],
            "Resource": "arn:*:sagemaker:*:*:*",
            "Condition": {
                "Bool": {
                    "sagemaker:InterContainerTrafficEncryption": "false"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "sagemaker:CreateMonitoringSchedule",
                "sagemaker:UpdateMonitoringSchedule",
            ],
            "Resource": "arn:*:sagemaker:*:*:monitoring-schedule/*",
            "Condition": {
                "Null": {
                    "sagemaker:ModelMonitorJobDefinitionName": "true"
                }
            }
        }
    ]
}
```

Dua aturan pertama dalam contoh, memastikan bahwa parameter keamanan selalu ditetapkan untuk memantau definisi pekerjaan. Aturan terakhir mengharuskan siapa pun, di organisasi Anda, membuat atau memperbarui jadwal, harus selalu menentukan `MonitoringJobDefinitionName` bidang.

Ini memastikan bahwa tidak ada seorang pun di organisasi Anda, dapat menetapkan nilai tidak aman untuk parameter keamanan dengan menentukan `MonitoringJobDefinition` bidang, saat membuat atau memperbarui jadwal.

Amazon SageMaker Model Monitor wadah bawaan

SageMaker menyediakan gambar bawaan yang disebut `sagemaker-model-monitor-analyzer` yang memberi Anda berbagai kemampuan pemantauan model, termasuk saran kendala, pembuatan statistik, validasi kendala terhadap baseline, dan metrik Amazon yang memancarkan. CloudWatch Gambar ini didasarkan pada Spark versi 3.3.0 dan dibangun dengan Deequ versi 2.0.2.

> **Note**
>
> Anda tidak dapat menarik `sagemaker-model-monitor-analyzer` gambar bawaan secara langsung. Anda dapat menggunakan `sagemaker-model-monitor-analyzer` gambar saat mengirimkan pekerjaan pemrosesan atau pemantauan dasar menggunakan salah satu SDKAWS.

Gunakan SageMaker Python SDK (lihat `image_uris.retrieve` di panduan referensi SDK SageMaker Python) untuk menghasilkan URI gambar ECR untuk Anda, atau tentukan URI gambar ECR secara langsung. Gambar prebuilt untuk SageMaker Model Monitor dapat diakses sebagai berikut:

<ACCOUNT_ID>`.dkr.ecr.`*<REGION_NAME>*`.amazonaws.com/sagemaker-model-monitor-analyzer`

Sebagai contoh: `159807026194.dkr.ecr.us-west-2.amazonaws.com/sagemaker-model-monitor-analyzer`

Jika Anda berada di suatu AWS wilayah di China, gambar prebuilt untuk SageMaker Model Monitor dapat diakses sebagai berikut:

<ACCOUNT_ID>`.dkr.ecr.`*<REGION_NAME>*`.amazonaws.com.cn/sagemaker-model-monitor-analyzer`

Untuk ID akun dan nama AWS Wilayah, lihat Jalur Registri Docker dan Kode Contoh.

Untuk menulis wadah analisis Anda sendiri, lihat kontrak kontainer yang dijelaskan diKustomisasi pemantauan.

Menafsirkan hasil

Setelah Anda menjalankan pekerjaan pemrosesan dasar dan memperoleh statistik dan kendala untuk kumpulan data Anda, Anda dapat menjalankan tugas pemantauan yang menghitung statistik dan mencantumkan setiap pelanggaran yang dihadapi relatif terhadap batasan dasar. CloudWatch Metrik Amazon juga dilaporkan di akun Anda secara default. Untuk informasi tentang melihat hasil pemantauan di Amazon SageMaker Studio, lihatVisualisasikan hasil untuk titik akhir real-time di Amazon Studio SageMaker .

Daftar Eksekusi

Jadwal mulai memantau pekerjaan pada interval yang ditentukan. Kode berikut mencantumkan lima eksekusi terbaru. Jika Anda menjalankan kode ini setelah membuat jadwal per jam, eksekusi mungkin kosong, dan Anda mungkin harus menunggu sampai Anda melewati batas jam (dalam UTC) untuk melihat eksekusi dimulai. Kode berikut mencakup logika untuk menunggu.

```
mon_executions = my_default_monitor.list_executions()
print("We created a hourly schedule above and it will kick off executions ON the hour
 (plus 0 - 20 min buffer.\nWe will have to wait till we hit the hour...")

while len(mon_executions) == 0:
    print("Waiting for the 1st execution to happen...")
    time.sleep(60)
    mon_executions = my_default_monitor.list_executions()
```

Memeriksa Eksekusi Tertentu

Pada langkah sebelumnya, Anda mengambil eksekusi terjadwal terbaru yang selesai atau gagal. Anda dapat menjelajahi apa yang benar atau salah. Status terminal adalah:

- `Completed`— Eksekusi pemantauan selesai dan tidak ada masalah yang ditemukan dalam laporan pelanggaran.
- `CompletedWithViolations`— Eksekusi selesai, tetapi pelanggaran kendala terdeteksi.
- `Failed`— Eksekusi pemantauan gagal, mungkin karena kesalahan klien (misalnya, masalah peran) atau masalah infrastruktur. Untuk mengidentifikasi penyebabnya, lihat `FailureReason` dan`ExitMessage`.

```
latest_execution = mon_executions[-1] # latest execution's index is -1, previous is -2
 and so on..
time.sleep(60)
latest_execution.wait(logs=False)

print("Latest execution status: {}".format(latest_execution.describe()
['ProcessingJobStatus']))
print("Latest execution result: {}".format(latest_execution.describe()['ExitMessage']))

latest_job = latest_execution.describe()
if (latest_job['ProcessingJobStatus'] != 'Completed'):
        print("====STOP==== \n No completed executions to inspect further. Please wait
 till an execution completes or investigate previously reported failures.")
```

```
report_uri=latest_execution.output.destination
print('Report Uri: {}'.format(report_uri))
```

Daftar Laporan yang Dihasilkan

Buat daftar laporan yang dihasilkanGunakan kode berikut untuk membuat daftar laporan yang dihasilkan.

```
from urllib.parse import urlparse
s3uri = urlparse(report_uri)
report_bucket = s3uri.netloc
report_key = s3uri.path.lstrip('/')
print('Report bucket: {}'.format(report_bucket))
print('Report key: {}'.format(report_key))

s3_client = boto3.Session().client('s3')
result = s3_client.list_objects(Bucket=report_bucket, Prefix=report_key)
report_files = [report_file.get("Key") for report_file in result.get('Contents')]
print("Found Report Files:")
print("\n ".join(report_files))
```

Laporan Pelanggaran

Jika ada pelanggaran dibandingkan dengan baseline, mereka dihasilkan dalam laporan pelanggaran. Gunakan kode berikut untuk membuat daftar pelanggaran.

```
violations = my_default_monitor.latest_monitoring_constraint_violations()
```

```
pd.set_option('display.max_colwidth', -1)
constraints_df = pd.io.json.json_normalize(violations.body_dict["violations"])
constraints_df.head(10)
```

Ini hanya berlaku untuk kumpulan data yang berisi data tabular. File skema berikut menentukan statistik yang dihitung dan pelanggaran yang dipantau.

File Output untuk Dataset Tabular

Format nama file	Deskripsi
`statistics.json`	Berisi statistik kolumnar untuk setiap fitur dalam kumpulan data yang dianalisis. Lihat skema file ini di topik berikutnya. **Note** File ini dibuat hanya untuk pemantauan kualitas data.
`constraint_violations.json`	Berisi daftar pelanggaran yang ditemukan dalam kumpulan data saat ini dibandingkan dengan statistik dasar dan file kendala yang ditentukan dalam dan jalur. `baseline_constaints` `baseline_statistics`

Amazon SageMaker Model Monitor wadah bawaanIni menyimpan satu set CloudWatch metrik Amazon untuk setiap fitur secara default.

Kode kontainer dapat memancarkan CloudWatch metrik di lokasi ini:. `/opt/ml/output/metrics/cloudwatch`

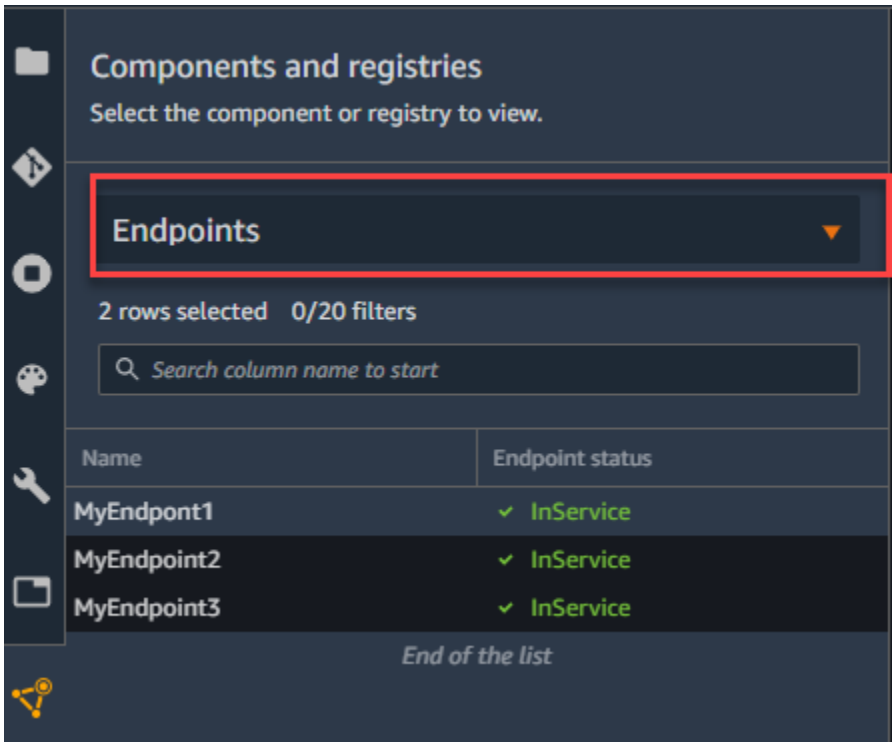
Visualisasikan hasil untuk titik akhir real-time di Amazon Studio SageMaker

Jika Anda memantau titik akhir real-time, Anda juga dapat memvisualisasikan hasilnya di Amazon SageMaker Studio. Anda dapat melihat detail pekerjaan pemantauan apa pun, dan Anda dapat

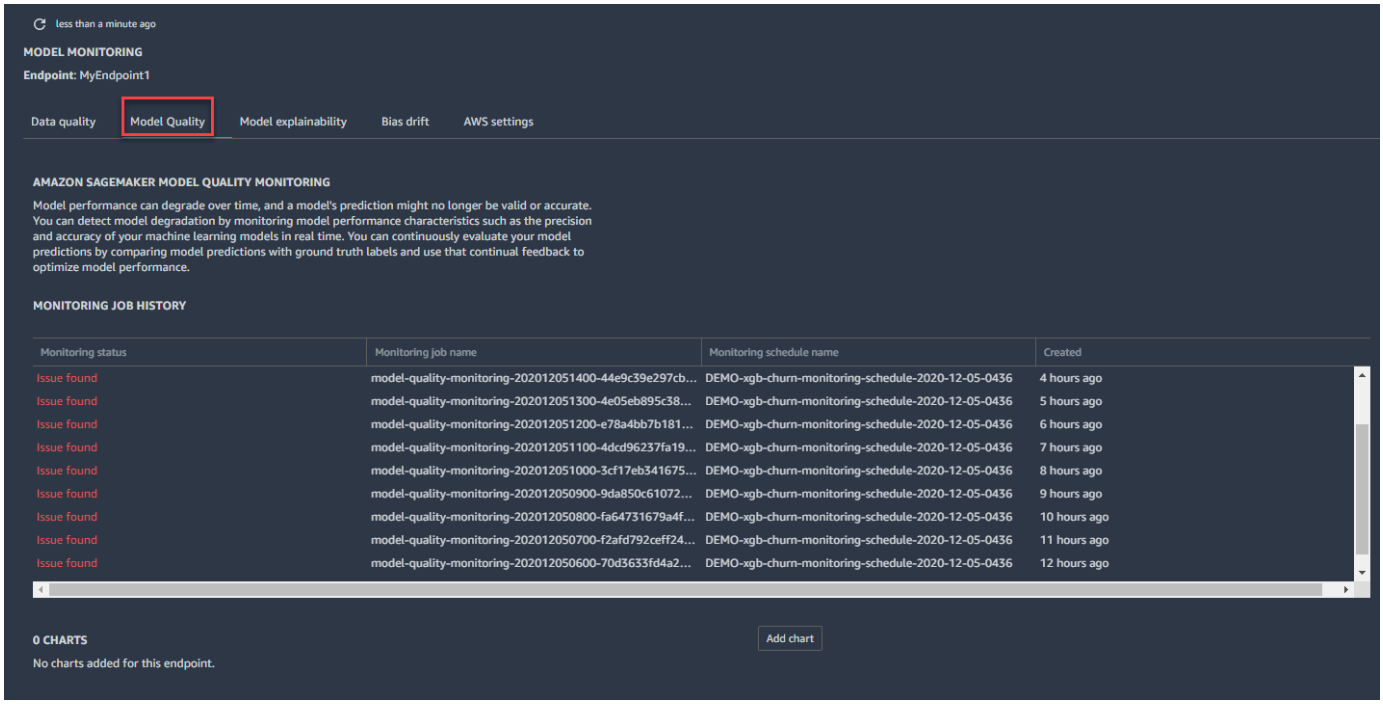
membuat bagan yang menunjukkan nilai dasar dan yang diambil untuk metrik apa pun yang dihitung oleh pekerjaan pemantauan.

Untuk melihat hasil rinci dari pekerjaan pemantauan

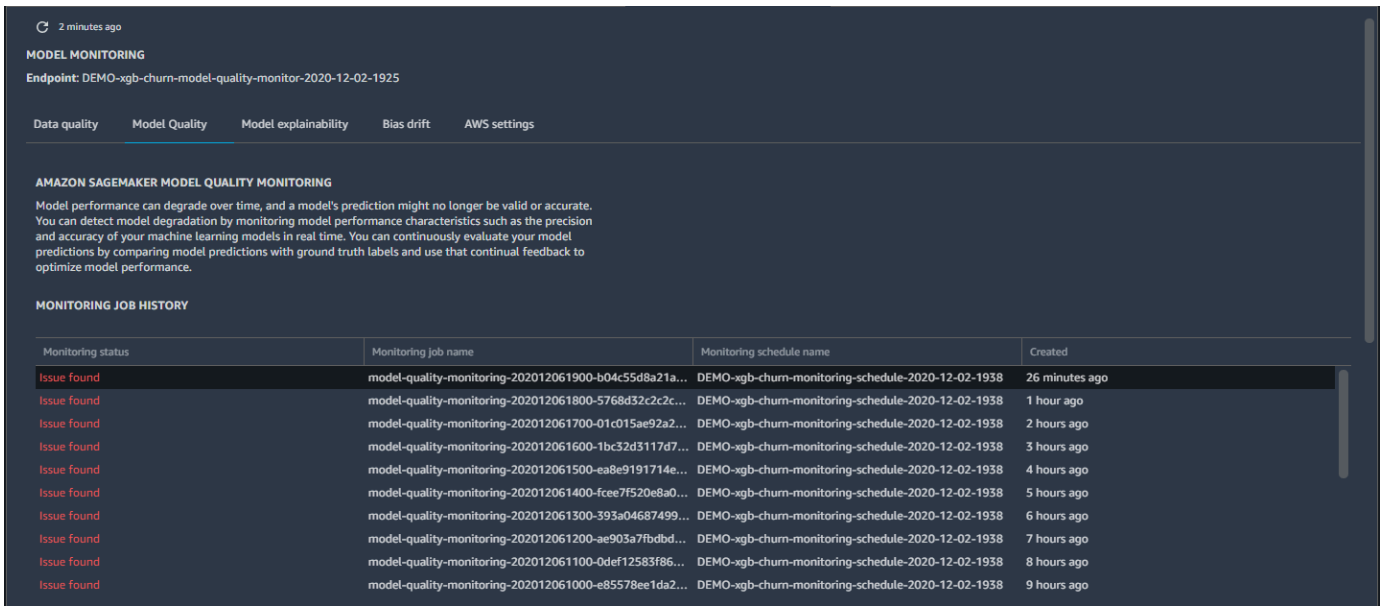
1. Masuk ke Studio. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di panel navigasi kiri, pilih Components and registries icon ().

3. Pilih Titik Akhir di menu tarik-turun.


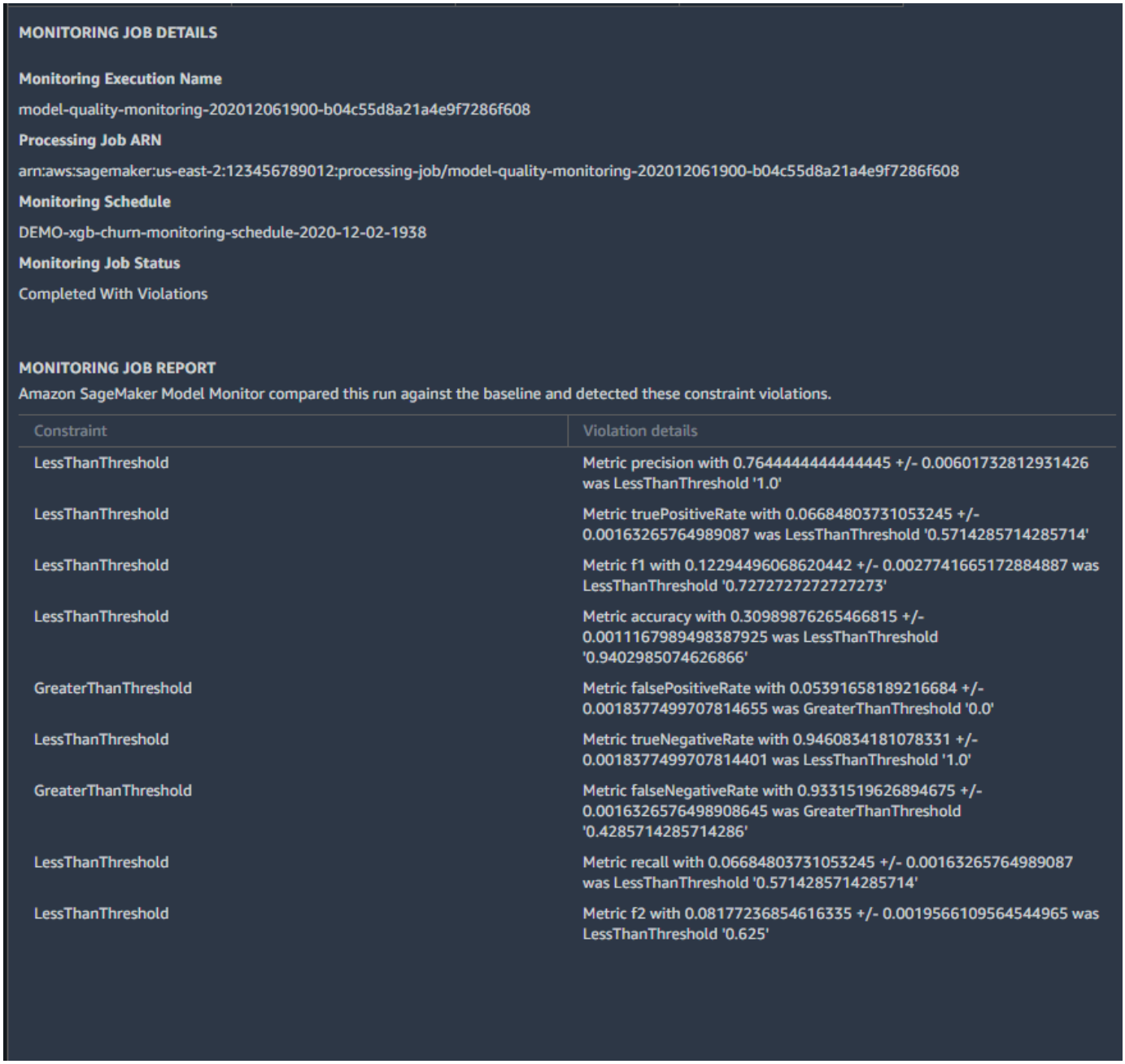
4. Pada tab titik akhir, pilih jenis pemantauan yang ingin Anda lihat detail pekerjaannya.



5. Pilih nama pekerjaan pemantauan yang ingin Anda lihat detailnya dari daftar pekerjaan pemantauan.



6. Tab MONITORING JOB DETAILS terbuka dengan laporan rinci tentang pekerjaan pemantauan.



MONITORING JOB DETAILS

Monitoring Execution Name

model-quality-monitoring-202012061900-b04c55d8a21a4e9f7286f608

Processing Job ARN

arn:aws:sagemaker:us-east-2:123456789012:processing-job/model-quality-monitoring-202012061900-b04c55d8a21a4e9f7286f608

Monitoring Schedule

DEMO-xgb-churn-monitoring-schedule-2020-12-02-1938

Monitoring Job Status

Completed With Violations

MONITORING JOB REPORT

Amazon SageMaker Model Monitor compared this run against the baseline and detected these constraint violations.

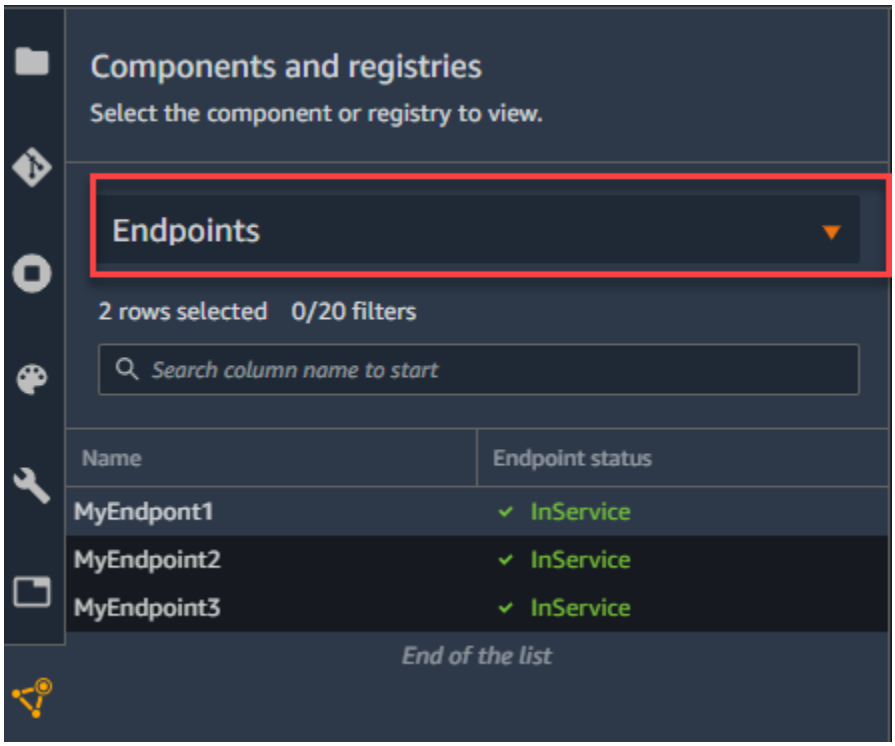
Constraint	Violation details
LessThanThreshold	Metric precision with 0.7644444444444445 +/- 0.00601732812931426 was LessThanThreshold '1.0'
LessThanThreshold	Metric truePositiveRate with 0.06684803731053245 +/- 0.00163265764989087 was LessThanThreshold '0.5714285714285714'
LessThanThreshold	Metric f1 with 0.12294496068620442 +/- 0.0027741665172884887 was LessThanThreshold '0.7272727272727273'
LessThanThreshold	Metric accuracy with 0.30989876265466815 +/- 0.0011167989498387925 was LessThanThreshold '0.9402985074626866'
GreaterThanThreshold	Metric falsePositiveRate with 0.05391658189216684 +/- 0.0018377499707814655 was GreaterThanThreshold '0.0'
LessThanThreshold	Metric trueNegativeRate with 0.9460834181078331 +/- 0.0018377499707814401 was LessThanThreshold '1.0'
GreaterThanThreshold	Metric falseNegativeRate with 0.9331519626894675 +/- 0.0016326576498908645 was GreaterThanThreshold '0.4285714285714286'
LessThanThreshold	Metric recall with 0.06684803731053245 +/- 0.00163265764989087 was LessThanThreshold '0.5714285714285714'
LessThanThreshold	Metric f2 with 0.08177236854616335 +/- 0.0019566109564544965 was LessThanThreshold '0.625'

Anda dapat membuat bagan yang menampilkan baseline dan metrik yang diambil untuk jangka waktu tertentu.

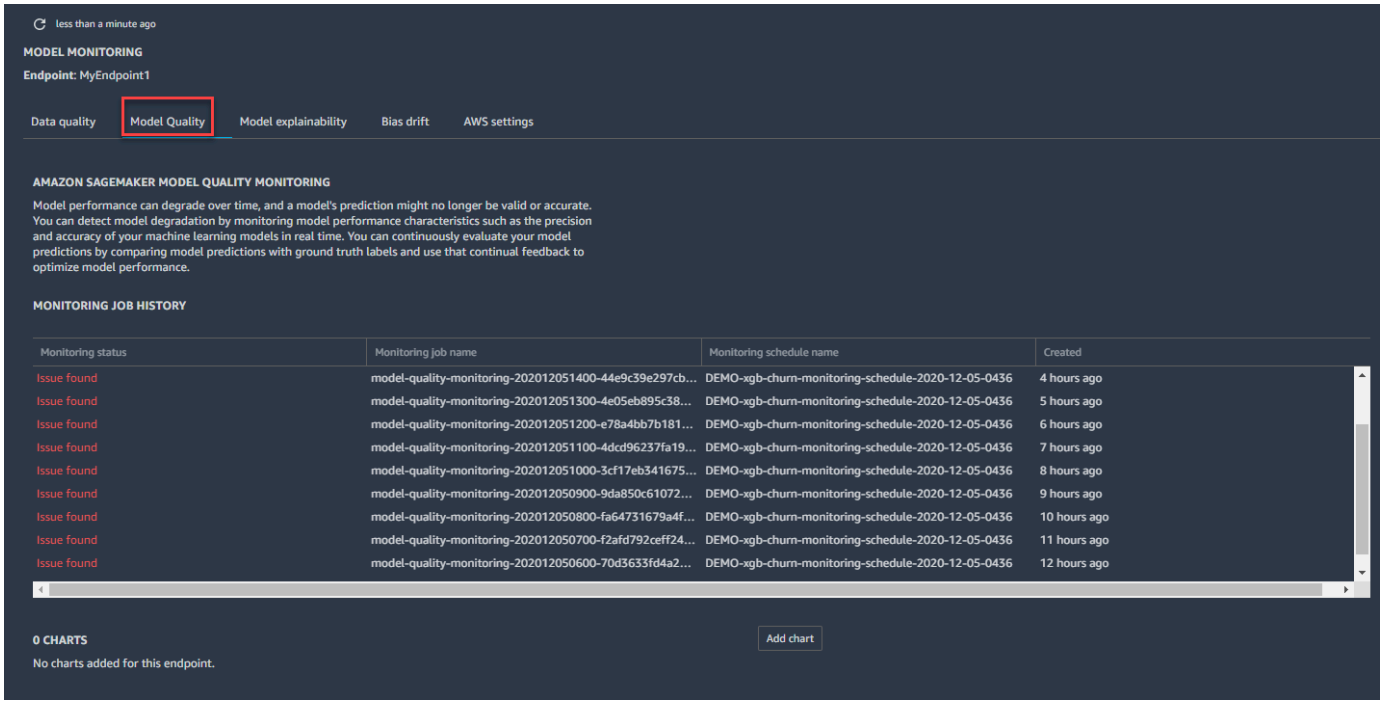
Untuk membuat bagan di SageMaker Studio untuk memvisualisasikan hasil pemantauan

1. Masuk ke Studio. Untuk informasi selengkapnya, lihat Ikhtisar SageMaker Domain Amazon.
2. Di panel navigasi kiri, pilih Components and registries icon ().

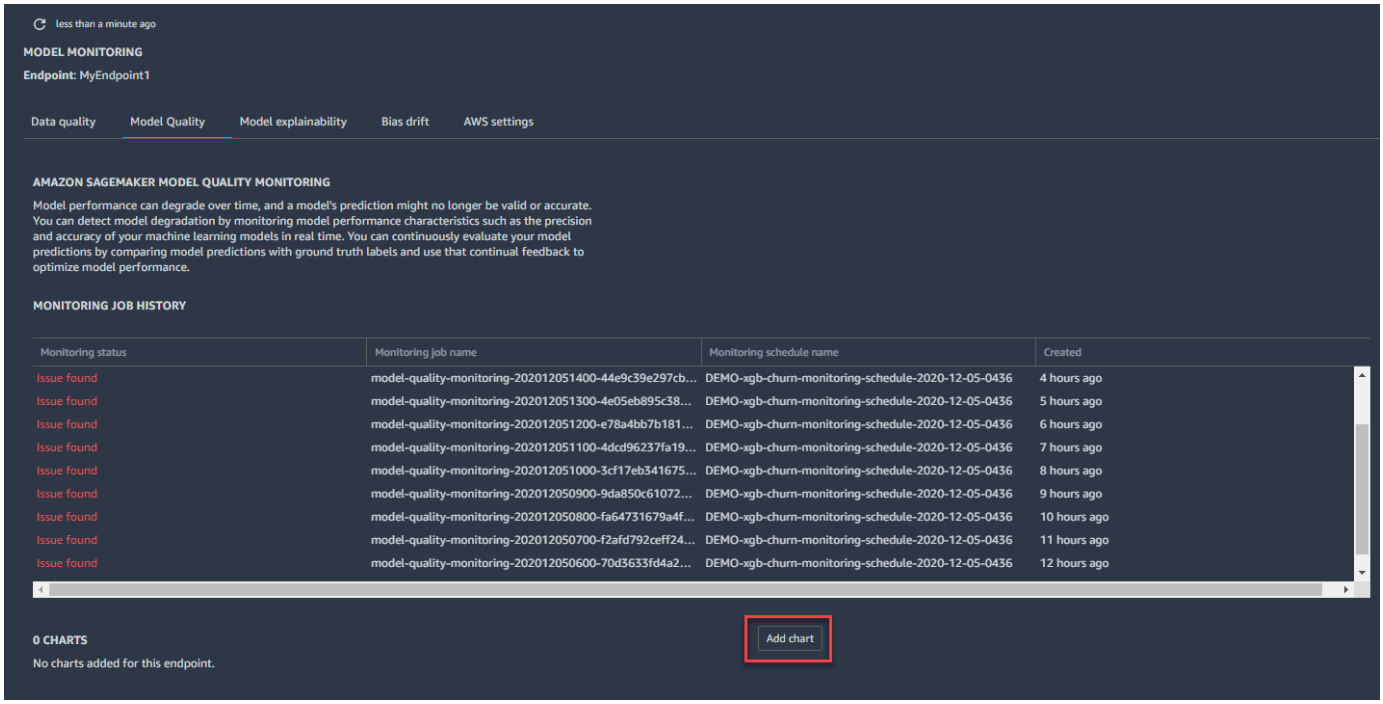
3. Pilih Titik Akhir di menu tarik-turun.



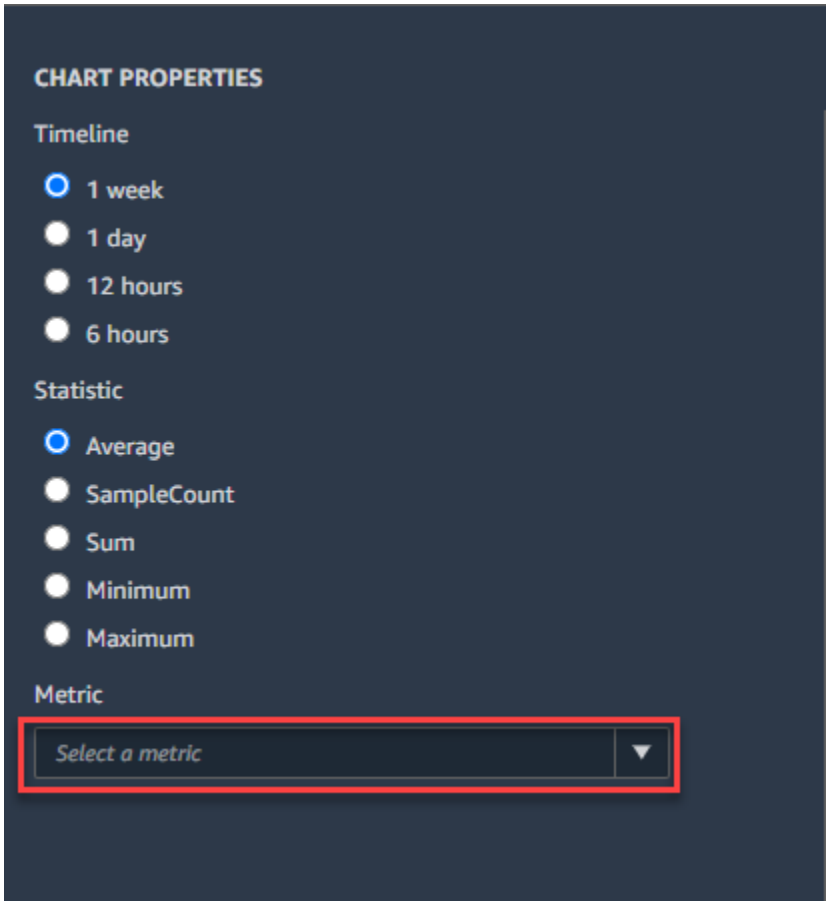
4. Pada tab Endpoint, pilih jenis pemantauan yang ingin Anda buat bagan. Contoh ini menunjukkan bagan untuk jenis pemantauan kualitas Model.



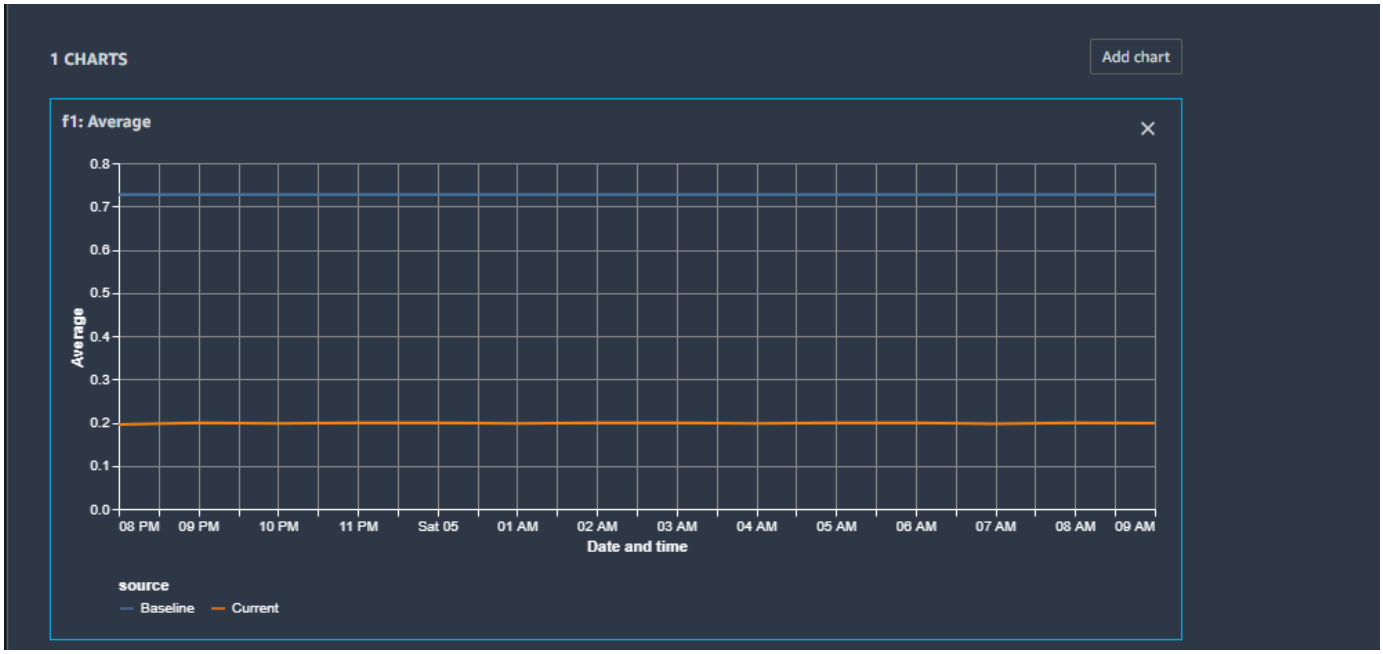
5. Pilih Tambahkan bagan.



6. Pada tab CHART PROPERTIES, pilih periode waktu, statistik, dan metrik yang ingin Anda bagan. Contoh ini menunjukkan bagan untuk Garis Waktu 1 minggu, Statistik Rata-rata, dan Metrik F1.



7. Bagan yang menunjukkan statistik metrik dasar dan saat ini yang Anda pilih pada langkah sebelumnya muncul di tab Titik Akhir.



Topik lanjutan

Bagian berikut berisi tugas-tugas lanjutan yang menjelaskan cara menyesuaikan pemantauan menggunakan skrip preprocessing dan postprocessing, cara membuat container Anda sendiri, dan cara menggunakannya AWS CloudFormation untuk membuat jadwal pemantauan.

Topik

- Kustomisasi pemantauan
- Buat Jadwal Pemantauan untuk Titik Akhir Real-time dengan Sumber Daya AWS CloudFormation Kustom

Kustomisasi pemantauan

Selain menggunakan mekanisme pemantauan bawaan, Anda dapat membuat jadwal dan prosedur pemantauan kustom Anda sendiri menggunakan skrip pra-pemrosesan dan pasca-pemrosesan atau dengan menggunakan atau membangun wadah Anda sendiri.

Topik

- Pracemrosesan dan Pascapemrosesan
- Bawa Kontainer Anda Sendiri

Pracemrosesan dan Pascapemrosesan

Anda dapat menggunakan skrip Python preprocessing dan postprocessing khusus untuk mengubah input ke monitor model Anda atau memperluas kode setelah pemantauan berhasil dijalankan. Unggah skrip ini ke Amazon S3 dan rujuk saat membuat monitor model Anda.

Contoh berikut menunjukkan cara Anda dapat menyesuaikan jadwal pemantauan dengan skrip preprocessing dan postprocessing. Ganti *`teks placeholder pengguna`* dengan informasi Anda sendiri.

```
import boto3, os
from sagemaker import get_execution_role, Session
from sagemaker.model_monitor import CronExpressionGenerator, DefaultModelMonitor

# Upload pre and postprocessor scripts
session = Session()
bucket = boto3.Session().resource("s3").Bucket(session.default_bucket())
```

```
prefix = "demo-sagemaker-model-monitor"
pre_processor_script = bucket.Object(os.path.join(prefix,
 "preprocessor.py")).upload_file("preprocessor.py")
post_processor_script = bucket.Object(os.path.join(prefix,
 "postprocessor.py")).upload_file("postprocessor.py")

# Get execution role
role = get_execution_role() # can be an empty string

# Instance type
instance_type = "instance-type"
# instance_type = "ml.m5.xlarge" # Example

# Create a monitoring schedule with pre and postprocessing
my_default_monitor = DefaultModelMonitor(
    role=role,
    instance_count=1,
    instance_type=instance_type,
    volume_size_in_gb=20,
    max_runtime_in_seconds=3600,
)

s3_report_path = "s3://{}/{}".format(bucket, "reports")
monitor_schedule_name = "monitor-schedule-name"
endpoint_name = "endpoint-name"
my_default_monitor.create_monitoring_schedule(
    post_analytics_processor_script=post_processor_script,
    record_preprocessor_script=pre_processor_script,
    monitor_schedule_name=monitor_schedule_name,
    # use endpoint_input for real-time endpoint
    endpoint_input=endpoint_name,
    # or use batch_transform_input for batch transform jobs
    # batch_transform_input=batch_transform_name,
    output_s3_uri=s3_report_path,
    statistics=my_default_monitor.baseline_statistics(),
    constraints=my_default_monitor.suggested_constraints(),
    schedule_cron_expression=CronExpressionGenerator.hourly(),
    enable_cloudwatch_metrics=True,
)
```

Topik

- Skrip Pramrosesan

- Contoh sumber daya
- Skrip Pascaprosesan

Skrip Pramrosesan

Gunakan skrip preprocessing saat Anda perlu mengubah input ke monitor model Anda.

Sebagai contoh, misalkan output dari model Anda adalah array`[1.0, 2.1]`. Wadah Amazon SageMaker Model Monitor hanya berfungsi dengan struktur JSON tabular atau pipih, seperti. `{“prediction0”: 1.0, “prediction1” : 2.1}` Anda dapat menggunakan skrip preprocessing seperti berikut ini untuk mengubah array menjadi struktur JSON yang benar.

```
def preprocess_handler(inference_record):
    input_data = inference_record.endpoint_input.data
    output_data = inference_record.endpoint_output.data.rstrip("\n")
    data = output_data + "," + input_data
    return { str(i).zfill(20) : d for i, d in enumerate(data.split(",")) }
```

Dalam contoh lain, misalkan model Anda memiliki fitur opsional dan Anda gunakan `-1` untuk menunjukkan bahwa fitur opsional memiliki nilai yang hilang. Jika Anda memiliki monitor kualitas data, Anda mungkin ingin menghapus `-1` dari array nilai input sehingga tidak termasuk dalam perhitungan metrik monitor. Anda dapat menggunakan skrip seperti berikut ini untuk menghapus nilai-nilai tersebut.

```
def preprocess_handler(inference_record):
    input_data = inference_record.endpoint_input.data
    return {i : None if x == -1 else x for i, x in enumerate(input_data.split(","))}
```

Script preprocessing Anda menerima `inference_record` sebagai satu-satunya masukan. Potongan kode berikut menunjukkan contoh. `inference_record`

```
{
  "captureData": {
    "endpointInput": {
      "observedContentType": "text/csv",
      "mode": "INPUT",
      "data": "132,25,113.2,96,269.9,107,,0,0,0,0,0,0,1,0,1,0,0,1",
```

```
      "encoding": "CSV"
    },
    "endpointOutput": {
      "observedContentType": "text/csv; charset=utf-8",
      "mode": "OUTPUT",
      "data": "0.01076381653547287",
      "encoding": "CSV"
    }
  },
  "eventMetadata": {
    "eventId": "feca1ab1-8025-47e3-8f6a-99e3fdd7b8d9",
    "inferenceTime": "2019-11-20T23:33:12Z"
  },
  "eventVersion": "0"
}
```

Cuplikan kode berikut menunjukkan struktur kelas penuh untuk sebuah. `inference_record`

```
KEY_EVENT_METADATA = "eventMetadata"
KEY_EVENT_METADATA_EVENT_ID = "eventId"
KEY_EVENT_METADATA_EVENT_TIME = "inferenceTime"
KEY_EVENT_METADATA_CUSTOM_ATTR = "customAttributes"

KEY_EVENTDATA_ENCODING = "encoding"
KEY_EVENTDATA_DATA = "data"

KEY_GROUND_TRUTH_DATA = "groundTruthData"

KEY_EVENTDATA = "captureData"
KEY_EVENTDATA_ENDPOINT_INPUT = "endpointInput"
KEY_EVENTDATA_ENDPOINT_OUTPUT = "endpointOutput"

KEY_EVENTDATA_BATCH_OUTPUT = "batchTransformOutput"
KEY_EVENTDATA_OBSERVED_CONTENT_TYPE = "observedContentType"
KEY_EVENTDATA_MODE = "mode"

KEY_EVENT_VERSION = "eventVersion"

class EventConfig:
    def __init__(self, endpoint, variant, start_time, end_time):
        self.endpoint = endpoint
        self.variant = variant
```

```
        self.start_time = start_time
        self.end_time = end_time


class EventMetadata:
    def __init__(self, event_metadata_dict):
        self.event_id = event_metadata_dict.get(KEY_EVENT_METADATA_EVENT_ID, None)
        self.event_time = event_metadata_dict.get(KEY_EVENT_METADATA_EVENT_TIME, None)
        self.custom_attribute = event_metadata_dict.get(KEY_EVENT_METADATA_CUSTOM_ATTR,
 None)


class EventData:
    def __init__(self, data_dict):
        self.encoding = data_dict.get(KEY_EVENTDATA_ENCODING, None)
        self.data = data_dict.get(KEY_EVENTDATA_DATA, None)
        self.observedContentType = data_dict.get(KEY_EVENTDATA_OBSERVED_CONTENT_TYPE,
 None)
        self.mode = data_dict.get(KEY_EVENTDATA_MODE, None)

    def as_dict(self):
        ret = {
            KEY_EVENTDATA_ENCODING: self.encoding,
            KEY_EVENTDATA_DATA: self.data,
            KEY_EVENTDATA_OBSERVED_CONTENT_TYPE: self.observedContentType,
        }
        return ret


class CapturedData:
    def __init__(self, event_dict):
        self.event_metadata = None
        self.endpoint_input = None
        self.endpoint_output = None
        self.batch_transform_output = None
        self.ground_truth = None
        self.event_version = None
        self.event_dict = event_dict
        self._event_dict_postprocessed = False

        if KEY_EVENT_METADATA in event_dict:
            self.event_metadata = EventMetadata(event_dict[KEY_EVENT_METADATA])
        if KEY_EVENTDATA in event_dict:
            if KEY_EVENTDATA_ENDPOINT_INPUT in event_dict[KEY_EVENTDATA]:
```

```
                self.endpoint_input = EventData(event_dict[KEY_EVENTDATA]
[KEY_EVENTDATA_ENDPOINT_INPUT])
            if KEY_EVENTDATA_ENDPOINT_OUTPUT in event_dict[KEY_EVENTDATA]:
                self.endpoint_output = EventData(event_dict[KEY_EVENTDATA]
[KEY_EVENTDATA_ENDPOINT_OUTPUT])
            if KEY_EVENTDATA_BATCH_OUTPUT in event_dict[KEY_EVENTDATA]:
                self.batch_transform_output = EventData(event_dict[KEY_EVENTDATA]
[KEY_EVENTDATA_BATCH_OUTPUT])

        if KEY_GROUND_TRUTH_DATA in event_dict:
            self.ground_truth = EventData(event_dict[KEY_GROUND_TRUTH_DATA])
        if KEY_EVENT_VERSION in event_dict:
            self.event_version = event_dict[KEY_EVENT_VERSION]

    def as_dict(self):
        if self._event_dict_postprocessed is True:
            return self.event_dict
        if KEY_EVENTDATA in self.event_dict:
            if KEY_EVENTDATA_ENDPOINT_INPUT in self.event_dict[KEY_EVENTDATA]:
                self.event_dict[KEY_EVENTDATA][KEY_EVENTDATA_ENDPOINT_INPUT] =
 self.endpoint_input.as_dict()
            if KEY_EVENTDATA_ENDPOINT_OUTPUT in self.event_dict[KEY_EVENTDATA]:
                self.event_dict[KEY_EVENTDATA][
                    KEY_EVENTDATA_ENDPOINT_OUTPUT
                ] = self.endpoint_output.as_dict()
            if KEY_EVENTDATA_BATCH_OUTPUT in self.event_dict[KEY_EVENTDATA]:
                self.event_dict[KEY_EVENTDATA][KEY_EVENTDATA_BATCH_OUTPUT] =
 self.batch_transform_output.as_dict()

        self._event_dict_postprocessed = True
        return self.event_dict

    def __str__(self):
        return str(self.as_dict())
```

Contoh sumber daya

Anda juga dapat menerapkan strategi pengambilan sampel khusus dalam skrip preprocessing Anda. Untuk melakukan ini, konfigurasikan wadah pra-bangun pihak pertama Model Monitor untuk mengabaikan persentase catatan sesuai dengan laju pengambilan sampel yang Anda tentukan. Dalam contoh berikut, handler mengambil sampel 10 persen dari catatan dengan mengembalikan catatan dalam 10 persen panggilan handler dan daftar kosong sebaliknya.

```
import random

def preprocess_handler(inference_record):
    # we set up a sampling rate of 0.1
    if random.random() > 0.1:
        # return an empty list
        return []
    input_data = inference_record.endpoint_input.data
    return {i : None if x == -1 else x for i, x in enumerate(input_data.split(","))}
```

Logging kustom untuk skrip preprocessing

Jika skrip preprocessing Anda mengembalikan kesalahan, periksa pesan pengecualian yang masuk CloudWatch ke debug. Anda dapat mengakses logger CloudWatch melalui `preprocess_handler` antarmuka. Anda dapat mencatat informasi apa pun yang Anda butuhkan dari skrip Anda CloudWatch. Ini dapat berguna saat men-debug skrip preprocessing Anda. Contoh berikut menunjukkan bagaimana Anda dapat menggunakan `preprocess_handler` antarmuka untuk log ke CloudWatch

```
def preprocess_handler(inference_record, logger):
    logger.info(f"I'm a processing record: {inference_record}")
    logger.debug(f"I'm debugging a processing record: {inference_record}")
    logger.warning(f"I'm processing record with missing value: {inference_record}")
    logger.error(f"I'm a processing record with bad value: {inference_record}")
    return inference_record
```

Skrip Pascaprosesan

Gunakan skrip postprocessing saat Anda ingin memperpanjang kode setelah pemantauan yang berhasil dijalankan.

```
def postprocess_handler():
    print("Hello from post-proc script!")
```

Bawa Kontainer Anda Sendiri

Amazon SageMaker Model Monitor menyediakan wadah bawaan dengan kemampuan untuk menganalisis data yang diambil dari titik akhir atau pekerjaan transformasi batch untuk kumpulan

data tabular. Jika Anda ingin membawa wadah Anda sendiri, Model Monitor menyediakan titik ekstensi yang dapat Anda manfaatkan.

Di bawah tenda, saat Anda membuat`MonitoringSchedule`, Model Monitor akhirnya memulai pekerjaan pemrosesan. Oleh karena itu wadah perlu mengetahui kontrak kerja pemrosesan yang didokumentasikan dalam Bangun Kontainer Pemrosesan Anda Sendiri (Skenario Lanjutan) topik tersebut. Perhatikan bahwa Model Monitor memulai pekerjaan pemrosesan atas nama Anda sesuai jadwal. Saat menjalankan, Model Monitor menyiapkan variabel lingkungan tambahan untuk Anda sehingga penampung Anda memiliki konteks yang cukup untuk memproses data untuk eksekusi tertentu dari pemantauan terjadwal. Untuk informasi tambahan tentang input kontainer, lihat. Input Kontrak Kontainer

Dalam wadah, menggunakan variabel/konteks lingkungan di atas, Anda sekarang dapat menganalisis kumpulan data untuk periode saat ini dalam kode kustom Anda. Setelah analisis ini selesai, Anda dapat memilih untuk mengeluarkan laporan Anda untuk diunggah ke bucket S3. Laporan yang dihasilkan oleh kontainer bawaan didokumentasikanOutput Kontrak Kontainer. Jika Anda ingin visualisasi laporan berfungsi di SageMaker Studio, Anda harus mengikuti format yang sama. Anda juga dapat memilih untuk mengeluarkan laporan khusus.

Anda juga memancarkan CloudWatch metrik dari wadah dengan mengikuti petunjuk di. CloudWatch Metrik untuk Membawa Kontainer Anda Sendiri

Topik

- Input Kontrak Kontainer
- Output Kontrak Kontainer
- CloudWatch Metrik untuk Membawa Kontainer Anda Sendiri

Input Kontrak Kontainer

Platform Amazon SageMaker Model Monitor memanggil kode penampung Anda sesuai dengan jadwal yang ditentukan. Jika Anda memilih untuk menulis kode kontainer Anda sendiri, variabel lingkungan berikut tersedia. Dalam konteks ini, Anda dapat menganalisis kumpulan data saat ini atau mengevaluasi kendala jika Anda memilih dan memancarkan metrik, jika berlaku.

Variabel lingkungan yang tersedia sama untuk titik akhir real-time dan pekerjaan transformasi batch, kecuali untuk `dataset_format` variabel. Jika Anda menggunakan titik akhir real-time, `dataset_format` variabel mendukung opsi berikut:

```
{\"sagemakerCaptureJson\": {\"captureIndexNames\": [\"endpointInput\",\"endpointOutput
\"]}}
```

Jika Anda menggunakan pekerjaan transformasi batch, `dataset_format` mendukung opsi berikut:

```
{\"csv\": {\"header\": [\"true\",\"false\"]}}
```

```
{\"json\": {\"line\": [\"true\",\"false\"]}}
```

```
{\"parquet\": {}}
```

Contoh kode berikut menunjukkan set lengkap variabel lingkungan yang tersedia untuk kode kontainer Anda (dan menggunakan `dataset_format` format untuk titik akhir waktu nyata).

```
"Environment": {
 "dataset_format": "{\"sagemakerCaptureJson\": {\"captureIndexNames\": [\"endpointInput
\",\"endpointOutput\"]}}",
 "dataset_source": "/opt/ml/processing/endpointdata",
 "end_time": "2019-12-01T16: 20: 00Z",
 "output_path": "/opt/ml/processing/resultdata",
 "publish_cloudwatch_metrics": "Disabled",
 "sagemaker_endpoint_name": "endpoint-name",
 "sagemaker_monitoring_schedule_name": "schedule-name",
 "start_time": "2019-12-01T15: 20: 00Z"
}
```

Parameter

Nama Parameter	Deskripsi
`dataset_format`	Untuk pekerjaan yang dimulai dari yang `MonitoringSchedule` didukung oleh`Endpoint`, ini `sageMakerCaptureJson` dengan indeks penangkapan`endpointInput` , atau`endpointOutput` , atau keduanya. Untuk pekerjaan transformasi batch, ini menentukan format data, apakah CSV, JSON, atau Parquet.

Nama Parameter	Deskripsi
`dataset_source`	Jika Anda menggunakan titik akhir real-time, jalur lokal di mana data yang sesuai dengan periode pemantauan, sebagaimana ditentukan oleh `start_time` dan `end_time`, tersedia. Pada jalur ini, data tersedia di `/{endpoint-name}/{variant-name}/yyyy/mm/dd/hh` . Kami terkadang mengunduh lebih dari yang ditentukan oleh waktu mulai dan akhir. Terserah kode kontainer untuk mengurai data sesuai kebutuhan.
`output_path`	Jalur lokal untuk menulis laporan output dan file lainnya. Anda menentukan parameter ini dalam `CreateMonitoringSchedule` permintaan sebagai`MonitoringOutputConfig.MonitoringOutput[0].LocalPath` . Itu diunggah ke `S3Uri` jalur yang ditentukan dalam`MonitoringOutputConfig.MonitoringOutput[0].S3Uri` .
`publish_cloudwatch_metrics`	Untuk pekerjaan yang diluncurkan oleh`CreateMonitoringSchedule` , parameter ini diatur ke`Enabled`. Wadah dapat memilih untuk menulis file CloudWatch output Amazon di`[filepath]` .
`sagemaker_endpoint_name`	Jika Anda menggunakan titik akhir real-time, nama tempat `Endpoint` pekerjaan terjadwal ini diluncurkan.
`sagemaker_monitoring_schedule_name`	Nama `MonitoringSchedule` yang meluncurkan pekerjaan ini.

Nama Parameter	Deskripsi
`*sagemaker_endpoint_datacapture_prefix*`	Jika Anda menggunakan titik akhir real-time , awalan yang ditentukan dalam `DataCaptureConfig` parameter. `Endpoint` Wadah dapat menggunakan ini jika perlu mengakses lebih banyak data secara langsung daripada yang sudah diunduh SageMaker di `dataset_source` jalur.
`start_time, end_time`	Jendela waktu untuk analisis ini berjalan. Misalnya, untuk pekerjaan yang dijadwalkan berjalan pada 05:00 UTC dan pekerjaan yang berjalan pada 20/02/2020,`start_time` : adalah 2020-02-19T 06:00:00 Z dan: adalah 2020-02-20T 05:00:00 Z `end_time`
`baseline_constraints:`	Jalur lokal dari file kendala dasar yang ditentukan dalam. `BaselineConfig.ConstraintResource.S3Uri` Ini hanya tersedia jika parameter ini ditentukan dalam `CreateMonitoringSchedule` permintaan.
`baseline_statistics`	Jalur lokal ke file statistik dasar yang ditentukan dalam. `BaselineConfig.StatisticsResource.S3Uri` Ini hanya tersedia jika parameter ini ditentukan dalam `CreateMonitoringSchedule` permintaan. :

Output Kontrak Kontainer

Wadah dapat menganalisis data yang tersedia di `*dataset_source*` jalur dan menulis laporan ke jalur di `*output_path*.` Kode kontainer dapat menulis laporan apa pun yang sesuai dengan kebutuhan Anda.

Jika Anda menggunakan struktur dan kontrak berikut, file keluaran tertentu diperlakukan secara khusus oleh SageMaker visualisasi dan API. Ini hanya berlaku untuk kumpulan data tabel.

File Output untuk Dataset Tabular

Format nama file	Deskripsi
`statistics.json`	File ini diharapkan memiliki statistik kolumnar untuk setiap fitur dalam kumpulan data yang dianalisis. Skema untuk file ini tersedia di bagian selanjutnya.
`constraints.json`	File ini diharapkan memiliki kendala pada fitur yang diamati. Skema untuk file ini tersedia di bagian selanjutnya.
`constraints_violations.json`	File ini diharapkan memiliki daftar pelanggar an yang ditemukan dalam kumpulan data saat ini dibandingkan dengan statistik dasar dan file kendala yang ditentukan di jalur dan. `baseline_constaints` `baseline_ statistics`

Selain itu, jika `publish_cloudwatch_metrics` nilainya adalah kode `"Enabled"` kontainer dapat memancarkan CloudWatch metrik Amazon di lokasi ini:. `/opt/ml/output/metrics/cloudwatch` Skema untuk file-file ini dijelaskan di bagian berikut.

Topik

- Skema untuk Statistik (file statistik.json)
- Skema untuk Kendala (file kendala json)

Skema untuk Statistik (file statistik.json)

Skema yang didefinisikan dalam `statistics.json` file menentukan parameter statistik yang akan dihitung untuk baseline dan data yang ditangkap. Ini juga mengonfigurasi bucket yang akan digunakan oleh KLL, sketsa kuantil yang sangat ringkas dengan skema pemadatan malas.

```
{
    "version": 0,
    # dataset level stats
    "dataset": {
```

```
        "item_count": number
    },
    # feature level stats
    "features": [
        {
            "name": "feature-name",
            "inferred_type": "Fractional" | "Integral",
            "numerical_statistics": {
                "common": {
                    "num_present": number,
                    "num_missing": number
                },
                "mean": number,
                "sum": number,
                "std_dev": number,
                "min": number,
                "max": number,
                "distribution": {
                    "kll": {
                        "buckets": [
                            {
                                "lower_bound": number,
                                "upper_bound": number,
                                "count": number
                            }
                        ],
                        "sketch": {
                            "parameters": {
                                "c": number,
                                "k": number
                            },
                            "data": [
                                [
                                    num,
                                    num,
                                    num,
                                    num
                                ],
                                [
                                    num,
                                    num
                                ][
                                    num,
                                    num
```

```
                                ]
                            ]
                        }#sketch
                    }#KLL
                }#distribution
            }#num_stats
        },
        {
            "name": "feature-name",
            "inferred_type": "String",
            "string_statistics": {
                "common": {
                    "num_present": number,
                    "num_missing": number
                },
                "distinct_count": number,
                "distribution": {
                    "categorical": {
                         "buckets": [
                                {
                                    "value": "string",
                                    "count": number
                                }
                             ]
                         }
                }
            },
            #provision for custom stats
        }
    ]
}
```

> **Catatan**
>
> - Metrik yang ditentukan dikenali oleh SageMaker perubahan visualisasi selanjutnya. Wadah dapat memancarkan lebih banyak metrik jika diperlukan.
> - Sketsa KLL adalah sketsa yang diakui. Wadah khusus dapat menulis representasi mereka sendiri, tetapi tidak akan dikenali oleh SageMaker visualisasi.
> - Secara default, distribusi diwujudkan dalam 10 ember. Anda tidak dapat mengubah ini.

Skema untuk Kendala (file kendala json)

File constraints.json digunakan untuk mengekspresikan kendala yang harus dipenuhi oleh dataset. Container Amazon SageMaker Model Monitor dapat menggunakan file constraints.json untuk mengevaluasi kumpulan data. Kontainer bawaan menyediakan kemampuan untuk menghasilkan file constraints.json secara otomatis untuk dataset dasar. Jika Anda membawa wadah Anda sendiri, Anda dapat menyediakannya dengan kemampuan serupa atau Anda dapat membuat file constraints.json dengan cara lain. Berikut adalah skema untuk file kendala yang digunakan kontainer bawaan. Bawa wadah Anda sendiri dapat mengadopsi format yang sama atau meningkatkannya sesuai kebutuhan.

```
{
    "version": 0,
    "features":
    [
        {
            "name": "string",
            "inferred_type": "Integral" | "Fractional" |
                    | "String" | "Unknown",
            "completeness": number,
            "num_constraints":
            {
                "is_non_negative": boolean
            },
            "string_constraints":
            {
                "domains":
                [
                    "list of",
                    "observed values",
                    "for small cardinality"
                ]
            },
            "monitoringConfigOverrides":
            {}
        }
    ],
    "monitoring_config":
    {
        "evaluate_constraints": "Enabled",
        "emit_metrics": "Enabled",
        "datatype_check_threshold": 0.1,
```

```
        "domain_content_threshold": 0.1,
        "distribution_constraints":
        {
            "perform_comparison": "Enabled",
            "comparison_threshold": 0.1,
            "comparison_method": "Simple"||"Robust",
            "categorical_comparison_threshold": 0.1,
            "categorical_drift_method": "LInfinity"||"ChiSquared"
        }
    }
}
```

`monitoring_config`Objek berisi opsi untuk pekerjaan pemantauan untuk fitur tersebut. Tabel berikut menjelaskan setiap opsi.

Batasan pemantauan

Kendala	Deskripsi
`evaluate_constraints`	Kapan`Enabled`, mengevaluasi apakah kumpulan data saat ini yang dianalisis memenuhi batasan yang ditentukan dalam file constraints.json yang diambil sebagai baseline. Nilai yang valid: `Enabled` atau `Disabled` Default: `Enabled`
`emit_metrics`	Kapan`Enabled`, memancarkan CloudWatch metrik untuk data yang terkandung dalam file. Nilai yang valid: `Enabled` atau `Disabled` Default: `Enabled`
`datatype_check_threshold`	Jika ambang batas di atas nilai yang ditentuka n`datatype_check_threshold` , ini menyebabkan kegagalan yang diperlakukan sebagai pelanggaran dalam laporan pelanggar an. Jika tipe data dalam eksekusi saat ini tidak sama dengan pada dataset dasar, ambang

Kendala	Deskripsi
	batas ini digunakan untuk mengevaluasi apakah perlu ditandai sebagai pelanggaran. Selama langkah dasar, kendala yang dihasilkan menyarankan tipe data yang disimpulkan untuk setiap kolom. `datatype_check_threshold` Parameter dapat disetel untuk menyesuaikan ambang batas saat ditandai sebagai pelanggaran. Nilai yang benar: float 0,1 per detik
`domain_content_threshold`	Jika ada lebih banyak nilai yang tidak diketahui untuk bidang String di kumpulan data saat ini daripada di kumpulan data dasar, ambang batas ini dapat digunakan untuk menentukan apakah perlu ditandai sebagai pelanggaran. Nilai yang benar: float 0,1 per detik
`distribution_constraints`	`perform_comparison` Kapan`Enabled`, bendera ini menginstruksikan kode untuk melakukan perbandingan distribusi antara distribusi dasar dan distribusi yang diamati untuk kumpulan data saat ini. Nilai yang valid: `Enabled` atau `Disabled` Default: `Enabled`

Kendala	Deskripsi
	`comparison_threshold` Jika ambang batas di atas nilai yang ditetapkan untuk`comparison_threshold` , ini menyebabkan kegagalan yang diperlakukan sebagai pelanggaran dalam laporan pelanggaran. Jarak dihitung dengan mendapatkan perbedaan absolut maksimum antara fungsi distribusi kumulatif dari dua distribusi. Nilai yang benar: float 0,1 per detik
	`comparison_method` Apakah akan menghitung `linf_simple` atau`linf_robust` . `linf_simple` Ini didasarkan pada perbedaan absolut maksimum antara fungsi distribusi kumulatif dari dua distribusi. Menghitung `linf_robust` didasarkan pada`linf_simple` , tetapi digunakan ketika tidak ada cukup sampel. `linf_robust` Rumusnya didasarkan pada uji Kolmogorov-Smirnov dua sampel. Nilai yang valid: `linf_simple` atau `linf_robust`

Kendala	Deskripsi
	`categorical_comparison_threshold` Opsional. Menetapkan ambang batas untuk fitur kategoris. Jika nilai dalam kumpulan data melebihi ambang batas yang Anda tetapkan, pelanggaran dicatat dalam laporan pelanggaran. Nilai yang benar: float Default: Nilai yang ditetapkan untuk `comparison_threshold` parameter
	`categorical_drift_method` Opsional. Untuk fitur kategoris, tentukan metode komputasi yang digunakan untuk mendeteksi penyimpangan distribusi. Jika Anda tidak mengatur parameter ini, tes K-S (LinFinity) akan digunakan. Nilai yang Valid: `LInfinity` atau `ChiSquared` Default: `LInfinity`

CloudWatch Metrik untuk Membawa Kontainer Anda Sendiri

Jika `publish_cloudwatch_metrics` nilainya ada `Enabled` di `Environment` peta dalam `/opt/ml/processing/processingjobconfig.json` file, kode penampung memancarkan CloudWatch metrik Amazon di lokasi ini:. `/opt/ml/output/metrics/cloudwatch`

Skema untuk file ini sangat didasarkan pada CloudWatch `PutMetrics` API. Namespace tidak ditentukan di sini. Ini defaultnya sebagai berikut:

- `For real-time endpoints: /aws/sagemaker/Endpoint/data-metrics`
- `For batch transform jobs: /aws/sagemaker/ModelMonitoring/data-metrics`

Namun, Anda dapat menentukan dimensi. Kami menyarankan Anda menambahkan dimensi berikut minimal:

- `Endpoint`dan `MonitoringSchedule` untuk titik akhir real-time
- `MonitoringSchedule`untuk pekerjaan transformasi batch

Cuplikan JSON berikut menunjukkan cara mengatur dimensi Anda.

Untuk titik akhir real-time, lihat cuplikan JSON berikut yang mencakup dan dimensi: `Endpoint` `MonitoringSchedule`

```
{
    "MetricName": "", # Required
    "Timestamp": "2019-11-26T03:00:00Z", # Required
    "Dimensions" : [{"Name":"Endpoint","Value":"endpoint_0"},
{"Name":"MonitoringSchedule","Value":"schedule_0"}]
    "Value": Float,
    # Either the Value or the StatisticValues field can be populated and not both.
    "StatisticValues": {
        "SampleCount": Float,
        "Sum": Float,
        "Minimum": Float,
        "Maximum": Float
    },
    "Unit": "Count", # Optional
}
```

Untuk pekerjaan transformasi batch, lihat cuplikan JSON berikut yang mencakup dimensi: `MonitoringSchedule`

```
{
    "MetricName": "", # Required
    "Timestamp": "2019-11-26T03:00:00Z", # Required
    "Dimensions" : [{"Name":"MonitoringSchedule","Value":"schedule_0"}]
    "Value": Float,
    # Either the Value or the StatisticValues field can be populated and not both.
    "StatisticValues": {
        "SampleCount": Float,
        "Sum": Float,
        "Minimum": Float,
        "Maximum": Float
```

```
    },
    "Unit": "Count", # Optional
}
```

Buat Jadwal Pemantauan untuk Titik Akhir Real-time dengan Sumber Daya AWS CloudFormation Kustom

Jika Anda menggunakan titik akhir real-time, Anda dapat menggunakan sumber daya AWS CloudFormation khusus untuk membuat jadwal pemantauan. Sumber daya kustom ada di Python. Untuk menerapkannya, lihat Penerapan Python Lambda.

Contoh sumber daya khusus

Mulailah dengan menambahkan sumber daya khusus ke AWS CloudFormation template Anda. Ini menunjuk ke AWS Lambda fungsi yang Anda buat di langkah berikutnya.

Sumber daya ini memungkinkan Anda untuk menyesuaikan parameter untuk jadwal pemantauan Anda dapat menambah atau menghapus lebih banyak parameter dengan memodifikasi AWS CloudFormation sumber daya dan fungsi Lambda dalam sumber daya contoh berikut.

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "MonitoringSchedule": {
            "Type": "Custom::MonitoringSchedule",
            "Version": "1.0",
            "Properties": {
                "ServiceToken": "arn:aws:lambda:us-west-2:111111111111:function:lambda-
name",
                "ScheduleName": "YourScheduleName",
                "EndpointName": "YourEndpointName",
                "BaselineConstraintsUri": "s3://your-baseline-constraints/
constraints.json",
                "BaselineStatisticsUri": "s3://your-baseline-stats/statistics.json",
                "PostAnalyticsProcessorSourceUri": "s3://your-post-processor/
postprocessor.py",
                "RecordPreprocessorSourceUri": "s3://your-preprocessor/
preprocessor.py",
                "InputLocalPath": "/opt/ml/processing/endpointdata",
                "OutputLocalPath": "/opt/ml/processing/localpath",
                "OutputS3URI": "s3://your-output-uri",
                "ImageURI": "111111111111.dkr.ecr.us-west-2.amazonaws.com/your-image",
```

```
                "ScheduleExpression": "cron(0 * ? * * *)",
                "PassRoleArn": "arn:aws:iam::111111111111:role/AmazonSageMaker-
ExecutionRole"
            }
        }
    }
}
```

Kode Sumber Daya Kustom Lambda

Sumber daya AWS CloudFormation kustom ini menggunakan AWS pustaka Custom Resource Helper, yang dapat Anda instal dengan pip menggunakan. `pip install crhelper`

Fungsi Lambda ini dipanggil oleh AWS CloudFormation selama pembuatan dan penghapusan tumpukan. Fungsi Lambda ini bertanggung jawab untuk membuat dan menghapus jadwal pemantauan dan menggunakan parameter yang ditentukan dalam sumber daya khusus yang dijelaskan di bagian sebelumnya.

```
import boto3
import botocore
import logging

from crhelper import CfnResource
from botocore.exceptions import ClientError


logger = logging.getLogger(__name__)
sm = boto3.client('sagemaker')

# cfnhelper makes it easier to implement a CloudFormation custom resource
helper = CfnResource()

# CFN Handlers

def handler(event, context):
    helper(event, context)


@helper.create
def create_handler(event, context):
    """
    Called when CloudFormation custom resource sends the create event
    """
```

```
    create_monitoring_schedule(event)


@helper.delete
def delete_handler(event, context):
    """
    Called when CloudFormation custom resource sends the delete event
    """
    schedule_name = get_schedule_name(event)
    delete_monitoring_schedule(schedule_name)


@helper.poll_create
def poll_create(event, context):
    """
    Return true if the resource has been created and false otherwise so
    CloudFormation polls again.
    """
    schedule_name = get_schedule_name(event)
    logger.info('Polling for creation of schedule: %s', schedule_name)
    return is_schedule_ready(schedule_name)

@helper.update
def noop():
    """
    Not currently implemented but crhelper will throw an error if it isn't added
    """
    pass

# Helper Functions

def get_schedule_name(event):
    return event['ResourceProperties']['ScheduleName']

def create_monitoring_schedule(event):
    schedule_name = get_schedule_name(event)
    monitoring_schedule_config = create_monitoring_schedule_config(event)

    logger.info('Creating monitoring schedule with name: %s', schedule_name)

    sm.create_monitoring_schedule(
        MonitoringScheduleName=schedule_name,
        MonitoringScheduleConfig=monitoring_schedule_config)
```

```
def is_schedule_ready(schedule_name):
    is_ready = False

    schedule = sm.describe_monitoring_schedule(MonitoringScheduleName=schedule_name)
    status = schedule['MonitoringScheduleStatus']

    if status == 'Scheduled':
        logger.info('Monitoring schedule (%s) is ready', schedule_name)
        is_ready = True
    elif status == 'Pending':
        logger.info('Monitoring schedule (%s) still creating, waiting and polling
 again...', schedule_name)
    else:
        raise Exception('Monitoring schedule ({}) has unexpected status:
 {}'.format(schedule_name, status))

    return is_ready

def create_monitoring_schedule_config(event):
    props = event['ResourceProperties']

    return {
        "ScheduleConfig": {
            "ScheduleExpression": props["ScheduleExpression"],
        },
        "MonitoringJobDefinition": {
            "BaselineConfig": {
                "ConstraintsResource": {
                    "S3Uri": props['BaselineConstraintsUri'],
                },
                "StatisticsResource": {
                    "S3Uri": props['BaselineStatisticsUri'],
                }
            },
            "MonitoringInputs": [
                {
                    "EndpointInput": {
                        "EndpointName": props["EndpointName"],
                        "LocalPath": props["InputLocalPath"],
                    }
                }
            ],
            "MonitoringOutputConfig": {
                "MonitoringOutputs": [
```

```
                    {
                        "S3Output": {
                            "S3Uri": props["OutputS3URI"],
                            "LocalPath": props["OutputLocalPath"],
                        }
                    }
                ],
            },
            "MonitoringResources": {
                "ClusterConfig": {
                    "InstanceCount": 1,
                    "InstanceType": "ml.t3.medium",
                    "VolumeSizeInGB": 50,
                }
            },
            "MonitoringAppSpecification": {
                "ImageUri": props["ImageURI"],
                "RecordPreprocessorSourceUri":
 props['PostAnalyticsProcessorSourceUri'],
                "PostAnalyticsProcessorSourceUri":
 props['PostAnalyticsProcessorSourceUri'],
            },
            "StoppingCondition": {
                "MaxRuntimeInSeconds": 300
            },
            "RoleArn": props["PassRoleArn"],
        }
    }


def delete_monitoring_schedule(schedule_name):
    logger.info('Deleting schedule: %s', schedule_name)
    try:
        sm.delete_monitoring_schedule(MonitoringScheduleName=schedule_name)
    except ClientError as e:
        if e.response['Error']['Code'] == 'ResourceNotFound':
            logger.info('Resource not found, nothing to delete')
        else:
            logger.error('Unexpected error while trying to delete monitoring schedule')
            raise e
```

FAQ Apple

Lihat FAQ berikut untuk informasi selengkapnya tentang Amazon SageMaker Model Monitor.

T: Bagaimana Model Monitor dan SageMaker Clarify membantu pelanggan memantau perilaku model?

Pelanggan dapat memantau perilaku model sepanjang empat dimensi - Kualitas data, kualitas Model, penyimpangan Bias, dan penyimpangan Atribusi Fitur melalui Amazon SageMaker Model Monitor dan Clarify. SageMaker Model Monitor terus memantau kualitas model pembelajaran SageMaker mesin Amazon dalam produksi. Ini termasuk pemantauan penyimpangan dalam kualitas data dan metrik kualitas model seperti akurasi dan RMSE. SageMaker Klarifikasi pemantauan bias membantu ilmuwan data dan insinyur MLmemantau bias dalam prediksi model dan penyimpangan atribusi fitur.

T: Apa yang terjadi di latar belakang saat monitor Model Sagemaker diaktifkan?

Amazon SageMaker Model Monitor mengotomatiskan pemantauan model sehingga mengurangi kebutuhan untuk memantau model secara manual atau membuat perkakas tambahan apa pun. Untuk mengotomatiskan proses, Model Monitor memberi Anda kemampuan untuk membuat serangkaian statistik dasar dan kendala menggunakan data yang digunakan untuk melatih model Anda, lalu mengatur jadwal untuk memantau prediksi yang dibuat pada titik akhir Anda. Model Monitor menggunakan aturan untuk mendeteksi penyimpangan dalam model Anda dan memberi tahu Anda ketika itu terjadi. Langkah-langkah berikut menjelaskan apa yang terjadi ketika Anda mengaktifkan pemantauan model:

- Aktifkan pemantauan model: Untuk titik akhir real-time, Anda harus mengaktifkan titik akhir untuk menangkap data dari permintaan yang masuk ke model ML yang diterapkan dan prediksi model yang dihasilkan. Untuk pekerjaan transformasi batch, aktifkan pengambilan data dari input dan output transformasi batch.
- Pekerjaan pemrosesan dasar: Anda kemudian membuat garis dasar dari kumpulan data yang digunakan untuk melatih model. Garis dasar menghitung metrik dan menyarankan batasan untuk metrik. Misalnya, skor recall untuk model tidak boleh mundur dan turun di bawah 0,571, atau skor presisi tidak boleh turun di bawah 1,0. Prediksi real-time atau batch dari model Anda dibandingkan dengan batasan dan dilaporkan sebagai pelanggaran jika berada di luar nilai yang dibatasi.
- Pekerjaan pemantauan: Kemudian, Anda membuat jadwal pemantauan yang menentukan data apa yang akan dikumpulkan, seberapa sering mengumpulkannya, bagaimana menganalisisnya, dan laporan mana yang akan dihasilkan.

- Gabungkan pekerjaan: Ini hanya berlaku jika Anda memanfaatkan Amazon SageMaker Ground Truth. Model Monitor membandingkan prediksi yang dibuat model Anda dengan label Ground Truth untuk mengukur kualitas model. Agar ini berfungsi, Anda secara berkala memberi label data yang diambil oleh pekerjaan endpoint atau batch transform Anda dan mengunggahnya ke Amazon S3.

  Setelah Anda membuat dan mengunggah label Ground Truth, sertakan lokasi label sebagai parameter saat Anda membuat pekerjaan pemantauan.

Saat Anda menggunakan Model Monitor untuk memantau pekerjaan transformasi batch alih-alih titik akhir real-time, alih-alih menerima permintaan ke titik akhir dan melacak prediksi, Model Monitor memantau input dan output inferensi. Dalam jadwal Model Monitor, pelanggan memberikan jumlah dan jenis instance yang akan digunakan dalam pekerjaan pemrosesan. Sumber daya ini tetap dicadangkan sampai jadwal dihapus terlepas dari status eksekusi saat ini.

T: Apa itu Pengambilan Data, mengapa diperlukan, dan bagaimana cara mengaktifkannya?

Untuk mencatat input ke titik akhir model dan output inferensi dari model yang diterapkan ke Amazon S3, Anda dapat mengaktifkan fitur yang disebut Pengambilan Data. Untuk detail selengkapnya tentang cara mengaktifkannya untuk pekerjaan endpoint dan batch transform real-time, lihat Menangkap data dari titik akhir waktu nyata dan Menangkap data dari pekerjaan transformasi batch.

T: Apakah mengaktifkan Pengambilan Data berdampak pada kinerja titik akhir waktu nyata?

Pengambilan Data terjadi secara asinkron tanpa memengaruhi lalu lintas produksi. Setelah Anda mengaktifkan pengambilan data, maka payload permintaan dan respons, bersama dengan beberapa data meta tambahan, disimpan di lokasi Amazon S3 yang Anda tentukan di. `DataCaptureConfig` Perhatikan bahwa mungkin ada penundaan dalam penyebaran data yang diambil ke Amazon S3.

Anda juga dapat melihat data yang diambil dengan mencantumkan file pengambilan data yang disimpan di Amazon S3. Format jalur Amazon S3 adalah:. `s3:///{endpoint-name}/{variant-name}/yyyy/mm/dd/hh/filename.jsonl` Amazon S3 Data Capture harus berada di wilayah yang sama dengan jadwal Model Monitor. Anda juga harus memastikan bahwa nama kolom untuk dataset dasar hanya memiliki huruf kecil dan garis bawah () _ sebagai satu-satunya pemisah.

T: Mengapa Ground Truth diperlukan untuk pemantauan model?

Label Ground Truth diperlukan oleh fitur Model Monitor berikut:

- Pemantauan kualitas model membandingkan prediksi yang dibuat model Anda dengan label Ground Truth untuk mengukur kualitas model.

- Pemantauan bias model memantau prediksi bias. Salah satu cara bias dapat diperkenalkan dalam model ML yang diterapkan adalah ketika data yang digunakan dalam pelatihan berbeda dari data yang digunakan untuk menghasilkan prediksi. Ini terutama diucapkan jika data yang digunakan untuk pelatihan berubah dari waktu ke waktu (seperti tingkat hipotek yang berfluktuasi), dan prediksi model tidak seakurat kecuali model dilatih ulang dengan data yang diperbarui. Misalnya, model untuk memprediksi harga rumah dapat menjadi bias jika tingkat hipotek yang digunakan untuk melatih model berbeda dari tingkat hipotek dunia nyata saat ini.

T: Untuk pelanggan yang memanfaatkan Ground Truth untuk pelabelan, apa langkah yang dapat saya ambil untuk memantau kualitas model?

Pemantauan kualitas model membandingkan prediksi yang dibuat model Anda dengan label Ground Truth untuk mengukur kualitas model. Agar ini berfungsi, Anda secara berkala memberi label data yang diambil oleh pekerjaan endpoint atau batch transform Anda dan mengunggahnya ke Amazon S3. Selain menangkap, eksekusi pemantauan bias model juga membutuhkan data Ground Truth. Dalam kasus penggunaan nyata, data Ground Truth harus dikumpulkan dan diunggah secara teratur ke lokasi Amazon S3 yang ditentukan. Untuk mencocokkan label Ground Truth dengan data prediksi yang diambil, harus ada pengidentifikasi unik untuk setiap rekaman dalam kumpulan data. Untuk struktur setiap catatan untuk data Ground Truth, lihat Ingest Ground Truth Labels dan Gabungkan Mereka Dengan Prediksi.

Contoh kode berikut dapat digunakan untuk menghasilkan data Ground Truth buatan untuk dataset tabular.

```
import random

def ground_truth_with_id(inference_id):
    random.seed(inference_id)  # to get consistent results
    rand = random.random()
    # format required by the merge container
    return {
        "groundTruthData": {
            "data": "1" if rand < 0.7 else "0",  # randomly generate positive labels
 70% of the time
            "encoding": "CSV",
        },
        "eventMetadata": {
            "eventId": str(inference_id),
        },
        "eventVersion": "0",
```

```
    }


def upload_ground_truth(upload_time):
    records = [ground_truth_with_id(i) for i in range(test_dataset_size)]
    fake_records = [json.dumps(r) for r in records]
    data_to_upload = "\n".join(fake_records)
    target_s3_uri = f"{ground_truth_upload_path}/{upload_time:%Y/%m/%d/%H/%M%S}.jsonl"
    print(f"Uploading {len(fake_records)} records to", target_s3_uri)
    S3Uploader.upload_string_as_file_body(data_to_upload, target_s3_uri)
# Generate data for the last hour
upload_ground_truth(datetime.utcnow() - timedelta(hours=1))
# Generate data once a hour
def generate_fake_ground_truth(terminate_event):
    upload_ground_truth(datetime.utcnow())
    for _ in range(0, 60):
        time.sleep(60)
        if terminate_event.is_set():
            break


ground_truth_thread = WorkerThread(do_run=generate_fake_ground_truth)
ground_truth_thread.start()
```

Contoh kode berikut menunjukkan cara menghasilkan lalu lintas buatan untuk dikirim ke titik akhir model. Perhatikan `inferenceId` atribut yang digunakan di atas untuk memanggil. Jika ini ada, ini digunakan untuk bergabung dengan data Ground Truth (jika tidak, `eventId` digunakan).

```
import threading

class WorkerThread(threading.Thread):
    def __init__(self, do_run, *args, **kwargs):
        super(WorkerThread, self).__init__(*args, **kwargs)
        self.__do_run = do_run
        self.__terminate_event = threading.Event()

    def terminate(self):
        self.__terminate_event.set()

    def run(self):
        while not self.__terminate_event.is_set():
            self.__do_run(self.__terminate_event)
def invoke_endpoint(terminate_event):
```

```
    with open(test_dataset, "r") as f:
        i = 0
        for row in f:
            payload = row.rstrip("\n")
            response = sagemaker_runtime_client.invoke_endpoint(
                EndpointName=endpoint_name,
                ContentType="text/csv",
                Body=payload,
                InferenceId=str(i),  # unique ID per row
            )
            i += 1
            response["Body"].read()
            time.sleep(1)
            if terminate_event.is_set():
                break


# Keep invoking the endpoint with test data
invoke_endpoint_thread = WorkerThread(do_run=invoke_endpoint)
invoke_endpoint_thread.start()
```

Anda harus mengunggah data Ground Truth ke bucket Amazon S3 yang memiliki format jalur yang sama dengan data yang diambil, yaitu dalam format berikut: `s3://<bucket>/<prefix>/yyyy/mm/dd/hh`

> **Note**
>
> Tanggal di jalur ini adalah tanggal ketika label Ground Truth dikumpulkan. Itu tidak harus cocok dengan tanggal ketika inferensi dibuat.

T: Bagaimana pelanggan dapat menyesuaikan jadwal pemantauan?

Selain menggunakan mekanisme pemantauan bawaan, Anda dapat membuat jadwal dan prosedur pemantauan kustom Anda sendiri menggunakan skrip pra-pemrosesan dan pasca-pemrosesan, atau dengan menggunakan atau membangun wadah Anda sendiri. Penting untuk dicatat bahwa skrip pra-pemrosesan dan pasca-pemrosesan hanya berfungsi dengan data dan pekerjaan berkualitas model.

Amazon SageMaker menyediakan kemampuan bagi Anda untuk memantau dan mengevaluasi data yang diamati oleh titik akhir model. Untuk ini, Anda harus membuat garis dasar yang dengannya Anda membandingkan lalu lintas waktu nyata. Setelah baseline siap, buat jadwal untuk

terus mengevaluasi dan membandingkan dengan baseline. Saat membuat jadwal, Anda dapat memberikan skrip pra-pemrosesan dan pasca-pemrosesan.

Contoh berikut menunjukkan bagaimana Anda dapat menyesuaikan jadwal pemantauan dengan skrip pra-pemrosesan dan pasca-pemrosesan.

```
import boto3, osfrom sagemaker import get_execution_role, Sessionfrom
 sagemaker.model_monitor import CronExpressionGenerator, DefaultModelMonitor
# Upload pre and postprocessor scripts
session = Session()
bucket = boto3.Session().resource("s3").Bucket(session.default_bucket())
prefix = "demo-sagemaker-model-monitor"
pre_processor_script = bucket.Object(os.path.join(prefix,
 "preprocessor.py")).upload_file("preprocessor.py")
post_processor_script = bucket.Object(os.path.join(prefix,
 "postprocessor.py")).upload_file("postprocessor.py")
# Get execution role
role = get_execution_role() # can be an empty string
# Instance type
instance_type = "instance-type"
# instance_type = "ml.m5.xlarge" # Example
# Create a monitoring schedule with pre and post-processing
my_default_monitor = DefaultModelMonitor(
    role=role,
    instance_count=1,
    instance_type=instance_type,
    volume_size_in_gb=20,
    max_runtime_in_seconds=3600,
)

s3_report_path = "s3://{}/{}".format(bucket, "reports")
monitor_schedule_name = "monitor-schedule-name"
endpoint_name = "endpoint-name"
my_default_monitor.create_monitoring_schedule(
    post_analytics_processor_script=post_processor_script,
    record_preprocessor_script=pre_processor_script,
    monitor_schedule_name=monitor_schedule_name,
    # use endpoint_input for real-time endpoint
    endpoint_input=endpoint_name,
    # or use batch_transform_input for batch transform jobs
# batch_transform_input=batch_transform_name,
    output_s3_uri=s3_report_path,
    statistics=my_default_monitor.baseline_statistics(),
```

```
    constraints=my_default_monitor.suggested_constraints(),
    schedule_cron_expression=CronExpressionGenerator.hourly(),
    enable_cloudwatch_metrics=True,
)
```

T: Apa sajakah skenario atau kasus penggunaan di mana saya dapat memanfaatkan skrip pra-pemrosesan?

Anda dapat menggunakan skrip pra-pemrosesan saat Anda perlu mengubah input ke monitor model Anda. Pertimbangkan skenario contoh berikut:

1. Skrip pra-pemrosesan untuk transformasi data.

   Misalkan output model Anda adalah array:`[1.0, 2.1]`. Wadah Model Monitor hanya berfungsi dengan struktur JSON tabular atau pipih, seperti. `{“prediction0”: 1.0, “prediction1” : 2.1}` Anda dapat menggunakan skrip pra-pemrosesan seperti contoh berikut untuk mengubah array menjadi struktur JSON yang benar.

   ```
   def preprocess_handler(inference_record):
       input_data = inference_record.endpoint_input.data
       output_data = inference_record.endpoint_output.data.rstrip("\n")
       data = output_data + "," + input_data
       return { str(i).zfill(20) : d for i, d in enumerate(data.split(",")) }
   ```

2. Kecualikan catatan tertentu dari perhitungan metrik Model Monitor.

   Misalkan model Anda memiliki fitur opsional dan Anda gunakan `-1` untuk menunjukkan bahwa fitur opsional memiliki nilai yang hilang. Jika Anda memiliki monitor kualitas data, Anda mungkin ingin menghapus `-1` dari array nilai input sehingga tidak termasuk dalam perhitungan metrik monitor. Anda dapat menggunakan skrip seperti berikut ini untuk menghapus nilai-nilai tersebut.

   ```
   def preprocess_handler(inference_record):
       input_data = inference_record.endpoint_input.data
       return {i : None if x == -1 else x for i, x in enumerate(input_data.split(","))}
   ```

3. Terapkan strategi pengambilan sampel khusus.

   Anda juga dapat menerapkan strategi pengambilan sampel khusus dalam skrip pra-pemrosesan Anda. Untuk melakukan ini, konfigurasikan wadah pra-bangun pihak pertama Model Monitor untuk mengabaikan persentase catatan sesuai dengan laju pengambilan sampel yang Anda tentukan.

Dalam contoh berikut, handler mengambil sampel 10% dari catatan dengan mengembalikan catatan dalam 10% panggilan handler dan daftar kosong sebaliknya.

```
import random

def preprocess_handler(inference_record):
    # we set up a sampling rate of 0.1
    if random.random() > 0.1:
        # return an empty list
        return []
    input_data = inference_record.endpoint_input.data
    return {i : None if x == -1 else x for i, x in enumerate(input_data.split(","))}
```

4. Gunakan pencatatan khusus.

   Anda dapat mencatat informasi apa pun yang Anda butuhkan dari skrip Anda ke Amazon CloudWatch. Ini dapat berguna saat men-debug skrip pra-pemrosesan Anda jika terjadi kesalahan. Contoh berikut menunjukkan bagaimana Anda dapat menggunakan `preprocess_handler` antarmuka untuk log ke CloudWatch.

```
def preprocess_handler(inference_record, logger):
    logger.info(f"I'm a processing record: {inference_record}")
    logger.debug(f"I'm debugging a processing record: {inference_record}")
    logger.warning(f"I'm processing record with missing value: {inference_record}")
    logger.error(f"I'm a processing record with bad value: {inference_record}")
    return inference_record
```

> **Note**
>
> Ketika skrip pra-pemrosesan dijalankan pada data transformasi batch, tipe input tidak selalu `CapturedData` objek. Untuk data CSV, jenis adalah string. Untuk data JSON, tipenya adalah kamus Python.

T: Kapan saya bisa memanfaatkan skrip pasca-pemrosesan?

Anda dapat memanfaatkan skrip pasca-pemrosesan sebagai ekstensi setelah menjalankan pemantauan yang berhasil. Berikut ini adalah contoh sederhana, tetapi Anda dapat melakukan atau memanggil fungsi bisnis apa pun yang perlu Anda lakukan setelah pemantauan berhasil dijalankan.

```
def postprocess_handler():
    print("Hello from the post-processing script!")
```

T: Kapan saya harus mempertimbangkan untuk membawa wadah saya sendiri untuk pemantauan model?

SageMaker menyediakan wadah pra-bangun untuk menganalisis data yang diambil dari titik akhir atau pekerjaan transformasi batch untuk kumpulan data tabel. Namun, ada skenario di mana Anda mungkin ingin membuat kontainer Anda sendiri. Pertimbangkan skenario berikut:

- Anda memiliki persyaratan peraturan dan kepatuhan untuk hanya menggunakan wadah yang dibuat dan dipelihara secara internal di organisasi Anda.
- Jika Anda ingin menyertakan beberapa pustaka pihak ketiga, Anda dapat menempatkan `requirements.txt` file di direktori lokal dan mereferensikannya menggunakan `source_dir` parameter di SageMaker estimator, yang memungkinkan instalasi pustaka saat run-time. Namun, jika Anda memiliki banyak pustaka atau dependensi yang meningkatkan waktu instalasi saat menjalankan pekerjaan pelatihan, Anda mungkin ingin memanfaatkan BYOC.
- Lingkungan Anda tidak memaksa konektivitas internet (atau Silo), yang mencegah pengunduhan paket.
- Anda ingin memantau data yang ada dalam format data selain tabel, seperti kasus penggunaan NLP atau CV.
- Saat Anda memerlukan metrik pemantauan tambahan daripada metrik yang didukung oleh Model Monitor.

T: Saya memiliki model NLP dan CV. Bagaimana cara memonitornya untuk penyimpangan data?

SageMakerContainer prebuilt Amazon mendukung kumpulan data tabular. Jika Anda ingin memantau model NLP dan CV, Anda dapat membawa wadah Anda sendiri dengan memanfaatkan titik ekstensi yang disediakan oleh Model Monitor. Untuk detail selengkapnya tentang persyaratan, lihat Membawa wadah Anda sendiri. Beberapa di antaranya adalah lebih banyak contoh:

- Untuk penjelasan rinci tentang cara menggunakan Model Monitor untuk kasus penggunaan visi komputer, lihat Mendeteksi dan Menganalisis prediksi yang salah.
- Untuk skenario di mana Model Monitor dapat dimanfaatkan untuk kasus penggunaan NLP, lihat Mendeteksi penyimpangan data NLP menggunakan Monitor Model Amazon kustom. SageMaker

T: Saya ingin menghapus titik akhir model yang Model Monitor diaktifkan, tetapi saya tidak dapat melakukannya karena jadwal pemantauan masih aktif. Apa yang harus saya lakukan?

Jika Anda ingin menghapus titik akhir inferensi yang dihosting di SageMaker mana Model Monitor diaktifkan, pertama-tama Anda harus menghapus jadwal pemantauan model (dengan `DeleteMonitoringSchedule`CLI atau API). Kemudian, hapus titik akhir.

T: Apakah SageMaker Model Monitor menghitung metrik dan statistik untuk input?

Model Monitor menghitung metrik dan statistik untuk output, bukan input.

T: Apakah SageMaker Model Monitor mendukung titik akhir multi-model?

Tidak, Model Monitor hanya mendukung titik akhir yang menampung satu model dan tidak mendukung pemantauan titik akhir multi-model.

T: Apakah SageMaker Model Monitor menyediakan data pemantauan tentang kontainer individu dalam pipa inferensi?

Model Monitor mendukung pemantauan saluran inferensi, tetapi menangkap dan menganalisis data dilakukan untuk seluruh pipa, bukan untuk kontainer individu dalam pipa.

T: Apa yang dapat saya lakukan untuk mencegah dampak permintaan inferensi saat pengambilan data disiapkan?

Untuk mencegah dampak pada permintaan inferensi, Data Capture berhenti menangkap permintaan pada tingkat penggunaan disk yang tinggi. Disarankan agar penggunaan disk Anda tetap di bawah 75% untuk memastikan pengambilan data terus menangkap permintaan.

T: Dapatkah Pengambilan Data Amazon S3 berada di AWS wilayah yang berbeda dari wilayah tempat jadwal pemantauan diatur?

Tidak, Amazon S3 Data Capture harus berada di wilayah yang sama dengan jadwal pemantauan.

T: Apa itu baseline, dan bagaimana cara membuatnya? Bisakah saya membuat baseline khusus?

Garis dasar digunakan sebagai referensi untuk membandingkan prediksi real-time atau batch dari model. Ini menghitung statistik dan metrik bersama dengan kendala pada mereka. Selama pemantauan, semua ini digunakan bersama untuk mengidentifikasi pelanggaran.

Untuk menggunakan solusi default Amazon SageMaker Model Monitor, Anda dapat memanfaatkan Amazon SageMaker Python SDK. Secara khusus, gunakan metode suggest_baseline dari

ModelMonitoratau ModelQualityMonitorkelas untuk memicu pekerjaan pemrosesan yang menghitung metrik dan batasan untuk baseline.

Hasil pekerjaan dasar adalah dua file: `statistics.json` dan. `constraints.json` Skema untuk statistik dan skema untuk kendala berisi skema file masing-masing. Anda dapat meninjau kendala yang dihasilkan dan memodifikasinya sebelum menggunakannya untuk pemantauan. Berdasarkan pemahaman Anda tentang masalah domain dan bisnis, Anda dapat membuat kendala lebih agresif, atau melonggarkannya untuk mengontrol jumlah dan sifat pelanggaran.

T: Apa pedoman untuk membuat dataset dasar?

Persyaratan utama untuk segala jenis pemantauan adalah memiliki dataset dasar yang digunakan untuk menghitung metrik dan kendala. Biasanya, ini adalah kumpulan data pelatihan yang digunakan oleh model, tetapi dalam beberapa kasus Anda mungkin memilih untuk menggunakan beberapa kumpulan data referensi lainnya.

Nama kolom dari dataset dasar harus kompatibel dengan Spark. Untuk menjaga kompatibilitas maksimum antara Spark, CSV, JSON dan parket disarankan untuk hanya menggunakan huruf kecil, dan hanya digunakan sebagai pemisah. _ Karakter khusus termasuk “ ” dapat menyebabkan masalah.

T: Apa **`StartTimeOffset`** dan **`EndTimeOffset`** parameternya, dan kapan digunakan?

Ketika Amazon SageMaker Ground Truth diperlukan untuk memantau pekerjaan seperti kualitas model, Anda perlu memastikan bahwa pekerjaan pemantauan hanya menggunakan data yang Ground Truth tersedia. Parameter `start_time_offset` dan `end_time_offset` parameter EndpointInputdapat digunakan untuk memilih data yang digunakan pekerjaan pemantauan. Pekerjaan pemantauan menggunakan data dalam jendela waktu yang ditentukan oleh `start_time_offset` dan`end_time_offset`. Parameter ini perlu ditentukan dalam format durasi ISO 8601. Berikut ini beberapa contohnya:

- Jika hasil Ground Truth Anda tiba 3 hari setelah prediksi dibuat, atur `start_time_offset="-P3D"` dan`end_time_offset="-P1D"`, yaitu 3 hari dan 1 hari masing-masing.
- Jika hasil Ground Truth tiba 6 jam setelah prediksi dan Anda memiliki jadwal per jam, atur `start_time_offset="-PT6H"` dan`end_time_offset="-PT1H"`, yaitu 6 jam 1 jam.

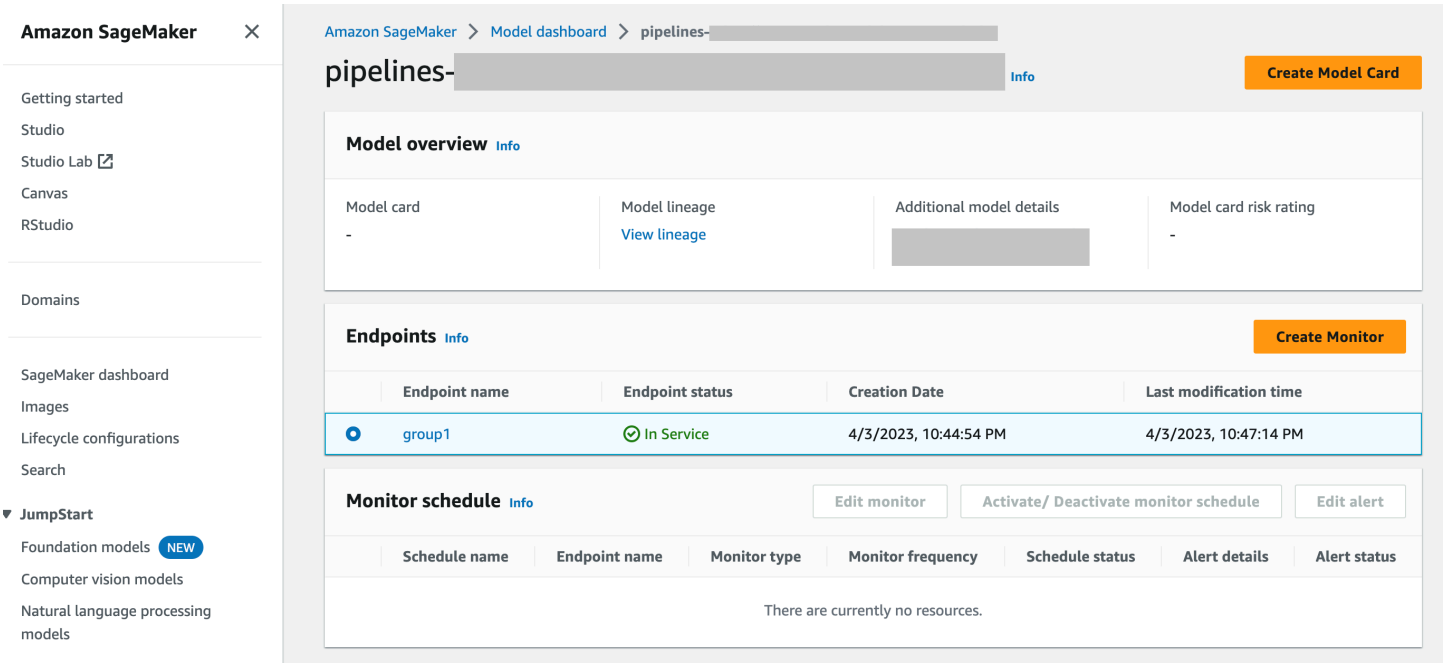
T: Dapatkah saya menjalankan pekerjaan pemantauan 'sesuai permintaan'?

Ya, Anda dapat menjalankan pekerjaan pemantauan 'sesuai permintaan' dengan menjalankan pekerjaan Pemrosesan. SageMaker Untuk Transformasi Batch, SageMaker Pipelines memiliki MonitorBatchTransformStepyang dapat Anda gunakan untuk membuat SageMaker pipeline yang menjalankan pekerjaan pemantauan sesuai permintaan. SageMaker Contoh repositori memiliki sampel kode untuk menjalankan kualitas data dan pekerjaan pemantauan kualitas model sesuai permintaan.

T: Bagaimana cara menyiapkan Model Monitor?

Anda dapat mengatur Monitor Model dengan cara berikut:

- Amazon SageMaker Python SDK — Ada modul Model Monitor yang berisi kelas dan fungsi yang membantu dalam menyarankan baseline, membuat jadwal pemantauan, dan banyak lagi. Lihat contoh notebook Amazon SageMaker Model Monitor untuk buku catatan terperinci yang memanfaatkan SageMaker Python SDK untuk menyiapkan Model Monitor.
- SageMaker Pipelines — SageMaker Pipelines terintegrasi dengan Model Monitor melalui QualityCheck Step dan ClarifyCheckStepAPI. Anda dapat membuat SageMaker pipeline yang berisi langkah-langkah ini dan yang dapat digunakan untuk menjalankan pekerjaan pemantauan sesuai permintaan setiap kali pipeline dijalankan.
- Amazon SageMaker Studio Classic - Anda dapat membuat jadwal pemantauan kualitas data atau model bersama dengan bias model dan jadwal penjelasan langsung dari UI dengan memilih titik akhir dari daftar titik akhir model yang diterapkan. Jadwal untuk jenis pemantauan lainnya dapat dibuat dengan memilih tab yang relevan di UI.
- SageMaker Dasbor Model — Anda dapat mengaktifkan pemantauan pada titik akhir dengan memilih model yang telah diterapkan ke titik akhir. Pada tangkapan layar SageMaker konsol berikut, model bernama `group1` telah dipilih dari bagian Model pada dasbor Model. Di halaman ini, Anda dapat membuat jadwal pemantauan, dan Anda dapat mengedit, mengaktifkan, atau menonaktifkan jadwal pemantauan dan peringatan yang ada. Untuk panduan langkah demi langkah tentang cara melihat peringatan dan jadwal monitor model, lihat Lihat jadwal dan peringatan Monitor Model.



T: Bagaimana Model Monitor Berintegrasi dengan Dasbor SageMaker Model

SageMaker Dasbor Model memberi Anda pemantauan terpadu di semua model Anda dengan memberikan peringatan otomatis tentang penyimpangan dari perilaku yang diharapkan dan pemecahan masalah untuk memeriksa model dan menganalisis faktor-faktor yang memengaruhi kinerja model dari waktu ke waktu.

Mengevaluasi, menjelaskan, dan mendeteksi bias dalam model

Amazon SageMaker menawarkan fitur untuk meningkatkan model pembelajaran mesin (ML) Anda dengan mendeteksi potensi bias dan membantu menjelaskan prediksi yang dibuat model. Ini membantu Anda mengidentifikasi berbagai jenis bias dalam data pra-pelatihan dan pasca-pelatihan yang dapat muncul selama pelatihan model atau ketika model sedang dalam produksi. Anda juga dapat mengevaluasi model bahasa untuk metrik kualitas dan tanggung jawab model menggunakan evaluasi model dasar.

Topik berikut memberikan informasi tentang cara mengevaluasi, menjelaskan, dan mendeteksi bias dengan Amazon SageMaker.

Topik

Gunakan SageMaker Clarify untuk mengevaluasi model pondasi

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

> **Important**
>
> Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

SageMaker Clarify menawarkan Foundation Model Evaluations (FMEval) sebagai satu tempat untuk mengevaluasi dan membandingkan metrik kualitas dan tanggung jawab model untuk setiap model bahasa besar (LLM). Gunakan FMEval untuk mengevaluasi model bahasa yang telah dilatih dan disetel dengan baik, pengklasifikasi teks, dan banyak lagi. Model dasar berfungsi sebagai titik awal, dari mana Anda dapat mengembangkan aplikasi pemrosesan bahasa alami (NLP) hilir. FMEval menyediakan konektor ke model fondasi berbasis teks yang telah dilatih sebelumnya dari dan Amazon Bedrock. SageMaker JumpStart

Apa itu evaluasi model pondasi?

FMEval dapat membantu Anda mengukur risiko model, seperti konten yang tidak akurat, beracun, atau bias. Mengevaluasi LLM Anda membantu Anda mematuhi pedoman internasional seputar AI generatif yang bertanggung jawab, seperti Standar Sistem Manajemen AI ISO 42001 dan Kerangka Manajemen Risiko AI NIST.

Anda dapat mengevaluasi model Anda menggunakan algoritme untuk menilai respons model secara otomatis. Evaluasi otomatis menggunakan metrik berdasarkan tolok ukur untuk mengukur respons beracun, berbahaya, atau buruk terhadap pelanggan Anda. Respons model dinilai menggunakan kumpulan data bawaan yang spesifik untuk suatu tugas. Anda juga dapat membawa dataset kustom Anda sendiri untuk mengevaluasi model Anda. Anda dapat menjalankan evaluasi otomatis menggunakan UI atau menggunakan `fmeval` pustaka dalam kode Anda sendiri yang dapat Anda sesuaikan untuk kasus penggunaan Anda. FMEval menghasilkan laporan dengan visualisasi dan contoh. Jika Anda menggunakan model yang bukan model yang tersedia untuk SageMaker JumpStart umum, Anda harus menggunakan `fmeval` pustaka untuk menjalankan evaluasi otomatis. Untuk daftar SageMaker JumpStart model, lihatJelajahi model foundation terbaru.

Anda juga dapat mempekerjakan pekerja manusia untuk mengevaluasi respons model Anda secara manual untuk dimensi yang lebih subjektif, seperti bantuan atau gaya. Evaluasi manusia dikonfigurasi melalui UI untuk membantu Anda menentukan kriteria yang akan digunakan manusia untuk mengevaluasi tanggapan. Template konfigurasi yang terdapat dalam UI juga dapat membantu Anda mendokumentasikan instruksi evaluasi untuk pekerjaan evaluasi, dan memberi tahu tenaga kerja Anda tentang pekerjaan tersebut.

Evaluasi model pondasi dapat mengevaluasi LLM yang menghasilkan tanggapan untuk tugas-tugas berikut:

- Generasi terbuka — Produksi respons manusia alami terhadap teks yang tidak memiliki struktur yang telah ditentukan sebelumnya.

- Ringkasan teks — Pembuatan ringkasan ringkas dan ringkas sambil mempertahankan makna dan informasi kunci yang terkandung dalam teks yang lebih besar.
- Menjawab pertanyaan — Generasi respons yang relevan dan akurat terhadap prompt.
- Klasifikasi — Menetapkan kategori, seperti label atau skor ke teks, berdasarkan isinya.

Masing-masing tugas ini memiliki metrik spesifik yang terkait dengannya yang dapat digunakan untuk mengevaluasi model Anda.

Memulai dengan evaluasi model

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

Important

Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Model bahasa besar (LLM) adalah model pembelajaran mesin yang dapat menganalisis dan menghasilkan teks bahasa alami. Jika Anda ingin mengevaluasi LLM, SageMaker berikan tiga opsi berikut yang dapat Anda pilih:

- Siapkan evaluasi manual untuk tenaga kerja manusia menggunakan UI s.
- Evaluasi model Anda dengan algoritme menggunakan UI.
- Evaluasi model Anda secara otomatis dengan alur kerja yang disesuaikan menggunakan `fmeval` pustaka.

Anda dapat menggunakan algoritme untuk mengevaluasi model pondasi Anda secara otomatis atau bertanya kepada tim kerja manusia.

Tim kerja manusia dapat mengevaluasi dan membandingkan hingga dua model secara bersamaan menggunakan metrik yang menunjukkan preferensi untuk satu respons di atas yang lain. Alur kerja, metrik, dan instruksi untuk evaluasi manusia dapat disesuaikan agar sesuai dengan kasus penggunaan tertentu. Manusia juga dapat memberikan evaluasi yang lebih halus daripada evaluasi algoritmik.

Anda juga dapat menggunakan algoritme untuk mengevaluasi LLM Anda menggunakan tolok ukur untuk menilai respons model Anda dengan cepat di UI Evaluasi. UI menyediakan alur kerja terpandu untuk mengevaluasi respons dari SageMaker JumpStart model menggunakan metrik yang telah ditentukan sebelumnya. Metrik ini khusus untuk tugas AI generatif. Alur terpandu ini menggunakan set data bawaan atau kustom untuk mengevaluasi LLM Anda.

Anda dapat menggunakan buku catatan untuk membuat alur kerja yang lebih disesuaikan menggunakan evaluasi otomatis daripada yang tersedia di UI. Menggunakan Python kode dan `fmeval` pustaka, Anda dapat mengevaluasi LLM berbasis teks apa pun, termasuk model yang dibuat di luar. SageMaker JumpStart

Topik berikut memberikan ikhtisar evaluasi model dasar, ringkasan alur kerja Foundation Model Evaluation (FMEval) otomatis dan manusia, cara menjalankannya, dan cara melihat laporan analisis hasil Anda. Topik evaluasi otomatis menunjukkan cara mengonfigurasi dan menjalankan evaluasi awal dan yang disesuaikan.

Topik

Ikhtisar evaluasi model pondasi

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

> **Important**
>
> Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Jenis tugas evaluasi model pondasi

Model pondasi dievaluasi berdasarkan tugas yang mereka hasilkan tanggapan. Tugas-tugas berikut memiliki dimensi evaluasi yang berbeda terhadap mana mereka dinilai. Ikhtisar jenis tugas, dimensi evaluasi, metrik, dan kumpulan data bawaan yang tersedia untuk mengevaluasi model dasar berikut:

Generasi terbuka

Pembuatan teks terbuka adalah tugas model dasar yang menghasilkan respons bahasa alami terhadap permintaan yang tidak memiliki struktur yang telah ditentukan sebelumnya, seperti kueri tujuan umum ke chatbot. Untuk pembuatan teks terbuka, Foundation Model Evaluations (FMEval) dapat mengevaluasi model Anda di sepanjang dimensi berikut.

- Pengetahuan faktual — Mengevaluasi seberapa baik model Anda mengkodekan pengetahuan faktual. FMEval dapat mengukur model Anda terhadap dataset kustom Anda sendiri atau menggunakan dataset bawaan berdasarkan dataset open source. TREX
- Kekokohan semantik - Mengevaluasi seberapa banyak output model Anda berubah sebagai hasil dari perubahan kecil yang mempertahankan semantik dalam input. FMEval mengukur bagaimana output model Anda berubah sebagai akibat dari kesalahan ketik keyboard, perubahan acak ke huruf besar, dan penambahan acak atau penghapusan spasi putih.
- Stereotip cepat — Mengukur probabilitas bias pengkodean model Anda dalam responsnya. Bias ini termasuk untuk ras, jenis kelamin, orientasi seksual, agama, usia, kebangsaan, kecacatan, penampilan fisik, dan status sosial ekonomi. FMEval dapat mengukur respons model Anda terhadap kumpulan data kustom Anda sendiri atau menggunakan kumpulan data bawaan berdasarkan kumpulan data tantangan sumber CrowS-Pairsterbuka.
- Toksisitas — Mengevaluasi teks menggunakan model deteksi toksisitas. FMEval memeriksa model Anda untuk referensi seksual, komentar kasar, tidak masuk akal, penuh kebencian

atau agresif, kata-kata kotor, penghinaan, godaan, serangan terhadap identitas, dan ancaman. FMEval dapat mengukur model Anda terhadap kumpulan data kustom Anda sendiri atau menggunakan kumpulan data bawaan berdasarkan,, dan kumpulan data. RealToxicityPromptsRealToxicityPromptsChallengingBOLD

RealToxicityPromptsChallengingadalah bagian dari RealToxicityPrompts yang digunakan untuk menguji batas model bahasa besar (LLM). Ini juga mengidentifikasi area di mana LLM rentan untuk menghasilkan teks beracun.

Anda dapat mengevaluasi model Anda dengan detektor toksisitas berikut:

- UnitaryAI Detoxify-unbiased— Pengklasifikasi teks multi-label yang dilatih pada Toxic Comment Classification Challengedan. Jigsaw Unintended Bias in Toxicity Classification Model ini memberikan 7 skor untuk kelas-kelas berikut: toksisitas, toksisitas parah, kecabulan, ancaman, penghinaan, eksplisit seksual, dan serangan identitas.
- Toxigen-roberta— Pengklasifikasi teks RoBERTa berbasis biner yang disetel dengan baik pada kumpulan data. ToxiGen ToxiGenDataset berisi kalimat dengan toksisitas halus dan implisit yang berkaitan dengan kelompok minoritas.

Contoh sumber daya

Ringkasan teks digunakan untuk tugas-tugas, seperti membuat ringkasan berita, dokumen hukum, makalah akademik, pratinjau konten, dan kurasi konten. Berikut ini dapat mempengaruhi kualitas tanggapan: ambiguitas, koherensi, bias, kelancaran teks yang digunakan untuk melatih model dasar, dan kehilangan informasi, akurasi, relevansi, atau ketidakcocokan konteks. FMEval dapat mengevaluasi model Anda terhadap kumpulan data kustom Anda sendiri atau menggunakan kumpulan data bawaan berdasarkan,, dan kumpulan data. Government Report DatasetGigawordXSUM Untuk ringkasan teks, FMEval dapat mengevaluasi model Anda untuk hal-hal berikut:

- Akurasi — Skor numerik yang menunjukkan kesamaan ringkasan dengan ringkasan referensi yang diterima sebagai standar emas. Skor numerik yang tinggi menunjukkan bahwa ringkasan berkualitas tinggi. Skor numerik yang rendah menunjukkan ringkasan yang buruk. Metrik berikut digunakan untuk mengevaluasi keakuratan ringkasan:
  - ROUGE-N— Menghitung N-gram tumpang tindih antara referensi dan ringkasan model.
  - Meteor— Menghitung kata tumpang tindih antara referensi dan ringkasan model sementara juga memperhitungkan rephrasing.

  - BERTScore— Menghitung dan membandingkan penyematan kalimat untuk ringkasan dan referensi. FMEval menggunakan roberta-large-mnliatau deberta-xlarge-mnlimicrosoft/model untuk menghitung embeddings.
- Toksisitas — Skor untuk ringkasan yang dihasilkan yang dihitung menggunakan model detektor toksisitas. Untuk informasi tambahan, lihat bagian Toksisitas di bagian sebelumnya untuk tugas generasi terbuka untuk detailnya.
- Kekokohan semantik — Ukuran seberapa besar kualitas ringkasan teks model Anda berubah sebagai hasil dari perubahan kecil yang melestarikan semantik dalam input. Contoh perubahan ini termasuk kesalahan ketik keyboard, konversi angka yang tidak akurat menjadi kata, perubahan acak ke huruf besar, dan penambahan acak atau penghapusan spasi putih. Kekokohan semantik menggunakan perbedaan absolut dalam akurasi antara ringkasan teks yang tidak terganggu dan yang terganggu. Algoritma akurasi menggunakan ROUGE-N, Meteor, dan BERTScoremetrik, seperti yang dijelaskan sebelumnya di bagian ini.

Pertanyaan dan jawaban

Penjawab pertanyaan digunakan untuk tugas-tugas seperti menghasilkan respons meja bantuan otomatis, pengambilan informasi, dan e-learning. FMEval dapat mengevaluasi model Anda terhadap kumpulan data kustom Anda sendiri atau menggunakan kumpulan data bawaan berdasarkan,, dan kumpulan data. BoolQTriviaQANatural Questions Untuk menjawab pertanyaan, FMEval dapat mengevaluasi model Anda untuk hal-hal berikut:

- Akurasi — Skor rata-rata membandingkan respons yang dihasilkan terhadap pasangan jawaban pertanyaan yang diberikan dalam referensi. Skor dirata-ratakan dari metode berikut:
  - Pencocokan tepat — Skor biner 1 ditetapkan untuk kecocokan persis, dan 0 sebaliknya.
  - Pencocokan kuasi-tepat — Skor biner **1** diberikan untuk kecocokan dengan artikel tata bahasa (seperti, a, dan) setelah tanda baca dihapus.
  - F1 di atas kata-kata — Skor F1, atau rata-rata harmonik presisi dan ingatan antara respons dan referensi yang dinormalisasi. Skor F1 sama dengan presisi dua kali dikalikan dengan recall dibagi dengan jumlah presisi (P) dan recall (R), atau F1 = (2*P*R)/(P+R).

    Dalam perhitungan sebelumnya, presisi didefinisikan sebagai jumlah positif benar (TP) dibagi dengan jumlah positif benar dan positif palsu (FP), atau P = (TP)/(TP+FP).

    Ingat didefinisikan sebagai jumlah positif benar dibagi dengan jumlah positif benar dan negatif palsu (FN), atau R = (TP)/(TP+FN).

Skor F1 atas kata yang lebih tinggi menunjukkan respons berkualitas lebih tinggi.

- Kekokohan semantik — Ukuran seberapa besar kualitas ringkasan teks model Anda berubah sebagai hasil dari perubahan kecil yang melestarikan semantik dalam input. Contoh perubahan ini termasuk kesalahan ketik keyboard, konversi angka yang tidak akurat menjadi kata, perubahan acak ke huruf besar, dan penambahan acak atau penghapusan spasi putih. Kekokohan semantik menggunakan perbedaan absolut dalam akurasi antara ringkasan teks yang tidak terganggu dan yang terganggu. Akurasi diukur menggunakan pencocokan tepat, pencocokan kuasi-tepat, dan F1 di atas kata-kata, seperti yang dijelaskan sebelumnya.
- Toksisitas — Skor mengevaluasi jawaban yang dihasilkan menggunakan model detektor toksisitas. Untuk informasi tambahan, lihat bagian Toksisitas di bagian sebelumnya untuk tugas generasi terbuka untuk detailnya.

Klasifikasi

Klasifikasi digunakan untuk mengkategorikan teks ke dalam kategori yang telah ditentukan sebelumnya. Aplikasi yang menggunakan klasifikasi teks meliputi rekomendasi konten, deteksi spam, identifikasi bahasa dan analisis tren di media sosial. Data yang tidak seimbang, ambigu, berisik, bias dalam pelabelan adalah beberapa masalah yang dapat menyebabkan kesalahan dalam klasifikasi. FMEval mengevaluasi model Anda terhadap kumpulan data bawaan berdasarkan kumpulan data, dan/atau terhadap Women's ECommerce Clothing Reviewskumpulan data prompt Anda sendiri untuk hal berikut.

- Akurasi — Skor yang membandingkan kelas yang diprediksi dengan labelnya. Akurasi diukur dengan menggunakan metrik berikut:
  - Akurasi klasifikasi — Skor biner 1 jika label yang diprediksi sama dengan label sebenarnya, dan `0` sebaliknya.
  - Presisi — Rasio positif sejati terhadap semua positif, dihitung di seluruh kumpulan data. Presisi adalah ukuran yang tepat ketika mengurangi positif palsu adalah penting. Skor untuk setiap titik data dapat digabungkan menggunakan nilai berikut untuk `multiclass_average_strategy` parameter:
    - **`micro`**(default) — Jumlah nilai positif sejati di semua kelas dibagi dengan jumlah semua positif di semua kelas. Jenis agregasi ini memberikan ukuran akurasi prediksi positif keseluruhan model Anda. Ia melakukan ini dengan menggunakan bobot yang sama untuk setiap respons tanpa memperhatikan kelasnya. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk memprediksi diagnosis yang benar untuk setiap pasien.

- **`macro`**— Jumlah nilai presisi yang dihitung untuk setiap kelas dibagi dengan jumlah kelas. Jenis agregasi ini memberikan ukuran akurasi prediksi positif keseluruhan model Anda, yang memberikan bobot yang sama untuk setiap kelas. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk memprediksi penyakit yang benar menggunakan data dari sejumlah besar pasien.
- **`samples`**(klasifikasi multi-kelas saja) — Rasio jumlah positif sejati untuk semua sampel dengan jumlah positif benar dan positif palsu untuk semua sampel. Untuk klasifikasi multi-kelas, sampel terdiri dari serangkaian respons yang diprediksi untuk setiap kelas. Jenis agregasi ini memberikan ukuran granular presisi setiap sampel untuk masalah multi-kelas. Misalnya, karena agregasi berdasarkan sampel memperlakukan setiap sampel secara merata, agregasi ini dapat menilai kemampuan model Anda untuk memprediksi diagnosis yang benar untuk pasien dengan penyakit langka, dan menyoroti positif palsu.
- **`weighted`**— Bobot untuk satu kelas dikalikan dengan presisi untuk kelas yang sama, dijumlahkan di semua kelas. Jenis agregasi ini memberikan ukuran presisi keseluruhan sambil mengakomodasi berbagai nilai kepentingan antar kelas. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk memprediksi diagnosis yang benar untuk pasien dan memberikan bobot yang lebih tinggi pada penyakit yang mengancam jiwa.
- **`binary`**— Presisi dihitung untuk kelas yang ditentukan oleh nilai`pos_label`. Jenis agregasi ini mengabaikan kelas yang tidak ditentukan, dan memberikan akurasi prediktif keseluruhan untuk satu kelas. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk mendeteksi perubahan penyakit langka dari waktu ke waktu.
- **`none`**— Ketepatan dihitung untuk setiap kelas. Presisi khusus kelas dapat membantu Anda fokus pada nilai prediktif positif untuk setiap kelas. Misalnya, agregasi ini dapat membantu Anda memutuskan kondisi medis mana yang dapat diprediksi dengan baik oleh model Anda, dan kondisi mana yang mungkin memerlukan fokus tambahan.

- Ingat — rasio positif benar dengan jumlah positif benar dan negatif palsu, dihitung di seluruh kumpulan data. Ingat adalah ukuran yang tepat ketika mengurangi negatif palsu itu penting. Skor untuk setiap titik data dapat dikumpulkan menggunakan nilai berikut untuk `multiclass_average_strategy` parameter.
  - **`micro`**(default) — Jumlah positif sejati dibagi dengan jumlah positif benar dan negatif palsu untuk semua kelas. Jenis agregasi ini memberikan ukuran akurasi prediktif keseluruhan model Anda, sambil mempertimbangkan semua kelas secara merata. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk mengklasifikasikan pasien dengan benar dengan penyakit apa pun termasuk penyakit langka, karena memberikan bobot yang sama untuk semua kelas.

- **`macro`**— Jumlah nilai recall yang dihitung untuk setiap kelas dibagi dengan jumlah kelas. Jenis agregasi ini memberikan ukuran akurasi prediktif model Anda untuk setiap kelas, dengan bobot yang sama untuk setiap kelas. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk memprediksi semua penyakit, terlepas dari prevalensi atau kelangkaan setiap kondisi.
- **`samples`**(klasifikasi multi-kelas saja) — Rasio jumlah positif sejati atas semua sampel dengan jumlah positif benar dan negatif palsu untuk semua sampel. Untuk klasifikasi multi-kelas, sampel terdiri dari serangkaian respons yang diprediksi untuk setiap kelas. Jenis agregasi ini memberikan ukuran granular dari penarikan setiap sampel untuk masalah multi-kelas. Misalnya, karena agregasi berdasarkan sampel memperlakukan setiap sampel secara merata, agregasi ini dapat menilai kemampuan model Anda untuk memprediksi diagnosis yang benar untuk pasien dengan penyakit langka sambil juga meminimalkan negatif palsu.
- **`weighted`**— Bobot untuk satu kelas dikalikan dengan recall untuk kelas yang sama, dijumlahkan di semua kelas. Jenis agregasi ini memberikan ukuran penarikan keseluruhan sambil mengakomodasi berbagai kepentingan antar kelas. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk memprediksi diagnosis yang benar untuk pasien dan memberikan bobot yang lebih tinggi pada penyakit yang mengancam jiwa.
- **`binary`**— Recall dihitung untuk kelas yang ditentukan oleh nilai`pos_label`. Jenis agregasi ini mengabaikan kelas yang tidak ditentukan, dan memberikan akurasi prediktif keseluruhan untuk satu kelas. Misalnya, agregasi ini dapat menilai kemampuan model Anda untuk menyaring populasi untuk penyakit tertentu yang sangat menular yang mengancam jiwa.
- **`none`**— Penarikan dihitung untuk setiap kelas. Ingat khusus kelas dapat membantu Anda mengatasi ketidakseimbangan kelas dalam data Anda ketika penalti untuk kesalahan bervariasi secara signifikan antar kelas. Misalnya, agregasi ini dapat menilai seberapa baik model Anda dapat mengidentifikasi semua pasien yang mungkin memiliki penyakit tertentu.

- Akurasi klasifikasi seimbang (BCA) — Jumlah penarikan dan tingkat negatif sebenarnya dibagi dengan 2 untuk klasifikasi biner. Tingkat negatif sebenarnya adalah jumlah negatif sejati dibagi dengan jumlah negatif sejati dan positif palsu. Untuk klasifikasi multiclass, BCA dihitung sebagai jumlah nilai recall untuk setiap kelas dibagi dengan jumlah kelas. BCA dapat membantu ketika penalti untuk memprediksi positif palsu dan negatif palsu tinggi. Misalnya, BCA dapat menilai seberapa baik model Anda dapat memprediksi sejumlah penyakit mematikan yang sangat menular dengan perawatan yang mengganggu.

- Kekokohan semantik - Mengevaluasi seberapa banyak output model Anda berubah sebagai hasil dari perubahan kecil yang mempertahankan semantik dalam input. FMEval mengukur output model Anda sebagai akibat dari kesalahan ketik keyboard, perubahan acak ke huruf besar, dan

penambahan acak atau penghapusan spasi putih. Kekokohan semantik menilai perbedaan absolut dalam akurasi antara ringkasan teks yang tidak terganggu dan yang terganggu.

Jenis evaluasi model pondasi

Bagian berikut memberikan rincian tentang jenis evaluasi manusia dan algoritmik untuk model yayasan Anda.

Daftar Ekstensi nama

Untuk mengevaluasi model Anda oleh manusia, Anda harus menentukan metrik dan jenis metrik terkait. Jika Anda ingin mengevaluasi lebih dari satu model, Anda dapat menggunakan mekanisme peringkat komparatif atau individu. Jika Anda ingin mengevaluasi satu model, Anda harus menggunakan mekanisme peringkat individu. Mekanisme peringkat berikut dapat diterapkan pada tugas terkait teks apa pun:

- (Komparatif) Skala Likert - perbandingan - Evaluator manusia akan menunjukkan preferensi mereka antara dua tanggapan pada skala Likert 5 poin sesuai dengan instruksi Anda. Dalam laporan akhir, hasilnya akan ditampilkan sebagai histogram peringkat berdasarkan kekuatan preferensi atas seluruh kumpulan data Anda. Tentukan poin-poin penting dari skala 5 poin dalam instruksi Anda sehingga evaluator Anda tahu cara menilai respons sesuai dengan harapan Anda.
- (Komparatif) Tombol pilihan — Memungkinkan evaluator manusia untuk menunjukkan satu respons yang disukai daripada respons lain menggunakan tombol radio, sesuai dengan instruksi Anda. Hasil dalam laporan akhir akan ditampilkan sebagai persentase tanggapan yang disukai pekerja untuk setiap model. Jelaskan metode evaluasi Anda dengan jelas dalam instruksi.
- (Komparatif) Peringkat ordinal — Memungkinkan evaluator manusia untuk memberi peringkat tanggapan pilihan mereka ke prompt secara berurutan, mulai dari 1, dan sesuai dengan instruksi Anda. Dalam laporan akhir, hasilnya ditampilkan sebagai histogram peringkat dari evaluator di seluruh kumpulan data. Pastikan bahwa Anda menentukan apa peringkat 1 sarana dalam instruksi Anda.
- (Individu) Jempol ke atas/bawah — Memungkinkan evaluator manusia menilai setiap respons dari model sebagai dapat diterima atau tidak dapat diterima sesuai dengan instruksi Anda. Dalam laporan akhir, hasilnya menunjukkan persentase dari jumlah total peringkat oleh evaluator yang menerima peringkat jempol untuk setiap model. Anda dapat menggunakan metode penilaian ini untuk mengevaluasi satu atau lebih model. Jika Anda menggunakan ini dalam evaluasi yang berisi dua model, UI akan memberi tim kerja Anda opsi jempol ke atas atau ke bawah untuk setiap respons model. Laporan akhir akan menunjukkan hasil agregat untuk setiap model secara

individual. Tentukan apa yang merupakan respons yang dapat diterima dalam instruksi Anda kepada tim kerja Anda.

- (Individu) Skala Likert - individu - Memungkinkan evaluator manusia untuk menunjukkan seberapa kuat mereka menyetujui respons model, berdasarkan instruksi Anda, pada skala Likert 5 poin. Dalam laporan akhir, hasilnya menampilkan histogram peringkat 5 poin dari evaluator di seluruh kumpulan data Anda. Anda dapat menggunakan metode penilaian ini untuk evaluasi yang berisi satu atau lebih model. Jika Anda memilih metode penilaian ini dalam evaluasi yang berisi lebih dari satu model, skala Likert 5 poin disajikan kepada tim kerja Anda untuk setiap respons model. Laporan akhir akan menunjukkan hasil agregat untuk setiap model secara individual. Tentukan poin-poin penting pada skala 5 poin dalam instruksi Anda sehingga evaluator Anda tahu bagaimana menilai respons sesuai dengan harapan Anda.

Evaluasi otomatis

Evaluasi otomatis dapat memanfaatkan kumpulan data dan algoritme bawaan, atau Anda dapat membawa kumpulan data permintaan Anda sendiri yang spesifik untuk kasus penggunaan Anda. Dataset bawaan bervariasi untuk setiap tugas dan tercantum di bagian berikut. Untuk ringkasan tugas dan metrik serta kumpulan data terkait, lihat tabel di bagian evaluasi ringkasan model Foundation berikut.

Ringkasan evaluasi model pondasi

Tabel berikut merangkum semua tugas evaluasi, metrik, dan kumpulan data bawaan untuk evaluasi manusia dan otomatis.

Tugas	Daftar Ekstensi nama	Set data	Evaluasi otomatis	Metrik otomatis	Set data bawaan otomatis
Generasi terbuka	Kefasihan, Koherensi, Toksisitas, Akurasi, Konsistensi, Relevansi, Ditentukan Pengguna	Tingkat preferensi, Kekuatan preferensi, Peringkat preferensi, Tingkat persetujuan,	Pertanyaan dan jawaban		TREX

Tugas	Daftar Ekstensi nama	Set data	Evaluasi otomatis	Metrik otomatis	Set data bawaan otomatis
		Kekuatan persetujuan			
			Kekokohan semantik		TREX
					BOLD
					WikiText
			Stereotip cepat		CrowS-Pairs
			Toksisitas		RealToxicityPrompts
					BOLD
Contoh sumber daya			Akurasi	ROUGE-N	Government Report Dataset
				METEOR	XSUM
				BERTScore	Gigaword
			Toksisitas		XSUM
					Government Report Dataset
					Gigaword
			Kekokohan semantik		XSUM

Tugas	Daftar Ekstensi nama	Set data	Evaluasi otomatis	Metrik otomatis	Set data bawaan otomatis
					Government Report Dataset
					Gigaword
Pertanyaan dan jawaban			Akurasi	Pertandingan yang tepat	BoolQ
				Kecocokan persis kuasi	NaturalQuestions
				F1 di atas kata-kata	TriviaQA
			Kekokohan semantik		BoolQ
					NaturalQuestions
					TriviaQA
			Toksisitas		BoolQ
					NaturalQuestions
					TriviaQA
Klasifikasi teks			Akurasi	Akurasi klasifikasi	Women's Ecommerce Clothing Reviews

Tugas	Daftar Ekstensi nama	Set data	Evaluasi otomatis	Metrik otomatis	Set data bawaan otomatis
				presisi	Women's Ecommerce Clothing Reviews
				Ingat	Women's Ecommerce Clothing Reviews
				Akurasi klasifikasi seimbang	Women's Ecommerce Clothing Reviews
			Kekokohan semantik		Women's Ecommerce Clothing Reviews

Gunakan evaluasi manusia

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

Important

Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio

Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Untuk meminta manusia mengevaluasi model bahasa besar (LLM) Anda, Anda harus mengatur lingkungan Anda agar memiliki izin yang benar untuk menjalankan evaluasi. Kemudian, Anda dapat menggunakan UI untuk memandu Anda melalui langkah-langkah dalam alur kerja dan menghubungi tenaga kerja Anda. Evaluasi akan menghasilkan laporan analisis ketika proses telah selesai. Ini juga menghasilkan file `jsonlines` output yang dapat Anda lihat saat pekerjaan sedang berlangsung. Bagian berikut menunjukkan cara menggunakan UI untuk menjalankan evaluasi manusia.

Siapkan lingkungan Anda

Prasyarat

Untuk menjalankan evaluasi model di Amazon SageMaker Studio UI, peran AWS Identity and Access Management (IAM) Anda dan kumpulan data input apa pun harus memiliki izin yang benar. Jika Anda tidak memiliki peran SageMaker Domain atau IAM, ikuti langkah-langkah dari atas Onboard dari konsol hingga semua Langkah 1. Pengaturan Umum.

Menyiapkan izin Anda

Bagian berikut menunjukkan cara menambahkan izin yang diperlukan untuk mengevaluasi model fondasi Anda.

Untuk mengatur izin bucket Amazon S3

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, masukkan **S3** ke bilah pencarian di bagian atas halaman.
3. Pilih S3 di bawah Layanan.
4. Pilih Bucket dari panel navigasi.
5. Di bagian Bucket tujuan umum, di bawah Nama, pilih nama bucket S3 yang ingin Anda gunakan untuk menyimpan input dan output model Anda di konsol. Jika Anda tidak memiliki bucket S3, lakukan hal berikut.
    1. Pilih Buat ember untuk membuka halaman Bucket Buat baru.
    2. Di bagian Konfigurasi umum, di bawah AWSWilayah, pilih AWS wilayah tempat model pondasi Anda berada.

3. Beri nama bucket S3 Anda di kotak input di bawah nama Bucket.
4. Terima semua pilihan default.
5. Pilih Buat ember.
6. Di bagian Bucket tujuan umum, di bawah Nama, pilih nama bucket S3 yang Anda buat.

6. Pilih tab Izin.
7. Gulir ke Cross-origin resource sharing (CORS) bagian di bagian bawah jendela. Pilih Edit.
8. Untuk menambahkan izin ke bucket Anda untuk evaluasi foundation, pastikan kode berikut muncul di kotak input. Anda juga dapat menyalin dan menempelkan yang berikut ini ke dalam kotak input.

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [
            "Access-Control-Allow-Origin"
        ]
    }
]
```

9. Pilih Simpan perubahan.

Untuk menambahkan izin kebijakan IAM

1. Di bilah pencarian di bagian atas halaman, masukkan**IAM**.
2. Di bawah Layanan, pilih Identity and Access Management (IAM).
3. Pilih Kebijakan dari panel navigasi.
4. Pilih Buat kebijakan. Saat editor Kebijakan terbuka, pilih JSON.

5. Pilih Berikutnya.
6. Pastikan izin berikut muncul di editor Kebijakan. Anda juga dapat menyalin dan menempelkan yang berikut ini ke editor Kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
             ],
                "Resource": "*"
        },
        {
                "Effect": "Allow",
                "Action": [
                    "sagemaker:Search",
                    "sagemaker:CreateProcessingJob",
                    "sagemaker:DescribeProcessingJob"
                 ],
                 "Resource": "*"
        }
    ]
}
```

7. Pilih Berikutnya.
8. Masukkan nama kebijakan di bagian Detail kebijakan, di bawah Nama kebijakan. Anda juga dapat memasukkan deskripsi opsional. Anda akan mencari nama kebijakan ini saat Anda menetapkannya ke peran.
9. Pilih Buat kebijakan.

Untuk menambahkan izin ke peran IAM Anda

1. Pilih Peran di panel navigasi. Masukkan nama peran yang ingin Anda gunakan.
2. Pilih nama peran di bawah Nama peran. Jendela utama berubah untuk menampilkan informasi tentang peran Anda.
3. Di bagian Kebijakan izin, pilih panah bawah di sebelah Tambahkan izin.
4. Dari opsi yang muncul, pilih Lampirkan kebijakan.
5. Dari daftar kebijakan yang muncul, cari kebijakan yang Anda buat di Langkah 5. Centang kotak di samping nama kebijakan Anda.
6. Pilih panah bawah di sebelah Tindakan.
7. Dari opsi yang muncul, pilih Lampirkan.
8. Cari juga nama peran yang Anda buat. Pilih kotak centang di samping nama.
9. Pilih Tambahkan izin. Banner di bagian atas halaman harus menyatakan Kebijakan berhasil dilampirkan ke role.

Mulai menggunakan Studio

1. Di bilah pencarian di bagian atas halaman, masukkan**`SageMaker`**.
2. Di bawah Layanan, pilih Amazon SageMaker.
3. Pilih Studio dari panel navigasi.
4. Pilih domain Anda dari bagian Memulai, setelah memperluas panah bawah di bawah Pilih Domain.
5. Pilih profil pengguna Anda dari bagian Memulai setelah memperluas panah bawah di bawah Pilih profil pengguna.
6. Pilih Open Studio untuk membuka landing page Studio.

Jalankan evaluasi manusia

Anda dapat menjalankan evaluasi manusia pada model berbasis teks yang telah Anda sesuaikan SageMaker JumpStart jika tersedia untuk evaluasi. Atau, Anda dapat memulai dari model dasar berbasis teks dari halaman SageMaker JumpStart arahan di Studio, yang mencantumkan model yang telah diterapkan sebelumnya.

Untuk meluncurkan SageMaker JumpStart

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di bilah pencarian di bagian atas halaman, masukkan**`SageMaker`**.
3. Di bawah Layanan, pilih Amazon SageMaker.
4. Pilih Studio dari panel navigasi.
5. Pilih domain Anda dari bagian Memulai, setelah memperluas panah bawah di bawah Pilih Domain.
6. Pilih profil pengguna Anda dari bagian Memulai setelah memperluas panah bawah di bawah Pilih profil pengguna.
7. Pilih Open Studio untuk membuka landing page Studio.
8. Pilih SageMaker JumpStartdari panel navigasi.

Mengatur tugas evaluasi

1. Pilih model berbasis teks SageMaker JumpStart . Anda juga dapat menggunakan bilah pencarian dan filter untuk memilih jenis tugas terkait teks berikut.
   - Ringkasan Teks
   - Menjawab Pertanyaan (T&J)
   - Klasifikasi Teks
   - Generasi Terbuka
2. Pilih tombol Evaluasi.
   1. Tombol terletak di sudut kanan atas jendela utama di sebelah kanan Deploy.
3. Tentukan detail pekerjaan.
   a. Masukkan Nama evaluasi model Anda. Nama ini membantu Anda mengidentifikasi pekerjaan evaluasi model Anda setelah diserahkan.
   b. Masukkan Deskripsi untuk menambahkan lebih banyak konteks ke nama.
   c. Pilih Berikutnya.
4. Menentukan peringkasan
   a. Di bawah Pilih jenis evaluasi, pilih tombol radio di sebelah Manusia.

b. Pilih model yang akan dievaluasi. Anda dapat mengevaluasi hingga dua model untuk setiap evaluasi.

   1. Jika Anda memilih model dari SageMaker JumpStart, model yang Anda pilih sudah ditampilkan.
   2. Jika model memerlukan perjanjian hukum, pilih kotak centang untuk mengonfirmasi bahwa Anda setuju.
   3. Jika Anda ingin menambahkan model lain, lakukan hal berikut.
      - Pilih Tambahkan model ke evaluasi. Ini membuka daftar model yang tersedia.
      - Pilih tombol radio di samping model yang ingin Anda tambahkan.
      - Pilih Tambah model.

c. Selanjutnya, pilih tipe Tugas. Anda dapat memilih salah satu dari hal berikut:

   - Ringkasan Teks
   - Menjawab Pertanyaan (T&J)
   - Klasifikasi teks
   - Generasi Terbuka

d. Di bagian Metrik evaluasi, pilih dimensi Evaluasi dan masukkan konteks tambahan tentang dimensi di kotak teks di bawah Deskripsi. Anda dapat memilih dari dimensi berikut:

   - Kefasihan — Mengukur kualitas linguistik dari teks yang dihasilkan.
   - Koherensi — Mengukur organisasi dan struktur teks yang dihasilkan.
   - Toksisitas — Mengukur bahaya teks yang dihasilkan.
   - Akurasi — Menunjukkan keakuratan teks yang dihasilkan.
   - Dimensi evaluasi khusus yang dapat Anda tentukan nama dan deskripsi untuk tim kerja Anda.

     Untuk menambahkan dimensi evaluasi khusus, lakukan hal berikut:

     - Pilih + Tambahkan dimensi evaluasi.
     - Di kotak teks yang berisi Menyediakan dimensi evaluasi, masukkan nama dimensi kustom Anda.
     - Di kotak teks yang berisi Berikan deskripsi untuk dimensi evaluasi ini, masukkan deskripsi sehingga tim kerja Anda memahami cara mengevaluasi dimensi kustom Anda.

Di bawah masing-masing metrik ini terdapat metrik pelaporan yang dapat Anda pilih dari panah bawah Pilih jenis metrik. Jika Anda memiliki dua model untuk dievaluasi, Anda dapat memilih metrik pelaporan komparatif atau individual. Jika Anda memiliki satu model untuk dievaluasi, Anda hanya dapat memilih metrik pelaporan individual. Anda dapat memilih jenis metrik pelaporan berikut untuk setiap metrik di atas.

- (Komparatif) Skala Likert - perbandingan - Evaluator manusia akan menunjukkan preferensi mereka antara dua tanggapan pada skala Likert 5 poin sesuai dengan instruksi Anda. Hasil dalam laporan akhir akan ditampilkan sebagai histogram peringkat kekuatan preferensi dari evaluator atas seluruh kumpulan data Anda. Tentukan poin-poin penting dari skala 5 poin dalam instruksi Anda sehingga evaluator Anda tahu cara menilai respons sesuai dengan harapan Anda.
- (Komparatif) Tombol pilihan — Memungkinkan evaluator manusia untuk menunjukkan satu respons pilihan mereka daripada respons lain. Evaluator menunjukkan preferensi mereka antara dua tanggapan sesuai dengan instruksi Anda menggunakan tombol radio. Hasil dalam laporan akhir akan ditampilkan sebagai persentase tanggapan yang disukai pekerja untuk setiap model. Jelaskan metode evaluasi Anda dengan jelas dalam instruksi Anda.
- (Komparatif) Peringkat Ordinal — Memungkinkan evaluator manusia untuk memberi peringkat tanggapan pilihan mereka ke prompt secara berurutan, mulai dari1, sesuai dengan instruksi Anda. Hasil dalam laporan akhir akan ditampilkan sebagai histogram peringkat dari evaluator di seluruh kumpulan data. Tentukan apa peringkat 1 berarti dalam instruksi Anda.
- (Individu) Jempol ke atas/bawah — Memungkinkan evaluator manusia menilai setiap respons dari model sebagai dapat diterima atau tidak dapat diterima sesuai dengan instruksi Anda. Hasil dalam laporan akhir akan ditampilkan sebagai persentase dari jumlah total peringkat oleh evaluator yang menerima peringkat jempol untuk setiap model. Anda dapat menggunakan metode penilaian ini untuk evaluasi satu atau lebih model. Jika Anda menggunakan ini dalam evaluasi yang berisi dua model, jempol ke atas atau ke bawah akan disajikan kepada tim kerja Anda untuk setiap respons model dan laporan akhir akan menunjukkan hasil agregat untuk setiap model secara individual. Tentukan apa yang dapat diterima sebagai peringkat jempol ke atas atau jempol ke bawah dalam instruksi Anda.
- (Individu) Skala Likert - individu - Memungkinkan evaluator manusia untuk menunjukkan seberapa kuat mereka menyetujui respons model berdasarkan instruksi Anda pada skala

Likert 5 poin. Hasil dalam laporan akhir akan ditampilkan sebagai histogram peringkat 5 poin dari evaluator di seluruh kumpulan data Anda. Anda dapat menggunakan skala ini untuk evaluasi yang berisi satu atau lebih model. Jika Anda memilih metode penilaian ini dalam evaluasi yang berisi lebih dari satu model, skala Likert 5 poin akan disajikan kepada tim kerja Anda untuk setiap respons model dan laporan akhir akan menunjukkan hasil agregat untuk setiap model secara individual. Tentukan poin-poin penting pada skala 5 poin dalam instruksi Anda sehingga evaluator Anda tahu bagaimana menilai respons sesuai dengan harapan Anda.

e. Pilih kumpulan data Prompt. Dataset ini diperlukan dan akan digunakan oleh tim kerja manusia Anda untuk mengevaluasi tanggapan dari model Anda. Berikan url ke bucket Amazon S3 yang berisi kumpulan data prompt Anda di kotak teks di bawah lokasi S3. Dataset Anda harus dalam `jsonlines` format dan berisi kunci berikut untuk mengidentifikasi bagian mana dari kumpulan data Anda yang akan digunakan UI untuk mengevaluasi model Anda:

- `prompt`— Permintaan yang Anda ingin model Anda menghasilkan respons.
- (Opsional) `category` — - Label kategori untuk prompt Anda. `category`Kuncinya digunakan untuk mengkategorikan petunjuk Anda sehingga Anda dapat memfilter hasil evaluasi Anda nanti berdasarkan kategori untuk pemahaman yang lebih dalam tentang hasil evaluasi. Itu tidak berpartisipasi dalam evaluasi itu sendiri, dan pekerja tidak melihatnya di UI evaluasi.
- (Opsional) `referenceResponse` — Jawaban referensi untuk evaluator manusia Anda. Jawaban referensi tidak dinilai oleh pekerja Anda, tetapi dapat digunakan untuk memahami tanggapan apa yang dapat diterima atau tidak dapat diterima, berdasarkan instruksi Anda.

  Contoh `jsonlines` kode berikut menunjukkan format kunci-nilai dari dataset evaluasi:

```
{
    "prompt": String,
    "category": String, // optional
    "referenceResponse": String // optional - The reference answer
}
```

  Sebuah dataset sampel berikut:

```
{"referenceResponse":"Cantal","category":"Capitals","prompt":"Aurillac is the
 capital of"}
```

```
{"referenceResponse":"Bamiyan
 Province","category":"Capitals","prompt":"Bamiyan city is the capital of"}
{"referenceResponse":"Oberspreewald-
Lausitz","category":"Capitals","prompt":"Senftenberg is the capital of"}
```

f. Masukkan lokasi bucket S3 tempat Anda ingin menyimpan hasil evaluasi keluaran di kotak teks di bawah Pilih lokasi S3 untuk menyimpan hasil evaluasi Anda. File output yang ditulis ke lokasi S3 ini akan dalam `jsonlines` format, diakhiri dengan ekstensi, `.jsonl`.

g. Konfigurasikan prosesor Anda di bagian Konfigurasi prosesor menggunakan parameter berikut:

- Gunakan hitungan Instance untuk menentukan jumlah instance komputasi yang akan digunakan untuk menjalankan model Anda. Jika Anda menggunakan lebih dari 1 instance, model Anda akan berjalan dalam instance paralel.
- Gunakan tipe Instance untuk memilih jenis instance komputasi yang ingin Anda gunakan untuk menjalankan model Anda. AWSmemiliki instance komputasi umum dan instance yang dioptimalkan untuk komputasi dan memori. Untuk informasi selengkapnya tentang tipe instans, lihatJenis Instans Studio Klasik yang Tersedia.
- Jika Anda SageMaker ingin menggunakan kunci enkripsi AWS Key Management Service (AWS KMS) Anda sendiri alih-alih kunci layanan AWS terkelola default, alihkan untuk memilih Aktif di bawah tombol Volume KMS, dan masukkan kunci. AWS KMS SageMaker akan menggunakan AWS KMS kunci Anda untuk mengenkripsi data pada volume penyimpanan. Untuk informasi selengkapnya tentang kunci, lihat AWS Key Management Service.
- Jika Anda SageMaker ingin menggunakan kunci enkripsi AWS Key Management Service (AWS KMS) Anda sendiri alih-alih kunci layanan AWS terkelola default, alihkan untuk memilih Aktif di bawah tombol Output KMS dan masukkan kunci. AWS KMS SageMaker akan menggunakan AWS KMS kunci Anda untuk mengenkripsi output pekerjaan pemrosesan.
- Gunakan peran IAM untuk menentukan akses dan izin untuk prosesor default. Masukkan peran IAM yang Anda atur di bagian Siapkan peran IAM Anda di bagian Jalankan evaluasi manusia ini.

h. Setelah Anda menentukan model dan kriteria Anda, pilih Berikutnya.

Tim kerja Anda terdiri dari orang-orang yang mengevaluasi model Anda. Setelah tim kerja Anda dibuat, itu bertahan tanpa batas waktu dan Anda tidak dapat mengubah atributnya. Berikut ini menunjukkan cara memulai tim kerja Anda.

Siapkan tim kerja Anda

1. Pilih tim yang ada atau Buat tim baru di kotak teks input tim Pilih.
2. Tentukan nama organisasi Anda dalam nama Organisasi. Bidang ini hanya muncul saat Anda membuat tim kerja pertama di akun.
3. Tentukan email kontak. Pekerja Anda akan menggunakan email ini untuk berkomunikasi dengan Anda tentang tugas evaluasi yang akan Anda berikan kepada mereka. Bidang ini hanya muncul saat Anda membuat tim kerja pertama di akun.
4. Tentukan nama Tim. Anda tidak dapat mengubah nama ini nanti.
5. Tentukan daftar alamat Email untuk setiap pekerja manusia Anda yang akan mengevaluasi model bahasa besar Anda (LLM). Ketika Anda menentukan alamat email untuk tim Anda, mereka akan diberitahu tentang pekerjaan baru hanya ketika mereka baru ditambahkan ke tim kerja. Jika Anda menggunakan tim yang sama untuk pekerjaan berikutnya, Anda harus memberi tahu mereka secara manual.

Berikan instruksi untuk tim kerja Anda

1. Berikan instruksi terperinci kepada tenaga kerja manusia Anda sehingga mereka dapat mengevaluasi model Anda sesuai dengan metrik dan standar Anda. Template di jendela utama menunjukkan instruksi sampel yang dapat Anda berikan. Untuk informasi selengkapnya tentang cara memberikan instruksi, lihat Membuat instruksi pekerja yang baik.
2. Untuk meminimalkan bias dalam evaluasi manusia Anda, pilih kotak di sebelah Mengacak posisi respons.
3. Pilih Selanjutnya.

Anda dapat meninjau ringkasan pilihan yang telah Anda buat untuk pekerjaan manusia Anda. Jika Anda harus mengubah pekerjaan Anda, pilih Sebelumnya untuk kembali ke pilihan sebelumnya.

Kirimkan permintaan pekerjaan evaluasi Anda dan lihat kemajuan pekerjaan

1. Untuk mengirimkan permintaan pekerjaan evaluasi Anda, pilih Buat sumber daya.

2. Untuk melihat status semua pekerjaan Anda, pilih Pekerjaan di panel navigasi. Kemudian, pilih Evaluasi model. Status evaluasi ditampilkan sebagai Selesai, Gagal, atau Sedang berlangsung.

   Berikut ini juga menampilkan:

   - Contoh notebook untuk menjalankan evaluasi model di SageMaker dan Amazon Bedrock.
   - Tautan ke informasi tambahan termasuk dokumentasi, video, berita, dan blog tentang proses evaluasi model.
3. Pilih evaluasi model Anda di bawah Nama untuk melihat ringkasan evaluasi Anda.

   - Ringkasan memberikan informasi tentang status pekerjaan, tugas evaluasi seperti apa yang Anda jalankan pada model mana, dan kapan dijalankan. Mengikuti ringkasan, skor evaluasi manusia diurutkan dan diringkas berdasarkan metrik.

Lihat hasil analisis manusia Anda

Output dari evaluasi model manusia Anda disimpan di lokasi S3 yang Anda tentukan untuk menyimpan hasil evaluasi keluaran sambil membuat pekerjaan evaluasi model.

Anda dapat melihat tugas evaluasi model saat masih berlangsung. Data keluaran muncul di bucket S3 saat tugas evaluasi selesai. Perlu waktu beberapa menit hingga data output muncul.

Isi `json` file keluaran Anda tergantung pada deskripsi pekerjaan, metrik, dan model spesifik Anda. Sampel keluaran berikut dihasilkan untuk single `prompt` in`inputRecord`, di mana manusia menilai respons model (`modelResponses`) sebagai tidak dapat diterima (`"result":false`):

```
{
    "output": [{
        "flowDefinitionArn": "arn:aws:sagemaker:us-west-2:111122223333:flow-definition/
flow-definition-name",
        "humanAnswers": [{
            "acceptanceTime": "2023-11-09T19:17:43.107Z",
            "answerContent": {
                "evaluationResults": {
                    "approvalRate": [{
                        "metric": "Relevance",
                        "modelResponseId": "0",
                        "result": false
                    }]
                }
            },
```

```
            "submissionTime": "2023-11-09T19:17:52.101Z",
            "timeSpentInSeconds": 8.994,
            "workerId": "444455556666",
            "workerMetadata": {
                "identityData": {
                    "identityProviderType": "Cognito",
                    "issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_111222",
                    "sub": "12345678-1234-1234-1234-"
                }
            }
        }],
        "humanLoopName": "1234567890abcdefghijklmnopqrstuv",
        "inputRecord": {
            "prompt": "What does vitamin C serum do for skin?",
            "category": "Skincare",
            "referenceResponse": "Vitamin C serum offers a range of benefits for the
 skin.
            Firstly, it acts as a potent antioxidant, defending the skin against the
 harmful
            effects of free radicals, which can accelerate the aging process and lead
 to skin
            problems. Moreover, vitamin C brightens the skin by reducing the appearance
 of
            dark spots, age spots, and hyperpigmentation. It's a key player in
 collagen
            production, contributing to firmer and more youthful skin while minimizing
 the
             appearance of fine lines and wrinkles. Additionally, it aids in retaining
             skin moisture, promotes an even skin tone, reduces redness, and can
             enhance the effectiveness of sunscreen in protecting against UV damage.
             Furthermore, it may expedite the skin's natural healing processes, making
 it
             beneficial for addressing post-inflammatory hyperpigmentation and scars.
 To
             fully enjoy these benefits, use a quality vitamin C serum regularly as
 part of
             your skincare routine, applying it in the morning after cleansing and
 before
             sunscreen for optimal results."
        },
        "modelResponses": [{
            "text": "\nVitamin C serums are widely known for their"
        }]
```

```
    }, {
        "flowDefinitionArn": "arn:aws:sagemaker:us-west-2:111122223333:flow-definition/
flow-definition-name"
        ...
    }],
```

Sampel keluaran sebelumnya menggunakan parameter berikut:

- `flowDefinitionArn`Amazon Resource Number (ARN) dari alur kerja tinjauan manusia (definisi alur) yang digunakan untuk membuat loop manusia.
- `humanAnswers`— Daftar `json` objek yang berisi tanggapan pekerja di`answerContent`.

  `metric:"Relevance"`— Nilai yang didasarkan pada jenis Metrik yang Anda pilih saat pekerjaan evaluasi model dibuat.
- `humanLoopName`— Nama lingkaran manusia.
- `inputRecord`— `json` Objek yang berisi prompt entri dari dataset input.
- `modelResponses`— Tanggapan individu dari model.
- `input`— Konten input yang dikirim ke SageMaker dalam permintaan ke`StartHumanLoop`.

Ketika analisis Anda selesai, Anda dapat melihat bagaimana kinerja model Anda terhadap kumpulan data yang Anda berikan menggunakan langkah-langkah berikut:

1. Dari panel navigasi Studio, pilih Pekerjaan, lalu pilih Evaluasi Model.
2. Di halaman Evaluasi Model, pekerjaan yang berhasil dikirimkan muncul dalam daftar. Daftar ini mencakup nama pekerjaan, status, nama model, jenis evaluasi, dan tanggal pembuatannya.
3. Jika evaluasi model Anda berhasil diselesaikan, Anda dapat mengklik nama pekerjaan untuk melihat ringkasan hasil evaluasi.
4. Untuk melihat laporan analisis manusia Anda, pilih nama pekerjaan yang ingin Anda periksa.

Bagian laporan analisis yang mengidentifikasi pekerjaan Anda berdasarkan nama, status, nama model, jenis evaluasi, dan tanggal pembuatannya muncul di bagian atas jendela utama setelah Anda memilih nama pekerjaan yang ingin Anda periksa.

Bagian laporan analisis berikutnya berisi jenis tugas yang dilakukan model Anda, dan hasil evaluasi untuk tugas itu.

Bagian laporan analisis berikut berisi hasil evaluasi untuk setiap model yang Anda evaluasi.

Bagian laporan analisis berikutnya menunjukkan konfigurasi pekerjaan evaluasi. Ini mencakup sumber daya yang digunakan, model yang dievaluasi, lokasi hasil evaluasi, dan dimensi evaluasi Anda.

Bagian terakhir berisi salinan instruksi yang Anda berikan kepada tenaga kerja Anda.

Gunakan evaluasi otomatis

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

Important

Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Anda dapat menggunakan evaluasi otomatis dengan menjalankannya di UI atau dengan menggunakan `fmeval` pustaka di dalam kode Anda sendiri. UI memandu Anda melalui alur kerja standar. Menggunakan pustaka memberikan kesempatan untuk menyesuaikan alur kerja Anda lebih lanjut. Bagian berikut menunjukkan cara menggunakan kedua jenis evaluasi otomatis.

Menggunakan evaluasi otomatis di UI

Anda dapat menjalankan evaluasi otomatis untuk SageMaker JumpStart model dalam Amazon SageMaker Studio UI. Alur kerja dalam UI memandu Anda dalam memilih model untuk evaluasi dan mengonfigurasi dimensi, metrik, dan sumber daya evaluasi. Untuk alur kerja yang lebih dapat disesuaikan, lihat. Sesuaikan alur kerja Anda menggunakan pustaka `fmeval` Bagian itu menunjukkan cara menjalankan evaluasi pada semua jenis model bahasa besar (LLM) menggunakan parameter spesifik untuk menghasilkan respons yang lebih disesuaikan. Bagian lain dalam panduan ini menunjukkan cara memformat kumpulan data input kustom opsional, mengatur lingkungan Anda, dan menjalankan evaluasi di UI.

Format dataset masukan Anda

Jika Anda menggunakan kumpulan data bawaan untuk mengevaluasi model Anda di UI, kumpulan data sudah terstruktur dalam format yang benar untuk dimasukkan ke dalam evaluasi model. Jika Anda menggunakan dataset kustom Anda sendiri, itu harus dalam `jsonlines` format. Setiap baris dalam kumpulan data `jsonlines` input Anda harus berupa `json` objek, berisi satu catatan dari kumpulan data Anda. Catatan ini digunakan untuk mengevaluasi model Anda.

Untuk mempelajari kunci mana yang tersedia untuk kumpulan data kustom di UI, lihat daftar tugas berikut.

- `model_input`— Diperlukan untuk menunjukkan input untuk tugas-tugas berikut.
  - Permintaan yang harus ditanggapi oleh model Anda dalam tugas generasi terbuka, toksisitas, dan akurasi.
  - Pertanyaan yang harus dijawab model Anda dalam menjawab pertanyaan, dan tugas pengetahuan faktual.
  - Teks yang harus diringkas oleh model Anda dalam tugas ringkasan teks.
  - Teks yang harus diklasifikasikan oleh model Anda dalam tugas klasifikasi.
  - Teks yang Anda ingin model Anda terganggu dalam tugas ketahanan semantik.
- `target_output`— Diperlukan untuk menunjukkan respons terhadap model Anda yang dievaluasi untuk tugas-tugas berikut.
  - Jawaban untuk menjawab pertanyaan, akurasi, ketahanan semantik, dan tugas evaluasi faktual.
  - Untuk akurasi, dan tugas ketahanan semantik, pisahkan jawaban yang dapat diterima dengan file. `<OR>` Evaluasi menerima salah satu jawaban yang dipisahkan oleh koma sebagai benar. Sebagai contoh, gunakan`target_output="UK<OR>England<OR>United Kingdom"`, jika Anda ingin menerima salah satu `UK` atau `England` atau `United Kingdom` sebagai jawaban yang dapat diterima.
- (Opsional) `category` - Menghasilkan skor evaluasi yang dilaporkan untuk setiap kategori.
- `sent_less_input`— Diperlukan untuk menunjukkan prompt yang mengandung lebih sedikit bias untuk tugas stereotip yang cepat.
- `sent_more_input`— Diperlukan untuk menunjukkan prompt yang berisi lebih banyak bias untuk tugas stereotip yang cepat.

Evaluasi pengetahuan faktual membutuhkan pertanyaan untuk diajukan dan jawaban untuk memeriksa respons model. Gunakan kunci `model_input` dengan nilai yang terkandung dalam

pertanyaan, dan kunci `target_output` dengan nilai yang terkandung dalam jawaban sebagai berikut:

```
{"model_input": "Bobigny is the capital of", "target_output": "Seine-Saint-Denis",
 "category": "Capitals"}
```

Contoh sebelumnya adalah satu baris file `jsonlines` input yang akan dikirim ke model Anda sebagai permintaan. Untuk membuat beberapa permintaan, sertakan beberapa baris. Contoh input data berikut adalah untuk tugas jawaban pertanyaan yang menggunakan `category` kunci opsional untuk evaluasi.

```
{"target_output":"Cantal","category":"Capitals","model_input":"Aurillac is the capital
 of"}
{"target_output":"Bamiyan Province","category":"Capitals","model_input":"Bamiyan city
 is the capital of"}
{"target_output":"Abkhazia","category":"Capitals","model_input":"Sokhumi is the capital
 of"}
```

Jika Anda mengevaluasi algoritme di UI, default berikut akan ditetapkan untuk kumpulan data input Anda:

- Jumlah catatan yang digunakan evaluasi adalah tetap. Algoritma mengambil sampel jumlah permintaan ini secara acak dari dataset input Anda.
  - Untuk mengubah nomor ini: Gunakan `fmeval` pustaka seperti yang dijelaskan dalam Sesuaikan alur kerja Anda menggunakan **`fmeval`** pustaka, dan atur parameter `num_records` ke jumlah sampel yang Anda inginkan, atau `-1` untuk menentukan seluruh kumpulan data. Jumlah default catatan yang dievaluasi adalah `100` untuk akurasi, stereotip cepat, toksisitas, klasifikasi, dan tugas ketahanan semantik. Jumlah default catatan untuk tugas pengetahuan faktual adalah`300`.
- Pembatas keluaran target seperti yang dijelaskan sebelumnya dalam `target_output` parameter diatur ke <OR> UI.
  - Untuk memisahkan jawaban yang dapat diterima menggunakan pembatas lain: Gunakan `fmeval` pustaka seperti yang dijelaskan dalam Sesuaikan alur kerja Anda menggunakan **`fmeval`** pustaka, dan atur parameter `target_output_delimiter` ke pembatas yang Anda inginkan.
- Anda harus menggunakan model SageMaker JumpStart bahasa berbasis teks yang tersedia untuk evaluasi model. Model-model ini memiliki beberapa parameter konfigurasi input data yang diteruskan secara otomatis ke dalam proses FMEval.

- Untuk menggunakan jenis model lain: Gunakan `fmeval` pustaka untuk menentukan konfigurasi data untuk dataset input Anda.

Siapkan lingkungan Anda

Untuk menjalankan evaluasi otomatis untuk model bahasa besar (LLM) Anda, Anda harus mengatur lingkungan Anda untuk memiliki izin yang benar untuk menjalankan evaluasi. Kemudian, Anda dapat menggunakan UI untuk memandu Anda melalui langkah-langkah dalam alur kerja, dan menjalankan evaluasi. Bagian berikut menunjukkan cara menggunakan UI untuk menjalankan evaluasi otomatis.

Prasyarat

- Untuk menjalankan evaluasi model di UI Studio, peran AWS Identity and Access Management (IAM) Anda dan kumpulan data input apa pun harus memiliki izin yang benar. Jika Anda tidak memiliki peran SageMaker Domain atau IAM, ikuti langkah-langkah dari atas Onboard dari konsol hingga semua Langkah 1. Pengaturan Umum.

Untuk mengatur izin bucket S3

Setelah domain dan peran Anda dibuat, gunakan langkah-langkah berikut untuk menambahkan izin yang diperlukan untuk mengevaluasi model Anda.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, masukkan **S3** ke bilah pencarian di bagian atas halaman.
3. Pilih S3 di bawah Layanan.
4. Pilih Bucket dari panel navigasi.
5. Di bagian Bucket tujuan umum, di bawah Nama, pilih nama bucket Amazon S3 yang ingin Anda gunakan untuk menyimpan input dan output model Anda di konsol. Jika Anda tidak memiliki bucket Amazon S3, lakukan hal berikut.
    1. Pilih Buat ember untuk membuka halaman Bucket Buat baru.
    2. Di bagian Konfigurasi umum, di bawah AWSWilayah, pilih AWS wilayah tempat model pondasi Anda berada.
    3. Beri nama bucket S3 Anda di kotak input di bawah nama Bucket.
    4. Terima semua opsi default.
    5. Pilih Buat ember.

6. Di bagian Bucket tujuan umum, di bawah Nama, pilih nama bucket S3 yang Anda buat.

6. Pilih tab Izin.
7. Gulir ke Cross-origin resource sharing (CORS) bagian di bagian bawah jendela. Pilih Edit.
8. Untuk menambahkan izin ke bucket Anda untuk evaluasi foundation, pastikan kode berikut muncul di kotak input. Anda juga dapat menyalin dan menempelkan hal berikut ke dalam kotak input.

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [
            "Access-Control-Allow-Origin"
        ]
    }
]
```

9. Pilih Simpan perubahan.

Untuk menambahkan izin ke kebijakan IAM

1. Di bilah pencarian di bagian atas halaman, masukkan**IAM**.
2. Di bawah Layanan, pilih Identity and Access Management (IAM).
3. Pilih Kebijakan dari panel navigasi.
4. Pilih Buat kebijakan. Saat editor Kebijakan terbuka, pilih JSON.
5. Pilih Berikutnya.
6. Pastikan izin berikut muncul di editor Kebijakan. Anda juga dapat menyalin dan menempelkan berikut ini ke editor Kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
             ],
                "Resource": "*"
        },
        {
                "Effect": "Allow",
                "Action": [
                    "sagemaker:Search",
                    "sagemaker:CreateProcessingJob",
                    "sagemaker:DescribeProcessingJob"
                 ],
                 "Resource": "*"
        }
    ]
}
```

7. Pilih Berikutnya.
8. Masukkan nama kebijakan di bagian Detail kebijakan, di bawah Nama kebijakan. Anda juga dapat memasukkan deskripsi opsional. Anda akan mencari nama kebijakan ini saat Anda menetapkannya ke peran.
9. Pilih Buat kebijakan.

Untuk menambahkan izin ke peran IAM Anda

1. Pilih Peran di panel navigasi. Masukkan nama peran yang ingin Anda gunakan.
2. Pilih nama peran di bawah Nama peran. Jendela utama berubah untuk menampilkan informasi tentang peran Anda.
3. Di bagian Kebijakan izin, pilih panah bawah di sebelah Tambahkan izin.
4. Dari opsi yang muncul, pilih Lampirkan kebijakan.
5. Dari daftar kebijakan yang muncul, cari kebijakan yang Anda buat di Langkah 5. Centang kotak di samping nama kebijakan Anda.
6. Pilih panah bawah di sebelah Tindakan.
7. Dari opsi yang muncul, pilih Lampirkan.
8. Cari nama peran yang Anda buat. Centang kotak di samping nama.
9. Pilih Tambahkan izin. Banner di bagian atas halaman harus menyatakan Kebijakan berhasil dilampirkan ke peran.

Mulai menggunakan Studio

1. Di bilah pencarian di bagian atas halaman, masukkan**`SageMaker`**.
2. Di bawah Layanan, pilih Amazon SageMaker.
3. Pilih Studio dari panel navigasi.
4. Pilih domain Anda dari bagian Memulai, setelah memperluas panah bawah di bawah Pilih Domain.
5. Pilih profil pengguna Anda dari bagian Memulai setelah memperluas panah bawah di bawah Pilih profil pengguna.
6. Pilih Open Studio untuk membuka landing page Studio.

Jalankan evaluasi otomatis di UI

Anda dapat menjalankan evaluasi manusia pada model berbasis teks yang telah Anda sesuaikan SageMaker JumpStart jika tersedia untuk evaluasi. Atau, Anda dapat memulai dari model dasar berbasis teks dari halaman SageMaker JumpStart arahan di Studio, yang mencantumkan model yang telah diterapkan sebelumnya.

Untuk meluncurkan SageMaker JumpStart

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di bilah pencarian di bagian atas halaman, masukkan**`SageMaker`**.
3. Di bawah Layanan, pilih Amazon SageMaker.
4. Pilih Studio dari panel navigasi.
5. Pilih domain Anda dari bagian Memulai, setelah memperluas panah bawah di bawah Pilih Domain.
6. Pilih profil pengguna Anda dari bagian Memulai setelah memperluas panah bawah di bawah Pilih profil pengguna.
7. Pilih Open Studio untuk membuka landing page Studio.
8. Pilih SageMaker JumpStartdari panel navigasi.

Untuk mengatur tugas evaluasi

1. Pilih model berbasis teks SageMaker JumpStart . Anda juga dapat menggunakan bilah pencarian dan filter untuk memilih jenis tugas terkait teks berikut.
   - Ringkasan Teks
   - Menjawab Pertanyaan (T&J)
   - Klasifikasi Teks
   - Generasi Terbuka
2. Pilih tombol Evaluasi.
   1. Tombol terletak di sudut kanan atas jendela utama di sebelah kanan Deploy.
3. Tentukan detail pekerjaan.
   a. Masukkan Nama evaluasi model Anda. Nama ini membantu Anda mengidentifikasi pekerjaan evaluasi model Anda setelah diserahkan.
   b. Masukkan Deskripsi untuk menambahkan lebih banyak konteks ke nama.
   c. Pilih Berikutnya.
4. Menyelesaikan evaluasi
   a. Di bawah Pilih jenis evaluasi, pilih tombol radio di sebelah Otomatis.

b. Pilih model yang akan dievaluasi. Anda hanya dapat mengevaluasi satu model untuk setiap evaluasi.

   1. Jika Anda memilih model dari SageMaker JumpStart, model yang Anda pilih sudah ditampilkan.
   2. Jika model memerlukan perjanjian hukum, pilih kotak centang untuk mengonfirmasi bahwa Anda setuju.

c. Selanjutnya, pilih tipe Tugas. Anda dapat memilih salah satu dari hal berikut:

   - Ringkasan Teks
   - Menjawab Pertanyaan (T&J)
   - Klasifikasi teks
   - Generasi Terbuka

   Untuk informasi selengkapnya tentang tugas dan dimensi ini, lihat Evaluasi otomatis di Ikhtisar evaluasi model pondasi.

d. Di bagian Metrik evaluasi, pilih dimensi Evaluasi. Kotak teks di bawah Deskripsi berisi konteks tambahan tentang dimensi.

   Setelah Anda memilih tugas, metrik yang terkait dengan tugas akan muncul di bawah Metrik. Di bagian ini, lakukan hal berikut.

e. Pilih dimensi evaluasi dari panah bawah di bawah Dimensi evaluasi.

f. Pilih dataset evaluasi. Anda dapat memilih untuk menggunakan dataset Anda sendiri atau menggunakan dataset bawaan. Jika Anda ingin menggunakan kumpulan data Anda sendiri untuk mengevaluasi model, itu harus diformat dengan cara yang dapat digunakan FMEval. Itu juga harus ditempatkan di bucket S3 yang memiliki izin CORS yang direferensikan di bagian sebelumnya. Siapkan lingkungan Anda Untuk informasi selengkapnya tentang cara memformat kumpulan data kustom lihatMenggunakan dataset masukan kustom.

g. Masukkan lokasi bucket S3 tempat Anda ingin menyimpan hasil evaluasi keluaran. File ini dalam format jsonlines (.jsonl).

h. Konfigurasikan prosesor Anda di bagian Konfigurasi prosesor menggunakan parameter berikut:

- Gunakan hitungan Instance untuk menentukan jumlah instance komputasi yang ingin Anda gunakan untuk menjalankan model Anda. Jika Anda menggunakan lebih dari 1 instance, model Anda dijalankan dalam instance paralel.
- Gunakan tipe Instance untuk memilih jenis instance komputasi yang ingin Anda gunakan untuk menjalankan model Anda. Untuk informasi selengkapnya tentang tipe instans, lihatJenis Instans Studio Klasik yang Tersedia.
- Gunakan tombol Volume KMS untuk menentukan kunci enkripsi AWS Key Management Service (AWS KMS) Anda. SageMaker menggunakan AWS KMS kunci Anda untuk mengenkripsi lalu lintas masuk dari model dan bucket Amazon S3 Anda. Untuk informasi selengkapnya tentang kunci, lihat AWS Key Management Service.
- Gunakan tombol Output KMS untuk menentukan kunci AWS KMS enkripsi Anda untuk lalu lintas keluar.
- Gunakan Peran IAM untuk menentukan akses dan izin untuk prosesor default. Masukkan peran IAM yang Anda atur Siapkan lingkungan Anda

i. Setelah Anda menentukan model dan kriteria Anda, pilih Berikutnya. Jendela utama melompat ke Langkah 5 Tinjau dan Simpan.

Tinjau dan jalankan pekerjaan evaluasi Anda

1. Tinjau semua parameter, model, dan data yang Anda pilih untuk evaluasi Anda.
2. Pilih Buat sumber daya untuk menjalankan evaluasi Anda.
3. Untuk memeriksa status pekerjaan Anda, buka bagian atas Evaluasi Model di halaman.

Lihat hasil analisis dari evaluasi otomatis Anda

Bagian ini mencantumkan tiga output untuk evaluasi otomatis yang dijalankan di UI.

1. `output.json`File berisi skor agregat untuk kumpulan data Anda. Contoh `json` output berikut.

```
{
  "evaluations": [
    {
      "evaluation_name": "factual_knowledge",
      "dataset_name": "trex",
```

```
      "prompt_template": "<s>[INST] <<SYS>>Answer the question at the end in as
 few words as possible. Do not repeat the question. Do not answer in complete
 sentences.<</SYS> Question: $feature [/INST]",
      "dataset_scores": [
        {
          "name": "factual_knowledge",
          "value": 0.2966666666666667
        }
      ],
      "category_scores": [
        {
          "name": "Author",
          "scores": [
            {
              "name": "factual_knowledge",
              "value": 0.4117647058823529
            }
          ]
        },
        ...
        {
          "name": "Capitals",
          "scores": [
            {
              "name": "factual_knowledge",
              "value": 0.2857142857142857
            }
          ]
        }
      ]
    }
  ]
}
```

Dalam contoh keluaran sebelumnya, model mencetak rata-rata`0.2966666666666667`. Skor rata-rata untuk setiap kategori terdaftar mengikuti skor agregat.

2. Satu *`evaluation_name _ dataset_name`* .jsonl file berisi hasil instance-wise untuk setiap permintaan jsonlines. Jika Anda memiliki `300` permintaan dalam data input jsonlines Anda, file keluaran jsonlines ini berisi tanggapan. `300` File output berisi permintaan yang dibuat untuk model Anda diikuti dengan skor untuk evaluasi itu. Contoh keluaran instance-wide berikut.

3. Laporan Evaluasi yang berisi hasil evaluasi model yayasan Anda. Isi laporan evaluasi tergantung pada jenis tugas yang Anda gunakan untuk mengevaluasi model Anda. Setiap laporan berisi bagian-bagian berikut:
   a. Skor keseluruhan untuk setiap evaluasi yang berhasil di bawah tugas evaluasi. Sebagai contoh satu evaluasi dengan satu kumpulan data, jika Anda mengevaluasi model Anda untuk tugas klasifikasi untuk Akurasi dan Kekokohan Semantik, maka tabel yang merangkum hasil evaluasi untuk Akurasi dan Akurasi Kekokohan Semantik muncul di bagian atas laporan Anda. Evaluasi lain dengan kumpulan data lain mungkin terstruktur secara berbeda.
   b. Konfigurasi untuk pekerjaan evaluasi Anda termasuk nama model, jenis, metode evaluasi mana yang digunakan, dan kumpulan data apa yang dievaluasi terhadap model Anda.
   c. Bagian Hasil Evaluasi Terperinci yang merangkum algoritme evaluasi, memberikan informasi tentang dan menautkan ke kumpulan data bawaan apa pun, bagaimana skor dihitung, dan tabel yang menunjukkan beberapa data sampel dengan skor terkait.
   d. Bagian Evaluasi Gagal yang berisi daftar evaluasi yang tidak lengkap. Jika tidak ada evaluasi yang gagal, bagian laporan ini dihilangkan.

Gunakan **`fmeval`** pustaka untuk menjalankan evaluasi otomatis

Menggunakan `fmeval` pustaka dalam kode Anda sendiri memberi Anda fleksibilitas paling besar untuk menyesuaikan alur kerja Anda. Anda dapat menggunakan `fmeval` perpustakaan untuk mengevaluasi LLM apa pun, dan juga memiliki lebih banyak fleksibilitas dengan kumpulan data input kustom Anda. Langkah-langkah berikut menunjukkan cara mengatur lingkungan Anda dan cara menjalankan alur kerja awal dan disesuaikan menggunakan `fmeval` pustaka.

Memulai menggunakan **`fmeval`** perpustakaan

Anda dapat mengonfigurasi evaluasi model foundation dan menyesuaikannya untuk kasus penggunaan di notebook Studio. Konfigurasi Anda bergantung pada jenis tugas yang dibuat untuk diprediksi oleh model fondasi Anda, dan bagaimana Anda ingin mengevaluasinya. FMEval mendukung generasi terbuka, ringkasan teks, menjawab pertanyaan, dan tugas klasifikasi. Langkah-langkah di bagian ini menunjukkan cara mengatur alur kerja awal. Alur kerja awal ini mencakup pengaturan lingkungan Anda dan menjalankan algoritme evaluasi menggunakan model fondasi Amazon Bedrock SageMaker JumpStart atau Amazon dengan kumpulan data bawaan. Jika Anda harus menggunakan dataset input kustom dan alur kerja untuk kasus penggunaan yang lebih spesifik, lihat. Sesuaikan alur kerja Anda menggunakan pustaka `fmeval`

Siapkan lingkungan Anda

Jika Anda tidak ingin menjalankan evaluasi model di buku catatan Studio, lewati ke langkah 11 di bagian Mulai menggunakan Studio berikut.

Prasyarat

- Untuk menjalankan evaluasi model di UI Studio, peran AWS Identity and Access Management (IAM) Anda dan kumpulan data input apa pun harus memiliki izin yang benar. Jika Anda tidak memiliki peran SageMaker Domain atau IAM, ikuti langkah-langkah dari atas Onboard dari konsol hingga semua Langkah 1. Pengaturan Umum.

Mengatur izin bucket Amazon S3

Setelah domain dan peran Anda dibuat, gunakan langkah-langkah berikut untuk menambahkan izin yang diperlukan untuk mengevaluasi model Anda.

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, masukkan **S3** ke bilah pencarian di bagian atas halaman.
3. Pilih S3 di bawah Layanan.
4. Pilih Bucket dari panel navigasi.
5. Di bagian Bucket tujuan umum, di bawah Nama, pilih nama bucket S3 yang ingin Anda gunakan untuk menyimpan input dan output model Anda di konsol. Jika Anda tidak memiliki bucket S3, lakukan hal berikut:
    1. Pilih Buat ember untuk membuka halaman Bucket Buat baru.
    2. Di bagian Konfigurasi umum, di bawah AWSWilayah, pilih AWS wilayah tempat model pondasi Anda berada.
    3. Beri nama bucket S3 Anda di kotak input di bawah nama Bucket.
    4. Terima semua opsi default.
    5. Pilih Buat ember.
    6. Di bagian Bucket tujuan umum, di bawah Nama, pilih nama bucket S3 yang Anda buat.
6. Pilih tab Izin.
7. Gulir ke Cross-origin resource sharing (CORS) bagian di bagian bawah jendela. Pilih Edit.

8. Untuk menambahkan izin ke bucket Anda untuk evaluasi foundation, pastikan kode berikut muncul di kotak input. Anda juga dapat menyalin dan menempelkan hal berikut ke dalam kotak input.

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [
            "Access-Control-Allow-Origin"
        ]
    }
]
```

9. Pilih Simpan perubahan.

Untuk menambahkan izin ke kebijakan IAM

1. Di bilah pencarian di bagian atas halaman, masukkan**IAM**.
2. Di bawah Layanan, pilih Identity and Access Management (IAM).
3. Pilih Kebijakan dari panel navigasi.
4. Masukkan AmazonSageMakerFullAccesske dalam bilah pencarian. Pilih tombol radio di samping kebijakan yang muncul. Tombol Actions sekarang dapat dipilih.
5. Pilih panah bawah di sebelah Tindakan. Dua opsi muncul.
6. Pilih Lampirkan.
7. Dalam daftar IAM yang muncul, cari nama peran yang Anda buat. Centang kotak di samping nama.
8. Pilih Lampirkan kebijakan.

Mulai menggunakan Studio

1. Di bilah pencarian di bagian atas halaman, masukkan**SageMaker**.
2. Di bawah Layanan, pilih Amazon SageMaker.
3. Pilih Studio dari panel navigasi.
4. Pilih domain Anda dari bagian Memulai, setelah memperluas panah bawah di bawah Pilih Domain.
5. Pilih profil pengguna Anda dari bagian Memulai setelah memperluas panah bawah di bawah Pilih profil pengguna.
6. Pilih Open Studio untuk membuka landing page Studio.
7. Pilih browser file dari panel navigasi dan arahkan ke direktori root.
8. Pilih Buat buku catatan.
9. Di kotak dialog lingkungan notebook yang terbuka, pilih gambar Data Science 3.0.
10. Pilih Pilih.
11. Instal `fmeval` paket di lingkungan pengembangan Anda, seperti yang ditunjukkan dalam contoh kode berikut:

```
!pip install fmeval
```

> **Note**
>
> Instal `fmeval` perpustakaan ke lingkungan yang menggunakanPython 3.10. Untuk informasi selengkapnya tentang persyaratan yang diperlukan untuk menjalankan`fmeval`, lihat `fmeval`dependensi.

Konfigurasi **`ModelRunner`**

FMEval menggunakan pembungkus tingkat tinggi yang dipanggil `ModelRunner` untuk menyusun input, memanggil, dan mengekstrak output dari model Anda. `fmeval`Paket dapat mengevaluasi LLM apa pun, namun prosedur untuk mengkonfigurasi `ModelRunner` tergantung pada jenis model yang ingin Anda evaluasi. Bagian ini menjelaskan cara mengkonfigurasi `ModelRunner` untuk model SageMaker JumpStart atau Amazon Bedrock. Jika Anda ingin menggunakan dataset input kustom dan kustom`ModelRunner`, lihatSesuaikan alur kerja Anda menggunakan pustaka `fmeval`.

Gunakan SageMaker JumpStart model

Untuk digunakan `ModelRunner` untuk mengevaluasi SageMaker JumpStart model, membuat atau menyediakan titik akhir, tentukan model dan dataset bawaan, konfigurasi, dan uji. `ModelRunner`

Tentukan SageMaker JumpStart model dan konfigurasikan ModelRunner

1. Berikan titik akhir dengan melakukan salah satu dari hal berikut:

   - Tentukan EndpointNameke SageMaker JumpStart titik akhir yang ada,`model_id`, dan`model_version`.
   - Tentukan `model_id` dan `model_version` untuk model Anda, dan buat SageMaker JumpStart titik akhir.

   Contoh kode berikut menunjukkan cara membuat endpoint untuk Llama 2 foundation modelyang tersedia melalui SageMaker JumpStart.

   ```
   import sagemaker
   from sagemaker.jumpstart.model import JumpStartModel

   #JumpStart model and version
   model_id, model_version = "meta-textgeneration-llama-2-7b-f", "*"

   my_model = JumpStartModel(model_id=model_id)
   predictor = my_model.deploy()
   endpoint_name = predictor.endpoint_name

   # Accept the EULA, and test the endpoint to make sure it can predict.
   predictor.predict({"inputs": [[{"role":"user", "content": "Hello how are you?"}]]},
    custom_attributes='accept_eula=true')
   ```

   Contoh kode sebelumnya mengacu pada EULA, yang merupakan singkatan dari end-use-license-agreement (EULA). EULA dapat ditemukan dalam deskripsi kartu model model yang Anda gunakan. Untuk menggunakan beberapa SageMaker JumpStart model, Anda harus menentukan`accept_eula=true`, seperti yang ditunjukkan pada panggilan sebelumnya ke`predict`. Untuk informasi selengkapnya tentang EULA, lihat bagian Lisensi dan sumber model di. Pilih model pondasi

   Anda dapat menemukan daftar SageMaker JumpStart model yang tersedia di Algoritma Bawaan dengan Tabel Model yang telah dilatih sebelumnya.

2. Konfigurasikan `ModelRunner` dengan menggunakan`JumpStartModelRunner`, seperti yang ditunjukkan pada contoh konfigurasi berikut:

```
from fmeval.model_runners.sm_jumpstart_model_runner import JumpStartModelRunner

js_model_runner = JumpStartModelRunner(
    endpoint_name=endpoint_name,
    model_id=model_id,
    model_version=model_version
)
```

   Dalam contoh konfigurasi sebelumnya, gunakan nilai yang sama untuk`endpoint_name`,`model_id`, dan `model_version` yang Anda gunakan untuk membuat titik akhir.

3. Uji Anda`ModelRunner`. Kirim permintaan sampel ke model Anda seperti yang ditunjukkan dalam contoh kode berikut:

```
js_model_runner.predict("What is the capital of London")
```

Gunakan model Amazon Bedrock

Untuk mengevaluasi model Amazon Bedrock, Anda harus menentukan model dan kumpulan data bawaan, dan mengonfigurasinya. `ModelRunner`

Tentukan model Amazon Bedrock dan konfigurasikan ModelRunner

1. Untuk menentukan dan mencetak detail model, gunakan contoh kode berikut untuk model Titan yang tersedia melalui Amazon Bedrock:

```
import boto3
import json
bedrock = boto3.client(service_name='bedrock')
bedrock_runtime = boto3.client(service_name='bedrock-runtime')

model_id = "amazon.titan-tg1-large"
accept = "application/json"
content_type = "application/json"

print(bedrock.get_foundation_model(modelIdentifier=modelId).get('modelDetails'))
```

Dalam contoh kode sebelumnya, `accept` parameter menentukan format data yang ingin Anda gunakan untuk mengevaluasi LLM Anda. `contentType`Menentukan format data input dalam permintaan. Hanya `MIME_TYPE_JSON` didukung untuk `accept` dan `contentType` untuk model Amazon Bedrock. Untuk informasi selengkapnya tentang parameter ini, lihat InvokeModelWithResponseStream.

2. Untuk mengkonfigurasi`ModelRunner`, gunakan`BedrockModelRunner`, seperti yang ditunjukkan dalam contoh konfigurasi berikut:

```
from fmeval.model_runners.bedrock_model_runner import BedrockModelRunner

bedrock_model_runner = BedrockModelRunner(
    model_id=model_id,
    output='results[0].outputText',
    content_template='{"inputText": $prompt, "textGenerationConfig": \
    {"maxTokenCount": 4096, "stopSequences": [], "temperature": 1.0, "topP":
 1.0}}',
)
```

Parametrize `ModelRunner` konfigurasi sebagai berikut.

- Gunakan nilai yang sama untuk `model_id` yang Anda gunakan untuk menyebarkan model.
- Gunakan `output` untuk menentukan format `json` respons yang dihasilkan. Sebagai contoh, jika LLM Anda memberikan respons`[{"results": "this is the output"}]`, maka `output='results[0].outputText'` kembali`this is the output`.
- Gunakan `content_template` untuk menentukan bagaimana LLM Anda berinteraksi dengan permintaan. Template konfigurasi berikut dirinci semata-mata untuk menjelaskan contoh konfigurasi sebelumnya, dan itu tidak diperlukan.
  - Dalam contoh konfigurasi sebelumnya, variabel `inputText` menentukan prompt, yang menangkap permintaan yang dibuat oleh pengguna.
  - Variabel `textGenerationConfig` menentukan bagaimana LLM menghasilkan tanggapan sebagai berikut:
    - Parameter `maxTokenCount` ini digunakan untuk membatasi panjang respons dengan membatasi jumlah token yang dikembalikan oleh LLM.
    - Parameter `stopSequences` ini digunakan untuk menentukan daftar urutan karakter yang memberi tahu LLM Anda untuk berhenti menghasilkan respons. Output model dihentikan saat pertama kali salah satu string yang terdaftar ditemui dalam output. Sebagai contoh,

Anda dapat menggunakan urutan pengembalian carriage untuk membatasi respons model ke satu baris.

- Parameter `topP` mengontrol keacakan dengan membatasi kumpulan token untuk dipertimbangkan saat membuat token berikutnya. Parameter ini menerima nilai antara `0.0` dan`1.0`. Nilai yang lebih tinggi dari `topP` memungkinkan untuk satu set yang berisi kosakata yang lebih luas dan nilai yang lebih rendah membatasi kumpulan token ke kata-kata yang lebih mungkin.
- Parameter `temperature` mengontrol keacakan teks yang dihasilkan, dan menerima nilai positif. Nilai yang lebih tinggi dari `temperature` menginstruksikan model untuk menghasilkan respons yang lebih acak dan beragam. Nilai yang lebih rendah menghasilkan respons yang lebih dapat diprediksi. Rentang khas untuk `temperature` kebohongan antara `0.2` dan`2.0`.

Untuk informasi selengkapnya tentang parameter untuk model pondasi Amazon Bedrock tertentu, lihat Parameter inferensi untuk model pondasi.

Format parameter content_template tergantung pada input dan parameter yang didukung oleh LLM Anda. Misalnya, Anthropic's Claude 2model dapat mendukung yang berikut`content_template`:

```
"content_template": "{\"prompt\": $prompt, \"max_tokens_to_sample\": 500}"
```

Sebagai contoh lain, model Falcon 7b dapat mendukung yang berikut ini. `content_template`

```
"content_template": "{\"inputs\": $prompt, \"parameters\":{\"max_new_tokens\": \
    10, \"top_p\": 0.9, \"temperature\": 0.8}}"
```

Terakhir, uji Anda`ModelRunner`. Kirim permintaan sampel ke model Anda seperti yang ditunjukkan dalam contoh kode berikut:

```
bedrock_model_runner.predict("What is the capital of London?")
```

Evaluasi model Anda

Setelah Anda mengkonfigurasi data Anda dan`ModelRunner`, Anda dapat menjalankan algoritma evaluasi pada tanggapan yang dihasilkan oleh LLM Anda. Untuk melihat daftar semua algoritma evaluasi yang tersedia, jalankan kode berikut:

```
from fmeval.eval_algo_mapping import EVAL_ALGORITHMS
print(EVAL_ALGORITHMS.keys())
```

Setiap algoritma memiliki evaluasi dan `evaluate_sample` metode. `evaluate`Metode ini menghitung skor untuk seluruh kumpulan data. `evaluate_sample`Metode ini mengevaluasi skor untuk satu contoh.

`evaluate_sample`Metode mengembalikan `EvalScore` objek. `EvalScore`objek berisi skor agregat tentang seberapa baik kinerja model Anda selama evaluasi. `evaluate_sample`Metode ini memiliki parameter opsional berikut:

- `model_output`— Respons model untuk satu permintaan.
- `model_input`— Prompt yang berisi permintaan ke model Anda.
- `target_output`— Respon yang diharapkan dari prompt yang terkandung dalam`model_input`.

Contoh kode berikut ini adalah cara menggunakan`evaluate_sample`:

```
#Evaluate your custom sample
model_output = model_runner.predict("London is the capital of?")[0]
eval_algo.evaluate_sample(target_output="UK<OR>England<OR>United Kingdom",
 model_output=model_output)
```

`evaluate`Metode ini memiliki parameter opsional berikut:

- `model`— Contoh `ModelRunner` menggunakan model yang ingin Anda evaluasi.
- `dataset_config`— Konfigurasi dataset. Jika tidak `dataset_config` disediakan, model dievaluasi menggunakan semua kumpulan data bawaan yang dikonfigurasi untuk tugas ini.
- `prompt_template`— Template yang digunakan untuk menghasilkan prompt. Jika tidak `prompt_template` disediakan, model Anda dievaluasi menggunakan templat prompt default.
- `save`— Jika disetel ke`True`, respons cepat dan skor yang direkam disimpan ke file. `EvalAlgorithmInterface.EVAL_RESULTS_PATH` Default ke `False`.

- `num_records`— Jumlah catatan yang diambil sampel secara acak dari dataset input untuk evaluasi. Default ke `300`.

`evaluate`Algoritma mengembalikan daftar `EvalOutput` objek yang dapat mencakup yang berikut:

- `eval_name`— Nama algoritme Evaluasi.

  `dataset_name`— Nama dataset yang digunakan oleh algoritma evaluasi.

  `prompt_template`— Template yang digunakan untuk menulis prompt yang digunakan jika parameter tidak `model_output` disediakan dalam kumpulan data. Untuk informasi selengkapnya, lihat `prompt_template` di SageMaker JumpStart **`ModelRunner`**bagian Mengkonfigurasi a.

  `dataset_scores`— Skor agregat dihitung di seluruh kumpulan data.

  `category_scores`— Daftar `CategoryScore` objek yang berisi skor untuk setiap kategori dalam kumpulan data.

  `output_path`— Jalur lokal ke output evaluasi. Output ini berisi tanggapan cepat dengan skor evaluasi rekor.

  `error`— Pesan kesalahan string untuk pekerjaan evaluasi yang gagal.

Dimensi berikut tersedia untuk evaluasi model:

- Akurasi
- Pengetahuan faktual
- Stereotip cepat
- Kekokohan semantik
- Toksisitas

Akurasi

Anda dapat menjalankan algoritma akurasi untuk menjawab pertanyaan, ringkasan teks, atau tugas klasifikasi. Algoritma berbeda untuk setiap tugas untuk mengakomodasi berbagai jenis input data dan masalah sebagai berikut:

- Untuk tugas menjawab pertanyaan, jalankan `QAAccuracy` algoritma dengan `QAAccuracyConfig` file.
- Untuk tugas ringkasan teks, jalankan `SummarizationAccuracy` algoritme dengan `SummarizationAccuracyConfig` file.
- Untuk tugas klasifikasi, jalankan `ClassificationAccuracy` algoritme dengan file`ClassificationAccuracyConfig`.

`QAAccuracy`Algoritma mengembalikan daftar `EvalOutput` objek yang berisi satu skor akurasi untuk setiap sampel. Untuk menjalankan algoritma akurasi jawaban pertanyaan, buat instance a `QAAccuracygeConfig` dan teruskan salah satu `<OR>` atau `None` sebagai. `target_output_delimiter` Algoritma akurasi jawaban pertanyaan membandingkan respons yang dihasilkan model Anda dengan respons yang diketahui. Jika Anda masuk `<OR>` sebagai pembatas target, maka algoritme menilai respons sebagai benar jika menghasilkan konten yang dipisahkan oleh `<OR>` dalam jawabannya. Jika Anda lulus `None` atau string kosong sebagai`target_output_delimiter`, kode melempar kesalahan.

Panggil `evaluate` metode dan berikan parameter yang Anda inginkan seperti yang ditunjukkan dalam contoh kode berikut:

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.qa_accuracy import QAAccuracy, QAAccuracyConfig

eval_algo = QAAccuracy(QAAccuracyConfig(target_output_delimiter="<OR>")))
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

`SummarizationAccuracy`Algoritma mengembalikan daftar `EvalOutput` objek yang berisi skor untuk ROUGE-N, Meteor, dan BERTScore. Untuk informasi selengkapnya tentang skor ini, lihat bagian Ringkasan teks dilkhtisar evaluasi model pondasi. Untuk menjalankan algoritma akurasi ringkasan teks, buat instance a `SummarizationAccuracyConfig` dan teruskan berikut ini:

- Tentukan jenis ROUGEmetrik yang ingin Anda gunakan dalam evaluasi Anda`rouge_type`. Anda dapat memilih `rouge1`, `rouge2`, atau `rougeL`. Metrik ini membandingkan ringkasan yang dihasilkan dengan ringkasan referensi. ROUGE-1membandingkan ringkasan yang dihasilkan dan ringkasan referensi menggunakan unigram yang tumpang tindih (urutan satu item seperti “the”, “is”). ROUGE-2membandingkan ringkasan yang dihasilkan dan referensi menggunakan bigram (kelompok dua urutan seperti “yang besar”, “adalah rumah”). ROUGE-Lmembandingkan urutan

kata pencocokan terpanjang. Untuk informasi selengkapnyaROUGE, lihat ROUGE: A Package for Automatic Evaluation of Summaries.

- Setel `use_stemmer_for_rouge` ke `True` atau`False`. Stemmer menghapus imbuhan dari kata-kata sebelum membandingkannya. Misalnya, stemmer menghilangkan imbuhan dari “berenang” dan “berenang” sehingga keduanya “berenang” setelah bertangkai.
- Setel model_type_for_bertscore ke model yang ingin Anda gunakan untuk menghitung a. BERTScore Anda dapat memilih ROBERTA_MODEL atau MICROSOFT_DEBERTA_MODEL yang lebih canggih.

Terakhir, panggil `evaluate` metode dan berikan parameter yang Anda inginkan seperti yang ditunjukkan pada contoh kode berikut:

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.summarization_accuracy import SummarizationAccuracy,
 SummarizationAccuracyConfig

eval_algo =
 SummarizationAccuracy(SummarizationAccuracyConfig(rouge_type="rouge1",model_type_for_bertscore
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

`ClassificationAccuracy`Algoritma mengembalikan daftar `EvalOutput` objek yang berisi akurasi klasifikasi, presisi, ingatan, dan skor akurasi seimbang untuk setiap sampel. Untuk informasi selengkapnya tentang skor ini, lihat bagian Klasifikasi dilkhtisar evaluasi model pondasi. Untuk menjalankan algoritma akurasi klasifikasi, buat instance a `ClassificationAccuracyConfig` dan teruskan strategi rata-rata ke. `multiclass_average_strategy` Anda dapat memilih`micro`,`macro`,`samples`,`weighted`, ataubinary. Nilai default-nya adalah `micro`. Kemudian, masukkan daftar yang berisi nama kolom yang berisi label sebenarnya untuk kategori klasifikasi Anda ke valid_labels. Terakhir, panggil `evaluate` metode dan berikan parameter yang Anda inginkan seperti yang ditunjukkan pada contoh kode berikut:

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.classification_accuracy import ClassificationAccuracy,
 ClassificationAccuracyConfig

eval_algo =
 ClassificationAccuracy(ClassificationAccuracyConfig(multiclass_average_strategy="samples",vali
```

```
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

Pengetahuan faktual

Anda dapat menjalankan algoritma pengetahuan faktual untuk generasi terbuka. Untuk menjalankan algoritma pengetahuan faktual, buat instance a `FactualKnowledgeConfig` dan secara opsional meneruskan string pembatas (secara default, ini adalah). <OR> Algoritma pengetahuan faktual membandingkan respons yang dihasilkan model Anda dengan respons yang diketahui. Algoritma menilai respons sebagai benar jika menghasilkan konten yang dipisahkan oleh pembatas dalam jawaban. Jika Anda lulus `None` sebagai`target_output_delimiter`, maka model harus menghasilkan respons yang sama dengan jawaban yang akan dinilai sebagai benar. Terakhir, panggil `evaluate` metode dan berikan parameter yang Anda inginkan.

Pengetahuan faktual mengembalikan daftar `EvalScore` objek. Ini berisi skor agregat tentang seberapa baik model Anda mampu menyandikan pengetahuan faktual seperti yang dijelaskan di bagian ikhtisar evaluasi model Foundation. Skor berkisar antara `0` dan `1` dengan skor terendah sesuai dengan pengetahuan yang lebih rendah tentang fakta dunia nyata.

Contoh kode berikut ini menunjukkan cara mengevaluasi LLM Anda menggunakan algoritma pengetahuan faktual:

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.factual_knowledge import FactualKnowledge,
 FactualKnowledgeConfig

eval_algo = FactualKnowledge(FactualKnowledgeConfig())
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

Stereotip cepat

Anda dapat menjalankan algoritma stereotip prompt untuk generasi terbuka. Untuk menjalankan algoritma stereotip prompt, Anda `DataConfig` harus mengidentifikasi kolom dalam kumpulan data input Anda yang berisi kalimat yang kurang stereotip `sent_less_input_location` dan kalimat yang lebih stereotip di. `sent_more_output_location` Untuk informasi selengkapnya tentang`DataConfig`, lihat bagian 2 sebelumnya. Konfigurasikan **`ModelRunner`**. Selanjutnya, panggil `evaluate` metode dan berikan parameter yang Anda inginkan.

Stereotip cepat mengembalikan daftar `EvalOutput` objek yang berisi skor untuk setiap catatan input dan skor keseluruhan untuk setiap jenis bias. Skor dihitung dengan membandingkan probabilitas kalimat stereotip yang semakin sedikit. Skor keseluruhan melaporkan seberapa sering model lebih menyukai kalimat stereotip karena model memberikan probabilitas yang lebih tinggi untuk lebih stereotip dibandingkan dengan kalimat yang kurang stereotip. Skor `0.5` menunjukkan bahwa model Anda tidak bias, atau bahwa ia lebih suka kalimat stereotip yang semakin sedikit pada tingkat yang sama. Skor lebih besar dari `0.5` menunjukkan bahwa model Anda cenderung menghasilkan respons yang lebih stereotip. Skor kurang dari `0.5` menunjukkan bahwa model Anda cenderung menghasilkan respons yang kurang stereotip.

Contoh kode berikut ini menunjukkan cara mengevaluasi LLM Anda menggunakan algoritma stereotip yang cepat:

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.prompt_stereotyping import PromptStereotyping

eval_algo = PromptStereotyping()
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

Kekokohan semantik

Anda dapat menjalankan algoritma ketahanan semantik untuk tugas FMEval apa pun, namun model Anda harus deterministik. Model deterministik adalah model yang selalu menghasilkan output yang sama untuk input yang sama. Seseorang biasanya dapat mencapai determinisme dengan menetapkan benih acak dalam proses decoding. Algoritma berbeda untuk setiap tugas untuk mengakomodasi berbagai jenis input data dan masalah sebagai berikut:

- Untuk generasi terbuka, penjawab pertanyaan, atau klasifikasi tugas, jalankan `GeneralSemanticRobustness` algoritme dengan `GeneralSemanticRobustnessConfig` file.
- Untuk ringkasan teks, jalankan `SummarizationAccuracySemanticRobustness` algoritma dengan `SummarizationAccuracySemanticRobustnessConfig` file.

`GeneralSemanticRobustness`Algoritma mengembalikan daftar `EvalScore` objek yang berisi akurasi dengan nilai antara `0` dan `1` mengukur perbedaan antara output model yang terganggu dan tidak terganggu. Untuk menjalankan algoritma ketahanan semantik umum, buat instance a dan teruskan a. `GeneralSemanticRobustnessConfig` `perturbation_type` Anda dapat memilih salah satu dari yang berikut ini untuk`perturbation_type`:

- `Butterfinger`— Gangguan yang meniru kesalahan ejaan menggunakan swap karakter berdasarkan jarak keyboard. Masukkan probabilitas bahwa karakter tertentu terganggu. Butterfinger adalah nilai default untuk`perturbation_type`.
- `RandomUpperCase`— Gangguan yang mengubah sebagian kecil karakter menjadi huruf besar. Masukan desimal dari `0` ke. 1
- `WhitespaceAddRemove`— Probabilitas bahwa karakter spasi putih ditambahkan di depan karakter spasi non-putih menjadi putih.

Anda juga dapat menentukan parameter berikut:

- `num_perturbations`— Jumlah gangguan untuk setiap sampel untuk dimasukkan ke dalam teks yang dihasilkan. Default-nya adalah `5`.
- `butter_finger_perturbation_prob`— Probabilitas bahwa karakter terganggu. Digunakan hanya ketika `perturbation_type` itu`Butterfinger`. Default-nya adalah `0.1`.
- `random_uppercase_corrupt_proportion`— Fraksi karakter yang akan diubah menjadi huruf besar. Digunakan hanya ketika `perturbation_type` itu`RandomUpperCase`. Default-nya adalah `0.1`.
- `whitespace_add_prob`— Diberikan ruang putih, probabilitas untuk menghapusnya dari sampel. Digunakan hanya ketika `perturbation_type` itu`WhitespaceAddRemove`. Default-nya adalah `0.05`.
- `whitespace_remove_prob`— Diberikan ruang non-putih, kemungkinan menambahkan spasi putih di depannya. Digunakan hanya ketika `perturbation_type` itu`WhitespaceAddRemove`. Default-nya adalah `0.1`.

Terakhir, panggil `evaluate` metode dan berikan parameter yang Anda inginkan seperti yang ditunjukkan pada contoh kode berikut:

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.general_semantic_robustness import
 GeneralSemanticRobustness, GeneralSemanticRobustnessConfig

eval_algo =
 GeneralSemanticRobustness(GeneralSemanticRobustnessConfig(perturbation_type="RandomUpperCase",
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

`SummarizationAccuracySemanticRobustness`Algoritma mengembalikan daftar `EvalScore` objek yang berisi perbedaan (atau delta) antara ROUGE-N, Meteor, dan BERTScorenilai antara ringkasan yang dihasilkan dan referensi. Untuk informasi lebih lanjut tentang skor ini, lihat bagian Ringkasan teks dilkhtisar evaluasi model pondasi. Untuk menjalankan algoritma ketahanan semantik ringkasan teks, buat instance a dan teruskan a. `SummarizationAccuracySemanticRobustnessConfig` `perturbation_type`

Anda dapat memilih salah satu dari yang berikut ini untuk`perturbation_type`:

- `Butterfinger`— Gangguan yang meniru kesalahan ejaan menggunakan swap karakter berdasarkan jarak keyboard. Masukkan probabilitas bahwa karakter tertentu terganggu. `Butterfinger`adalah nilai default untuk`perturbation_type`.
- `RandomUpperCase`— Gangguan yang mengubah sebagian kecil karakter menjadi huruf besar. Masukan desimal dari `0` ke. `1`
- `WhitespaceAddRemove`— Masukkan probabilitas bahwa karakter spasi putih ditambahkan di depan karakter spasi non-putih menjadi putih.

Anda juga dapat menentukan parameter berikut:

- `num_perturbations`— Jumlah gangguan untuk setiap sampel untuk dimasukkan ke dalam teks yang dihasilkan. Default-nya adalah `5`.
- `butter_finger_perturbation_prob`— Probabilitas bahwa karakter terganggu. Digunakan hanya ketika `perturbation_type` itu`Butterfinger`. Default-nya adalah `0.1`.
- `random_uppercase_corrupt_proportion`— Fraksi karakter yang akan diubah menjadi huruf besar. Digunakan hanya ketika `perturbation_type` itu`RandomUpperCase`. Default-nya adalah `0.1`.
- `whitespace_add_prob`— Diberikan ruang putih, probabilitas untuk menghapusnya dari sampel. Digunakan hanya ketika `perturbation_type` itu`WhitespaceAddRemove`. Default-nya adalah `0.05`.
- `whitespace_remove_prob`— Diberikan ruang non-putih, kemungkinan menambahkan spasi putih di depannya. Digunakan hanya ketika `perturbation_type` adalah`WhitespaceAddRemove`, Default adalah`0.1`.
- `rouge_type`— Metrik yang membandingkan ringkasan yang dihasilkan dengan ringkasan referensi. Tentukan jenis ROUGEmetrik yang ingin Anda gunakan dalam evaluasi Anda`rouge_type`. Anda dapat memilih`rouge1`,`rouge2`, atau`rougeL`. ROUGE-1membandingkan

ringkasan yang dihasilkan dan ringkasan referensi menggunakan unigram yang tumpang tindih (urutan satu item seperti “the”, “is”). ROUGE-2membandingkan ringkasan yang dihasilkan dan referensi menggunakan bigram (kelompok dua urutan seperti “yang besar”, “adalah rumah”). ROUGE-Lmembandingkan urutan kata pencocokan terpanjang. Untuk informasi selengkapnyaROUGE, lihat ROUGE: A Package for Automatic Evaluation of Summaries.

- Setel `user_stemmer_for_rouge` ke `True` atau`False`. Stemmer menghapus imbuhan dari kata-kata sebelum membandingkannya. Misalnya, stemmer menghilangkan imbuhan dari “berenang” dan “berenang” sehingga keduanya “berenang” setelah bertangkai.
- Setel `model_type_for_bertscore` ke model yang ingin Anda gunakan untuk menghitung a BERTScore. Anda dapat memilih ROBERTA_MODEL atau MICROSOFT_DEBERTA_MODEL yang lebih canggih.

Panggil `evaluate` metode dan berikan parameter yang Anda inginkan seperti yang ditunjukkan dalam contoh kode berikut:

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.summarization_accuracy_semantic_robustness import
 SummarizationAccuracySemanticRobustness,
 SummarizationAccuracySemanticRobustnessConfig

eval_algo =
 SummarizationAccuracySemanticRobustness(SummarizationAccuracySemanticRobustnessConfig(pertur
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

Toksisitas

Anda dapat menjalankan algoritma toksisitas untuk generasi terbuka, ringkasan teks, atau menjawab pertanyaan. Ada tiga kelas yang berbeda tergantung pada tugas.

- Untuk generasi terbuka, jalankan algoritma Toxicity dengan file. `ToxicityConfig`
- Untuk meringkas, gunakan kelas`Summarization_Toxicity`.
- Untuk menjawab pertanyaan, gunakan kelas`QAToxicity`.

Algoritma toksisitas mengembalikan satu atau lebih daftar `EvalScore` objek (tergantung pada detektor toksisitas) yang berisi skor antara `0` dan. `1` Untuk menjalankan algoritme toksisitas, buat instance a `ToxicityConfig` dan teruskan model toksisitas yang akan digunakan untuk

mengevaluasi model Anda terhadap in. `model_type` Anda dapat memilih yang berikut ini untuk`model_type`:

- `detoksify` untuk UnitaryAI Detoxify-unbias, pengklasifikasi teks multilabel yang dilatih tentang Tantangan Klasifikasi Komentar Beracun dan Bias Tidak Disengaja Jigsaw dalam Klasifikasi Toksisitas. Model ini memberikan 7 skor untuk kelas-kelas berikut: toksisitas, toksisitas parah, kecabulan, ancaman, penghinaan, eksplisit seksual, dan serangan identitas.

  Berikut ini adalah contoh keluaran dari model detoksiitas:

```
EvalScore(name='toxicity', value=0.01936926692724228),

EvalScore(name='severe_toxicity', value=3.3755677577573806e-06),

EvalScore(name='obscene', value=0.00022437423467636108),

EvalScore(name='identity_attack', value=0.0006707844440825284),

EvalScore(name='insult', value=0.005559926386922598),

EvalScore(name='threat', value=0.00016682750720065087),

EvalScore(name='sexual_explicit', value=4.828436431125738e-05)
```

- `toxigen` untuk Toksigen-Roberta, pengklasifikasi teks berbasis Roberta biner yang disetel dengan baik pada kumpulan data, yang berisi kalimat dengan toksisitas halus dan implisit yang berkaitan dengan kelompok minoritas. ToxiGen 13

Terakhir, panggil `evaluate` metode dan berikan parameter yang Anda inginkan seperti yang ditunjukkan pada contoh kode berikut.

```
from fmeval.eval import get_eval_algorithm
from fmeval.eval_algorithms.toxicity import Toxicity, ToxicityConfig

eval_algo = Toxicity(ToxicityConfig(model_type="detoxify"))
eval_output = eval_algo.evaluate(model=model_runner, dataset_config=config,
 prompt_template="$feature", save=True)
```

Sesuaikan alur kerja Anda menggunakan pustaka `fmeval`

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

> **Important**
>
> Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Anda dapat menyesuaikan evaluasi model Anda untuk memungkinkan model yang bukan model SageMaker JumpStart atau Amazon Bedrock atau menggunakan alur kerja khusus untuk evaluasi. Jika Anda menggunakan model Anda sendiri, Anda harus membuat kustom`ModelRunner`. Jika Anda menggunakan dataset Anda sendiri untuk evaluasi, Anda harus mengkonfigurasi `DataConfig` objek. Bagian berikut menunjukkan cara memformat dataset input Anda, menyesuaikan `DataConfig` objek untuk menggunakan dataset kustom Anda, dan membuat kustom. `ModelRunner`

Menggunakan dataset masukan kustom

Jika Anda ingin menggunakan kumpulan data Anda sendiri untuk mengevaluasi model Anda, Anda harus menggunakan `DataConfig` objek untuk menentukan `dataset_name` dan kumpulan data yang ingin Anda evaluasi. `dataset_uri` Jika Anda menggunakan dataset bawaan, `DataConfig` objek sudah dikonfigurasi sebagai default untuk algoritma evaluasi.

Anda dapat menggunakan satu kumpulan data khusus setiap kali Anda menggunakan `evaluate` fungsi tersebut. Anda dapat memanggil `evaluate` beberapa kali untuk menggunakan sejumlah kumpulan data yang Anda inginkan.

Konfigurasikan kumpulan data khusus dengan permintaan model Anda yang ditentukan di kolom pertanyaan, dan jawaban target yang ditentukan dalam jawaban kolom, sebagai berikut:

```
from fmeval.data_loaders.data_config import DataConfig
from fmeval.constants import MIME_TYPE_JSONLINES

config = DataConfig(
    dataset_name="tiny_dataset",
    dataset_uri="tiny_dataset.jsonl",
    dataset_mime_type=MIME_TYPE_JSONLINES,
    model_input_location="question",
    target_output_location="answer",
)
```

`DataConfig`Kelas berisi parameter berikut:

- `dataset_name`— Nama kumpulan data yang ingin Anda gunakan untuk mengevaluasi LLM Anda.

  `dataset_uri`— Jalur lokal atau pengidentifikasi sumber daya seragam (URI) ke lokasi S3 dari kumpulan data Anda.
- `dataset_mime_type`— Format data input yang ingin Anda gunakan untuk mengevaluasi LLM Anda. Pustaka FMEval dapat mendukung keduanya dan`MIME_TYPE_JSON`. `MIME_TYPE_JSONLINES`
- `model_input_location`— (Opsional) Nama kolom dalam dataset Anda yang berisi input model atau prompt yang ingin Anda evaluasi.

  Gunakan `model_input_location` yang menentukan nama kolom Anda. Kolom harus berisi nilai-nilai berikut yang sesuai dengan tugas-tugas terkait berikut:
  - Untuk evaluasi generasi terbuka, toksisitas, dan akurasi, tentukan kolom yang berisi prompt yang harus ditanggapi oleh model Anda.
  - Untuk tugas menjawab pertanyaan, tentukan kolom yang berisi pertanyaan yang harus direspon oleh model Anda.
  - Untuk tugas meringkas teks, tentukan nama kolom yang berisi teks yang ingin diringkas oleh model Anda.
  - Untuk tugas klasifikasi, tentukan nama kolom yang berisi teks yang ingin diklasifikasikan oleh model Anda.
  - Untuk evaluasi pengetahuan faktual, tentukan nama kolom yang berisi pertanyaan yang Anda inginkan model untuk memprediksi jawabannya.
  - Untuk evaluasi ketahanan semantik, tentukan nama kolom yang berisi input yang Anda inginkan untuk diganggu oleh model Anda.

  - Untuk evaluasi stereotip yang cepat, gunakan `sent_more_input_location` dan `sent_less_input_location` bukan`model_input_location`, seperti yang ditunjukkan pada parameter berikut.
- `model_output_location`— (Opsional) Nama kolom dalam dataset Anda yang berisi output prediksi yang ingin Anda bandingkan dengan output referensi yang terkandung di `target_output_location` dalamnya. Jika Anda memberikan`model_output_location`, maka FMEval tidak akan mengirim permintaan ke model Anda untuk inferensi. Sebaliknya, ia menggunakan output yang terkandung dalam kolom yang ditentukan untuk mengevaluasi model Anda.
- `target_output_location`— Nama kolom dalam dataset referensi yang berisi nilai sebenarnya untuk dibandingkan dengan nilai prediksi yang terkandung di `model_output_location` dalamnya. Diperlukan hanya untuk pengetahuan faktual, akurasi, dan ketahanan semantik. Untuk pengetahuan faktual, setiap baris di kolom ini harus berisi semua kemungkinan jawaban yang dipisahkan oleh pembatas. Misalnya, jika jawaban untuk pertanyaan adalah [“UK”, “Inggris”], maka kolom harus berisi “Inggris <OR>Inggris”. Prediksi model benar jika berisi salah satu jawaban yang dipisahkan oleh pembatas.
- `category_location`- Nama kolom yang berisi nama kategori. Jika Anda memberikan nilai untuk`category_location`, maka skor dikumpulkan dan dilaporkan untuk setiap kategori.
- `sent_more_input_location`— Nama kolom yang berisi prompt dengan bias lebih. Diperlukan hanya untuk stereotip cepat. Hindari bias bawah sadar. Untuk contoh bias, lihat kumpulan data Crows-pairs.
- `sent_less_input_location`— Nama kolom yang berisi prompt dengan bias yang lebih sedikit. Diperlukan hanya untuk stereotip cepat. Hindari bias bawah sadar. Untuk contoh bias, lihat kumpulan data Crows-pairs.
- `sent_more_output_location`— (Opsional) Nama kolom yang berisi probabilitas prediksi bahwa respons yang dihasilkan model Anda akan mengandung lebih banyak bias. Parameter ini hanya digunakan dalam tugas stereotip cepat.
- `sent_less_output_location`— (Opsional) Nama kolom yang berisi probabilitas prediksi bahwa respons yang dihasilkan model Anda akan mengandung lebih sedikit bias. Parameter ini hanya digunakan dalam tugas stereotip cepat.

Jika Anda ingin menambahkan atribut baru yang sesuai dengan kolom dataset ke `DataConfig` kelas, Anda harus menambahkan `suffix _location` ke akhir nama atribut.

Gunakan kustom **`ModelRunner`**

Untuk mengevaluasi model kustom, gunakan kelas data dasar untuk mengonfigurasi model Anda dan membuat kustom`ModelRunner`. Kemudian, Anda dapat menggunakan ini `ModelRunner` untuk mengevaluasi model bahasa apa pun. Gunakan langkah-langkah berikut untuk menentukan konfigurasi model, membuat kustom`ModelRunner`, dan mengujinya.

`ModelRunner`Antarmuka memiliki satu metode abstrak sebagai berikut:

```
def predict(self, prompt: str) # Tuple[Optional[str], Optional[float]]
```

Metode ini mengambil prompt sebagai input string, dan mengembalikan Tuple yang berisi respons teks model dan probabilitas log masukan. Setiap orang `ModelRunner` harus menerapkan `predict` metode.

Buat kustom **`ModelRunner`**

1. Tentukan konfigurasi model.

   Contoh kode berikut menunjukkan cara menerapkan `dataclass` dekorator ke `HFModelConfig` kelas khusus sehingga Anda dapat menentukan konfigurasi model untuk Hugging Facemodel:

   ```
   from dataclasses import dataclass

   @dataclass
   class HFModelConfig:
       model_name: str
       max_new_tokens: int
       seed: int = 0
       remove_prompt_from_generated_text: bool = True
   ```

   Pada contoh kode sebelumnya, hal berikut berlaku:

   - Parameter `max_new_tokens` ini digunakan untuk membatasi panjang respons dengan membatasi jumlah token yang dikembalikan oleh LLM. Jenis model diatur dengan melewatkan nilai `model_name` ketika kelas dipakai. Dalam contoh ini, nama model diatur ke`gpt2`, seperti yang ditunjukkan pada akhir bagian ini. Parameter `max_new_tokens` adalah salah satu opsi untuk mengonfigurasi strategi pembuatan teks menggunakan konfigurasi `gpt2` model untuk model GPT OpenAI yang telah dilatih sebelumnya. Lihat AutoConfiguntuk jenis model lainnya.

- Jika parameter `remove_prompt_from_generated_text` disetel ke`True`, maka respons yang dihasilkan tidak akan berisi prompt asal yang dikirim dalam permintaan.

Untuk parameter pembuatan teks lainnya, lihat Hugging Facedokumentasi untuk GenerationConfig.

2. Buat kustom `ModelRunner` dan terapkan metode prediksi. Contoh kode berikut menunjukkan cara membuat kustom `ModelRunner` untuk Hugging Face model menggunakan `HFModelConfig` kelas yang dibuat dalam contoh kode sebelumnya.

```
from typing import Tuple, Optional
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer
from fmeval.model_runners.model_runner import ModelRunner

class HuggingFaceCausalLLMModelRunner(ModelRunner):
    def __init__(self, model_config: HFModelConfig):
        self.config = model_config
        self.model = AutoModelForCausalLM.from_pretrained(self.config.model_name)
        self.tokenizer = AutoTokenizer.from_pretrained(self.config.model_name)

    def predict(self, prompt: str) -> Tuple[Optional[str], Optional[float]]:
        input_ids = self.tokenizer(prompt,
 return_tensors="pt").to(self.model.device)
        generations = self.model.generate(
            **input_ids,
            max_new_tokens=self.config.max_new_tokens,
            pad_token_id=self.tokenizer.eos_token_id,
        )
        generation_contains_input = (
            input_ids["input_ids"][0] == generations[0][:
 input_ids["input_ids"].shape[1]]
        ).all()
        if self.config.remove_prompt_from_generated_text and not
 generation_contains_input:
            warnings.warn(
                "Your model does not return the prompt as part of its generations.
 "
                "`remove_prompt_from_generated_text` does nothing."
            )
        if self.config.remove_prompt_from_generated_text and
 generation_contains_input:
```

```
            output = self.tokenizer.batch_decode(generations[:,
input_ids["input_ids"].shape[1] :])[0]
        else:
            output = self.tokenizer.batch_decode(generations,
skip_special_tokens=True)[0]

        with torch.inference_mode():
            input_ids = self.tokenizer(self.tokenizer.bos_token + prompt,
return_tensors="pt")["input_ids"]
            model_output = self.model(input_ids, labels=input_ids)
            probability = -model_output[0].item()

        return output, probability
```

Kode sebelumnya menggunakan `HuggingFaceCausalLLMModelRunner` kelas kustom yang mewarisi properti dari kelas FMEval`ModelRunner`. Kelas kustom berisi konstruktor dan definisi untuk fungsi prediksi, yang mengembalikan a`Tuple`.

Untuk `ModelRunner` contoh lainnya, lihat bagian model_runner perpustakaan. `fmeval`

`HuggingFaceCausalLLMModelRunner`Konstruktor berisi definisi berikut:

- Konfigurasi diatur ke`HFModelConfig`, didefinisikan di awal bagian ini.
- Model diatur ke model pra-terlatih dari Hugging Face Auto Class yang ditentukan menggunakan parameter model_name pada instantiation.
- Tokenizer diatur ke kelas dari pustaka Hugging Face tokenizer yang cocok dengan model pra-terlatih yang ditentukan oleh. `model_name`

`predict`Metode di `HuggingFaceCausalLLMModelRunner` kelas menggunakan definisi berikut:

- `input_ids`— Variabel yang berisi masukan untuk model Anda. Model menghasilkan input sebagai berikut.
  - A `tokenizer` Mengonversi permintaan yang terkandung `prompt` dalam pengenal token (ID). ID token ini, yang merupakan nilai numerik yang mewakili token tertentu (kata, sub-kata atau karakter), dapat digunakan langsung oleh model Anda sebagai input. ID token dikembalikan sebagai objek PyTorch tensor, seperti yang ditentukan oleh`return_tensors="pt"`. Untuk tipe tipe tensor kembali lainnya, lihat Hugging Face dokumentasi untuk apply_chat_template.

    - ID Token dikirim ke perangkat tempat model berada sehingga dapat digunakan oleh model.
- `generations`— Variabel yang berisi respons yang dihasilkan oleh LLM Anda. Fungsi generate model menggunakan input berikut untuk menghasilkan respons:
    - `input_ids`Dari langkah sebelumnya.
    - Parameter `max_new_tokens` yang ditentukan dalam`HFModelConfig`.
    - A `pad_token_id` menambahkan token akhir kalimat (eos) ke respons. Untuk token lain yang dapat Anda gunakan, lihat Hugging Face dokumentasi untuk PreTrainedTokenizer.
- `generation_contains_input`- Variabel boolean yang kembali `True` ketika respons yang dihasilkan menyertakan prompt input dalam responsnya, dan `False` sebaliknya. Nilai pengembalian dihitung menggunakan perbandingan elemen antara yang berikut ini.
    - Semua ID token dalam prompt input yang terkandung di dalamnya`input_ids["input_ids"][0]`.
    - Awal dari konten yang dihasilkan yang terkandung di dalamnya`generations[0][: input_ids["input_ids"].shape[1]]`.

    `predict`Metode ini mengembalikan peringatan jika Anda mengarahkan LLM ke `remove_prompt_from_generated_text` dalam konfigurasi Anda tetapi respons yang dihasilkan tidak berisi prompt input.

    Output dari `predict` metode ini berisi string yang dikembalikan oleh `batch_decode` metode, yang mengubah ID token yang dikembalikan dalam respons menjadi teks yang dapat dibaca manusia. Jika Anda menentukan `remove_prompt_from_generated_text` sebagai`True`, maka prompt input dihapus dari teks yang dihasilkan. Jika Anda menentukan `remove_prompt_from_generated_text` sebagai`False`, teks yang dihasilkan akan dikembalikan tanpa token khusus yang Anda sertakan dalam kamus`special_token_dict`, seperti yang ditentukan oleh`skip_special_tokens=True`.

3. Uji Anda`ModelRunner`. Kirim permintaan sampel ke model Anda.

   Contoh berikut menunjukkan cara menguji model menggunakan model `gpt2` pra-terlatih dari Hugging Face `AutoConfig` kelas:

```
hf_config = HFModelConfig(model_name="gpt2", max_new_tokens=32)
model = HuggingFaceCausalLLMModelRunner(model_config=hf_config)
```

Dalam contoh kode sebelumnya, `model_name` tentukan nama model pra-terlatih. `HFModelConfig`Kelas ini dipakai sebagai hf_config dengan nilai untuk parameter`max_new_tokens`, dan digunakan untuk menginisialisasi. `ModelRunner`

Jika Anda ingin menggunakan model pra-terlatih lainnyaHugging Face, pilih `pretrained_model_name_or_path` di `from_pretrained` bawah AutoClass.

Terakhir, uji Anda`ModelRunner`. Kirim permintaan sampel ke model Anda seperti yang ditunjukkan dalam contoh kode berikut:

```
model_output = model.predict("London is the capital of?")[0]
print(model_output)
eval_algo.evaluate_sample()
```

Tutorial Notebook

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

> **Important**
>
> Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Bagian ini menyediakan tutorial notebook berikut, yang mencakup contoh kode dan penjelasan:

- Bagaimana mengevaluasi SageMaker JumpStart model untuk stereotip cepat.
- Cara mengevaluasi model Amazon Bedrock untuk akurasi ringkasan teks.

Bagaimana mengevaluasi SageMaker JumpStart model untuk stereotip cepat

Anda dapat menggunakan `ModelRunner` pembungkus tingkat tinggi untuk mengevaluasi SageMaker JumpStart model Amazon untuk stereotip cepat. Algoritma stereotip cepat mengukur probabilitas bias pengkodean model Anda dalam responsnya. Bias ini termasuk untuk ras, jenis kelamin, orientasi seksual, agama, usia, kebangsaan, kecacatan, penampilan fisik, dan status sosial ekonomi.

Tutorial ini menunjukkan cara memuat model Falcon 7-B dari Technology Innovation Institute, tersedia di SageMaker JumpStart, dan meminta model ini untuk menghasilkan tanggapan terhadap petunjuk. Kemudian, tutorial ini menunjukkan cara mengevaluasi respons untuk stereotip cepat terhadap kumpulan data tantangan sumber terbuka Crows-pair bawaan.

Bagian dari tutorial ini menunjukkan cara melakukan hal berikut:

- Siapkan lingkungan Anda.
- Jalankan evaluasi model Anda.
- Lihat hasil analisis Anda.

Siapkan lingkungan Anda

Prasyarat

- Gunakan lingkungan kernel dasar Python 3.10 dan instans `ml.g4dn.2xlarge` Amazon Elastic Compute Cloud (Amazon EC2) sebelum memulai tutorial ini.

  Untuk informasi selengkapnya tentang tipe instans dan kasus penggunaan yang direkomendasikan, lihatJenis Instans Studio Klasik yang Tersedia.

Menginstal pustaka yang diperlukan

1. Instal SageMaker,`fmeval`, dan pustaka lain yang diperlukan dalam kode Anda sebagai berikut:

```
!pip3 install sagemaker
!pip3 install -U pyarrow
!pip3 install -U accelerate
!pip3 install "ipywidgets>=8"
!pip3 install jsonlines
!pip install fmeval
!pip3 install boto3==1.28.65
```

```
import sagemaker
```

2. Unduh `JSON Lines` kumpulan data sampel crows-pairs_sample.jsonl, ke direktori kerja Anda saat ini.
3. Periksa apakah lingkungan Anda berisi file input sampel menggunakan kode berikut:

```
import glob

# Check for fmeval wheel and built-in dataset
if not glob.glob("crows-pairs_sample.jsonl"):
    print("ERROR - please make sure file exists: crows-pairs_sample.jsonl")
```

4. Tentukan SageMaker JumpStart model sebagai berikut:

```
from sagemaker.jumpstart.model import JumpStartModel

model_id, model_version, = (
"huggingface-llm-falcon-7b-instruct-bf16",
"*",
)
```

5. Terapkan SageMaker JumpStart model dan buat titik akhir sebagai berikut:

```
my_model = JumpStartModel(model_id=model_id)
predictor = my_model.deploy()
endpoint_name = predictor.endpoint_name
```

6. Tentukan prompt dan format permintaan model, atau payload, sebagai berikut:

```
prompt = "London is the capital of"
payload = {
    "inputs": prompt,
    "parameters": {
        "do_sample": True,
        "top_p": 0.9,
        "temperature": 0.8,
        "max_new_tokens": 1024,
        "decoder_input_details" : True,
        "details" : True
    },
}
```

Dalam contoh kode sebelumnya, parameter berikut disertakan dalam permintaan model:

- `do_sample`— Menginstruksikan model untuk mengambil sampel dari output model mentah (sebelum normalisasi) selama inferensi model untuk memperkenalkan keragaman dan kreativitas ke dalam respons model. Default ke `False`. Jika Anda mengatur `do_sample` ke`True`, maka Anda harus menentukan nilai untuk salah satu parameter berikut:`temperature`,`top_k`,`top_p`, atau`typical_p`.
- `top_p`— Mengontrol keacakan dengan membatasi set token untuk dipertimbangkan saat membuat token berikutnya. Nilai yang lebih tinggi dari `top_p` memungkinkan untuk satu set yang berisi kosakata yang lebih luas. Nilai yang lebih rendah membatasi kumpulan token ke kata-kata yang lebih mungkin. Rentang untuk `top_p` lebih besar dari `0` dan kurang dari`1`.
- `temperature`— Mengontrol keacakan teks yang dihasilkan. Nilai yang lebih tinggi dari `temperature` menginstruksikan model untuk menghasilkan respons yang lebih acak dan beragam. Nilai yang lebih rendah menghasilkan respons yang lebih dapat diprediksi. Nilai untuk `temperature` harus positif.
- `max_new_tokens`— Membatasi panjang respons dengan membatasi jumlah token yang dikembalikan oleh model Anda. Default ke `20`.
- `decoder_input_details`— Mengembalikan informasi tentang probabilitas log yang ditetapkan oleh model untuk setiap token potensial berikutnya dan ID token yang sesuai. Jika `decoder_input_details` diatur ke`True`, Anda juga harus mengatur `details` untuk `True` menerima detail yang diminta. Default ke `False`.

Untuk informasi selengkapnya tentang parameter untuk `Hugging Face` model ini, lihat types.py.

Kirim permintaan inferensi sampel

Untuk menguji model Anda, kirim permintaan sampel ke model Anda dan cetak respons model sebagai berikut:

```
response = predictor.predict(payload)
print(response[0]["generated_text"])
```

Dalam contoh kode sebelumnya, jika model Anda memberikan respons`[{"response": "this is the output"}]`, maka `print` pernyataan akan kembali`this is the output`.

Menyelesaikan FMEval

1. Muat pustaka yang diperlukan untuk menjalankan FMEval sebagai berikut:

```
import fmeval
from fmeval.data_loaders.data_config import DataConfig
from fmeval.model_runners.sm_jumpstart_model_runner import JumpStartModelRunner
from fmeval.constants import MIME_TYPE_JSONLINES
from fmeval.eval_algorithms.prompt_stereotyping import PromptStereotyping,
 PROMPT_STEREOTYPING
from fmeval.eval_algorithms import EvalAlgorithm
```

2. Siapkan konfigurasi data untuk dataset input Anda.

   Jika Anda tidak menggunakan kumpulan data bawaan, konfigurasi data Anda harus mengidentifikasi kolom yang berisi lebih banyak bias. `sent_more_input_location` Anda juga harus mengidentifikasi kolom yang mengandung lebih sedikit bias di`sent_less_input_location`. Jika Anda menggunakan dataset bawaan dari SageMaker JumpStart, parameter ini diteruskan ke FMEval secara otomatis melalui metadata model.

   Tentukan `sent_more_input_location` dan `sent_less_input_location` kolom untuk tugas stereotip cepat, nama, pengidentifikasi sumber daya seragam (URI), dan `MIME` jenis.

```
config = DataConfig(
    dataset_name="crows-pairs_sample",
    dataset_uri="crows-pairs_sample.jsonl",
    dataset_mime_type=MIME_TYPE_JSONLINES,
    sent_more_input_location="sent_more",
    sent_less_input_location="sent_less",
    category_location="bias_type",
)
```

   Untuk informasi selengkapnya tentang informasi kolom yang diperlukan tugas lain, lihat bagian Menggunakan kumpulan data input kustom diMenggunakan dataset masukan kustom.

3. Siapkan kustom `ModelRunner` seperti yang ditunjukkan dalam contoh kode berikut:

```
js_model_runner = JumpStartModelRunner(
    endpoint_name=endpoint_name,
    model_id=model_id,
    model_version=model_version,
    output='[0].generated_text',
```

```
    log_probability='[0].details.prefill[*].logprob',
    content_template='{"inputs": $prompt, "parameters":
    {"do_sample": true, "top_p": 0.9, "temperature": 0.8, "max_new_tokens": 1024,
    "decoder_input_details": true,"details": true}}',
)
```

Contoh kode sebelumnya menentukan yang berikut:

- `endpoint_name`— Nama titik akhir yang Anda buat di langkah Instal diperlukan library sebelumnya.
- `model_id`— Id yang digunakan untuk menentukan model Anda. Parameter ini ditentukan ketika SageMaker JumpStart model didefinisikan.
- `model_version`— Versi model Anda yang digunakan untuk menentukan model Anda. Parameter ini ditentukan ketika SageMaker JumpStart model didefinisikan.
- `output`— Menangkap output dari model Falcon 7b, yang mengembalikan responsnya dalam kunci. `generated_text` Jika model Anda memberikan respons`[{"generated_text": "this is the output"}]`, maka `[0].generated_text` kembali`this is the output`.
- `log_probability`— Menangkap probabilitas log yang dikembalikan oleh SageMaker JumpStart model ini.
- `content_template`— Menentukan bagaimana model Anda berinteraksi dengan permintaan. Contoh template konfigurasi dirinci semata-mata untuk menjelaskan contoh sebelumnya, dan itu tidak diperlukan. Parameter dalam template konten adalah parameter yang sama yang dideklarasikan untuk`payload`. Untuk informasi selengkapnya tentang parameter untuk `Hugging Face` model ini, lihat types.py.

4. Konfigurasikan laporan evaluasi Anda dan simpan ke direktori seperti yang ditunjukkan pada contoh kode berikut:

```
import os
eval_dir = "results-eval-prompt-stereotyping"
curr_dir = os.getcwd()
eval_results_path = os.path.join(curr_dir, eval_dir) + "/"
os.environ["EVAL_RESULTS_PATH"] = eval_results_path
if os.path.exists(eval_results_path):
    print(f"Directory '{eval_results_path}' exists.")
else:
    os.mkdir(eval_results_path)
```

5. Siapkan faktor paralelisasi sebagai berikut:

```
os.environ["PARALLELIZATION_FACTOR"] = "1"
```

A `PARALLELIZATION_FACTOR` adalah pengganda untuk jumlah batch bersamaan yang dikirim ke instance komputasi Anda. Jika perangkat keras Anda memungkinkan paralelisasi, Anda dapat mengatur nomor ini untuk melipatgandakan jumlah pemanggilan untuk pekerjaan evaluasi Anda. Misalnya, jika Anda memiliki `100` pemanggilan, dan `PARALLELIZATION_FACTOR` disetel ke2, maka pekerjaan Anda akan menjalankan `200` pemanggilan. Anda dapat meningkatkan `PARALLELIZATION_FACTOR` hingga`10`, atau menghapus variabel seluruhnya. Untuk membaca blog tentang cara penggunaan AWS Lambda, `PARALLELIZATION_FACTOR` lihat Kontrol penskalaan AWS Lambda Baru untuk sumber peristiwa Kinesis dan DynamoDB.

Jalankan evaluasi model Anda

1. Tentukan algoritma evaluasi Anda. Contoh berikut menunjukkan cara mendefinisikan `PromptStereotyping` algoritma:

```
eval_algo = PromptStereotyping()
```

Untuk contoh algoritme yang menghitung metrik untuk tugas evaluasi lainnya, lihat Mengevaluasi model Anda. Gunakan `fmeval` pustaka untuk menjalankan evaluasi otomatis

2. Jalankan algoritma evaluasi Anda. Contoh kode berikut menggunakan model dan konfigurasi data yang telah ditentukan sebelumnya, dan `prompt_template` yang digunakan `feature` untuk meneruskan prompt Anda ke model sebagai berikut:

```
eval_output = eval_algo.evaluate(model=js_model_runner, dataset_config=config,
prompt_template="$feature", save=True)
```

Output model Anda mungkin berbeda dari output sampel sebelumnya.

Melihat hasil analisis Anda

1. Parse laporan evaluasi dari `eval_output` objek yang dikembalikan oleh algoritma evaluasi sebagai berikut:

```
import json
```

```
print(json.dumps(eval_output, default=vars, indent=4))
```

Perintah sebelumnya mengembalikan output berikut (diringkas untuk singkatnya):

```
[
    {
        "eval_name": "prompt_stereotyping",
        "dataset_name": "crows-pairs_sample",
        "dataset_scores": [
            {
                "name": "prompt_stereotyping",
                "value": 0.6666666666666666
            }
        ],
        "prompt_template": "$feature",
        "category_scores": [
            {
                "name": "disability",
                "scores": [
                    {
                        "name": "prompt_stereotyping",
                        "value": 0.5
                    }
                ]
            },
            ...
        ],
        "output_path": "/home/sagemaker-user/results-eval-prompt-stereotyping/
prompt_stereotyping_crows-pairs_sample.jsonl",
        "error": null
    }
]
```

Output contoh sebelumnya menampilkan skor keseluruhan untuk dataset berikut`"name": prompt_stereotyping`. Skor ini adalah perbedaan yang dinormalisasi dalam probabilitas log antara respons model yang memberikan bias lebih banyak versus lebih sedikit. Jika skornya lebih besar dari`0.5`, ini berarti respons model Anda lebih cenderung mengembalikan respons yang mengandung lebih banyak bias. Jika skornya kurang dari`0.5`, model Anda lebih cenderung mengembalikan respons yang mengandung lebih sedikit bias. Jika skornya`0.5`, respons model tidak mengandung bias yang diukur dengan dataset input. Anda akan menggunakan `output_path` untuk membuat `Pandas DataFrame` di langkah berikut.

2. Impor hasil Anda dan baca ke dalam`DataFrame`, dan lampirkan skor stereotip prompt ke input model, output model, dan output target sebagai berikut:

```
import pandas as pd
data = []
with open(os.path.join(eval_results_path,
"prompt_stereotyping_crows-pairs_sample.jsonl"), "r") as file:
for line in file:
data.append(json.loads(line))
df = pd.DataFrame(data)
df['eval_algo'] = df['scores'].apply(lambda x: x[0]['name'])
df['eval_score'] = df['scores'].apply(lambda x: x[0]['value'])
df
```

   Untuk buku catatan yang berisi contoh kode yang diberikan di bagian ini, lihat jumpstart-falcon-stereotyping.ipnyb.

Cara mengevaluasi model Amazon Bedrock untuk akurasi ringkasan teks

Anda dapat menggunakan `ModelRunner` pembungkus tingkat tinggi untuk membuat evaluasi khusus berdasarkan model yang di-host di luar. SageMaker JumpStart

Tutorial ini menunjukkan cara memuat model Anthropic Claude 2, yang tersedia di Amazon Bedrock, dan meminta model ini untuk meringkas petunjuk teks. Kemudian, tutorial ini menunjukkan cara mengevaluasi respons model untuk akurasi menggunakan Rouge-L, Meteor, dan BERTScoremetrik.

Tutorial menunjukkan cara melakukan hal berikut:

- Siapkan lingkungan Anda.
- Jalankan evaluasi model Anda.
- Lihat hasil analisis Anda.

Siapkan lingkungan Anda

Prasyarat

- Gunakan lingkungan kernel dasar Python 3.10 dan instans `ml.m5.2xlarge` Amazon Elastic Compute Cloud (Amazon EC2) sebelum memulai tutorial ini.

Untuk informasi tambahan tentang jenis instans dan kasus penggunaan yang direkomendasikan, lihatJenis Instans Studio Klasik yang Tersedia.

Siapkan Bedrock Amazon

Sebelum Anda dapat menggunakan model Amazon Bedrock, Anda harus meminta akses ke sana.

1. Masuk ke AndaAkun AWS.
   - Jika Anda tidak memiliki AWS akun, lihat Mendaftar untuk AWS akun di Mengatur Amazon Bedrock.
2. Buka konsol Amazon Bedrock.
3. Selamat Datang di Amazon Bedrock! bagian yang terbuka, pilih Kelola akses model.
4. Di bagian Akses model yang muncul, pilih Kelola akses model.
5. Di bagian Model dasar yang muncul, centang kotak di sebelah Claude yang tercantum di bawah subbagian Antropik Model.
6. Pilih Minta akses model.
7. Jika permintaan Anda berhasil, tanda centang dengan Akses yang diberikan akan muncul di bawah Status akses di sebelah model yang Anda pilih.
8. Anda mungkin perlu masuk kembali Akun AWS ke Anda untuk dapat mengakses model.

Menginstal pustaka yang diperlukan

1. Dalam kode Anda, instal `fmeval` dan `boto3` pustaka sebagai berikut:

```
!pip install fmeval
!pip3 install boto3==1.28.65
```

2. Impor pustaka, tetapkan faktor paralelisasi, dan panggil klien Amazon Bedrock sebagai berikut:

```
import boto3
import json
import os

# Dependent on available hardware and memory
os.environ["PARALLELIZATION_FACTOR"] = "1"
```

```
# Bedrock clients for model inference
bedrock = boto3.client(service_name='bedrock')
bedrock_runtime = boto3.client(service_name='bedrock-runtime')
```

Pada contoh kode sebelumnya, hal berikut berlaku:

- `PARALLELIZATION_FACTOR`— Pengganda untuk jumlah batch bersamaan yang dikirim ke instance komputasi Anda. Jika perangkat keras Anda memungkinkan paralelisasi, Anda dapat mengatur nomor ini untuk mengalikan jumlah pemanggilan untuk pekerjaan evaluasi Anda dengan. Misalnya, jika Anda memiliki `100` pemanggilan, dan `PARALLELIZATION_FACTOR` disetel ke`2`, maka pekerjaan Anda akan menjalankan `200` pemanggilan. Anda dapat meningkatkan `PARALLELIZATION_FACTOR` hingga`10`, atau menghapus variabel seluruhnya. Untuk membaca blog tentang cara penggunaan AWS Lambda, `PARALLELIZATION_FACTOR` lihat Kontrol penskalaan AWS Lambda Baru untuk sumber peristiwa Kinesis dan DynamoDB.

3. Unduh `JSON Lines` kumpulan data sampel, xsum_sample.jsonl, ke dalam direktori kerja Anda saat ini.
4. Periksa apakah lingkungan Anda berisi file input sampel sebagai berikut:

```
import glob

# Check for the built-in dataset
if not glob.glob("xsum_sample.jsonl"):
    print("ERROR - please make sure file exists: xsum_sample.jsonl")
```

Kirim permintaan inferensi sampel ke model Anda

1. Tentukan model dan `MIME` jenis prompt Anda. Untuk model Anthropic Claude 2 yang dihosting di Amazon Bedrock, prompt Anda harus disusun sebagai berikut:

```
import json
model_id = 'anthropic.claude-v2'
accept = "application/json"
contentType = "application/json"
# Ensure that your prompt has the correct format
prompt_data = """Human: Who is Barack Obama?
Assistant:
"""
```

Untuk informasi selengkapnya tentang cara menyusun isi permintaan Anda, lihat Bidang badan permintaan pemanggilan model. Model lain mungkin memiliki format yang berbeda.

2. Kirim permintaan sampel ke model Anda. Isi permintaan Anda berisi prompt dan parameter tambahan apa pun yang ingin Anda atur. Permintaan sampel dengan `max_tokens_to_sample` set untuk `500` berikut:

```
body = json.dumps({"prompt": prompt_data, "max_tokens_to_sample": 500})
response = bedrock_runtime.invoke_model(
body=body, modelId=model_id, accept=accept, contentType=contentType
)
response_body = json.loads(response.get("body").read())
print(response_body.get("completion"))
```

Dalam contoh kode sebelumnya, Anda dapat mengatur parameter berikut:

- `temperature`— Mengontrol keacakan teks yang dihasilkan, dan menerima nilai positif. Nilai yang lebih tinggi dari `temperature` menginstruksikan model untuk menghasilkan respons yang lebih acak dan beragam. Nilai yang lebih rendah menghasilkan respons yang lebih dapat diprediksi. Rentang untuk `temperature` kebohongan antara `0` dan`1`, dengan nilai default 0,5.
- `topP`— Mengontrol keacakan dengan membatasi set token untuk dipertimbangkan saat membuat token berikutnya. Nilai yang lebih tinggi dari `topP` memungkinkan untuk satu set yang berisi kosakata yang lebih luas dan nilai yang lebih rendah membatasi kumpulan token ke kata-kata yang lebih mungkin. Rentang `0` untuk `topP` are to`1`, dengan nilai default`1`.
- `topK`— Membatasi prediksi model ke token k paling mungkin teratas. Nilai yang lebih tinggi `topK` memungkinkan respons yang lebih inventif. Nilai yang lebih rendah menghasilkan respons yang lebih koheren. Rentang `0` untuk `topK` are to`500`, dengan nilai default`250`.
- `max_tokens_to_sample`— Membatasi panjang respons dengan membatasi jumlah token yang dikembalikan oleh model Anda. Rentang `0` untuk `max_tokens_to_sample` are to`4096`, dengan nilai default`200`.
- `stop_sequences`— Menentukan daftar urutan karakter yang memberi tahu model Anda untuk berhenti menghasilkan respons. Output model dihentikan saat pertama kali salah satu string yang terdaftar ditemui dalam output. Respons tidak mengandung urutan berhenti. Misalnya, Anda dapat menggunakan urutan pengembalian carriage untuk membatasi respons model ke satu baris. Anda dapat mengonfigurasi hingga `4` menghentikan urutan.

Untuk informasi lebih lanjut tentang parameter yang dapat Anda tentukan dalam permintaan, lihat Anthropic Claude.

Menyelesaikan FMEval

1. Muat pustaka yang diperlukan untuk menjalankan FMEval sebagai berikut:

```
from fmeval.data_loaders.data_config import DataConfig
from fmeval.model_runners.bedrock_model_runner import BedrockModelRunner
from fmeval.constants import MIME_TYPE_JSONLINES
from fmeval.eval_algorithms.summarization_accuracy import SummarizationAccuracy,
 SummarizationAccuracyConfig
```

2. Siapkan konfigurasi data untuk dataset input Anda.

   Masukan sampel berikut adalah satu baris dari`xsum_sample.jsonl`:

```
{
    "document": "23 October 2015 Last updated at 17:44
        BST\nIt's the highest rating a tropical storm
        can get and is the first one of this magnitude
        to hit mainland Mexico since 1959.\nBut how are
        the categories decided and what do they mean?
        Newsround reporter Jenny Lawrence explains.",
    "summary": "Hurricane Patricia has been rated as
        a category 5 storm.",
    "id": "34615665",
}
```

   Masukan sampel sebelumnya berisi teks untuk diringkas di dalam `document` kunci. Referensi yang digunakan untuk mengevaluasi respons model Anda ada di `summary` kuncinya. Anda harus menggunakan kunci ini di dalam konfigurasi data Anda untuk menentukan kolom mana yang berisi informasi yang dibutuhkan FMEval untuk mengevaluasi respons model.

   Konfigurasi data Anda harus mengidentifikasi teks yang harus diringkas oleh model Anda. `model_input_location` Anda harus mengidentifikasi nilai referensi dengan`target_output_location`.

Contoh konfigurasi data berikut mengacu pada contoh input sebelumnya untuk menentukan kolom yang diperlukan untuk tugas ringkasan teks, nama, pengidentifikasi sumber daya seragam (URI), dan `MIME` jenis:

```
config = DataConfig(
    dataset_name="xsum_dataset",
    dataset_uri="xsum_sample.jsonl",
    dataset_mime_type=MIME_TYPE_JSONLINES,
    model_input_location="document",
    target_output_location="summary"
)
```

Untuk informasi selengkapnya tentang informasi kolom yang diperlukan untuk tugas lain, lihat bagian Menggunakan kumpulan data input kustom diGunakan evaluasi otomatis.

3. Siapkan kustom `ModelRunner` seperti yang ditunjukkan dalam contoh kode berikut:

```
bedrock_model_runner = BedrockModelRunner(
    model_id=model_id,
    output='completion',
    content_template='{"prompt": $prompt, "max_tokens_to_sample": 500}'
)
```

Contoh kode sebelumnya menentukan yang berikut:

- `model_id`— Id yang digunakan untuk menentukan model Anda.
- `output`— Menangkap output dari model Anthropic Claude 2, yang mengembalikan responsnya dalam kunci. `completion`
- `content_template`— Menentukan bagaimana model Anda berinteraksi dengan permintaan. Contoh template konfigurasi dirinci sebagai berikut semata-mata untuk menjelaskan contoh sebelumnya, dan itu tidak diperlukan.
  - Pada `content_template` contoh sebelumnya, berikut ini berlaku:
    - Variabel `prompt` menentukan prompt input, yang menangkap permintaan yang dibuat oleh pengguna.
    - Variabel `max_tokens_to_sample` menentukan jumlah maksimum token untuk`500`, untuk membatasi panjang respon.

Untuk informasi lebih lanjut tentang parameter yang dapat Anda tentukan dalam permintaan Anda, lihat Model Anthropic Claude.

Format `content_template` parameter tergantung pada input dan parameter yang didukung oleh LLM Anda. Dalam tutorial ini, model Claude 2 Anthropic menggunakan yang berikut: `content_template`

```
   "content_template": "{\"prompt\": $prompt, \"max_tokens_to_sample\": 500}"
```

Sebagai contoh lain, model Falcon 7b dapat mendukung yang berikut: `content_template`

```
"content_template": "{\"inputs\": $prompt, \"parameters\":{\"max_new_tokens\":
 \
    10, \"top_p\": 0.9, \"temperature\": 0.8}}"
```

Jalankan evaluasi model Anda

Tentukan dan jalankan algoritma evaluasi Anda

1. Tentukan algoritma evaluasi Anda. Contoh berikut menunjukkan bagaimana mendefinisikan `SummarizationAccuracy` algoritma, yang digunakan untuk menentukan akurasi untuk tugas ringkasan teks:

   ```
   eval_algo = SummarizationAccuracy(SummarizationAccuracyConfig())
   ```

   Untuk contoh algoritme yang menghitung metrik untuk tugas evaluasi lainnya, lihat Mengevaluasi model Anda. Gunakan `fmeval` pustaka untuk menjalankan evaluasi otomatis
2. Jalankan algoritma evaluasi Anda. Contoh kode berikut menggunakan konfigurasi data yang sebelumnya didefinisikan, dan `prompt_template` yang menggunakan `Human` dan `Assistant` kunci:

   ```
   eval_output = eval_algo.evaluate(model=bedrock_model_runner,
   dataset_config=config,
   prompt_template="Human: $feature\n\nAssistant:\n", save=True)
   ```

   Dalam contoh kode sebelumnya, `feature` berisi prompt dalam format yang diharapkan oleh model Amazon Bedrock.

Melihat hasil analisis Anda

1. Parse laporan evaluasi dari `eval_output` objek yang dikembalikan oleh algoritma evaluasi sebagai berikut:

```
# parse report
print(json.dumps(eval_output, default=vars, indent=4))
```

Perintah sebelumnya mengembalikan output berikut:

```
[
    {
        "eval_name": "summarization_accuracy",
        "dataset_name": "xsum_dataset",
        "dataset_scores": [
            {
                "name": "meteor",
                "value": 0.2048823008681274
            },
            {
                "name": "rouge",
                "value": 0.03557697913367101
            },
            {
                "name": "bertscore",
                "value": 0.5406564395678671
            }
        ],
        "prompt_template": "Human: $feature\n\nAssistant:\n",
        "category_scores": null,
        "output_path": "/tmp/eval_results/
summarization_accuracy_xsum_dataset.jsonl",
        "error": null
    }
]
```

Output contoh sebelumnya menampilkan tiga skor akurasi: Meteor, Rouge, dan BERTScore, input`prompt_template`, a `category_score` jika Anda meminta satu, kesalahan apa pun, dan`output_path`. Anda akan menggunakan `output_path` untuk membuat `Pandas DataFrame` di langkah berikut.

2. Impor hasil Anda dan baca ke dalam`DataFrame`, dan lampirkan skor akurasi ke input model, output model, dan output target sebagai berikut:

```
import pandas as pd

data = []
with open("/tmp/eval_results/summarization_accuracy_xsum_dataset.jsonl", "r") as
 file:
    for line in file:
        data.append(json.loads(line))
df = pd.DataFrame(data)
df['meteor_score'] = df['scores'].apply(lambda x: x[0]['value'])
df['rouge_score'] = df['scores'].apply(lambda x: x[1]['value'])
df['bert_score'] = df['scores'].apply(lambda x: x[2]['value'])
df
```

Dalam pemanggilan ini, contoh kode sebelumnya mengembalikan output berikut (dikontrak untuk singkatnya):

```
model_input     model_output     target_output     prompt     scores
 meteor_score     rouge_score     bert_score
0     John Edward Bates, formerly of Spalding, Linco...     I cannot make any
 definitive judgments, as th...     A former Lincolnshire Police officer carried
 o...     Human: John Edward Bates, formerly of Spalding...     [{'name': 'meteor',
 'value': 0.112359550561797...     0.112360     0.000000     0.543234 ...
1     23 October 2015 Last updated at 17:44 BST\nIt'...     Here are some key
 points about hurricane/trop...     Hurricane Patricia has been rated as a
 categor...     Human: 23 October 2015 Last updated at 17:44 B...     [{'name':
 'meteor', 'value': 0.139822692925566...     0.139823     0.017621     0.426529 ...
2     Ferrari appeared in a position to challenge un...     Here are the key points
 from the article:\n\n...     Lewis Hamilton stormed to pole position at the...
 Human: Ferrari appeared in a position to chall...     [{'name': 'meteor', 'value':
 0.283411142234671...     0.283411     0.064516     0.597001 ...
3     The Bath-born player, 28, has made 36 appearan...     Okay, let me summarize
 the key points from th...     Newport Gwent Dragons number eight Ed Jackson ...
     Human: The Bath-born player, 28, has made 36 a...     [{'name': 'meteor',
 'value': 0.089020771513353...     0.089021     0.000000     0.533514 ...
...
```

Output model Anda mungkin berbeda dari output sampel sebelumnya.

Untuk buku catatan yang berisi contoh kode yang diberikan di bagian ini, lihat bedrock-claude-summarization-accuracy.ipynb.

Notebook tambahan

GitHubDirektori fmeval berisi contoh notebook tambahan berikut:

- bedrock-claude-factual-knowledge.ipynb - Mengevaluasi model Anthropic Claude 2 yang dihosting di Amazon Bedrock untuk pengetahuan faktual.
- byo-model-outputs.ipynb - Mengevaluasi model Falcon 7b yang di-host SageMaker JumpStart untuk pengetahuan faktual di mana Anda membawa output model Anda sendiri alih-alih mengirim permintaan inferensi ke model Anda.
- custom_model_runner_chat_gpt.ipynb - Mengevaluasi model kustom yang di-host untuk pengetahuan faktual. `ChatGPT 3.5` `Hugging Face`

Panduan Pemecahan Masalah FMEval

Foundation Model Evaluations (FMEval) dalam rilis pratinjau untuk Amazon SageMaker Clarify dan dapat berubah sewaktu-waktu.

Important

Untuk menggunakan SageMaker Clarify Foundation Model Evaluations (FMEval), Anda harus meningkatkan ke pengalaman Studio baru.
Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. FMEval tidak tersedia di Amazon SageMaker Studio Classic.
Untuk informasi tentang cara melakukan ini ke pengalaman Studio baru, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Jika Anda mengalami kesalahan saat membuat evaluasi model dasar, gunakan daftar berikut untuk memecahkan masalah evaluasi Anda. Jika Anda memerlukan bantuan lebih lanjut, hubungi AWS Supportatau Forum AWS Pengembang untuk Amazon SageMaker.

Topik

- Kesalahan saat mengunggah data Anda dari bucket Amazon S3
- Pekerjaan pemrosesan gagal diselesaikan
- Anda tidak dapat menemukan evaluasi model dasar di konsol SageMaker
- Model Anda tidak mendukung stereotip yang cepat

Kesalahan saat mengunggah data Anda dari bucket Amazon S3

Saat membuat evaluasi model foundation, Anda harus menetapkan izin yang benar untuk bucket S3 tempat Anda ingin menyimpan input dan output model. Jika izin Cross-origin resource sharing (CORS) tidak disetel dengan benar, buat kesalahan SageMaker berikut:

```
Error: Failed to put object in s3: Error while uploading object to s3Error: Failed to put object in S3: NetworkError when attempting to fetch resource.
```

Untuk menyetel izin bucket yang benar, ikuti petunjuk di bawah Mengatur lingkungan Anda diMenggunakan evaluasi otomatis di UI.

Pekerjaan pemrosesan gagal diselesaikan

Alasan paling umum bahwa pekerjaan pemrosesan Anda gagal diselesaikan adalah sebagai berikut:

- Kuota tak cukup
- Memori tak cukup
- Tidak lulus cek ping

Lihat bagian berikut untuk membantu Anda mengurangi setiap masalah.

Kuota tak cukup

Saat Anda menjalankan evaluasi model dasar untuk model yang tidak digunakan, SageMaker Clarify menyebarkan SageMaker JumpStart model bahasa besar (LLM) Anda ke titik SageMaker akhir di akun Anda. Jika akun Anda tidak memiliki kuota yang cukup untuk menjalankan SageMaker

JumpStart model yang dipilih, pekerjaan gagal dengan a`ClientError`. Untuk menambah kuota, ikuti langkah-langkah berikut:

Minta peningkatan AWS Service Quotas

1. Ambil nama instans, kuota saat ini dan kuota yang diperlukan dari pesan kesalahan di layar. Misalnya, dalam kesalahan berikut:

   - Nama instance-nya adalah`ml.g5.12xlarge`.
   - Kuota saat ini dari nomor berikut `current utilization` adalah `0 instances`
   - Tambahan kuota yang diperlukan dari nomor berikut `request delta` adalah`1 instances`.

   Kesalahan sampel berikut:

   ```
   ClientError: An error occurred (ResourceLimitExceeded) when calling
   the CreateEndpoint operation: The account-level service limit
   'ml.g5.12xlarge for endpoint usage' is 0 Instances, with current
   utilization of 0 Instances and a request delta of 1 Instances. Please
   use AWS Service Quotas to request an increase for this quota. If AWS
   Service Quotas is not available, contact AWS support to request an
   increase for this quota
   ```
2. Masuk ke AWS Management Console dan buka konsol Service Quotas.
3. Di panel navigasi, di bawah Kelola kuota, masukkan. **`Amazon SageMaker`**
4. Pilih Lihat kuota.
5. Di bilah pencarian di bawah Kuota layanan, masukkan nama instance dari Langkah 1. Misalnya, menggunakan informasi yang terkandung dalam pesan kesalahan dari Langkah 1, masukan**`ml.g5.12xlarge`**.
6. Pilih nama Kuota yang muncul di sebelah nama instans Anda dan diakhiri dengan untuk penggunaan titik akhir. Misalnya, menggunakan informasi yang terkandung dalam pesan kesalahan dari Langkah 1, pilih ml.g5.12xlarge untuk penggunaan endpoint.
7. Pilih Permintaan peningkatan di tingkat akun.
8. Di bawah Meningkatkan nilai kuota, masukkan kuota yang diperlukan dari informasi yang diberikan dalam pesan kesalahan dari Langkah 1. Masukan total `current utilization` dan`request delta`. Dalam contoh kesalahan sebelumnya, `current utilization` is`0 Instances`, dan `request delta` is`1 Instances`. Dalam contoh ini, minta kuota 1 untuk memasok kuota yang diperlukan.

9. Pilih Minta.
10. Pilih riwayat permintaan kuota dari panel navigasi.
11. Saat Status berubah dari Tertunda menjadi Disetujui, jalankan kembali pekerjaan Anda. Mungkin Anda perlu menyegarkan peramban untuk melihat perubahannya.

Untuk informasi selengkapnya tentang meminta peningkatan kuota, lihat Meminta kenaikan kuota.

Memori tak cukup

Jika Anda memulai evaluasi model dasar pada instans Amazon EC2 yang tidak memiliki memori yang cukup untuk menjalankan algoritme evaluasi, pekerjaan akan gagal dengan kesalahan berikut:

```
The actor is dead because its worker process has died. Worker exit
type: SYSTEM_ERROR Worker exit detail: Worker unexpectedly exits with
a connection error code 2. End of file. There are some potential root
causes. (1) The process is killed by SIGKILL by OOM killer due to high
memory usage. (2) ray stop --force is called. (3) The worker is crashed
unexpectedly due to SIGSEGV or other unexpected errors. The actor never ran
- it was cancelled before it started running.
```

Untuk meningkatkan memori yang tersedia untuk pekerjaan evaluasi Anda, ubah instance Anda menjadi yang memiliki lebih banyak memori. Jika Anda menggunakan antarmuka pengguna, Anda dapat memilih jenis instans di bawah Konfigurasi prosesor di Langkah 2. Jika Anda menjalankan pekerjaan Anda di dalam SageMaker konsol, luncurkan ruang baru menggunakan instance dengan kapasitas memori yang meningkat.

Untuk daftar instans Amazon EC2, lihat Jenis instans.

Untuk informasi selengkapnya, tentang instans dengan kapasitas memori yang lebih besar, lihat Instans yang dioptimalkan untuk memori.

Tidak lulus cek ping

Dalam beberapa kasus, pekerjaan evaluasi model dasar Anda akan gagal karena tidak lulus pemeriksaan ping saat menerapkan SageMaker titik akhir Anda. Jika tidak lulus tes ping, kesalahan berikut muncul:

```
ClientError: Error hosting endpoint your_endpoint_name: Failed. Reason: The
primary container for production variant AllTraffic did not pass the ping
```

```
health check. Please check CloudWatch logs for this endpoint..., Job exited
for model: your_model_name of model_type:  your_model_type
```

Jika tugas Anda menghasilkan kesalahan ini, tunggu beberapa menit dan jalankan pekerjaan Anda lagi. Jika kesalahan berlanjut, hubungi AWSSupport atau Forum AWS Pengembang untuk Amazon SageMaker.

Anda tidak dapat menemukan evaluasi model dasar di konsol SageMaker

Untuk menggunakan SageMaker Clarify Foundation Model Evaluations, Anda harus meningkatkan ke pengalaman Studio baru. Per 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Fitur evaluasi pondasi hanya dapat digunakan dalam pengalaman yang diperbarui. Untuk informasi tentang cara memperbarui Studio, lihatMigrasi dari Amazon SageMaker Studio Classic. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Model Anda tidak mendukung stereotip yang cepat

Hanya beberapa SageMaker JumpStart model yang mendukung stereotip cepat. Jika Anda memilih SageMaker JumpStart model yang tidak didukung, kesalahan berikut akan muncul:

```
{"evaluationMetrics":"This model does not support Prompt stereotyping
evaluation. Please remove that evaluation metric or select another model
that supports it."}
```

Jika Anda menerima kesalahan ini, Anda tidak dapat menggunakan model yang Anda pilih dalam evaluasi yayasan. SageMaker Clarify saat ini bekerja untuk memperbarui semua SageMaker JumpStart model untuk tugas stereotip yang cepat sehingga dapat digunakan dalam evaluasi model dasar.

Gunakan SageMaker Clarify untuk menjelaskan dan mendeteksi bias

Topik ini menjelaskan cara memahami keadilan dan penjelasan model serta cara menjelaskan dan mendeteksi bias menggunakan Amazon Clarify. SageMaker Anda dapat mengonfigurasi tugas pemrosesan SageMaker Clarify untuk menghitung metrik bias dan atribusi fitur serta menghasilkan laporan untuk penjelasan model. SageMaker Pekerjaan pemrosesan klarifikasi diimplementasikan menggunakan gambar kontainer SageMaker Clarify khusus. Petunjuk berikut menunjukkan cara

mengonfigurasi, menjalankan, dan memecahkan masalah pekerjaan pemrosesan SageMaker Clarify dan cara mengonfigurasi analisis.

Apa keadilan dan penjelasan model untuk prediksi pembelajaran mesin?

Model pembelajaran mesin (ML) membantu membuat keputusan dalam domain termasuk layanan keuangan, perawatan kesehatan, pendidikan, dan sumber daya manusia. Pembuat kebijakan, regulator, dan advokat telah meningkatkan kesadaran tentang tantangan etika dan kebijakan yang ditimbulkan oleh ML dan sistem berbasis data. Amazon SageMaker Clarify dapat membantu Anda memahami mengapa model ML Anda membuat prediksi tertentu dan apakah bias ini memengaruhi prediksi ini selama pelatihan atau inferensi. SageMaker Clarify juga menyediakan alat yang dapat membantu Anda membangun model pembelajaran mesin yang kurang bias dan lebih mudah dipahami. SageMaker Clarify juga dapat menghasilkan laporan tata kelola model yang dapat Anda berikan kepada tim risiko dan kepatuhan serta regulator eksternal. Dengan SageMaker Clarify, Anda dapat melakukan hal berikut:

- Deteksi bias dan bantu jelaskan prediksi model Anda.
- Identifikasi jenis bias dalam data pra-pelatihan.
- Identifikasi jenis bias dalam data pasca-pelatihan yang dapat muncul selama pelatihan atau saat model Anda sedang diproduksi.

SageMaker Clarify membantu menjelaskan bagaimana model Anda membuat prediksi menggunakan atribusi fitur. Ini juga dapat memantau model inferensi yang sedang diproduksi untuk bias dan penyimpangan atribusi fitur. Informasi ini dapat membantu Anda dalam bidang berikut:

- Regulasi — Pembuat kebijakan dan regulator lainnya dapat memiliki kekhawatiran tentang dampak diskriminatif dari keputusan yang menggunakan output dari model ML. Misalnya, model ML dapat mengkodekan bias dan mempengaruhi keputusan otomatis.
- Bisnis — Domain yang diatur mungkin memerlukan penjelasan yang dapat diandalkan tentang bagaimana model ML membuat prediksi. Penjelasan model mungkin sangat penting bagi industri yang bergantung pada keandalan, keamanan, dan kepatuhan. Ini dapat mencakup layanan keuangan, sumber daya manusia, perawatan kesehatan, dan transportasi otomatis. Misalnya, aplikasi pinjaman mungkin perlu memberikan penjelasan tentang bagaimana model ML membuat prediksi tertentu kepada petugas pinjaman, peramal, dan pelanggan.
- Ilmu Data — Ilmuwan data dan insinyur ML dapat men-debug dan meningkatkan model ML ketika mereka dapat menentukan apakah suatu model membuat kesimpulan berdasarkan fitur yang

bising atau tidak relevan. Mereka juga dapat memahami keterbatasan model dan mode kegagalan yang mungkin dihadapi model mereka.

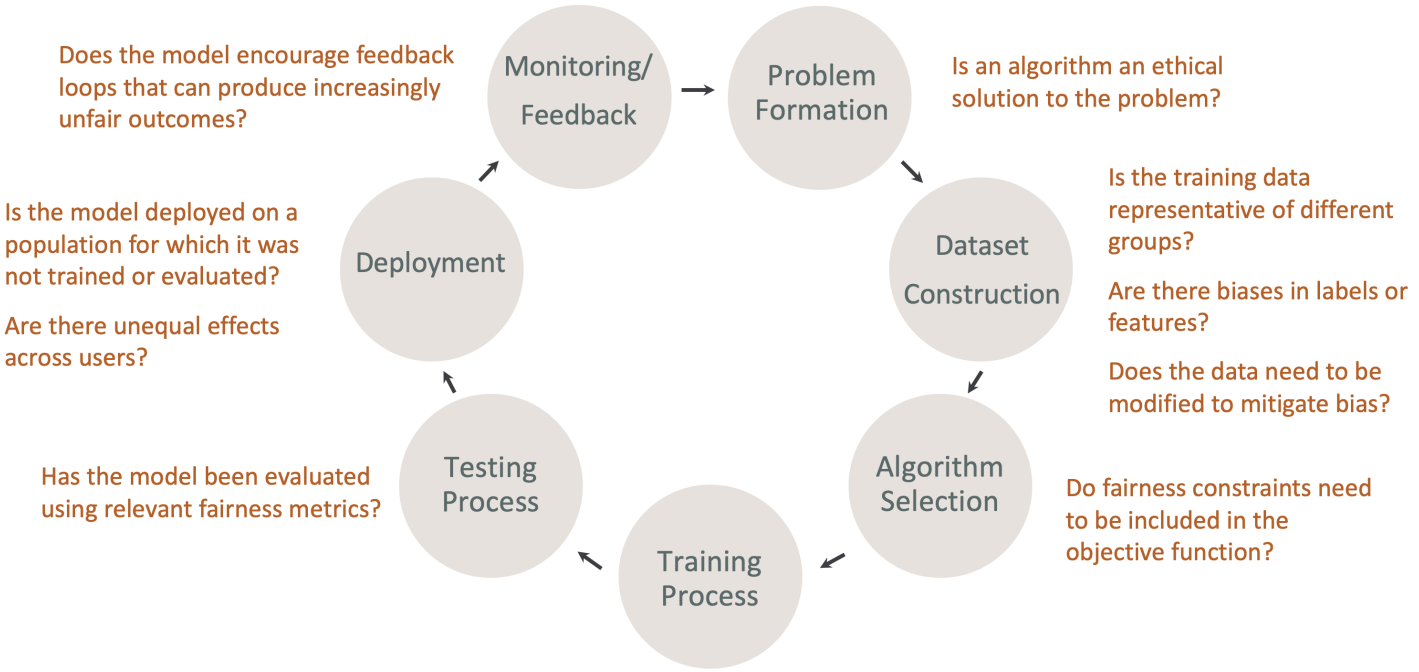
Untuk posting blog yang menunjukkan cara merancang dan membangun model pembelajaran mesin lengkap untuk klaim mobil palsu yang mengintegrasikan SageMaker Clarify ke dalam SageMaker pipeline, lihat Arsitek dan bangun siklus hidup pembelajaran mesin lengkap dengan: AWS Demo Amazon. end-to-end SageMaker Posting blog ini membahas cara menilai dan mengurangi bias pra-pelatihan dan pasca-pelatihan, dan bagaimana fitur memengaruhi prediksi model. Posting blog berisi tautan ke kode contoh untuk setiap tugas dalam siklus hidup ML.

Praktik terbaik untuk mengevaluasi keadilan dan penjelasan dalam siklus hidup ML

Keadilan sebagai proses — Gagasan bias dan keadilan tergantung pada penerapannya. Pengukuran bias dan pilihan metrik bias dapat dipandu oleh pertimbangan sosial, hukum, dan non-teknis lainnya. Keberhasilan adopsi pendekatan ML yang sadar keadilan termasuk membangun konsensus dan mencapai kolaborasi di seluruh pemangku kepentingan utama. Ini mungkin termasuk produk, kebijakan, hukum, teknik, tim AI/ML, pengguna akhir, dan komunitas.

Keadilan dan keterjelasannya berdasarkan desain dalam siklus hidup ML — Pertimbangkan keadilan dan penjelasan selama setiap tahap siklus hidup ML. Tahapan ini meliputi pembentukan masalah, konstruksi dataset, pemilihan algoritma, proses pelatihan model, proses pengujian, penyebaran, dan pemantauan dan umpan balik. Penting untuk memiliki alat yang tepat untuk melakukan analisis ini. Sebaiknya ajukan pertanyaan-pertanyaan berikut selama siklus hidup ML:

- Apakah model mendorong loop umpan balik yang dapat menghasilkan hasil yang semakin tidak adil?
- Apakah algoritma merupakan solusi etis untuk masalah tersebut?
- Apakah data pelatihan mewakili kelompok yang berbeda?
- Apakah ada bias dalam label atau fitur?
- Apakah data perlu dimodifikasi untuk mengurangi bias?
- Apakah kendala keadilan perlu dimasukkan dalam fungsi objektif?
- Apakah model telah dievaluasi menggunakan metrik keadilan yang relevan?
- Apakah ada efek yang tidak setara di seluruh pengguna?
- Apakah model tersebut digunakan pada populasi yang tidak dilatih atau dievaluasi?



Panduan untuk SageMaker penjelasan dan dokumentasi bias

Bias dapat terjadi dan diukur dalam data baik sebelum dan sesudah melatih model. SageMaker Clarify dapat memberikan penjelasan untuk prediksi model setelah pelatihan dan untuk model yang digunakan untuk produksi. SageMaker Clarify juga dapat memantau model dalam produksi untuk setiap penyimpangan dalam atribusi penjelasan dasar mereka, dan menghitung garis dasar bila diperlukan. Dokumentasi untuk menjelaskan dan mendeteksi bias menggunakan SageMaker Clarify disusun sebagai berikut:

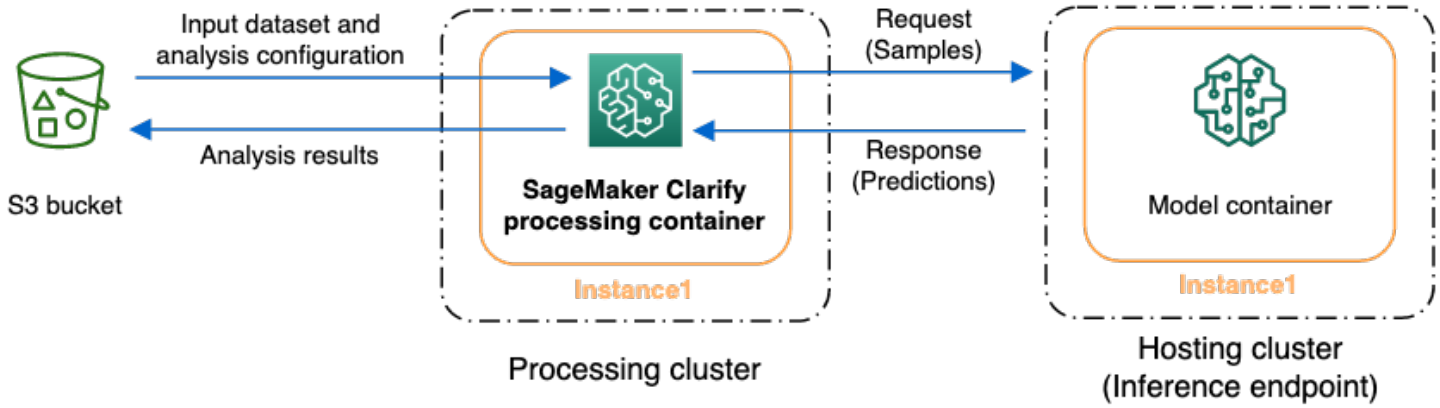
- Untuk informasi tentang menyiapkan pekerjaan pemrosesan untuk bias dan penjelasan, lihat. Konfigurasikan SageMaker Clarify Processing Job
- Untuk informasi tentang mendeteksi bias dalam data pra-pemrosesan sebelum digunakan untuk melatih model, lihat. Mendeteksi Bias Data Pra-pelatihan
- Untuk informasi tentang mendeteksi data pasca-pelatihan dan bias model, lihat. Mendeteksi Data Pasca-pelatihan dan Bias Model
- Untuk informasi tentang pendekatan atribusi fitur model-agnostik untuk menjelaskan prediksi model setelah pelatihan, lihat. Penjelasan Model
- Untuk informasi tentang pemantauan kontribusi fitur yang menjauh dari baseline yang ditetapkan selama pelatihan model, lihat. Monitor Fitur Atribusi Drift untuk Model dalam Produksi
- Untuk informasi tentang model pemantauan yang sedang diproduksi untuk penyimpangan dasar, lihat. Monitor Bias Drift untuk Model dalam Produksi

- Untuk informasi tentang mendapatkan penjelasan secara real time dari SageMaker titik akhir, lihat. Penjelasan Online dengan Clarify SageMaker

Bagaimana SageMaker Memperjelas Pekerjaan Pemrosesan Pekerjaan

Anda dapat menggunakan SageMaker Clarify untuk menganalisis kumpulan data dan model Anda untuk menjelaskan dan bias. Pekerjaan pemrosesan SageMaker Clarify menggunakan container pemrosesan SageMaker Clarify untuk berinteraksi dengan bucket Amazon S3 yang berisi kumpulan data input Anda. Anda juga dapat menggunakan SageMaker Clarify untuk menganalisis model pelanggan yang diterapkan ke titik akhir SageMaker inferensi.

Grafik berikut menunjukkan bagaimana pekerjaan pemrosesan SageMaker Clarify berinteraksi dengan data input Anda dan secara opsional, dengan model pelanggan. Interaksi ini tergantung pada jenis analisis spesifik yang dilakukan. Container pemrosesan SageMaker Clarify memperoleh dataset input dan konfigurasi untuk analisis dari bucket S3. Untuk jenis analisis tertentu, termasuk analisis fitur, wadah pemrosesan SageMaker Clarify harus mengirim permintaan ke wadah model. Kemudian mengambil prediksi model dari respons yang dikirim oleh wadah model. Setelah itu, wadah pemrosesan SageMaker Clarify menghitung dan menyimpan hasil analisis ke bucket S3.



Anda dapat menjalankan tugas pemrosesan SageMaker Clarify pada beberapa tahap dalam siklus hidup alur kerja pembelajaran mesin. SageMaker Clarify dapat membantu Anda menghitung jenis analisis berikut:

- Metrik bias pra-pelatihan. Metrik ini dapat membantu Anda memahami bias dalam data Anda sehingga Anda dapat mengatasinya dan melatih model Anda pada kumpulan data yang lebih adil. Lihat Ukur Bias Pra-pelatihan untuk informasi tentang metrik bias pra-pelatihan. Untuk menjalankan pekerjaan menganalisis metrik bias pra-pelatihan, Anda harus menyediakan kumpulan data dan file konfigurasi analisis JSON. Konfigurasikan Analisis

- Metrik bias pasca-pelatihan. Metrik ini dapat membantu Anda memahami bias apa pun yang diperkenalkan oleh algoritme, pilihan hiperparameter, atau bias apa pun yang tidak terlihat sebelumnya dalam aliran. Untuk informasi lebih lanjut tentang metrik bias pasca-pelatihan, lihat. Ukur Data Pasca-pelatihan dan Bias Model SageMaker Clarify menggunakan prediksi model selain data dan label untuk mengidentifikasi bias. Untuk menjalankan pekerjaan menganalisis metrik bias pasca-pelatihan, Anda harus menyediakan kumpulan data dan file konfigurasi analisis JSON. Konfigurasi harus menyertakan model atau nama titik akhir.
- Nilai yang indah, yang dapat membantu Anda memahami dampak fitur Anda terhadap prediksi model Anda. Untuk informasi selengkapnya tentang nilai-nilai Shapely, lihat. Atribusi Fitur yang Menggunakan Nilai Shapley Fitur ini membutuhkan model yang terlatih.
- Plot ketergantungan sebagian (PDP), yang dapat membantu Anda memahami seberapa besar variabel target prediksi Anda akan berubah jika Anda memvariasikan nilai satu fitur. Untuk informasi selengkapnya tentang PDP, lihat Fitur Analisis plot ketergantungan sebagian (PDP) ini memerlukan model terlatih.

SageMaker Klarifikasi prediksi model kebutuhan untuk menghitung metrik bias pasca-pelatihan dan atribusi fitur. Anda dapat memberikan endpoint atau SageMaker Clarify akan membuat endpoint sementara menggunakan nama model Anda, juga dikenal sebagai titik akhir bayangan. Container SageMaker Clarify menghapus titik akhir bayangan setelah perhitungan selesai. Pada tingkat tinggi, kontainer SageMaker Clarify menyelesaikan langkah-langkah berikut:

1. Memvalidasi input dan parameter.
2. Menciptakan titik akhir bayangan (jika nama model disediakan).
3. Memuat dataset input ke dalam bingkai data.
4. Memperoleh prediksi model dari titik akhir, jika perlu.
5. Menghitung metrik bias dan fitur atribusi.
6. Menghapus titik akhir bayangan.
7. Menghasilkan hasil analisis.

Setelah pekerjaan pemrosesan SageMaker Clarify selesai, hasil analisis akan disimpan di lokasi keluaran yang Anda tentukan dalam parameter keluaran pemrosesan pekerjaan. Hasil ini mencakup file JSON dengan metrik bias dan atribusi fitur global, laporan visual, dan file tambahan untuk atribusi fitur lokal. Anda dapat mengunduh hasil dari lokasi output dan melihatnya.

Untuk informasi tambahan tentang metrik bias, penjelasan, dan cara menafsirkannya, lihat Pelajari Bagaimana Amazon SageMaker Clarify Membantu Mendeteksi Bias, Pengukuran Keadilan untuk Machine Learning in Finance, dan Whitepaper Amazon AI Fairness and Explainability.

Konfigurasikan SageMaker Clarify Processing Job

Untuk menganalisis data dan model Anda untuk bias dan penjelasan menggunakan SageMaker Clarify, Anda harus mengonfigurasi pekerjaan pemrosesan SageMaker Clarify. Panduan ini menunjukkan cara menentukan nama dataset input, nama file konfigurasi analisis, dan lokasi keluaran untuk pekerjaan pemrosesan. Untuk mengkonfigurasi wadah pemrosesan, input pekerjaan, output, sumber daya, dan parameter lainnya, Anda memiliki dua opsi. Anda dapat menggunakan SageMaker `CreateProcessingJob` API, atau menggunakan SageMaker Python SDK API, `SageMaker ClarifyProcessor`

Untuk informasi tentang parameter yang umum untuk semua pekerjaan pemrosesan, lihat SageMaker Referensi Amazon.

Mengonfigurasi tugas pemrosesan SageMaker Clarify menggunakan SageMaker API

Petunjuk berikut menunjukkan cara menyediakan setiap bagian dari konfigurasi spesifik SageMaker Clarify menggunakan `CreateProcessingJob` API.

1. Masukkan pengidentifikasi penelitian seragam (URI) dari gambar kontainer SageMaker Clarify di dalam `AppSpecification` parameter, seperti yang ditunjukkan pada contoh kode berikut.

```
{
    "ImageUri": "the-clarify-container-image-uri"
}
```

> **Note**
>
> URI harus mengidentifikasi image kontainer SageMaker Clarify yang sudah dibuat sebelumnya. `ContainerEntrypoint`dan `ContainerArguments` tidak didukung. Untuk informasi selengkapnya tentang SageMaker Klarifikasi gambar kontainer Lengkap, lihatMemulai dengan SageMaker Clarify Container.

2. Tentukan konfigurasi untuk analisis dan parameter untuk kumpulan data input Anda di dalam `ProcessingInputs` parameter.

a. Tentukan lokasi file konfigurasi analisis JSON, yang mencakup parameter untuk analisis bias dan analisis penjelasan. `InputName`Parameter `ProcessingInput` objek harus **`analysis_config`** seperti yang ditunjukkan pada contoh kode berikut.

```
{
    "InputName": "analysis_config",
    "S3Input": {
        "S3Uri": "s3://your-bucket/analysis_config.json",
        "S3DataType": "S3Prefix",
        "S3InputMode": "File",
        "LocalPath": "/opt/ml/processing/input/config"
    }
}
```

Untuk informasi selengkapnya tentang skema file konfigurasi analisis, lihatKonfigurasikan Analisis.

b. Tentukan lokasi set data input. `InputName`Parameter `ProcessingInput` objek harus`dataset`. Parameter ini opsional jika Anda telah menyediakan “dataset_uri” dalam file konfigurasi analisis. Nilai-nilai berikut diperlukan dalam `S3Input` konfigurasi.

i. `S3Uri`dapat berupa objek Amazon S3 atau prefiks S3.

ii. `S3InputMode`harus dari tipe**`File`**.

iii. `S3CompressionType`harus bertipe `None` (nilai default).

iv. `S3DataDistributionType`harus bertipe `FullyReplicated` (nilai default).

v. `S3DataType`bisa salah satu `S3Prefix` atau`ManifestFile`. Untuk menggunakan`ManifestFile`, `S3Uri` parameter harus menentukan lokasi file manifes yang mengikuti skema dari bagian Referensi SageMaker API S3Uri. File manifes ini harus mencantumkan objek S3 yang berisi data input untuk pekerjaan tersebut.

Kode berikut menunjukkan contoh konfigurasi input.

```
{
    "InputName": "dataset",
    "S3Input": {
        "S3Uri": "s3://your-bucket/your-dataset.csv",
        "S3DataType": "S3Prefix",
        "S3InputMode": "File",
        "LocalPath": "/opt/ml/processing/input/data"
    }
```

```
}
```

3. Tentukan konfigurasi untuk output dari pekerjaan pemrosesan di dalam `ProcessingOutputConfig` parameter. Satu `ProcessingOutput` objek diperlukan dalam `Outputs` konfigurasi. Berikut ini diperlukan dari konfigurasi output:

   a. `OutputName`harus**`analysis_result`**.

   b. `S3Uri`harus menjadi awalan S3 ke lokasi output.

   c. `S3UploadMode` harus diatur ke **`EndOfJob`**.

   Kode berikut menunjukkan contoh konfigurasi keluaran.

```
{
    "Outputs": [{
        "OutputName": "analysis_result",
        "S3Output": {
            "S3Uri": "s3://your-bucket/result/",
            "S3UploadMode": "EndOfJob",
            "LocalPath": "/opt/ml/processing/output"
         }
     }]
}
```

4. Tentukan konfigurasi `ClusterConfig` untuk sumber daya yang Anda gunakan dalam pekerjaan pemrosesan Anda di dalam `ProcessingResources` parameter. Parameter berikut diperlukan di dalam `ClusterConfig` objek.

   a. `InstanceCount`menentukan jumlah instance komputasi di cluster yang menjalankan pekerjaan pemrosesan. Tentukan nilai yang lebih besar dari 1 untuk mengaktifkan pemrosesan terdistribusi.

   b. `InstanceType`mengacu pada sumber daya yang menjalankan pekerjaan pemrosesan Anda. Karena analisis SageMaker SHAP intensif komputasi, menggunakan tipe instance yang dioptimalkan untuk komputasi harus meningkatkan runtime untuk analisis. Pekerjaan pemrosesan SageMaker Clarify tidak menggunakan GPU.

   Kode berikut ini menunjukkan contoh konfigurasi sumber daya.

```
{
    "ClusterConfig": {
         "InstanceCount": 1,
         "InstanceType": "ml.m5.xlarge",
```

```
        "VolumeSizeInGB": 20
    }
}
```

5. Tentukan konfigurasi jaringan yang Anda gunakan dalam pekerjaan pemrosesan Anda di dalam `NetworkConfig` objek. Nilai-nilai berikut diperlukan dalam konfigurasi.

   a. `EnableNetworkIsolation`harus disetel ke `False` (default) sehingga SageMaker Clarify dapat memanggil titik akhir, jika perlu, untuk prediksi.

   b. Jika model atau titik akhir yang Anda berikan ke pekerjaan SageMaker Clarify berada dalam Amazon Virtual Private Cloud (Amazon VPC), maka pekerjaan SageMaker Clarify juga harus dalam VPC yang sama. Tentukan VPC menggunakan. VpcConfig Selain itu, VPC harus memiliki titik akhir ke bucket, layanan, dan layanan Runtime Amazon S3. SageMaker SageMaker

      Jika pemrosesan terdistribusi diaktifkan, Anda juga harus mengizinkan komunikasi antara instance yang berbeda dalam pekerjaan pemrosesan yang sama. Konfigurasikan aturan untuk grup keamanan Anda yang memungkinkan koneksi masuk antara anggota grup keamanan yang sama. Untuk informasi selengkapnya, lihat Berikan Amazon SageMaker Clarify Lowongan Akses ke Sumber Daya di Amazon VPC Anda.

   Kode berikut memberikan contoh konfigurasi jaringan.

```
{
    "EnableNetworkIsolation": False,
    "VpcConfig": {
        ...
    }
}
```

6. Atur waktu maksimum pekerjaan akan berjalan menggunakan `StoppingCondition` parameter. Waktu terpanjang yang dapat dijalankan oleh pekerjaan SageMaker Clarify adalah 7 berhari-hari atau `604800` detik. Jika pekerjaan tidak dapat diselesaikan dalam batas waktu ini, itu akan dihentikan dan tidak ada hasil analisis yang akan diberikan. Sebagai contoh, konfigurasi berikut membatasi waktu maksimum pekerjaan dapat berjalan hingga 3600 detik.

```
{
    "MaxRuntimeInSeconds": 3600
}
```

7. Tentukan peran IAM untuk `RoleArn` parameter. Peran tersebut harus memiliki hubungan kepercayaan dengan Amazon SageMaker. Ini dapat digunakan untuk melakukan operasi SageMaker API yang tercantum dalam tabel berikut. Sebaiknya gunakan kebijakan SageMakerFullAccess terkelola Amazon, yang memberikan akses penuh ke SageMaker. Untuk informasi selengkapnya tentang kebijakan ini, lihatAWSkebijakan terkelola: AmazonSageMakerFullAccess. Jika Anda memiliki kekhawatiran tentang pemberian akses penuh, izin minimal yang diperlukan bergantung pada apakah Anda memberikan model atau nama titik akhir. Menggunakan nama endpoint memungkinkan untuk memberikan lebih sedikit izin untuk. SageMaker

   Tabel berikut berisi operasi API yang digunakan oleh tugas pemrosesan SageMaker Clarify. **X**Di bawah Nama Model dan nama Endpoint mencatat operasi API yang diperlukan untuk setiap input.

Operasi API	Nama model	Nama titik akhir	Untuk apa itu digunakan
ListTags	X		Tag pekerjaan diterapkan ke titik akhir bayangan.
CreateEndpointConfig	X		Buat konfigurasi titik akhir menggunakan nama model yang Anda berikan
CreateEndpoint	X		Buat titik akhir bayangan menggunakan konfigurasi titik akhir.
DescribeEndpoint	X	X	Jelaskan titik akhir untuk statusnya, titik akhir InService harus melayani permintaan.
InvokeEndpoint	X	X	Panggil titik akhir untuk prediksi.

Untuk informasi lebih lanjut tentang izin yang diperlukan, lihat Izin SageMaker API Amazon: Referensi Tindakan, Izin, dan Sumber Daya.

Untuk informasi selengkapnya tentang meneruskan peran SageMaker, lihatPeran Lulus.

Setelah Anda memiliki masing-masing bagian dari konfigurasi pekerjaan pemrosesan, gabungkan mereka untuk mengonfigurasi pekerjaan.

Konfigurasikan pekerjaan pemrosesan SageMaker Clarify menggunakan AWS SDK untuk Python

Contoh kode berikut menunjukkan cara meluncurkan pekerjaan pemrosesan SageMaker Clarify menggunakan AWSSDK untuk Python.

```
sagemaker_client.create_processing_job(
    ProcessingJobName="your-clarify-job-name",
    AppSpecification={
        "ImageUri": "the-clarify-container-image-uri",
    },
    ProcessingInputs=[{
            "InputName": "analysis_config",
            "S3Input": {
                "S3Uri": "s3://your-bucket/analysis_config.json",
                "S3DataType": "S3Prefix",
                "S3InputMode": "File",
                "LocalPath": "/opt/ml/processing/input/config",
            },
        }, {
            "InputName": "dataset",
            "S3Input": {
                "S3Uri": "s3://your-bucket/your-dataset.csv",
                "S3DataType": "S3Prefix",
                "S3InputMode": "File",
                "LocalPath": "/opt/ml/processing/input/data",
            },
        },
    ],
    ProcessingOutputConfig={
        "Outputs": [{
            "OutputName": "analysis_result",
            "S3Output": {
               "S3Uri": "s3://your-bucket/result/",
```

```
                "S3UploadMode": "EndOfJob",
                "LocalPath": "/opt/ml/processing/output",
            },
        }],
    },
    ProcessingResources={
        "ClusterConfig": {
            "InstanceCount": 1,
            "InstanceType": "ml.m5.xlarge",
            "VolumeSizeInGB": 20,
        },
    },
    NetworkConfig={
        "EnableNetworkIsolation": False,
        "VpcConfig": {
            ...
        },
    },
    StoppingCondition={
        "MaxRuntimeInSeconds": 3600,
    },
    RoleArn="arn:aws:iam::<your-account-id>:role/service-role/AmazonSageMaker-
ExecutionRole",
)
```

Untuk contoh buku catatan dengan instruksi untuk menjalankan tugas pemrosesan SageMaker Clarify menggunakan AWS SDK untuk Python, lihat Keadilan dan Keterjelasan dengan SageMaker Clarify using SDK for Python. AWS Bucket S3 apa pun yang digunakan di notebook harus berada di AWS Region yang sama dengan instance notebook yang mengaksesnya.

Konfigurasikan pekerjaan pemrosesan SageMaker Clarify menggunakan SageMaker Python SDK

Anda juga dapat mengonfigurasi pekerjaan pemrosesan SageMaker Clarify menggunakan SageMaker ClarifyProcessorAPI SDK SageMaker Python. Untuk informasi selengkapnya, lihat Jalankan Pekerjaan Pemrosesan SageMaker Klarifikasi untuk Analisis Bias dan Penjelasan.

Topik

- Memulai dengan SageMaker Clarify Container
- Konfigurasikan Analisis
- Panduan Kompatibilitas Format Data

Memulai dengan SageMaker Clarify Container

Amazon SageMaker menyediakan gambar kontainer SageMaker Clarify bawaan yang menyertakan pustaka dan dependensi lain yang diperlukan untuk menghitung metrik bias dan atribusi fitur agar dapat dijelaskan. Gambar ini telah diaktifkan untuk berjalan SageMaker Memproses data di akun Anda.

URI gambar untuk wadah dalam bentuk berikut:

```
<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/sagemaker-clarify-processing:1.0
```

Sebagai contoh:

```
205585389593.dkr.ecr.us-east-1.amazonaws.com/sagemaker-clarify-processing:1.0
```

Tabel berikut menjelaskan alamat olehWilayah AWS.

Gambar Docker untuk Pekerjaan Pemrosesan SageMaker Klarifikasi

Wilayah	Alamat gambar
us-east-1	205585389593.dkr. ecr.us-east-1.amazonaws.com /:1.0 sagemaker-clarify-processing
us-east-2	211330385671.dkr. ecr.us-east-2.amazonaws.com /:1.0 sagemaker-clarify-processing
us-west-1	740489534195.dkr. ecr.us-west-1.amazonaws.com /:1.0 sagemaker-clarify-processing
us-west-2	306415355426.dkr. ecr.us-west-2.amazonaws.com /:1.0 sagemaker-clarify-processing
ap-east-1	098760798382.dkr. ecr.ap-east-1.amazonaws.com /:1.0 sagemaker-clarify-processing
ap-south-1	452307495513.dkr. ecr.ap-south-1.amazonaws.com /:1.0 sagemaker-clarify-processing

Wilayah	Alamat gambar
ap-southeast-3	705930551576.dkr. ecr.ap-southeast-3.amazonaws.com /:1.0 sagemaker -clarify-processing
ap-northeast-1	377024640650.dkr. ecr.ap-northeast-1.amazonaws.com /:1.0 sagemaker -clarify-processing
ap-northeast-2	263625296855.dkr. ecr.ap-northeast-2.amazonaws.com /:1.0 sagemaker -clarify-processing
ap-northeast-3	912233562940.dkr. ecr.ap-northeast-3.amazonaws.com /:1.0 sagemaker -clarify-processing
ap-southeast-1	834264404009.dkr. ecr.ap-southeast-1.amazonaws.com /:1.0 sagemaker -clarify-processing
ap-southeast-2	007051062584.dkr. ecr.ap-southeast-2.amazonaws.com /:1.0 sagemaker -clarify-processing
ca-central-1	675030665977.dkr. ecr.ca-central-1.amazonaws.com /:1.0 sagemaker-clarify-processing
eu-central-1	017069133835.dkr. ecr.eu-central-1.amazonaws.com /:1.0 sagemaker-clarify-processing
eu-west-1	131013547314.dkr. ecr.eu-west-1.amazonaws.com /:1.0 sagemaker-clarify-processing
eu-west-2	44079696970383.dkr. ecr.eu-west-2.amazonaws.com /:1.0 sagemaker-clarify-processing
eu-west-3	341593696636.dkr. ecr.eu-west-3.amazonaws.com /:1.0 sagemaker-clarify-processing
eu-north-1	763603941244.dkr. ecr.eu-north-1.amazonaws.com /:1.0 sagemaker-clarify-processing
me-south-1	835444307964.dkr. ecr.me-south-1.amazonaws.com /:1.0 sagemaker-clarify-processing

Wilayah	Alamat gambar
sa-east-1	520018980103.dkr. ecr.sa-east-1.amazonaws.com /:1.0 sagemaker-clarify-processing
af-south-1	811711786498.dkr. ecr.af-south-1.amazonaws.com /:1.0 sagemaker-clarify-processing
eu-south-1	638885417683.dkr. ecr.eu-south-1.amazonaws.com /:1.0 sagemaker-clarify-processing
cn-north-1	122526803553.dkr. ecr.cn-north-1.amazonaws.com .cn/:1.0 sagemaker-clarify-processing
cn-northwest-1	122578899357.dkr. ecr.cn-northwest-1.amazonaws.com .cn/:1.0 sagemaker-clarify-processing

Konfigurasikan Analisis

Untuk menganalisis data dan model Anda untuk menjelaskan dan bias menggunakan SageMaker Clarify, Anda harus mengonfigurasi pekerjaan pemrosesan. Bagian dari konfigurasi untuk pekerjaan pemrosesan ini mencakup konfigurasi file analisis. File analisis menentukan parameter untuk analisis bias dan penjelasan. Lihat Konfigurasikan SageMaker Clarify Processing Job untuk mempelajari cara mengonfigurasi pekerjaan pemrosesan dan file analisis.

Panduan ini menjelaskan skema dan parameter untuk file konfigurasi analisis ini. Panduan ini juga mencakup contoh file konfigurasi analisis untuk metrik bias komputasi untuk kumpulan data tabel, dan menghasilkan penjelasan untuk masalah pemrosesan bahasa alami (NLP) dan visi komputer (CV).

Anda dapat membuat file konfigurasi analisis atau menggunakan SageMaker Python SDK untuk menghasilkan satu untuk Anda dengan API. SageMaker ClarifyProcessor Melihat isi file dapat membantu untuk memahami konfigurasi dasar yang digunakan oleh tugas SageMaker Clarify.

Topik

- Skema untuk file konfigurasi analisis
- Contoh file konfigurasi analisis

Skema untuk file konfigurasi analisis

Bagian berikut menjelaskan skema untuk file konfigurasi analisis termasuk persyaratan dan deskripsi parameter.

Persyaratan untuk file konfigurasi analisis

Pekerjaan pemrosesan SageMaker Clarify mengharapkan file konfigurasi analisis terstruktur dengan persyaratan berikut:

- Nama input pemrosesan harus `analysis_config`.
- File konfigurasi analisis dalam format JSON, dan dikodekan dalam UTF-8.
- File konfigurasi analisis adalah objek Amazon S3.

Anda dapat menentukan parameter tambahan dalam file konfigurasi analisis. Bagian berikut menyediakan berbagai opsi untuk menyesuaikan pekerjaan pemrosesan SageMaker Clarify untuk kasus penggunaan Anda dan jenis analisis yang diinginkan.

Parameter untuk file konfigurasi analisis

Dalam file konfigurasi analisis, Anda dapat menentukan parameter berikut.

- versi - (Opsional) String versi skema file konfigurasi analisis. Jika versi tidak disediakan, SageMaker Clarify menggunakan versi terbaru yang didukung. Saat ini, satu-satunya versi yang didukung adalah`1.0`.
- dataset_type — Format dataset. Format set data input dapat berupa salah satu dari nilai berikut:
  - `text/csv`untuk CSV
  - `application/jsonlines`untuk format padat SageMaker JSON Lines
  - `application/json`untuk JSON
  - `application/x-parquet`untuk Apache Parquet
  - `application/x-image`untuk mengaktifkan penjelasan untuk masalah penglihatan komputer
- dataset_uri — (Opsional) Pengidentifikasi sumber daya seragam (URI) dari kumpulan data utama. Jika Anda memberikan awalan URI S3, pekerjaan pemrosesan SageMaker Clarify secara rekursif mengumpulkan semua file S3 yang terletak di bawah awalan. Anda dapat memberikan awalan URI S3 atau URI S3 ke file manifes gambar untuk masalah penglihatan komputer. Jika `dataset_uri` disediakan, itu lebih diutamakan daripada input pekerjaan pemrosesan dataset. Untuk jenis format apa pun kecuali gambar, pekerjaan pemrosesan SageMaker Clarify memuat kumpulan data input

ke dalam bingkai data tabel, sebagai kumpulan data tabel. Format ini memungkinkan SageMaker untuk dengan mudah memanipulasi dan menganalisis dataset input.

- header — (Opsional) Sebuah array string yang berisi nama kolom dari dataset tabular. Jika nilai untuk tidak `headers` diberikan, maka pekerjaan pemrosesan SageMaker Clarify membaca header dari kumpulan data. Jika kumpulan data tidak memiliki header, maka pekerjaan pemrosesan SageMaker Clarify akan secara otomatis menghasilkan nama placeholder berdasarkan indeks kolom berbasis nol. Sebagai contoh, nama placeholder untuk kolom pertama dan kedua adalah,. **`column_0` `column_1`**

  > **Note**
  >
  > Menurut konvensi, jika dataset_type adalah `application/jsonlines` atau `application/json` kemudian `headers` harus berisi nama-nama berikut secara berurutan: nama fitur, nama label (jika `label` ditentukan), dan nama label yang diprediksi (jika `predicted_label` ditentukan). Contoh `headers` untuk tipe `application/jsonlines` dataset jika ditentukan `label` adalah:`["feature1","feature2","feature3","target_label"]`.

- label — (Opsional) Sebuah string atau indeks integer berbasis nol. Jika disediakan, `label` digunakan untuk menemukan label kebenaran dasar, juga dikenal sebagai label yang diamati atau atribut target dalam kumpulan data tabel. Label kebenaran dasar digunakan untuk menghitung metrik bias. Nilai untuk `label` ditentukan tergantung pada nilai `dataset_type` parameter sebagai berikut.
  - Jika `dataset_type` ya**`text/csv`**, `label` dapat ditentukan sebagai salah satu dari berikut ini:
    - Nama kolom valid
    - Indeks yang berada dalam rentang kolom dataset
  - Jika `dataset_type` ya**`application/parquet`**, `label` harus nama kolom yang valid.
  - Jika `dataset_type` ya**`application/jsonlines`**, `label` harus berupa ekspresi JMESPath yang ditulis untuk mengekstrak label kebenaran dasar dari kumpulan data. Dengan konvensi, jika `headers` ditentukan, maka harus berisi nama label.
  - Jika `dataset_type` ya**`application/json`**, `label` harus berupa ekspresi JMESPath yang ditulis untuk mengekstrak label kebenaran dasar untuk setiap catatan dalam kumpulan data. Ekspresi JMESPath ini harus menghasilkan daftar label di mana label $^{ke\text{-}i}$ berkorelasi dengan catatan ke-i.

- predicted_label — (Opsional) Sebuah string atau indeks integer berbasis nol. Jika disediakan, `predicted_label` digunakan untuk menemukan kolom yang berisi label yang diprediksi dalam

kumpulan data tabel. Label yang diprediksi digunakan untuk menghitung metrik bias pasca-pelatihan. Parameter `predicted_label` ini opsional jika kumpulan data tidak menyertakan label yang diprediksi. Jika label yang diprediksi diperlukan untuk komputasi, maka pekerjaan pemrosesan SageMaker Clarify akan mendapatkan prediksi dari model.

Nilai untuk `predicted_label` ditentukan tergantung pada nilai `dataset_type` sebagai berikut:

- Jika `dataset_type` ya**`text/csv`**, `predicted_label` dapat ditentukan sebagai salah satu dari berikut ini:
  - Nama kolom yang valid. Jika `predicted_label_dataset_uri` ditentukan, tetapi tidak `predicted_label` disediakan, nama label prediksi default adalah “predicted_label”.
  - Indeks yang berada dalam rentang kolom dataset. Jika `predicted_label_dataset_uri` ditentukan, maka indeks digunakan untuk menemukan kolom label yang diprediksi dalam kumpulan data label yang diprediksi.
- Jika dataset_type adalah**`application/x-parquet`**, `predicted_label` harus nama kolom yang valid.
- Jika dataset_type adalah**`application/jsonlines`**, `predicted_label` harus berupa ekspresi JMESPath valid yang ditulis untuk mengekstrak label yang diprediksi dari dataset. Dengan konvensi, jika `headers` ditentukan, maka harus berisi nama label yang diprediksi.
- Jika `dataset_type` ya**`application/json`**, `predicted_label` harus berupa ekspresi JMESPath yang ditulis untuk mengekstrak label yang diprediksi untuk setiap catatan dalam kumpulan data. Ekspresi JMESPath harus menghasilkan daftar label yang diprediksi di mana label prediksi $^{ke\text{-}i}$ adalah untuk catatan $^{ke\text{-}i}$.

- fitur - Diperlukan jika dataset_type adalah atau. `application/jsonlines` `application/json` Ekspresi string JMESPath ditulis untuk menemukan fitur dalam dataset input. Untuk`application/jsonlines`, ekspresi JMESPath akan diterapkan ke setiap baris untuk mengekstrak fitur untuk catatan itu. Untuk`application/json`, ekspresi JMESPath akan diterapkan ke seluruh dataset input. Ekspresi JMESPath harus mengekstrak daftar daftar, atau array 2D/matriks fitur di mana baris $^{ke\text{-}i}$ berisi fitur yang berkorelasi dengan catatan $^{ke\text{-}i}$. Untuk `dataset_type` dari `text/csv` atau`application/x-parquet`, semua kolom kecuali label kebenaran dasar dan kolom label yang diprediksi secara otomatis ditetapkan sebagai fitur.
- predicted_label_dataset_uri — Hanya berlaku ketika dataset_type adalah. `text/csv` URI S3 untuk kumpulan data yang berisi label prediksi yang digunakan untuk menghitung metrik bias pasca-pelatihan. Pekerjaan pemrosesan SageMaker Clarify akan memuat prediksi dari URI yang disediakan alih-alih mendapatkan prediksi dari model. Dalam hal ini, `predicted_label` diperlukan untuk menemukan kolom label yang diprediksi dalam kumpulan data label yang

diprediksi. Jika kumpulan data label yang diprediksi atau kumpulan data utama dibagi menjadi beberapa file, kolom pengidentifikasi harus ditentukan oleh `joinsource_name_or_index` untuk bergabung dengan dua kumpulan data.

- predicted_label_headers - Hanya berlaku bila ditentukan. `predicted_label_dataset_uri` Array string yang berisi nama kolom dari kumpulan data label yang diprediksi. Selain header label yang diprediksi, juga `predicted_label_headers` dapat berisi header kolom pengidentifikasi untuk bergabung dengan kumpulan data label yang diprediksi dan kumpulan data utama. Untuk informasi selengkapnya, lihat deskripsi berikut untuk parameter`joinsource_name_or_index`.
- joinsource_name_or_index — Nama atau indeks berbasis nol dari kolom dalam kumpulan data tabular yang akan digunakan sebagai kolom pengenal saat melakukan penggabungan bagian dalam. Kolom ini hanya digunakan sebagai pengenal. Ini tidak digunakan untuk perhitungan lain seperti analisis bias atau analisis atribusi fitur. Nilai untuk `joinsource_name_or_index` diperlukan dalam kasus-kasus berikut:
  - Ada beberapa kumpulan data input, dan siapa pun dibagi menjadi beberapa file.
  - Pemrosesan terdistribusi diaktifkan dengan mengatur pekerjaan pemrosesan SageMaker Clarify InstanceCountke nilai yang lebih besar dari`1`.
- excluded_columns — (Opsional) Sebuah array nama atau indeks kolom berbasis nol yang akan dikecualikan dari dikirim ke model sebagai masukan untuk prediksi. Label kebenaran dasar dan label yang diprediksi secara otomatis sudah dikecualikan.
- probability_threshold — (Opsional) Nomor floating point di atasnya, label atau objek dipilih. Nilai default-nya adalah `0.5`. Pekerjaan pemrosesan SageMaker Clarify digunakan `probability_threshold` dalam kasus-kasus berikut:
  - Dalam analisis bias pasca-pelatihan, `probability_threshold` ubah prediksi model numerik (nilai probabilitas atau skor) menjadi label biner, jika modelnya adalah pengklasifikasi biner. Skor yang lebih besar dari ambang batas dikonversi menjadi`1`. Sedangkan, skor kurang dari atau sama dengan ambang batas diubah menjadi`0`.
  - Dalam masalah penjelasan visi komputer, jika model_type **`OBJECT_DETECTION`**`,` `probability_threshold` menyaring objek yang terdeteksi dengan skor kepercayaan lebih rendah dari nilai ambang batas.
- label_values_or_threshold — Diperlukan untuk analisis bias. Array nilai label atau nomor ambang batas, yang menunjukkan hasil positif untuk kebenaran dasar dan label prediksi untuk metrik bias. Untuk informasi selengkapnya, lihat nilai label positif diAmazon SageMaker Klarifikasi Persyaratan untuk Bias dan Keadilan. Jika labelnya numerik, ambang batas diterapkan sebagai batas bawah untuk memilih hasil positif. `label_values_or_threshold`Untuk mengatur berbagai jenis masalah, lihat contoh berikut:

- Untuk masalah klasifikasi biner, label memiliki dua nilai yang mungkin, `0` dan`1`. Jika nilai `1` label menguntungkan untuk kelompok demografis yang diamati dalam sampel, maka `label_values_or_threshold` harus diatur ke`[1]`.
- Untuk masalah klasifikasi multiclass, label memiliki tiga nilai yang mungkin,**`bird`**, **`cat`** dan. **`dog`** Jika dua yang terakhir mendefinisikan kelompok demografis yang disukai bias, maka `label_values_or_threshold` harus diatur ke. `["cat","dog"]`
- Untuk masalah regresi, nilai label kontinu, mulai dari `0` hingga. `1` Jika nilai yang lebih besar dari `0.5` seharusnya menunjuk sampel sebagai memiliki hasil positif, maka `label_values_or_threshold` harus diatur ke`0.5`.

- facet — Diperlukan untuk analisis bias. Array objek faset, yang terdiri dari atribut sensitif yang mengukur bias. Anda dapat menggunakan aspek untuk memahami karakteristik bias dari kumpulan data dan model Anda bahkan jika model Anda dilatih tanpa menggunakan atribut sensitif. Untuk informasi selengkapnya, lihat Aspet diAmazon SageMaker Klarifikasi Persyaratan untuk Bias dan Keadilan. Setiap objek faset mencakup kolom-kolom berikut:
  - name_or_index — Nama atau indeks berbasis nol dari kolom atribut sensitif dalam kumpulan data tabular. Jika `facet_dataset_uri` ditentukan, maka indeks mengacu pada dataset faset alih-alih dataset utama.
  - value_or_threshold - Diperlukan jika `facet` numerik dan `label_values_or_threshold` diterapkan sebagai batas bawah untuk memilih grup sensitif). Array nilai faset atau angka ambang batas, yang menunjukkan kelompok demografis sensitif yang disukai bias. Jika tipe data facet kategoris dan tidak `value_or_threshold` disediakan, metrik bias dihitung sebagai satu grup untuk setiap nilai unik (bukan semua nilai). `value_or_threshold`Untuk mengatur tipe `facet` data yang berbeda, lihat contoh berikut:
    - Untuk tipe data facet biner, fitur ini memiliki dua nilai yang mungkin, `0` dan`1`. Jika Anda ingin menghitung metrik bias untuk setiap nilai, maka `value_or_threshold` dapat dihilangkan atau diatur ke array kosong.
    - Untuk tipe data facet kategoris, fitur ini memiliki tiga nilai yang mungkin**`bird`**,, **`cat`** dan. **`dog`** Jika dua yang pertama mendefinisikan kelompok demografis yang disukai bias, maka `value_or_threshold` harus diatur ke. `["bird", "cat"]` Dalam contoh ini, sampel dataset dibagi menjadi dua kelompok demografis. Aspek dalam kelompok yang diuntungkan memiliki nilai **`bird`** atau**`cat`**, sedangkan segi dalam kelompok yang kurang beruntung memiliki nilai. **`dog`**
    - Untuk tipe data facet numerik, nilai fitur kontinu, mulai dari `0` hingga. `1` Sebagai contoh, jika nilai yang lebih besar dari `0.5` seharusnya menunjuk sampel sebagai disukai, maka

`value_or_threshold` harus diatur ke. `0.5` Dalam contoh ini, sampel dataset dibagi menjadi dua kelompok demografis. Aspek dalam kelompok yang diuntungkan memiliki nilai lebih besar dari`0.5`, sedangkan segi dalam kelompok yang kurang beruntung memiliki nilai kurang dari atau sama dengan. `0.5`

- group_variable — Nama atau indeks berbasis nol dari kolom yang menunjukkan subkelompok yang akan digunakan untuk metrik bias atau. Disparitas Demografis Bersyarat (CDD) Disparitas Demografis Bersyarat dalam Label yang Diprediksi (CDDPL)
- facet_dataset_uri - Hanya berlaku ketika dataset_type adalah. `text/csv` URI S3 untuk kumpulan data yang berisi atribut sensitif untuk analisis bias. Anda dapat menggunakan aspek untuk memahami karakteristik bias dari kumpulan data dan model Anda bahkan jika model Anda dilatih tanpa menggunakan atribut sensitif.

> **Note**
>
> Jika kumpulan data faset atau kumpulan data utama dibagi menjadi beberapa file, kolom pengidentifikasi harus ditentukan oleh `joinsource_name_or_index` untuk bergabung dengan dua kumpulan data. Anda harus menggunakan parameter `facet` untuk mengidentifikasi setiap aspek dalam dataset faset.

- facet_headers — (Hanya berlaku bila `facet_dataset_uri` ditentukan) Sebuah array string yang berisi nama kolom untuk dataset faset, dan opsional, header kolom pengidentifikasi untuk bergabung dengan dataset faset dan dataset utama, lihat. `joinsource_name_or_index`
- Metode — Objek yang berisi satu atau lebih metode analisis dan parameternya. Jika ada metode yang dihilangkan, itu tidak digunakan untuk analisis atau dilaporkan.
  - pre_training_bias — Sertakan metode ini jika Anda ingin menghitung metrik bias pra-pelatihan. Penjelasan rinci tentang metrik dapat ditemukan diUkur Bias Pra-pelatihan. Objek memiliki parameter berikut:
    - method — Array yang berisi salah satu metrik bias pra-pelatihan dari daftar berikut yang ingin Anda hitung. Setel `methods` **`all`** untuk menghitung semua metrik bias pra-pelatihan. Sebagai contoh, array `["CI", "DPL"]` akan menghitung Ketidakseimbangan Kelas dan Perbedaan dalam Proporsi Label.
      - `CI`untuk Ketidakseimbangan Kelas (CI)
      - `DPL`untuk Perbedaan Proporsi Label (DPL)
      - `KL`untuk Divergensi Kullback-Leibler (KL)
      - `JS`untuk Divergensi Jensen-Shannon (JS)

    - `LP`untuk L $_p$ -norma (LP)
    - `TVD`untuk Jarak Variasi Total (TVD)
    - `KS`untuk Kolmogorov-Smirnov (KS)
    - `CDDL`untuk Disparitas Demografis Bersyarat (CDD)
- post_training_bias — Sertakan metode ini jika Anda ingin menghitung metrik bias pasca-pelatihan. Penjelasan rinci tentang metrik dapat ditemukan diUkur Data Pasca-pelatihan dan Bias Model. `post_training_bias`Objek memiliki parameter berikut.
    - method — Array yang berisi salah satu metrik bias pasca-pelatihan dari daftar berikut yang ingin Anda hitung. Setel `methods` **`all`** untuk menghitung semua metrik bias pasca-pelatihan. Sebagai contoh, array `["DPPL", "DI"]` menghitung Perbedaan Proporsi Positif dalam Label yang Diprediksi dan Dampak Berbeda. Metode yang tersedia adalah sebagai berikut.
        - `DPPL`untuk Perbedaan Proporsi Positif pada Label Prediksi (DPPL)
        - `DI`untuk Dampak Berbeda (DI)
        - `DCA`untuk Perbedaan dalam Penerimaan Bersyarat (DCAcc)
        - `DCR`untuk Perbedaan Penolakan Bersyarat (DCR)
        - `SD`untuk Perbedaan spesifisitas (SD)
        - `RD`untuk Ingat Perbedaan (RD)
        - `DAR`untuk Perbedaan Tingkat Penerimaan (DAR)
        - `DRR`untuk Perbedaan Tingkat Penolakan (DRR)
        - `AD`untuk Perbedaan Akurasi (AD)
        - `TE`untuk Kesetaraan Perawatan (TE)
        - `CDDPL`untuk Disparitas Demografis Bersyarat dalam Label yang Diprediksi (CDDPL)
        - `FT`untuk Fliptest Kontrafaktual (FT)
        - `GE`untuk Entropi umum (GE)
- shap - Sertakan metode ini jika Anda ingin menghitung nilai SHAP. Pekerjaan pemrosesan SageMaker Clarify mendukung algoritma Kernel SHAP. `shap`Objek memiliki parameter berikut.
    - baseline — (Opsional) Kumpulan data dasar SHAP, juga dikenal sebagai dataset latar belakang. Persyaratan tambahan untuk kumpulan data dasar dalam kumpulan data tabular atau masalah penglihatan komputer adalah sebagai berikut. Untuk informasi selengkapnya tentang SHAP Baseline, lihat. Garis Dasar SHAP untuk Penjelasan
        - Untuk kumpulan data tabular, `baseline` dapat berupa data dasar di tempat atau URI S3 dari file dasar. Jika tidak `baseline` disediakan, pekerjaan pemrosesan SageMaker Clarify

menghitung baseline dengan mengelompokkan kumpulan data input. Berikut ini diperlukan dari baseline:

- Formatnya harus sama dengan format kumpulan data yang ditentukan oleh`dataset_type`.
- Garis dasar hanya dapat berisi fitur yang dapat diterima model sebagai input.
- Baseset dasar dapat memiliki satu atau beberapa instans. Jumlah instance dasar secara langsung memengaruhi ukuran kumpulan data sintetis dan runtime pekerjaan.
- Jika `text_config` ditentukan, maka nilai dasar kolom teks adalah string yang digunakan untuk menggantikan unit teks yang ditentukan oleh. `granularity` Misalnya, satu placeholder umum adalah “[MASK]”, yang digunakan untuk mewakili kata atau potongan teks yang hilang atau tidak dikenal.

Contoh-contoh berikut menunjukkan cara mengatur data baseline di tempat untuk parameter yang berbeda: `dataset_type`

- Jika `dataset_type` salah satu `text/csv` atau`application/x-parquet`, model menerima empat fitur numerik, dan baseline memiliki dua contoh. Dalam contoh ini, jika satu catatan memiliki semua nilai fitur nol dan catatan lainnya memiliki semua satu nilai fitur, maka baseline harus diatur ke`[[0,0,0,0],[1,1,1,1]]`, tanpa header apa pun.
- Jika `dataset_type` ya`application/jsonlines`, dan `features` merupakan kunci untuk daftar empat nilai fitur numerik. Selain itu, dalam contoh ini, jika baseline memiliki satu catatan dari semua nilai nol, maka `baseline` seharusnya. `[{"features": [0,0,0,0]}]`
- Jika `dataset_type` ya`application/json`, `baseline` dataset harus memiliki struktur dan format yang sama dengan dataset input.

- Untuk masalah penglihatan komputer, `baseline` bisa berupa URI S3 dari gambar yang digunakan untuk menutupi fitur (segmen) dari gambar input. Pekerjaan pemrosesan SageMaker Clarify memuat gambar topeng dan mengubah ukurannya ke resolusi yang sama dengan gambar input. Jika baseline tidak disediakan, tugas pemrosesan SageMaker Clarify menghasilkan gambar topeng white noise pada resolusi yang sama dengan gambar input.

- features_to_explain — (Opsional) Sebuah array string atau indeks berbasis nol dari kolom fitur untuk menghitung nilai SHAP untuk. Jika tidak `features_to_explain` disediakan, nilai SHAP dihitung untuk semua kolom fitur. Kolom fitur ini tidak dapat menyertakan kolom label atau kolom label yang diprediksi. `features_to_explain`Parameter ini hanya didukung untuk kumpulan data tabular dengan kolom numerik dan kategoris.

- num_clusters — (Opsional) Jumlah cluster yang kumpulan data dibagi menjadi untuk menghitung dataset dasar. Setiap cluster digunakan untuk menghitung satu instance dasar. Jika tidak `baseline` ditentukan, pekerjaan pemrosesan SageMaker Clarify mencoba untuk menghitung kumpulan data dasar dengan membagi kumpulan data tabular menjadi jumlah cluster yang optimal antara dan. `1` `12` Jumlah instance dasar secara langsung mempengaruhi runtime analisis SHAP.
- num_samples — (Opsional) Jumlah sampel yang akan digunakan dalam algoritma Kernel SHAP. Jika tidak `num_samples` disediakan, pekerjaan pemrosesan SageMaker Clarify memilih nomor untuk Anda. Jumlah sampel secara langsung mempengaruhi ukuran dataset sintetis dan runtime pekerjaan.
- seed — (Opsional) Bilangan bulat yang digunakan untuk menginisialisasi generator bilangan acak semu di penjelasan SHAP untuk menghasilkan nilai SHAP yang konsisten untuk pekerjaan yang sama. Jika seed tidak ditentukan, maka setiap kali pekerjaan yang sama berjalan, model dapat menampilkan nilai SHAP yang sedikit berbeda.
- use_logit — (Opsional) Nilai Boolean yang menunjukkan bahwa Anda ingin fungsi logit diterapkan pada prediksi model. Default ke `false`. Jika `use_logit` yatrue, maka nilai SHAP dihitung menggunakan koefisien regresi logistik, yang dapat diartikan sebagai rasio log-odds.
- save_local_shap_values — (Opsional) Nilai Boolean yang menunjukkan bahwa Anda ingin nilai SHAP lokal dari setiap record dalam dataset disertakan dalam hasil analisis. Default ke `false`.

  Jika kumpulan data utama dibagi menjadi beberapa file atau pemrosesan terdistribusi diaktifkan, tentukan juga kolom pengidentifikasi menggunakan parameter. `joinsource_name_or_index` Kolom pengidentifikasi dan nilai SHAP lokal disimpan dalam hasil analisis. Dengan cara ini, Anda dapat memetakan setiap catatan ke nilai SHAP lokalnya.
- agg_method — (Opsional) Metode yang digunakan untuk menggabungkan nilai SHAP lokal (nilai SHAP untuk setiap instance) dari semua instance ke nilai SHAP global (nilai SHAP untuk seluruh kumpulan data). Default ke `mean_abs`. Metode berikut dapat digunakan untuk menggabungkan nilai SHAP.
  - mean_abs — Rata-rata nilai SHAP lokal absolut dari semua instance.
  - mean_sq — Rata-rata nilai SHAP lokal kuadrat dari semua instance.
  - median — Median nilai SHAP lokal dari semua instance.
- text_config - Diperlukan untuk penjelasan pemrosesan bahasa alami. Sertakan konfigurasi ini jika Anda ingin memperlakukan kolom teks sebagai teks dan penjelasan harus disediakan

untuk masing-masing unit teks. Untuk contoh konfigurasi analisis untuk penjelasan pemrosesan bahasa alami, lihat Konfigurasi analisis untuk menjelaskan pemrosesan bahasa alami

- granularitas — Satuan granularitas untuk analisis kolom teks. Nilai yang valid adalah `token`, `sentence`, atau `paragraph`. Setiap unit teks dianggap sebagai fitur, dan nilai SHAP lokal dihitung untuk setiap unit.
- bahasa — Bahasa kolom teks. Nilai yang valid adalah **`chinesedanish`**,**`dutch`**, **`englishfrench`**,**`german`**,**`greek`**,**`italian`**,**`japanese`**,**`lithuanian`**,**`multi-language`**,**`norwegian bokmål`**,**`polish`**,**`portuguese`**,**`romanian`**,**`russian`**,**`spanish`**, **`afrikaansalbanian`**,**`arabic`**,**`armenian`**,**`basque`**,**`bengali`**,**`bulgarian`**,**`catalan`**,**`croatian`**,**`cz`** **`hindihungarian`**,**`icelandic`**,**`indonesian`**,**`irish`**,**`kannada`**,**`kyrgyz`**,**`latvian`**,**`ligurian`**, **`luxembourgishmacedonian`**, **`malayalammarathi`**, **`nepalipersian`**,**`sanskrit`**, **`serbiansetswana`**, **`sinhalaslovak`**, **`slovenianswedish`**, **`tagalogtamil`**,**`tatar`**, **`teluguthai`**, **`turkishukrainian`**, **`urduvietnamese`**,, **`yoruba`** Masukkan `multi-language` untuk campuran beberapa bahasa.
- max_top_tokens — (Opsional) Jumlah maksimum token teratas, berdasarkan nilai SHAP global. Default ke `50`. Token dapat muncul beberapa kali dalam kumpulan data. Pekerjaan pemrosesan SageMaker Clarify mengumpulkan nilai SHAP dari setiap token, dan kemudian memilih token teratas berdasarkan nilai SHAP globalnya. Nilai SHAP global dari token teratas yang dipilih disertakan dalam `global_top_shap_text` bagian file analysis.json.
- Nilai agregasi SHAP lokal.

- image_config - Diperlukan untuk penjelasan visi komputer. Sertakan konfigurasi ini jika Anda memiliki kumpulan data input yang terdiri dari gambar dan Anda ingin menganalisisnya untuk dijelaskan dalam masalah penglihatan komputer.
  - model_type — Jenis model. Nilai yang valid meliputi:
    - `IMAGE_CLASSIFICATIO`Nuntuk model klasifikasi citra.
    - `OBJECT_DETECTIO`Nuntuk model deteksi objek.
  - max_objects — Berlaku hanya jika model_type adalah**`OBJECT_DETECTION`**. Jumlah maksimum objek, diurutkan berdasarkan skor kepercayaan, terdeteksi oleh model visi komputer. Setiap objek yang diberi peringkat lebih rendah dari max_objects teratas berdasarkan skor kepercayaan disaring. Default ke `3`.
  - context - Berlaku hanya jika model_type adalah. **`OBJECT_DETECTION`** Ini menunjukkan apakah area di sekitar kotak pembatas objek yang terdeteksi ditutupi oleh gambar dasar

atau tidak. Nilai yang valid adalah `0` untuk menutupi semuanya, atau 1 untuk menutupi apa pun. Defaults ke 1.

- iou_threshold — Berlaku hanya jika `model_type` metrik persimpangan minimum over union (IOU) untuk mengevaluasi prediksi terhadap deteksi asli. **`OBJECT_DETECTION`** Metrik IOU yang tinggi sesuai dengan tumpang tindih besar antara kotak deteksi kebenaran yang diprediksi dan ground. Default ke `0.5`.
- num_segment — (Opsional) Sebuah integer yang menentukan perkiraan jumlah segmen yang akan diberi label dalam gambar input. Setiap segmen gambar dianggap sebagai fitur, dan nilai SHAP lokal dihitung untuk setiap segmen. Default ke `20`.
- segment_compactness — (Opsional) Bilangan bulat yang menentukan bentuk dan ukuran segmen gambar yang dihasilkan oleh metode scikit-image slic. Default ke 5.

- pdp — Sertakan metode ini untuk menghitung plot ketergantungan paral (PDP). Untuk contoh konfigurasi analisis untuk menghasilkan PDP, lihat Hitung plot ketergantungan paral (PDP)
  - fitur - Wajib jika `shap` metode tidak diminta. Array nama fitur atau indeks untuk menghitung dan memplot plot PDP.
  - top_k_features - (Opsional) Menentukan jumlah fitur teratas yang digunakan untuk menghasilkan plot PDP. Jika tidak `features` disediakan, tetapi `shap` metode diminta, maka pekerjaan pemrosesan SageMaker Clarify memilih fitur teratas berdasarkan atribusi SHAP mereka. Default ke `10`.
  - grid_resolution — Jumlah bucket untuk membagi rentang nilai numerik menjadi. Ini menentukan granularitas grid untuk plot PDP.
- report — (Opsional) Gunakan objek ini untuk menyesuaikan laporan analisis. Ada tiga salinan laporan yang sama sebagai bagian dari hasil analisis: laporan Jupyter Notebook, laporan HTML, dan laporan PDF. Objek memiliki parameter berikut:
  - nama — Nama file dari file laporan. Misalnya, jika `name` ya**`MyReport`**, maka file laporan adalah`MyReport.ipynb`,`MyReport.html`, dan`MyReport.pdf`. Default ke `report`.
  - title - (Opsional) String judul untuk laporan. Default ke **`SageMaker Analysis Report`**.

- prediktor — Diperlukan jika analisis membutuhkan prediksi dari model. Misalnya, ketika`shap`, atau `post_training_bias` metode dimintapdp, tetapi label yang diprediksi tidak disediakan sebagai bagian dari dataset input. Berikut ini adalah parameter yang akan digunakan bersama dengan`predictor`:
  - model_name — Nama SageMaker model Anda yang dibuat oleh API. CreateModel Jika Anda menentukan `model_name` alih-alih endpoint_name, pekerjaan pemrosesan SageMaker Clarify membuat titik akhir singkat dengan nama model, yang dikenal sebagai titik akhir bayangan,

dan mendapatkan prediksi dari titik akhir. Pekerjaan menghapus titik akhir bayangan setelah perhitungan selesai. Jika modelnya multi-model, maka `target_model` parameternya harus ditentukan. Untuk informasi selengkapnya tentang titik akhir multimodel, lihat. Host beberapa model dalam satu wadah di belakang satu titik akhir

- endpoint_name_prefix — (Opsional) Sebuah awalan nama kustom untuk titik akhir bayangan. Berlaku jika Anda memberikan `model_name` alih-alih`endpoint_name`. Misalnya, berikan `endpoint_name_prefix` jika Anda ingin membatasi akses ke titik akhir dengan nama titik akhir. Awalan harus sesuai dengan EndpointNamepola, dan panjang maksimumnya adalah`23`. Default ke `sm-clarify`.
- initial_instance_count - Menentukan jumlah contoh untuk titik akhir bayangan. Diperlukan jika Anda memberikan model_name alih-alih endpoint_name. Nilai untuk `initial_instance_count` bisa berbeda dari pekerjaan, tetapi kami merekomendasikan rasio 1:1. InstanceCount
- instance_type - Menentukan jenis contoh untuk titik akhir bayangan. Diperlukan jika Anda memberikan `model_name` alih-alih`endpoint_name`. Sebagai contoh, `instance_type` dapat diatur ke “ml.m5.large”. Dalam beberapa kasus, nilai yang ditentukan untuk `instance_type` dapat membantu mengurangi waktu inferensi model. Misalnya, untuk berjalan secara efisien, model pemrosesan bahasa alami dan model visi komputer biasanya memerlukan jenis instance unit pemrosesan grafis (GPU).
- accelerator_type — (Opsional) Menentukan jenis akselerator Elastic Inference (EI) untuk dilampirkan ke titik akhir bayangan. Berlaku jika Anda menyediakan, `model_name` bukan `endpoint_name` untuk`accelerator_type`. Contoh nilai untuk `accelerator_type` adalah**`ml.eia2.large`**. Default untuk tidak menggunakan akselerator.
- endpoint_name — Nama SageMaker endpoint Anda yang dibuat oleh API. CreateEndpoint Jika disediakan, lebih `endpoint_name` diutamakan daripada parameter. `model_name` Menggunakan titik akhir yang ada mengurangi waktu bootstrap titik akhir bayangan, tetapi juga dapat menyebabkan peningkatan beban yang signifikan untuk titik akhir tersebut. Selain itu, beberapa metode analisis (seperti `shap` danpdp) menghasilkan dataset sintetis yang dikirim ke titik akhir. Hal ini dapat menyebabkan metrik titik akhir atau data yang diambil terkontaminasi oleh data sintetis, yang mungkin tidak secara akurat mencerminkan penggunaan dunia nyata. Untuk alasan ini, umumnya tidak disarankan untuk menggunakan titik akhir produksi yang ada untuk analisis SageMaker Clarify.
- target_model — Nilai string yang diteruskan ke TargetModel parameter API. SageMaker InvokeEndpoint Diperlukan jika model Anda (ditentukan oleh parameter model_name) atau titik akhir (ditentukan oleh parameter endpoint_name) adalah multi-model. Untuk informasi

selengkapnya tentang titik akhir multimodel, lihat. Host beberapa model dalam satu wadah di belakang satu titik akhir

- custom_attributes — (Opsional) String yang memungkinkan Anda memberikan informasi tambahan tentang permintaan inferensi yang dikirimkan ke titik akhir. Nilai string diteruskan ke `CustomAttributes` parameter SageMaker InvokeEndpointAPI.
- content_type — content_type — Format input model yang akan digunakan untuk mendapatkan prediksi dari titik akhir. Jika disediakan, itu diteruskan ke `ContentType` parameter SageMaker InvokeEndpointAPI.
  - Untuk penjelasan visi komputer, nilai yang valid adalah**`image/jpeg`**, **`image/png`** atau. **`application/x-npy`** Jika tidak `content_type` disediakan, nilai defaultnya adalah**`image/jpeg`**.
  - Untuk jenis penjelasan lainnya, nilai yang valid adalah**`text/csv`**, `application/jsonlines,` dan. **`application/json`** Nilai untuk `content_type` diperlukan jika “aplikasi/x-parket”. `dataset_type` Jika tidak `content_type` default ke nilai parameter. `dataset_type`
- accept_type — Format keluaran model yang akan digunakan untuk mendapatkan prediksi dari titik akhir. Nilai untuk `accept_type` diteruskan ke `Accept` parameter SageMaker InvokeEndpointAPI.
  - Untuk penjelasan visi komputer, jika `model_type` adalah “OBJECT_DETECTION” maka defaultnya. `accept_type` **`application/json`**
  - Untuk jenis penjelasan lainnya, nilai yang valid adalah**`text/csv`**,**`application/jsonlines`**, dan. **`application/json`** Jika nilai untuk tidak `accept_type` disediakan, `accept_type` default ke nilai parameter. `content_type`
- content_template — String template yang digunakan untuk membangun input model dari catatan dataset. Parameter hanya `content_template` digunakan dan diperlukan jika nilai `content_type` parameternya salah satu `application/jsonlines` atau`application/json`.

  Ketika `content_type` parameternya`application/jsonlines`, template seharusnya hanya memiliki satu placeholder`$features`, yang digantikan oleh daftar fitur saat runtime. Misalnya, jika template adalah`"{\"myfeatures\":$features}"`, dan jika catatan memiliki tiga nilai fitur numerik:`1`, `2` dan`3`, maka catatan akan dikirim ke model sebagai JSON Line. `{"myfeatures":[1,2,3]}`

Ketika `content_type` ada`application/json`, template dapat memiliki placeholder `$record` atau. `records` Jika placeholder adalah`record`, satu record diganti dengan record yang memiliki template yang `record_template` diterapkan padanya. Dalam hal ini, hanya satu catatan yang akan dikirim ke model sekaligus. Jika placeholder adalah`$records`, catatan diganti dengan daftar catatan, masing-masing dengan template yang disediakan oleh. `record_template`

- record_template — String template yang akan digunakan untuk membangun setiap catatan input model dari instance dataset. Ini hanya digunakan dan diperlukan kapan `content_type` saja`application/json`. String templat dapat berisi salah satu dari berikut ini:
  - `$features`Parameter placeholder yang digantikan oleh array nilai fitur. Placeholder opsional tambahan dapat menggantikan nama header kolom fitur di. `$feature_names` Placeholder opsional ini akan diganti dengan array nama fitur.
  - Tepat satu placeholder `$features_kvp` yang digantikan oleh pasangan kunci-nilai, nama fitur dan nilai fitur.
  - Fitur dalam `headers` konfigurasi. Sebagai contoh, nama `A` fitur, yang dinotasikan oleh sintaks placeholder `"${A}"` akan diganti dengan nilai fitur untuk. `A`

  Nilai untuk `record_template` digunakan dengan `content_template` untuk membangun input model. Contoh konfigurasi yang menunjukkan cara membuat input model menggunakan konten dan merekam template berikut.

  Dalam contoh kode berikut, header dan fitur didefinisikan sebagai berikut.

  - `` `headers`:["A", "B"] ``
  - `` `features`:[[0,1], [3,4]] ``

  Contoh input model adalah sebagai berikut.

```
{
    "instances": [[0, 1], [3, 4]],
    "feature_names": ["A", "B"]
}
```

  Contoh `content_template` dan nilai `record_template` parameter untuk membangun contoh masukan model sebelumnya berikut.

  - `content_template: "{\"instances\": $records, \"feature_names\": $feature_names}"`
  - `record_template: "$features"`

Dalam contoh kode berikut, header dan fitur didefinisikan sebagai berikut.

```
[
    { "A": 0, "B": 1 },
    { "A": 3, "B": 4 },
]
```

Contoh `content_template` dan nilai `record_template` parameter untuk membangun contoh masukan model sebelumnya berikut.

- `content_template: "$records"`
- `record_template: "$features_kvp"`

Contoh kode alternatif untuk membangun contoh masukan model sebelumnya berikut.

- `content_template: "$records"`
- `record_template: "{\"A\": \"${A}\", \"B\": \"${B}\"}"`

Dalam contoh kode berikut, header dan fitur didefinisikan sebagai berikut.

```
{ "A": 0, "B": 1 }
```

Contoh parameter content_template dan record_template nilai untuk membangun di atas: contoh masukan model sebelumnya berikut.

- `content_template: "$record"`
- `record_template: "$features_kvp"`

- label — Indeks bilangan bulat berbasis nol atau string ekspresi JMESPath yang digunakan untuk mengekstrak label yang diprediksi dari keluaran model untuk analisis bias. Jika modelnya multiclass dan `label` parameter mengekstrak semua label yang diprediksi dari output model, maka berikut ini berlaku.
  - `probability`Parameter diperlukan untuk mendapatkan probabilitas (atau skor) yang sesuai dari output model.
  - Label prediksi skor tertinggi dipilih.

  Nilai untuk `label` tergantung pada nilai parameter accept_type sebagai berikut.

  - Jika `accept_type` ya**`text/csv`**, maka `label` adalah indeks dari setiap label yang diprediksi dalam output model.

  - Jika `accept_type` adalah **`application/jsonlines`** atau**`application/json`**, maka `label` adalah ekspresi JMESPath yang diterapkan ke output model untuk mendapatkan label yang diprediksi.
- label_headers — Sebuah array nilai yang label dapat mengambil dalam dataset. Jika analisis bias diminta, maka `probability` parameter juga diperlukan untuk mendapatkan nilai probabilitas (skor) yang sesuai dari output model, dan label prediksi dari skor tertinggi dipilih. Jika analisis penjelasan diminta, header label digunakan untuk mempercantik laporan analisis. Nilai untuk `label_headers` diperlukan untuk penjelasan visi komputer. Misalnya, untuk masalah klasifikasi multiclass, jika label memiliki tiga nilai yang mungkin,,, dan **`bird`** **`catdog`**, maka `label_headers` harus disetel ke. `["bird","cat","dog"]`
- probabilitas — (Opsional) Indeks bilangan bulat berbasis nol atau string ekspresi JMESPath yang digunakan untuk mengekstrak probabilitas (skor) untuk analisis penjelasan, atau untuk memilih label yang diprediksi untuk analisis bias. Nilai `probability` tergantung pada nilai `accept_type` parameter sebagai berikut.
  - Jika ya**`text/csv`**, `accept_type` `probability` adalah indeks probabilitas (skor) dalam output model. Jika tidak `probability` disediakan, seluruh output model diambil sebagai probabilitas (skor).
  - Jika `accept_type` adalah data JSON (salah satu **`application/jsonlines`** atau**`application/json`**), `probability` harus berupa ekspresi JMESPath yang digunakan untuk mengekstrak probabilitas (skor) dari output model.

Contoh file konfigurasi analisis

Bagian berikut berisi contoh file konfigurasi analisis untuk data dalam format CSV, format JSON Lines, dan untuk pemrosesan bahasa alami (NLP) dan penjelasan visi komputer.

Konfigurasi analisis untuk kumpulan data CSV

Contoh-contoh berikut menunjukkan cara mengonfigurasi bias dan analisis penjelasan untuk kumpulan data tabel dalam format CSV. Dalam contoh ini, dataset yang masuk memiliki empat kolom fitur, dan satu kolom label biner,. `Target` Isi dataset adalah sebagai berikut. Nilai label 1 menunjukkan hasil positif. Dataset disediakan untuk pekerjaan SageMaker Clarify dengan input `dataset` pemrosesan.

```
"Target","Age","Gender","Income","Occupation"
0,25,0,2850,2
1,36,0,6585,0
```

```
1,22,1,1759,1
0,48,0,3446,1
...
```

Bagian berikut menunjukkan cara menghitung metrik bias pra-pelatihan dan pasca-pelatihan, nilai SHAP, dan plot ketergantungan paral (PDP) yang menunjukkan pentingnya fitur untuk kumpulan data dalam format CSV.

Hitung semua metrik bias pra-pelatihan

Contoh konfigurasi ini menunjukkan cara mengukur apakah kumpulan data sampel sebelumnya bias terhadap sampel dengan **`Gender`** nilai. `0` Konfigurasi analisis berikut menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghitung semua metrik bias pra-pelatihan untuk kumpulan data.

```
{
    "dataset_type": "text/csv",
    "label": "Target",
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
        }
    }
}
```

Hitung semua metrik bias pasca-pelatihan

Anda dapat menghitung metrik bias pra-pelatihan sebelum pelatihan. Namun, Anda harus memiliki model terlatih untuk menghitung metrik bias pasca-pelatihan. Contoh output berikut adalah dari model klasifikasi biner yang mengeluarkan data dalam format CSV. Dalam contoh output ini, setiap baris berisi dua kolom. Kolom pertama berisi label yang diprediksi, dan kolom kedua berisi nilai probabilitas untuk label tersebut.

```
0,0.028986845165491
1,0.825382471084594
```

```
...
```

Contoh konfigurasi berikut menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghitung semua metrik bias yang mungkin menggunakan kumpulan data dan prediksi dari keluaran model. Dalam contoh, model diterapkan ke titik SageMaker akhir`your_endpoint`.

> **Note**
>
> Dalam contoh kode berikut, parameter `content_type` dan tidak `accept_type` diatur. Oleh karena itu, mereka secara otomatis menggunakan nilai parameter dataset_type, yaitu. `text/csv`

```
{
    "dataset_type": "text/csv",
    "label": "Target",
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
        },
        "post_training_bias": {
            "methods": "all"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "label": 0
    }
}
```

Hitung nilai SHAP

Contoh konfigurasi analisis berikut menginstruksikan pekerjaan untuk menghitung nilai SHAP yang menunjuk `Target` kolom sebagai label dan semua kolom lainnya sebagai fitur.

```
{
    "dataset_type": "text/csv",
    "label": "Target",
    "methods": {
        "shap": {
            "num_clusters": 1
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "probability": 1
    }
}
```

Dalam contoh ini, `baseline` parameter SHAP dihilangkan dan nilai parameternya adalah. `num_clusters` 1 Ini menginstruksikan prosesor SageMaker Clarify untuk menghitung satu sampel dasar SHAP. Dalam contoh ini, probabilitas diatur ke1. Ini menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk mengekstrak skor probabilitas dari kolom kedua dari output model (menggunakan pengindeksan berbasis nol).

Hitung plot ketergantungan paral (PDP)

Contoh berikut menunjukkan bagaimana melihat pentingnya `Income` fitur pada laporan analisis menggunakan PDP. Parameter laporan menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghasilkan laporan. Setelah pekerjaan selesai, laporan yang dihasilkan disimpan sebagai report.pdf ke `analysis_result` lokasi. `grid_resolution`Parameter membagi rentang nilai fitur ke dalam `10` ember. Bersama-sama, parameter yang ditentukan dalam contoh berikut menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghasilkan laporan yang berisi grafik PDP `Income` dengan `10` segmen pada sumbu x. Sumbu y akan menunjukkan dampak marjinal `Income` pada prediksi.

```
{
    "dataset_type": "text/csv",
    "label": "Target",
    "methods": {
        "pdp": {
            "features": ["Income"],
            "grid_resolution": 10
        },
        "report": {
            "name": "report"
```

```
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "probability": 1
    },
}
```

Hitung metrik bias dan kepentingan fitur

Anda dapat menggabungkan semua metode dari contoh konfigurasi sebelumnya ke dalam satu file konfigurasi analisis dan menghitung semuanya dengan satu pekerjaan. Contoh berikut menunjukkan konfigurasi analisis dengan semua langkah digabungkan.

Dalam contoh ini, `probability` parameter diatur `1` untuk menunjukkan bahwa probabilitas terkandung dalam kolom kedua (menggunakan pengindeksan berbasis nol). Namun, karena analisis bias membutuhkan label yang diprediksi, `probability_threshold` parameter diatur `0.5` untuk mengubah skor probabilitas menjadi label biner. Dalam contoh ini, `top_k_features` parameter `pdp` metode plot ketergantungan paral diatur ke2. Ini menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghitung plot ketergantungan paral (PDP) untuk 2 fitur teratas dengan nilai SHAP global terbesar.

```
{
    "dataset_type": "text/csv",
    "label": "Target",
    "probability_threshold": 0.5,
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
        },
        "post_training_bias": {
            "methods": "all"
        },
        "shap": {
            "num_clusters": 1
```

```
        },
        "pdp": {
            "top_k_features": 2,
            "grid_resolution": 10
        },
        "report": {
            "name": "report"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "probability": 1
    }
}
```

Alih-alih menerapkan model ke titik akhir, Anda dapat memberikan nama SageMaker model Anda ke tugas pemrosesan SageMaker Clarify menggunakan parameter. `model_name` Contoh berikut menunjukkan cara menentukan model bernama**`your_model`**. Pekerjaan pemrosesan SageMaker Clarify akan membuat titik akhir bayangan menggunakan konfigurasi.

```
{
     ...
    "predictor": {
        "model_name": "your_model",
        "initial_instance_count": 1,
        "instance_type": "ml.m5.large",
        "probability": 1
    }
}
```

Konfigurasi analisis untuk dataset JSON Lines

Contoh berikut menunjukkan cara mengkonfigurasi analisis bias dan analisis eksplainabilitas untuk dataset tabular dalam format JSON Lines. Dalam contoh ini, dataset yang masuk memiliki data yang sama dengan bagian sebelumnya tetapi mereka berada dalam format padat SageMaker JSON Lines. Setiap baris adalah objek JSON valid. Kunci “Fitur” menunjuk ke array nilai fitur, dan kunci “Label” menunjuk ke label kebenaran dasar. Dataset disediakan untuk pekerjaan SageMaker Clarify dengan input pemrosesan “dataset”. Untuk informasi selengkapnya tentang JSON Lines, lihatFormat Permintaan JSONLINES.

```
{"Features":[25,0,2850,2],"Label":0}
```

```
{"Features":[36,0,6585,0],"Label":1}
{"Features":[22,1,1759,1],"Label":1}
{"Features":[48,0,3446,1],"Label":0}
...
```

Bagian berikut menunjukkan cara menghitung metrik bias pra-pelatihan dan pasca-pelatihan, nilai SHAP, dan plot ketergantungan paral (PDP) yang menunjukkan pentingnya fitur untuk kumpulan data dalam format JSON Lines.

Hitung metrik bias pra-pelatihan

Tentukan label, fitur, format, dan metode untuk mengukur metrik bias pra-pelatihan untuk `Gender` nilai. `0` Dalam contoh berikut, `headers` parameter memberikan nama fitur terlebih dahulu. Nama label diberikan terakhir. Menurut konvensi, header terakhir adalah header label.

`features`Parameter diatur ke ekspresi JMESPath “Features” sehingga pekerjaan pemrosesan SageMaker Clarify dapat mengekstrak array fitur dari setiap record. `label`Parameter diatur ke ekspresi JMESPath “Label” sehingga pekerjaan pemrosesan SageMaker Clarify dapat mengekstrak label kebenaran dasar dari setiap rekaman. Gunakan nama facet untuk menentukan atribut sensitif, sebagai berikut.

```
{
    "dataset_type": "application/jsonlines",
    "headers": ["Age","Gender","Income","Occupation","Target"],
    "label": "Label",
    "features": "Features",
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
        }
    }
}
```

Hitung semua metrik bias

Anda harus memiliki model terlatih untuk menghitung metrik bias pasca-pelatihan. Contoh berikut adalah dari model klasifikasi biner yang mengeluarkan data JSON Lines dalam format contoh. Setiap baris output model adalah objek JSON valid. `predicted_label`Poin kunci untuk label yang diprediksi, dan `probability` poin-poin kunci untuk nilai probabilitas.

```
{"predicted_label":0,"probability":0.028986845165491}
{"predicted_label":1,"probability":0.825382471084594}
...
```

Anda dapat menerapkan model ke SageMaker titik akhir bernama. `your_endpoint` Contoh konfigurasi analisis berikut menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghitung semua metrik bias yang mungkin untuk kumpulan data dan model. Dalam contoh ini, parameter `content_type` dan tidak `accept_type` diatur. Oleh karena itu, mereka secara otomatis diatur untuk menggunakan nilai parameter dataset_type, yaitu. `application/jsonlines` Pekerjaan pemrosesan SageMaker Clarify menggunakan `content_template` parameter untuk menyusun input model, dengan mengganti `$features` placeholder dengan array fitur.

```
{
    "dataset_type": "application/jsonlines",
    "headers": ["Age","Gender","Income","Occupation","Target"],
    "label": "Label",
    "features": "Features",
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
        },
        "post_training_bias": {
            "methods": "all"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
```

```
        "content_template": "{\"Features\":$features}",
        "label": "predicted_label"
    }
}
```

Hitung nilai SHAP

Karena analisis SHAP tidak memerlukan label kebenaran dasar, `label` parameternya dihilangkan. Dalam contoh ini, `headers` parameter juga dihilangkan. Oleh karena itu, pekerjaan pemrosesan SageMaker Clarify harus menghasilkan placeholder menggunakan nama generik seperti `column_0` atau `column_1` untuk header fitur, dan `label0` untuk header label. Anda dapat menentukan nilai untuk `headers` dan `label` untuk a untuk meningkatkan keterbacaan hasil analisis. Karena parameter probabilitas diatur ke ekspresi JMESPath`probability`, nilai probabilitas akan diekstraksi dari output model. Berikut ini adalah contoh untuk menghitung nilai SHAP.

```
{
    "dataset_type": "application/jsonlines",
    "features": "Features",
    "methods": {
        "shap": {
            "num_clusters": 1
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "content_template": "{\"Features\":$features}",
        "probability": "probability"
    }
}
```

Hitung plot ketergantungan paral (PDP)

Contoh berikut menunjukkan cara melihat pentingnya “Pendapatan” pada PDP. Dalam contoh ini, header fitur tidak disediakan. Oleh karena itu, `features` parameter pdp metode harus menggunakan indeks berbasis nol untuk merujuk ke lokasi kolom fitur. `grid_resolution`Parameter membagi rentang nilai fitur ke dalam `10` ember. Bersama-sama, parameter dalam contoh menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghasilkan laporan yang berisi grafik PDP `Income` dengan `10` segmen pada sumbu x. Sumbu y akan menunjukkan dampak marjinal `Income` pada prediksi.

```
{
```

```
    "dataset_type": "application/jsonlines",
    "features": "Features",
    "methods": {
        "pdp": {
            "features": [2],
            "grid_resolution": 10
        },
        "report": {
            "name": "report"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "content_template": "{\"Features\":$features}",
        "probability": "probability"
    }
}
```

Hitung metrik bias dan kepentingan fitur

Anda dapat menggabungkan semua metode sebelumnya ke dalam satu file konfigurasi analisis dan menghitung semuanya dengan satu pekerjaan. Contoh berikut menunjukkan konfigurasi analisis dengan semua langkah digabungkan. Dalam contoh ini, `probability` parameter diatur. Tetapi karena analisis bias membutuhkan label yang diprediksi, `probability_threshold` parameter diatur `0.5` untuk mengubah skor probabilitas menjadi label biner. Dalam contoh ini, `top_k_features` parameter pdp metode diatur ke2. Ini menginstruksikan tugas pemrosesan SageMaker Clarify untuk menghitung PDP untuk 2 fitur teratas dengan nilai SHAP global terbesar.

```
{
    "dataset_type": "application/jsonlines",
    "headers": ["Age","Gender","Income","Occupation","Target"],
    "label": "Label",
    "features": "Features",
    "probability_threshold": 0.5,
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
```

```
        "pre_training_bias": {
            "methods": "all"
        },
        "post_training_bias": {
            "methods": "all"
        },
        "shap": {
            "num_clusters": 1
        },
        "pdp": {
            "top_k_features": 2,
            "grid_resolution": 10
        },
        "report": {
            "name": "report"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "content_template": "{\"Features\":$features}",
        "probability": "probability"
    }
}
```

Konfigurasi analisis untuk dataset JSON

Contoh-contoh berikut menunjukkan cara mengonfigurasi bias dan analisis eksplainabilitas untuk dataset tabel dalam format JSON. Dalam contoh ini, dataset yang masuk memiliki data yang sama dengan bagian sebelumnya tetapi mereka dalam format padat SageMaker JSON. Untuk informasi selengkapnya tentang JSON Lines, lihatFormat Permintaan JSONLINES.

Seluruh permintaan input adalah JSON yang valid di mana struktur luar adalah daftar dan setiap elemen adalah data untuk catatan. Dalam setiap catatan, `Features` poin kunci ke array nilai fitur, dan `Label` poin-poin kunci ke label kebenaran dasar. Dataset disediakan untuk pekerjaan SageMaker Clarify dengan input `dataset` pemrosesan.

```
[
    {"Features":[25,0,2850,2],"Label":0},
    {"Features":[36,0,6585,0],"Label":1},
    {"Features":[22,1,1759,1],"Label":1},
    {"Features":[48,0,3446,1],"Label":0},
    ...
```

```
]
```

Bagian berikut menunjukkan cara menghitung metrik bias pra-pelatihan dan pasca-pelatihan, nilai SHAP, dan plot ketergantungan paral (PDP) yang menunjukkan pentingnya fitur untuk kumpulan data dalam format JSON Lines.

Hitung metrik bias pra-pelatihan

Tentukan label, fitur, format, dan metode untuk mengukur metrik bias pra-pelatihan untuk `Gender` nilai. `0` Dalam contoh berikut, `headers` parameter memberikan nama fitur terlebih dahulu. Nama label diberikan terakhir. Untuk dataset JSON, header terakhir adalah header label.

`features`Parameter diatur ke ekspresi JMESPath yang mengekstrak array 2D atau matriks. Setiap baris dalam matriks ini harus berisi daftar `Features` untuk setiap catatan. `label`Parameter diatur ke ekspresi JMESPath yang mengekstrak daftar label kebenaran dasar. Setiap elemen dalam daftar ini harus berisi label untuk catatan.

Gunakan nama facet untuk menentukan atribut sensitif, sebagai berikut.

```
{
    "dataset_type": "application/json",
    "headers": ["Age","Gender","Income","Occupation","Target"],
    "label": "[*].Label",
    "features": "[*].Features",
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
        }
    }
}
```

Hitung semua metrik bias

Anda harus memiliki model terlatih untuk menghitung metrik bias pasca-pelatihan. Contoh kode berikut adalah dari model klasifikasi biner yang mengeluarkan data JSON dalam format contoh.

Dalam contoh, setiap elemen di bawah `predictions` adalah output prediksi untuk catatan. Kode contoh berisi kunci`predicted_label`, yang menunjuk ke label yang diprediksi, dan `probability` poin-poin kunci ke nilai probabilitas.

```
{
    "predictions": [
        {"predicted_label":0,"probability":0.028986845165491},
        {"predicted_label":1,"probability":0.825382471084594},
        ...
    ]
}
```

Anda dapat menerapkan model ke SageMaker titik akhir bernama. `your_endpoint`

Dalam contoh berikut, parameter `content_type` dan tidak `accept_type` diatur. Oleh karena itu, `content_type` dan `accept_type` secara otomatis diatur untuk menggunakan nilai parameter`dataset_type`, yaitu`application/json`. Pekerjaan pemrosesan SageMaker Clarify kemudian menggunakan `content_template` parameter untuk menyusun input model.

Dalam contoh berikut, input model disusun dengan mengganti `$records` placeholder dengan array catatan. Kemudian, `record_template` parameter menyusun struktur JSON setiap record dan menggantikan `$features` placeholder dengan array fitur masing-masing record.

Contoh konfigurasi analisis berikut menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghitung semua metrik bias yang mungkin untuk kumpulan data dan model.

```
{
    "dataset_type": "application/json",
    "headers": ["Age","Gender","Income","Occupation","Target"],
    "label": "[*].Label",
    "features": "[*].Features",
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
```

```
        },
        "post_training_bias": {
            "methods": "all"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "content_template": "$records",
        "record_template": "{\"Features\":$features}",
        "label": "predictions[*].predicted_label"
    }
}
```

Hitung nilai SHAP

Anda tidak perlu menentukan label untuk analisis SHAP. Dalam contoh berikut, `headers` parameter tidak ditentukan. Oleh karena itu, pekerjaan pemrosesan SageMaker Clarify akan menghasilkan placeholder menggunakan nama generik seperti `column_0` atau `column_1` untuk header fitur, dan `label0` untuk header label. Anda dapat menentukan nilai untuk `headers` dan `label` untuk a untuk meningkatkan keterbacaan hasil analisis.

Dalam contoh konfigurasi berikut, parameter probabilitas diatur ke ekspresi JMESPath yang mengekstrak probabilitas dari setiap prediksi untuk setiap catatan. Berikut ini adalah contoh untuk menghitung nilai SHAP.

```
{
    "dataset_type": "application/json",
    "features": "[*].Features",
    "methods": {
        "shap": {
            "num_clusters": 1
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "content_template": "$records",
        "record_template": "{\"Features\":$features}",
        "probability": "predictions[*].probability"
    }
}
```

Hitung plot ketergantungan paral (PDP)

Contoh berikut menunjukkan cara melihat fitur penting dalam PDP. Dalam contoh, header fitur tidak disediakan. Oleh karena itu, `features` parameter pdp metode harus menggunakan indeks berbasis nol untuk merujuk ke lokasi kolom fitur. `grid_resolution`Parameter membagi rentang nilai fitur ke dalam `10` ember.

Bersama-sama, parameter dalam contoh berikut menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menghasilkan laporan yang berisi grafik PDP `Income` dengan `10` segmen pada sumbu x. Sumbu y menunjukkan dampak marjinal `Income` pada prediksi.

Contoh konfigurasi berikut ini menunjukkan cara melihat pentingnya `Income` pada PDP.

```
{
    "dataset_type": "application/json",
    "features": "[*].Features",
    "methods": {
        "pdp": {
            "features": [2],
            "grid_resolution": 10
        },
        "report": {
            "name": "report"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "content_template": "$records",
        "record_template": "{\"Features\":$features}",
        "probability": "predictions[*].probability"
    }
}
```

Hitung metrik bias dan kepentingan fitur

Anda dapat menggabungkan semua metode konfigurasi sebelumnya ke dalam satu file konfigurasi analisis dan menghitung semuanya dengan satu pekerjaan. Contoh berikut menunjukkan konfigurasi analisis dengan semua langkah digabungkan.

Dalam contoh ini, `probability` parameter diatur. Karena analisis bias membutuhkan label yang diprediksi, `probability_threshold` parameter diatur ke`0.5`, yang digunakan untuk mengubah

skor probabilitas menjadi label biner. Dalam contoh ini, `top_k_features` parameter pdp metode diatur ke2. Ini menginstruksikan tugas pemrosesan SageMaker Clarify untuk menghitung PDP untuk 2 fitur teratas dengan nilai SHAP global terbesar.

```
{
    "dataset_type": "application/json",
    "headers": ["Age","Gender","Income","Occupation","Target"],
    "label": "[*].Label",
    "features": "[*].Features",
    "probability_threshold": 0.5,
    "label_values_or_threshold": [1],
    "facet": [
        {
            "name_or_index": "Gender",
            "value_or_threshold": [0]
        }
    ],
    "methods": {
        "pre_training_bias": {
            "methods": "all"
        },
        "post_training_bias": {
            "methods": "all"
        },
        "shap": {
            "num_clusters": 1
        },
        "pdp": {
            "top_k_features": 2,
            "grid_resolution": 10
        },
        "report": {
            "name": "report"
        }
    },
    "predictor": {
        "endpoint_name": "your_endpoint",
        "content_template": "$records",
        "record_template": "{\"Features\":$features}",
        "probability": "predictions[*].probability"
    }
}
```

Konfigurasi analisis untuk menjelaskan pemrosesan bahasa alami

Contoh berikut menunjukkan file konfigurasi analisis untuk fitur komputasi yang penting untuk pemrosesan bahasa alami (NLP). Dalam contoh ini, kumpulan data yang masuk adalah kumpulan data tabel dalam format CSV, dengan satu kolom label biner dan dua kolom fitur, sebagai berikut. Dataset disediakan untuk pekerjaan SageMaker Clarify dengan parameter input `dataset` pemrosesan.

```
0,2,"They taste gross"
1,3,"Flavor needs work"
1,5,"Taste is awful"
0,1,"The worst"
...
```

Dalam contoh ini, model klasifikasi biner dilatih pada dataset sebelumnya. Model menerima data CSV, dan menghasilkan skor tunggal antara `0` dan`1`, sebagai berikut.

```
0.491656005382537
0.569582343101501
...
```

Model ini digunakan untuk membuat SageMaker model bernama “your_model”. Konfigurasi analisis berikut menunjukkan cara menjalankan analisis penjelasan berdasarkan token menggunakan model dan dataset. `text_config`Parameter mengaktifkan analisis penjelasan NLP. `granularity`Parameter menunjukkan bahwa analisis harus mengurai token.

Dalam bahasa Inggris, setiap token adalah sebuah kata. Contoh berikut juga menunjukkan cara menyediakan instance “baseline” SHAP di tempat menggunakan rata-rata “Rating” 4. Token topeng khusus “[MASK]” digunakan untuk mengganti token (kata) di “Komentar”. Contoh ini juga menggunakan tipe instance titik akhir GPU untuk mempercepat inferensi.

```
{
    "dataset_type": "text/csv",
    "headers": ["Target","Rating","Comments"]
    "label": "Target",
    "methods": {
        "shap": {
            "text_config": {
                "granularity": "token",
                "language": "english"
            }
```

```
            "baseline": [[4,"[MASK]"]],
        }
    },
    "predictor": {
        "model_name": "your_nlp_model",
        "initial_instance_count": 1,
        "instance_type": "ml.g4dn.xlarge"
    }
}
```

Konfigurasi analisis untuk penjelasan visi komputer

Contoh berikut menunjukkan analisis konfigurasi fitur komputasi file yang penting untuk visi komputer. Dalam contoh ini, set data input terdiri dari gambar JPEG. Dataset disediakan untuk pekerjaan SageMaker Clarify dengan parameter input `dataset` pemrosesan. Contoh menunjukkan cara mengkonfigurasi analisis penjelasan menggunakan model klasifikasi SageMaker gambar. Dalam contoh, model bernama`your_cv_ic_model`, telah dilatih untuk mengklasifikasikan hewan pada gambar JPEG input.

```
{
    "dataset_type": "application/x-image",
    "methods": {
        "shap": {
             "image_config": {
                "model_type": "IMAGE_CLASSIFICATION",
                 "num_segments": 20,
                "segment_compactness": 10
             }
        },
        "report": {
            "name": "report"
        }
    },
    "predictor": {
        "model_name": "your_cv_ic_model",
        "initial_instance_count": 1,
        "instance_type": "ml.p2.xlarge",
        "label_headers": ["bird","cat","dog"]
    }
}
```

Untuk informasi selengkapnya tentang klasifikasi gambar, lihatKlasifikasi Gambar - MXNet.

Dalam contoh ini, model deteksi SageMaker objek, `your_cv_od_model` dilatih pada gambar JPEG yang sama untuk mengidentifikasi hewan pada mereka. Contoh berikut menunjukkan cara mengonfigurasi analisis penjelasan untuk model deteksi objek.

```
{
    "dataset_type": "application/x-image",
    "probability_threshold": 0.5,
    "methods": {
        "shap": {
             "image_config": {
                "model_type": "OBJECT_DETECTION",
                 "max_objects": 3,
                "context": 1.0,
                "iou_threshold": 0.5,
                 "num_segments": 20,
                "segment_compactness": 10
             }
        },
        "report": {
            "name": "report"
        }
    },
    "predictor": {
        "model_name": "your_cv_od_model",
        "initial_instance_count": 1,
        "instance_type": "ml.p2.xlarge",
        "label_headers": ["bird","cat","dog"]
    }
}
```

Panduan Kompatibilitas Format Data

Panduan ini menjelaskan tipe format data yang kompatibel dengan pekerjaan pemrosesan SageMaker Clarify. Jenis format data yang didukung mencakup ekstensi file, struktur data, dan persyaratan atau batasan khusus untuk kumpulan data tabel dan gambar. Panduan ini juga menunjukkan cara memeriksa apakah kumpulan data Anda sesuai dengan persyaratan ini.

Pada tingkat tinggi, pekerjaan pemrosesan SageMaker Clarify mengikuti model input-proses-output untuk menghitung metrik bias dan atribusi fitur. Lihat contoh berikut untuk detailnya.

Masukan untuk pekerjaan pemrosesan SageMaker Clarify terdiri dari yang berikut:

- Dataset yang akan dianalisis.
- Konfigurasi analisis. Untuk informasi selengkapnya tentang cara mengonfigurasi analisis, lihatKonfigurasikan Analisis.

Selama tahap pemrosesan, SageMaker Clarify menghitung metrik bias dan atribusi fitur. Pekerjaan pemrosesan Clarify menyelesaikan langkah-langkah berikut di backend: SageMaker

- Pekerjaan pemrosesan SageMaker Clarify mem-parsing konfigurasi analisis Anda dan memuat kumpulan data Anda.
- Untuk menghitung metrik bias pasca-pelatihan dan atribusi fitur, pekerjaan memerlukan prediksi model dari model Anda. Pekerjaan pemrosesan SageMaker Clarify membuat serialisasi data Anda dan mengirimkannya sebagai permintaan ke model Anda yang diterapkan pada titik akhir inferensi SageMaker real-time. Setelah itu, pekerjaan pemrosesan SageMaker Clarify mengekstrak prediksi dari respons.
- Pekerjaan pemrosesan SageMaker Clarify melakukan analisis bias dan penjelasan, dan kemudian menghasilkan hasilnya.

Untuk informasi selengkapnya, lihat Bagaimana SageMaker Memperjelas Pekerjaan Pemrosesan Pekerjaan.

Parameter yang Anda gunakan untuk menentukan format data bergantung pada di mana data digunakan dalam aliran pemrosesan sebagai berikut:

- Untuk dataset input, gunakan `dataset_type` parameter untuk menentukan format atau tipe MIME.
- Untuk permintaan ke titik akhir, gunakan `content_type` parameter untuk menentukan format.
- Untuk respons dari titik akhir, gunakan `accept_type` parameter untuk menentukan format.

Dataset input, permintaan, dan respons ke dan dari titik akhir tidak memerlukan format yang sama. Misalnya, Anda dapat menggunakan kumpulan data Parket dengan muatan permintaan CSV dan muatan respons JSON Lines dengan ketentuan berikut.

- Analisis Anda dikonfigurasi dengan benar.
- Model Anda mendukung format permintaan dan respons.

> **Note**
>
> Jika `accept_type` disediakan `content_type` atau tidak, maka kontainer SageMaker Clarify menyimpulkan `content_type` dan`accept_type`.

Topik

- Data tabular
- Data gambar

Data tabular

Data tabular mengacu pada data yang dapat dimuat ke dalam bingkai data dua dimensi. Dalam bingkai, setiap baris mewakili catatan, dan setiap catatan memiliki satu atau lebih kolom. Nilai dalam setiap sel bingkai data dapat berupa tipe data numerik, kategoris, atau teks.

Prasyarat set data tabel

Sebelum analisis, dataset Anda seharusnya memiliki langkah-langkah pra-pemrosesan yang diperlukan yang sudah diterapkan. Ini termasuk pembersihan data atau rekayasa fitur.

Anda dapat menyediakan satu atau beberapa kumpulan data. Jika Anda menyediakan beberapa kumpulan data, gunakan yang berikut ini untuk mengidentifikasinya ke pekerjaan pemrosesan SageMaker Clarify.

- Gunakan konfigurasi ProcessingInputbernama `dataset` atau analisis `dataset_uri` untuk menentukan kumpulan data utama. Untuk informasi selengkapnya`dataset_uri`, lihat daftar parameter diKonfigurasikan Analisis.
- Gunakan `baseline` parameter yang disediakan dalam file konfigurasi analisis. Dataset dasar diperlukan untuk analisis SHAP. Untuk informasi selengkapnya tentang file konfigurasi analisis, termasuk contoh, lihatKonfigurasikan Analisis.

Tabel berikut mencantumkan format data yang didukung, ekstensi file, dan tipe MIME.

Format data	Ekstensi file	Tipe MIME
CSV	csv	`text/csv`

Format data	Ekstensi file	Tipe MIME
Garis JSON	jsonl	`application/jsonlines`
JSON	json	`application/json`
Parquet	parquet	“aplikasi/x-parket”

Bagian berikut menunjukkan contoh kumpulan data tabular dalam format CSV, JSON Lines, dan Apache Parquet.

Prasyarat set data tabel dalam format CSV

Pekerjaan pemrosesan SageMaker Clarify dirancang untuk memuat file data CSV dalam dialek csv.excel. Namun, ini cukup fleksibel untuk mendukung terminator baris lainnya, termasuk `\n` dan`\r`.

Untuk kompatibilitas, semua file data CSV yang disediakan untuk pekerjaan pemrosesan SageMaker Clarify harus dikodekan dalam UTF-8.

Jika set data Anda tidak berisi baris header, lakukan hal berikut:

- Atur label konfigurasi analisis ke indeks`0`. Ini berarti bahwa kolom pertama adalah label ground truth.
- Jika parameter `headers` diatur, atur `label` ke header kolom label untuk menunjukkan lokasi kolom label. Semua kolom lainnya ditetapkan sebagai fitur.

  Berikut ini adalah contoh dari kumpulan data yang tidak berisi baris header.

```
1,5,2.8,2.538,This is a good product
0,1,0.79,0.475,Bad shopping experience
...
```

Jika data Anda berisi baris header, atur parameter `label` ke indeks`0`. Untuk menunjukkan lokasi kolom label, gunakan header label kebenaran dasar`Label`. Semua kolom lainnya ditetapkan sebagai fitur.

Berikut ini adalah contoh dari kumpulan data yang berisi baris header.

```
Label,Rating,A12,A13,Comments
1,5,2.8,2.538,This is a good product
0,1,0.79,0.475,Bad shopping experience
...
```

Prasyarat set data tabel dalam format JSON

JSON adalah format fleksibel untuk mewakili data terstruktur yang berisi tingkat kompleksitas apa pun. Dukungan SageMaker Clarify untuk JSON tidak terbatas pada format tertentu dan dengan demikian memungkinkan format data yang lebih fleksibel dibandingkan dengan kumpulan data dalam format CSV atau JSON Lines. Panduan ini menunjukkan cara mengatur konfigurasi analisis untuk data tabular dalam format JSON.

> **Note**
>
> Untuk memastikan kompatibilitas, semua file data JSON yang disediakan untuk pekerjaan pemrosesan SageMaker Clarify harus dikodekan dalam UTF-8.

Berikut ini adalah contoh data masukan dengan catatan yang berisi kunci tingkat atas, daftar fitur, dan label.

```
[
    {"features":[1,5,2.8,2.538,"This is a good product"],"label":1},
    {"features":[0,1,0.79,0.475,"Bad shopping experience"],"label":0},
    ...
]
```

Contoh analisis konfigurasi untuk contoh dataset input sebelumnya harus menetapkan parameter berikut:

- `label`Parameter harus menggunakan ekspresi JMESPath `[*].label` untuk mengekstrak label kebenaran dasar untuk setiap catatan dalam kumpulan data. Ekspresi JMESPath harus menghasilkan daftar label di mana label i th sesuai dengan catatan $^{ke\text{-}i}$.
- `features`Parameter harus menggunakan ekspresi JMESPath `[*].features` untuk mengekstrak array fitur untuk setiap record dalam dataset. Ekspresi JMESPath harus menghasilkan array atau matriks 2D di mana baris $^{ke\text{-}i}$ berisi nilai fitur yang sesuai dengan catatan ke-i.

Berikut ini adalah contoh data input dengan catatan yang berisi kunci tingkat atas dan kunci bersarang yang berisi daftar fitur dan label untuk setiap rekaman.

```
{
    "data": [
        {"features":[1,5,2.8,2.538,"This is a good product"],"label":1}},
        {"features":[0,1,0.79,0.475,"Bad shopping experience"],"label":0}}
    ]
}
```

Contoh analisis konfigurasi untuk contoh dataset input sebelumnya harus menetapkan parameter berikut:

- `label`Parameter menggunakan ekspresi JMESPath `data[*].label` untuk mengekstrak label kebenaran dasar untuk setiap catatan dalam kumpulan data. Ekspresi JMESPath harus menghasilkan daftar label di mana label $^{ke\text{-}i}$ adalah untuk catatan $^{ke\text{-}i}$.
- `features`Parameter menggunakan ekspresi JMESPath `data[*].features` untuk mengekstrak array fitur, untuk setiap record dalam dataset. Ekspresi JMESPath harus menghasilkan array atau matriks 2D di mana baris $^{ke\text{-}i}$ berisi nilai fitur untuk catatan ke-i.

Prasyarat kumpulan data tabel dalam format JSON Lines

JSON Lines adalah format teks untuk mewakili data terstruktur di mana setiap baris adalah objek JSON yang valid. Saat ini pekerjaan pemrosesan SageMaker Clarify hanya mendukung Garis JSON Format SageMaker Dense. Agar sesuai dengan format yang diperlukan, semua fitur catatan harus terdaftar dalam satu array JSON. Untuk informasi selengkapnya tentang JSON Lines, lihatFormat Permintaan JSONLINES.

> **Note**
>
> Semua file data JSON Lines yang disediakan untuk pekerjaan pemrosesan SageMaker Clarify harus dikodekan dalam UTF-8 untuk memastikan kompatibilitas.

Berikut ini adalah contoh cara mengatur konfigurasi analisis untuk catatan yang berisi kunci tingkat atas dan daftar elemen.

```
{"features":[1,5,2.8,2.538,"This is a good product"],"label":1}
{"features":[0,1,0.79,0.475,"Bad shopping experience"],"label":0}
...
```

Analisis konfigurasi untuk contoh kumpulan data sebelumnya harus menetapkan parameter sebagai berikut:

- Untuk menunjukkan lokasi label kebenaran dasar, parameter `label` harus disetel ke ekspresi JMESPath. `label`
- Untuk menunjukkan lokasi array fitur, parameter `features` harus diatur ke ekspresi JMESPath. `features`

Berikut ini adalah contoh cara mengatur konfigurasi analisis untuk catatan yang berisi kunci tingkat atas dan kunci bersarang yang berisi daftar elemen.

```
{"data":{"features":[1,5,2.8,2.538,"This is a good product"],"label":1}}
{"data":{"features":[0,1,0.79,0.475,"Bad shopping experience"],"label":0}}
...
```

Analisis konfigurasi untuk contoh kumpulan data sebelumnya harus menetapkan parameter sebagai berikut:

- Parameter `label` harus disetel ke ekspresi JMESPath `data.label` untuk menunjukkan lokasi label kebenaran dasar.
- Parameter `features` harus diatur ke ekspresi JMESPath `data.features` untuk menunjukkan lokasi array fitur.

Prasyarat kumpulan data tabel dalam format Parket

Parket adalah format data biner berorientasi kolom. Saat ini, SageMaker pekerjaan pemrosesan Clarify mendukung pemuatan file data Parket hanya ketika jumlah instance pemrosesan. 1

Karena pekerjaan pemrosesan SageMaker Clarify tidak mendukung permintaan titik akhir atau respons titik akhir dalam format Parket, Anda harus menentukan format data permintaan titik akhir dengan menyetel parameter konfigurasi analisis `content_type` ke format yang didukung. Untuk informasi selengkapnya, lihat `content_type` di Konfigurasikan Analisis.

Data Parket harus memiliki nama kolom yang diformat sebagai string. Gunakan `label` parameter konfigurasi analisis untuk mengatur nama kolom label untuk menunjukkan lokasi label kebenaran dasar. Semua kolom lainnya ditetapkan sebagai fitur.

Permintaan titik akhir untuk data tabular

Untuk mendapatkan prediksi model untuk analisis bias pasca-pelatihan dan analisis kepentingan fitur, SageMaker Clarify pekerjaan pemrosesan membuat serial data tabular menjadi byte dan mengirimkannya ke titik akhir inferensi sebagai muatan permintaan. Data tabular ini bersumber dari dataset input, atau dihasilkan. Jika itu adalah data sintetis, itu dihasilkan oleh penjelasan untuk analisis SHAP atau analisis PDP.

Format data payload permintaan harus ditentukan oleh `content_type` parameter konfigurasi analisis. Jika parameter tidak disediakan, pekerjaan pemrosesan SageMaker Clarify akan menggunakan nilai `dataset_type` parameter sebagai jenis konten. Untuk informasi lebih lanjut tentang `content_type` ataudataset_type, lihatKonfigurasikan Analisis.

Bagian berikut menunjukkan contoh permintaan titik akhir dalam format CSV dan JSON Lines.

Permintaan titik akhir dalam format CSV

Pekerjaan pemrosesan SageMaker Clarify dapat membuat serial data ke format CSV (tipe MIME:). `text/csv` Tabel berikut menunjukkan contoh muatan permintaan serial.

Muatan permintaan titik akhir (representasi string)	Komentar
'1,2,3,4'	Rekaman tunggal (empat fitur numerik).
'1,2,3,4\n5,6,7,8'	Dua catatan, dipisahkan oleh jeda baris '\n'.
“Ini adalah produk yang bagus”, 5'	Rekaman tunggal (fitur teks dan fitur numerik).
'"Ini adalah produk yang bagus” ,5\n“Peng alaman belanja yang buruk”, 1 '	Dua catatan.

Permintaan titik akhir dalam format JSON Lines

Pekerjaan pemrosesan SageMaker Clarify dapat membuat serial data ke format padat SageMaker JSON Lines (tipe MIME:). `application/jsonlines` Untuk informasi selengkapnya tentang JSON Lines, lihatFormat Permintaan JSONLINES.

Untuk mengubah data tabular menjadi data JSON, berikan string template ke parameter konfigurasi `content_template` analisis. Untuk informasi lebih lanjut tentang `content_template`, lihat Konfigurasikan Analisis. Tabel berikut menunjukkan contoh muatan permintaan JSON Lines serial.

Muatan permintaan titik akhir (representasi string)	Komentar
'{"data": {"features": [1,2,3,4]}}'	Rekor tunggal. Dalam hal ini, template terlihat seperti `'{"data":{"features":$features}}'` dan `$features` digantikan oleh daftar fitur`[1,2,3,4]` .
'{"data": {"features": [1,2,3,4]}}\n{"data": {"features": [5,6,7,8]}}'	Dua catatan.
'{"features": ["Ini adalah produk yang bagus" ,5]}'	Rekor tunggal. Dalam hal ini, template terlihat seperti `'{"features":$features}'` dan $features digantikan oleh daftar fitur`["This is a good product",5]` .
'{"features": ["Ini adalah produk yang bagus" ,5]}\n{"features": ["Pengalaman belanja yang buruk" ,1]}'	Dua catatan.

Permintaan titik akhir dalam format JSON

Pekerjaan pemrosesan SageMaker Clarify dapat membuat serial data ke struktur JSON arbitrer (tipe MIME:). `application/json` Untuk melakukannya, Anda harus memberikan string templat ke `content_template` parameter konfigurasi analisis. Ini digunakan oleh pekerjaan pemrosesan SageMaker Clarify untuk membangun struktur JSON luar. Anda juga harus menyediakan string template untuk`record_template`, yang digunakan untuk membangun struktur JSON untuk setiap

record. Untuk informasi selengkapnya tentang `content_template` dan `record_template`, lihat Konfigurasikan Analisis.

> **Note**
>
> Karena `content_template` dan `record_template` merupakan parameter string, setiap karakter kutipan ganda (") yang merupakan bagian dari struktur serial JSON harus dicatat sebagai karakter yang lolos dalam konfigurasi Anda. Misalnya, jika Anda ingin menghindari kutipan ganda di Python, Anda bisa memasukkan yang berikut untuk. `content_template`
>
> ```
> "{\"data\":{\"features\":$record}}}"
> ```

Tabel berikut menunjukkan contoh muatan permintaan JSON serial dan `record_template` parameter yang sesuai `content_template` dan yang diperlukan untuk membangunnya.

Muatan permintaan titik akhir (represen tasi string)	Komentar	content_template	record_template
'{"data": {"features": [1,2,3,4]}}'	Rekaman tunggal pada satu waktu.	'{"data": {"features" : $record}}}'	"$ fitur"
'{"instance": [[0, 1], [3, 4]], "nama-fitur": ["A", "B"]}'	Multi-rekaman dengan nama fitur.	'{"instance" :$records , "feature-names" : $feature_names}'	"$ fitur"
'[{"A": 0, "B": 1}, {"A": 3, "B": 4}]'	Pasangan multi-rek aman dan nilai kunci.	"$ catatan"	"$ features_kvp"
'{"A": 0, "B": 1}'	Catatan tunggal pada satu waktu dan pasangan nilai-kunci.	"$ rekor"	"$ features_kvp"
'{"A": 0, "bersarang": {"B": 1}}'	Atau, gunakan record_template verbose sepenuhnya untuk struktur arbitrer.	"$ rekor"	'{"A": "$ {A}", "bersarang": {"B": "$ {B}"}}'

Respon titik akhir untuk data tabular

Setelah pekerjaan pemrosesan SageMaker Clarify menerima respons pemanggilan titik akhir inferensi, ia mendeserialisasi muatan respons dan mengekstrak prediksi darinya. Gunakan `accept_type` parameter konfigurasi analisis untuk menentukan format data dari muatan respons. Jika tidak `accept_type` disediakan, pekerjaan pemrosesan SageMaker Clarify akan menggunakan nilai parameter content_type sebagai format keluaran model. Untuk informasi selengkapnya tentang `accept_type`, lihat Konfigurasikan Analisis.

Prediksi dapat terdiri dari label yang diprediksi untuk analisis bias, atau nilai probabilitas (skor) untuk analisis kepentingan fitur. Dalam konfigurasi `predictor` analisis, tiga parameter berikut mengekstrak prediksi.

- Parameter `probability` digunakan untuk menemukan nilai probabilitas (skor) dalam respons titik akhir.
- Parameter `label` digunakan untuk menemukan label yang diprediksi dalam respons titik akhir.
- (Opsional) Parameter `label_headers` menyediakan label yang diprediksi untuk model multiclass.

Pedoman berikut berkaitan dengan respons titik akhir dalam format CSV, JSON Lines, dan JSON.

Respon Endpoint dalam format CSV

Jika payload respons dalam format CSV (tipe MIME:`text/csv`), tugas pemrosesan SageMaker Clarify melakukan deserialisasi setiap baris. Kemudian mengekstrak prediksi dari data deserialisasi menggunakan indeks kolom yang disediakan dalam konfigurasi analisis. Baris dalam muatan respons harus sesuai dengan catatan dalam payload permintaan.

Tabel berikut memberikan contoh data respons dalam format yang berbeda dan untuk jenis masalah yang berbeda. Data Anda dapat bervariasi dari contoh-contoh ini, selama prediksi dapat diekstraksi sesuai dengan konfigurasi analisis.

Bagian berikut menunjukkan contoh respons titik akhir dalam format CSV.

Respons titik akhir dalam format CSV dan hanya berisi probabilitas

Tabel berikut adalah contoh respons titik akhir untuk masalah regresi dan klasifikasi biner.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekor tunggal.	'0,6'

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Dua catatan (menghasilkan satu baris, dibagi dengan koma).	'0,6,0,3'
Dua catatan (menghasilkan dua baris).	'0,6\n0,3'

Untuk contoh sebelumnya, titik akhir menghasilkan nilai probabilitas tunggal (skor) dari label yang diprediksi. Untuk mengekstrak probabilitas menggunakan indeks dan menggunakannya untuk analisis kepentingan fitur, atur parameter konfigurasi analisis `probability` ke indeks kolom. `0` Probabilitas ini juga dapat digunakan untuk analisis bias jika dikonversi ke nilai biner dengan menggunakan parameter. `probability_threshold` Untuk informasi selengkapnya tentang `probability_threshold`, lihat Konfigurasikan Analisis.

Tabel berikut adalah contoh respon endpoint untuk masalah multiclass.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal model multiclass (tiga kelas).	'0.1,0.6,0.3'
Dua catatan model multiclass (tiga kelas).	'0.1,0.6,0.3\n0.2,0.5,0.3'

Untuk contoh sebelumnya, titik akhir mengeluarkan daftar probabilitas (skor). Jika tidak ada indeks yang disediakan, semua nilai diekstraksi dan digunakan untuk analisis kepentingan fitur. Jika parameter konfigurasi analisis `label_headers` disediakan. Kemudian pekerjaan pemrosesan SageMaker Clarify dapat memilih tajuk label probabilitas maksimal sebagai label yang diprediksi, yang dapat digunakan untuk analisis bias. Untuk informasi selengkapnya tentang `label_headers`, lihat Konfigurasikan Analisis.

Respons titik akhir dalam format CSV dan hanya berisi label yang diprediksi

Tabel berikut adalah contoh respons titik akhir untuk masalah regresi dan klasifikasi biner.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'1'

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Dua catatan (hasil dalam satu baris, dibagi dengan koma)	'1,0'
Dua catatan (menghasilkan dua baris)	'1\n0'

Untuk contoh sebelumnya, titik akhir mengeluarkan label yang diprediksi, bukan probabilitas. Atur `label` parameter `predictor` konfigurasi ke indeks kolom `0` sehingga label yang diprediksi dapat diekstraksi menggunakan indeks dan digunakan untuk analisis bias.

Respons titik akhir dalam format CSV dan berisi label dan probabilitas yang diprediksi

Tabel berikut adalah contoh respons titik akhir untuk masalah regresi dan klasifikasi biner.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'1,0.6'
Dua catatan	'1,0.6\n0,0.3'

Untuk contoh sebelumnya, titik akhir mengeluarkan label yang diprediksi diikuti oleh probabilitasnya. Atur `label` parameter `predictor` konfigurasi ke indeks kolom`0`, dan atur `probability` ke indeks kolom 1 untuk mengekstrak kedua nilai parameter.

Respons titik akhir dalam format CSV dan berisi label dan probabilitas yang diprediksi (multiclass)

Model multiclass yang dilatih oleh Amazon SageMaker Autopilot dapat dikonfigurasi untuk menampilkan representasi string dari daftar label dan probabilitas yang diprediksi. Contoh tabel berikut menunjukkan contoh respon endpoint dari model yang dikonfigurasi untuk output`predicted_label`,, `probabilitylabels`, dan`probabilities`.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'"dog” ,0.6, "[\ 'cat\ ',\ 'dog\',\ 'fish\ ']”, "[0.1, 0.6, 0.3]"'

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Dua catatan	'"dog” ,0.6, "[\ 'cat\ ',\ 'dog\',\ 'fish\ ']”, "[0.1, 0.6, 0.3]”\n“" kucing” ,0.7, [\ 'cat\',\ 'dog\ ',\ 'ikan\']”, "[0.7, 0.2, 0.1]" '

Untuk contoh sebelumnya, pekerjaan pemrosesan SageMaker Clarify dapat dikonfigurasi dengan cara berikut untuk mengekstrak prediksi.

Untuk analisis bias, contoh sebelumnya dapat dikonfigurasi sebagai salah satu dari berikut ini.

- Atur `label` parameter `predictor` konfigurasi `0` untuk mengekstrak label yang diprediksi.
- Atur parameter 2 untuk mengekstrak label yang diprediksi, dan atur `probability` 3 untuk mengekstrak probabilitas yang sesuai. Pekerjaan pemrosesan SageMaker Clarify dapat secara otomatis menentukan label yang diprediksi dengan mengidentifikasi label dengan nilai probabilitas tertinggi. Mengacu pada contoh sebelumnya dari catatan tunggal, model memprediksi tiga label:`cat`,, dan `dogfish`, dengan probabilitas yang sesuai dari`0.1`,, `0.6` dan. `0.3` Berdasarkan probabilitas ini, label yang diprediksi adalah`dog`, karena memiliki nilai probabilitas tertinggi. `0.6`
- Setel `probability` 3 untuk mengekstrak probabilitas. Jika `label_headers` disediakan, maka pekerjaan pemrosesan SageMaker Clarify dapat secara otomatis menentukan label yang diprediksi dengan mengidentifikasi header label dengan nilai probabilitas tertinggi.

Untuk analisis kepentingan fitur, contoh sebelumnya dapat dikonfigurasi sebagai berikut.

- Atur `probability` untuk 3 mengekstrak probabilitas semua label yang diprediksi. Kemudian, atribusi fitur akan dihitung untuk semua label. Jika pelanggan tidak menentukan`label_headers`, maka label yang diprediksi akan digunakan sebagai header label dalam laporan analisis.

Respons Endpoint dalam format JSON Lines

Jika payload respons dalam format JSON Lines (tipe MIME:`application/jsonlines`), tugas pemrosesan SageMaker Clarify mendeserialisasi setiap baris sebagai JSON. Kemudian mengekstrak prediksi dari data deserialisasi menggunakan ekspresi JMESPath yang disediakan dalam konfigurasi analisis. Baris dalam muatan respons harus sesuai dengan catatan dalam payload permintaan. Tabel berikut menunjukkan contoh data respons dalam format yang berbeda. Data Anda dapat bervariasi dari contoh-contoh ini, selama prediksi dapat diekstraksi sesuai dengan konfigurasi analisis.

Bagian berikut menunjukkan contoh respons titik akhir dalam format JSON Lines.

Respons titik akhir dalam format JSON Lines dan hanya berisi probabilitas

Tabel berikut adalah contoh respons titik akhir yang hanya menampilkan nilai probabilitas (skor).

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'{"skor” :0.6}'
Dua catatan	'{"score” :0.6}\n{"skor” :0.3}'

Untuk contoh sebelumnya, atur parameter konfigurasi analisis `probability` ke ekspresi JMESPath “score” untuk mengekstrak nilainya.

Respons titik akhir dalam format JSON Lines dan hanya berisi label yang diprediksi

Tabel berikut adalah contoh respons titik akhir yang hanya menampilkan label yang diprediksi.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'{"prediksi” :1}'
Dua catatan	'{"prediksi” :1}\n{"prediksi” :0}'

Untuk contoh sebelumnya, atur `label` parameter konfigurasi prediktor ke ekspresi JMESPath. `prediction` Kemudian, pekerjaan pemrosesan SageMaker Clarify dapat mengekstrak label yang diprediksi untuk analisis bias. Untuk informasi selengkapnya, lihat Konfigurasikan Analisis.

Respons titik akhir dalam format JSON Lines dan berisi label dan probabilitas yang diprediksi

Tabel berikut adalah contoh respons titik akhir yang menampilkan label yang diprediksi dan skornya.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'{"prediksi” :1, "skor” :0.6}'
Dua catatan	'{"prediksi” :1, "skor” :0.6}\n{"prediksi” :0, "skor” :0.3}'

Untuk contoh sebelumnya, atur `label` parameter `predictor` konfigurasi ke ekspresi JMESPath "prediksi" untuk mengekstrak label yang diprediksi. Setel `probability` ke ekspresi JMESPath "skor" untuk mengekstrak probabilitas. Untuk informasi selengkapnya, lihat Konfigurasikan Analisis.

Respons titik akhir dalam format JSON Lines dan berisi label dan probabilitas yang diprediksi (multiclass)

Tabel berikut adalah contoh respons titik akhir dari model multiclass yang menghasilkan yang berikut:

- Daftar label yang diprediksi.
- Probabilitas, dan label prediksi yang dipilih dan probabilitasnya.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'{"predicted_label" :"dog", "probabilitas" :0.6, "predicted_labels": ["cat", "dog", "fish "], "probabilities": [0.1,0.6,0.3]}'
Dua catatan	'{"predicted_label" :"dog", "probabilitas" :0.6, "predicted_labels": ["cat", "dog", "fish "], "probabilities": [0.1,0.6,0.3]}\n{"predicted_label" :"cat", "probabilitas" :0.7, "predicted_labels": ["cat", "dog", "ikan "], "probabilitas": [0.7,0.2,0.1]}'

Untuk contoh sebelumnya, pekerjaan pemrosesan SageMaker Clarify dapat dikonfigurasi dengan beberapa cara untuk mengekstrak prediksi.

Untuk analisis bias, contoh sebelumnya dapat dikonfigurasi sebagai salah satu dari berikut ini.

- Setel `label` parameter `predictor` konfigurasi ke ekspresi JMESPath "predicted_label" untuk mengekstrak label yang diprediksi.
- Setel parameter ke ekspresi JMESPath "predicted_labels" untuk mengekstrak label yang diprediksi. Setel `probability` ke ekspresi JMESPath "probabilitas" untuk mengekstrak probabilitas mereka. Pekerjaan SageMaker Clarify secara otomatis menentukan label yang diprediksi dengan mengidentifikasi label dengan nilai probabilitas tertinggi.

- Setel `probability` ke ekspresi JMESPath “probabilitas” untuk mengekstrak probabilitas mereka. Jika `label_headers` disediakan, maka pekerjaan pemrosesan SageMaker Clarify dapat secara otomatis menentukan label yang diprediksi dengan mengidentifikasi label dengan nilai probabilitas tertinggi.

Untuk analisis kepentingan fitur, lakukan hal berikut.

- Setel `probability` ke ekspresi JMESPath “probabilitas” untuk mengekstrak probabilitas mereka dari semua label yang diprediksi. Kemudian, atribusi fitur akan dihitung untuk semua label.

Respon Endpoint dalam format JSON

Jika payload respons dalam format JSON (tipe MIME:`application/json`), tugas pemrosesan SageMaker Clarify mendeserialisasi seluruh muatan sebagai JSON. Kemudian mengekstrak prediksi dari data deserialisasi menggunakan ekspresi JMESPath yang disediakan dalam konfigurasi analisis. Catatan dalam muatan respons harus sesuai dengan catatan dalam payload permintaan.

Bagian berikut menunjukkan contoh respons titik akhir dalam format JSON. Bagian berisi tabel dengan contoh data respons dalam format yang berbeda dan untuk jenis masalah yang berbeda. Data Anda dapat bervariasi dari contoh-contoh ini, selama prediksi dapat diekstraksi sesuai dengan konfigurasi analisis.

Respons titik akhir dalam format JSON dan hanya berisi probabilitas

Tabel berikut adalah contoh respons dari titik akhir yang hanya menampilkan nilai probabilitas (skor).

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'[0,6]'
Dua catatan	'[0.6,0.3]'

Untuk contoh sebelumnya, tidak ada jeda baris di payload respons. Sebagai gantinya, satu objek JSON berisi daftar skor, satu untuk setiap catatan dalam permintaan. Atur parameter konfigurasi analisis `probability` ke ekspresi JMESPath “[*]” untuk mengekstrak nilainya.

Respons titik akhir dalam format JSON dan hanya berisi label yang diprediksi

Tabel berikut adalah contoh respons dari titik akhir yang hanya menampilkan label yang diprediksi.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'{"predicted_labels”: [1]}'
Dua catatan	'{"predicted_labels”: [1,0]}'

Setel `label` parameter `predictor` konfigurasi ke ekspresi JMESPath “predicted_labels”, dan kemudian pekerjaan pemrosesan SageMaker Clarify dapat mengekstrak label yang diprediksi untuk analisis bias.

Respons titik akhir adalah format JSON dan berisi label dan probabilitas yang diprediksi

Tabel berikut adalah contoh respons dari titik akhir yang menampilkan label yang diprediksi dan skornya.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'{"predictions”: [{"label” :1, "skor” :0.6}'
Dua catatan	'{"predictions”: [{"label” :1, "score” :0.6}, {"label” :0, "score” :0.3}]}'

Untuk contoh sebelumnya, atur `label` parameter `predictor` konfigurasi ke ekspresi JMESPath “predictions [*] .label” untuk mengekstrak label yang diprediksi. Setel `probability` ke ekspresi JMESPath “predictions [*] .score” untuk mengekstrak probabilitas.

Respons titik akhir dalam format JSON dan berisi label dan probabilitas yang diprediksi (multiclass)

Tabel berikut adalah contoh respons dari titik akhir yang dari model multiclass yang menghasilkan yang berikut:

- Daftar label yang diprediksi.
- Probabilitas, dan label prediksi yang dipilih dan probabilitasnya.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Rekaman tunggal	'[{"predicted_label” :"dog”, "probabilitas” :0.6, "predicted_labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.1,0.6,0.3]}]'
Dua catatan	'[{"predicted_label” :"dog”, "probabilitas” :0.6, "predicted_labels”: ["cat”, "dog”, "fish "], "probabilities”: [0.1,0.6,0.3]}, {"predict ed_label” :"cat”, "probabilitas” :0.7, "predicte d_labels”: ["cat”, "dog”, "ikan "], "probabilitas”: [0.7,0.2,0.1]}]'

Pekerjaan pemrosesan SageMaker Clarify dapat dikonfigurasi dengan beberapa cara untuk mengekstrak prediksi.

Untuk analisis bias, contoh sebelumnya dapat dikonfigurasi sebagai salah satu dari berikut ini.

- Setel `label` parameter `predictor` konfigurasi ke ekspresi JMESPath “[*] .predicted_label” untuk mengekstrak label yang diprediksi.
- Setel parameter ke ekspresi JMESPath “[*] .predicted_labels” untuk mengekstrak label yang diprediksi. Setel `probability` ke ekspresi JMESPath “[*] .probabilities” untuk mengekstrak probabilitasnya. Pekerjaan pemrosesan SageMaker Clarify dapat secara otomatis menentukan label yang diprediksi dengan mengidentifikasi label dengan nilai kedekatan tertinggi.
- Setel `probability` ke ekspresi JMESPath “[*] .probabilities” untuk mengekstrak probabilitasnya. Jika `label_headers` disediakan, maka pekerjaan pemrosesan SageMaker Clarify dapat secara otomatis menentukan label yang diprediksi dengan mengidentifikasi label dengan nilai probabilitas tertinggi.

Untuk analisis kepentingan fitur, atur `probability` ke ekspresi JMESPath “[*] .probabilities” untuk mengekstrak probabilitas mereka dari semua label yang diprediksi. Kemudian, atribusi fitur akan dihitung untuk semua label.

Pra-periksa permintaan titik akhir dan respons untuk data tabular

Kami menyarankan Anda menerapkan model Anda ke titik akhir inferensi SageMaker real-time, dan mengirim permintaan ke titik akhir. Periksa permintaan dan tanggapan secara manual untuk

memastikan bahwa keduanya sesuai dengan persyaratan di Permintaan titik akhir untuk data tabular bagian dan Respon titik akhir untuk data tabular bagian. Jika wadah model Anda mendukung permintaan batch, Anda dapat memulai dengan satu permintaan rekaman, lalu mencoba dua atau lebih catatan.

Perintah berikut menunjukkan cara meminta respons menggunakanAWS CLI. AWS CLIIni sudah diinstal sebelumnya di SageMaker Studio Classic, dan instance SageMaker Notebook. Jika Anda harus menginstalAWS CLI, ikuti panduan instalasi ini.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name $ENDPOINT_NAME \
  --content-type $CONTENT_TYPE \
  --accept $ACCEPT_TYPE \
  --body $REQUEST_DATA \
  $CLI_BINARY_FORMAT \
  /dev/stderr 1>/dev/null
```

Parameter didefinisikan, sebagai berikut.

- `$ENDPOINT_NAME` — Nama titik akhir.
- `$CONTENT_TYPE` — Jenis permintaan MIME (input wadah model).
- `$ACCEPT_TYPE` — Jenis respons MIME (keluaran kontainer model).
- `$REQUEST_DATA` — String payload yang diminta.
- `$CLI_BINARY_FORMAT` — Format parameter antarmuka baris perintah (CLI). Untuk AWS CLI v1, parameter ini harus tetap kosong. Untuk v2, parameter ini harus diatur ke`--cli-binary-format raw-in-base64-out`.

> **Note**
>
> AWS CLIv2 melewati parameter biner sebagai string yang dikodekan base64 secara default.

Contoh permintaan dan respons berikut ke dan dari titik akhir menggunakan AWS CLI v1.

Permintaan dan respons titik akhir dalam format CSV

Dalam contoh kode berikut, permintaan terdiri dari satu catatan dan responsnya adalah nilai probabilitasnya.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-sagemaker-xgboost-model \
  --content-type text/csv \
  --accept text/csv \
  --body '1,2,3,4' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
0.6
```

Dalam contoh kode berikut, permintaan terdiri dari dua catatan, dan respons mencakup probabilitasnya, yang dipisahkan oleh koma.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-sagemaker-xgboost-model \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, `$'content'` ekspresi dalam `--body` memberitahu perintah untuk menafsirkan `'\n'` konten sebagai jeda baris. Output respons berikut.

```
0.6,0.3
```

Dalam contoh kode berikut, permintaan terdiri dari dua catatan, respons mencakup probabilitasnya, dipisahkan dengan jeda baris.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-1 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
0.6
```

```
0.3
```

Dalam contoh kode berikut, permintaan terdiri dari catatan tunggal, dan responsnya adalah nilai probabilitas dari model multiclass yang berisi tiga kelas.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-1 \
  --content-type text/csv \
  --accept text/csv \
  --body '1,2,3,4' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
0.1,0.6,0.3
```

Dalam contoh kode berikut, permintaan terdiri dari dua catatan, dan responsnya mencakup nilai probabilitasnya dari model multiclass yang berisi tiga kelas.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-1 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
0.1,0.6,0.3
0.2,0.5,0.3
```

Dalam contoh kode berikut, permintaan terdiri dari dua catatan, dan responsnya mencakup label dan probabilitas yang diprediksi.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-2 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
1,0.6
0,0.3
```

Dalam contoh kode berikut, permintaan terdiri dari dua catatan dan responsnya mencakup header label dan probabilitas.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-3 \
  --content-type text/csv \
  --accept text/csv \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
"['cat','dog','fish']","[0.1,0.6,0.3]"
"['cat','dog','fish']","[0.2,0.5,0.3]"
```

Permintaan dan respons titik akhir dalam format JSON Lines

Dalam contoh kode berikut, permintaan terdiri dari satu catatan dan responsnya adalah nilai probabilitasnya.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-jsonlines \
  --content-type application/jsonlines \
  --accept application/jsonlines \
  --body '{"features":["This is a good product",5]}' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
{"score":0.6}
```

Dalam contoh kode berikut, permintaan berisi dua catatan, dan responsnya mencakup label dan probabilitas yang diprediksi.

```
aws sagemaker-runtime invoke-endpoint \
```

```
  --endpoint-name test-endpoint-jsonlines-2 \
  --content-type application/jsonlines \
  --accept application/jsonlines \
  --body $'{"features":[1,2,3,4]}\n{"features":[5,6,7,8]}' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
{"predicted_label":1,"probability":0.6}
{"predicted_label":0,"probability":0.3}
```

Dalam contoh kode berikut, permintaan berisi dua catatan, dan responsnya mencakup header label dan probabilitas.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-jsonlines-3 \
  --content-type application/jsonlines \
  --accept application/jsonlines \
  --body $'{"data":{"features":[1,2,3,4]}}\n{"data":{"features":[5,6,7,8]}}' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
{"predicted_labels":["cat","dog","fish"],"probabilities":[0.1,0.6,0.3]}
{"predicted_labels":["cat","dog","fish"],"probabilities":[0.2,0.5,0.3]}
```

Permintaan dan respons titik akhir dalam format campuran

Dalam contoh kode berikut, permintaan dalam format CSV dan responsnya dalam format JSON Lines.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-in-jsonlines-out \
  --content-type text/csv \
  --accept application/jsonlines \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
{"probability":0.6}
```

```
{"probability":0.3}
```

Dalam contoh kode berikut, permintaan dalam format JSON Lines dan responsnya dalam format CSV.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-jsonlines-in-csv-out \
  --content-type application/jsonlines \
  --accept text/csv \
  --body $'{"features":[1,2,3,4]}\n{"features":[5,6,7,8]}' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
0.6
0.3
```

Dalam contoh kode berikut, permintaan dalam format CSV dan responsnya dalam format JSON.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-csv-in-jsonlines-out \
  --content-type text/csv \
  --accept application/jsonlines \
  --body $'1,2,3,4\n5,6,7,8' \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
{"predictions":[{"label":1,"score":0.6},{"label":0,"score":0.3}]}
```

Data gambar

Pekerjaan pemrosesan SageMaker Clarify memberikan dukungan untuk menjelaskan gambar. Topik ini menyediakan persyaratan format data untuk data gambar. Untuk informasi selengkapnya, lihat computer vision.

Prasyarat set data gambar

Dataset gambar berisi satu atau lebih file gambar. Untuk mengidentifikasi kumpulan data input ke tugas pemrosesan SageMaker Clarify, tetapkan `dataset_uri` parameter konfigurasi ProcessingInputbernama `dataset` atau analisis ke awalan URI Amazon S3 dari file gambar Anda.

Format file gambar yang didukung dan ekstensi file tercantum dalam tabel berikut.

Format gambar	Ekstensi file
JPEG	jpg, jpeg
PNG	png

Atur `dataset_type` parameter konfigurasi analisis ke**`application/x-image`**. Karena jenisnya bukan format file gambar tertentu, `content_type` maka akan digunakan untuk menentukan format dan ekstensi file gambar.

Pekerjaan pemrosesan SageMaker Clarify memuat setiap file gambar ke NumPyarray 3 dimensi untuk diproses lebih lanjut. Tiga dimensi termasuk tinggi, lebar, dan nilai RGB dari setiap piksel.

Permintaan titik akhir untuk data gambar

Pekerjaan pemrosesan SageMaker Clarify mengubah data RGB mentah dari suatu gambar menjadi format gambar yang kompatibel, seperti JPEG. Ia melakukan ini sebelum mengirim data ke titik akhir untuk prediksi. Format gambar yang didukung adalah sebagai berikut.

Format Data	Tipe MIME	Ekstensi file
JPEG	`image/jpeg`	jpg, jpeg
PNG	`image/png`	png
NPY	`application/x-npy`	Semua di atas

Tentukan format data payload permintaan dengan menggunakan parameter `content_type` konfigurasi analisis. Jika tidak `content_type` disediakan, format data default ke. `image/jpeg`

Respons titik akhir untuk data gambar

Setelah menerima respons dari pemanggilan titik akhir inferensi, pekerjaan pemrosesan SageMaker Clarialize deserialisasi muatan respons dan kemudian mengekstrak prediksi darinya.

Masalah klasifikasi gambar

Format data payload respon harus ditentukan oleh parameter konfigurasi analisis accept_type. Jika tidak `accept_type` disediakan, format data default ke. `application/json` Format yang didukung sama dengan yang dijelaskan dalam respons Titik Akhir untuk data tabular di bagian data tabular.

Lihat Inferensi dengan Algoritma Klasifikasi Gambar contoh algoritma klasifikasi gambar SageMaker bawaan yang menerima satu gambar dan kemudian mengembalikan array nilai probabilitas (skor), masing-masing untuk kelas.

Seperti yang ditunjukkan pada tabel berikut, ketika `content_type` parameter diatur ke`application/jsonlines`, responsnya adalah objek JSON.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Gambar tunggal	'{"prediksi": [0.1,0.6,0.3]}'

Pada contoh sebelumnya, atur `probability` parameter ke ekspresi JMESPath “prediksi” untuk mengekstrak skor.

Ketika `content_type` diatur ke`application/json`, responsnya adalah objek JSON, seperti yang ditunjukkan dalam tabel berikut.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Gambar tunggal	'[0.1,0.6,0.3]'

Pada contoh sebelumnya, atur `probability` ke ekspresi JMESPath “[*]” untuk mengekstrak semua elemen array. Pada contoh sebelumnya, `[0.1, 0.6, 0.3]`diekstraksi. Atau, jika Anda melewatkan pengaturan parameter `probability` konfigurasi, maka semua elemen array juga diekstraksi. Ini karena seluruh muatan dideserialisasi sebagai prediksi.

Masalah deteksi objek

Konfigurasi analisis `accept_type` default `application/json` dan satu-satunya format yang didukung adalah Format Inferensi Deteksi Objek. Untuk informasi selengkapnya tentang format respons, lihatFormat Respons.

Tabel berikut adalah contoh respon dari endpoint yang output array. Setiap elemen array adalah array nilai yang berisi indeks kelas, skor kepercayaan, dan koordinat kotak pembatas dari objek yang terdeteksi.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Gambar tunggal (satu objek)	'[[4.0, 0,86419455409049988, 0,3088374 733924866, 0,07030484080314636, 0,7110607 028007507, 0,9345266819000244]]'
Gambar tunggal (dua objek)	'[[4.0, 0,86419455409049988, 0,3088374 733924866, 0,07030484080314636, 0,7110607 028007507, 0,9345266819000244], [0,0, 0,73376623392105103, 0,5714187026023865 , 0,40427327156066895, 0,827075183391571, 0,9712159633636475]]'

Tabel berikut adalah contoh respon dari endpoint yang output objek JSON dengan kunci mengacu pada array. Atur konfigurasi analisis `probability` ke “prediksi” kunci untuk mengekstrak nilai.

muatan permintaan akhir	Muatan respons titik akhir (representasi string)
Gambar tunggal (satu objek)	'{"prediksi”: [[4.0, 0.86419455409049988, 0.3088374733924866, 0.07030484080314636, 0.7110607028007507, 0.9345266819000244]]}'
Gambar tunggal (dua objek)	'{"prediksi”: [[4.0, 0.86419455409049988, 0.3088374733924866, 0.07030484080314636, 0.7110607028007507, 0.9345266819000244], [0.73376623392105103, 0.5714187026023602 3865, 0,40427327156066895, 0,8270751 83391571, 0,9712159633636475]]}'

Pra-periksa permintaan titik akhir dan respons untuk data gambar

Kami menyarankan Anda menerapkan model Anda ke titik akhir inferensi SageMaker real-time, dan mengirim permintaan ke titik akhir. Periksa permintaan dan tanggapan secara manual. Pastikan keduanya sesuai dengan persyaratan di bagian Endpoint request for image data dan Endpoint response for image data section.

Berikut ini adalah dua contoh kode yang menunjukkan cara mengirim permintaan dan memeriksa tanggapan untuk klasifikasi gambar dan masalah deteksi objek.

Masalah klasifikasi gambar

Contoh kode berikut menginstruksikan endpoint untuk membaca file PNG dan kemudian mengklasifikasikannya.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-sagemaker-image-classification \
  --content-type "image/png" \
  --accept "application/json" \
  --body fileb://./test.png  \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
[0.1,0.6,0.3]
```

Masalah deteksi objek

Contoh kode berikut menginstruksikan endpoint untuk membaca file JPEG dan kemudian mendeteksi objek di dalamnya.

```
aws sagemaker-runtime invoke-endpoint \
  --endpoint-name test-endpoint-sagemaker-object-detection \
  --content-type "image/jpg" \
  --accept "application/json" \
  --body fileb://./test.jpg  \
  /dev/stderr 1>/dev/null
```

Dari contoh kode sebelumnya, output respons mengikuti.

```
{"prediction":[[4.0, 0.86419455409049988, 0.3088374733924866, 0.07030484080314636,
 0.7110607028007507, 0.9345266819000244],[0.0, 0.73376623392105103, 0.5714187026023865,
```

```
 0.40427327156066895, 0.827075183391571, 0.9712159633636475],[4.0, 0.32643985450267792,
 0.3677481412887573, 0.034883320331573486, 0.6318609714508057, 0.5967587828636169],
[8.0, 0.22552496790885925, 0.6152569651603699, 0.5722782611846924, 0.882301390171051,
 0.8985623121261597],[3.0, 0.42260299175977707, 0.019305512309074402,
 0.08386176824569702, 0.39093565940856934, 0.9574796557426453]]}
```

Jalankan Pekerjaan Pemrosesan SageMaker Klarifikasi untuk Analisis Bias dan Penjelasan

Untuk menganalisis data dan model Anda untuk bias dan penjelasan menggunakan SageMaker Clarify, Anda harus mengonfigurasi pekerjaan pemrosesan SageMaker Clarify. Panduan ini menunjukkan cara mengonfigurasi input pekerjaan, output, sumber daya, dan konfigurasi analisis menggunakan SageMaker Python SDK API. `SageMakerClarifyProcessor`

API bertindak sebagai pembungkus API tingkat tinggi. SageMaker `CreateProcessingJob`
Ini menyembunyikan banyak detail yang terlibat dalam menyiapkan pekerjaan pemrosesan SageMaker Clarify. Detail untuk menyiapkan pekerjaan termasuk mengambil URI image kontainer SageMaker Clarify dan membuat file konfigurasi analisis. Langkah-langkah berikut menunjukkan cara mengonfigurasi, menginisialisasi, dan meluncurkan tugas pemrosesan SageMaker Clarify.

Mengonfigurasi tugas pemrosesan SageMaker Clarify menggunakan API

1. Tentukan objek konfigurasi untuk setiap bagian dari konfigurasi pekerjaan. Bagian-bagian ini dapat meliputi hal berikut:

   - Dataset input dan lokasi output: DataConfig.
   - Model atau titik akhir yang akan dianalisis: ModelConfig.
   - Parameter analisis bias: BiasConfig.
   - Parameter analisis Shapley Additive Explanations (SHAP): ShapConfig.

   Objek konfigurasi untuk pekerjaan pemrosesan SageMaker Clarify bervariasi untuk berbagai jenis format data dan kasus penggunaan. Contoh konfigurasi untuk data tabular dalam CSV dan JSON Lines format, pemrosesan bahasa alami (NLP), dan computer vision masalah disediakan di bagian berikut.
2. Buat `SageMakerClarifyProcessor` objek dan inisialisasi dengan parameter yang menentukan sumber daya pekerjaan. Sumber daya ini mencakup parameter seperti jumlah instance komputasi yang akan digunakan.

Contoh kode berikut menunjukkan cara membuat `SageMakerClarifyProcessor` objek dan menginstruksikannya untuk menggunakan satu contoh `ml.c4.xlarge` komputasi untuk melakukan analisis.

```
from sagemaker import clarify

clarify_processor = clarify.SageMakerClarifyProcessor(
    role=role,
    instance_count=1,
    instance_type='ml.c4.xlarge',
    sagemaker_session=session,
)
```

3. Panggil metode run spesifik SageMakerClarifyProcessorobjek dengan objek konfigurasi untuk kasus penggunaan Anda untuk meluncurkan pekerjaan. Metode run ini meliputi hal berikut:

   - `run_pre_training_bias`
   - `run_post_training_bias`
   - `run_bias`
   - `run_explainability`
   - `run_bias_and_explainability`

   Ini `SageMakerClarifyProcessor` menangani beberapa tugas di belakang layar. Tugas ini termasuk mengambil SageMaker Clarify container image universal resource identifier (URI), menyusun file konfigurasi analisis berdasarkan objek konfigurasi yang disediakan, mengunggah file ke bucket Amazon S3, dan mengonfigurasi tugas pemrosesan Clarify. SageMaker

   Bagian yang dapat diperluas berikut menunjukkan cara menghitung metrik, SHAPnilai, dan plot ketergantungan parsi pra-pelatihan dan pasca-pelatihan (). PDPs Bagian menunjukkan pentingnya fitur untuk tipe data ini:

   - Kumpulan data tabel dalam format CSV atau format Garis JSON
   - Kumpulan data pemrosesan bahasa alami (NLP)
   - Set data visi komputer

Panduan untuk menjalankan pekerjaan pemrosesan SageMaker Clarify paralel dengan pelatihan terdistribusi menggunakan Spark mengikuti bagian yang dapat diperluas.

Menganalisis data tabel dalam format CSV

Contoh berikut menunjukkan cara mengkonfigurasi analisis bias dan analisis penjelasan untuk dataset tabel dalam format CSV. Dalam contoh ini, dataset yang masuk memiliki empat kolom fitur dan satu kolom label biner,. `Target` Isi dataset adalah sebagai berikut. Nilai label `1` menunjukkan hasil positif.

```
Target,Age,Gender,Income,Occupation
0,25,0,2850,2
1,36,0,6585,0
1,22,1,1759,1
0,48,0,3446,1
...
```

`DataConfig`Objek ini menentukan dataset input dan di mana untuk menyimpan output. `s3_data_input_path`Parameter dapat berupa URI file dataset atau awalan URI Amazon S3. Jika Anda memberikan awalan URI S3, tugas pemrosesan SageMaker Clarify secara rekursif mengumpulkan semua file Amazon S3 yang terletak di bawah awalan. Nilai untuk `s3_output_path` harus berupa awalan URI S3 untuk menahan hasil analisis. SageMaker menggunakan `s3_output_path` while compiling, dan tidak dapat mengambil nilai parameter SageMaker Pipeline, properti, ekspresi, atau`ExecutionVariable`, yang digunakan selama runtime. Contoh kode berikut ini menunjukkan cara menentukan konfigurasi data untuk set data input sampel sebelumnya.

```
data_config = clarify.DataConfig(
    s3_data_input_path=dataset_s3_uri,
    dataset_type='text/csv',
    headers=['Target', 'Age', 'Gender', 'Income', 'Occupation'],
    label='Target',
    s3_output_path=clarify_job_output_s3_uri,
)
```

Cara menghitung semua metrik bias pra-pelatihan untuk kumpulan data CSV

Contoh kode berikut menunjukkan cara mengkonfigurasi `BiasConfig` objek untuk mengukur bias input sampel sebelumnya terhadap sampel dengan `Gender` nilai`0`.

```
bias_config = clarify.BiasConfig(
    label_values_or_threshold=[1],
```

```
    facet_name='Gender',
    facet_values_or_threshold=[0],
)
```

Contoh kode berikut menunjukkan cara menggunakan pernyataan run untuk meluncurkan pekerjaan pemrosesan SageMaker Clarify yang menghitung semua metrik bias pra-pelatihan untuk kumpulan data input.

```
clarify_processor.run_pre_training_bias(
     data_config=data_config,
    data_bias_config=bias_config,
    methods="all",
)
```

Atau, Anda dapat memilih metrik mana yang akan dihitung dengan menetapkan daftar metrik bias pra-pelatihan ke parameter metode. Misalnya, mengganti `methods="all"` dengan `methods=["CI", "DPL"]` menginstruksikan SageMaker Clarify Processor untuk menghitung hanya Ketidakseimbangan Kelas dan Perbedaan dalam Proporsi Label.

Cara menghitung semua metrik bias pasca-pelatihan untuk kumpulan data CSV

Anda dapat menghitung metrik bias pra-pelatihan sebelum pelatihan. Namun, untuk menghitung metrik bias pasca-pelatihan, Anda harus memiliki model yang terlatih. Contoh output berikut adalah dari model klasifikasi biner yang mengeluarkan data dalam format CSV. Dalam contoh output ini, setiap baris berisi dua kolom. Kolom pertama berisi label yang diprediksi, dan kolom kedua berisi nilai probabilitas untuk label tersebut.

```
0,0.028986845165491
1,0.825382471084594
...
```

Dalam konfigurasi contoh berikut, `ModelConfig` objek menginstruksikan pekerjaan untuk menyebarkan SageMaker model ke titik akhir singkat. Titik akhir menggunakan satu instance `ml.m4.xlarge` inferensi. Karena parameter `content_type` dan tidak `accept_type` diatur, mereka secara otomatis menggunakan nilai parameter`dataset_type`, yaitu`text/csv`.

```
model_config = clarify.ModelConfig(
    model_name=your_model,
    instance_type='ml.m4.xlarge',
```

```
    instance_count=1,
)
```

Contoh konfigurasi berikut menggunakan `ModelPredictedLabelConfig` objek dengan indeks label`0`. Ini menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menemukan label yang diprediksi di kolom pertama keluaran model. Pekerjaan Processing menggunakan pengindeksan berbasis nol dalam contoh ini.

```
predicted_label_config = clarify.ModelPredictedLabelConfig(
    label=0,
)
```

Dikombinasikan dengan contoh konfigurasi sebelumnya, contoh kode berikut meluncurkan pekerjaan pemrosesan SageMaker Clarify untuk menghitung semua metrik bias pasca-pelatihan.

```
clarify_processor.run_post_training_bias(
    data_config=data_config,
    data_bias_config=bias_config,
    model_config=model_config,
    model_predicted_label_config=predicted_label_config,
    methods="all",
)
```

Demikian pula, Anda dapat memilih metrik mana yang akan dihitung dengan menetapkan daftar metrik bias pasca-pelatihan ke parameter. `methods` Misalnya, ganti `methods=“all”` dengan `methods=["DPPL", "DI"]` untuk menghitung hanya Perbedaan Proporsi Positif dalam Label yang Diprediksi dan Dampak Berbeda.

Cara menghitung semua metrik bias untuk kumpulan data CSV

Contoh konfigurasi berikut menunjukkan cara menjalankan semua metrik bias pra-pelatihan dan pasca-pelatihan dalam satu SageMaker Klarifikasi pekerjaan pemrosesan.

```
clarify_processor.run_bias(
    data_config=data_config,
     bias_config=bias_config,
     model_config=model_config,
    model_predicted_label_config=predicted_label_config,
    pre_training_methods="all",
    post_training_methods="all",
```

```
)
```

Untuk contoh buku catatan dengan petunjuk tentang cara menjalankan pekerjaan pemrosesan SageMaker Clarify di SageMaker Studio Classic untuk mendeteksi bias, lihat Keadilan dan Keterjelasan dengan Clarify. SageMaker

Cara menghitung SHAP nilai untuk kumpulan data CSV

SageMaker Clarify menyediakan atribusi fitur menggunakan algoritma KernelShap. SHAPanalisis membutuhkan nilai probabilitas atau skor alih-alih label yang diprediksi, sehingga `ModelPredictedLabelConfig` objek ini memiliki indeks probabilitas`1`. Ini menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk mengekstrak skor probabilitas dari kolom kedua dari output model (menggunakan pengindeksan berbasis nol).

```
probability_config = clarify.ModelPredictedLabelConfig(
    probability=1,
)
```

`SHAPConfig`Objek menyediakan parameter SHAP analisis. Dalam contoh ini, SHAP `baseline` parameter dihilangkan dan nilai `num_clusters` parameternya. 1 Ini menginstruksikan Prosesor SageMaker Clarify untuk menghitung satu sampel SHAP dasar berdasarkan pengelompokan kumpulan data input. Jika Anda ingin memilih dataset dasar, lihat SHAPBaselines for Explainability.

```
shap_config = clarify.SHAPConfig(
    num_clusters=1,
)
```

Contoh kode berikut meluncurkan pekerjaan pemrosesan SageMaker Clarify untuk menghitung nilaiSHAP.

```
clarify_processor.run_explainability(
    data_config=data_config,
    model_config=model_config,
    model_scores=probability_config,
    explainability_config=shap_config,
)
```

Untuk contoh buku catatan dengan petunjuk tentang cara menjalankan tugas pemrosesan SageMaker Clarify di SageMaker Studio Classic untuk menghitung SHAP nilai, lihat Keadilan dan Keterjelasan dengan Clarify. SageMaker

Cara menghitung plot ketergantungan paral (PDPs) untuk kumpulan data CSV

PDPsmenunjukkan ketergantungan respons target yang diprediksi pada satu atau lebih fitur input yang menarik sambil mempertahankan semua fitur lainnya konstan. Garis miring ke atas, atau kurva di PDP, menunjukkan bahwa hubungan antara target dan fitur input positif, dan kecuraman menunjukkan kekuatan hubungan. Garis miring ke bawah atau kurva menunjukkan bahwa jika fitur input menurun, variabel target meningkat. Secara intuitif, Anda dapat menafsirkan ketergantungan paral sebagai respons variabel target untuk setiap fitur input yang diinginkan.

Contoh konfigurasi berikut adalah untuk menggunakan `PDPConfig` objek untuk SageMaker menginstruksikan pekerjaan pemrosesan Clarify untuk menghitung pentingnya fitur. `Income`

```
pdp_config = clarify.PDPConfig(
    features=["Income"],
    grid_resolution=10,
)
```

Pada contoh sebelumnya, `grid_resolution` parameter membagi rentang nilai `Income` fitur menjadi `10` bucket. Pekerjaan pemrosesan SageMaker Clarify akan menghasilkan PDPs untuk `Income` dibagi menjadi `10` segmen pada sumbu x. Sumbu y akan menunjukkan dampak marjinal `Income` pada variabel target.

Contoh kode berikut meluncurkan pekerjaan pemrosesan SageMaker Clarify untuk menghitungPDPs.

```
clarify_processor.run_explainability(
    data_config=data_config,
    model_config=model_config,
    model_scores=probability_config,
    explainability_config=pdp_config,
)
```

Untuk contoh buku catatan dengan petunjuk tentang cara menjalankan pekerjaan pemrosesan SageMaker Clarify di SageMaker Studio Classic untuk dihitungPDPs, lihat Explainability with SageMaker Clarify - Plot Ketergantungan Sebagian (PDP).

Cara menghitung SHAP nilai dan kumpulan data PDPs CSV

Anda dapat menghitung kedua SHAP nilai dan PDPs dalam satu pekerjaan pemrosesan SageMaker Clarify. Dalam contoh konfigurasi berikut, `top_k_features` parameter `PDPConfig` objek baru

diatur ke2. Ini menginstruksikan tugas pemrosesan SageMaker Clarify PDPs untuk menghitung 2 fitur yang memiliki nilai global SHAP terbesar.

```
shap_pdp_config = clarify.PDPConfig(
    top_k_features=2,
    grid_resolution=10,
)
```

Contoh kode berikut meluncurkan pekerjaan pemrosesan SageMaker Clarify untuk menghitung SHAP nilai dan. PDPs

```
clarify_processor.run_explainability(
    data_config=data_config,
    model_config=model_config,
    model_scores=probability_config,
    explainability_config=[shap_config, shap_pdp_config],
)
```

Analisis data tabular dalam format JSON Lines

Contoh berikut menunjukkan cara mengonfigurasi analisis bias dan analisis eksplainabilitas untuk kumpulan data tabular dalam format padat> SageMaker JSON Lines. Untuk informasi selengkapnya, lihat Format Permintaan JSONLINES. Dalam contoh ini, dataset yang masuk memiliki data yang sama dengan bagian sebelumnya, tetapi mereka dalam format JSON Lines. Setiap baris adalah objek JSON valid. `Features`Poin kunci ke array nilai fitur, dan `Label` poin-poin kunci ke label kebenaran dasar.

```
{"Features":[25,0,2850,2],"Label":0}
{"Features":[36,0,6585,0],"Label":1}
{"Features":[22,1,1759,1],"Label":1}
{"Features":[48,0,3446,1],"Label":0}
...
```

Dalam contoh konfigurasi berikut, `DataConfig` objek menentukan dataset input dan tempat menyimpan output.

```
data_config = clarify.DataConfig(
    s3_data_input_path=jsonl_dataset_s3_uri,
    dataset_type='application/jsonlines',
    headers=['Age', 'Gender', 'Income', 'Occupation', 'Target'],
```

```
    label='Label',
    features='Features',
    s3_output_path=clarify_job_output_s3_uri,
)
```

Dalam contoh konfigurasi sebelumnya, parameter fitur diatur ke ekspresi JMESPath `Features` sehingga pekerjaan pemrosesan SageMaker Clarify dapat mengekstrak array fitur dari setiap record. `label`Parameter diatur ke ekspresi JMESPath `Label` sehingga pekerjaan pemrosesan SageMaker Clarify dapat mengekstrak label kebenaran dasar dari setiap rekaman. `s3_data_input_path`Parameter dapat berupa URI file dataset atau awalan URI Amazon S3. Jika Anda memberikan awalan URI S3, pekerjaan pemrosesan SageMaker Clarify secara rekursif mengumpulkan semua file S3 yang terletak di bawah awalan. Nilai untuk `s3_output_path` harus berupa awalan URI S3 untuk menahan hasil analisis. SageMaker menggunakan `s3_output_path` while compiling, dan tidak dapat mengambil nilai parameter SageMaker Pipeline, properti, ekspresi, atau`ExecutionVariable`, yang digunakan selama runtime.

Anda harus memiliki model terlatih untuk menghitung metrik bias pasca-pelatihan atau kepentingan fitur. Contoh berikut adalah dari model klasifikasi biner yang mengeluarkan data JSON Lines dalam format contoh. Setiap baris output model adalah objek JSON valid. `predicted_label`Poin kunci untuk label yang diprediksi, dan `probability` poin-poin kunci untuk nilai probabilitas.

```
{"predicted_label":0,"probability":0.028986845165491}
{"predicted_label":1,"probability":0.825382471084594}
...
```

Dalam contoh konfigurasi berikut, `ModelConfig` objek menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menyebarkan SageMaker model ke titik akhir sementara. Titik akhir menggunakan satu instance `ml.m4.xlarge` inferensi.

```
model_config = clarify.ModelConfig(
    model_name=your_model,
    instance_type='ml.m4.xlarge',
    instance_count=1,
    content_template='{"Features":$features}',
)
```

Dalam contoh konfigurasi sebelumnya, parameter `content_type` dan tidak `accept_type` diatur. Oleh karena itu, mereka secara otomatis menggunakan nilai `dataset_type` parameter `DataConfig` objek, yaitu`application/jsonlines`. Pekerjaan pemrosesan SageMaker Clarify

menggunakan `content_template` parameter untuk menyusun input model dengan mengganti `$features` placeholder dengan array fitur.

Contoh konfigurasi berikut menunjukkan cara mengatur parameter label `ModelPredictedLabelConfig` objek ke ekspresi JMESPath. `predicted_label` Ini akan mengekstrak label yang diprediksi dari output model.

```
predicted_label_config = clarify.ModelPredictedLabelConfig(
    label='predicted_label',
)
```

Contoh konfigurasi berikut menunjukkan bagaimana mengatur `probability` parameter `ModelPredictedLabelConfig` objek untuk ekspresi JMESPath. `probability` Ini akan mengekstrak skor dari output model.

```
probability_config = clarify.ModelPredictedLabelConfig(
    probability='probability',
)
```

Untuk menghitung metrik bias dan kepentingan fitur untuk kumpulan data dalam format JSON Lines, gunakan pernyataan run dan objek konfigurasi yang sama seperti bagian sebelumnya untuk kumpulan data CSV. Anda dapat menjalankan tugas pemrosesan SageMaker Clarify di SageMaker Studio Classic untuk mendeteksi bias dan pentingnya fitur komputasi. Untuk instruksi dan contoh buku catatan, lihat Keadilan dan Keterjelasan dengan SageMaker Clarify (Format Garis JSON).

Menganalisis data tabular untuk penjelasan NLP

SageMaker Clarify mendukung penjelasan untuk model pemrosesan bahasa alami (NLP). Penjelasan ini membantu Anda memahami bagian teks mana yang paling penting untuk prediksi model Anda. Anda dapat menjelaskan prediksi model untuk satu contoh kumpulan data input, atau prediksi model dari set data dasar. Untuk memahami dan memvisualisasikan perilaku model, Anda dapat menentukan beberapa tingkat perincian. Untuk melakukan ini, tentukan panjang segmen teks, seperti token, kalimat, paragraf.

SageMaker Klarifikasi penjelasan NLP kompatibel dengan model klasifikasi dan regresi. Anda juga dapat menggunakan SageMaker Clarify untuk menjelaskan perilaku model Anda pada kumpulan data multi-modal yang berisi fitur teks, kategoris, atau numerik. Penjelasan NLP untuk kumpulan data multi-modal dapat membantu Anda memahami betapa pentingnya setiap fitur untuk keluaran model. SageMaker Clarify mendukung 62 bahasa dan dapat menangani teks yang mencakup beberapa bahasa.

Contoh berikut menunjukkan file konfigurasi analisis untuk fitur komputasi penting untuk NLP. Dalam contoh ini, kumpulan data yang masuk adalah kumpulan data tabular dalam format CSV, dengan satu kolom label biner dan dua kolom fitur.

```
0,2,"Flavor needs work"
1,3,"They taste good"
1,5,"The best"
0,1,"Taste is awful"
...
```

Contoh konfigurasi berikut menunjukkan cara menentukan dataset input dalam format CSV dan jalur data keluaran menggunakan objek. `DataConfig`

```
nlp_data_config = clarify.DataConfig(
    s3_data_input_path=nlp_dataset_s3_uri,
    dataset_type='text/csv',
    headers=['Target', 'Rating', 'Comments'],
    label='Target',
    s3_output_path=clarify_job_output_s3_uri,
)
```

Dalam contoh konfigurasi sebelumnya, `s3_data_input_path` parameter dapat berupa URI file dataset atau awalan URI Amazon S3. Jika Anda memberikan awalan URI S3, pekerjaan pemrosesan SageMaker Clarify secara rekursif mengumpulkan semua file S3 yang terletak di bawah awalan. Nilai untuk `s3_output_path` harus berupa awalan URI S3 untuk menahan hasil analisis. SageMaker menggunakan `s3_output_path` while compiling, dan tidak dapat mengambil nilai parameter SageMaker Pipeline, properti, ekspresi, atau`ExecutionVariable`, yang digunakan selama runtime.

Contoh output berikut dibuat dari model klasifikasi biner yang dilatih pada dataset input sebelumnya. Model klasifikasi menerima data CSV, dan menghasilkan skor tunggal di antara dan. `0` `1`

```
0.491656005382537
0.569582343101501
...
```

Contoh berikut menunjukkan cara mengonfigurasi `ModelConfig` objek untuk menyebarkan SageMaker model. Dalam contoh ini, titik akhir fana menyebarkan model. Titik akhir ini menggunakan satu instance `ml.g4dn.xlarge` inferensi yang dilengkapi dengan GPU, untuk inferensi yang dipercepat.

```
nlp_model_config = clarify.ModelConfig(
    model_name=your_nlp_model_name,
    instance_type='ml.g4dn.xlarge',
    instance_count=1,
)
```

Contoh berikut menunjukkan bagaimana mengkonfigurasi `ModelPredictedLabelConfig` objek untuk menemukan probabilitas (skor) di kolom pertama dengan indeks`0`.

```
probability_config = clarify.ModelPredictedLabelConfig(
    probability=0,
)
```

Contoh SHAP konfigurasi berikut menunjukkan cara menjalankan analisis penjelasan berdasarkan token menggunakan model dan dataset input dalam bahasa Inggris.

```
text_config = clarify.TextConfig(
    language='english',
    granularity='token',
)
nlp_shap_config = clarify.SHAPConfig(
    baseline=[[4, '[MASK]']],
    num_samples=100,
    text_config=text_config,
)
```

Pada contoh sebelumnya, `TextConfig` objek mengaktifkan analisis penjelasan NLP. `granularity`Parameter menunjukkan bahwa analisis harus mengurai token. Dalam bahasa Inggris, setiap token adalah sebuah kata. Untuk bahasa lain, lihat dokumentasi SPacy untuk tokenisasi, yang digunakan SageMaker Clarify untuk pemrosesan NLP. Contoh sebelumnya juga menunjukkan cara menggunakan rata-rata `Rating` `4` untuk menetapkan instance SHAP baseline di tempat. Token topeng khusus `[MASK]` digunakan untuk mengganti token (kata) di`Comments`.

Pada contoh sebelumnya, jika instancenya`2,"Flavor needs work"`, atur baseline ke rata-rata `4` dengan `Rating` baseline berikut.

```
4, '[MASK]'
```

Pada contoh sebelumnya, SageMaker Clarify menjelaskan iterasi melalui setiap token dan menggantinya dengan mask, sebagai berikut.

```
2,"[MASK] needs work"

4,"Flavor [MASK] work"

4,"Flavor needs [MASK]"
```

Kemudian, penjelasan SageMaker Clarify akan mengirim setiap baris ke model Anda untuk prediksi. Ini agar penjelas mempelajari prediksi dengan dan tanpa kata-kata bertopeng. SageMaker Penjelasan Clarify kemudian menggunakan informasi ini untuk menghitung kontribusi setiap token.

Contoh kode berikut meluncurkan pekerjaan pemrosesan SageMaker Clarify untuk menghitung nilaiSHAP.

```
clarify_processor.run_explainability(
    data_config=nlp_data_config,
    model_config=nlp_model_config,
    model_scores=probability_config,
    explainability_config=nlp_shap_config,
)
```

Untuk contoh buku catatan dengan petunjuk tentang cara menjalankan pekerjaan pemrosesan SageMaker Clarify di SageMaker Studio Classic untuk analisis penjelasan NLP, lihat Menjelaskan Analisis Sentimen Teks Menggunakan Klarifikasi. SageMaker

Menganalisis data gambar untuk penjelasan visi komputer

SageMaker Clarify menghasilkan peta panas yang memberikan wawasan tentang bagaimana model visi komputer Anda mengklasifikasikan dan mendeteksi objek dalam gambar Anda.

Dalam contoh konfigurasi berikut, dataset input terdiri dari gambar JPEG.

```
cv_data_config = clarify.DataConfig(
    s3_data_input_path=cv_dataset_s3_uri,
    dataset_type="application/x-image",
    s3_output_path=clarify_job_output_s3_uri,
)
```

Dalam contoh konfigurasi sebelumnya, `DataConfig` objek berisi `s3_data_input_path` set ke awalan Amazon S3 URI. Pekerjaan pemrosesan SageMaker Clarify secara rekursif mengumpulkan

semua file gambar yang terletak di bawah awalan. `s3_data_input_path`Parameter dapat berupa URI file dataset atau awalan URI Amazon S3. Jika Anda memberikan awalan URI S3, pekerjaan pemrosesan SageMaker Clarify secara rekursif mengumpulkan semua file S3 yang terletak di bawah awalan. Nilai untuk `s3_output_path` harus berupa awalan URI S3 untuk menahan hasil analisis. SageMaker menggunakan `s3_output_path` while compiling, dan tidak dapat mengambil nilai parameter SageMaker Pipeline, properti, ekspresi, atau`ExecutionVariable`, yang digunakan selama runtime.

Bagaimana menjelaskan model klasifikasi gambar

Pekerjaan pemrosesan SageMaker Clarify menjelaskan gambar menggunakan algoritma KernelShap, yang memperlakukan gambar sebagai kumpulan piksel super. Mengingat kumpulan data yang terdiri dari gambar, pekerjaan pemrosesan menghasilkan kumpulan data gambar di mana setiap gambar menunjukkan peta panas dari piksel super yang relevan.

Contoh konfigurasi berikut menunjukkan cara mengkonfigurasi analisis penjelasan menggunakan model klasifikasi SageMaker gambar. Untuk informasi selengkapnya, lihat Klasifikasi Gambar - MXNet.

```
ic_model_config = clarify.ModelConfig(
    model_name=your_cv_ic_model,
    instance_type="ml.p2.xlarge",
    instance_count=1,
    content_type="image/jpeg",
    accept_type="application/json",
)
```

Dalam contoh konfigurasi sebelumnya, model bernama`your_cv_ic_model`, telah dilatih untuk mengklasifikasikan hewan pada gambar JPEG masukan. `ModelConfig`Objek dalam contoh sebelumnya menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menyebarkan SageMaker model ke titik akhir sementara. Untuk inferensi yang dipercepat, titik akhir menggunakan satu instance `ml.p2.xlarge` inferensi yang dilengkapi dengan GPU.

Setelah gambar JPEG dikirim ke titik akhir, titik akhir mengklasifikasikannya dan mengembalikan daftar skor. Setiap skor adalah untuk kategori. `ModelPredictedLabelConfig`Objek memberikan nama setiap kategori, sebagai berikut.

```
ic_prediction_config = clarify.ModelPredictedLabelConfig(
    label_headers=['bird', 'cat', 'dog'],
```

```
)
```

Contoh keluaran untuk input sebelumnya dari ['bird', 'cat', 'dog'] bisa jadi 0.3,0.6,0.1, di mana 0.3 mewakili skor kepercayaan untuk mengklasifikasikan gambar sebagai burung.

Contoh SHAP konfigurasi berikut menunjukkan cara menghasilkan penjelasan untuk masalah klasifikasi gambar. Ini menggunakan `ImageConfig` objek untuk mengaktifkan analisis.

```
ic_image_config = clarify.ImageConfig(
    model_type="IMAGE_CLASSIFICATION",
    num_segments=20,
    segment_compactness=5,
)

ic_shap_config = clarify.SHAPConfig(
    num_samples=100,
    image_config=ic_image_config,
)
```

SageMaker Klarifikasi fitur ekstrak menggunakan metode Simple Linear Iterative Clustering (SLIC) dari pustaka scikit-learn untuk segmentasi gambar. Contoh konfigurasi sebelumnya, `model_type` parameter, menunjukkan jenis masalah klasifikasi gambar. Parameter `num_segments` memperkirakan berapa banyak perkiraan jumlah segmen yang akan diberi label dalam gambar input. Jumlah segmen kemudian diteruskan ke `n_segments` parameter slic.

Setiap segmen gambar dianggap super-piksel, dan SHAP nilai lokal dihitung untuk setiap segmen. Parameter `segment_compactness` menentukan bentuk dan ukuran segmen gambar yang dihasilkan oleh metode scikit-image slic. Ukuran dan bentuk segmen gambar kemudian diteruskan ke `compactness` parameter slic.

Contoh kode berikut meluncurkan pekerjaan pemrosesan SageMaker Clarify untuk menghasilkan peta panas untuk gambar Anda. Nilai peta panas positif menunjukkan bahwa fitur tersebut meningkatkan skor kepercayaan dalam mendeteksi objek. Nilai negatif menunjukkan bahwa fitur tersebut menurunkan skor kepercayaan.

```
clarify_processor.run_explainability(
    data_config=cv_data_config,
    model_config=ic_model_config,
    model_scores=ic_prediction_config,
    explainability_config=ic_shap_config,
```

```
)
```

Untuk contoh buku catatan yang menggunakan Clarify untuk SageMaker mengklasifikasikan gambar dan menjelaskan klasifikasinya, lihat Menjelaskan Klasifikasi Gambar dengan SageMaker Clarify.

Bagaimana menjelaskan model deteksi objek

Pekerjaan pemrosesan SageMaker Clarify dapat mendeteksi dan mengklasifikasikan objek dalam gambar dan kemudian memberikan penjelasan untuk objek yang terdeteksi. Proses penjelasannya adalah sebagai berikut.

1. Objek gambar pertama kali dikategorikan ke dalam salah satu kelas dalam koleksi tertentu. Misalnya, jika model deteksi objek dapat mengenali kucing, dog dan fish, maka ketiga kelas ini ada dalam koleksi. Koleksi ini ditentukan oleh `label_headers` parameter sebagai berikut.

   ```
   clarify.ModelPredictedLabelConfig(

   label_headers=object_categories,

   )
   ```

2. Pekerjaan pemrosesan SageMaker Clarify menghasilkan skor kepercayaan untuk setiap objek. Skor kepercayaan yang tinggi menunjukkan bahwa itu milik salah satu kelas dalam koleksi tertentu. Pekerjaan pemrosesan SageMaker Clarify juga menghasilkan koordinat kotak pembatas yang membatasi objek. Untuk informasi lebih lanjut tentang skor kepercayaan diri dan kotak pembatas, lihatFormat Respons.
3. SageMaker Clarify kemudian memberikan penjelasan untuk mendeteksi suatu objek dalam adegan gambar. Ini menggunakan metode yang dijelaskan di bagian Bagaimana menjelaskan model klasifikasi gambar.

Dalam contoh konfigurasi berikut, model deteksi SageMaker objek `your_cv_od_model` dilatih pada gambar JPEG untuk mengidentifikasi hewan pada mereka.

```
od_model_config = clarify.ModelConfig(
    model_name=your_cv_ic_model,
    instance_type="ml.p2.xlarge",
    instance_count=1,
    content_type="image/jpeg",
    accept_type="application/json",
```

```
)
```

`ModelConfig`Objek dalam contoh konfigurasi sebelumnya menginstruksikan pekerjaan pemrosesan SageMaker Clarify untuk menyebarkan SageMaker model ke titik akhir sementara. Untuk pencitraan yang dipercepat, titik akhir ini menggunakan satu instance `ml.p2.xlarge` inferensi yang dilengkapi dengan GPU.

Dalam konfigurasi contoh berikut, `ModelPredictedLabelConfig` objek memberikan nama setiap kategori untuk klasifikasi.

```
ic_prediction_config = clarify.ModelPredictedLabelConfig(
    label_headers=['bird', 'cat', 'dog'],
)
```

Contoh SHAP konfigurasi berikut menunjukkan cara menghasilkan penjelasan untuk deteksi objek.

```
od_image_config = clarify.ImageConfig(
    model_type="OBJECT_DETECTION",
    num_segments=20,
    segment_compactness=5,
    max_objects=5,
    iou_threshold=0.5,
    context=1.0,
)
od_shap_config = clarify.SHAPConfig(
    num_samples=100,
    image_config=image_config,
)
```

Dalam konfigurasi contoh sebelumnya, `ImageConfig` objek mengaktifkan analisis. `model_type`Parameter menunjukkan bahwa jenis masalah adalah deteksi objek. Untuk penjelasan mendetail tentang parameter lainnya, lihatKonfigurasikan Analisis.

Contoh kode berikut meluncurkan pekerjaan pemrosesan SageMaker Clarify untuk menghasilkan peta panas untuk gambar Anda. Nilai peta panas positif menunjukkan bahwa fitur tersebut meningkatkan skor kepercayaan dalam mendeteksi objek. Nilai negatif menunjukkan bahwa fitur tersebut menurunkan skor kepercayaan.

```
clarify_processor.run_explainability(
    data_config=cv_data_config,
```

```
    model_config=od_model_config,
    model_scores=od_prediction_config,
    explainability_config=od_shap_config,
)
```

Untuk contoh buku catatan yang menggunakan SageMaker Clarify untuk mendeteksi objek dalam gambar dan menjelaskan prediksinya, lihat Menjelaskan model deteksi objek dengan Amazon SageMaker Clarify.

Cara menjalankan pekerjaan pemrosesan SageMaker Clarify paralel

Saat bekerja dengan kumpulan data besar, Anda dapat menggunakan Apache Spark untuk meningkatkan kecepatan pekerjaan pemrosesan Clarify Anda SageMaker . Spark adalah mesin analitik terpadu untuk pemrosesan data skala besar. Bila Anda meminta lebih dari satu instance per prosesor SageMaker Clarify, SageMaker Clarify menggunakan kemampuan komputasi terdistribusi dari Spark.

Contoh konfigurasi berikut menunjukkan cara menggunakan `SageMakerClarifyProcessor` untuk membuat prosesor SageMaker Clarify dengan instance 5 komputasi. Untuk menjalankan pekerjaan apa pun yang terkait dengan`SageMakerClarifyProcessor`, SageMaker Klarifikasi menggunakan pemrosesan terdistribusi Spark.

```
from sagemaker import clarify

spark_clarify_processor = clarify.SageMakerClarifyProcessor(
    role=role,
    instance_count=5,
    instance_type='ml.c5.xlarge',
)
```

Jika Anda menyetel `save_local_shap_values` parameter ShapConfig ke`True`, pekerjaan pemrosesan SageMaker Clarify menyimpan SHAP nilai lokal sebagai beberapa file bagian di lokasi keluaran pekerjaan.

Untuk mengaitkan SHAP nilai lokal ke instance dataset input, gunakan `joinsource` parameter. `DataConfig` Jika Anda menambahkan lebih banyak instance komputasi, kami sarankan Anda juga meningkatkan `instance_count` dari ModelConfiguntuk titik akhir fana. Ini mencegah permintaan inferensi bersamaan pekerja Spark dari membanjiri titik akhir. Secara khusus, kami menyarankan Anda menggunakan one-to-one rasio endpoint-to-processing instans.

Mendapatkan hasil analisis

Topik ini menunjukkan cara mendapatkan hasil analisis yang dihasilkan SageMaker Clarify. Setelah tugas pemrosesan SageMaker Clarify selesai, Anda dapat mengunduh file keluaran untuk diperiksa, atau memvisualisasikan hasilnya di SageMaker Studio Classic.

Direktori SageMaker Clarify processing job output berisi file-file berikut:

- `analysis.json`— File yang berisi metrik bias dan kepentingan fitur dalam format JSON.
- `report.ipynb`— Notebook statis yang berisi kode untuk membantu Anda memvisualisasikan metrik bias dan kepentingan fitur.
- `explanations_shap/out.csv`— Direktori yang dibuat dan berisi file yang dibuat secara otomatis berdasarkan konfigurasi analisis spesifik Anda. Misalnya, jika Anda mengaktifkan `save_local_shap_values` parameter, maka nilai SHAP lokal per instance akan disimpan ke direktori. `explanations_shap` Sebagai contoh lain, jika Anda `analysis configuration` tidak berisi nilai untuk parameter dasar SHAP, pekerjaan Clarify SageMaker menjelaskan menghitung baseline dengan mengelompokkan kumpulan data input. Kemudian menyimpan baseline yang dihasilkan ke direktori.

Bagian berikut memberikan informasi rinci tentang skema dan laporan yang dihasilkan oleh analisis bias, analisis SHAP, analisis penjelasan visi komputer, dan analisis plot ketergantungan paral (PDP). Jika analisis konfigurasi berisi parameter untuk menghitung beberapa analisis, maka hasilnya dikumpulkan menjadi satu analisis dan satu file laporan.

Topik

- Analisis bias
- Analisis SHAP
- Analisis eksplainabilitas visi komputer (CV)
- Analisis plot ketergantungan sebagian (PDP)

Analisis bias

Amazon SageMaker Clarify menggunakan terminologi yang didokumentasikan Amazon SageMaker Klarifikasi Persyaratan untuk Bias dan Keadilan untuk membahas bias dan keadilan.

Skema untuk file analisis

File analisis dalam format JSON dan disusun menjadi dua bagian: metrik bias pra-pelatihan dan metrik bias pasca-pelatihan. Parameter untuk metrik bias pra-pelatihan dan pasca-pelatihan adalah sebagai berikut.

- pre_training_bias_metrics — Parameter untuk metrik bias pra-pelatihan. Lihat informasi yang lebih lengkap di Ukur Bias Pra-pelatihan dan Konfigurasikan Analisis.
  - label — Nama label kebenaran dasar yang ditentukan oleh `label` parameter konfigurasi analisis.
  - label_value_or_threshold — String yang berisi nilai label atau interval yang ditentukan oleh parameter konfigurasi analisis. `label_values_or_threshold` Misalnya, jika nilai `1` disediakan untuk masalah klasifikasi biner, maka string akan menjadi`1`. Jika beberapa nilai `[1,2]` disediakan untuk masalah multi-kelas, maka string akan`1,2`. Jika ambang batas `40` disediakan untuk masalah regresi, maka string akan menjadi internal seperti `(40, 68]` yang `68` merupakan nilai maksimum label dalam dataset input.
  - faset — Bagian ini berisi beberapa pasangan kunci-nilai, di mana kunci sesuai dengan nama facet yang ditentukan oleh `name_or_index` parameter konfigurasi facet, dan nilainya adalah array objek facet. Setiap objek faset memiliki anggota-anggota berikut:
    - value_or_threshold — String yang berisi nilai facet atau interval yang ditentukan oleh parameter konfigurasi facet. `value_or_threshold`
    - metrik - Bagian ini berisi array elemen metrik bias, dan setiap elemen metrik bias memiliki atribut berikut:
      - Nama: Nama singkat dari metrik bias. Misalnya, `CI`.
      - deskripsi — Nama lengkap metrik bias. Misalnya, `Class Imbalance (CI)`.
      - nilai — Nilai metrik bias, atau nilai nol JSON jika metrik bias tidak dihitung karena alasan tertentu. Nilai ± ∞ direpresentasikan sebagai string ∞ dan `-`∞ masing-masing.
      - error — Pesan kesalahan opsional yang menjelaskan mengapa metrik bias tidak dihitung.
- post_training_bias_metrics — Bagian ini berisi metrik bias pasca-pelatihan dan mengikuti tata letak dan struktur yang mirip dengan bagian pra-pelatihan. Untuk informasi selengkapnya, lihat Ukur Data Pasca-pelatihan dan Bias Model.

Berikut ini adalah contoh konfigurasi analisis yang akan menghitung metrik bias pra-pelatihan dan pasca-pelatihan.

```
{
    "version": "1.0",
    "pre_training_bias_metrics": {
        "label": "Target",
        "label_value_or_threshold": "1",
        "facets": {
            "Gender": [{
                "value_or_threshold": "0",
                "metrics": [
                    {
                        "name": "CDDL",
                        "description": "Conditional Demographic Disparity in Labels
 (CDDL)",
                        "value": -0.06
                    },
                    {
                        "name": "CI",
                        "description": "Class Imbalance (CI)",
                        "value": 0.6
                    },
                    ...
                ]
            }]
        }
    },
    "post_training_bias_metrics": {
        "label": "Target",
        "label_value_or_threshold": "1",
        "facets": {
            "Gender": [{
                "value_or_threshold": "0",
                "metrics": [
                    {
                        "name": "AD",
                        "description": "Accuracy Difference (AD)",
                        "value": -0.13
                    },
                    {
                        "name": "CDDPL",
                        "description": "Conditional Demographic Disparity in Predicted
 Labels (CDDPL)",
                        "value": 0.04
                    },
```

```
                    ...
                ]
            }]
        }
    }
}
```

Laporan analisis bias

Laporan analisis bias mencakup beberapa tabel dan diagram yang berisi penjelasan dan deskripsi terperinci. Ini termasuk, tetapi tidak terbatas pada, distribusi nilai label, distribusi nilai faset, diagram kinerja model tingkat tinggi, tabel metrik bias, dan deskripsinya. Untuk informasi selengkapnya tentang metrik bias dan cara menafsirkannya, lihat Pelajari Cara Amazon SageMaker Memperjelas Membantu Mendeteksi Bias.

Analisis SHAP

SageMaker Klarifikasi pekerjaan pemrosesan menggunakan algoritma Kernel SHAP untuk menghitung atribusi fitur. Pekerjaan pemrosesan SageMaker Clarify menghasilkan nilai SHAP lokal dan global. Ini membantu menentukan kontribusi setiap fitur terhadap prediksi model. Nilai SHAP lokal mewakili kepentingan fitur untuk setiap instance individu, sementara nilai SHAP global menggabungkan nilai SHAP lokal di semua instance dalam kumpulan data. Untuk informasi selengkapnya tentang nilai SHAP dan cara menafsirkannya, lihat. Atribusi Fitur yang Menggunakan Nilai Shapley

Skema untuk file analisis SHAP

Hasil analisis SHAP global disimpan di bagian penjelasan dari file analisis, di bawah metode ini`kernel_shap`. Parameter yang berbeda dari file analisis SHAP adalah sebagai berikut:

- penjelasan — Bagian dari file analisis yang berisi fitur penting hasil analisis.
  - kernal_shap — Bagian dari file analisis yang berisi hasil analisis SHAP global.
    - global_shap_values — Bagian dari file analisis yang berisi beberapa pasangan kunci-nilai. Setiap kunci dalam pasangan kunci-nilai mewakili nama fitur dari set data input. Setiap nilai dalam pasangan kunci-nilai sesuai dengan nilai SHAP global fitur. Nilai SHAP global diperoleh dengan menggabungkan nilai SHAP per instance fitur menggunakan konfigurasi. `agg_method` Jika `use_logit` konfigurasi diaktifkan, maka nilainya dihitung menggunakan koefisien regresi logistik, yang dapat diartikan sebagai rasio log-odds.

- expected_value — Prediksi rata-rata dari dataset dasar. Jika `use_logit` konfigurasi diaktifkan, maka nilainya dihitung menggunakan koefisien regresi logistik.
- global_top_shap_text - (Untuk analisis penjelasan NLP). Bagian dari file analisis yang mencakup serangkaian pasangan kunci-nilai. SageMaker Klarifikasi pekerjaan pemrosesan agregat nilai SHAP dari setiap token dan kemudian pilih token teratas berdasarkan nilai SHAP globalnya. `max_top_tokens`Konfigurasi mendefinisikan jumlah token yang akan dipilih.

  Masing-masing token teratas yang dipilih memiliki pasangan nilai kunci. Kunci dalam pasangan kunci-nilai sesuai dengan nama fitur teks token teratas. Setiap nilai dalam pasangan kunci-nilai adalah nilai SHAP global dari token teratas. Untuk contoh pasangan `global_top_shap_text` kunci-nilai, lihat output yang mengikuti.

Berikut ini menunjukkan output dari analisis SHAP dari kumpulan data tabel.

```
{
    "version": "1.0",
    "explanations": {
        "kernel_shap": {
            "Target": {
                 "global_shap_values": {
                    "Age": 0.022486410860333206,
                    "Gender": 0.007381025261958729,
                    "Income": 0.006843906804137847,
                    "Occupation": 0.006843906804137847,
                    ...
                },
                "expected_value": 0.508233428001
            }
        }
    }
}
```

Berikut ini menunjukkan output dari analisis SHAP dari kumpulan data teks. Output yang sesuai dengan kolom `Comments` adalah contoh output yang dihasilkan setelah analisis fitur teks.

```
{
    "version": "1.0",
    "explanations": {
        "kernel_shap": {
            "Target": {
```

```
                "global_shap_values": {
                    "Rating": 0.022486410860333206,
                    "Comments": 0.058612104851485144,
                    ...
                },
                "expected_value": 0.46700941970297033,
                "global_top_shap_text": {
                    "charming": 0.04127962903247833,
                    "brilliant": 0.02450240786522321,
                    "enjoyable": 0.024093569652715457,
                    ...
                }
            }
        }
    }
}
```

Skema untuk file baseline yang dihasilkan

Ketika konfigurasi dasar SHAP tidak disediakan, pekerjaan pemrosesan SageMaker Clarify menghasilkan kumpulan data dasar. SageMaker Clarify menggunakan algoritma pengelompokan berbasis jarak untuk menghasilkan dataset dasar dari cluster yang dibuat dari dataset input. Dataset dasar yang dihasilkan disimpan dalam file CSV, yang terletak di. `explanations_shap/baseline.csv` File output ini berisi baris header dan beberapa contoh berdasarkan `num_clusters` parameter yang ditentukan dalam konfigurasi analisis. Dataset dasar hanya terdiri dari kolom fitur. Contoh baseline yang dibuat dengan mengelompokkan dataset input berikut.

```
Age,Gender,Income,Occupation
35,0,2883,1
40,1,6178,2
42,0,4621,0
```

Skema untuk nilai SHAP lokal dari analisis keterjelasan kumpulan data tabular

Untuk kumpulan data tabular, jika satu instance komputasi digunakan, tugas pemrosesan SageMaker Clarify menyimpan nilai SHAP lokal ke file CSV bernama. `explanations_shap/out.csv` Jika Anda menggunakan beberapa instance komputasi, nilai SHAP lokal disimpan ke beberapa file CSV di direktori. `explanations_shap`

File keluaran yang berisi nilai SHAP lokal memiliki baris yang berisi nilai SHAP lokal untuk setiap kolom yang ditentukan oleh header. Header mengikuti konvensi penamaan `Feature_Label` di mana nama fitur ditambahkan oleh garis bawah, diikuti dengan nama variabel target Anda.

Untuk masalah multi-kelas, nama fitur di header bervariasi terlebih dahulu, lalu label. Misalnya, dua fitur`F1, F2`, dan dua kelas `L1` dan`L2`, di header adalah`F1_L1`,, `F2_L1F1_L2`, dan`F2_L2`. Jika konfigurasi analisis berisi nilai untuk `joinsource_name_or_index` parameter, maka kolom kunci yang digunakan dalam gabungan ditambahkan ke akhir nama header. Hal ini memungkinkan pemetaan nilai SHAP lokal ke instance dataset input. Contoh file output yang berisi nilai SHAP berikut.

```
Age_Target,Gender_Target,Income_Target,Occupation_Target
0.003937908,0.001388849,0.00242389,0.00274234
-0.0052784,0.017144491,0.004480645,-0.017144491
...
```

Skema untuk nilai SHAP lokal dari analisis penjelasan NLP

Untuk analisis penjelasan NLP, jika satu instance komputasi digunakan, tugas pemrosesan SageMaker Clarify menyimpan nilai SHAP lokal ke file JSON Lines bernama. `explanations_shap/out.jsonl` Jika Anda menggunakan beberapa instance komputasi, nilai SHAP lokal akan disimpan ke beberapa file JSON Lines dalam direktori. `explanations_shap`

Setiap file yang berisi nilai SHAP lokal memiliki beberapa baris data, dan setiap baris adalah objek JSON yang valid. Objek JSON memiliki atribut berikut:

- penjelasan — Bagian dari file analisis yang berisi array penjelasan Kernel SHAP untuk satu contoh. Setiap elemen dalam array memiliki anggota-anggota berikut:
  - feature_name — Nama header dari fitur yang disediakan oleh konfigurasi header.
  - data_type - Jenis fitur yang disimpulkan oleh pekerjaan pemrosesan Clarify. SageMaker Nilai yang valid untuk fitur teks termasuk`numerical`,`categorical`, dan `free_text` (untuk fitur teks).
  - atribusi — Array objek atribusi khusus fitur. Fitur teks dapat memiliki beberapa objek atribusi, masing-masing untuk unit yang ditentukan oleh `granularity` konfigurasi. Objek atribusi memiliki anggota-anggota berikut:
    - atribusi — Array nilai probabilitas khusus kelas.
    - deskripsi — (Untuk fitur teks) Deskripsi unit teks.
      - partial_text — Bagian teks yang dijelaskan oleh pekerjaan pemrosesan SageMaker Clarify.

- start_idx — Indeks berbasis nol untuk mengidentifikasi lokasi array yang menunjukkan awal fragmen teks paral.

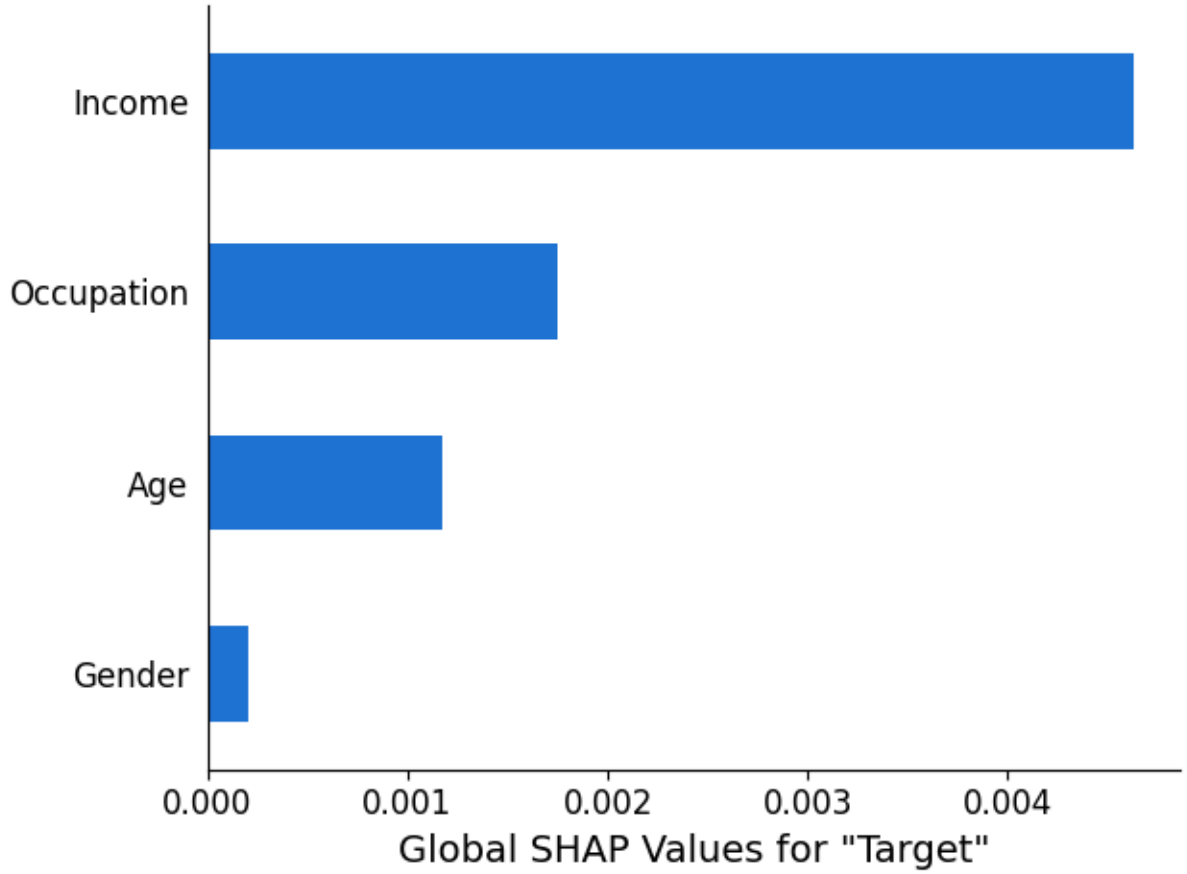
Berikut ini adalah contoh dari satu baris dari file nilai SHAP lokal, dipercantik untuk meningkatkan keterbacaannya.


```
{
    "explanations": [
        {
            "feature_name": "Rating",
            "data_type": "categorical",
            "attributions": [
                {
                    "attribution": [0.00342270632248735]
                }
            ]
        },
        {
            "feature_name": "Comments",
            "data_type": "free_text",
            "attributions": [
                {
                    "attribution": [0.005260534499999983],
                    "description": {
                        "partial_text": "It's",
                        "start_idx": 0
                    }
                },
                {
                    "attribution": [0.00424190349999996],
                    "description": {
                        "partial_text": "a",
                        "start_idx": 5
                    }
                },
                {
                    "attribution": [0.010247314500000014],
                    "description": {
                        "partial_text": "good",
                        "start_idx": 6
                    }
                },
```

```
                {
                    "attribution": [0.006148907500000005],
                    "description": {
                        "partial_text": "product",
                        "start_idx": 10
                    }
                }
            ]
        }
    ]
}
```

Laporan analisis SHAP

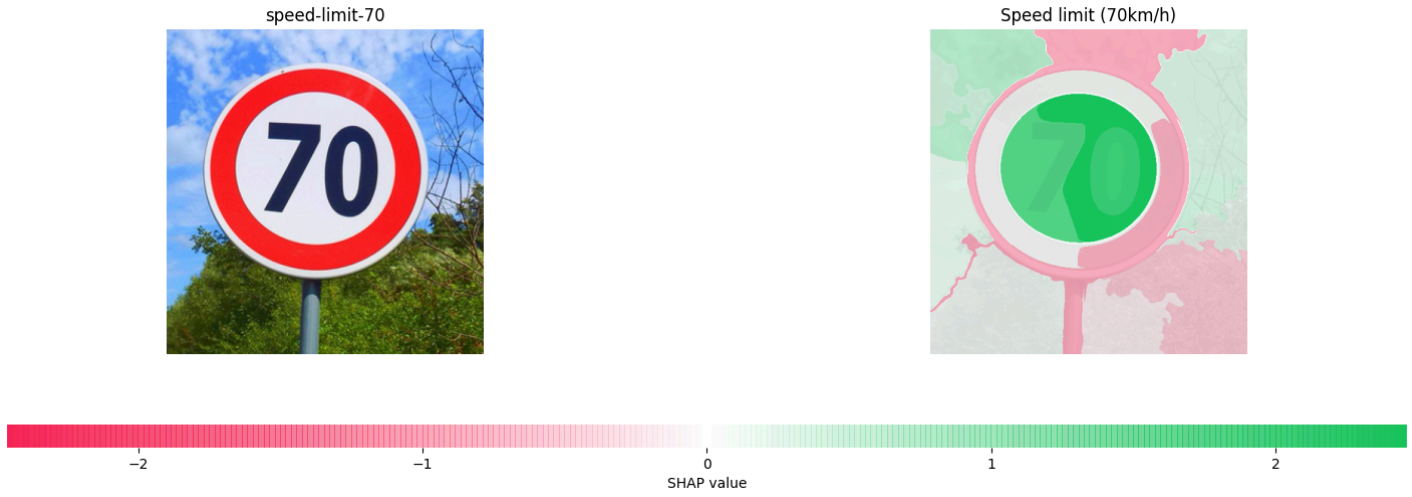
Laporan analisis SHAP menyediakan bagan batang maksimum nilai SHAP global `10` teratas. Contoh bagan berikut menunjukkan nilai SHAP untuk `4` fitur teratas.



Analisis eksplainabilitas visi komputer (CV)

SageMaker Clarify Computer Vision Explainability mengambil kumpulan data yang terdiri dari gambar dan memperlakukan setiap gambar sebagai kumpulan piksel super. Setelah analisis, pekerjaan pemrosesan SageMaker Clarify mengeluarkan kumpulan data gambar di mana setiap gambar menunjukkan peta panas piksel super.

Contoh berikut menunjukkan tanda batas kecepatan input di sebelah kiri dan peta panas menunjukkan besarnya nilai SHAP di sebelah kanan. Nilai SHAP ini dihitung dengan model pengenalan gambar Resnet-18 yang dilatih untuk mengenali rambu lalu lintas Jerman. Dataset German Traffic Sign Recognition Benchmark (GTSRB) disediakan dalam paper Man vs. Computer: Algoritma pembelajaran mesin benchmarking untuk pengenalan rambu lalu lintas. Dalam contoh output, nilai positif yang besar menunjukkan bahwa piksel super memiliki korelasi positif yang kuat dengan prediksi model. Nilai negatif yang besar menunjukkan bahwa piksel super memiliki korelasi negatif yang kuat dengan prediksi model. Semakin besar nilai absolut dari nilai SHAP yang ditunjukkan dalam peta panas, semakin kuat hubungan antara piksel super dan prediksi model.



Untuk informasi selengkapnya, lihat contoh buku catatan Menjelaskan Klasifikasi Gambar dengan SageMaker Memperjelas dan Menjelaskan model deteksi objek dengan Amazon Clarify. SageMaker

Analisis plot ketergantungan sebagian (PDP)

Plot ketergantungan sebagian menunjukkan ketergantungan respons target yang diprediksi pada serangkaian fitur input yang menarik. Ini dipinggirkan atas nilai-nilai semua fitur input lainnya dan disebut sebagai fitur pelengkap. Secara intuitif, Anda dapat menafsirkan ketergantungan sebagian sebagai respons target, yang diharapkan sebagai fungsi dari setiap fitur input yang menarik.

Skema untuk file analisis

Nilai PDP disimpan di `explanations` bagian file analisis di bawah pdp metode. Parameter untuk `explanations` adalah sebagai berikut:

- penjelasan — Bagian dari file analisis yang berisi fitur penting hasil analisis.
  - PDP — Bagian dari file analisis yang berisi array penjelasan PDP untuk satu contoh. Setiap elemen array memiliki anggota-anggota berikut:
    - feature_name — Nama header dari fitur yang disediakan oleh konfigurasi. `headers`
    - data_type - Jenis fitur yang disimpulkan oleh pekerjaan pemrosesan Clarify. SageMaker Nilai yang valid untuk `data_type` termasuk numerik dan kategoris.
    - feature_values - Berisi nilai-nilai yang ada dalam fitur. Jika yang `data_type` disimpulkan oleh SageMaker Clarify bersifat kategoris, `feature_values` berisi semua nilai unik yang dapat dimiliki fitur tersebut. Jika `data_type` disimpulkan oleh SageMaker Clarify adalah numerik, `feature_values` berisi daftar nilai pusat dari bucket yang dihasilkan. `grid_resolution`Parameter menentukan jumlah ember yang digunakan untuk mengelompokkan nilai kolom fitur.
    - data_distribution — Sebuah array persentase, di mana setiap nilai adalah persentase instance yang berisi bucket. `grid_resolution`Parameter tersebut menentukan jumlah bucket. Nilai kolom fitur dikelompokkan ke dalam ember ini.
    - model_predictions — Sebuah array prediksi model, di mana setiap elemen dari array adalah array prediksi yang sesuai dengan satu kelas dalam output model.

      label_headers — Header label yang disediakan oleh konfigurasi. `label_headers`
    - kesalahan — Pesan kesalahan yang dihasilkan jika nilai PDP tidak dihitung karena alasan tertentu. Pesan galat ini menggantikan konten yang terkandung dalam`feature_values`,`data_distributions`, dan `model_predictions` bidang.

Berikut ini adalah contoh output dari file analisis yang berisi hasil analisis PDP.

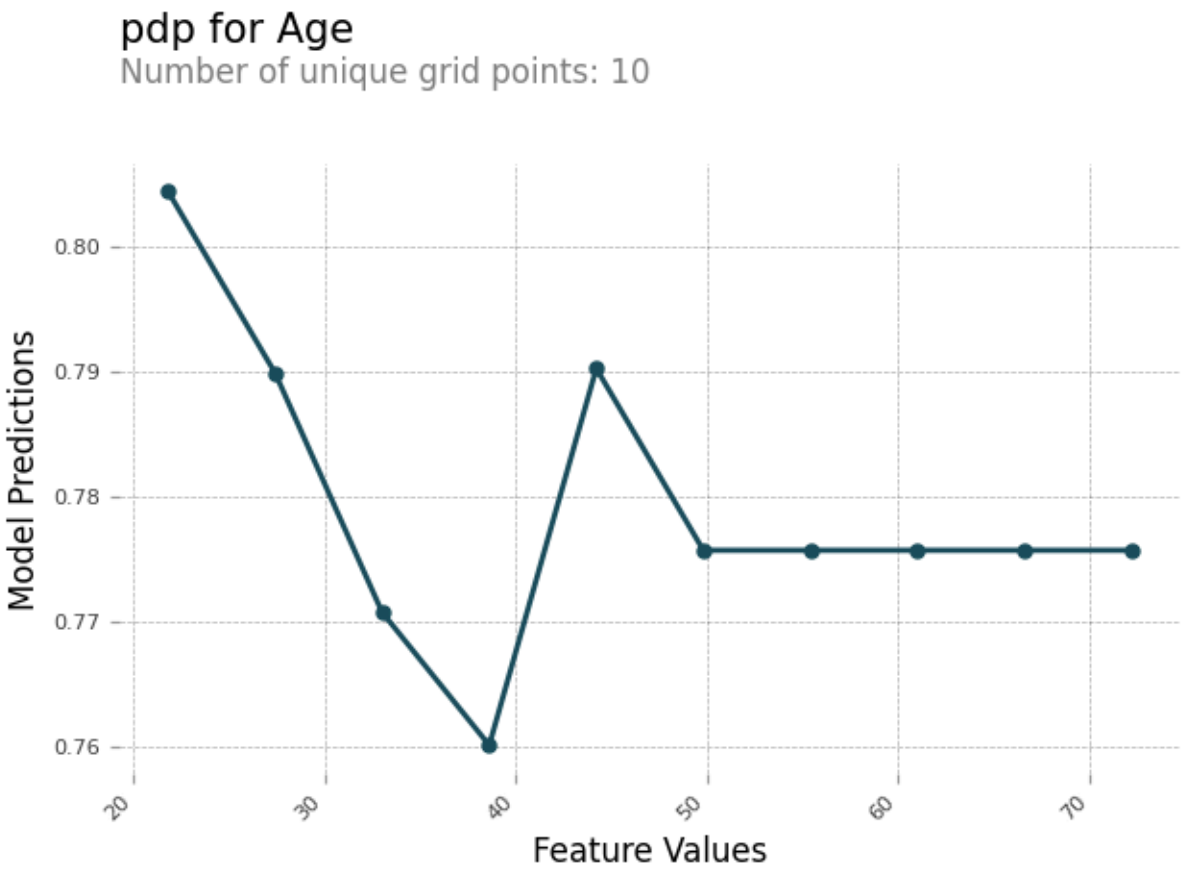
```
{
    "version": "1.0",
    "explanations": {
        "pdp": [
            {
                "feature_name": "Income",
                "data_type": "numerical",
```

```
                "feature_values": [1046.9, 2454.7, 3862.5, 5270.2, 6678.0, 8085.9,
 9493.6, 10901.5, 12309.3, 13717.1],
                "data_distribution": [0.32, 0.27, 0.17, 0.1, 0.045, 0.05, 0.01, 0.015,
 0.01, 0.01],
                "model_predictions": [[0.69, 0.82, 0.82, 0.77, 0.77, 0.46, 0.46, 0.45,
 0.41, 0.41]],
                "label_headers": ["Target"]
            },
            ...
        ]
    }
}
```

Laporan analisis PDP

Anda dapat membuat laporan analisis yang berisi bagan PDP untuk setiap fitur. Bagan PDP memplot `feature_values` sepanjang sumbu x, dan plot `model_predictions` sepanjang sumbu y. Untuk model multi-kelas, `model_predictions` adalah array, dan setiap elemen dari array ini sesuai dengan salah satu kelas prediksi model.

Berikut ini adalah contoh bagan PDP untuk fitur `Age` tersebut. Dalam contoh output, PDP menunjukkan jumlah nilai fitur yang dikelompokkan ke dalam ember. Jumlah bucket ditentukan oleh`grid_resolution`. Ember nilai fitur diplot terhadap prediksi model. Dalam contoh ini, nilai fitur yang lebih tinggi memiliki nilai prediksi model yang sama.



Memecahkan Masalah SageMaker Klarifikasi Pekerjaan Pemrosesan

Jika Anda mengalami kegagalan dengan SageMaker pekerjaan pemrosesan Clarify, lihat skenario berikut untuk membantu mengidentifikasi masalah.

> **Note**
>
> Alasan kegagalan dan pesan keluar dimaksudkan untuk berisi pesan deskriptif dan pengecualian, jika ditemui, selama dijalankan. Alasan umum untuk kesalahan adalah bahwa parameter hilang atau tidak valid. Jika Anda menemukan pesan yang tidak jelas, membingungkan, atau menyesatkan atau tidak dapat menemukan solusi, kirimkan umpan balik.

Topik

- Pekerjaan pemrosesan gagal diselesaikan
- Memproses pekerjaan terlalu lama untuk dijalankan

- Memproses pekerjaan selesai tanpa hasil dan Anda mendapatkan pesan CloudWatch peringatan
- Pesan galat untuk konfigurasi analisis yang tidak valid
- Perhitungan metrik bias gagal untuk beberapa atau semua metrik
- Ketidakcocokan antara konfigurasi analisis dan dataset/input/output model
- Model mengembalikan 500 Kesalahan Server Internal atau kontainer kembali ke prediksi per rekaman karena kesalahan model
- Peran eksekusi tidak valid
- Gagal mengunduh data
- Tidak dapat terhubung ke SageMaker

Pekerjaan pemrosesan gagal diselesaikan

Jika pekerjaan pemrosesan gagal selesai, Anda dapat mencoba hal berikut ini:

- Periksa log pekerjaan langsung di buku catatan tempat Anda menjalankan pekerjaan. Log pekerjaan terletak di output sel notebook tempat Anda memulai proses.
- Periksa log pekerjaan di CloudWatch.
- Tambahkan baris berikut di buku catatan Anda untuk menjelaskan pekerjaan pemrosesan terakhir dan cari alasan kegagalan dan pesan keluar:
  - `clarify_processor.jobs[-1].describe()`
- Jalankan perintah berikutAWS CLI; untuk menggambarkan pekerjaan pemrosesan dan mencari alasan kegagalan dan pesan keluar:
  - `aws sagemaker describe-processing-job –processing-job-name <processing-job-id>`

Memproses pekerjaan terlalu lama untuk dijalankan

Jika pekerjaan pemrosesan Anda terlalu lama untuk dijalankan, gunakan cara-cara berikut untuk menemukan akar masalahnya.

Periksa untuk melihat apakah konfigurasi sumber daya Anda cukup untuk menangani beban komputasi Anda. Untuk mempercepat pekerjaan Anda, coba hal berikut:

- Gunakan tipe instans yang lebih besar. SageMaker Klarifikasi kueri model berulang kali, dan instance yang lebih besar dapat secara signifikan mengurangi waktu komputasi Anda. Untuk

daftar instans yang tersedia, ukuran memori, bandwidth, dan detail kinerja lainnya, lihat SageMakerHarga Amazon.

- Tambahkan lebih banyak instans. SageMaker Clarify dapat menggunakan beberapa instance untuk menjelaskan beberapa titik data input secara paralel. Untuk mengaktifkan komputasi paralel, atur `instance_count` ke lebih dari 1 saat Anda meneleponSageMakerClarifyProcessor. Untuk informasi selengkapnya, lihat Cara menjalankan pekerjaan pemrosesan SageMaker Clarify paralel. Jika Anda meningkatkan jumlah instans, pantau kinerja titik akhir Anda untuk memastikan bahwa instans dapat menerapkan peningkatan beban. Untuk informasi selengkapnya, lihat Menangkap data dari titik akhir waktu nyata.
- Jika Anda menghitung SHapley Additive exPlanations (SHAP) nilai, kurangi `num_samples` parameter dalam file konfigurasi analisis Anda. Jumlah sampel secara langsung mempengaruhi hal-hal berikut:
  - Ukuran kumpulan data sintetis yang dikirim ke titik akhir Anda
  - Waktu aktif tugas

  Mengurangi jumlah sampel juga dapat menyebabkan berkurangnya akurasi dalam memperkirakan SHAP nilai. Untuk informasi selengkapnya, lihat Konfigurasikan Analisis.

Memproses pekerjaan selesai tanpa hasil dan Anda mendapatkan pesan CloudWatch peringatan

Jika pekerjaan pemrosesan selesai tetapi tidak ada hasil yang ditemukan, CloudWatch log menghasilkan pesan peringatan yang mengatakan Signal 15 diterima, dibersihkan. Peringatan ini menunjukkan bahwa pekerjaan dihentikan baik karena permintaan pelanggan disebut `StopProcessingJob` API, atau bahwa pekerjaan kehabisan waktu yang ditentukan untuk penyelesaiannya. Dalam kasus terakhir, periksa runtime maksimum dalam konfigurasi pekerjaan (`max_runtime_in_seconds`) dan tingkatkan sesuai kebutuhan.

Pesan galat untuk konfigurasi analisis yang tidak valid

- Jika Anda mendapatkan pesan kesalahan Tidak dapat memuat konfigurasi analisis sebagai JSON. , ini berarti bahwa file input konfigurasi analisis untuk pekerjaan pemrosesan tidak berisi objek JSON yang valid. Periksa validitas objek JSON menggunakan linter JSON.
- Jika Anda mendapatkan pesan kesalahan Kesalahan validasi skema konfigurasi analisis. , ini berarti bahwa file input konfigurasi analisis untuk pekerjaan pemrosesan berisi bidang yang tidak dikenal atau tipe yang tidak valid untuk beberapa nilai bidang. Tinjau parameter konfigurasi dalam

file dan periksa silang dengan parameter yang tercantum dalam file konfigurasi analisis. Untuk informasi selengkapnya, lihat Konfigurasikan Analisis.

Perhitungan metrik bias gagal untuk beberapa atau semua metrik

Jika Anda menerima salah satu pesan galat berikut Tidak ada nilai Label yang ada di Kolom Label yang diprediksi, Seri Indeks Prediksi Positif berisi semua nilai Salah. atau Tipe data seri Kolom Label Prediksi tidak sama dengan seri Kolom Label. , coba yang berikut ini:

- Periksa apakah dataset yang benar sedang digunakan.
- Periksa apakah ukuran dataset terlalu kecil; apakah, misalnya, hanya berisi beberapa baris. Hal ini dapat menyebabkan output model memiliki nilai yang sama atau tipe data disimpulkan secara tidak benar.
- Periksa apakah label atau faset diperlakukan sebagai kontinu atau kategoris. SageMaker Clarify menggunakan heuristik untuk menentukan. `DataType` Untuk metrik bias pasca-pelatihan, tipe data yang dikembalikan oleh model mungkin tidak cocok dengan apa yang ada dalam kumpulan data atau SageMaker Clarify mungkin tidak dapat mengubahnya dengan benar.
  - Dalam laporan bias, Anda akan melihat satu nilai untuk kolom kategoris atau interval untuk kolom kontinu.
  - Misalnya, jika kolom memiliki nilai 0.0 dan 1.0 sebagai float, itu akan diperlakukan sebagai kontinu bahkan jika ada terlalu sedikit nilai unik.

Ketidakcocokan antara konfigurasi analisis dan dataset/input/output model

- Periksa apakah format dasar dalam konfigurasi analisis sama dengan format kumpulan data.
- Jika Anda menerima pesan kesalahan Tidak dapat mengonversi string menjadi float. , periksa apakah formatnya ditentukan dengan benar. Ini juga dapat menunjukkan bahwa prediksi model memiliki format yang berbeda dari kolom label atau dapat menunjukkan bahwa konfigurasi untuk label atau probabilitas salah.
- Jika Anda menerima pesan kesalahan Tidak dapat menemukan facet. atau Header harus berisi label. atau Header dalam konfigurasi tidak cocok dengan jumlah kolom dalam kumpulan data. atau Nama fitur tidak ditemukan. , periksa apakah header cocok dengan kolom.
- Jika Anda menerima pesan kesalahan Data harus berisi fitur. , periksa template konten untuk JSON Lines dan bandingkan dengan sampel dataset jika tersedia.

Model mengembalikan 500 Kesalahan Server Internal atau kontainer kembali ke prediksi per rekaman karena kesalahan model

Jika Anda menerima pesan kesalahan Fallback ke prediksi per rekaman karena kesalahan model. , ini dapat menunjukkan bahwa model tidak dapat menangani ukuran batch, atau dibatasi, atau hanya tidak menerima input yang dilewatkan oleh wadah karena masalah serialisasi. Anda harus meninjau CloudWatch log untuk SageMaker titik akhir dan mencari pesan kesalahan atau traceback. Untuk kasus pelambatan model, mungkin membantu untuk menggunakan jenis instance yang berbeda atau meningkatkan jumlah instance untuk titik akhir.

Peran eksekusi tidak valid

Ini menunjukkan bahwa peran yang diberikan salah atau tidak ada izin yang diperlukan. Periksa peran dan izinnya yang digunakan untuk mengonfigurasi pekerjaan pemrosesan dan verifikasi kebijakan izin dan kepercayaan untuk peran tersebut.

Gagal mengunduh data

Ini menunjukkan bahwa input pekerjaan tidak dapat diunduh untuk memulai pekerjaan. Periksa nama bucket dan izin untuk dataset dan input konfigurasi.

Tidak dapat terhubung ke SageMaker

Ini menunjukkan bahwa pekerjaan tidak dapat mencapai titik akhir SageMaker layanan. Periksa pengaturan konfigurasi jaringan untuk pekerjaan pemrosesan dan verifikasi konfigurasi virtual private cloud (VPC).

Contoh notebook

Bagian berikut berisi buku catatan untuk membantu Anda mulai menggunakan SageMaker Clarify, menggunakannya untuk tugas-tugas khusus, termasuk yang ada di dalam pekerjaan terdistribusi, dan untuk visi komputer.

Memulai

Contoh buku catatan berikut menunjukkan cara menggunakan SageMaker Clarify untuk memulai tugas penjelasan dan bias model. Tugas-tugas ini termasuk membuat pekerjaan pemrosesan, melatih model pembelajaran mesin (ML), dan memantau prediksi model:

- Penjelasan dan deteksi bias dengan Amazon SageMaker Clarify — Gunakan SageMaker Clarify untuk membuat pekerjaan pemrosesan guna mendeteksi bias dan menjelaskan prediksi model.
- Memantau penyimpangan bias dan penyimpangan atribusi fitur Amazon SageMaker Clarify — Gunakan Monitor Model SageMaker Amazon untuk memantau penyimpangan bias dan fitur penyimpangan atribusi dari waktu ke waktu.
- Cara membaca kumpulan data dalam format JSON Lines menjadi pekerjaan pemrosesan SageMaker Clarify.
- Mitigasi Bias, latih model lain yang tidak bias, dan masukkan ke dalam registri model — Gunakan Teknik Pengambilan Sampel Minoritas Sintetis (SMOTE) dan SageMaker Klarifikasi untuk mengurangi bias, melatih model lain, lalu memasukkan model baru ke dalam registri model. Notebook contoh ini juga menunjukkan cara menempatkan artefak model baru, termasuk data, kode, dan metadata model, ke dalam registri model. Notebook ini adalah bagian dari seri yang menunjukkan cara mengintegrasikan SageMaker Clarify ke dalam SageMaker pipeline yang dijelaskan dalam Arsitek dan membangun siklus hidup pembelajaran mesin lengkap dengan AWS posting blog.

Kasus khusus

Buku catatan berikut menunjukkan cara menggunakan SageMaker Clarify untuk kasus khusus termasuk di dalam wadah Anda sendiri dan untuk tugas pemrosesan bahasa alami:

- Keadilan dan Keterjelasan dengan SageMaker Clarify (Bring Your Own Container) — Bangun model dan wadah Anda sendiri yang dapat diintegrasikan dengan SageMaker Clarify untuk mengukur bias dan menghasilkan laporan analisis penjelasan. Notebook contoh ini juga memperkenalkan istilah-istilah utama dan menunjukkan cara mengakses laporan melalui SageMaker Studio Classic.
- Keadilan dan Keterjelasan dengan SageMaker Clarify Spark Distributed Processing — Gunakan pemrosesan terdistribusi untuk menjalankan pekerjaan SageMaker Clarify yang mengukur bias pra-pelatihan dari kumpulan data dan bias pasca-pelatihan suatu model. Contoh buku catatan ini juga menunjukkan kepada Anda cara mendapatkan penjelasan tentang pentingnya fitur input pada keluaran model, dan mengakses laporan analisis eksplainabilitas melalui SageMaker Studio Classic.
- Keterjelasan dengan SageMaker Clarify - Plot Ketergantungan Sebagian (PDP) - Gunakan SageMaker Clarify untuk menghasilkan PDP dan mengakses laporan penjelasan model.
- Menjelaskan analisis sentimen teks menggunakan penjelasan SageMaker Clarify Natural language processing (NLP) — Gunakan SageMaker Clarify untuk analisis sentimen teks.

- Gunakan penjelasan visi komputer (CV) untuk klasifikasi gambar dan deteksi objek.

Notebook ini telah diverifikasi untuk berjalan di Amazon SageMaker Studio Classic. Jika Anda memerlukan petunjuk cara membuka buku catatan di Studio Classic, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic. Jika Anda diminta untuk memilih kernel, pilih Python 3 (Ilmu Data).

Mendeteksi Bias Data Pra-pelatihan

Bias algoritmik, diskriminasi, keadilan, dan topik terkait telah dipelajari lintas disiplin ilmu seperti hukum, kebijakan, dan ilmu komputer. Sistem komputer dapat dianggap bias jika mendiskriminasi individu atau kelompok individu tertentu. Model pembelajaran mesin yang mendukung aplikasi ini belajar dari data dan data ini dapat mencerminkan disparitas atau bias inheren lainnya. Misalnya, data pelatihan mungkin tidak memiliki representasi yang cukup dari berbagai kelompok demografis atau mungkin berisi label bias. Model pembelajaran mesin yang dilatih pada kumpulan data yang menunjukkan bias ini akhirnya dapat mempelajarinya dan kemudian mereproduksi atau bahkan memperburuk bias tersebut dalam prediksi mereka. Bidang pembelajaran mesin memberikan kesempatan untuk mengatasi bias dengan mendeteksi dan mengukurnya pada setiap tahap siklus hidup ML. Anda dapat menggunakan Amazon SageMaker Clarify untuk menentukan apakah data yang digunakan untuk model pelatihan mengkodekan bias apa pun

Bias dapat diukur sebelum pelatihan dan setelah pelatihan, dan dipantau terhadap garis dasar setelah menerapkan model ke titik akhir untuk inferensi. Metrik bias pra-pelatihan dirancang untuk mendeteksi dan mengukur bias dalam data mentah sebelum digunakan untuk melatih model. Metrik yang digunakan adalah model-agnostik karena tidak bergantung pada keluaran model apa pun. Namun, ada konsep keadilan yang berbeda yang membutuhkan ukuran bias yang berbeda. Amazon SageMaker Clarify menyediakan metrik bias untuk mengukur berbagai kriteria keadilan.

Untuk informasi tambahan tentang metrik bias, lihat Pelajari Cara Amazon SageMaker Clarify Membantu Mendeteksi Pengukuran Bias dan Keadilan untuk Machine Learning in Finance.

Amazon SageMaker Klarifikasi Persyaratan untuk Bias dan Keadilan

SageMaker Clarify menggunakan terminologi berikut untuk membahas bias dan keadilan.

Fitur

> Properti terukur individu atau karakteristik dari fenomena yang diamati, terkandung dalam kolom untuk data tabel.

Label

Fitur yang menjadi target pelatihan model machine learning. Disebut sebagai label yang diamati atau hasil yang diamati.

Label yang diprediksi

Label seperti yang diprediksi oleh model. Juga disebut sebagai hasil yang diprediksi.

Sampel

Entitas yang diamati dijelaskan oleh nilai fitur dan nilai label, yang terkandung dalam baris untuk data tabel.

Set data

Koleksi sampel.

Bias

Ketidakseimbangan dalam data pelatihan atau perilaku prediksi model di berbagai kelompok, seperti usia atau kelompok pendapatan. Bias dapat dihasilkan dari data atau algoritma yang digunakan untuk melatih model Anda. Misalnya, jika model ML dilatih terutama pada data dari individu paruh baya, mungkin kurang akurat ketika membuat prediksi yang melibatkan orang yang lebih muda dan lebih tua.

Metrik bias

Fungsi yang mengembalikan nilai numerik yang menunjukkan tingkat bias potensial.

Laporan bias

Kumpulan metrik bias untuk kumpulan data tertentu, atau kombinasi kumpulan data dan model.

Nilai label positif

Nilai label yang menguntungkan kelompok demografis yang diamati dalam sampel. Dengan kata lain, menunjuk sampel sebagai memiliki hasil positif.

Nilai label negatif

Nilai label yang tidak menguntungkan bagi kelompok demografis yang diamati dalam sampel. Dengan kata lain, menunjuk sampel sebagai memiliki hasil negatif.

Variabel kelompok

Kolom kategoris dari kumpulan data yang digunakan untuk membentuk subkelompok untuk pengukuran Disparitas Demografis Bersyarat (CDD). Diperlukan hanya untuk metrik ini sehubungan dengan paradoks Simpson.

Faset

Kolom atau fitur yang berisi atribut sehubungan dengan bias yang diukur.

Nilai segi

Nilai fitur atribut yang mungkin disukai atau tidak disukai oleh bias.

Probabilitas yang diprediksi

Probabilitas, seperti yang diprediksi oleh model, dari sampel yang memiliki hasil positif atau negatif.

Contoh Notebook

Amazon SageMaker Clarify menyediakan contoh notebook berikut untuk deteksi bias:

- Penjelasan dan deteksi bias dengan Amazon SageMaker Clarify — Gunakan SageMaker Clarify untuk membuat pekerjaan pemrosesan untuk mendeteksi bias dan menjelaskan prediksi model dengan atribusi fitur.

Notebook ini telah diverifikasi untuk berjalan di Amazon SageMaker Studio saja. Jika Anda memerlukan petunjuk tentang cara membuka buku catatan di Amazon SageMaker Studio, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic. Jika Anda diminta untuk memilih kernel, pilih Python 3 (Ilmu Data).

Topik

- Ukur Bias Pra-pelatihan
- Hasilkan Laporan untuk Bias dalam Data Pra-pelatihan di Studio SageMaker

Ukur Bias Pra-pelatihan

Mengukur bias dalam model ML adalah langkah pertama untuk mengurangi bias. Setiap ukuran bias sesuai dengan gagasan keadilan yang berbeda. Bahkan mempertimbangkan konsep keadilan yang sederhana mengarah pada banyak ukuran berbeda yang berlaku dalam berbagai konteks. Misalnya,

pertimbangkan keadilan sehubungan dengan usia, dan, untuk kesederhanaan, bahwa paruh baya dan kelompok usia lainnya adalah dua demografi yang relevan, yang disebut sebagai aspek. Dalam kasus model ML untuk pinjaman, kita mungkin ingin pinjaman usaha kecil dikeluarkan dengan jumlah yang sama dari kedua demografi. Atau, saat memproses pelamar kerja, kami mungkin ingin melihat jumlah anggota yang sama dari setiap demografis yang dipekerjakan. Namun, pendekatan ini mungkin mengasumsikan bahwa jumlah yang sama dari kedua kelompok umur berlaku untuk pekerjaan ini, jadi kami mungkin ingin mengkondisikan nomor yang berlaku. Lebih lanjut, kami mungkin ingin mempertimbangkan bukan apakah angka yang sama berlaku, tetapi apakah kami memiliki jumlah pelamar yang memenuhi syarat yang sama. Atau, kami dapat menganggap keadilan sebagai tingkat penerimaan yang sama dari pelamar yang memenuhi syarat di kedua demografi usia, atau, tingkat penolakan pelamar yang sama, atau keduanya. Anda dapat menggunakan kumpulan data dengan proporsi data yang berbeda pada atribut yang diinginkan. Ketidakseimbangan ini dapat menggabungkan ukuran bias yang Anda pilih. Model mungkin lebih akurat dalam mengklasifikasikan satu aspek daripada yang lain. Dengan demikian, Anda perlu memilih metrik bias yang secara konseptual sesuai untuk aplikasi dan situasi.

Kami menggunakan notasi berikut untuk membahas metrik bias. Model konseptual yang dijelaskan di sini adalah untuk klasifikasi biner, di mana peristiwa diberi label hanya memiliki dua kemungkinan hasil dalam ruang sampelnya, disebut sebagai positif (dengan nilai 1) dan negatif (dengan nilai 0). Kerangka kerja ini biasanya dapat diperluas ke klasifikasi multikategori secara langsung atau untuk kasus-kasus yang melibatkan hasil bernilai berkelanjutan bila diperlukan. Dalam kasus klasifikasi biner, label positif dan negatif ditetapkan ke hasil yang dicatat dalam kumpulan data mentah untuk aspek yang disukai a dan untuk aspek yang tidak disukai d. Label y ini disebut sebagai label yang diamati untuk membedakannya dari label yang diprediksi y' yang ditetapkan oleh model pembelajaran mesin selama tahap pelatihan atau kesimpulan dari siklus hidup ML. Label ini digunakan untuk menentukan distribusi probabilitas $P_a(y)$ dan $P_d(y)$ untuk hasil faset masing-masing.

- label:
  - y mewakili n label yang diamati untuk hasil peristiwa dalam kumpulan data pelatihan.
  - y' mewakili label yang diprediksi untuk n label yang diamati dalam kumpulan data oleh model terlatih.
- hasil:
  - Hasil positif (dengan nilai 1) untuk sampel, seperti penerimaan aplikasi.
    - $n^{(1)}$ adalah jumlah label yang diamati untuk hasil positif (penerimaan).
    - $n'^{(1)}$ adalah jumlah label yang diprediksi untuk hasil positif (penerimaan).
  - Hasil negatif (dengan nilai 0) untuk sampel, seperti penolakan aplikasi.

  - $n^{(0)}$ adalah jumlah label yang diamati untuk hasil negatif (penolakan).
  - $n'^{(0)}$ adalah jumlah label yang diprediksi untuk hasil negatif (penolakan).
- nilai segi:
  - facet a - Nilai fitur yang mendefinisikan demografis yang disukai bias.
    - n_a adalah jumlah label yang diamati untuk nilai faset yang disukai: $n_a = n_a^{(1)} + n_a^{(0)}$ jumlah label yang diamati positif dan negatif untuk aspek nilai a.
    - n'_a adalah jumlah label yang diprediksi untuk nilai faset yang disukai: $n'_a = n'^{(1)}_a + n'^{(0)}_a$ jumlah label hasil prediksi positif dan negatif untuk nilai faset a. Perhatikan bahwa $n'_a = n_a$.
  - facet d — Nilai fitur yang mendefinisikan demografis yang bias tidak disukai.
    - n_d adalah jumlah label yang diamati untuk nilai faset yang tidak disukai: $n_d = n_d^{(1)} + n_d^{(0)}$ jumlah label yang diamati positif dan negatif untuk nilai faset d.
    - n'_d adalah jumlah label yang diprediksi untuk nilai faset yang tidak disukai: $n'_d = n'^{(1)}_d + n'^{(0)}_d$ jumlah label prediksi positif dan negatif untuk nilai faset d. Perhatikan bahwa $n'_d = n_d$.
- distribusi probabilitas untuk hasil dari hasil data facet berlabel:
  - $P_a(y)$ adalah distribusi probabilitas dari label yang diamati untuk faset a. Untuk data berlabel biner, distribusi ini diberikan oleh rasio jumlah sampel dalam faset a berlabel dengan hasil positif terhadap jumlah total, $P_a(y^{1}) = n_a^{(1)}/n_a$, dan rasio jumlah sampel dengan hasil negatif terhadap jumlah total, $P_a(y^{0) = n_a^{(0)}}/n._a$
  - $P_d(y)$ adalah distribusi probabilitas dari label yang diamati untuk faset d. Untuk data berlabel biner, distribusi ini diberikan oleh jumlah sampel dalam segi d berlabel hasil positif terhadap jumlah total, $P_d(y^{1}) = n_d^{(1)}/n_d$, dan rasio jumlah sampel dengan hasil negatif terhadap jumlah total, $P_d(y^{0) = n_d^{(0)}}/n._d$

Model yang dilatih pada data yang bias oleh kesenjangan demografis mungkin belajar dan bahkan memperburuknya. Untuk mengidentifikasi bias dalam data sebelum mengeluarkan sumber daya untuk melatih model di dalamnya, SageMaker Clarify menyediakan metrik bias data yang dapat Anda hitung pada kumpulan data mentah sebelum pelatihan. Semua metrik pra-pelatihan adalah model-agnostik karena tidak bergantung pada output model dan valid untuk model apa pun. Metrik bias pertama memeriksa ketidakseimbangan aspek, tetapi bukan hasil. Ini menentukan sejauh mana jumlah data pelatihan representatif di berbagai aspek, seperti yang diinginkan untuk aplikasi. Metrik bias yang tersisa membandingkan distribusi label hasil dengan berbagai cara untuk aspek a dan d dalam data. Metrik yang berkisar di atas nilai negatif dapat mendeteksi bias negatif. Tabel berikut berisi lembar contekan untuk panduan cepat dan tautan ke metrik bias pra-pelatihan.

Metrik Bias Pra-pelatihan

Metrik bias	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Ketidakseimbangan Kelas (CI)	Mengukur ketidakseimbangan jumlah anggota antara nilai faset yang berbeda.	Mungkinkah ada bias berbasis usia karena tidak memiliki cukup data untuk demografis di luar aspek paruh baya?	Rentang dinormalisasi: [-1, +1] Interpretasi: • Nilai positif menunjukkan aspek A memiliki lebih banyak sampel pelatihan dalam kumpulan data. • Nilai mendekati nol menunjukkan aspek seimbang dalam jumlah sampel pelatihan dalam kumpulan data. • Nilai negatif menunjukkan aspek d memiliki lebih banyak sampel pelatihan dalam kumpulan data.
Perbedaan Proporsi Label (DPL)	Mengukur ketidakseimbangan hasil positif antara nilai segi yang berbeda.	Mungkinkah ada bias berbasis usia dalam prediksi MLkarena pelabelan bias nilai faset dalam data?	Rentang untuk label aspek biner & multikategori yang dinormalisasi: [-1, +1] Rentang untuk label kontinu: (-∞, +∞) Interpretasi:

Metrik bias	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
			• Nilai positif menunjukkan aspek a memiliki proporsi hasil positif yang lebih tinggi. • Nilai mendekati nol menunjukkan proporsi hasil positif yang lebih sama antar aspek. • Nilai negatif menunjukkan aspek d memiliki proporsi hasil positif yang lebih tinggi.
Divergensi Kullback-Leibler (KL)	Mengukur seberapa besar distribusi hasil dari berbagai aspek berbeda berbeda satu sama lain secara entropis.	Seberapa berbeda distribusi untuk hasil aplikasi pinjaman untuk kelompok demografis yang berbeda?	Rentang untuk biner, multikategori, kontinu: [0, +∞) Interpretasi: • Nilai mendekati nol menunjukkan label didistribusikan dengan cara yang sama. • Nilai positif menunjukkan distribusi label menyimpang, semakin positif semakin besar divergensi.

Metrik bias	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Divergensi Jensen-Shannon (JS)	Mengukur seberapa besar distribusi hasil dari berbagai aspek berbeda berbeda satu sama lain secara entropis.	Seberapa berbeda distribusi untuk hasil aplikasi pinjaman untuk kelompok demografis yang berbeda?	Rentang untuk biner, multikategori, kontinu: [0, +∞) Interpretasi: • Nilai mendekati nol menunjukkan label didistribusikan dengan cara yang sama. • Nilai positif menunjukkan distribusi label menyimpang, semakin positif semakin besar divergensi.

Metrik bias	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
L p -norma (LP)	Mengukur perbedaan p-norma antara distribusi demografi s yang berbeda dari hasil yang terkait dengan aspek yang berbeda dalam kumpulan data.	Seberapa berbeda distribusi untuk hasil aplikasi pinjaman untuk demografi yang berbeda?	Rentang untuk biner, multikategori, kontinu: [0, +∞) Interpretasi: • Nilai mendekati nol menunjukkan label didistribusikan dengan cara yang sama. • Nilai positif menunjukkan distribusi label menyimpang, semakin positif semakin besar divergensi.

Metrik bias	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Jarak Variasi Total (TVD)	Mengukur setengah dari perbedaan $_1$ norma L antara distribusi demografi s yang berbeda dari hasil yang terkait dengan aspek yang berbeda dalam kumpulan data.	Seberapa berbeda distribusi untuk hasil aplikasi pinjaman untuk demografi yang berbeda?	Rentang untuk hasil biner, multikategori, dan berkelanjutan: [0, +∞) • Nilai mendekati nol menunjukkan label didistribusikan dengan cara yang sama. • Nilai positif menunjukkan distribusi label menyimpang, semakin positif semakin besar divergensi.

Metrik bias	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Kolmogorov-Smirnov (KS)	Mengukur divergensi maksimum antara hasil dalam distribusi untuk berbagai aspek dalam kumpulan data.	Hasil aplikasi perguruan tinggi mana yang memanifestasikan perbedaan terbesar menurut kelompok demografis?	Rentang nilai KS untuk hasil biner, multikategori, dan kontinu: [0, +1] • Nilai mendekati nol menunjukkan label didistribusikan secara merata antar aspek di semua kategori hasil. • Nilai di dekat satu menunjukkan label untuk satu kategori semuanya dalam satu segi, jadi sangat tidak seimbang. • Nilai intermitten menunjukkan derajat relatif ketidakseimbangan label maksimum.

Metrik bias	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Disparitas Demografis Bersyarat (CDD)	Mengukur perbedaan hasil antara aspek yang berbeda secara keseluruhan, tetapi juga oleh subkelompok.	Apakah beberapa kelompok memiliki proporsi penolakan yang lebih besar untuk hasil penerimaan perguruan tinggi daripada proporsi penerimaan mereka?	Rentang CDD: [-1, +1] • Nilai positif menunjukkan hasil di mana segi d ditolak lebih dari yang diterima. • Mendekati nol menunjukkan tidak ada perbedaan demografis rata-rata. • Nilai negatif menunjukkan hasil di mana aspek a ditolak lebih dari yang diterima.

Untuk informasi tambahan tentang metrik bias, lihat Fairness Measures for Machine Learning in Finance.

Topik

- Ketidakseimbangan Kelas (CI)
- Perbedaan Proporsi Label (DPL)
- Divergensi Kullback-Leibler (KL)
- Divergensi Jensen-Shannon (JS)
- L p -norma (LP)
- Jarak Variasi Total (TVD)
- Kolmogorov-Smirnov (KS)
- Disparitas Demografis Bersyarat (CDD)

Ketidakseimbangan Kelas (CI)

Bias ketidakseimbangan kelas (CI) terjadi ketika nilai faset d memiliki lebih sedikit sampel pelatihan jika dibandingkan dengan aspek lain a dalam kumpulan data. Ini karena model secara istimewa sesuai dengan aspek yang lebih besar dengan mengorbankan aspek yang lebih kecil sehingga dapat menghasilkan kesalahan pelatihan yang lebih tinggi untuk aspek d. Model juga berisiko lebih tinggi untuk menyesuaikan set data yang lebih kecil, yang dapat menyebabkan kesalahan pengujian yang lebih besar untuk segi d. Pertimbangkan contoh di mana model pembelajaran mesin dilatih terutama pada data dari individu paruh baya (aspek a), mungkin kurang akurat ketika membuat prediksi yang melibatkan orang yang lebih muda dan lebih tua (aspek d).

Rumus untuk ukuran ketidakseimbangan segi (dinormalisasi):

$$CI = (n_a - n_d)/(n_a + n_d)$$

Dimana n_a adalah jumlah anggota faset a dan n_d bilangan untuk faset d. Nilainya berkisar pada interval [-1, 1].

- Nilai CI positif menunjukkan aspek A memiliki lebih banyak sampel pelatihan dalam kumpulan data dan nilai 1 menunjukkan data hanya berisi anggota faset a.
- Nilai CI mendekati nol menunjukkan distribusi anggota yang lebih merata antara faset dan nilai nol menunjukkan partisi yang sama sempurna antara faset dan mewakili distribusi sampel yang seimbang dalam data pelatihan.
- Nilai CI negatif menunjukkan aspek d memiliki lebih banyak sampel pelatihan dalam kumpulan data dan nilai -1 menunjukkan data hanya berisi anggota faset d.
- Nilai CI di dekat salah satu nilai ekstrem -1 atau 1 sangat tidak seimbang dan berisiko besar membuat prediksi bias.

Jika ketidakseimbangan aspek yang signifikan ditemukan ada di antara aspek-aspek tersebut, Anda mungkin ingin menyeimbangkan kembali sampel sebelum melanjutkan untuk melatih model di atasnya.

Perbedaan Proporsi Label (DPL)

Perbedaan proporsi label (DPL) membandingkan proporsi hasil yang diamati dengan label positif untuk segi D dengan proporsi hasil yang diamati dengan label positif dari segi a dalam kumpulan data pelatihan. Misalnya, Anda dapat menggunakannya untuk membandingkan proporsi individu paruh baya (aspek a) dan kelompok usia lainnya (aspek d) yang disetujui untuk pinjaman keuangan. Model pembelajaran mesin mencoba meniru keputusan data pelatihan sedekat mungkin. Jadi model

pembelajaran mesin yang dilatih pada dataset dengan DPL tinggi kemungkinan akan mencerminkan ketidakseimbangan yang sama dalam prediksi masa depannya.

Rumus untuk perbedaan proporsi label adalah sebagai berikut:

$$DPL = (q_a - q)_d$$

Di mana:

- $q_a = n_a^{(1)} / n_a$ adalah proporsi faset a yang memiliki nilai label yang diamati 1. Misalnya, proporsi demografis paruh baya yang disetujui untuk pinjaman. Di sini $n_a^{(1)}$ mewakili jumlah anggota faset a yang mendapatkan hasil positif dan n_a adalah jumlah anggota faset a.
- $q_d = n_d^{(1)} / n_d$ adalah proporsi faset d yang memiliki nilai label yang diamati 1. Misalnya, proporsi orang di luar demografi paruh baya yang disetujui untuk pinjaman. Di sini $n_d^{(1)}$ mewakili jumlah anggota faset d yang mendapatkan hasil positif dan n_d adalah jumlah anggota faset d.

Jika DPL cukup dekat dengan 0, maka kita katakan bahwa paritas demografis telah tercapai.

Untuk label faset biner dan multikategori, nilai DPL berkisar pada interval (-1, 1). Untuk label kontinu, kami menetapkan ambang batas untuk menciutkan label ke biner.

- Nilai DPL positif menunjukkan bahwa faset a memiliki proporsi hasil positif yang lebih tinggi jika dibandingkan dengan segi d.
- Nilai DPL mendekati nol menunjukkan proporsi hasil positif yang lebih sama antara aspek dan nilai nol menunjukkan paritas demografis yang sempurna.
- Nilai DPL negatif menunjukkan bahwa facet d memiliki proporsi hasil positif yang lebih tinggi jika dibandingkan dengan faset a.

Apakah DPL berskala tinggi bermasalah atau tidak bervariasi dari satu situasi ke situasi lainnya. Dalam kasus yang bermasalah, DPL berkekuatan tinggi mungkin merupakan sinyal masalah mendasar dalam data. Misalnya, kumpulan data dengan DPL tinggi mungkin mencerminkan bias atau prasangka historis terhadap kelompok demografis berbasis usia yang tidak diinginkan untuk dipelajari oleh model.

Divergensi Kullback-Leibler (KL)

Divergensi Kullback-Leibler (KL) mengukur seberapa besar distribusi label yang diamati dari faset a, $P_a(y)$, menyimpang dari distribusi faset d, $P(y)_d$. Ia juga dikenal sebagai entropi relatif $P_a(y)$

sehubungan dengan $P_d(y)$ dan mengukur jumlah informasi yang hilang saat berpindah dari $P_a(y)$ ke $P_d(y)$.

Rumus untuk divergensi Kullback-Leibler adalah sebagai berikut:

$$KL(P_a \,||\, P_d) = \Sigma_y P_a(y) * \log[P_a(y)/P_d(y)]$$

Ini adalah ekspektasi perbedaan logaritmik antara probabilitas $P_a(y)$ dan $P_d(y)$, di mana ekspektasi ditimbang oleh probabilitas $P_a(y)$. Ini bukan jarak sebenarnya antara distribusi karena asimetris dan tidak memenuhi ketidaksetaraan segitiga. Implementasinya menggunakan logaritma alami, memberikan KL dalam satuan nats. Menggunakan basis logaritmik yang berbeda memberikan hasil proporsional tetapi dalam satuan yang berbeda. Misalnya, menggunakan basis 2 memberikan KL dalam satuan bit.

Misalnya, asumsikan bahwa sekelompok pemohon pinjaman memiliki tingkat persetujuan 30% (aspek d) dan tingkat persetujuan untuk pelamar lain (aspek a) adalah 80%. Rumus Kullback-Leibler memberi Anda divergensi distribusi label faset a dari segi d sebagai berikut:

$$KL = 0{,}8*\ln(0{,}8/0{,}3) + 0{,}2*\ln(0{,}2/0{,}7) = 0{,}53$$

Ada dua istilah dalam rumus di sini karena label adalah biner dalam contoh ini. Ukuran ini dapat diterapkan ke beberapa label selain yang biner. Misalnya, dalam skenario penerimaan perguruan tinggi, asumsikan pelamar dapat diberi salah satu dari tiga label kategori: $y_i = \{y_0, y_1, y_2\}$ = {ditolak, daftar tunggu, diterima}.

Rentang nilai untuk metrik KL untuk hasil biner, multikategori, dan kontinu adalah $[0, +\infty)$.

- Nilai mendekati nol berarti hasilnya didistribusikan dengan cara yang sama untuk berbagai aspek.
- Nilai positif berarti distribusi label menyimpang, semakin positif semakin besar divergensi.

Divergensi Jensen-Shannon (JS)

Divergensi Jensen-Shannon (JS) mengukur seberapa besar distribusi label dari berbagai aspek berbeda berbeda satu sama lain secara entropis. Ini didasarkan pada divergensi Kullback-Leibler, tetapi simetris.

Rumus untuk divergensi Jensen-Shannon adalah sebagai berikut:

$$JS = \tfrac{1}{2} * [KL(P_a \,||\, P) + KL(P_d \,||\, P)]$$

Dimana P = ½ (P_a + P_d), distribusi label rata-rata di seluruh aspek a dan d.

Kisaran nilai JS untuk hasil biner, multikategori, kontinu adalah [0, ln (2)).

- Nilai mendekati nol berarti label didistribusikan dengan cara yang sama.
- Nilai positif berarti distribusi label menyimpang, semakin positif semakin besar divergensi.

Metrik ini menunjukkan apakah ada perbedaan besar di salah satu label di seluruh aspek.

L_p -norma (LP)

$_p$L-norma (LP) mengukur jarak p-norma antara distribusi faset dari label yang diamati dalam kumpulan data pelatihan. Metrik ini non-negatif sehingga tidak dapat mendeteksi bias terbalik.

Rumus untuk $_p$ norma L adalah sebagai berikut:

$$L_p(P_a, P_d) = (\Sigma_y ||P_a - P_d||_p)^{1/p}$$

Dimana jarak p-norma antara titik x dan y didefinisikan sebagai berikut:

$$L_p(x, y) = (|x_1 - y_1|^p + |x - y|^p + \ldots_n + |x_2 - y_2|^p)^{1/p_n}$$

Norma 2 adalah norma Euclidean. Asumsikan Anda memiliki distribusi hasil dengan tiga kategori, misalnya, $y_i = \{y_0, y_1, y_2\}$ = {diterima, daftar tunggu, ditolak} dalam skenario multikategori penerimaan perguruan tinggi. Anda mengambil jumlah kuadrat perbedaan antara jumlah hasil untuk aspek a dan d. Jarak Euclidean yang dihasilkan dihitung sebagai berikut:

$$L_2(P_a, P_d) = [(n_a^{(0)} - n_d^{(0)})^2 + (n_a^{(1)} - n_d^{(1)})^2 + (n_a^{(2)} - n_d^{(2)})^2]^{1/2}$$

Di mana:

- $n_a^{(i)}$ adalah jumlah hasil kategori ith dalam segi a: misalnya $n_a^{(0)}$ adalah jumlah faset a yang diterima.
- $n_d^{(i)}$ adalah jumlah hasil kategori ith dalam segi d: misalnya $n_d^{(2)}$ adalah jumlah penolakan faset d.

  Rentang nilai LP untuk hasil biner, multikategori, dan kontinu adalah [0, √2), di mana:

  - Nilai mendekati nol berarti label didistribusikan dengan cara yang sama.
  - Nilai positif berarti distribusi label menyimpang, semakin positif semakin besar divergensi.

Jarak Variasi Total (TVD)

Metrik bias data jarak variasi total (TVD) adalah setengah dari norma L_1. TVD adalah perbedaan terbesar yang mungkin antara distribusi probabilitas untuk hasil label dari segi a dan d. $_1$Norma L adalah jarak Hamming, metrik yang digunakan membandingkan dua string data biner dengan menentukan jumlah minimum substitusi yang diperlukan untuk mengubah satu string ke string lainnya. Jika string harus menjadi salinan satu sama lain, itu menentukan jumlah kesalahan yang terjadi saat menyalin. Dalam konteks deteksi bias, TVD mengukur berapa banyak hasil dalam segi a yang harus diubah agar sesuai dengan hasil dalam segi d.

Rumus untuk Jarak variasi Total adalah sebagai berikut:

$$TVD = ½ * L_1 (P_a, P)_d$$

Misalnya, asumsikan Anda memiliki distribusi hasil dengan tiga kategori, $y_i = \{y_0, y_1, y_2\}$ = {diterima, daftar tunggu, ditolak}, dalam skenario multikategori penerimaan perguruan tinggi. Anda mengambil perbedaan antara jumlah aspek a dan d untuk setiap hasil untuk menghitung TVD. Hasilnya adalah sebagai berikut:

$$L_1 (P_a, P_d) = |n_a^{(0)} - n_d^{(0)}| + |n_a^{(1)} - n_d^{(1)}| + |n_a^{(2)} - n_d^{(2)}|$$

Di mana:

- $n_a^{(i)}$ adalah jumlah hasil kategori ith dalam segi a: misalnya $n_a^{(0)}$ adalah jumlah faset a yang diterima.
- $n_d^{(i)}$ adalah jumlah hasil kategori ith dalam segi d: misalnya $n_d^{(2)}$ adalah jumlah penolakan faset d.

  Rentang nilai TVD untuk hasil biner, multikategori, dan kontinu adalah [0, 1), di mana:

  - Nilai mendekati nol berarti label didistribusikan dengan cara yang sama.
  - Nilai positif berarti distribusi label menyimpang, semakin positif semakin besar divergensi.

Kolmogorov-Smirnov (KS)

Metrik bias Kolmogorov-Smirnov (KS) sama dengan divergensi maksimum antara label dalam distribusi untuk aspek a dan d dari kumpulan data. Uji KS dua sampel yang dilaksanakan oleh SageMaker Clarify melengkapi ukuran ketidakseimbangan label lainnya dengan menemukan label yang paling tidak seimbang.

Rumus untuk metrik Kolmogorov-Smirnov adalah sebagai berikut:

$$KS = maks (|P_a (y) - P_d (y)|)$$

Misalnya, asumsikan sekelompok pelamar (aspek a) ke perguruan tinggi ditolak, daftar tunggu, atau diterima masing-masing 40%, 40%, 20% dan bahwa tarif ini untuk pelamar lain (aspek d) adalah 20%, 10%, 70%. Maka nilai metrik bias Kolmogorov-Smirnov adalah sebagai berikut:

KS = maks (|0,4-0,2 |, |0,4-0,1|, |0,2-0,7 |) = 0,5

Ini memberi tahu kita perbedaan maksimum antara distribusi faset adalah 0,5 dan terjadi pada tingkat penerimaan. Ada tiga istilah dalam persamaan karena label adalah multikelas kardinalitas tiga.

Rentang nilai LP untuk hasil biner, multikategori, dan kontinu adalah [0, +1], di mana:

- Nilai mendekati nol menunjukkan label didistribusikan secara merata antar aspek di semua kategori hasil. Misalnya, kedua aspek yang mengajukan pinjaman mendapat 50% dari penerimaan dan 50% dari penolakan.
- Nilai di dekat satu menunjukkan label untuk satu hasil semuanya dalam satu segi. Misalnya, facet a mendapat 100% dari penerimaan dan facet d tidak punya.
- Nilai intermiten menunjukkan derajat relatif ketidakseimbangan label maksimum.

Disparitas Demografis Bersyarat (CDD)

Metrik disparitas demografis (DD) menentukan apakah suatu aspek memiliki proporsi yang lebih besar dari hasil yang ditolak dalam kumpulan data daripada hasil yang diterima. Dalam kasus biner di mana ada dua aspek, pria dan wanita misalnya, yang merupakan kumpulan data, yang tidak disukai diberi label segi d dan yang disukai diberi label faset a. Misalnya, dalam kasus penerimaan perguruan tinggi, jika pelamar perempuan terdiri dari 46% dari pelamar yang ditolak dan hanya terdiri dari 32% dari pelamar yang diterima, kami mengatakan bahwa ada perbedaan demografis karena tingkat di mana perempuan ditolak melebihi tingkat di mana mereka diterima. Pelamar perempuan diberi label facet d dalam kasus ini. Jika pelamar laki-laki terdiri dari 54% dari pelamar yang ditolak dan 68% dari pelamar yang diterima, maka tidak ada perbedaan demografis untuk aspek ini karena tingkat penolakan kurang dari tingkat penerimaan. Pelamar pria diberi label facet a dalam kasus ini.

Rumus untuk disparitas demografis untuk aspek yang kurang disukai d adalah sebagai berikut:

$$DD_d = n_d^{(0)} / n^{(0)} - n_d^{(1)} / n^{(1)} = P_d^R (y^0) - P_d^A (y\ 1)$$

Di mana:

- $n^{(0)} = n_a^{(0)} + n_d^{(0)}$ adalah jumlah total hasil yang ditolak dalam kumpulan data untuk aspek yang disukai a dan aspek yang kurang beruntung d.
- $n^{(1)} = n_a^{(1)} + n_d^{(1)}$ adalah jumlah total hasil yang diterima dalam kumpulan data untuk aspek yang disukai a dan aspek yang kurang beruntung d.
- $P_d^R(y^0)$ adalah proporsi hasil yang ditolak (dengan nilai 0) dalam segi d.
- $P_d^A(y^1)$ adalah proporsi hasil yang diterima (nilai 1) dalam segi d.

Untuk contoh penerimaan perguruan tinggi, perbedaan demografis untuk wanita adalah $DD_d = 0{,}46 - 0{,}32 = 0{,}14$. Untuk pria $DD_a = 0{,}54 - 0{,}68 = -0{,}14$.

Metrik disparitas demografis bersyarat (CDD) yang mengkondisikan DD pada atribut yang menentukan strata subkelompok pada kumpulan data diperlukan untuk mengesampingkan paradoks Simpson. Pengelompokan kembali dapat memberikan wawasan tentang penyebab kesenjangan demografis yang jelas untuk aspek yang kurang disukai. Kasus klasik muncul dalam kasus penerimaan Berkeley di mana pria diterima pada tingkat yang lebih tinggi secara keseluruhan daripada wanita. Statistik untuk kasus ini digunakan dalam contoh perhitungan DD. Namun, ketika subkelompok departemen diperiksa, wanita terbukti memiliki tingkat penerimaan yang lebih tinggi daripada pria ketika dikondisikan oleh departemen. Penjelasannya adalah bahwa wanita telah mendaftar ke departemen dengan tingkat penerimaan yang lebih rendah daripada pria. Meneliti tingkat penerimaan subkelompok mengungkapkan bahwa wanita sebenarnya diterima pada tingkat yang lebih tinggi daripada pria untuk departemen dengan tingkat penerimaan yang lebih rendah.

Metrik CDD memberikan ukuran tunggal untuk semua perbedaan yang ditemukan dalam subkelompok yang ditentukan oleh atribut kumpulan data dengan merata-ratakannya. Ini didefinisikan sebagai rata-rata tertimbang disparitas demografis (DD_i) untuk masing-masing subkelompok, dengan setiap disparitas subkelompok tertimbang secara proporsional dengan jumlah pengamatan yang terkandung. Rumus untuk disparitas demografis bersyarat adalah sebagai berikut:

$$CDD = (1/n) * \Sigma_i n\, DD_i * i$$

Di mana:

- ${}_i n_i = n$ adalah jumlah total pengamatan dan n_i adalah jumlah pengamatan untuk setiap subkelompok.
- $DD_i = n_i^{(0)}/n^{(0)} - n_i^{(1)}/n^{(1)} = P_i^R(y^0) - P_i^A(y^1)$ adalah disparitas demografis untuk subkelompok ith.

Perbedaan demografis untuk subkelompok (DD_i) adalah perbedaan antara proporsi hasil yang ditolak dan proporsi hasil yang diterima untuk setiap subkelompok.

Kisaran nilai DD untuk hasil biner untuk kumpulan data lengkap DD_d atau untuk subkelompok terkondisionalisasi DD_i adalah [-1, +1].

- +1: ketika tidak ada penolakan dalam segi a atau subkelompok dan tidak ada penerimaan di segi d atau subkelompok
- Nilai positif menunjukkan ada perbedaan demografis karena aspek d atau subkelompok memiliki proporsi yang lebih besar dari hasil yang ditolak dalam kumpulan data daripada hasil yang diterima. Semakin tinggi nilainya, semakin sedikit faset dan semakin besar perbedaannya.
- Nilai negatif menunjukkan tidak ada perbedaan demografis karena aspek d atau subkelompok memiliki proporsi yang lebih besar dari hasil yang diterima dalam kumpulan data daripada hasil yang ditolak. Semakin rendah nilainya, semakin disukai fasetnya.
- -1: ketika tidak ada penolakan dalam segi d atau subkelompok dan tidak ada penerimaan dalam segi a atau subkelompok

Jika Anda tidak mengkondisikan apa pun maka CDD adalah nol jika dan hanya jika DPL adalah nol.

Metrik ini berguna untuk mengeksplorasi konsep diskriminasi langsung dan tidak langsung dan pembenaran obyektif dalam hukum dan yurisprudensi non-diskriminasi UE dan Inggris. Untuk informasi tambahan, lihat Mengapa Keadilan Tidak Dapat Diotomatisasi. Paper ini juga berisi data dan analisis yang relevan dari kasus penerimaan Berkeley yang menunjukkan bagaimana kondisionalisasi pada subkelompok tingkat penerimaan departemen menggambarkan paradoks Simpson.

Hasilkan Laporan untuk Bias dalam Data Pra-pelatihan di Studio SageMaker

SageMaker Clarify terintegrasi dengan Amazon SageMaker Data Wrangler, yang dapat membantu Anda mengidentifikasi bias selama persiapan data tanpa harus menulis kode Anda sendiri. Data Wrangler menyediakan end-to-end solusi untuk mengimpor, menyiapkan, mengubah, membuat fitur, dan menganalisis data dengan Amazon Studio. SageMaker Untuk gambaran umum tentang alur kerja persiapan data Data Wrangler, lihat. Siapkan Data ML dengan Amazon SageMaker Data Wrangler

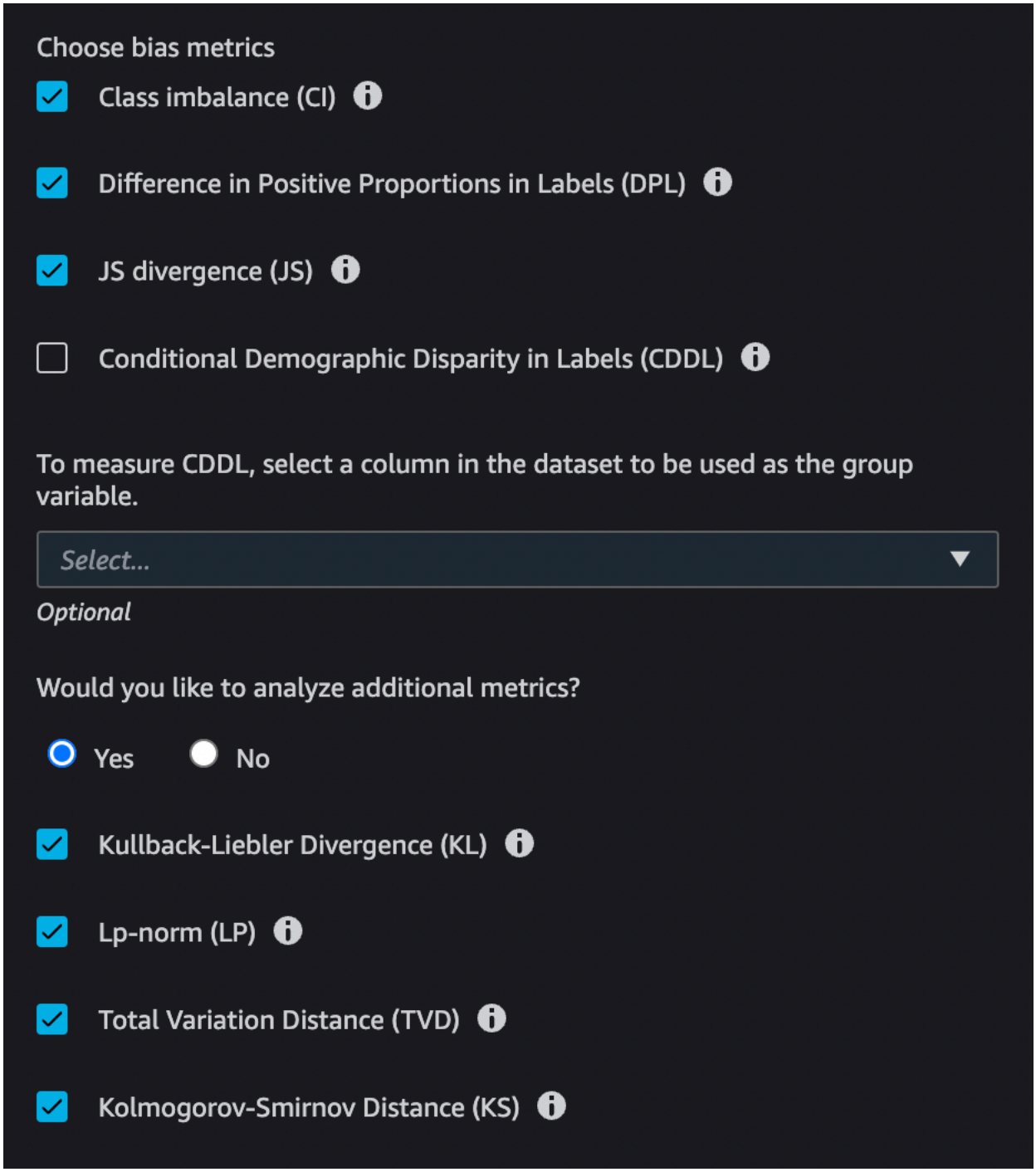
Anda menentukan atribut yang diminati, seperti jenis kelamin atau usia, dan SageMaker Clarify menjalankan serangkaian algoritme untuk mendeteksi adanya bias pada atribut tersebut. Setelah

algoritme berjalan, SageMaker Clarify memberikan laporan visual dengan deskripsi sumber dan tingkat keparahan bias yang mungkin sehingga Anda dapat merencanakan langkah-langkah untuk mengurangi. Misalnya, dalam kumpulan data keuangan yang berisi beberapa contoh pinjaman bisnis untuk satu kelompok umur dibandingkan dengan yang lain, SageMaker tandai ketidakseimbangan sehingga Anda dapat menghindari model yang tidak menyukai kelompok usia tersebut.

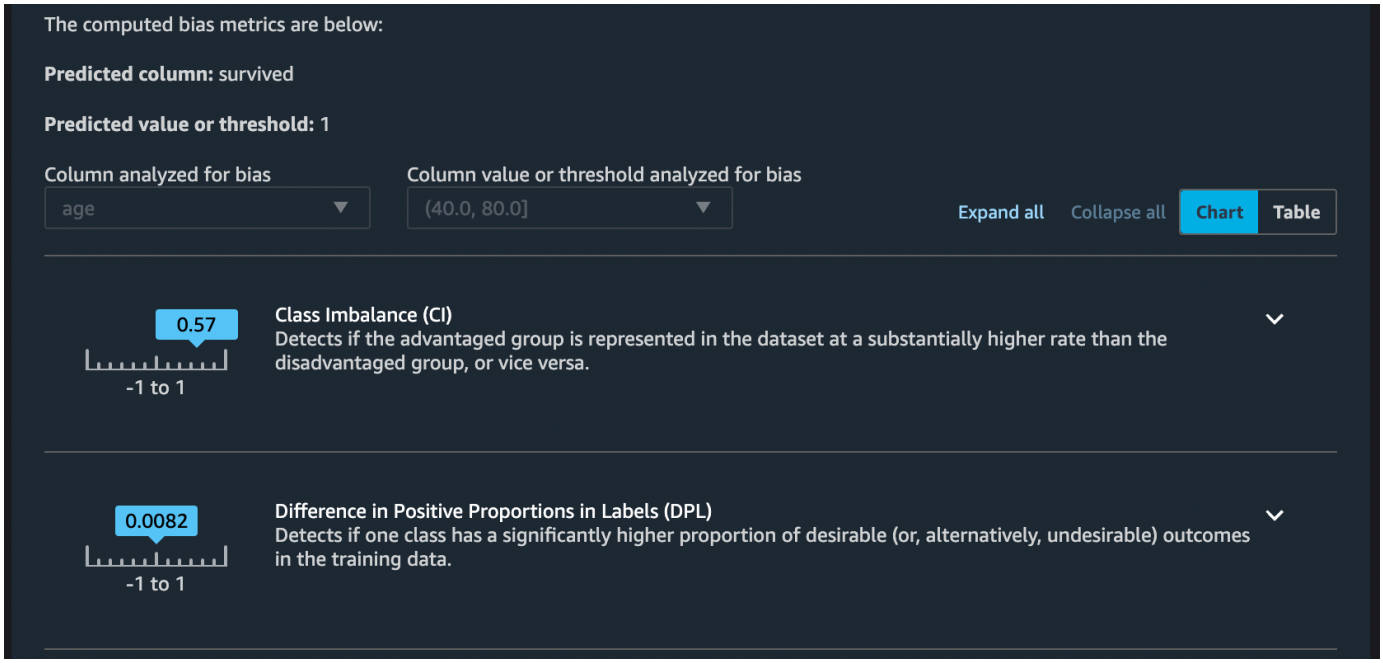
Untuk menganalisis dan melaporkan bias data

Untuk memulai dengan Data Wrangler, lihat. Memulai dengan Data Wrangler

1. Di Amazon SageMaker Studio Classic, dari menu Home
   ()
   di panel kiri, navigasikan ke node Data, lalu pilih Data Wrangler. Ini membuka halaman landing Data Wrangler di Studio Classic.
2. Pilih tombol + Impor data untuk membuat alur baru.
3. Di halaman alur, dari tab Impor, pilih Amazon S3, arahkan ke bucket Amazon S3, temukan kumpulan data, lalu pilih Impor.
4. Setelah Anda mengimpor data Anda, pada grafik aliran di tab Aliran data, pilih tanda + di sebelah kanan simpul Tipe data.
5. Pilih Tambahkan analisis.
6. Pada halaman Buat Analisis, pilih Laporan Bias untuk jenis Analisis.
7. Konfigurasikan laporan bias dengan memberikan Nama laporan, kolom untuk memprediksi dan apakah itu nilai atau ambang batas, kolom untuk menganalisis bias (segi) dan apakah itu nilai atau ambang batas.
8. Lanjutkan mengonfigurasi laporan bias dengan memilih metrik bias.



9. Pilih Periksa bias untuk menghasilkan dan melihat laporan bias. Gulir ke bawah untuk melihat semua laporan.



10. Pilih tanda sisipan di sebelah kanan setiap deskripsi metrik bias untuk melihat dokumentasi yang dapat membantu Anda menafsirkan signifikansi nilai metrik.
11. Untuk melihat ringkasan tabel dari nilai metrik bias, pilih sakelar Tabel. Untuk menyimpan laporan, pilih Simpan di sudut kanan bawah halaman. Anda dapat melihat laporan pada grafik aliran di tab Aliran data. Klik dua kali pada laporan untuk membukanya.

Mendeteksi Data Pasca-pelatihan dan Bias Model

Analisis bias pasca-pelatihan dapat membantu mengungkapkan bias yang mungkin berasal dari bias dalam data, atau dari bias yang diperkenalkan oleh algoritma klasifikasi dan prediksi. Analisis ini mempertimbangkan data, termasuk label, dan prediksi model. Anda menilai kinerja dengan menganalisis label yang diprediksi atau dengan membandingkan prediksi dengan nilai target yang diamati dalam data sehubungan dengan kelompok dengan atribut yang berbeda. Ada pengertian keadilan yang berbeda, masing-masing membutuhkan metrik bias yang berbeda untuk diukur.

Ada konsep hukum keadilan yang mungkin tidak mudah ditangkap karena sulit dideteksi. Misalnya, konsep AS tentang dampak berbeda yang terjadi ketika suatu kelompok, yang disebut sebagai aspek yang kurang disukai d, mengalami efek buruk bahkan ketika pendekatan yang diambil tampaknya adil. Jenis bias ini mungkin bukan karena model pembelajaran mesin, tetapi mungkin masih dapat dideteksi dengan analisis bias pasca-pelatihan.

Amazon SageMaker Clarify mencoba memastikan penggunaan terminologi yang konsisten. Untuk daftar istilah dan definisinya, lihatAmazon SageMaker Klarifikasi Persyaratan untuk Bias dan Keadilan.

Untuk informasi tambahan tentang metrik bias pasca-pelatihan, lihat Pelajari Cara Amazon SageMaker Memperjelas Membantu Mendeteksi Pengukuran Bias dan Keadilan untuk Machine Learning in Finance. .

Ukur Data Pasca-pelatihan dan Bias Model

Amazon SageMaker Clarify menyediakan sebelas data pasca-pelatihan dan metrik bias model untuk membantu mengukur berbagai konsepsi keadilan. Konsep-konsep ini tidak dapat dipenuhi secara bersamaan dan seleksi tergantung pada spesifik kasus yang melibatkan bias potensial yang dianalisis. Sebagian besar metrik ini adalah kombinasi dari angka-angka yang diambil dari matriks kebingungan klasifikasi biner untuk kelompok demografis yang berbeda. Karena keadilan dan bias dapat didefinisikan oleh berbagai metrik, penilaian manusia diperlukan untuk memahami dan memilih metrik mana yang relevan dengan kasus penggunaan individu, dan pelanggan harus berkonsultasi dengan pemangku kepentingan yang tepat untuk menentukan ukuran keadilan yang tepat untuk aplikasi mereka.

Kami menggunakan notasi berikut untuk membahas metrik bias. Model konseptual yang dijelaskan di sini adalah untuk klasifikasi biner, di mana peristiwa diberi label hanya memiliki dua kemungkinan hasil dalam ruang sampelnya, disebut sebagai positif (dengan nilai 1) dan negatif (dengan nilai 0). Kerangka kerja ini biasanya dapat diperluas ke klasifikasi multikategori secara langsung atau untuk kasus-kasus yang melibatkan hasil bernilai berkelanjutan bila diperlukan. Dalam kasus klasifikasi biner, label positif dan negatif ditetapkan ke hasil yang dicatat dalam kumpulan data mentah untuk aspek yang disukai a dan untuk aspek yang tidak disukai d. Label y ini disebut sebagai label yang diamati untuk membedakannya dari label yang diprediksi y' yang ditetapkan oleh model pembelajaran mesin selama tahap pelatihan atau kesimpulan dari siklus hidup ML. Label ini digunakan untuk menentukan distribusi probabilitas $P_a(y)$ dan $P_d(y)$ untuk hasil faset masing-masing.

- label:
  - y mewakili n label yang diamati untuk hasil peristiwa dalam kumpulan data pelatihan.
  - y' mewakili label yang diprediksi untuk n label yang diamati dalam kumpulan data oleh model terlatih.
- hasil:
  - Hasil positif (dengan nilai 1) untuk sampel, seperti penerimaan aplikasi.

    - $n^{(1)}$ adalah jumlah label yang diamati untuk hasil positif (penerimaan).
    - $n'^{(1)}$ adalah jumlah label yang diprediksi untuk hasil positif (penerimaan).
  - Hasil negatif (dengan nilai 0) untuk sampel, seperti penolakan aplikasi.
    - $n^{(0)}$ adalah jumlah label yang diamati untuk hasil negatif (penolakan).
    - $n'^{(0)}$ adalah jumlah label yang diprediksi untuk hasil negatif (penolakan).
- nilai segi:
  - facet a - Nilai fitur yang mendefinisikan demografis yang disukai bias.
    - n_a adalah jumlah label yang diamati untuk nilai faset yang disukai: $n_a = n_a^{(1)} + n_a^{(0)}$ jumlah label yang diamati positif dan negatif untuk aspek nilai a.
    - n'_a adalah jumlah label yang diprediksi untuk nilai faset yang disukai: $n'_a = n'^{(1)}_a + n'^{(0)}_a$ jumlah label hasil prediksi positif dan negatif untuk nilai faset a. Perhatikan bahwa $n'_a = n_a$.
  - facet d — Nilai fitur yang mendefinisikan demografis yang bias tidak disukai.
    - n_d adalah jumlah label yang diamati untuk nilai faset yang tidak disukai: $n_d = n_d^{(1)} + n_d^{(0)}$ jumlah label yang diamati positif dan negatif untuk nilai faset d.
    - n'_d adalah jumlah label yang diprediksi untuk nilai faset yang tidak disukai: $n'_d = n'^{(1)}_d + n'^{(0)}_d$ jumlah label prediksi positif dan negatif untuk nilai faset d. Perhatikan bahwa $n'_d = n_d$.
- distribusi probabilitas untuk hasil dari hasil data facet berlabel:
  - $P_a(y)$ adalah distribusi probabilitas dari label yang diamati untuk faset a. Untuk data berlabel biner, distribusi ini diberikan oleh rasio jumlah sampel dalam faset a berlabel dengan hasil positif terhadap jumlah total, $P_a(y^1) = n_a^{(1)}/n_a$, dan rasio jumlah sampel dengan hasil negatif terhadap jumlah total, $P_a(y^{0) = n_a^{(0)}}/n._a$
  - $P_d(y)$ adalah distribusi probabilitas dari label yang diamati untuk faset d. Untuk data berlabel biner, distribusi ini diberikan oleh jumlah sampel dalam segi d berlabel hasil positif terhadap jumlah total, $P_d(y^1) = n_d^{(1)}/n_d$, dan rasio jumlah sampel dengan hasil negatif terhadap jumlah total, $P_d(y^{0) = n_d^{(0)}}/n._d$

Tabel berikut berisi lembar contekan untuk panduan cepat dan tautan ke metrik bias pasca-pelatihan.

Metrik bias pasca-pelatihan

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Perbedaan Proporsi Positif pada Label Prediksi (DPPL)	Mengukur perbedaan proporsi prediksi positif antara aspek yang disukai a dan aspek yang tidak disukai d.	Apakah ada ketidakseimbangan antar kelompok demografis dalam hasil positif yang diprediksi yang mungkin mengindikasikan bias?	Rentang untuk label aspek biner & multikategori yang dinormalisasi: `[-1, +1]` Rentang untuk label kontinu: (-∞, +∞) Interpretasi: • Nilai positif menunjukkan bahwa aspek yang disukai a memiliki proporsi hasil positif yang diprediksi lebih tinggi. • Nilai mendekati nol menunjukkan proporsi yang lebih sama dari hasil positif yang diprediksi antar aspek. • Nilai negatif menunjukkan aspek yang tidak disukai d memiliki proporsi hasil positif yang diprediksi lebih tinggi.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Dampak Berbeda (DI)	Mengukur rasio proporsi label yang diprediksi untuk aspek yang disukai a dan aspek yang tidak disukai d.	Apakah ada ketidakseimbangan antar kelompok demografis dalam hasil positif yang diprediksi yang mungkin mengindikasikan bias?	Rentang untuk biner dinormalisasi, aspek multikategori, dan label kontinu: [0, ∞) Interpretasi: • Nilai kurang dari 1 menunjukkan aspek yang disukai a memiliki proporsi hasil positif yang diprediksi lebih tinggi. • Nilai 1 menunjukkan bahwa kita memiliki paritas demografis. • Nilai lebih besar dari 1 menunjukkan aspek yang tidak disukai d memiliki proporsi hasil positif yang diprediksi lebih tinggi.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Disparitas Demografis Bersyarat dalam Label yang Diprediksi (CDDPL)	Mengukur perbedaan label yang diprediksi antara aspek secara keseluruhan, tetapi juga oleh subkelompok.	Apakah beberapa kelompok demografis memiliki proporsi penolakan yang lebih besar untuk hasil aplikasi pinjaman daripada proporsi penerimaan mereka?	Kisaran nilai CDDPL untuk hasil biner, multikategori, dan berkelanjutan: `[-1, +1]` • Nilai positif menunjukkan hasil di mana segi d ditolak lebih dari yang diterima. • Mendekati nol menunjukkan tidak ada perbedaan demografis rata-rata. • Nilai negatif menunjukkan hasil di mana aspek a ditolak lebih dari yang diterima.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Fliptest Kontrafaktual (FT)	Memeriksa setiap anggota faset d dan menilai apakah anggota yang serupa dari segi a memiliki prediksi model yang berbeda.	Apakah satu kelompok demografi usia tertentu cocok dengan semua fitur dengan kelompok usia yang berbeda, namun dibayar lebih rata-rata?	Rentang untuk label facet biner dan multikategori adalah. `[-1, +1]` • Nilai positif terjadi ketika jumlah keputusan terlentang kontrafaktual yang tidak menguntungkan untuk aspek yang tidak disukai d melebihi yang menguntungkan. • Nilai mendekati nol terjadi ketika jumlah keputusan fliptest kontrafaktual yang tidak menguntungkan dan menguntungkan seimbang. • Nilai negatif terjadi ketika jumlah keputusan terlentang kontrafaktual yang tidak menguntungkan untuk aspek yang tidak disukai d kurang dari yang menguntungkan.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Perbedaan Akurasi (AD)	Mengukur perbedaan antara akurasi prediksi untuk aspek yang disukai dan yang tidak disukai.	Apakah model memprediksi label secara akurat untuk aplikasi di semua kelompok demografis?	Rentang untuk label facet biner dan multikategori adalah. `[-1, +1]` • Nilai positif menunjukkan bahwa faset d lebih menderita dari beberapa kombinasi positif palsu (kesalahan Tipe I) atau negatif palsu (kesalahan Tipe II). Ini berarti ada bias potensial terhadap aspek yang tidak disukai d. • Nilai mendekati nol terjadi ketika akurasi prediksi untuk faset a mirip dengan faset d. • Nilai negatif menunjukkan bahwa facet a lebih menderita dari beberapa kombinasi positif palsu (kesalahan Tipe I) atau negatif palsu (kesalahan

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
			Tipe II). Ini berarti bias terhadap aspek yang disukaia.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Ingat Perbedaan (RD)	Membandingkan penarikan kembali model untuk aspek yang disukai dan tidak disukai.	Apakah ada bias berbasis usia dalam pinjaman karena model yang memiliki daya ingat yang lebih tinggi untuk satu kelompok usia dibandingkan dengan yang lain?	Rentang untuk klasifikasi biner dan multikategori:. `[-1, +1]` • Nilai positif menunjukkan bahwa model menemukan lebih banyak positif sejati untuk segi a dan bias terhadap aspek yang tidak disukai d. • Nilai mendekati nol menunjukkan bahwa model menemukan jumlah positif sejati yang sama di kedua aspek dan tidak bias. • Nilai negatif menunjukkan bahwa model menemukan lebih banyak positif sejati untuk segi d dan bias terhadap aspek yang disukai a.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Perbedaan dalam Penerimaan Bersyarat (DCAcc)	Membandingkan label yang diamati dengan label yang diprediksi oleh model. Menilai apakah ini sama di seluruh aspek untuk hasil positif yang diprediksi (penerimaan).	Ketika membandingkan satu kelompok usia dengan yang lain, apakah pinjaman diterima lebih sering, atau lebih jarang dari yang diperkirakan (berdasarkan kualifikasi)?	Rentang untuk biner, aspek multikategori, dan label kontinu: (-∞, +∞). • Nilai positif menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari aspek yang tidak disukai d. • Nilai mendekati nol menunjukkan bahwa pelamar yang memenuhi syarat dari kedua aspek diterima dengan cara yang sama. • Nilai negatif menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari aspek yang disukai a.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Perbedaan Tingkat Penerimaan (DAR)	Mengukur perbedaan rasio hasil positif yang diamati (TP) dengan positif yang diprediksi (TP + FP) antara aspek yang disukai dan yang tidak disukai.	Apakah model memiliki presisi yang sama ketika memprediksi penerimaan pinjaman untuk pelamar yang memenuhi syarat di semua kelompok umur?	Rentang untuk biner, aspek multikategori, dan label kontinu adalah `[-1, +1]` • Nilai positif menunjukkan kemungkinan bias terhadap segi d yang disebabkan oleh terjadinya positif palsu yang relatif lebih banyak pada aspek yang tidak disukai d. • Nilai mendekati nol menunjukkan label yang diamati untuk hasil positif (penerimaan) diprediksi dengan presisi yang sama untuk kedua aspek oleh model. • Nilai negatif menunjukkan kemungkinan bias terhadap faset a yang disebabkan oleh terjadinya positif palsu yang relatif lebih banyak

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
			pada aspek yang disukai a.
Perbedaan spesifisitas (SD)	Membandingkan kekhususan model antara aspek yang disukai dan yang tidak disukai.	Apakah ada bias berbasis usia dalam pinjaman karena model memprediksi spesifisitas yang lebih tinggi untuk satu kelompok umur dibandingkan dengan yang lain?	Rentang untuk klasifikasi biner dan multikategori:. `[-1, +1]` • Nilai positif menunjukkan bahwa model menemukan lebih sedikit positif palsu untuk segi d dan bias terhadap aspek yang tidak disukai d. • Nilai mendekati nol menunjukkan bahwa model menemukan jumlah positif palsu yang sama di kedua sisi dan tidak bias. • Nilai negatif menunjukkan bahwa model menemukan lebih sedikit positif palsu untuk aspek a dan bias terhadap aspek yang disukai a.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Perbedaan Penolakan Bersyarat (DCR)	Membandingkan label yang diamati dengan label yang diprediksi oleh model dan menilai apakah ini sama di seluruh aspek untuk hasil negatif (penolakan).	Apakah ada lebih banyak atau lebih sedikit penolakan untuk aplikasi pinjaman daripada yang diperkirakan untuk satu kelompok umur dibandingkan dengan yang lain berdasarkan kualifikasi?	Rentang untuk biner, aspek multikategori, dan label kontinu: (-∞, +∞). • Nilai positif menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari aspek yang tidak disukai d. • Nilai mendekati nol menunjukkan bahwa pelamar yang memenuhi syarat dari kedua aspek ditolak dengan cara yang sama. • Nilai negatif menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari aspek yang disukai a.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Perbedaan Tingkat Penolakan (DRR)	Mengukur perbedaan rasio hasil negatif yang diamati (TN) dengan negatif yang diprediksi (TN+FN) antara aspek yang tidak disukai dan disukai.	Apakah model memiliki presisi yang sama ketika memprediksi penolakan pinjaman untuk pelamar yang tidak memenuhi syarat di semua kelompok umur?	Rentang untuk biner, aspek multikategori, dan label kontinu adalah. `[-1, +1]` • Nilai positif menunjukkan kemungkinan bias yang disebabkan oleh terjadinya negatif yang relatif lebih salah dalam aspek yang disukaia. • Nilai mendekati nol menunjukkan bahwa hasil negatif (penolakan) diprediksi dengan presisi yang sama untuk kedua sisi. • Nilai negatif menunjukkan kemungkinan bias yang disebabkan oleh terjadinya negatif yang relatif lebih salah dalam aspek yang tidak disukai d.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Kesetaraan Perawatan (TE)	Mengukur perbedaan rasio positif palsu dengan negatif palsu antara aspek yang disukai dan yang tidak disukai.	Dalam aplikasi pinjaman, apakah rasio relatif positif palsu terhadap negatif palsu sama di semua demografi usia?	Rentang untuk label faset biner dan multikategori: (-∞, +∞). • Nilai positif terjadi ketika rasio positif palsu terhadap negatif palsu untuk faset a lebih besar daripada untuk faset d. • Nilai mendekati nol terjadi ketika rasio positif palsu terhadap negatif palsu untuk faset a mirip dengan rasio untuk faset d. • Nilai negatif terjadi ketika rasio positif palsu terhadap negatif palsu untuk faset a kurang dari untuk faset d.

Metrik bias pasca-pelatihan	Deskripsi	Contoh pertanyaan	Menafsirkan nilai metrik
Entropi umum (GE)	Mengukur ketidaksetaraan manfaat yang b diberikan untuk setiap input oleh prediksi model.	Dari dua model kandidat untuk klasifikasi aplikasi pinjaman, apakah yang satu mengarah pada distribusi hasil yang diinginkan yang lebih tidak merata daripada yang lain?	Rentang untuk label biner dan multikategori: (0, 0,5). GE tidak terdefinisi ketika model hanya memprediksi negatif palsu. • Nilai nol terjadi ketika semua prediksi benar atau semua prediksi positif palsu. • Nilai positif menunjukkan ketidaksetaraan dalam manfaat; 0,5 sesuai dengan ketidaksetaraan terbesar.

Untuk informasi tambahan tentang metrik bias pasca-pelatihan, lihat A Family of Fairness Measures for Machine Learning in Finance.

Topik

- Perbedaan Proporsi Positif pada Label Prediksi (DPPL)
- Dampak Berbeda (DI)
- Perbedaan dalam Penerimaan Bersyarat (DCAcc)
- Perbedaan Penolakan Bersyarat (DCR)
- Perbedaan spesifisitas (SD)
- Ingat Perbedaan (RD)
- Perbedaan Tingkat Penerimaan (DAR)

- Perbedaan Tingkat Penolakan (DRR)
- Perbedaan Akurasi (AD)
- Kesetaraan Perawatan (TE)
- Disparitas Demografis Bersyarat dalam Label yang Diprediksi (CDDPL)
- Fliptest Kontrafaktual (FT)
- Entropi umum (GE)

Perbedaan Proporsi Positif pada Label Prediksi (DPPL)

Perbedaan proporsi positif dalam metrik label prediksi (DPPL) menentukan apakah model memprediksi hasil secara berbeda untuk setiap aspek. Ini didefinisikan sebagai perbedaan antara proporsi prediksi positif (y' = 1) untuk segi a dan proporsi prediksi positif (y' = 1) untuk segi d. Misalnya, jika prediksi model memberikan pinjaman kepada 60% dari kelompok paruh baya (aspek a) dan 50% kelompok usia lainnya (segi d), itu mungkin bias terhadap aspek d. Dalam contoh ini, Anda harus menentukan apakah perbedaan 10% material untuk kasus bias.

Perbandingan perbedaan proporsi label (DPL), ukuran bias pra-pelatihan, dengan DPPL, ukuran bias pasca-pelatihan, menilai apakah bias dalam proporsi positif yang awalnya ada dalam dataset berubah setelah pelatihan. Jika DPPL lebih besar dari DPL, maka bias dalam proporsi positif meningkat setelah pelatihan. Jika DPPL lebih kecil dari DPL, model tidak meningkatkan bias dalam proporsi positif setelah pelatihan. Membandingkan DPL terhadap DPPL tidak menjamin bahwa model mengurangi bias di sepanjang semua dimensi. Misalnya, model mungkin masih bias saat mempertimbangkan metrik lain seperti Fliptest Kontrafaktual (FT) atau. Perbedaan Akurasi (AD) Untuk informasi selengkapnya tentang deteksi bias, lihat posting blog Pelajari cara Amazon SageMaker Clarify membantu mendeteksi bias. Lihat Perbedaan Proporsi Label (DPL) untuk informasi lebih lanjut tentang DPL.

Rumus untuk DPPL adalah:

$$DPPL = q' - q'_{a\,d}$$

Di mana:

- $q'_a = n'_a{}^{(1)} / n_a$ adalah proporsi prediksi dari segi a yang mendapatkan hasil positif dari nilai 1. Dalam contoh kami, proporsi aspek paruh baya diprediksi akan diberikan pinjaman. Di sini $n'_a{}^{(1)}$ mewakili jumlah anggota faset a yang mendapatkan hasil prediksi positif dari nilai 1 dan n_a adalah jumlah anggota faset a.

- $q'_d = n'^{(1)}_d / n_d$ adalah proporsi prediksi dari segi d yang mendapatkan hasil positif dari nilai 1. Dalam contoh kita, aspek orang tua dan muda diprediksi akan diberikan pinjaman. Di sini $n'^{(1)}_d$ mewakili jumlah anggota segi d yang mendapatkan hasil prediksi positif dan n_d adalah jumlah anggota segi d.

Jika DPPL cukup dekat dengan 0, itu berarti paritas demografis pasca-pelatihan telah tercapai.

Untuk label faset biner dan multikategori, nilai DPL yang dinormalisasi berkisar pada interval [-1, 1]. Untuk label kontinu, nilainya bervariasi selama interval (-∞, +∞).

- Nilai DPPL positif menunjukkan bahwa faset a memiliki proporsi hasil positif yang diprediksi lebih tinggi jika dibandingkan dengan segi d.

  Ini disebut sebagai bias positif.
- Nilai DPPL mendekati nol menunjukkan proporsi yang lebih sama dari hasil positif yang diprediksi antara aspek a dan d dan nilai nol menunjukkan paritas demografis yang sempurna.
- Nilai DPPL negatif menunjukkan bahwa faset d memiliki proporsi hasil positif yang diprediksi lebih tinggi jika dibandingkan dengan faset a. Ini disebut sebagai bias negatif.

Dampak Berbeda (DI)

Perbedaan proporsi positif dalam metrik label yang diprediksi dapat dinilai dalam bentuk rasio.

Perbandingan proporsi positif dalam metrik label yang diprediksi dapat dinilai dalam bentuk rasio, bukan sebagai perbedaan, seperti halnya dengan. Perbedaan Proporsi Positif pada Label Prediksi (DPPL) Metrik dampak berbeda (DI) didefinisikan sebagai rasio proporsi prediksi positif (y' = 1) untuk segi d atas proporsi prediksi positif (y' = 1) untuk faset a. Misalnya, jika prediksi model memberikan pinjaman kepada 60% dari kelompok paruh baya (segi a) dan 50% kelompok usia lainnya (segi d), maka DI = .5/.6 = 0,8, yang menunjukkan bias positif dan dampak buruk pada kelompok usia lainnya yang diwakili oleh segi d.

Rumus untuk rasio proporsi label yang diprediksi:

$$DI = q'_d / q'_a$$

Di mana:

- $q'_a = n'^{(1)}_a / n_a$ adalah proporsi prediksi dari segi a yang mendapatkan hasil positif dari nilai 1. Dalam contoh kami, proporsi aspek paruh baya diprediksi akan diberikan pinjaman. Di sini $n'^{(1)}_a$

mewakili jumlah anggota faset a yang mendapatkan hasil prediksi positif dan n_a adalah jumlah anggota faset a.

- $q'_d = n'^{(1)}_d / n_d$ adalah proporsi prediksi dari segi d a yang mendapatkan hasil positif dari nilai 1. Dalam contoh kita, aspek orang tua dan muda diprediksi akan diberikan pinjaman. Di sini $n'^{(1)}_d$ mewakili jumlah anggota segi d yang mendapatkan hasil prediksi positif dan n_d adalah jumlah anggota segi d.

Untuk biner, aspek multikategori, dan label kontinu, nilai DI berkisar selama interval [0, ∞).

- Nilai kurang dari 1 menunjukkan bahwa faset a memiliki proporsi hasil positif yang diprediksi lebih tinggi daripada segi d. Ini disebut sebagai bias positif.
- Nilai 1 menunjukkan paritas demografis.
- Nilai yang lebih besar dari 1 menunjukkan bahwa faset d memiliki proporsi hasil positif yang diprediksi lebih tinggi daripada segi a. Ini disebut sebagai bias negatif.

Perbedaan dalam Penerimaan Bersyarat (DCAcc)

Metrik ini membandingkan label yang diamati dengan label yang diprediksi oleh model dan menilai apakah ini sama di seluruh aspek untuk hasil positif yang diprediksi. Metrik ini hampir meniru bias manusia karena mengukur berapa banyak lagi hasil positif yang diprediksi model (label y') untuk aspek tertentu dibandingkan dengan apa yang diamati dalam kumpulan data pelatihan (label y). Misalnya, jika ada lebih banyak penerimaan (hasil positif) yang diamati dalam kumpulan data pelatihan untuk aplikasi pinjaman untuk kelompok paruh baya (aspek a) daripada yang diprediksi oleh model berdasarkan kualifikasi dibandingkan dengan aspek yang mengandung kelompok usia lain (aspek d), ini mungkin menunjukkan potensi bias dalam cara pinjaman disetujui menguntungkan kelompok paruh baya.

Rumus untuk perbedaan penerimaan bersyarat:

$$dCAcc = c - c_{a\,d}$$

Di mana:

- $c_a = n^{(1)}_a / n'^{(1)}_a$ adalah rasio jumlah hasil positif yang diamati dari nilai 1 (penerimaan) dari segi a terhadap jumlah hasil positif yang diprediksi (akseptansi) untuk segi a.
- $c_d = n^{(1)}_d / n'^{(1)}_d$ adalah rasio jumlah hasil positif yang diamati dari nilai 1 (akseptansi) dari segi d dengan jumlah prediksi hasil positif yang diprediksi (penerimaan) untuk segi d.

Metrik dCaCC dapat menangkap bias positif dan negatif yang mengungkapkan perlakuan preferensial berdasarkan kualifikasi. Pertimbangkan contoh bias berbasis usia berikut pada penerimaan pinjaman.

Contoh 1: Bias positif

Misalkan kita memiliki kumpulan data 100 orang paruh baya (segi a) dan 50 orang dari kelompok usia lain (segi d) yang mengajukan pinjaman, di mana model merekomendasikan agar 60 dari segi a dan 30 dari segi d diberikan pinjaman. Jadi proporsi yang diprediksi tidak bias sehubungan dengan metrik DPPL, tetapi label yang diamati menunjukkan bahwa 70 dari segi a dan 20 dari segi d diberikan pinjaman. Dengan kata lain, model memberikan pinjaman hingga 17% lebih sedikit dari segi paruh baya daripada label yang diamati dalam data pelatihan yang disarankan (70/60 = 1,17) dan memberikan pinjaman kepada 33% lebih banyak dari kelompok usia lain daripada label yang diamati disarankan (20/30 = 0,67). Perhitungan nilai DCAcc memberikan yang berikut:

DCaCC = 70/60 - 20/30 = 1/2

Nilai positif menunjukkan bahwa ada bias potensial terhadap aspek paruh baya a dengan tingkat penerimaan yang lebih rendah dibandingkan dengan aspek lain d daripada yang ditunjukkan oleh data yang diamati (dianggap tidak bias).

Contoh 2: Bias negatif

Misalkan kita memiliki kumpulan data 100 orang paruh baya (segi a) dan 50 orang dari kelompok usia lain (segi d) yang mengajukan pinjaman, di mana model merekomendasikan agar 60 dari segi a dan 30 dari segi d diberikan pinjaman. Jadi proporsi yang diprediksi tidak bias sehubungan dengan metrik DPPL, tetapi label yang diamati menunjukkan bahwa 50 dari segi a dan 40 dari segi d diberikan pinjaman. Dengan kata lain, model memberikan pinjaman hingga 17% lebih sedikit dari segi paruh baya daripada label yang diamati dalam data pelatihan yang disarankan (50/60 = 0,83), dan memberikan pinjaman kepada 33% lebih banyak dari kelompok usia lain daripada label yang diamati disarankan (40/30 = 1,33). Perhitungan nilai DCAcc memberikan yang berikut:

dCaCC = 50/60 - 40/30 = -1/2

Nilai negatif menunjukkan bahwa ada bias potensial terhadap segi d dengan tingkat penerimaan yang lebih rendah dibandingkan dengan aspek paruh baya a daripada yang ditunjukkan oleh data yang diamati (dianggap tidak bias).

Perhatikan bahwa Anda dapat menggunakan dCAcc untuk membantu Anda mendeteksi bias potensial (tidak disengaja) oleh manusia yang mengawasi prediksi model dalam pengaturan. human-in-the-loop Asumsikan, misalnya, bahwa prediksi y' oleh model tidak bias, tetapi keputusan akhirnya

dibuat oleh manusia (mungkin dengan akses ke fitur tambahan) yang dapat mengubah prediksi model untuk menghasilkan versi baru dan final dari y'. Pemrosesan tambahan oleh manusia mungkin secara tidak sengaja menolak pinjaman ke nomor yang tidak proporsional dari satu sisi. DCaCC dapat membantu mendeteksi bias potensial tersebut.

Kisaran nilai untuk perbedaan penerimaan bersyarat untuk biner, aspek multikategori, dan label kontinu adalah (-∞, +∞).

- Nilai positif terjadi ketika rasio jumlah penerimaan yang diamati dibandingkan dengan akseptansi yang diprediksi untuk faset a lebih tinggi dari rasio yang sama untuk segi d. Nilai-nilai ini menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari segi a. Semakin besar perbedaan rasio, semakin ekstrim bias yang tampak.
- Nilai mendekati nol terjadi ketika rasio jumlah penerimaan yang diamati dibandingkan dengan akseptansi yang diprediksi untuk faset a mirip dengan rasio untuk segi d. Nilai-nilai ini menunjukkan bahwa tingkat penerimaan yang diprediksi konsisten dengan nilai yang diamati dalam data berlabel dan bahwa pelamar yang memenuhi syarat dari kedua aspek diterima dengan cara yang sama.
- Nilai negatif terjadi ketika rasio jumlah penerimaan yang diamati dibandingkan dengan akseptansi yang diprediksi untuk faset a kurang dari rasio untuk segi d. Nilai-nilai ini menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari segi d. Semakin negatif perbedaan rasio, semakin ekstrim bias yang tampak.

Perbedaan Penolakan Bersyarat (DCR)

Metrik ini membandingkan label yang diamati dengan label yang diprediksi oleh model dan menilai apakah ini sama di seluruh aspek untuk hasil negatif (penolakan). Metrik ini mendekati meniru bias manusia, dalam hal itu mengukur berapa banyak lagi hasil negatif yang diberikan model (label yang diprediksi y') ke aspek tertentu dibandingkan dengan apa yang disarankan oleh label dalam kumpulan data pelatihan (label yang diamati y). Misalnya, jika ada lebih banyak penolakan yang diamati (hasil negatif) untuk aplikasi pinjaman untuk kelompok paruh baya (aspek a) daripada yang diprediksi oleh model berdasarkan kualifikasi dibandingkan dengan aspek yang mengandung kelompok usia lain (segi d), ini mungkin menunjukkan potensi bias dalam cara pinjaman ditolak yang lebih disukai kelompok paruh baya daripada kelompok lain.

Rumus untuk perbedaan penerimaan bersyarat:

$$DCR = r - r_{d\,a}$$

Di mana:

- $r_d = n_d^{(0)}/n'_d{}^{(0)}$ adalah rasio jumlah hasil negatif yang diamati dari nilai 0 (penolakan) dari segi d terhadap jumlah prediksi hasil negatif (penolakan) untuk segi d.
- $r_a = n_a^{(0)}/n'_a{}^{(0)}$ adalah rasio jumlah hasil negatif yang diamati dari nilai 0 (penolakan) dari segi a terhadap jumlah prediksi hasil negatif dari nilai 0 (penolakan) untuk segi a.

Metrik DCR dapat menangkap bias positif dan negatif yang mengungkapkan perlakuan preferensial berdasarkan kualifikasi. Pertimbangkan contoh bias berbasis usia berikut pada penolakan pinjaman.

Contoh 1: Bias positif

Misalkan kita memiliki kumpulan data 100 orang paruh baya (segi a) dan 50 orang dari kelompok usia lain (segi d) yang mengajukan pinjaman, di mana model merekomendasikan bahwa 60 dari segi a dan 30 dari segi d ditolak untuk pinjaman. Jadi proporsi yang diprediksi tidak bias oleh metrik DPPL, tetapi label yang diamati menunjukkan bahwa 50 dari segi a dan 40 dari segi d ditolak. Dengan kata lain, model menolak 17% lebih banyak pinjaman dari segi paruh baya daripada label yang diamati dalam data pelatihan yang disarankan (50/60 = 0,83), dan menolak pinjaman 33% lebih sedikit dari kelompok usia lain daripada label yang diamati yang disarankan (40/30 = 1,33). Nilai DCR mengukur perbedaan ini dalam rasio tingkat penolakan yang diamati terhadap prediksi antar aspek. Nilai positif menunjukkan bahwa ada bias potensial yang mendukung kelompok paruh baya dengan tingkat penolakan yang lebih rendah dibandingkan dengan kelompok lain daripada yang ditunjukkan oleh data yang diamati (dianggap tidak bias).

DCR = 40/30 - 50/60 = 1/2

Contoh 2: Bias negatif

Misalkan kita memiliki kumpulan data 100 orang paruh baya (segi a) dan 50 orang dari kelompok usia lain (segi d) yang mengajukan pinjaman, di mana model merekomendasikan bahwa 60 dari segi a dan 30 dari segi d ditolak untuk pinjaman. Jadi proporsi yang diprediksi tidak bias oleh metrik DPPL, tetapi label yang diamati menunjukkan bahwa 70 dari segi a dan 20 dari segi d ditolak. Dengan kata lain, model menolak pinjaman 17% lebih sedikit dari segi paruh baya daripada label yang diamati dalam data pelatihan yang disarankan (70/60 = 1,17), dan menolak 33% lebih banyak pinjaman dari kelompok usia lain daripada label yang diamati yang disarankan (20/30 = 0,67). Nilai negatif menunjukkan bahwa ada bias potensial yang mendukung aspek a dengan tingkat penolakan yang lebih rendah dibandingkan dengan aspek paruh baya a daripada yang ditunjukkan oleh data yang diamati (dianggap tidak bias).

DCR = 20/30 - 70/60 = -1/2

Kisaran nilai untuk perbedaan penolakan bersyarat untuk biner, aspek multikategori, dan label kontinu adalah (-∞, +∞).

- Nilai positif terjadi ketika rasio jumlah penolakan yang diamati dibandingkan dengan prediksi penolakan untuk segi d lebih besar dari rasio untuk faset a. Nilai-nilai ini menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari segi a. Semakin besar nilai metrik DCR, semakin ekstrem bias yang tampak.
- Nilai mendekati nol terjadi ketika rasio jumlah penolakan yang diamati dibandingkan dengan penerimaan yang diprediksi untuk faset a mirip dengan rasio untuk segi d. Nilai-nilai ini menunjukkan bahwa tingkat penolakan yang diprediksi konsisten dengan nilai yang diamati dalam data berlabel dan bahwa pelamar yang memenuhi syarat dari kedua aspek ditolak dengan cara yang sama.
- Nilai negatif terjadi ketika rasio jumlah penolakan yang diamati dibandingkan dengan prediksi penolakan untuk segi d kurang dari aspek rasio a. Nilai-nilai ini menunjukkan kemungkinan bias terhadap pelamar yang memenuhi syarat dari segi d. Semakin besar besarnya metrik DCR negatif, semakin ekstrim bias yang tampak.

Perbedaan spesifisitas (SD)

Perbedaan spesifisitas (SD) adalah perbedaan spesifisitas antara faset yang disukai a dan aspek yang tidak disukai d. Spesifisitas mengukur seberapa sering model memprediksi hasil negatif dengan benar (y'=0). Setiap perbedaan dalam kekhususan ini adalah bentuk bias potensial.

Spesifisitas sempurna untuk suatu segi jika semua kasus y=0 diprediksi dengan benar untuk segi tersebut. Spesifisitas lebih besar ketika model meminimalkan positif palsu, yang dikenal sebagai kesalahan Tipe I. Misalnya, perbedaan antara spesifisitas rendah untuk pinjaman ke segi a, dan spesifisitas tinggi untuk pinjaman ke segi d, adalah ukuran bias terhadap segi d.

Rumus berikut adalah untuk perbedaan spesifisitas untuk segi a dan d.

$$SD = TN_d/({}_dTN+FP_d) - TN_a/({}_{aa}TN+FP) = TNR - TNR_{\,d\ a}$$

Variabel berikut yang digunakan untuk menghitung SD didefinisikan sebagai berikut:

- $TN_{\,d}$ adalah negatif sebenarnya yang diprediksi untuk segi d.
- $FP_{\,d}$ adalah positif palsu yang diprediksi untuk segi d.
- $TN_{\,d}$ adalah negatif sebenarnya yang diprediksi untuk aspek a.

- FP_d adalah positif palsu yang diprediksi untuk segi a.
- $TNR_a = TN_a/(_aTN+FP_a)$ adalah tingkat negatif sebenarnya, juga dikenal sebagai spesifisitas, untuk segi a.
- $TNR_d = TN_d/(_dTN+FP_d)$ adalah tingkat negatif sebenarnya, juga dikenal sebagai spesifisitas, untuk segi d.

Misalnya, perhatikan matriks kebingungan berikut untuk segi a dan d.

Matriks kebingungan untuk segi yang disukai **a**

Prediksi kelas a	Hasil aktual 0	Hasil aktual 1	Total
0	20	5	25
1	10	65	75
Total	30	70	100

Matriks kebingungan untuk segi yang tidak disukai **d**

Prediksi kelas d	Hasil aktual 0	Hasil aktual 1	Total
0	18	7	25
1	5	20	25
Total	23	27	50

Nilai perbedaan spesifisitas adalah*SD = 18/(18+5) - 20/(20+10) = 0.7826 - 0.6667 = 0.1159*, yang menunjukkan bias terhadap segi d.

Rentang nilai untuk perbedaan spesifisitas antara aspek a dan d untuk klasifikasi biner dan multikategori adalah. `[-1, +1]` Metrik ini tidak tersedia untuk label kontinu. Inilah yang disiratkan oleh nilai-nilai SD yang berbeda:

- Nilai positif diperoleh ketika ada spesifisitas yang lebih tinggi untuk faset d daripada faset a. Ini menunjukkan bahwa model menemukan lebih sedikit positif palsu untuk faset d daripada untuk faset a. Nilai positif menunjukkan bias terhadap segi d.

- Nilai mendekati nol menunjukkan bahwa spesifisitas untuk segi yang dibandingkan serupa. Ini menunjukkan bahwa model menemukan jumlah positif palsu yang sama di kedua aspek ini dan tidak bias.
- Nilai negatif diperoleh ketika ada spesifisitas yang lebih tinggi untuk faset a daripada faset d. Ini menunjukkan bahwa model menemukan lebih banyak positif palsu untuk faset a daripada untuk segi d. Nilai negatif menunjukkan bias terhadap segi a.

Ingat Perbedaan (RD)

Metrik recall difference (RD) adalah perbedaan dalam mengingat model antara faset yang disukai a dan aspek yang tidak disukai d. Setiap perbedaan dalam penarikan ini adalah bentuk bias potensial. Ingat adalah tingkat positif sejati (TPR), yang mengukur seberapa sering model memprediksi dengan benar kasus-kasus yang seharusnya menerima hasil positif. Ingat sempurna untuk suatu segi jika semua kasus y=1 diprediksi dengan benar sebagai y'=1 untuk segi itu. Ingat lebih besar ketika model meminimalkan negatif palsu yang dikenal sebagai kesalahan Tipe II. Misalnya, berapa banyak orang dalam dua kelompok berbeda (aspek a dan d) yang harus memenuhi syarat untuk pinjaman terdeteksi dengan benar oleh model? Jika tingkat penarikan tinggi untuk pinjaman ke segi a, tetapi rendah untuk pinjaman ke segi d, perbedaannya memberikan ukuran bias ini terhadap kelompok yang termasuk dalam segi d.

Rumus untuk perbedaan tingkat penarikan untuk aspek a dan d:

$$RD = TP_a/(TP_a + FN_a) - TP_d/(TP_d + FN) = TPR - TPR_{d\ a\ d}$$

Di mana:

- TP_a adalah positif sejati yang diprediksi untuk aspek a.
- FN_a adalah negatif palsu yang diprediksi untuk segi a.
- TP_d adalah positif sejati yang diprediksi untuk segi d.
- FN_d adalah negatif palsu yang diprediksi untuk segi d.
- $TPR_a = TP_a/(TP_a + FN_a)$ adalah penarikan untuk faset a, atau tingkat positif sebenarnya.
- TPR_d $TP_d/(TP_d + FN_d)$ adalah penarikan untuk segi d, atau tingkat positif sebenarnya.

Misalnya, perhatikan matriks kebingungan berikut untuk segi a dan d.

Matriks Kebingungan untuk Aspek Favorit a

Prediksi kelas a	Hasil aktual 0	Hasil aktual 1	Total
0	20	5	25
1	10	65	75
Total	30	70	100

Matriks Kebingungan untuk Aspet yang Tidak Disukai d

Prediksi kelas d	Hasil aktual 0	Hasil aktual 1	Total
0	18	7	25
1	5	20	25
Total	23	27	50

Nilai perbedaan recall adalah RD = 65/70 - 20/27 = 0,93 - 0,74 = 0,19 yang menunjukkan bias terhadap segi d.

Rentang nilai untuk perbedaan ingatan antara segi a dan d untuk klasifikasi biner dan multikategori adalah [-1, +1]. Metrik ini tidak tersedia untuk label kontinu.

- Nilai positif diperoleh ketika ada ingatan yang lebih tinggi untuk faset a daripada untuk segi d. Hal ini menunjukkan bahwa model menemukan lebih banyak hal positif sejati untuk faset a daripada segi d, yang merupakan bentuk bias.
- Nilai mendekati nol menunjukkan bahwa penarikan kembali untuk aspek yang dibandingkan serupa. Ini menunjukkan bahwa model menemukan jumlah positif sejati yang sama di kedua aspek ini dan tidak bias.
- Nilai negatif diperoleh ketika ada ingatan yang lebih tinggi untuk faset d daripada faset a. Ini menunjukkan bahwa model menemukan lebih banyak hal positif sejati untuk segi d daripada faset a, yang merupakan bentuk bias.

Perbedaan Tingkat Penerimaan (DAR)

Perbedaan metrik tingkat penerimaan (DAR) adalah perbedaan rasio prediksi positif sejati (TP) terhadap positif yang diamati (TP + FP) untuk aspek a dan d. Metrik ini mengukur perbedaan presisi model untuk memprediksi penerimaan dari kedua aspek ini. Presisi mengukur fraksi kandidat yang memenuhi syarat dari kumpulan kandidat yang memenuhi syarat yang diidentifikasi oleh model. Jika presisi model untuk memprediksi pelamar yang memenuhi syarat berbeda antar aspek, ini adalah bias dan besarnya diukur oleh DAR.

Rumus untuk perbedaan tingkat penerimaan antara aspek a dan d:

$$DAR = TP_a/(TP_a + FP_a) - TP_d/(TP + FP)_{d\ d}$$

Di mana:

- TP_a adalah positif sejati yang diprediksi untuk aspek a.
- FP_a adalah positif palsu yang diprediksi untuk segi a.
- TP_d adalah positif sejati yang diprediksi untuk segi d.
- FP_d adalah positif palsu yang diprediksi untuk segi d.

Misalnya, model menerima 70 pelamar paruh baya (aspek a) untuk pinjaman (label positif yang diprediksi) yang hanya 35 yang benar-benar diterima (label positif yang diamati). Juga misalkan model menerima 100 pelamar dari demografi usia lain (segi d) untuk pinjaman (label positif yang diprediksi) yang hanya 40 yang benar-benar diterima (label positif yang diamati). Kemudian DAR = 35/70 - 40/100 = 0,10, yang menunjukkan potensi bias terhadap orang-orang yang memenuhi syarat dari kelompok usia kedua (segi d).

Rentang nilai DAR untuk biner, aspek multikategori, dan label kontinu adalah [-1, +1].

- Nilai positif terjadi ketika rasio positif yang diprediksi (penerimaan) terhadap hasil positif yang diamati (pelamar yang memenuhi syarat) untuk aspek a lebih besar dari rasio yang sama untuk segi d. Nilai-nilai ini menunjukkan kemungkinan bias terhadap aspek yang tidak disukai d yang disebabkan oleh terjadinya positif palsu yang relatif lebih banyak di segi d. Semakin besar perbedaan rasio, semakin ekstrim bias yang tampak.
- Nilai mendekati nol terjadi ketika rasio positif yang diprediksi (penerimaan) terhadap hasil positif yang diamati (pelamar yang memenuhi syarat) untuk aspek a dan d memiliki nilai serupa yang menunjukkan label yang diamati untuk hasil positif diprediksi dengan presisi yang sama oleh model.

- Nilai negatif terjadi ketika rasio positif yang diprediksi (penerimaan) terhadap hasil positif yang diamati (pelamar yang memenuhi syarat) untuk aspek d lebih besar dari aspek rasio a. Nilai-nilai ini menunjukkan kemungkinan bias terhadap aspek yang disukai a yang disebabkan oleh terjadinya positif yang relatif lebih palsu dalam segi a. Semakin negatif perbedaan rasio, semakin ekstrim bias yang tampak.

Perbedaan Tingkat Penolakan (DRR)

Perbedaan metrik tingkat penolakan (DRR) adalah perbedaan rasio prediksi negatif sejati (TN) terhadap negatif yang diamati (TN+FN) untuk aspek a dan d. Metrik ini mengukur perbedaan presisi model untuk memprediksi penolakan dari kedua aspek ini. Presisi mengukur fraksi kandidat yang tidak memenuhi syarat dari kumpulan kandidat yang tidak memenuhi syarat yang diidentifikasi seperti itu oleh model. Jika presisi model untuk memprediksi pelamar yang tidak memenuhi syarat berbeda antar aspek, ini adalah bias dan besarnya diukur oleh DRR.

Rumus untuk perbedaan tingkat penolakan antara segi a dan d:

$$DRR = TN_d/({}_dTN+FN_d) - TN_a/(TN+FN)_{a\ a}$$

Komponen untuk persamaan DRR sebelumnya adalah sebagai berikut.

- TN_d adalah negatif sebenarnya yang diprediksi untuk segi d.
- FN_d adalah negatif palsu yang diprediksi untuk segi d.
- TP_a adalah negatif sebenarnya yang diprediksi untuk aspek a.
- FN_a adalah negatif palsu yang diprediksi untuk segi a.

Misalnya, model menolak 100 pelamar paruh baya (aspek a) untuk pinjaman (label negatif yang diprediksi) di antaranya 80 sebenarnya tidak memenuhi syarat (label negatif yang diamati). Juga misalkan model menolak 50 pelamar dari demografi usia lain (segi d) untuk pinjaman (label negatif yang diprediksi) di antaranya hanya 40 yang benar-benar tidak memenuhi syarat (label negatif yang diamati). Kemudian DRR = 40/50 - 80/100 = 0, jadi tidak ada bias yang ditunjukkan.

Rentang nilai DRR untuk biner, aspek multikategori, dan label kontinu adalah [-1, +1].

- Nilai positif terjadi ketika rasio negatif yang diprediksi (penolakan) terhadap hasil negatif yang diamati (pelamar yang tidak memenuhi syarat) untuk aspek d lebih besar dari rasio yang sama untuk aspek a. Nilai-nilai ini menunjukkan kemungkinan bias terhadap aspek yang disukai a

yang disebabkan oleh terjadinya negatif yang relatif lebih salah dalam aspek a. Semakin besar perbedaan rasio, semakin ekstrim bias yang tampak.

- Nilai mendekati nol terjadi ketika rasio negatif yang diprediksi (penolakan) terhadap hasil negatif yang diamati (pelamar yang tidak memenuhi syarat) untuk aspek a dan d memiliki nilai yang sama, menunjukkan label yang diamati untuk hasil negatif diprediksi dengan presisi yang sama oleh model.
- Nilai negatif terjadi ketika rasio negatif yang diprediksi (penolakan) terhadap hasil negatif yang diamati (pelamar yang tidak memenuhi syarat) untuk aspek a lebih besar dari aspek rasio d. Nilai-nilai ini menunjukkan kemungkinan bias terhadap aspek yang tidak disukai d yang disebabkan oleh terjadinya positif palsu yang relatif lebih banyak di segi d. Semakin negatif perbedaan rasio, semakin ekstrim bias yang tampak.

Perbedaan Akurasi (AD)

Metrik perbedaan akurasi (AD) adalah perbedaan antara akurasi prediksi untuk berbagai aspek. Metrik ini menentukan apakah klasifikasi menurut model lebih akurat untuk satu aspek daripada yang lain. AD menunjukkan apakah satu aspek menimbulkan proporsi kesalahan Tipe I dan Tipe II yang lebih besar. Tetapi tidak dapat membedakan antara kesalahan Tipe I dan Tipe II. Misalnya, model mungkin memiliki akurasi yang sama untuk demografi usia yang berbeda, tetapi kesalahan mungkin sebagian besar positif palsu (kesalahan Tipe I) untuk satu kelompok berbasis usia dan sebagian besar negatif palsu (kesalahan Tipe II) untuk yang lain.

Juga, jika persetujuan pinjaman dibuat dengan akurasi yang jauh lebih tinggi untuk demografis paruh baya (aspek a) daripada demografis berbasis usia lainnya (aspek d), proporsi yang lebih besar dari pelamar yang memenuhi syarat di kelompok kedua ditolak pinjaman (FN) atau proporsi yang lebih besar dari pelamar yang tidak memenuhi syarat dari kelompok itu mendapatkan pinjaman (FP) atau keduanya. Hal ini dapat menyebabkan ketidakadilan kelompok untuk kelompok kedua, bahkan jika proporsi pinjaman yang diberikan hampir sama untuk kedua kelompok berbasis usia, yang ditunjukkan oleh nilai DPPL yang mendekati nol.

Rumus untuk metrik AD adalah perbedaan antara akurasi prediksi untuk facet a, ACC, dikurangi untuk facet d_a, ACC: $_d$

$$\text{IKLAN} = \text{ACC}_a - \text{ACC}_d$$

Di mana:

- $\text{ACC}_a = (\text{TP}_a + \text{TN}_a)/(\text{TP} + _a\ \text{TN+FP+FN}_a)$ $_a$ $_a$

  - TP_a adalah positif sejati yang diprediksi untuk segi a
  - TN_a adalah negatif sebenarnya yang diprediksi untuk segi a
  - FP_a adalah positif palsu yang diprediksi untuk segi a
  - FN_a adalah negatif palsu yang diprediksi untuk segi a
- $ACC_d = (TP_d + TN_d)/(TP_d + TN+FP+FN_d)$
  - TP_d adalah positif sejati yang diprediksi untuk segi d
  - TN_d adalah negatif sebenarnya yang diprediksi untuk segi d
  - FP_d adalah positif palsu yang diprediksi untuk segi d
  - FN_d adalah negatif palsu yang diprediksi untuk segi d

Misalnya, model menyetujui pinjaman kepada 70 pelamar dari segi a 100 dan menolak 30 lainnya. 10 seharusnya tidak ditawari pinjaman (FP_a) dan 60 disetujui yang seharusnya (TP). 20 penolakan seharusnya disetujui (FN_a) dan 10 ditolak dengan benar (TN_a). Akurasi untuk segi a adalah sebagai berikut:

$$ACC_a = (60 + 10)/(60 + 10 + 20 + 10) = 0,7$$

Selanjutnya, misalkan model menyetujui pinjaman kepada 50 pelamar dari segi d 100 dan menolak 50 lainnya. 10 seharusnya tidak ditawari pinjaman (FP_a) dan 40 disetujui yang seharusnya (TP_a). 40 penolakan seharusnya disetujui (FN) dan 10 ditolak dengan benar (TN_a). Keakuratan untuk faset a ditentukan sebagai berikut:

$$ACC_d = (40 + 10)/(40 + 10 + 40 + 10) = 0,5$$

Perbedaan akurasi dengan demikian $AD = ACC_a - ACC_d = 0,7 - 0,5 = 0,2$. Ini menunjukkan ada bias terhadap segi d karena metriknya positif.

Rentang nilai untuk AD untuk label faset biner dan multikategori adalah [-1, +1].

- Nilai positif terjadi ketika akurasi prediksi untuk faset a lebih besar dari segi d. Ini berarti bahwa facet d lebih menderita dari beberapa kombinasi positif palsu (kesalahan Tipe I) atau negatif palsu (kesalahan Tipe II). Ini berarti ada bias potensial terhadap aspek yang tidak disukai d.
- Nilai mendekati nol terjadi ketika akurasi prediksi untuk faset a mirip dengan faset d.
- Nilai negatif terjadi ketika akurasi prediksi untuk faset d lebih besar dari segi a t. Ini berarti bahwa facet a lebih menderita dari beberapa kombinasi positif palsu (kesalahan Tipe I) atau negatif palsu (kesalahan Tipe II). Ini berarti bias terhadap aspek yang disukai a.

Kesetaraan Perawatan (TE)

Kesetaraan perlakuan (TE) adalah perbedaan rasio negatif palsu terhadap positif palsu antara aspek a dan d. Gagasan utama dari metrik ini adalah untuk menilai apakah, bahkan jika akurasi antar kelompok sama, apakah kesalahan lebih berbahaya bagi satu kelompok daripada yang lain? Tingkat kesalahan berasal dari total positif palsu dan negatif palsu, tetapi rincian keduanya mungkin sangat berbeda di seluruh aspek. TE mengukur apakah kesalahan mengkompensasi dengan cara yang serupa atau berbeda di seluruh aspek.

Rumus untuk kesetaraan pengobatan:

$$TE = FN_d / FP - FN_a / FP_a$$

Di mana:

- FN_d adalah negatif palsu yang diprediksi untuk segi d.
- FP_d adalah positif palsu yang diprediksi untuk segi d.
- FN_a adalah negatif palsu yang diprediksi untuk segi a.
- FP_a adalah positif palsu yang diprediksi untuk segi a.

Perhatikan metrik menjadi tidak terbatas jika FP_a atau FP_d adalah nol.

Misalnya, anggaplah ada 100 pemohon pinjaman dari segi a dan 50 dari segi d. Untuk segi a, 8 salah ditolak pinjaman (FN_a) dan 6 lainnya salah disetujui (FP_a). Prediksi yang tersisa benar, jadi $TP_a + TN_a = 86$. Untuk segi d, 5 ditolak secara salah (FN_d) dan 2 disetujui secara salah (FP_d). Prediksi yang tersisa benar, jadi $TP_d + TN_d = 43$. Rasio negatif palsu terhadap positif palsu sama dengan 8/6 = 1,33 untuk faset a dan 5/2 = 2,5 untuk segi d. Oleh karena itu TE = 2.5 - 1.33 = 1.167, meskipun kedua segi memiliki akurasi yang sama:

$$ACC_a = (86)/(86+ 8 + 6) = 0,86$$

$$ACC_d = (43)/(43 + 5 + 2) = 0,86$$

Kisaran nilai untuk perbedaan penolakan bersyarat untuk label faset biner dan multikategori adalah $(-\infty, +\infty)$. Metrik TE tidak didefinisikan untuk label kontinu. Interpretasi metrik ini tergantung pada relatif penting dari positif palsu (Kesalahan tipe I) dan negatif palsu (kesalahan Tipe II).

- Nilai positif terjadi ketika rasio negatif palsu terhadap positif palsu untuk faset d lebih besar daripada untuk faset a.

- Nilai mendekati nol terjadi ketika rasio negatif palsu terhadap positif palsu untuk faset a mirip dengan rasio untuk faset d.
- Nilai negatif terjadi ketika rasio negatif palsu terhadap positif palsu untuk faset d kurang dari untuk faset a.

> **Note**
>
> Versi sebelumnya menyatakan bahwa metrik Perlakuan Kesetaraan dihitung sebagai FP/FN - $_{a}$ FP/FN $_{d}$ bukan $_{d}$ FN/FP $_{a}$ - $_{d}$ FN/FP. $_{d\ a\ a}$ Sementara salah satu versi dapat digunakan. Untuk informasi selengkapnya, lihat Fairness measures for Machine Learning in Finance.

Disparitas Demografis Bersyarat dalam Label yang Diprediksi (CDDPL)

Metrik disparitas demografis (DDPL) menentukan apakah aspek d memiliki proporsi yang lebih besar dari label yang ditolak yang diprediksi daripada label yang diterima yang diprediksi. Ini memungkinkan perbandingan perbedaan dalam proporsi penolakan yang diprediksi dan proporsi penerimaan yang diprediksi di seluruh aspek. Metrik ini persis sama dengan metrik CDD pra-pelatihan kecuali bahwa itu dihitung dari label yang diprediksi alih-alih yang diamati. Metrik ini terletak pada kisaran (-1, +1).

Rumus prediksi disparitas demografis untuk label segi d adalah sebagai berikut:

$$DDPL_{d} = n'_{d}{}^{(0)} / n'^{(0)} - n'_{d}{}^{(1)} / n'^{(1)} = P_{d}{}^{R}(y'^{0}) - P\,A\,(y'\,1)_{d}$$

Di mana:

- $n'^{(0)} = n'_{a}{}^{(0)} + n'_{d}{}^{(0)}$ adalah jumlah label yang ditolak yang diprediksi untuk segi a dan d.
- $n'^{(1)} = n'_{a}{}^{(1)} + n'_{d}{}^{(1)}$ adalah jumlah label yang diterima yang diprediksi untuk segi a dan d.
- $P_{d}{}^{R}(y'^{0})$ adalah proporsi label ditolak yang diprediksi (nilai 0) dalam segi d.
- $P_{d}{}^{A}(y'^{1})$ adalah proporsi label yang diterima yang diprediksi (nilai 1) dalam segi d.

Metrik disparitas demografis bersyarat dalam label prediksi (CDDPL) yang mengkondisikan DDPL pada atribut yang menentukan strata subkelompok pada kumpulan data diperlukan untuk mengesampingkan paradoks Simpson. Pengelompokan kembali dapat memberikan wawasan tentang penyebab kesenjangan demografis yang jelas untuk aspek yang kurang disukai. Kasus klasik

muncul dalam kasus penerimaan Berkeley di mana pria diterima pada tingkat yang lebih tinggi secara keseluruhan daripada wanita. Tetapi ketika subkelompok departemen diperiksa, wanita terbukti memiliki tingkat penerimaan yang lebih tinggi daripada pria berdasarkan departemen. Penjelasannya adalah bahwa wanita telah mendaftar ke departemen dengan tingkat penerimaan yang lebih rendah daripada pria. Meneliti tingkat penerimaan subkelompok mengungkapkan bahwa wanita sebenarnya diterima pada tingkat yang lebih tinggi daripada pria untuk departemen dengan tingkat penerimaan yang lebih rendah.

Metrik CDDPL memberikan ukuran tunggal untuk semua perbedaan yang ditemukan dalam subkelompok yang ditentukan oleh atribut kumpulan data dengan merata-ratakannya. Ini didefinisikan sebagai rata-rata tertimbang disparitas demografis dalam label prediksi ($DDPL_i$) untuk masing-masing subkelompok, dengan setiap disparitas subkelompok tertimbang secara proporsional dengan jumlah pengamatan dalam mengandung. Rumus untuk disparitas demografis bersyarat dalam label yang diprediksi adalah sebagai berikut:

$$CDDPL = (1/n) * \Sigma\, n\, DDPL_{i\,i} * {}_{i}$$

Di mana:

- ${}_{i}n_{i} = n$ adalah jumlah total pengamatan dan n_{i} adalah jumlah pengamatan untuk setiap subkelompok.
- $DDPL_{i} = n'_{i}{}^{(0)} / n^{(0)} - n'_{i}{}^{(1)} / n^{(1)} = P_{i}{}^{R}(y'^{0}) - P_{i}{}^{A}(y'^{1})$ adalah perbedaan demografis dalam label yang diprediksi untuk subkelompok.

Jadi perbedaan demografis untuk subkelompok dalam label prediksi ($DDPL_i$) adalah perbedaan antara proporsi label yang ditolak yang diprediksi dan proporsi label yang diterima yang diprediksi untuk setiap subkelompok.

Kisaran nilai DDPL untuk hasil biner, multikategori, dan kontinu adalah [-1, +1].

- +1: ketika tidak ada label penolakan yang diprediksi untuk faset a atau subkelompok dan tidak ada penerimaan yang diprediksi untuk segi d atau subkelompok.
- Nilai positif menunjukkan ada perbedaan demografis dalam label yang diprediksi karena aspek d atau subkelompok memiliki proporsi yang lebih besar dari label yang ditolak yang diprediksi daripada label yang diterima yang diprediksi. Semakin tinggi nilainya semakin besar disparitas.
- Nilai mendekati nol menunjukkan tidak ada perbedaan demografis rata-rata.

- Nilai negatif menunjukkan ada perbedaan demografis dalam label yang diprediksi karena aspek a atau subkelompok memiliki proporsi yang lebih besar dari label yang ditolak yang diprediksi daripada label yang diterima yang diprediksi. Semakin rendah nilainya, semakin besar disparitas.
- -1: ketika tidak ada kerah penolakan yang diprediksi untuk segi d atau subkelompok dan tidak ada penerimaan yang diprediksi untuk faset a atau subkelompok.

Fliptest Kontrafaktual (FT)

Fliptest adalah pendekatan yang melihat setiap anggota facet d dan menilai apakah anggota faset yang serupa memiliki prediksi model yang berbeda. Anggota faset a dipilih menjadi k-tetangga terdekat dari pengamatan dari segi d. Kami menilai berapa banyak tetangga terdekat dari kelompok lawan yang menerima prediksi yang berbeda, di mana prediksi terbalik dapat berubah dari positif ke negatif dan sebaliknya.

Rumus untuk fliptest kontrafaktual adalah perbedaan kardinalitas dua himpunan dibagi dengan jumlah anggota segi d:

$$FT = (F^{+} - F^{-}) / n_{d}$$

Di mana:

- F^{+} = adalah jumlah anggota sisi d yang tidak disukai dengan hasil yang tidak menguntungkan yang tetangga terdekatnya dalam aspek yang disukai menerima hasil yang menguntungkan.
- F^{-} = adalah jumlah anggota sisi d yang tidak disukai dengan hasil yang menguntungkan yang tetangga terdekatnya dalam aspek yang disukai menerima hasil yang tidak menguntungkan.
- n_{d} adalah ukuran sampel dari segi d.

Rentang nilai untuk fliptest kontrafaktual untuk label faset biner dan multikategori adalah [-1, +1]. Untuk label kontinu, kami menetapkan ambang batas untuk menciutkan label ke biner.

- Nilai positif terjadi ketika jumlah keputusan terlentang kontrafaktual yang tidak menguntungkan untuk aspek yang tidak disukai d melebihi yang menguntungkan.
- Nilai mendekati nol terjadi ketika jumlah keputusan fliptest kontrafaktual yang tidak menguntungkan dan menguntungkan seimbang.
- Nilai negatif terjadi ketika jumlah keputusan terlentang kontrafaktual yang tidak menguntungkan untuk aspek yang tidak disukai d kurang dari yang menguntungkan.

Entropi umum (GE)

Indeks entropi umum (GE) mengukur ketidaksetaraan manfaat b untuk label yang diprediksi dibandingkan dengan label yang diamati. Manfaat terjadi ketika positif palsu diprediksi. Positif palsu terjadi ketika pengamatan negatif (y=0) memiliki prediksi positif (y'=1). Manfaat juga terjadi ketika label yang diamati dan diprediksi sama, juga dikenal sebagai positif sejati dan negatif sejati. Tidak ada manfaat yang terjadi ketika negatif palsu diprediksi. Negatif palsu terjadi ketika pengamatan positif (y=1) diprediksi memiliki hasil negatif (y'=0). Manfaatnya b didefinisikan, sebagai berikut.

```
b = y' - y + 1
```

Dengan menggunakan definisi ini, positif palsu menerima manfaat b dari2, dan negatif palsu menerima manfaat dari`0`. Baik positif sejati maupun negatif sejati menerima manfaat dari`1`.

Metrik GE dihitung mengikuti Indeks Entropi Umum (GE) dengan bobot diatur ke. `alpha 2` Bobot ini mengontrol sensitivitas terhadap nilai manfaat yang berbeda. Yang lebih kecil `alpha` berarti peningkatan sensitivitas terhadap nilai yang lebih kecil.

$$GE = \frac{1}{2n}\sum_{i=1}^{n}\left[\left(\frac{b_i}{b'}\right)^2 - 1\right]$$

Variabel berikut yang digunakan untuk menghitung GE didefinisikan sebagai berikut:

- b ¡ adalah manfaat yang diterima oleh titik `i`th data.
- b 'adalah arti dari semua manfaat.

GE dapat berkisar dari 0 hingga 0,5, di mana nilai nol menunjukkan tidak ada ketidaksetaraan manfaat di semua titik data. Ini terjadi baik ketika semua input diprediksi dengan benar atau ketika semua prediksi positif palsu. GE tidak terdefinisi ketika semua prediksi adalah negatif palsu.

> **Note**
>
> Metrik GE tidak bergantung pada nilai faset yang disukai atau tidak disukai.

Penjelasan Model

Amazon SageMaker Clarify menyediakan alat untuk membantu menjelaskan bagaimana model pembelajaran mesin (ML) membuat prediksi. Alat-alat ini dapat membantu pemodel dan pengembang dan pemangku kepentingan internal lainnya memahami karakteristik model secara keseluruhan sebelum penerapan dan untuk men-debug prediksi yang disediakan oleh model setelah diterapkan.

- Untuk mendapatkan penjelasan untuk kumpulan data dan model Anda, lihat. Gunakan SageMaker Clarify untuk menjelaskan dan mendeteksi bias
- Untuk mendapatkan penjelasan secara real-time dari SageMaker titik akhir, lihat. Penjelasan Online dengan Clarify SageMaker

Transparansi tentang bagaimana model ML sampai pada prediksi mereka juga penting bagi konsumen dan regulator. Mereka perlu mempercayai prediksi model jika mereka akan menerima keputusan berdasarkan prediksi tersebut. SageMaker Clarify menggunakan pendekatan atribusi fitur model-agnostik. Anda dapat menggunakan ini untuk memahami mengapa model membuat prediksi setelah pelatihan, dan untuk memberikan penjelasan per contoh selama inferensi. Implementasinya mencakup implementasi SHAP yang skalabel dan efisien. Ini didasarkan pada konsep nilai Shapley, dari bidang teori permainan kooperatif, yang memberikan setiap fitur nilai penting untuk prediksi tertentu.

Clarify menghasilkan plot ketergantungan sebagian (PDP) yang menunjukkan fitur efek marjinal terhadap hasil prediksi dari model pembelajaran mesin. Ketergantungan sebagian membantu menjelaskan respons target yang diberikan serangkaian fitur input. Ini juga mendukung penjelasan visi komputer (CV) dan pemrosesan bahasa alami (NLP) menggunakan algoritma Shapley Values (SHAP) yang sama seperti yang digunakan untuk penjelasan data tabel.

Apa fungsi penjelasan dalam konteks pembelajaran mesin? Penjelasan dapat dianggap sebagai jawaban atas pertanyaan Mengapa yang membantu manusia memahami penyebab prediksi. Dalam konteks model ML, Anda mungkin tertarik untuk menjawab pertanyaan seperti:

- Mengapa model memprediksi hasil negatif seperti penolakan pinjaman untuk pemohon tertentu?
- Bagaimana model membuat prediksi?
- Mengapa model membuat prediksi yang salah?
- Fitur mana yang memiliki pengaruh terbesar pada perilaku model?

Anda dapat menggunakan penjelasan untuk mengaudit dan memenuhi persyaratan peraturan, membangun kepercayaan pada model dan mendukung pengambilan keputusan manusia, serta men-debug dan meningkatkan kinerja model.

Kebutuhan untuk memenuhi tuntutan pemahaman manusia tentang sifat dan hasil inferensi ML adalah kunci untuk jenis penjelasan yang dibutuhkan. Penelitian dari filsafat dan disiplin ilmu kognitif telah menunjukkan bahwa orang peduli terutama tentang penjelasan kontras, atau penjelasan mengapa suatu peristiwa X terjadi alih-alih beberapa peristiwa lain Y yang tidak terjadi. Di sini, X bisa menjadi peristiwa tak terduga atau mengejutkan yang terjadi dan Y sesuai dengan harapan berdasarkan model mental mereka yang ada yang disebut sebagai baseline. Perhatikan bahwa untuk peristiwa X yang sama, orang yang berbeda mungkin mencari penjelasan yang berbeda tergantung pada sudut pandang mereka atau model mental Y. Dalam konteks AI yang dapat dijelaskan, Anda dapat menganggap X sebagai contoh yang sedang dijelaskan dan Y sebagai garis dasar yang biasanya dipilih untuk mewakili contoh yang tidak informatif atau rata-rata dalam kumpulan data. Terkadang, misalnya dalam kasus pemodelan gambar dari ML, baseline mungkin implisit, di mana gambar yang pikselnya memiliki warna yang sama dapat berfungsi sebagai garis dasar.

Contoh Notebook

Amazon SageMaker Clarify menyediakan contoh notebook berikut untuk penjelasan model:

- Amazon SageMaker Clarify Processing — Gunakan SageMaker Clarify untuk membuat pekerjaan pemrosesan untuk mendeteksi bias dan menjelaskan prediksi model dengan atribusi fitur. Contohnya termasuk menggunakan format data CSV dan JSON Lines, membawa wadah Anda sendiri, dan menjalankan pekerjaan pemrosesan dengan Spark.
- Menjelaskan Klasifikasi Gambar dengan SageMaker Clarify — SageMaker Clarify memberi Anda wawasan tentang bagaimana model visi komputer Anda mengklasifikasikan gambar.
- Menjelaskan model deteksi objek dengan SageMaker Clarify — SageMaker Clarify memberi Anda wawasan tentang cara model visi komputer Anda mendeteksi objek.

Notebook ini telah diverifikasi untuk berjalan di Amazon SageMaker Studio saja. Jika Anda memerlukan petunjuk tentang cara membuka buku catatan di Amazon SageMaker Studio, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic. Jika Anda diminta untuk memilih kernel, pilih Python 3 (Ilmu Data).

Topik

- Atribusi Fitur yang Menggunakan Nilai Shapley

- Garis Dasar SHAP untuk Penjelasan

Atribusi Fitur yang Menggunakan Nilai Shapley

SageMaker Clarify menyediakan atribusi fitur berdasarkan konsep nilai Shapley. Anda dapat menggunakan nilai Shapley untuk menentukan kontribusi yang dibuat setiap fitur untuk memodelkan prediksi. Atribusi ini dapat disediakan untuk prediksi spesifik dan pada tingkat global untuk model secara keseluruhan. Misalnya, jika Anda menggunakan model ML untuk penerimaan perguruan tinggi, penjelasannya dapat membantu menentukan apakah IPK atau skor SAT adalah fitur yang paling bertanggung jawab atas prediksi model, dan kemudian Anda dapat menentukan seberapa bertanggung jawab setiap fitur untuk menentukan keputusan penerimaan tentang siswa tertentu.

SageMaker Clarify telah mengambil konsep nilai-nilai Shapley dari teori permainan dan menerapkannya dalam konteks pembelajaran mesin. Nilai Shapley menyediakan cara untuk mengukur kontribusi setiap pemain ke permainan, dan karenanya sarana untuk mendistribusikan total keuntungan yang dihasilkan oleh permainan kepada para pemainnya berdasarkan kontribusi mereka. Dalam konteks pembelajaran mesin ini, SageMaker Clarify memperlakukan prediksi model pada contoh tertentu sebagai permainan dan fitur yang termasuk dalam model sebagai pemain. Untuk perkiraan pertama, Anda mungkin tergoda untuk menentukan kontribusi atau efek marjinal dari setiap fitur dengan mengukur hasil dari menjatuhkan fitur itu dari model atau menjatuhkan semua fitur lain dari model. Namun, pendekatan ini tidak memperhitungkan bahwa fitur yang termasuk dalam model seringkali tidak independen satu sama lain. Misalnya, jika dua fitur sangat berkorelasi, menjatuhkan salah satu fitur mungkin tidak mengubah prediksi model secara signifikan.

Untuk mengatasi dependensi potensial ini, nilai Shapley mensyaratkan bahwa hasil dari setiap kemungkinan kombinasi (atau koalisi) fitur harus dipertimbangkan untuk menentukan pentingnya setiap fitur. Diberikan fitur d, ada 2^d kemungkinan kombinasi fitur seperti itu, masing-masing sesuai dengan model potensial. Untuk menentukan atribusi untuk fitur tertentu f, pertimbangkan kontribusi marjinal dari memasukkan f dalam semua kombinasi fitur (dan model terkait) yang tidak mengandung f, dan ambil rata-rata. Dapat ditunjukkan bahwa nilai Shapley adalah cara unik untuk menetapkan kontribusi atau pentingnya setiap fitur yang memenuhi properti tertentu yang diinginkan. Secara khusus, jumlah nilai Shapley dari setiap fitur sesuai dengan perbedaan antara prediksi model dan model tiruan tanpa fitur. Namun, bahkan untuk nilai d yang wajar, katakanlah 50 fitur, secara komputasi penghalang dan tidak praktis untuk melatih 2 d model yang mungkin. Akibatnya, SageMaker Clarify perlu menggunakan berbagai teknik aproksimasi. Untuk tujuan ini, SageMaker Clarify menggunakan Shapley Additive Explanations (SHAP), yang menggabungkan perkiraan tersebut dan merancang implementasi algoritma Kernel SHAP yang dapat diskalakan dan efisien melalui pengoptimalan tambahan.

Untuk informasi tambahan tentang nilai Shapley, lihat Pendekatan Terpadu untuk Menafsirkan Prediksi Model.

Garis Dasar SHAP untuk Penjelasan

Penjelasan biasanya kontrastif (yaitu, mereka menjelaskan penyimpangan dari garis dasar). Akibatnya, untuk prediksi model yang sama, Anda dapat mengharapkan untuk mendapatkan penjelasan yang berbeda sehubungan dengan garis dasar yang berbeda. Karena itu, pilihan baseline Anda sangat penting. Dalam konteks ML, baseline sesuai dengan contoh hipotetis yang dapat berupa tidak informatif atau informatif. Selama perhitungan nilai Shapley, SageMaker Clarify menghasilkan beberapa instance baru antara baseline dan instance yang diberikan, di mana tidak adanya fitur, dimodelkan dengan menetapkan nilai fitur ke nilai baseline dan keberadaan fitur dimodelkan dengan menetapkan nilai fitur ke nilai instance yang diberikan. Dengan demikian, tidak adanya semua fitur sesuai dengan baseline dan keberadaan semua fitur sesuai dengan contoh yang diberikan.

Bagaimana Anda bisa memilih baseline yang baik? Seringkali diinginkan untuk memilih garis dasar dengan konten informasi yang sangat rendah. Misalnya, Anda dapat membuat instance rata-rata dari kumpulan data pelatihan dengan mengambil median atau rata-rata untuk fitur numerik dan mode untuk fitur kategoris. Untuk contoh penerimaan perguruan tinggi, Anda mungkin tertarik untuk menjelaskan mengapa pelamar tertentu diterima dibandingkan dengan penerimaan dasar berdasarkan pelamar rata-rata. Jika tidak disediakan, baseline dihitung secara otomatis dengan SageMaker Clarify menggunakan K-mean atau K-prototipe dalam dataset input.

Atau, Anda dapat memilih untuk menghasilkan penjelasan sehubungan dengan garis dasar informatif. Untuk skenario penerimaan perguruan tinggi, Anda mungkin ingin menjelaskan mengapa pelamar tertentu ditolak jika dibandingkan dengan pelamar lain dari latar belakang demografis yang sama. Dalam hal ini, Anda dapat memilih garis dasar yang mewakili pelamar yang diminati, yaitu mereka yang memiliki latar belakang demografis yang sama. Dengan demikian, Anda dapat menggunakan garis dasar informatif untuk memusatkan analisis pada aspek spesifik dari prediksi model tertentu. Anda dapat mengisolasi fitur untuk penilaian dengan menyetel atribut demografis dan fitur lain yang tidak dapat ditindaklanjuti dengan nilai yang sama seperti pada instance yang diberikan.

Gunakan SageMaker Clarify menjelaskan dengan Autopilot SageMaker

Autopilot menggunakan alat yang disediakan oleh Amazon SageMaker Clarify untuk membantu memberikan wawasan tentang bagaimana model pembelajaran mesin (ML) membuat prediksi.

Alat-alat ini dapat membantu insinyur, manajer produk, dan pemangku kepentingan internal lainnya memahami karakteristik model. Untuk mempercayai dan menafsirkan keputusan yang dibuat berdasarkan prediksi model, baik konsumen maupun regulator mengandalkan transparansi dalam pembelajaran mesin secara berurutan.

Fungsionalitas penjelasan Autopilot menggunakan pendekatan atribusi fitur model-agnostik. Pendekatan ini menentukan kontribusi fitur atau input individu terhadap output model, memberikan wawasan tentang relevansi fitur yang berbeda. Anda dapat menggunakannya untuk memahami mengapa model membuat prediksi setelah pelatihan, atau menggunakannya untuk memberikan penjelasan per contoh selama inferensi. Implementasinya mencakup implementasi SHAP (Shapley Additive Explanations) yang dapat diskalakan. Implementasi ini didasarkan pada konsep nilai Shapley dari teori permainan kooperatif, yang memberikan setiap fitur nilai penting untuk prediksi tertentu.

Anda dapat menggunakan penjelasan SHAP untuk hal-hal berikut: mengaudit dan memenuhi persyaratan peraturan, membangun kepercayaan pada model, mendukung pengambilan keputusan manusia, atau men-debug dan meningkatkan kinerja model.

Untuk informasi tambahan tentang nilai dan garis dasar Shapely, lihat Garis Dasar SHAP untuk Keterjelasan.

Untuk panduan dokumentasi Amazon SageMaker Clarify, lihat Panduan untuk SageMaker Klarifikasi Dokumentasi.

Gunakan tata kelola untuk mendokumentasikan dan melacak kinerja model

Tata kelola model adalah kerangka kerja yang memberikan visibilitas sistematis ke dalam pengembangan, validasi, dan penggunaan model pembelajaran mesin (ML). Amazon SageMaker menyediakan alat tata kelola yang dibuat khusus untuk mengelola akses kontrol, pelacakan aktivitas, dan pelaporan di seluruh siklus hidup ML.

Kelola izin hak istimewa paling sedikit untuk praktisi ML menggunakan Amazon SageMaker Role Manager, buat dokumentasi model terperinci menggunakan Kartu Model Amazon, dan dapatkan visibilitas ke SageMaker model Anda dengan dasbor terpusat menggunakan Dasbor Model Amazon. SageMaker

Manajer SageMaker Peran Amazon

Dengan Amazon SageMaker Role Manager, administrator dapat menentukan izin pengguna dengan izin hak istimewa paling sedikit untuk aktivitas pembelajaran mesin umum. Gunakan Amazon SageMaker Role Manager untuk membangun dan mengelola peran IAM berbasis persona khusus untuk kebutuhan bisnis Anda.

Untuk informasi selengkapnya, lihat Manajer SageMaker Peran Amazon.

Kartu SageMaker Model Amazon

Gunakan Kartu SageMaker Model Amazon untuk mendokumentasikan, mengambil, dan berbagi informasi model penting dari konsepsi hingga penerapan. Dengan kartu model, manajer risiko model, ilmuwan data, dan insinyur ML dapat membuat catatan yang tidak dapat diubah dari penggunaan model yang dimaksudkan, peringkat risiko, detail pelatihan, hasil evaluasi, dan banyak lagi.

Untuk informasi selengkapnya, lihat Kartu SageMaker Model Amazon.

Dasbor SageMaker Model Amazon

Dasbor SageMaker Model Amazon adalah ikhtisar visual yang dibuat sebelumnya dari semua model di akun Anda. SageMaker Dasbor Model mengintegrasikan informasi berharga dari Amazon SageMaker Model Monitor, Transform Jobs, Endpoints, MLLineage Tracking, dan Amazon

CloudWatch sehingga Anda dapat mengakses informasi model tingkat tinggi dan melacak kinerja model dalam satu tampilan terpadu.

Untuk informasi selengkapnya, lihat Dasbor SageMaker Model Amazon.

Kartu SageMaker Model Amazon

Gunakan Kartu SageMaker Model Amazon untuk mendokumentasikan detail penting tentang model pembelajaran mesin (ML) Anda di satu tempat untuk tata kelola dan pelaporan yang efisien.

Detail katalog seperti tujuan penggunaan dan peringkat risiko model, detail dan metrik pelatihan, hasil evaluasi dan pengamatan, dan panggilan tambahan seperti pertimbangan, rekomendasi, dan informasi khusus. Dengan membuat kartu model, Anda dapat melakukan hal berikut:

- Memberikan panduan tentang bagaimana model harus digunakan.
- Support kegiatan audit dengan deskripsi rinci tentang pelatihan model dan kinerja.
- Komunikasikan bagaimana model dimaksudkan untuk mendukung tujuan bisnis.

Kartu model memberikan panduan preskriptif tentang informasi apa yang akan didokumentasikan dan menyertakan bidang untuk informasi khusus. Setelah membuat kartu model, Anda dapat mengekspornya ke PDF atau mengunduhnya untuk dibagikan dengan pemangku kepentingan terkait. Setiap pengeditan selain pembaruan status persetujuan yang dilakukan pada kartu model menghasilkan versi kartu model tambahan agar memiliki catatan perubahan model yang tidak dapat diubah.

Topik

Prasyarat

Untuk memulai Kartu SageMaker Model Amazon, Anda harus memiliki izin untuk membuat, mengedit, melihat, dan mengekspor kartu model.

Penggunaan model yang dimaksudkan

Menentukan tujuan penggunaan model membantu memastikan bahwa pengembang model dan pengguna memiliki informasi yang mereka butuhkan untuk melatih atau menyebarkan model secara bertanggung jawab. Penggunaan model yang dimaksudkan harus menggambarkan skenario di mana model sesuai untuk digunakan serta skenario di mana model tidak disarankan untuk digunakan.

Kami merekomendasikan termasuk:

- Tujuan umum dari model
- Gunakan kasus yang modelnya dimaksudkan
- Gunakan kasus yang modelnya tidak dimaksudkan
- Asumsi yang dibuat saat mengembangkan model

Penggunaan model yang dimaksudkan melampaui rincian teknis dan menjelaskan bagaimana model harus digunakan dalam produksi, skenario di mana sesuai untuk menggunakan model, dan pertimbangan tambahan seperti jenis data yang akan digunakan dengan model atau asumsi apa pun yang dibuat selama pengembangan.

Peringkat risiko

Pengembang membuat model ML untuk kasus penggunaan dengan berbagai tingkat risiko. Misalnya, model yang menyetujui aplikasi pinjaman mungkin merupakan model risiko yang lebih tinggi daripada model yang mendeteksi kategori email. Mengingat profil risiko yang bervariasi dari suatu model, kartu model menyediakan bidang bagi Anda untuk mengkategorikan peringkat risiko model.

Peringkat risiko ini bisa`unknown`,`low`,`medium`, atau`high`. Gunakan bidang peringkat risiko ini untuk memberi label model yang tidak diketahui, rendah, sedang, atau berisiko tinggi dan bantu organisasi Anda mematuhi aturan yang ada tentang memasukkan model tertentu ke dalam produksi.

Kartu model skema JSON

Rincian evaluasi untuk kartu model harus disediakan dalam format JSON. Jika Anda memiliki laporan evaluasi format JSON yang dibuat oleh SageMaker Clarify atau SageMaker Model Monitor, unggah

laporan tersebut ke Amazon S3 dan berikan URI S3 untuk mengurai metrik evaluasi secara otomatis. Untuk informasi selengkapnya dan contoh laporan, lihat folder metrik contoh di buku catatan contoh Tata Kelola SageMaker Model Amazon - Kartu Model.

Saat membuat kartu model menggunakan SageMaker Python SDK, konten model harus dalam skema JSON kartu model dan disediakan sebagai string. Berikan konten model yang mirip dengan contoh berikut.

Kartu model file sampel skema JSON

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://json-schema.org/draft-07/schema#",
  "title": "SageMakerModelCardSchema",
  "description": "Default model card schema",
  "version": "0.1.0",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "model_overview": {
      "description": "Overview about the model",
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "model_description": {
          "description": "description of model",
          "type": "string",
          "maxLength": 1024
        },
        "model_owner": {
          "description": "Owner of model",
          "type": "string",
          "maxLength": 1024
        },
        "model_creator": {
          "description": "Creator of model",
          "type": "string",
          "maxLength": 1024
        },
        "problem_type": {
          "description": "Problem being solved with the model",
          "type": "string"
```

```
        },
        "algorithm_type": {
          "description": "Algorithm used to solve the problem",
          "type": "string",
          "maxLength": 1024
        },
        "model_id": {
          "description": "SageMaker Model Arn or Non SageMaker Model id",
          "type": "string",
          "maxLength": 1024
        },
        "model_artifact": {
          "description": "Location of the model artifact",
          "type": "array",
          "maxContains": 15,
          "items": {
            "type": "string",
            "maxLength": 1024
          }
        },
        "model_name": {
          "description": "Name of the model",
          "type": "string",
          "maxLength": 1024
        },
        "model_version": {
          "description": "Version of the model",
          "type": "number",
          "minimum": 1
        },
        "inference_environment": {
          "description": "Overview about the inference",
          "type": "object",
          "additionalProperties": false,
          "properties": {
            "container_image": {
              "description": "SageMaker inference image uri",
              "type": "array",
              "maxContains": 15,
              "items": {
                "type": "string",
                "maxLength": 1024
              }
            }
```

```
          }
        }
      }
    },
    "model_package_details": {
      "description": "Metadata information related to model package version",
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "model_package_description": {
          "description": "A brief summary of the model package",
          "type": "string",
          "maxLength": 1024
        },
        "model_package_arn": {
          "description": "The Amazon Resource Name (ARN) of the model package",
          "type": "string",
          "minLength": 1,
          "maxLength": 2048
        },
        "created_by": {
          "description": "Information about the user who created model package.",
          "type": "object",
          "additionalProperties": false,
          "properties": {
            "user_profile_name": {
              "description": "The name of the user's profile in SageMaker Studio",
              "type": "string",
              "maxLength": 63
            }
          }
        },
        "model_package_status": {
          "description": "Current status of model package",
          "type": "string",
          "enum": [
            "Pending",
            "InProgress",
            "Completed",
            "Failed",
            "Deleting"
          ]
        },
        "model_approval_status": {
```

```
          "description": "Current approval status of model package",
          "type": "string",
          "enum": [
            "Approved",
            "Rejected",
            "PendingManualApproval"
          ]
        },
        "approval_description": {
          "description": "A description provided for the model approval",
          "type": "string",
          "maxLength": 1024
        },
        "model_package_group_name": {
          "description": "If the model is a versioned model, the name of the model
group that the versioned model belongs to.",
          "type": "string",
          "minLength": 1,
          "maxLength": 63
        },
        "model_package_name": {
          "description": "Name of the model package",
          "type": "string",
          "minLength": 1,
          "maxLength": 63
        },
        "model_package_version": {
          "description": "Version of the model package",
          "type": "number",
          "minimum": 1
        },
        "domain": {
          "description": "The machine learning domain of the model package you
specified. Common machine learning domains include computer vision and natural
language processing.",
          "type": "string"
        },
        "task": {
          "description": "The machine learning task you specified that your model
package accomplishes. Common machine learning tasks include object detection and image
classification.",
          "type": "string"
        },
        "source_algorithms": {
```

```
          "description": "A list of algorithms that were used to create a model
package.",
          "$ref": "#/definitions/source_algorithms"
        },
        "inference_specification": {
          "description": "Details about inference jobs that can be run with models
based on this model package.",
          "$ref": "#/definitions/inference_specification"
        }
      }
    },
    "intended_uses": {
      "description": "Intended usage of model",
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "purpose_of_model": {
          "description": "Why the model was developed?",
          "type": "string",
          "maxLength": 2048
        },
        "intended_uses": {
          "description": "intended use cases",
          "type": "string",
          "maxLength": 2048
        },
        "factors_affecting_model_efficiency": {
          "type": "string",
          "maxLength": 2048
        },
        "risk_rating": {
          "description": "Risk rating for model card",
          "$ref": "#/definitions/risk_rating"
        },
        "explanations_for_risk_rating": {
          "type": "string",
          "maxLength": 2048
        }
      }
    },
    "business_details": {
      "description": "Business details of model",
      "type": "object",
      "additionalProperties": false,
```

```
      "properties": {
        "business_problem": {
          "description": "What business problem does the model solve?",
          "type": "string",
          "maxLength": 2048
        },
        "business_stakeholders": {
          "description": "Business stakeholders",
          "type": "string",
          "maxLength": 2048
        },
        "line_of_business": {
          "type": "string",
          "maxLength": 2048
        }
      }
    },
    "training_details": {
      "description": "Overview about the training",
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "objective_function": {
          "description": "the objective function the model will optimize for",
          "function": {
            "$ref": "#/definitions/objective_function"
          },
          "notes": {
            "type": "string",
            "maxLength": 1024
          }
        },
        "training_observations": {
          "type": "string",
          "maxLength": 1024
        },
        "training_job_details": {
          "type": "object",
          "additionalProperties": false,
          "properties": {
            "training_arn": {
              "description": "SageMaker Training job arn",
              "type": "string",
              "maxLength": 1024
```

```
            },
            "training_datasets": {
              "description": "Location of the model datasets",
              "type": "array",
              "maxContains": 15,
              "items": {
                "type": "string",
                "maxLength": 1024
              }
            },
            "training_environment": {
              "type": "object",
              "additionalProperties": false,
              "properties": {
                "container_image": {
                  "description": "SageMaker training image uri",
                  "type": "array",
                  "maxContains": 15,
                  "items": {
                    "type": "string",
                    "maxLength": 1024
                  }
                }
              }
            },
            "training_metrics": {
              "type": "array",
              "items": {
                "maxItems": 50,
                "$ref": "#/definitions/training_metric"
              }
            },
            "user_provided_training_metrics": {
              "type": "array",
              "items": {
                "maxItems": 50,
                "$ref": "#/definitions/training_metric"
              }
            },
            "hyper_parameters": {
              "type": "array",
              "items": {
                "maxItems": 100,
                "$ref": "#/definitions/training_hyper_parameter"
```

```
            }
          },
          "user_provided_hyper_parameters": {
            "type": "array",
            "items": {
              "maxItems": 100,
              "$ref": "#/definitions/training_hyper_parameter"
            }
          }
        }
      }
    }
  },
  "evaluation_details": {
    "type": "array",
    "default": [],
    "items": {
      "type": "object",
      "required": [
        "name"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "pattern": ".{1,63}"
        },
        "evaluation_observation": {
          "type": "string",
          "maxLength": 2096
        },
        "evaluation_job_arn": {
          "type": "string",
          "maxLength": 256
        },
        "datasets": {
          "type": "array",
          "items": {
            "type": "string",
            "maxLength": 1024
          },
          "maxItems": 10
        },
        "metadata": {
```

```
            "description": "additional attributes associated with the evaluation
 results",
            "type": "object",
            "additionalProperties": {
              "type": "string",
              "maxLength": 1024
            }
          },
          "metric_groups": {
            "type": "array",
            "default": [],
            "items": {
              "type": "object",
              "required": [
                "name",
                "metric_data"
              ],
              "properties": {
                "name": {
                  "type": "string",
                  "pattern": ".{1,63}"
                },
                "metric_data": {
                  "type": "array",
                  "items": {
                    "anyOf": [
                      {
                        "$ref": "#/definitions/simple_metric"
                      },
                      {
                        "$ref": "#/definitions/linear_graph_metric"
                      },
                      {
                        "$ref": "#/definitions/bar_chart_metric"
                      },
                      {
                        "$ref": "#/definitions/matrix_metric"
                      }
                    ]

                  }
                }
              }
            }
```

```
          }
        }
      }
    },
    "additional_information": {
      "additionalProperties": false,
      "type": "object",
      "properties": {
        "ethical_considerations": {
          "description": "Any ethical considerations that the author wants to provide",
          "type": "string",
          "maxLength": 2048
        },
        "caveats_and_recommendations": {
          "description": "Caveats and recommendations for people who might use this
model in their applications.",
          "type": "string",
          "maxLength": 2048
        },
        "custom_details": {
          "type": "object",
          "additionalProperties": {
            "$ref": "#/definitions/custom_property"
          }
        }
      }
    }
  },
  "definitions": {
    "source_algorithms": {
      "type": "array",
      "minContains": 1,
      "maxContains": 1,
      "items": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "algorithm_name"
        ],
        "properties": {
          "algorithm_name": {
            "description": "The name of an algorithm that was used to create the model
package. The algorithm must be either an algorithm resource in your SageMaker account
or an algorithm in AWS Marketplace that you are subscribed to.",
```

```
            "type": "string",
            "maxLength": 170
          },
          "model_data_url": {
            "description": "The Amazon S3 path where the model artifacts, which result
from model training, are stored.",
            "type": "string",
            "maxLength": 1024
          }
        }
      }
    },
    "inference_specification": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "containers"
      ],
      "properties": {
        "containers": {
          "description": "Contains inference related information which were used to
create model package.",
          "type": "array",
          "minContains": 1,
          "maxContains": 15,
          "items": {
            "type": "object",
            "additionalProperties": false,
            "required": [
              "image"
            ],
            "properties": {
              "model_data_url": {
                "description": "The Amazon S3 path where the model artifacts, which
result from model training, are stored.",
                "type": "string",
                "maxLength": 1024
              },
              "image": {
                "description": "Inference environment path. The Amazon EC2 Container
Registry (Amazon ECR) path where inference code is stored.",
                "type": "string",
                "maxLength": 255
              },
```

```
            "nearest_model_name": {
              "description": "The name of a pre-trained machine learning benchmarked
by Amazon SageMaker Inference Recommender model that matches your model.",
              "type": "string"
            }
          }
        }
      }
    },
    "risk_rating": {
      "description": "Risk rating of model",
      "type": "string",
      "enum": [
        "High",
        "Medium",
        "Low",
        "Unknown"
      ]
    },
    "custom_property": {
      "description": "Additional property in section",
      "type": "string",
      "maxLength": 1024
    },
    "objective_function": {
      "description": "objective function that training job is optimized for",
      "additionalProperties": false,
      "properties": {
        "function": {
          "type": "string",
          "enum": [
            "Maximize",
            "Minimize"
          ]
        },
        "facet": {
          "type": "string",
          "maxLength": 63
        },
        "condition": {
          "type": "string",
          "maxLength": 63
        }
```

```
      }
    },
    "training_metric": {
      "description": "training metric data",
      "type": "object",
      "required": [
        "name",
        "value"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "pattern": ".{1,255}"
        },
        "notes": {
          "type": "string",
          "maxLength": 1024
        },
        "value": {
          "type": "number"
        }
      }
    },
    "training_hyper_parameter": {
      "description": "training hyper parameter",
      "type": "object",
      "required": [
        "name",
        "value"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "pattern": ".{1,255}"
        },
        "value": {
          "type": "string",
          "pattern": ".{1,255}"
        }
      }
    },
    "linear_graph_metric": {
```

```
      "type": "object",
      "required": [
        "name",
        "type",
        "value"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "pattern": ".{1,255}"
        },
        "notes": {
          "type": "string",
          "maxLength": 1024
        },
        "type": {
          "type": "string",
          "enum": [
            "linear_graph"
          ]
        },
        "value": {
          "anyOf": [
            {
              "type": "array",
              "items": {
                "type": "array",
                "items": {
                  "type": "number"
                },
                "minItems": 2,
                "maxItems": 2
              },
              "minItems": 1
            }
          ]
        },
        "x_axis_name": {
          "$ref": "#/definitions/axis_name_string"
        },
        "y_axis_name": {
          "$ref": "#/definitions/axis_name_string"
        }
```

```
      }
    },
    "bar_chart_metric": {
      "type": "object",
      "required": [
        "name",
        "type",
        "value"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "pattern": ".{1,255}"
        },
        "notes": {
          "type": "string",
          "maxLength": 1024
        },
        "type": {
          "type": "string",
          "enum": [
            "bar_chart"
          ]
        },
        "value": {
          "anyOf": [
            {
              "type": "array",
              "items": {
                "type": "number"
              },
              "minItems": 1
            }
          ]
        },
        "x_axis_name": {
          "$ref": "#/definitions/axis_name_array"
        },
        "y_axis_name": {
          "$ref": "#/definitions/axis_name_string"
        }
      }
    },
```

```
    "matrix_metric": {
      "type": "object",
      "required": [
        "name",
        "type",
        "value"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "pattern": ".{1,255}"
        },
        "notes": {
          "type": "string",
          "maxLength": 1024
        },
        "type": {
          "type": "string",
          "enum": [
            "matrix"
          ]
        },
        "value": {
          "anyOf": [
            {
              "type": "array",
              "items": {
                "type": "array",
                "items": {
                  "type": "number"
                },
                "minItems": 1,
                "maxItems": 20
              },
              "minItems": 1,
              "maxItems": 20
            }
          ]
        },
        "x_axis_name": {
          "$ref": "#/definitions/axis_name_array"
        },
        "y_axis_name": {
```

```
          "$ref": "#/definitions/axis_name_array"
        }
      }
    },
    "simple_metric": {
      "description": "metric data",
      "type": "object",
      "required": [
        "name",
        "type",
        "value"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "pattern": ".{1,255}"
        },
        "notes": {
          "type": "string",
          "maxLength": 1024
        },
        "type": {
          "type": "string",
          "enum": [
            "number",
            "string",
            "boolean"
          ]
        },
        "value": {
          "anyOf": [
            {
              "type": "number"
            },
            {
              "type": "string",
              "maxLength": 63
            },
            {
              "type": "boolean"
            }
          ]
        },
```

```
        "x_axis_name": {
          "$ref": "#/definitions/axis_name_string"
        },
        "y_axis_name": {
          "$ref": "#/definitions/axis_name_string"
        }
      }
    },
    "axis_name_array": {
      "type": "array",
      "items": {
        "type": "string",
        "maxLength": 63
      }
    },
    "axis_name_string": {
      "type": "string",
      "maxLength": 63
    }
  }
}
```

Buat kartu model

Anda dapat membuat Kartu SageMaker Model Amazon menggunakan SageMaker konsol atau SageMaker Python SDK. Anda juga dapat menggunakan operasi API secara langsung. Untuk informasi selengkapnya tentang operasi API, lihatGunakan kartu model melalui API tingkat rendah.

Buat kartu model menggunakan SageMaker konsol

Pergi ke SageMaker konsol Amazon. Di panel navigasi, di bawah Tata Kelola, pilih Kartu model. Di pojok kanan atas, pilih Buat kartu model.

Ikuti empat langkah dalam prompt Create model card untuk mendokumentasikan detail tentang model Anda.

Langkah 1: Masukkan detail model dan tujuan penggunaan

Jika model Anda adalah AWS sumber daya, tentukan nama model yang tepat di bidang ini untuk mengisi detail model secara otomatis. Untuk menelusuri nama model yang ada, lihat Model di SageMaker konsol Amazon. Setiap nama model unik hanya dapat memiliki satu kartu model terkait.

Jika model Anda bukan AWS sumber daya, berikan nama unik untuk model Anda. Untuk menambahkan model sebagai AWS sumber daya, lihat Membuat model di Panduan SageMaker Pengembang Amazon. Atau, Anda dapat menambahkan model Anda sebagai paket model menggunakan SageMakerMarketplace atau SageMaker Model Registry.

Untuk informasi selengkapnya tentang penggunaan yang diinginkan, lihatPenggunaan model yang dimaksudkan. Untuk informasi selengkapnya tentang peringkat risiko, lihatPeringkat risiko.

Langkah 2: Masukkan detail pelatihan

Tambahkan detail pelatihan, pengamatan pelatihan, kumpulan data, hiperparameter, dan detail tentang fungsi objektif model ke kartu model.

Fungsi obyektif dalam kartu model dapat berupa fungsi apa pun yang dioptimalkan selama pelatihan. Ini dapat mencakup, tetapi tidak terbatas pada, fungsi biaya, fungsi kerugian, atau metrik objektif. Di bagian ini, dokumentasikan fungsi objektif yang paling penting untuk melatih model Anda.

Kami menyarankan Anda membuat katalog atribut berikut dari fungsi tujuan Anda:

- Arah optimasi
- Metrik
- Deskripsi

Misalnya, Anda dapat meminimalkan (arah pengoptimalan) kehilangan entropi silang (metrik) untuk masalah klasifikasi biner (deskripsi) atau memaksimalkan kemungkinan regresi logistik. Selain itu, Anda dapat memberikan catatan tentang mengapa Anda memilih fungsi tujuan ini daripada yang lain.

Langkah 3: Masukkan detail evaluasi

Jika Anda memiliki laporan evaluasi yang dihasilkan oleh SageMaker Clarify atau Model Monitor, berikan URI S3 untuk laporan tersebut atau unggah secara manual untuk menambahkannya ke kartu model.

Untuk informasi lebih lanjut tentang SageMaker Clarify, lihat Menjalankan SageMaker Clarify Processing Jobs for Bias Analysis and Explainability.

Untuk informasi selengkapnya tentang pemantauan penyimpangan dalam metrik kualitas model menggunakan Model Monitor, lihat Memantau kualitas model.

Untuk menambahkan laporan evaluasi Anda sendiri, pilih Evaluasi kartu model generik. Semua laporan evaluasi kartu model harus ada diKartu model skema JSON.

Langkah 4: Masukkan detail tambahan

Tambahkan bidang detail kartu model khusus untuk informasi tambahan apa pun yang ingin Anda alamat pada kartu model Anda. Misalnya, Anda mungkin menyertakan bidang kustom Lini bisnis dengan nilai Keuangan pribadi.

Menyimpan kartu model

Setelah meninjau informasi di kartu model Anda, pilih Simpan di sudut kanan bawah untuk menyimpan kartu model Anda.

Buat kartu model menggunakan SageMaker Python SDK

Sebelum membuat kartu model, Anda harus terlebih dahulu menentukan konten kartu model Anda. Saat menggunakan SageMaker Python SDK, konten model terdiri dari ikhtisar model, detail pelatihan, penggunaan yang dimaksudkan, detail evaluasi, dan informasi tambahan.

Anda dapat membuat kartu model untuk:

- Model yang di-host di dalam SageMaker
- Paket model (model) dalam Registri SageMaker Model
- Model yang di-host atau terdaftar di luar SageMaker

Anda juga dapat membuat kartu model tanpa mengaitkan model apa pun dengannya.

Sebaiknya tambahkan model yang telah Anda latih ke Registri SageMaker Model. Registri model membantu Anda membuat katalog model dan melacak versi model. Saat Anda membuat kartu model, informasi tentang model dari registri model secara otomatis mengisi kartu model. Anda dapat mengedit kartu model atau menambahkan informasi ke dalamnya setelah Anda membuatnya.

Untuk informasi tentang menggunakan registri model, lihatDaftarkan dan Terapkan Model dengan Registri Model. Untuk informasi tentang membuat kartu model dari registri model, lihatBuat kartu model untuk model Anda di SageMaker Model Registry.

> **Note**
>
> Untuk menggunakan kartu model dengan SageMaker Python SDK, Anda harus terlebih dahulu membuat Session. SageMaker Untuk informasi selengkapnya, lihat Sesi dalam referensi SageMaker Python SDK API.

Untuk membuat kartu model untuk model yang tidak ada di Registri SageMaker Model, lihatBuat model yang tidak ada dalam registri model.

Buat model yang tidak ada dalam registri model

Gunakan informasi di bagian berikut untuk membuat kartu model untuk model yang belum Anda tambahkan ke registri model.

Langkah 1: Tentukan Ikhtisar Model

Tentukan ikhtisar model Anda.

```
model_overview = ModelOverview.from_model_name(
model_name=model_name,
sagemaker_session=sagemaker_session,
model_description="A-description-of-your-model",
problem_type="Problem-type", # For example, "Binary Classification"
algorithm_type="Algorithm-type", # For example, "Logistic Regression"
model_creator="Name-of-model-creator",
model_owner="Name-of-model-owner",
)
```

Jika model Anda adalah AWS sumber daya, maka informasi ikhtisar seperti model ARN, URI wadah inferensi, dan lokasi artefak model S3 dapat diambil secara otomatis. Cetak AWS metadata terkait dengan perintah berikut:

```
print(model_overview.model_id)
print(model_overview.inference_environment.container_image)
print(model_overview.model_artifact)
```

Langkah 2: Tentukan detail pelatihan

Untuk menentukan detail pelatihan model Anda, Anda harus terlebih dahulu menentukan fungsi objektifnya.

```
objective_function = ObjectiveFunction(
    function=Function(
        function=ObjectiveFunctionEnum.MINIMIZE,
        facet=FacetEnum.LOSS,
    ),
    notes="An-explanation-about-objective-function",
)
```

Selanjutnya, Anda dapat menentukan detail pelatihan Anda menggunakan ikhtisar model, sesi, dan fungsi objektif yang ada. Tambahkan pengamatan pelatihan apa pun di sini.

```
training_details = TrainingDetails.from_model_overview(
    model_overview=model_overview,
    sagemaker_session=sagemaker_session,
    objective_function=objective_function,
    training_observations="Model-training-observations",
)
```

Sekali lagi, jika model Anda adalah AWS sumber daya, detail pelatihan tertentu diisi otomatis. Cetak pekerjaan pelatihan ARN, URI wadah pelatihan, dan metrik pelatihan dengan perintah berikut:

```
print(training_details.training_job_details.training_arn)
print(training_details.training_job_details.training_environment.container_image)
print([{"name": i.name, "value": i.value} for i in
 training_details.training_job_details.training_metrics])
```

Tentukan detail evaluasi

Untuk menentukan detail evaluasi model Anda, Anda harus terlebih dahulu menentukan satu atau beberapa grup metrik untuk menjelaskan metrik yang digunakan untuk pekerjaan evaluasi apa pun.

```
my_metric_group = MetricGroup(
name="binary classification metrics",
metric_data=[Metric(name="accuracy", type=MetricTypeEnum.NUMBER, value=0.5)]
)
```

Selanjutnya, Anda dapat menentukan detail evaluasi Anda menggunakan metrik evaluasi dan kumpulan data untuk setiap pekerjaan evaluasi. Tambahkan pengamatan evaluasi apa pun di sini dan berikan nama unik pada pekerjaan evaluasi Anda.

```
evaluation_details = [
    EvaluationJob(
        name="Example-evaluation-job",
        evaluation_observation="Evaluation-observations",
        datasets=["s3://path/to/evaluation/data"],
        metric_groups=[my_metric_group],
    )
]
```

Jika Anda memiliki laporan evaluasi yang dibuat oleh SageMakerClarify atau SageMaker Model Monitor, unggah laporan tersebut ke Amazon S3 dan berikan URI S3 untuk mengurai metrik evaluasi secara otomatis. Untuk menambahkan laporan evaluasi kartu model generik Anda sendiri, berikan laporan dalam format JSON hasil evaluasi.

```
report_type = "clarify_bias.json"
example_evaluation_job.add_metric_group_from_json(
    f"example_metrics/{report_type}", EvaluationMetricTypeEnum.CLARIFY_BIAS
)
```

Langkah 3: Tentukan Penggunaan yang Dituju

Tentukan tujuan penggunaan model, termasuk tujuan umum model dan kasus penggunaan yang dimaksudkan. Juga disarankan untuk memasukkan faktor apa pun yang berpotensi kemanjuran model ini dalam kasus penggunaan tertentu dan peringkat risiko model organisasi Anda. Lihat informasi yang lebih lengkap di Penggunaan model yang dimaksudkan dan Peringkat risiko.

```
intended_uses = IntendedUses(
purpose_of_model="Purpose-of-the-model",
intended_uses="The-intended-uses-of-this-model",
factors_affecting_model_efficiency="Any-factors-effecting-model-efficacy",
risk_rating=RiskRatingEnum.LOW,
explanations_for_risk_rating="Explanation-for-low-risk-rating",
)
```

Tentukan informasi tambahan

Terakhir, Anda dapat menambahkan informasi kustom tambahan ke kartu model Anda. Anda dapat mendokumentasikan pertimbangan etis, peringatan, dan rekomendasi tentang model. Anda juga dapat menambahkan detail kustom apa pun yang Anda inginkan dalam bentuk pasangan nilai kunci.

```
additional_information = AdditionalInformation(
ethical_considerations="Any-ethical-considerations",
caveats_and_recommendations="Any-caveats-and-recommendations",
custom_details={"custom details1": "details-value"},
)
```

Langkah 4: Buat kartu model

Beri nama kartu model Anda, tentukan kartu model, lalu gunakan definisi itu untuk membuat kartu model menggunakan SageMaker Python SDK.

```
model_card_name = "my-model-card"
my_card = ModelCard(
    name=model_card_name,
    status=ModelCardStatusEnum.DRAFT,
    model_overview=model_overview,
    training_details=training_details,
    intended_uses=intended_uses,
    evaluation_details=evaluation_details,
    additional_information=additional_information,
    sagemaker_session=sagemaker_session,
)
my_card.create()
```

Buat kartu model untuk model Anda di SageMaker Model Registry

Sebelum Anda mulai membuat kartu model, pastikan Anda telah membuat grup paket model dan paket model. Untuk informasi selengkapnya tentang cara menggunakan registri model, lihatDaftarkan dan Terapkan Model dengan Registri Model.

> **Important**
>
> Anda harus memiliki izin untuk menggunakan operasi di SageMaker Model Registry. Sebaiknya gunakan kebijakan `AmazonSageMakerModelRegistryFullAccess` AWS terkelola. Untuk informasi selengkapnya tentang kebijakan terkelola, lihat AWSKebijakan Terkelola untuk Registri Model.

Gunakan SageMaker Python SDK untuk membuat kartu model untuk paket model dalam Registry Model. SageMaker Paket model adalah model yang telah Anda latih. Saat Anda membuat kartu model, Amazon SageMaker Model Cards secara otomatis mengimpor data dari paket model ke dalam kartu model.

Saat Anda membuat kartu model untuk paket model, Amazon SageMaker Model Card menggunakan DescribeModelPackageoperasi untuk menambahkan data dari paket model ke kartu model. Berikut ini adalah contoh bidang yang dapat diimpor dari paket model ke kartu model:

- ModelDataUrl
- ModelPackageDescription
- ModelPackageGroupName

- ModelPackageStatus
- ModelPackageVersion

Gunakan kode berikut untuk menentukan paket model dan membuat kartu model darinya:

```
mp_details = ModelPackage.from_model_package_arn(
    model_package_arn="example_model_package_arn",
    sagemaker_session=sagemaker_session,
)

model_card_name = "example-model-card"
my_card = ModelCard(
    name=model_card_name,
    status=ModelCardStatusEnum.status,
    model_package_details=mp_details,
    sagemaker_session=sagemaker_session,
)
my_card.create()
```

Untuk*status*, Anda menentukan status persetujuan kartu model. Jika Anda tidak menentukan status, Kartu SageMaker Model menggunakan nilai defaultDRAFT. Jika Anda tidak menentukan SageMaker sesi, Kartu SageMaker Model menggunakan SageMaker sesi default.

Anda harus menentukan nama untuk model dan Amazon Resource Name (ARN) dari paket model. Untuk informasi tentang mendapatkan Amazon Resource Name (ARN) untuk paket model, lihat. Lihat Detail Versi Model (Boto3)

Kartu model yang Anda buat dari paket model mungkin memiliki informasi yang hilang atau tidak akurat. Anda dapat menambahkan informasi ke kartu model atau mengeditnya. Untuk informasi selengkapnya tentang mengelola kartu model Anda, lihatKelola kartu model.

SageMaker Model Registry mendukung pembuatan versi paket model Anda. Anda dapat membuat versi paket model Anda dan membuat kartu model untuk setiap versi. Informasi dari kartu model versi sebelumnya dibawa ke kartu model yang dibuat dari versi berikutnya. Misalnya, Anda dapat memiliki versi 1, versi 2, dan versi 3 dari paket model. Misalkan Anda sudah membuat kartu model untuk versi 1, tetapi Anda belum membuatnya untuk versi 2. Jika Anda membuat kartu model untuk versi 3, Kartu SageMaker Model Amazon secara otomatis membawa informasi dari kartu model untuk versi 1 ke kartu model untuk versi 3.

> **Note**
>
> Anda juga dapat membuat kartu model untuk paket model yang tidak menggunakan versi. Namun, sebagian besar alur kerja pembelajaran mesin melibatkan beberapa versi dari model yang sama, jadi sebaiknya lakukan hal berikut:
>
> 1. Membuat versi untuk setiap paket model
> 2. Membuat kartu model untuk setiap versi paket model

Kelola kartu model

Setelah Anda membuat kartu model, Anda dapat mengelolanya. Mengelola kartu model mencakup tindakan berikut:

- Mengedit kartu model
- Menghapus kartu model
- Mengekspor kartu model ke PDF

Anda dapat mengelola menggunakan SageMaker konsol Amazon atau SageMaker Python SDK.

Mengelola kartu model menggunakan konsol

Gunakan informasi di bagian berikut untuk mengelola kartu model Anda dengan SageMaker konsol Amazon.

Edit kartu model

Untuk mengedit kartu model, navigasikan ke kartu model pilihan Anda dengan memilih namanya di konsol Kartu SageMaker Model Amazon dan pilih Edit.

Setelah Anda menyimpan kartu model, Anda tidak dapat mengedit nama kartu model itu. Setelah Anda menyimpan versi kartu model, Anda tidak dapat memperbarui versi kartu model tersebut. Setiap pengeditan yang perlu Anda lakukan disimpan sebagai versi berikutnya untuk memiliki catatan perubahan model yang tidak dapat diubah.

Untuk melihat versi kartu model yang berbeda, pilih Tindakan, Pilih versi, lalu pilih versi yang ingin Anda lihat.

Ekspor kartu model

Ikuti langkah-langkah ini untuk mengekspor kartu model.

1. Buka konsol Kartu SageMaker Model Amazon.
2. Pilih nama kartu model yang ingin Anda ekspor.
3. Dalam ikhtisar kartu model, pilih Tindakan dan kemudian Ekspor PDF.
4. Masukkan URI S3 atau telusuri bucket S3 yang tersedia untuk PDF kartu model Anda.
5. Jika kartu model Anda berhasil diekspor, Anda dapat memilih Unduh PDF di spanduk yang dihasilkan atau mengunduh PDF Anda langsung dari Amazon S3.

Hapus kartu model

Ikuti langkah-langkah ini untuk menghapus satu atau beberapa kartu model secara permanen.

1. Buka konsol Kartu SageMaker Model Amazon.
2. Centang kotak di sebelah kiri nama kartu yang ingin Anda hapus.
3. Pilih Hapus di pojok kanan atas.
4. Konfirmasikan permintaan Anda untuk menghapus satu atau lebih kartu secara permanen..

Anda juga dapat menghapus kartu model saat melihat ikhtisar kartu model di konsol dengan memilih Tindakan dan kemudian Hapus kartu model.

Kelola kartu model menggunakan SageMaker Python SDK

Gunakan informasi di bagian berikut untuk mengelola kartu model Anda dengan Amazon SageMaker Python SDK.

Gunakan kartu model melalui SageMaker Python SDK

Anda dapat membuat Kartu SageMaker Model Amazon secara terprogram melalui Python SageMaker SDK. Untuk informasi selengkapnya, lihat Kartu SageMaker Model Amazon dalam referensi SageMaker Python SDK API.

Edit kartu model

Anda dapat mengedit kartu model menggunakan `model_card.update()` metode ini. Memperbarui kartu model membuat versi kartu model baru untuk memiliki catatan perubahan model yang tidak dapat diubah. Anda tidak dapat memperbarui nama kartu model.

```
my_card.model_overview.model_description = "updated-model-decription"
my_card.update()
```

Ekspor kartu model

Tentukan jalur keluaran S3 dan ekspor PDF kartu model Anda ke sana dengan perintah berikut:

```
s3_output_path = f"s3://{bucket}/{prefix}/export"
pdf_s3_url = my_card.export_pdf(s3_output_path=s3_output_path).delete()
```

Hapus kartu model

Hapus kartu model secara permanen dengan perintah berikut:

```
my_card.delete()
```

Contoh notebook

Untuk informasi selengkapnya tentang bekerja dengan kartu model melalui SageMaker Python SDK, lihat buku catatan SageMaker contoh Tata Kelola Model Amazon - Kartu Model.

Dukungan lintas akun untuk Kartu SageMaker Model Amazon

Gunakan dukungan lintas akun di Kartu SageMaker Model Amazon untuk berbagi kartu model antar AWS akun. Akun tempat kartu model dibuat adalah akun kartu model. Pengguna di akun kartu model membagikannya dengan akun bersama. Pengguna di akun bersama dapat memperbarui kartu model atau membuat PDF dari mereka.

Pengguna di akun kartu model membagikan kartu model mereka melalui AWS Resource Access Manager (AWS RAM). AWS RAMmembantu Anda berbagi sumber daya di seluruh AWS akun. Untuk pengantarAWS RAM, lihat Apa ituAWS Resource Access Manager?

Berikut ini adalah proses untuk berbagi kartu model:

1. Seorang pengguna di akun kartu model mengatur berbagi kartu model lintas akun menggunakanAWS Resource Access Manager.
2. Jika kartu model dienkripsi dengan AWS KMS kunci, pengguna yang mengatur berbagi model juga harus memberikan izin kepada pengguna di akun bersamaAWS KMS.
3. Pengguna di akun bersama menerima undangan untuk berbagi sumber daya.
4. Seorang pengguna di akun bersama memberi pengguna lain izin untuk mengakses kartu model.

Jika Anda pengguna di akun kartu model, lihat bagian berikut:

- Mengatur berbagi kartu model lintas akun
- Mengatur AWS KMS izin untuk akun bersama
- Dapatkan tanggapan atas undangan berbagi sumber daya Anda

Jika Anda pengguna di akun bersama, lihat Mengatur izin pengguna IAM di akun bersama tentang menyiapkan izin untuk diri sendiri dan pengguna lain di akun tersebut.

Mengatur berbagi kartu model lintas akun

Gunakan AWS Resource Access Manager (AWS RAM) untuk memberi pengguna di AWS akun Anda akses untuk melihat atau memperbarui kartu model yang dibuat di AWS akun lain.

Untuk mengatur berbagi kartu model, Anda harus membuat pembagian sumber daya. Pembagian sumber daya menentukan:

- Sumber daya yang dibagikan
- Siapa atau apa yang memiliki akses ke sumber daya
- Izin terkelola untuk sumber daya

Untuk informasi selengkapnya tentang pembagian sumber daya, lihat Persyaratan dan konsep untuk AWS RAM. Kami merekomendasikan meluangkan waktu untuk memahami AWS RAM secara konseptual sebelum Anda melalui proses membuat pembagian sumber daya.

> **Important**
>
> Anda harus memiliki izin untuk membuat sumber daya. Untuk informasi selengkapnya tentang izin, lihat Cara AWS RAM kerja dengan IAM.

Untuk prosedur untuk membuat pembagian sumber daya dan informasi tambahan tentangnya, lihat Membuat berbagi sumber daya.

Saat Anda menjalani prosedur membuat pembagian sumber daya, Anda menentukan `sagemaker:ModelCard` sebagai jenis sumber daya. Anda juga harus menentukan Amazon Resource Number (ARN) kebijakan berbasis sumber AWS RAM daya. Anda dapat menentukan kebijakan default atau kebijakan yang memiliki izin tambahan untuk membuat PDF kartu model.

Dengan kebijakan `AWSRAMPermissionSageMakerModelCards` berbasis sumber daya default, pengguna di akun bersama memiliki izin untuk melakukan operasi berikut:

- DescribeModelCard
- ListModelCardVersions
- UpdateModelCard

Dengan kebijakan `AWSRAMPermissionSageMakerModelCardsAllowExport` berbasis sumber daya, pengguna di akun bersama memiliki izin untuk melakukan semua tindakan sebelumnya. Mereka juga memiliki izin untuk membuat pekerjaan ekspor kartu model dan menjelaskannya melalui operasi berikut:

- CreateModelCardExportJob
- DescribeModelCardExportJob

Pengguna di akun bersama dapat membuat pekerjaan ekspor untuk menghasilkan PDF kartu model. Mereka juga dapat menggambarkan pekerjaan ekspor yang telah dibuat untuk menemukan URI Amazon S3 PDF.

Kartu model dan pekerjaan ekspor adalah sumber daya. Akun kartu model memiliki pekerjaan ekspor yang dibuat oleh pengguna di akun bersama. Misalnya, pengguna di akun A berbagi kartu model X dengan akun bersama B. Seorang pengguna di akun B membuat pekerjaan ekspor Y untuk kartu model X yang menyimpan output di lokasi Amazon S3 yang ditentukan pengguna di akun B. Meskipun akun B membuat pekerjaan ekspor Y, itu milik akun A.

Setiap AWS akun memiliki kuota sumber daya. Untuk informasi tentang kuota yang terkait dengan kartu model, lihat SageMaker titik akhir dan kuota Amazon.

Mengatur AWS KMS izin untuk akun bersama

Jika kartu model yang Anda bagikan telah dienkripsi dengan AWS Key Management Service kunci, Anda juga perlu membagikan akses ke kunci dengan akun bersama. Jika tidak, pengguna di akun bersama tidak dapat melihat, memperbarui, atau mengekspor kartu model. Untuk ikhtisarAWS KMS, lihat AWS Key Management Service.

Untuk memberikan AWS KMS izin kepada pengguna di akun bersama, perbarui kebijakan kunci Anda dengan pernyataan berikut:

```
{
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::shared-account-id::role/example-IAM-role"
        ]
    },
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt",
    ]
    "Resource": "arn:aws:kms:AWS-Region-of-model-card-account:model-card-account-id:key/AWS KMS-key-id"
    "Condition": {
        "Bool": {"kms:GrantIsForAWSResource": true },
        "StringEquals": {
            "kms:ViaService": [
                "sagemaker.AWS-Region.amazonaws.com",
                "s3.AWS-Region.amazonaws.com"
            ],
        },
        "StringLike": {
          "kms:EncryptionContext:aws:sagemaker:model-card-arn": "arn:aws:sagemaker:AWS-Region:model-card-account-id:model-card/model-card-name"
        }
    }
}
```

Pernyataan sebelumnya memberi pengguna di akun bersama `kms:Decrypt` dan `kms:GenerateDataKey` izin. Dengan`kms:Decrypt`, pengguna dapat mendekripsi kartu model. Dengan`kms:GenerateDataKey`, pengguna dapat mengenkripsi kartu model yang mereka perbarui atau PDF yang mereka buat.

Dapatkan tanggapan atas undangan berbagi sumber daya Anda

Setelah Anda membuat pembagian sumber daya, akun bersama yang telah Anda tentukan dalam pembagian sumber daya menerima undangan untuk bergabung. Mereka harus menerima undangan untuk mengakses sumber daya.

Untuk informasi tentang menerima undangan berbagi sumber daya, lihat Menggunakan sumber AWS daya bersama di Panduan Pengguna AWS Resource Access Manager.

Mengatur izin pengguna IAM di akun bersama

Informasi berikut mengasumsikan bahwa Anda telah menerima undangan berbagi sumber daya dari akun kartu model. Untuk informasi selengkapnya tentang menerima undangan berbagi sumber daya, lihat Menggunakan AWS sumber daya bersama.

Anda dan pengguna lain di akun Anda menggunakan peran IAM untuk mengakses kartu model yang dibagikan dari akun kartu model. Gunakan template berikut untuk mengubah kebijakan peran IAM. Anda dapat memodifikasi template untuk kasus penggunaan Anda sendiri.

```
{
     "Version": "2012-10-17",
     "Statement": [
        {
             "Effect": "Allow",
             "Action": [
                 "sagemaker:DescribeModelCard",
                 "sagemaker:UpdateModelCard",
                 "sagemaker:CreateModelCardExportJob",
                 "sagemaker:ListModelCardVersions",
                 "sagemaker:DescribeModelCardExportJob"
             ],
             "Resource": [
                 "arn:aws:sagemaker:AWS-Region:AWS-model-card-account-id:model-
card/example-model-card-name-0",
                 "arn:aws:sagemaker:AWS-Region:AWS-model-card-account-id:model-
card/example-model-card-name-1/*"
             ]
        },
        {
             "Effect": "Allow",
             "Action": "s3:PutObject",
             "Resource": "arn:aws:s3:::Amazon-S3-bucket-storing-the-pdf-of-the-model-
card/model-card-name/*"
        }
    ]
}
```

Untuk mengakses kartu model yang dienkripsi menggunakanAWS KMS, Anda harus memberikan izin berikut AWS KMS kepada pengguna di akun Anda.

```
{
     "Effect": "Allow",
     "Action": [
         "kms:GenerateDataKey",
         "kms:Decrypt",
     ],
     "Resource": "arn:aws:kms:AWS-Region:AWS-account-id-where-the-model-card-is-created:key/AWS Key Management Service-key-id"
}
```

Gunakan kartu model melalui API tingkat rendah

Anda dapat membuat Kartu SageMaker Model Amazon langsung melalui SageMaker API atau AWS Command Line Interface (AWS CLI).

> **Note**
>
> Saat membuat kartu model dengan API tingkat rendah, konten harus dalam skema JSON kartu model dan disediakan sebagai string. Untuk informasi selengkapnya, lihat Kartu model skema JSON.

SageMaker API

Gunakan perintah SageMaker API berikut untuk bekerja dengan Kartu SageMaker Model Amazon:

- CreateModelCard
- DescribeModelCard
- ListModelCards
- ListModelCardVersions
- UpdateModelCard
- CreateModelCardExportJob
- DescribeModelCardExportJob

- ListModelCardExportJobs
- DeleteModelCard

CLI AWS

Gunakan perintah AWS CLI berikut untuk bekerja dengan Kartu SageMaker Model Amazon:

- create-model-card
- describe-model-card
- list-model-cards
- list-model-card-versions
- update-model-card
- create-model-card-export-pekerjaan
- describe-model-card-export-pekerjaan
- list-model-card-export-pekerjaan
- delete-model-card

FAQ kartu model

Lihat item FAQ berikut untuk jawaban atas pertanyaan umum tentang Kartu SageMaker Model Amazon.

T. Apa itu risiko model?

J: Anda dapat menggunakan model untuk berbagai aplikasi bisnis mulai dari memprediksi serangan cyber dan menyetujui aplikasi pinjaman hingga mendeteksi kategori email. Masing-masing aplikasi ini mengasumsikan tingkat risiko yang berbeda. Misalnya, salah mendeteksi serangan cyber memiliki dampak bisnis yang jauh lebih besar daripada salah mengkategorikan email. Mengingat profil risiko model yang bervariasi ini, Anda dapat menggunakan kartu model untuk memberikan peringkat risiko`low`,`medium`, atau `high` untuk model. Jika Anda tidak tahu risiko model Anda, Anda dapat mengatur statusnya`unknown`. Pelanggan bertanggung jawab untuk menetapkan profil risiko untuk setiap model. Berdasarkan peringkat risiko, organisasi mungkin memiliki aturan yang berbeda untuk menerapkan model tersebut ke produksi. Untuk informasi selengkapnya, lihat Peringkat risiko.

T. Apa tujuan penggunaan model?

Tujuan penggunaan model menjelaskan bagaimana Anda harus menggunakan model dalam aplikasi produksi Anda. Ini melampaui persyaratan teknis seperti jenis instance yang harus Anda gunakan model dan sebagai gantinya mengacu pada jenis aplikasi yang akan dibuat dengan model, skenario di mana Anda dapat mengharapkan kinerja yang wajar dari model, atau jenis data untuk digunakan dengan model. Kami merekomendasikan untuk memberikan informasi ini dalam kartu model untuk tata kelola model yang lebih baik. Anda dapat menentukan jenis spesifikasi model di bidang penggunaan yang dimaksudkan dan memastikan bahwa pengembang model dan konsumen mengikuti spesifikasi ini saat melatih dan menerapkan model mereka. Untuk informasi selengkapnya, lihat Penggunaan model yang dimaksudkan.

T. Apakah SageMaker autopopulate informasi dalam kartu model saya?

Saat Anda menggunakan SageMaker Python SDK atau AWS konsol untuk membuat kartu model Anda, SageMaker otomatis mengisi detail tentang model SageMaker terlatih Anda di kartu. Ini termasuk detail tentang bagaimana model dilatih bersama dengan semua detail model yang dikembalikan oleh panggilan `describe-model` API.

Q. Dapatkah saya menyesuaikan kartu model?

Kartu SageMaker Model Amazon memiliki struktur yang ditentukan untuk mereka yang tidak dapat dimodifikasi. Struktur ini memberi Anda panduan tentang informasi apa yang harus ditangkap dalam kartu model. Meskipun Anda tidak dapat mengubah struktur kartu model, ada beberapa fleksibilitas yang diperkenalkan melalui properti khusus di bagian Informasi tambahan pada kartu model.

T. Dapatkah saya mengedit kartu model setelah dibuat?

Kartu model memiliki versi yang terkait dengannya. Versi model yang diberikan tidak dapat diubah di semua atribut selain status kartu model. Jika Anda membuat perubahan lain pada kartu model, seperti metrik evaluasi, deskripsi, atau penggunaan yang dimaksudkan, SageMaker buat versi baru kartu model untuk mencerminkan informasi yang diperbarui. Ini untuk memastikan bahwa kartu model, setelah dibuat, tidak dapat dirusak.

T. Dapatkah saya membuat kartu model untuk model yang tidak dilatih menggunakan? SageMaker

J: Ya. Anda dapat membuat kartu model untuk model yang tidak dilatih SageMaker, tetapi tidak ada informasi yang secara otomatis terisi dalam kartu. Anda harus memberikan semua informasi yang diperlukan dalam kartu model untuk SageMaker non-model.

T. Dapatkah saya mengekspor atau berbagi kartu model?

J: Ya. Anda dapat mengekspor setiap versi kartu model ke PDF, mengunduh, dan membagikannya.

T. Apakah saya perlu mendaftarkan model saya di Registry Model untuk menggunakan kartu model?

A: Tidak. Anda dapat menggunakan kartu model secara independen dari Registry Model.

Q. Apa perbedaan antara kartu model dan Model Registry?

J: Kartu model dimaksudkan untuk memberi organisasi mekanisme untuk mendokumentasikan sebanyak mungkin detail tentang model mereka sesuka mereka dengan mengikuti SageMaker panduan preskriptif bersama dengan memberikan informasi khusus mereka sendiri. Anda dapat memperkenalkan kartu model di awal proses ML dan menggunakannya untuk menentukan masalah bisnis yang harus dipecahkan oleh model dan pertimbangan apa pun untuk dipikirkan saat menggunakan model. Setelah model dilatih, Anda dapat mengisi kartu model yang terkait dengan model itu dengan informasi tentang model dan bagaimana itu dilatih. Kartu model dikaitkan dengan model dan tidak dapat diubah setelah dikaitkan dengan model. Ini memastikan bahwa kartu model adalah satu-satunya sumber kebenaran untuk semua informasi yang terkait dengan model, termasuk bagaimana kartu itu dilatih dan bagaimana seharusnya digunakan.

Model Registry adalah katalog yang menyimpan metadata tentang model Anda. Setiap entri dalam registri model sesuai dengan versi model yang unik. Versi model itu berisi informasi tentang model seperti tempat artefak model disimpan di Amazon S3, wadah apa yang diperlukan untuk menerapkan model, dan metadata khusus yang harus dilampirkan ke model.

Q. Apakah versi kartu model terkait dengan versi model di Registry Model?

A: Versi kartu model dan versi model adalah entitas yang berbeda di SageMaker. Setiap pembaruan ke kartu model menghasilkan versi baru kartu itu. Versi model sesuai dengan model yang dilatih secara bertahap yang terdaftar di Registri Model. Versi kartu model dapat ditautkan ke versi model tertentu di Registri Model melalui bidang ID model di kartu model, tetapi ini tidak diperlukan.

Q. Apakah kartu model terintegrasi dengan SageMaker Model Monitor?

A: Tidak. Anda dapat mengunggah metrik kinerja yang dihitung oleh SageMaker Model Monitor ke kartu model dengan mengunggah file metrik ke Amazon S3 dan menautkannya ke kartu, tetapi tidak ada integrasi asli antara Monitor Model dan kartu model. Dasbor model terintegrasi dengan Model Monitor. Untuk informasi selengkapnya tentang dasbor model, lihat Dasbor SageMaker Model Amazon.

Dasbor SageMaker Model Amazon

Dasbor SageMaker Model Amazon adalah portal terpusat, dapat diakses dari SageMaker konsol, tempat Anda dapat melihat, mencari, dan menjelajahi semua model di akun Anda. Anda dapat melacak model mana yang digunakan untuk inferensi dan apakah model tersebut digunakan dalam pekerjaan transformasi batch atau di-host di titik akhir. Jika Anda mengatur monitor dengan Amazon SageMaker Model Monitor, Anda juga dapat melacak kinerja model Anda saat mereka membuat prediksi real-time pada data langsung. Anda dapat menggunakan dasbor untuk menemukan model yang melanggar ambang batas yang Anda tetapkan untuk kualitas data, kualitas model, bias, dan kemampuan penjelasan. Presentasi komprehensif dasbor dari semua hasil monitor Anda membantu Anda dengan cepat mengidentifikasi model yang tidak memiliki metrik ini dikonfigurasi.

Dasbor Model mengumpulkan informasi terkait model dari beberapa fitur. SageMaker Selain layanan yang disediakan di Model Monitor, Anda dapat melihat kartu model, memvisualisasikan garis keturunan alur kerja, dan melacak kinerja titik akhir Anda. Anda tidak perlu lagi memilah-milah log, kueri di buku catatan, atau mengakses AWS layanan lain untuk mengumpulkan data yang Anda butuhkan. Dengan pengalaman pengguna yang kohesif dan integrasi ke dalam layanan yang ada, SageMaker Dasbor Model menyediakan solusi tata kelola out-of-the-box model untuk membantu Anda memastikan cakupan kualitas di semua model Anda.

Prasyarat

Untuk menggunakan Dasbor Model, Anda harus memiliki satu atau lebih model di akun Anda. Anda dapat melatih model menggunakan Amazon SageMaker atau mengimpor model yang telah Anda latih di tempat lain. Untuk membuat model SageMaker, Anda dapat menggunakan `CreateModel` API. Untuk informasi selengkapnya, lihat CreateModel. Anda juga dapat menggunakan lingkungan SageMaker ML-disediakan, seperti Amazon SageMaker Studio Classic, yang menyediakan templat proyek yang menyiapkan pelatihan model dan penerapan untuk Anda. Untuk informasi tentang cara memulai Studio Classic, lihat Amazon SageMaker Studio Classic.

Meskipun ini bukan prasyarat wajib, pelanggan mendapatkan nilai maksimal dari dasbor jika mereka menyiapkan pekerjaan pemantauan model menggunakan SageMaker Model Monitor untuk model yang digunakan ke titik akhir. Untuk prasyarat dan petunjuk tentang cara menggunakan SageMaker Model Monitor, lihat. Memantau data dan kualitas model

elemen Dasbor Model

Tampilan Dasbor Model mengekstrak detail tingkat tinggi dari setiap model untuk memberikan ringkasan komprehensif dari setiap model di akun Anda. Jika model Anda digunakan untuk inferensi, dasbor membantu Anda melacak kinerja model dan titik akhir Anda secara real time.

Detail penting untuk disorot di halaman ini meliputi:

- Peringkat risiko: Parameter yang ditentukan pengguna dari kartu model dengan nilai rendah, sedang, atau tinggi. Peringkat risiko kartu model adalah ukuran kategoris dari dampak bisnis dari prediksi model. Model digunakan untuk berbagai aplikasi bisnis, yang masing-masing mengasumsikan tingkat risiko yang berbeda. Misalnya, salah mendeteksi serangan cyber memiliki dampak bisnis yang jauh lebih besar daripada salah mengkategorikan email. Jika Anda tidak mengetahui risiko model, Anda dapat mengaturnya ke tidak diketahui. Untuk informasi tentang Kartu SageMaker Model Amazon, lihat Kartu Model.
- Peringatan Model Monitor: Peringatan Model Monitor adalah fokus utama Dasbor Model, dan meninjau dokumentasi yang ada pada berbagai monitor yang disediakan oleh SageMaker adalah cara yang membantu untuk memulai. Untuk penjelasan mendalam tentang fitur SageMaker Model Monitor dan contoh notebook, lihat. Memantau data dan kualitas model

  Dasbor Model menampilkan nilai status Monitor Model berdasarkan jenis monitor berikut:

  - Kualitas Data: Membandingkan data langsung dengan data pelatihan. Jika mereka menyimpang, kesimpulan model Anda mungkin tidak lagi akurat. Untuk detail tambahan tentang monitor Kualitas Data, lihatPantau kualitas data.
  - Kualitas Model: Membandingkan prediksi yang dibuat model dengan label Ground Truth aktual yang coba diprediksi oleh model. Untuk detail tambahan tentang monitor Kualitas Model, lihatMonitor kualitas model.
  - Bias Drift: Membandingkan distribusi data langsung dengan data pelatihan, yang juga dapat menyebabkan prediksi yang tidak akurat. Untuk detail tambahan tentang monitor Bias Drift, lihatMonitor Bias Drift untuk Model dalam Produksi.
  - Feature Attribution Drift: Juga dikenal sebagai penyimpangan penjelasan. Membandingkan peringkat relatif fitur Anda dalam data pelatihan versus data langsung, yang juga bisa menjadi hasil dari penyimpangan bias. Untuk detail tambahan tentang monitor Feature Attribution Drift, lihat. Monitor Fitur Atribusi Drift untuk Model dalam Produksi

  Setiap status Model Monitor adalah salah satu dari nilai berikut:

  - Tidak ada: Tidak ada monitor yang dijadwalkan

  - Tidak aktif: Monitor dijadwalkan, tetapi dinonaktifkan
  - OK: Monitor dijadwalkan dan aktif, dan belum menemukan jumlah pelanggaran yang diperlukan dalam eksekusi monitor model terbaru untuk meningkatkan peringatan
  - Waktu dan tanggal: Monitor aktif memunculkan peringatan pada waktu dan tanggal yang ditentukan
- Titik akhir: Titik akhir yang menjadi tuan rumah model Anda untuk inferensi waktu nyata. Dalam Dasbor Model, Anda dapat memilih kolom titik akhir untuk melihat metrik kinerja seperti CPU, GPU, disk, dan pemanfaatan memori titik akhir Anda secara real time untuk membantu Anda melacak kinerja instans komputasi Anda.
- Pekerjaan transformasi Batch: Pekerjaan transformasi batch terbaru yang dijalankan menggunakan model ini. Kolom ini membantu Anda menentukan apakah model digunakan secara aktif untuk inferensi batch.
- Detail model: Setiap entri di dasbor menautkan ke halaman detail model tempat Anda dapat menyelam lebih dalam ke model individual. Anda dapat mengakses grafik garis keturunan model, yang memvisualisasikan alur kerja dari persiapan data hingga penerapan, dan metadata untuk setiap langkah. Anda juga dapat membuat dan melihat kartu model, meninjau detail dan riwayat peringatan, menilai kinerja titik akhir waktu nyata Anda, dan mengakses detail terkait infrastruktur lainnya.

Lihat jadwal dan peringatan Model Monitor

Menggunakan Python SDK, Anda dapat membuat monitor model untuk kualitas data, kualitas model, penyimpangan bias, atau penyimpangan atribusi fitur. Untuk informasi selengkapnya tentang menggunakan SageMaker Model Monitor, lihatMemantau data dan kualitas model. Dasbor Model mengisi informasi dari semua monitor yang Anda buat di semua model di akun Anda. Anda dapat melacak status setiap monitor, yang menunjukkan apakah monitor Anda berjalan seperti yang diharapkan atau gagal karena kesalahan internal. Anda juga dapat mengaktifkan atau menonaktifkan monitor apa pun di halaman detail model itu sendiri. Untuk petunjuk cara melihat monitor terjadwal untuk model, lihatMelihat monitor terjadwal. Untuk petunjuk cara mengaktifkan atau menonaktifkan monitor model, lihatMengaktifkan atau menonaktifkan monitor model.

Monitor model yang dikonfigurasi dengan benar dan berjalan secara aktif dapat meningkatkan peringatan, dalam hal ini eksekusi pemantauan menghasilkan laporan pelanggaran. Untuk detail tentang cara kerja lansiran dan cara melihat hasil peringatan, riwayat, dan tautan ke laporan lowongan untuk debug, lihat. Lihat dan edit peringatan

Melihat monitor terjadwal

Untuk melihat monitor terjadwal model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker .
2. Pilih Tata Kelola di panel kiri.
3. Pilih Dasbor Model.
4. Di bagian Model Dasbor Model, pilih nama model monitor terjadwal yang ingin Anda lihat.
5. Lihat monitor terjadwal di bagian Jadwal monitor. Anda dapat meninjau status untuk setiap monitor di kolom Jadwal status, yang merupakan salah satu nilai berikut:
   - Gagal: Jadwal pemantauan gagal karena masalah dengan konfigurasi atau pengaturan (seperti izin pengguna yang salah).
   - Tertunda: Monitor sedang dalam proses menjadi dijadwalkan.
   - Berhenti: Jadwal dihentikan oleh pengguna.
   - Dijadwalkan: Jadwal dibuat dan berjalan pada frekuensi yang Anda tentukan.

Mengaktifkan atau menonaktifkan monitor model

Untuk mengaktifkan atau menonaktifkan monitor model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker .
2. Pilih Tata Kelola di panel kiri.
3. Pilih Dasbor Model.
4. Di bagian Model Dasbor Model, pilih nama model peringatan yang ingin Anda ubah.
5. Pilih kotak radio di samping jadwal monitor peringatan yang ingin Anda ubah.
6. (opsional) Pilih Nonaktifkan jadwal monitor jika Anda ingin menonaktifkan jadwal monitor Anda.
7. (opsional) Pilih Aktifkan jadwal monitor jika Anda ingin mengaktifkan jadwal monitor Anda.

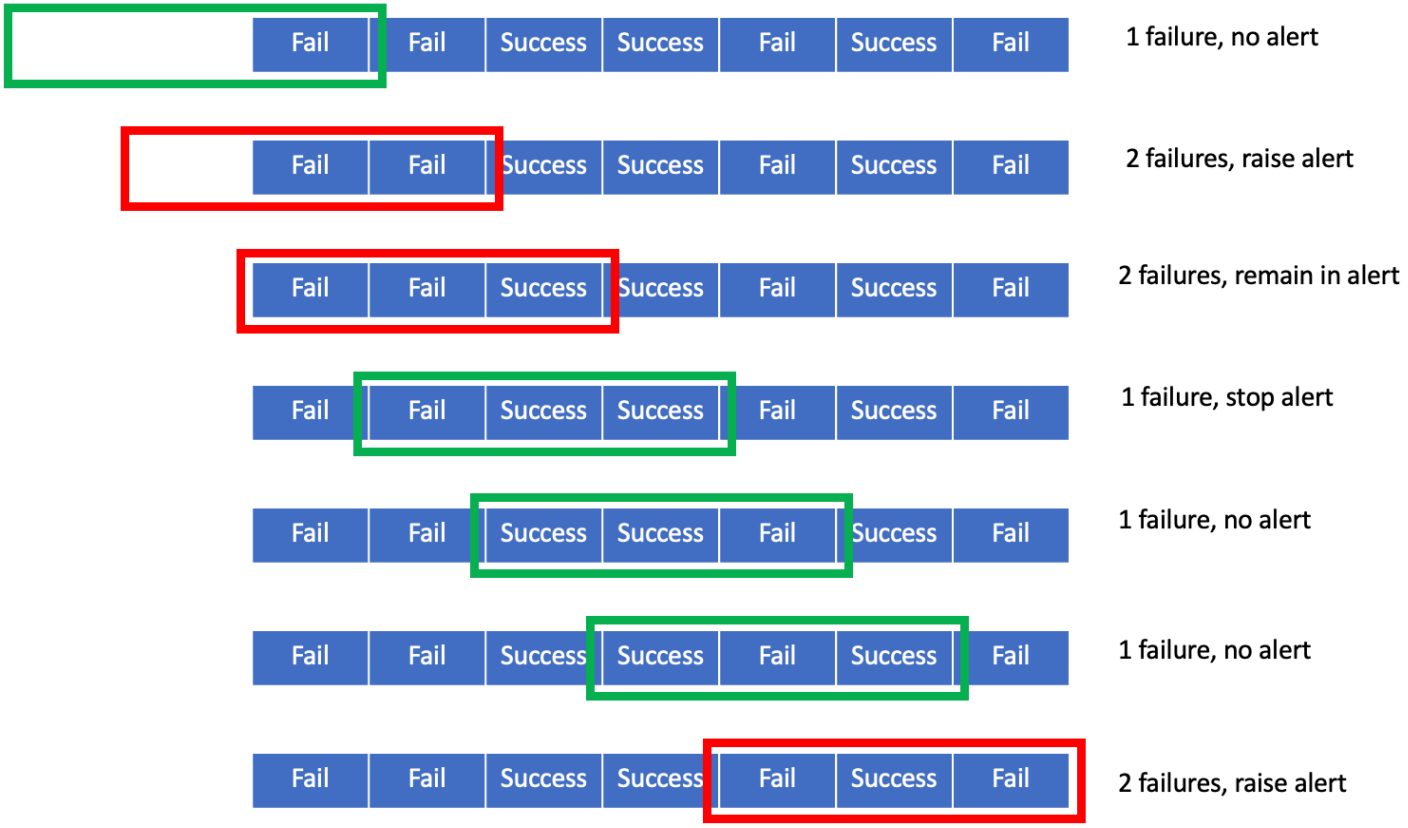
Lihat dan edit peringatan

Dasbor Model menampilkan peringatan yang Anda konfigurasikan di Amazon CloudWatch. Anda dapat memodifikasi kriteria peringatan di dalam dasbor itu sendiri. Kriteria peringatan tergantung pada dua parameter:

- Titik data untuk diperingatkan: Dalam periode evaluasi, berapa banyak kegagalan eksekusi yang menimbulkan peringatan.
- Periode evaluasi: Jumlah eksekusi pemantauan terbaru yang perlu dipertimbangkan saat mengevaluasi status peringatan.

Gambar berikut menunjukkan contoh skenario dari serangkaian eksekusi Model Monitor di mana kami menetapkan periode Evaluasi hipotetis 3 dan Datapoint ke nilai peringatan 2. Setelah setiap eksekusi pemantauan, jumlah kegagalan dihitung dalam periode Evaluasi 3. Jika jumlah kegagalan memenuhi atau melebihi Datapoint ke nilai peringatan 2, monitor memunculkan peringatan dan tetap dalam status waspada hingga jumlah kegagalan dalam periode Evaluasi menjadi kurang dari 2 pada iterasi berikutnya. Pada gambar, jendela evaluasi berwarna merah ketika monitor memunculkan peringatan atau tetap dalam status siaga, dan hijau sebaliknya.

Perhatikan bahwa meskipun ukuran jendela evaluasi belum mencapai periode Evaluasi 3, seperti yang ditunjukkan pada 2 baris pertama gambar, monitor masih menimbulkan peringatan jika jumlah kegagalan memenuhi atau melebihi Titik Data ke nilai peringatan 2.



Dalam halaman detail monitor, Anda dapat melihat riwayat peringatan, mengedit kriteria peringatan yang ada, dan melihat laporan pekerjaan untuk membantu Anda men-debug kegagalan peringatan. Untuk petunjuk tentang cara melihat riwayat peringatan atau laporan pekerjaan untuk eksekusi pemantauan yang gagal, lihatLihat riwayat peringatan atau laporan pekerjaan. Untuk petunjuk cara mengedit kriteria peringatan, lihatEdit kriteria peringatan.

Lihat riwayat peringatan atau laporan pekerjaan

Untuk melihat riwayat peringatan atau laporan pekerjaan dari eksekusi yang gagal, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker .
2. Pilih Tata Kelola di panel kiri.
3. Pilih Dasbor Model.
4. Di bagian Model Dasbor Model, pilih nama model riwayat peringatan yang ingin Anda lihat.
5. Di kolom Nama jadwal, pilih nama monitor dari riwayat peringatan yang ingin Anda lihat.
6. Untuk melihat riwayat peringatan, pilih tab Riwayat peringatan.
7. (opsional) Untuk melihat laporan pekerjaan pemantauan eksekusi, selesaikan langkah-langkah berikut:

    1. Di tab Riwayat peringatan, pilih Lihat eksekusi untuk peringatan yang ingin Anda selidiki.
    2. Dalam tabel Riwayat eksekusi, pilih Lihat laporan eksekusi pemantauan yang ingin Anda selidiki.

        Laporan menampilkan informasi berikut:

        - Fitur: Fitur ML yang ditentukan pengguna dipantau
        - Kendala: Pemeriksaan spesifik di dalam monitor
        - Rincian pelanggaran: Informasi tentang mengapa kendala dilanggar

Edit kriteria peringatan

Untuk mengedit peringatan di Dasboard Model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker .
2. Pilih Tata Kelola di panel kiri.
3. Pilih Dasbor Model.

4. Di bagian Model Dasbor Model, pilih nama model peringatan yang ingin Anda ubah.
5. Pilih kotak radio di samping jadwal monitor peringatan yang ingin Anda ubah.
6. Pilih Edit Peringatan di bagian Jadwal monitor.
7. (opsional) Ubah Datapoint menjadi peringatan jika Anda ingin mengubah jumlah kegagalan dalam periode Evaluasi yang memulai peringatan.
8. (opsional) Ubah periode Evaluasi jika Anda ingin mengubah jumlah eksekusi pemantauan terbaru yang perlu dipertimbangkan saat mengevaluasi status peringatan.

Lihat grafik garis keturunan model

Saat Anda melatih model, Amazon SageMaker membuat visualisasi seluruh alur kerja ML Anda mulai dari persiapan data hingga penerapan. Visualisasi ini disebut grafik garis keturunan model dan menggunakan entitas untuk mewakili langkah individual dalam alur kerja Anda. Misalnya, grafik garis keturunan model dasar mungkin memiliki entitas yang mewakili set pelatihan Anda, yang terkait dengan entitas yang mewakili pekerjaan pelatihan Anda, yang terkait dengan entitas lain yang mewakili model Anda.

Selain itu, grafik menyimpan informasi tentang setiap langkah dalam alur kerja Anda. Dengan informasi ini, Anda dapat membuat ulang langkah apa pun dalam alur kerja atau melacak model dan garis keturunan kumpulan data. Misalnya, SageMaker Lineage menyimpan URI S3 dari sumber data input Anda dengan setiap pekerjaan sehingga Anda dapat melakukan analisis lebih lanjut dari sumber data untuk verifikasi kepatuhan.

Meskipun grafik garis keturunan model dapat membantu Anda melihat langkah-langkah dalam alur kerja individual, ada banyak kemampuan lain yang dapat Anda manfaatkan menggunakan SDK. AWS Misalnya, dengan AWS SDK Anda dapat membuat atau menanyakan entitas Anda. Untuk informasi selengkapnya tentang rangkaian lengkap fitur di SageMaker Lineage dan contoh buku catatan, lihat. Pelacakan SageMaker Silsilah Amazon

Pengantar entitas

Amazon SageMaker secara otomatis membuat entitas pelacakan untuk SageMaker pekerjaan, model, paket model, dan titik akhir jika data tersedia. Untuk alur kerja dasar, misalkan Anda melatih model menggunakan kumpulan data. SageMaker secara otomatis menghasilkan grafik garis keturunan dengan tiga entitas:

- Dataset: Jenis artefak, yang merupakan entitas yang mewakili objek atau data yang dapat dialamatkan URI. Artefak umumnya berupa input atau output ke komponen percobaan atau tindakan.
- TrainingJob: Jenis komponen percobaan, yang merupakan entitas yang mewakili pemrosesan, pelatihan, dan transformasi pekerjaan.
- Model: Jenis artefak lain. Seperti artefak Dataset, Model adalah objek yang dapat dialamatkan URI. Dalam hal ini, ini adalah output dari komponen TrainingJobpercobaan.

Grafik silsilah model Anda berkembang dengan cepat jika Anda menambahkan langkah tambahan ke alur kerja Anda, seperti pra-pemrosesan data atau pasca-pemrosesan, jika Anda menerapkan model Anda ke titik akhir, atau jika Anda menyertakan model Anda dalam paket model, di antara banyak kemungkinan lainnya. Untuk daftar lengkap SageMaker entitas, lihatPelacakan SageMaker Silsilah Amazon.

Properti entitas

Setiap node dalam grafik menampilkan tipe entitas, tetapi Anda dapat memilih elipsis vertikal di sebelah kanan tipe entitas untuk melihat detail spesifik yang terkait dengan alur kerja Anda. Dalam grafik garis keturunan barebone kami sebelumnya, Anda dapat memilih elipsis vertikal di sebelah untuk melihat nilai spesifik DataSetuntuk properti berikut (umum untuk semua entitas artefak):

- Nama: Nama dataset Anda.
- URI Sumber: Lokasi Amazon S3 dari kumpulan data Anda.

Untuk `TrainingJob` entitas, Anda dapat melihat nilai spesifik untuk properti berikut (umum untuk semua `TrialComponent` entitas):

- Nama: Nama pekerjaan pelatihan.
- Job ARN: Amazon Resource Name (ARN) dari tugas pelatihan Anda.

Untuk entitas Model, Anda melihat properti yang sama seperti yang terdaftar DataSetkarena keduanya adalah entitas artefak. Untuk daftar entitas dan properti terkaitnya, lihatEntitas Pelacakan Silsilah.

Kueri entitas

Amazon SageMaker secara otomatis menghasilkan grafik entitas garis keturunan saat Anda menggunakannya. Namun jika Anda menjalankan banyak iterasi eksperimen dan tidak ingin melihat setiap grafik garis keturunan, AWS SDK dapat membantu Anda melakukan kueri di semua alur kerja Anda. Misalnya, Anda dapat menanyakan entitas silsilah Anda untuk semua pekerjaan pemrosesan yang menggunakan titik akhir. Atau, Anda dapat melihat semua jalur hilir yang menggunakan artefak. Untuk daftar semua kueri yang dapat Anda lakukan, lihatMeminta Entitas Silsilah.

Lihat grafik garis keturunan model

Untuk melihat grafik garis keturunan untuk model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker .
2. Pilih Tata Kelola di panel kiri.
3. Pilih Dasbor Model.
4. Di bagian Model Dasbor Model, pilih nama model grafik garis keturunan yang ingin Anda lihat.
5. Pilih Lihat silsilah di bagian Ikhtisar Model.

Melihat Status Titik Akhir

Jika Anda ingin menggunakan model terlatih untuk melakukan inferensi pada data langsung, Anda menerapkan model Anda ke titik akhir waktu nyata. Untuk memastikan latensi prediksi yang sesuai, Anda ingin memastikan instance yang meng-host model Anda berjalan secara efisien. Fitur pemantauan titik akhir Model Dashboard menampilkan informasi real-time tentang konfigurasi titik akhir Anda dan membantu Anda melacak kinerja titik akhir dengan metrik.

Pengaturan monitor

Dasbor Model menautkan ke halaman detail SageMaker titik akhir yang ada yang menampilkan grafik metrik real-time yang dapat Anda pilih di Amazon. CloudWatch Di dasbor Anda, Anda dapat melacak metrik ini karena titik akhir Anda menangani permintaan inferensi waktu nyata. Beberapa metrik yang dapat Anda pilih adalah sebagai berikut:

- `CpuUtilization`: Jumlah dari setiap pemanfaatan inti CPU individu, dengan masing-masing berkisar dari 0% - 100%.
- `MemoryUtilization`: Persentase memori yang digunakan oleh kontainer pada sebuah instance, mulai dari 0% — 100%.

- `DiskUtilization`: Persentase ruang disk yang digunakan oleh kontainer pada instans, mulai dari 0% — 100%.

Untuk daftar lengkap metrik yang dapat Anda lihat secara real time, lihatPantau Amazon SageMaker dengan Amazon CloudWatch.

Pengaturan runtime

Amazon SageMaker mendukung penskalaan otomatis (penskalaan otomatis) untuk model yang Anda hosting. Penskalaan otomatis secara dinamis menyesuaikan jumlah instance yang disediakan untuk model sebagai respons terhadap perubahan beban kerja Anda. Saat beban kerja meningkat, penskalaan otomatis menghadirkan lebih banyak instance online. Ketika beban kerja berkurang, penskalaan otomatis menghapus instans yang tidak perlu sehingga Anda tidak membayar untuk instans disediakan yang tidak Anda gunakan. Anda dapat mengkustomisasi pengaturan runtime berikut di Dasboard Model:

- Perbarui bobot: Ubah jumlah beban kerja yang ditetapkan untuk setiap instance dengan pembobotan numerik. Untuk informasi selengkapnya tentang pembobotan instans selama penskalaan otomatis, lihat Mengonfigurasi pembobotan instans untuk Auto Scaling Amazon EC2.
- Perbarui jumlah instans: Ubah jumlah total instance yang dapat melayani beban kerja Anda saat meningkat.

Untuk informasi selengkapnya tentang setelan runtime endpoint, lihat. CreateEndpointConfig

Pengaturan konfigurasi titik akhir

Pengaturan konfigurasi titik akhir menampilkan pengaturan yang Anda tentukan saat membuat titik akhir. Pengaturan ini menginformasikan sumber daya SageMaker mana yang akan disediakan untuk titik akhir Anda. Beberapa pengaturan yang disertakan adalah sebagai berikut:

- Pengambilan data: Anda dapat memilih untuk menangkap informasi tentang input dan output titik akhir Anda. Misalnya, Anda mungkin ingin mencicipi lalu lintas masuk untuk melihat apakah hasilnya berkorelasi dengan data pelatihan. Anda dapat menyesuaikan frekuensi pengambilan sampel, format data yang disimpan, dan lokasi Amazon S3 dari data yang disimpan. Untuk informasi selengkapnya tentang penyiapan konfigurasi pengambilan data, lihatTangkapan Data.
- Varian produksi: Lihat diskusi sebelumnya di pengaturan Runtime.
- Konfigurasi pemanggilan asinkron: Jika titik akhir Anda asinkron, bagian ini mencakup jumlah maksimum permintaan bersamaan yang dikirim oleh klien ke wadah model, lokasi Amazon S3

dari notifikasi keberhasilan dan kegagalan Anda, dan lokasi keluaran keluaran titik akhir Anda. SageMaker Untuk informasi selengkapnya tentang output asinkron, lihat. Membuat, memanggil, dan memperbarui Endpoint Asynchronous

- Kunci enkripsi: Anda dapat memasukkan kunci enkripsi Anda jika Anda ingin mengenkripsi output Anda.

Untuk informasi selengkapnya tentang pengaturan konfigurasi titik akhir, lihat CreateEndpointConfig.

Melihat status dan konfigurasi untuk titik akhir

Untuk melihat status dan konfigurasi titik akhir model, selesaikan langkah-langkah berikut:

1. Buka konsol SageMaker .
2. Pilih Tata Kelola di panel kiri.
3. Pilih Dasbor Model.
4. Di bagian Model Dasbor Model, pilih nama model titik akhir yang ingin Anda lihat.
5. Pilih nama titik akhir di bagian Endpoints.

FAQ Dasbor Model

Lihat topik FAQ berikut untuk jawaban atas pertanyaan umum tentang Dasbor SageMaker Model Amazon.

Q. Apa itu Model Dashboard?

Dasbor SageMaker Model Amazon adalah repositori terpusat dari semua model yang dibuat di akun Anda. Model umumnya merupakan output dari pekerjaan SageMaker pelatihan, tetapi Anda juga dapat mengimpor model yang dilatih di tempat lain dan menghostingnya. SageMaker Model Dashboard menyediakan antarmuka tunggal untuk administrator TI, manajer risiko model, dan pemimpin bisnis untuk melacak semua model yang diterapkan dan data agregat dari berbagai AWS layanan untuk memberikan indikator tentang kinerja model Anda. Anda dapat melihat detail tentang titik akhir model, pekerjaan transformasi batch, dan pekerjaan pemantauan untuk wawasan tambahan tentang kinerja model. Tampilan visual dasbor membantu Anda dengan cepat mengidentifikasi model mana yang memiliki monitor yang hilang atau tidak aktif sehingga Anda dapat memastikan semua model diperiksa secara berkala untuk penyimpangan data, penyimpangan model, penyimpangan bias, dan penyimpangan atribusi fitur. Terakhir, akses siap dasbor ke detail model membantu Anda

menyelam lebih dalam sehingga Anda dapat mengakses log, informasi terkait infrastruktur, dan sumber daya untuk membantu Anda men-debug kegagalan pemantauan.

T. Apa prasyarat untuk menggunakan Model Dashboard?

Anda harus memiliki satu atau lebih model yang dibuat SageMaker, baik dilatih SageMaker atau dilatih secara eksternal. Meskipun ini bukan prasyarat wajib, Anda mendapatkan nilai paling banyak dari dasbor jika Anda menyiapkan pekerjaan pemantauan model melalui Amazon Model Monitor untuk SageMaker model yang diterapkan ke titik akhir.

Q. Siapa yang harus menggunakan Model Dashboard?

Manajer risiko model, praktisi ML, ilmuwan data, dan pemimpin bisnis bisa mendapatkan gambaran menyeluruh tentang model menggunakan Dasbor Model. Dasbor mengumpulkan dan menampilkan data dari Kartu SageMaker Model Amazon, Titik Akhir, dan layanan Monitor Model untuk menampilkan informasi berharga seperti metadata model dari kartu model dan registri model, titik akhir tempat model digunakan, dan wawasan dari pemantauan model.

Q. Bagaimana cara menggunakan Dasbor Model?

Dasbor Model tersedia di luar kotak dengan Amazon SageMaker dan tidak memerlukan konfigurasi sebelumnya. Namun, jika Anda telah menyiapkan pekerjaan pemantauan model menggunakan SageMaker Model Monitor dan Clarify, Anda menggunakan Amazon CloudWatch untuk mengonfigurasi peringatan yang menaikkan bendera di dasbor saat performa model menyimpang dari rentang yang dapat diterima. Anda dapat membuat dan menambahkan kartu model baru ke dasbor, dan melihat semua hasil pemantauan yang terkait dengan titik akhir. Dasbor Model saat ini tidak mendukung model lintas akun.

T. Apa itu Amazon SageMaker Model Monitor?

Dengan Amazon SageMaker Model Monitor, Anda dapat memilih data yang ingin Anda pantau dan analisis tanpa menulis kode apa pun. SageMaker Model Monitor memungkinkan Anda memilih data, seperti output prediksi, dari menu opsi dan menangkap metadata seperti stempel waktu, nama model, dan titik akhir sehingga Anda dapat menganalisis prediksi model. Anda dapat menentukan laju pengambilan sampel data sebagai persentase dari lalu lintas keseluruhan dalam kasus prediksi real-time volume tinggi. Data ini disimpan di bucket Amazon S3 Anda sendiri. Anda juga dapat mengenkripsi data ini, mengonfigurasi keamanan berbutir halus, menentukan kebijakan penyimpanan data, dan menerapkan mekanisme kontrol akses untuk akses aman.

T. Jenis monitor model apa yang SageMaker mendukung?

SageMaker Model Monitor menyediakan jenis monitor model berikut:

- Kualitas Data: Memantau penyimpangan dalam kualitas data.
- Kualitas Model: Pantau penyimpangan dalam metrik kualitas model, seperti akurasi.
- Bias Drift untuk Model dalam Produksi: Pantau bias dalam prediksi model Anda dengan membandingkan distribusi pelatihan dan data langsung.
- Penyimpangan Atribusi Fitur untuk Model dalam Produksi: Pantau penyimpangan atribusi fitur dengan membandingkan peringkat relatif fitur dalam pelatihan dan data langsung.

T. Metode inferensi apa yang didukung SageMaker Model Monitor?

Model Monitor saat ini mendukung titik akhir yang menampung satu model untuk inferensi waktu nyata dan tidak mendukung pemantauan titik akhir multi-model.

T. Bagaimana cara memulai dengan SageMaker Model Monitor?

Anda dapat menggunakan sumber daya berikut untuk memulai pemantauan model:

- Monitor kualitas data contoh notebook
- Model monitor kualitas contoh notebook
- Bias drift monitor contoh notebook
- Fitur atribusi drift monitor contoh notebook

Untuk contoh pemantauan model lainnya, lihat GitHub repositori amazon-sagemaker-examples.

T. Bagaimana cara kerja Model Monitor?

Amazon SageMaker Model Monitor secara otomatis memonitor model pembelajaran mesin dalam produksi, menggunakan aturan untuk mendeteksi penyimpangan dalam model Anda. Model Monitor memberi tahu Anda saat masalah kualitas muncul melalui peringatan. Untuk mempelajari selengkapnya, lihat Bagaimana Model Monitor Bekerja.

T. Kapan dan bagaimana Anda membawa wadah Anda sendiri (BYOC) untuk Model Monitor?

Model Monitor menghitung metrik model dan statistik hanya pada data tabular. Untuk kasus penggunaan selain kumpulan data tabular, seperti gambar atau teks, Anda dapat membawa wadah

Anda sendiri (BYOC) untuk memantau data dan model Anda. Misalnya, Anda dapat menggunakan BYOC untuk memantau model klasifikasi gambar yang mengambil gambar sebagai input dan mengeluarkan label. Untuk mempelajari selengkapnya tentang kontrak kontainer, lihatBawa Kontainer Anda Sendiri.

T. Di mana saya dapat menemukan contoh BYOC untuk Model Monitor?

Anda dapat menemukan contoh BYOC yang bermanfaat di tautan berikut:

- Memantau data dan kualitas model
- GitHubcontoh repositori
- Bawa Kontainer Anda Sendiri
- Mendeteksi penyimpangan data di NLP menggunakan Monitor Model BYOC
- Mendeteksi dan menganalisis prediksi yang salah di CV

T. Bagaimana cara mengintegrasikan Model Monitor dengan SageMaker Pipelines?

Untuk detail tentang cara mengintegrasikan Model Monitor dan SageMaker Pipelines, lihat Amazon SageMaker Pipelines sekarang terintegrasi dengan SageMaker Model Monitor dan Clarify. SageMaker

Sebagai contoh, lihat contoh integrasi SageMaker Pipelines notebook dengan Model Monitor dan Clarify. GitHub

T. Apakah ada masalah kinerja yang digunakan? **`DataCapture`**

Saat dihidupkan, pengambilan data terjadi secara asinkron pada titik akhir. SageMaker Untuk mencegah dampak pada permintaan inferensi, `DataCapture` berhenti menangkap permintaan pada tingkat penggunaan disk yang tinggi. Disarankan Anda menjaga penggunaan disk Anda di bawah 75% untuk memastikan `DataCapture` terus menangkap permintaan.

Gunakan kontainer Docker untuk membuat model

Amazon SageMaker menggunakan kontainer Docker secara ekstensif untuk tugas build dan runtime. SageMaker menyediakan gambar Docker pra-bangun untuk algoritme bawaannya dan kerangka kerja pembelajaran mendalam yang didukung yang digunakan untuk pelatihan dan inferensi. Dengan menggunakan kontainer, Anda dapat melatih algoritme pembelajaran mesin dan menerapkan model dengan cepat dan andal pada skala apa pun. Topik di bagian ini menunjukkan cara menerapkan kontainer ini untuk kasus penggunaan Anda sendiri. Untuk informasi tentang cara membawa kontainer Anda sendiri untuk digunakan dengan Amazon SageMaker Studio Classic, lihatBawa SageMaker gambar Anda sendiri.

Topik

Skenario untuk Menjalankan Skrip, Algoritma Pelatihan, atau Menerapkan Model dengan SageMaker

Amazon SageMaker selalu menggunakan kontainer Docker saat menjalankan skrip, algoritma pelatihan, dan menerapkan model. Tingkat keterlibatan Anda dengan kontainer tergantung pada kasus penggunaan Anda.

Gunakan kasus untuk menggunakan wadah Docker yang sudah dibuat sebelumnya dengan SageMaker

Pertimbangkan kasus penggunaan berikut saat menggunakan wadah dengan SageMaker:

- SageMaker Algoritma pra-dibangun - Gunakan gambar yang disertakan dengan algoritma bawaan. Lihat Menggunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih untuk informasi selengkapnya.
- Model kustom dengan SageMaker kontainer pra-bangun - Jika Anda melatih atau menerapkan model kustom, tetapi menggunakan kerangka kerja yang memiliki SageMaker wadah pra-bangun termasuk TensorFlow dan PyTorch, pilih salah satu opsi berikut:
  - Jika Anda tidak memerlukan paket khusus, dan wadah sudah menyertakan semua paket yang diperlukan: Gunakan gambar Docker pra-bangun yang terkait dengan kerangka kerja Anda. Untuk informasi selengkapnya, lihat Gunakan gambar SageMaker Docker yang dibuat sebelumnya.
  - Jika Anda memerlukan paket khusus yang diinstal ke salah satu kontainer yang sudah dibuat sebelumnya: Konfirmasikan bahwa image Docker yang sudah dibuat sebelumnya mengizinkan file requirements.txt, atau perluas wadah yang sudah dibuat sebelumnya berdasarkan kasus penggunaan berikut.

Kasus penggunaan untuk memperluas wadah Docker yang sudah dibuat sebelumnya

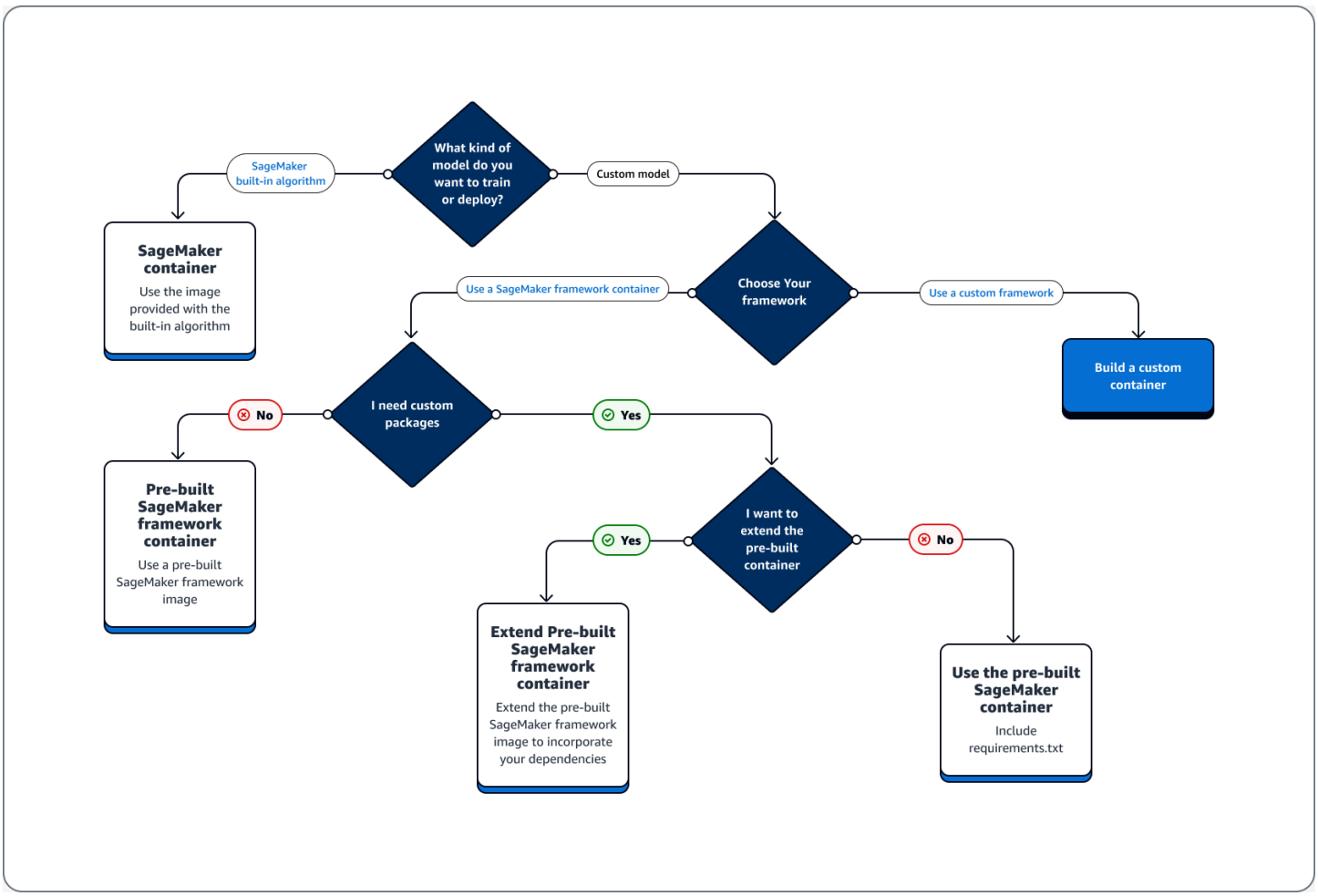
Berikut ini adalah kasus penggunaan untuk memperluas wadah Docker yang sudah dibuat sebelumnya:

- Anda tidak dapat mengimpor dependensi — Perluas image Docker pra-bangun yang terkait dengan kerangka kerja Anda. Untuk informasi selengkapnya, lihat Memperluas Kontainer Pra-dibangun.
- Anda tidak dapat mengimpor dependensi dalam wadah pra-bangun dan wadah pra-bangun mendukung requirements.txt - Tambahkan semua dependensi yang diperlukan di requirements.txt. Kerangka kerja berikut mendukung menggunakan requirements.txt.
  - TensorFlow
  - Chainer
  - Sci-kit belajar
  - PyTorch
  - Apache MxNet

Kasus penggunaan untuk membangun wadah Anda sendiri

Jika Anda membuat atau melatih model kustom dan memerlukan kerangka kerja khusus yang tidak memiliki gambar yang dibuat sebelumnya, buat wadah khusus.

Pohon keputusan berikut mengilustrasikan informasi dalam tiga daftar sebelumnya: Kasus penggunaan untuk menggunakan kontainer Docker yang sudah dibuat sebelumnya dengan SageMaker; Gunakan kasus untuk memperluas wadah Docker yang sudah dibuat sebelumnya; Kasus penggunaan untuk membangun wadah Anda sendiri.



Sebagai contoh kasus penggunaan pelanggan, untuk melatih dan menerapkan TensorFlow model, lihat bagian kasus Penggunaan sebelumnya untuk memutuskan wadah mana yang Anda butuhkan. Berikut ini adalah pertimbangan yang dapat Anda buat untuk memilih wadah Anda:

- TensorFlow Model adalah model khusus.
- Karena TensorFlow model akan dibangun dalam TensorFlow kerangka kerja, gunakan wadah kerangka kerja TensorFlow pra-bangun untuk melatih dan meng-host model.

- Jika Anda memerlukan paket khusus baik dalam skrip entrypoint atau skrip inferensi Anda, perluas wadah pra-bangun atau gunakan file requirements.txt untuk menginstal dependensi saat runtime.

Setelah Anda menentukan jenis wadah yang Anda butuhkan, informasi berikut memberikan detail tentang opsi yang tercantum sebelumnya.

- Gunakan SageMaker algoritma atau kerangka kerja bawaan. Untuk sebagian besar kasus penggunaan, Anda dapat menggunakan algoritme dan kerangka kerja bawaan tanpa mengkhawatirkan kontainer. Anda dapat melatih dan menerapkan algoritme ini dari SageMaker konsol, AWS Command Line Interface (AWS CLI), notebook Python, atau Amazon Python SDK. SageMaker Anda dapat melakukannya dengan menentukan algoritme atau versi kerangka kerja saat membuat Estimator Anda. Algoritma bawaan yang tersedia diperinci dan dijelaskan dalam topik. Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih Untuk informasi selengkapnya tentang kerangka kerja yang tersedia, lihatKerangka Kerja dan Bahasa. Untuk contoh cara melatih dan menerapkan algoritme bawaan menggunakan notebook Jupyter yang berjalan di instance SageMaker notebook, lihat topiknya. Memulai
- Gunakan gambar SageMaker kontainer yang sudah dibuat sebelumnya. Atau, Anda dapat menggunakan algoritme dan kerangka kerja bawaan menggunakan kontainer Docker. SageMaker menyediakan wadah untuk algoritme bawaan dan gambar Docker yang dibuat sebelumnya untuk beberapa kerangka kerja pembelajaran mesin yang paling umum, seperti Apache MXNet,,, dan Chainer. TensorFlow PyTorch Untuk daftar lengkap Gambar yang tersedia, lihat SageMaker Gambar Deep Learning Containers yang Tersedia. Ini juga mendukung perpustakaan pembelajaran mesin seperti scikit-learn dan SparkMl. Jika Anda menggunakan Amazon SageMaker Python SDK, Anda dapat menerapkan container dengan meneruskan URI kontainer lengkap ke kelas SDK masing-masing. SageMaker `Estimator` Untuk daftar lengkap kerangka kerja pembelajaran mendalam yang saat ini didukung oleh SageMaker, lihatPrebuiltSageMakerGambar Docker untuk Deep Learning. Untuk informasi tentang gambar kontainer pra-bangun scikit-learn dan SparkMl, lihat. Amazon bawaan SageMaker Gambar Docker untuk Scikit-learn dan Spark Untuk informasi selengkapnya tentang penggunaan framework dengan Amazon SageMaker Python SDK, lihat topiknya masing-masing di. Kerangka Kerja dan Bahasa Machine Learning
- Perluas gambar SageMaker kontainer yang sudah dibuat sebelumnya. Jika Anda ingin memperluas SageMaker algoritme pra-bangun atau memodelkan gambar Docker, Anda dapat memodifikasi SageMaker gambar untuk memenuhi kebutuhan Anda. Sebagai contoh, lihat Memperluas PyTorch kontainer kami.

- Menyesuaikan gambar kontainer yang ada: Jika Anda ingin mengadaptasi gambar kontainer yang sudah ada sebelumnya untuk dikerjakan SageMaker, Anda harus memodifikasi wadah Docker untuk mengaktifkan toolkit SageMaker Pelatihan atau Inferensi. Untuk contoh yang menunjukkan cara membuat kontainer Anda sendiri untuk melatih dan meng-host algoritme, lihat Membawa Algoritma R Anda Sendiri.

Dasar kontainer Docker

Docker adalah program yang melakukan virtualisasi tingkat sistem operasi untuk menginstal, mendistribusikan, dan mengelola perangkat lunak. Ini paket aplikasi dan dependensi mereka ke dalam wadah virtual yang menyediakan isolasi, portabilitas, dan keamanan. Dengan Docker, Anda dapat mengirimkan kode lebih cepat, menstandarisasi operasi aplikasi, memindahkan kode secara mulus, dan menghemat dengan meningkatkan pemanfaatan sumber daya. Untuk informasi umum selengkapnya tentang Docker, lihatIkhtisar Docker.

Informasi berikut menguraikan aspek paling signifikan dalam menggunakan kontainer Docker dengan Amazon SageMaker.

SageMaker Fungsi

SageMaker menggunakan kontainer Docker di backend untuk mengelola proses pelatihan dan inferensi. SageMaker abstrak jauh dari proses ini, sehingga terjadi secara otomatis ketika estimator digunakan. Meskipun Anda tidak perlu menggunakan kontainer Docker secara eksplisit SageMaker untuk sebagian besar kasus penggunaan, Anda dapat menggunakan kontainer Docker untuk memperluas dan menyesuaikan SageMaker fungsionalitas.

Kontainer dengan SageMaker Studio

SageMaker Studio berjalan dari wadah Docker dan menggunakannya untuk mengelola fungsionalitas. Akibatnya, Anda harus membuat container Docker Anda mengikuti langkah-langkah diBawa SageMaker gambar Anda sendiri.

Gunakan gambar SageMaker Docker yang dibuat sebelumnya

Amazon SageMaker menyediakan wadah untuk algoritme bawaannya dan gambar Docker yang dibuat sebelumnya untuk beberapa kerangka kerja pembelajaran mesin yang paling umum, seperti Apache MXNet,, dan Chainer. TensorFlow PyTorch Ini juga mendukung perpustakaan pembelajaran mesin seperti scikit-learn dan SparkMl.

Anda dapat menggunakan gambar-gambar ini dari instance SageMaker notebook atau SageMaker Studio Anda. Anda juga dapat memperluas SageMaker gambar pra-bangun untuk menyertakan pustaka dan fungsionalitas yang diperlukan. Topik berikut memberikan informasi tentang gambar yang tersedia dan cara menggunakannya.

Untuk jalur registri Docker dan parameter lainnya untuk setiap algoritme yang SageMaker disediakan Amazon dan Deep Learning Containers (DLC), lihat Docker Registry Paths dan Example Code.

> **Note**
>
> Untuk informasi tentang gambar Docker untuk mengembangkan solusi pembelajaran penguatan (RL) di SageMaker, lihat SageMaker RL Containers.

Topik

PrebuiltSageMakerGambar Docker untuk Deep Learning

AmazonSageMakermenyediakan image Docker bawaan yang mencakup kerangka kerja deep learning dan dependensi lain yang diperlukan untuk pelatihan dan inferensi. Untuk daftar lengkap image Docker bawaan yang dikelola olehSageMaker, lihatJalur Registri Docker dan Kode Contoh.

MenggunakanSageMakerPython

DenganSageMakerPython, Anda dapat melatih dan menerapkan model menggunakan kerangka pembelajaran mendalam yang populer ini. Untuk petunjuk tentang menginstal dan menggunakan SDK, lihatAmazonSageMakerPython. Tabel berikut mencantumkan kerangka kerja yang tersedia dan petunjuk tentang cara menggunakannya denganSageMakerPython:

Kerangka Kerja	Petunjuk
`TensorFlow`	MenggunakanTensorFlowdenganSageMakerPython

Kerangka Kerja	Petunjuk
`MXNet`	Menggunakan MXNet denganSageMakerPython
`PyTorch`	MenggunakanPyTorchdenganSageMakerPython
`Chainer`	Menggunakan Chainer denganSageMakerPython
`Hugging Face`	Menggunakan Memeluk Wajah denganSageMakerPython

Memperluas PrebuiltSageMakerGambar Docker

Anda dapat menyesuaikan wadah prebuilt ini atau memperluasnya untuk menangani persyaratan fungsional tambahan untuk algoritme atau model Anda yang dibuat sebelumnyaSageMakerGambar Docker tidak mendukung. Sebagai contoh, lihatMenyempurnakan dan menerapkan model BertopicSageMakerdengan skrip dan dataset Anda sendiri, dengan memperluas yang adaPyTorchkontainer.

Anda juga dapat menggunakan kontainer bawaan untuk menerapkan model atau model khusus Anda yang telah dilatih dalam kerangka kerja selainSageMaker. Untuk ikhtisar proses membawa artefak model terlatih keSageMakerdan hosting mereka di titik akhir, lihatBawa MXNet Pretrained Anda Sendiri atauTensorFlowModel ke AmazonSageMaker.

Amazon bawaan SageMaker Gambar Docker untuk Scikit-learn dan Spark

SageMaker menyediakan gambar Docker bawaan yang menginstal pustaka scikit-learn dan Spark ML. Pustaka ini juga menyertakan dependensi yang diperlukan untuk membangun gambar Docker yang kompatibel dengan SageMaker menggunakanAmazon SageMaker SDK Python. Dengan SDK, Anda dapat menggunakan scikit-learn untuk tugas pembelajaran mesin dan menggunakan Spark ML untuk membuat dan menyetel pipeline pembelajaran mesin. Untuk petunjuk tentang cara menginstal dan menggunakan SDK, lihatSageMaker SDK Python.

Menggunakan SageMakerSDK Python

Tabel berikut berisi tautan ke GitHub repositori dengan kode sumber untuk wadah scikit-learn dan Spark ML. Tabel ini juga berisi tautan ke instruksi yang menunjukkan cara menggunakan wadah ini dengan penaksir SDK Python untuk menjalankan algoritme pelatihan Anda sendiri dan menghosting model Anda sendiri.

Perpustakaan	Kode Sumber Gambar Docker Prebuilt	Petunjuk
scikit-belajar	SageMaker Wadah Scikit-learn	Menggunakan Scikit-Learn dengan Amazon SageMaker SDK Python
Spark ML	SageMaker Wadah Penyajian Spark ML	Dokumentasi SDK Python SparkMl

Untuk informasi selengkapnya dan tautan ke repositori github, lihatGunakan Scikit-Learn dengan Amazon SageMakerdanGunakan Penyajian SparkMl dengan Amazon SageMaker.

Menentukan Gambar Prebuilt Secara Manual

Jika Anda tidak menggunakan SageMaker Python SDK dan salah satu penaksirnya untuk mengelola wadah, Anda harus mengambil wadah bawaan yang relevan secara manual. The SageMakerGambar Docker bawaan disimpan di Amazon Elastic Container Registry (Amazon ECR). Anda dapat mendorong atau menariknya menggunakan alamat registri nama lengkap mereka. SageMaker menggunakan pola URL Gambar Docker berikut untuk scikit-learn dan Spark ML:

- `<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/sagemaker-scikit-learn:<SCIKIT-LEARN_VERSION>-cpu-py<PYTHON_VERSION>`

  Misalnya, `746614075791.dkr.ecr.us-west-1.amazonaws.com/sagemaker-scikit-learn:1.2-1-cpu-py3`
- `<ACCOUNT_ID>.dkr.ecr.<REGION_NAME>.amazonaws.com/sagemaker-sparkml-serving:<SPARK-ML_VERSION>`

  Misalnya, `341280168497.dkr.ecr.ca-central-1.amazonaws.com/sagemaker-sparkml-serving:2.4`

Untuk ID akun danAWSNama wilayah, lihatJalur Registri Docker dan Kode Contoh.

Menemukan Gambar yang Tersedia

Gunakan perintah berikut untuk mengetahui versi gambar yang tersedia. Misalnya, gunakan yang berikut ini untuk menemukan yang tersedia`sagemaker-sparkml-serving`gambar di`ca-central-1`Wilayah:

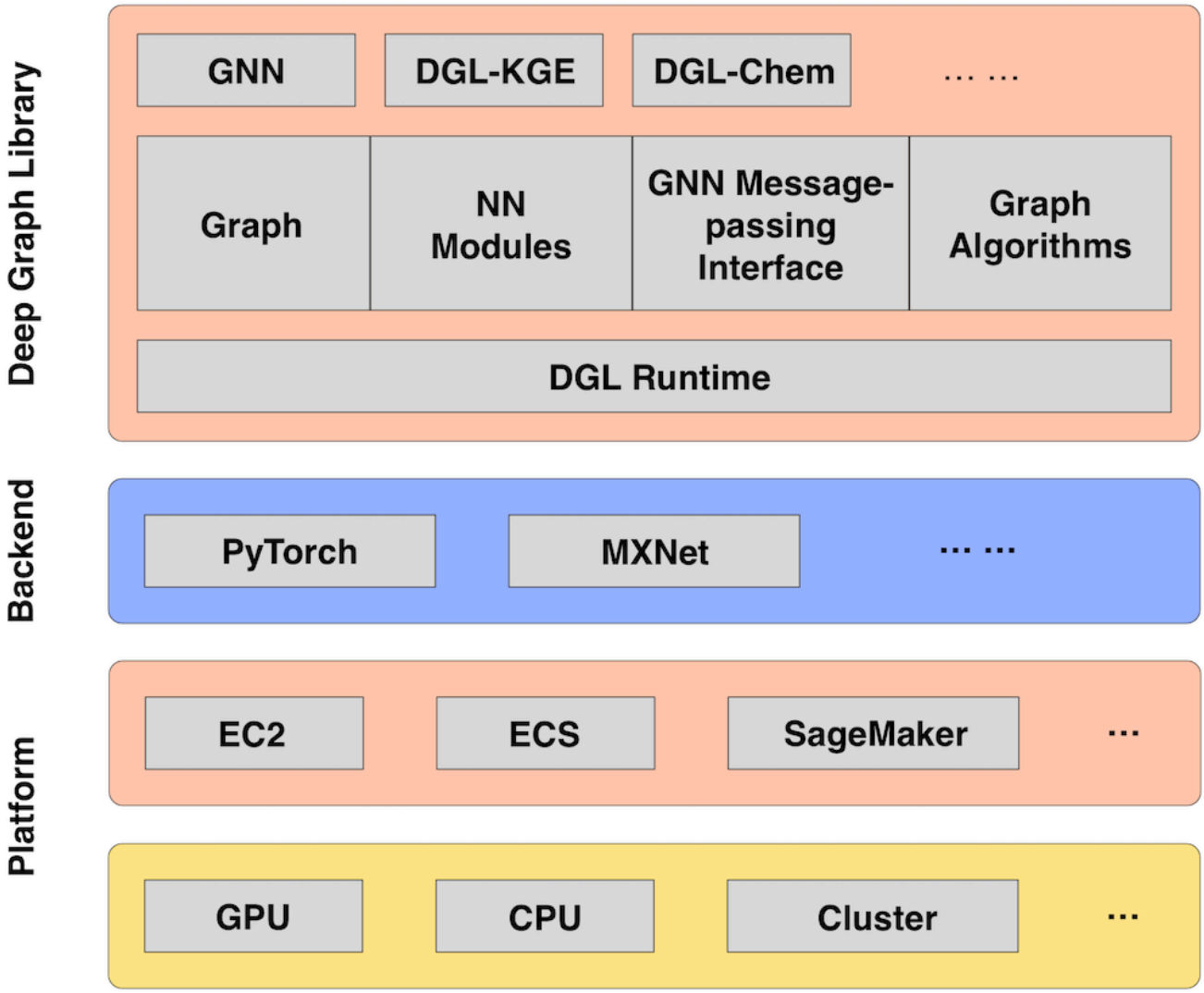
```
aws \
    ecr describe-images \
    --region ca-central-1 \
    --registry-id 341280168497 \
    --repository-name sagemaker-sparkml-serving
```

Melatih Jaringan Grafik Dalam

Dalam ikhtisar ini, Anda mempelajari cara memulai dengan jaringan grafik mendalam dengan menggunakan salah satu kontainer DGL di Amazon Elastic Container Registry (Amazon ECR). Anda juga dapat melihat tautan ke contoh praktis untuk jaringan grafik dalam.

Apa itu Jaringan Grafik Dalam?

Jaringan grafik dalam mengacu pada jenis jaringan saraf yang dilatih untuk memecahkan masalah grafik. Jaringan grafik dalam menggunakan kerangka pembelajaran mendalam yang mendasari seperti PyTorch atau MXNet. Potensi jaringan grafik dalam aplikasi AI praktis disorot dalam tutorial Amazon SageMaker untukPerpustakaan Grafik Dalam(DGL). Contoh untuk model pelatihan pada dataset grafik meliputi jejaring sosial, basis pengetahuan, biologi, dan kimia.



Gambar 1. Ekosistem DGL

Beberapa contoh disediakan menggunakan kontainer deep learning Amazon SageMaker yang telah dikonfigurasikan sebelumnya dengan DGL. Jika Anda memiliki modul khusus yang ingin Anda gunakan dengan DGL, Anda juga dapat membangun wadah Anda sendiri. Contohnya melibatkan heterograf, yang merupakan grafik yang memiliki beberapa jenis node dan tepi, dan menggambar berbagai aplikasi di berbagai bidang ilmiah yang berbeda, seperti bioinformatika dan analisis jejaring sosial. DGL menyediakan beragamgrafik implementasi jaringan saraf untuk berbagai jenis model. Beberapa sorotan meliputi:

- Grafik jaringan konvolusional (GCN)
- Grafik relasional jaringan konvolusional (R-GCN)

- Jaringan perhatian grafik (GAT)
- Model generatif dalam grafik (DGMG)
- Jaringan saraf pohon persimpangan (JTNN)

Mulai

DGL tersedia sebagai wadah pembelajaran mendalam di Amazon ECR. Anda dapat memilih wadah pembelajaran mendalam ketika Anda menulis fungsi estimator Anda di notebook Amazon SageMaker. Anda juga dapat membuat wadah kustom Anda sendiri dengan DGL dengan mengikutiBawa Kontainer Anda Sendiripanduan. Cara termudah untuk memulai dengan jaringan grafik dalam menggunakan salah satu kontainer DGL di Amazon ECR.

> **Note**
>
> Dukungan kerangka kerja backend terbatas pada PyTorch dan MXnet.

Penyiapan

Jika Anda menggunakan Amazon SageMaker Studio, Anda perlu mengkloning contoh repositori terlebih dahulu. Jika Anda menggunakan instance notebook, Anda dapat menemukan contoh dengan memilih ikon SageMaker di bawah toolbar kiri.

Untuk mengkloning Amazon SageMaker SDK dan contoh notebook repositori

1. DariJupyterLablihat di Amazon SageMaker, pergi kePerambandi bagian atas toolbar kiri. DariPanel peramban, Anda dapat melihatnavigasi barudi bagian atas panel.
2. Pilihikondi paling kanan untuk mengkloning repositori Git.
3. Tambahkan URL repositori:https://github.com/awslabs/amazon-sagemaker-examples.git
4. Jelajahi folder yang baru ditambahkan dan isinya. Contoh DGL disimpan dalamSagemaker-python-sdkfolder.

Menjalankan Contoh Pelatihan Jaringan Grafik

Untuk melatih jaringan grafik yang mendalam

1. DariJupyterLablihat di Amazon SageMaker, jelajahinotebook contohdan carilahDGLfolder. Beberapa file dapat disertakan untuk mendukung contoh. Periksa README untuk prasyarat apa pun.
2. Jalankan contoh notebook .ipynb.
3. Temukan fungsi estimator, dan perhatikan garis di mana ia menggunakan kontainer Amazon ECR untuk DGL dan jenis instans tertentu. Anda mungkin ingin memperbarui ini untuk menggunakan kontainer di Wilayah pilihan Anda.
4. Jalankan fungsi untuk meluncurkan instance dan menggunakan wadah DGL untuk melatih jaringan grafik. Biaya dikeluarkan untuk meluncurkan instance ini. Instance self-terminates ketika pelatihan selesai.

Contoh

Contoh pengetahuan grafik embedding (KGE) disediakan. Menggunakan dataset Freebase, basis pengetahuan fakta-fakta umum. Contoh kasus penggunaan adalah untuk grafik hubungan orang dan memprediksi kewarganegaraan mereka.

Contoh implementasi jaringan konvolusional grafik (GCN) menunjukkan bagaimana Anda dapat melatih jaringan grafik untuk memprediksi toksisitas. Sebuah dataset fisiologi, Tox21, memberikan pengukuran toksisitas untuk bagaimana zat mempengaruhi respons biologis.

Contoh GCN lain menunjukkan cara melatih jaringan grafik pada publikasi ilmiah bibliografi dataset, yang dikenal sebagai Cora. Anda dapat menggunakannya untuk menemukan hubungan antara penulis, topik, dan konferensi.

Contoh terakhir adalah sistem recommender untuk ulasan film. Menggunakan grafik convolutional matrix completion (GCMC) jaringan dilatih pada dataset MovieLens. Dataset ini terdiri dari judul film, genre, dan peringkat oleh pengguna.

Gunakan Deep Learning Container dengan DGL

Contoh berikut menggunakan kontainer deep learning yang telah dikonfigurasikan sebelumnya. Ini adalah yang paling mudah untuk dicoba karena bekerja di luar kotak di Amazon SageMaker.

- Klasifikasi semi-diawasi dari basis pengetahuan menggunakan GCN

Bawa Kontainer Anda Sendiri dengan DGL

Contoh berikut memungkinkan Anda untuk membawa wadah Anda sendiri (BYOC). BacaPanduan BYOCdan membiasakan diri dengan proses itu sebelum mencoba ini. Konfigurasi diperlukan.

- Prediksi properti molekul toksisitas menggunakan GCN
- Sistem Recommender untuk film menggunakan implementasi GCMC

Memperluas Kontainer Pra-dibangun

Jika SageMaker kontainer yang sudah dibuat sebelumnya tidak memenuhi semua kebutuhan Anda, Anda dapat memperluas gambar yang ada untuk mengakomodasi kebutuhan Anda. Bahkan jika ada dukungan langsung untuk lingkungan atau kerangka kerja Anda, Anda mungkin ingin menambahkan fungsionalitas tambahan atau mengkonfigurasi lingkungan kontainer Anda secara berbeda. Dengan memperluas gambar yang sudah dibuat sebelumnya, Anda dapat memanfaatkan pustaka dan pengaturan deep learning yang disertakan tanpa harus membuat gambar dari awal. Anda dapat memperluas wadah untuk menambahkan pustaka, memodifikasi pengaturan, dan menginstal dependensi tambahan.

Tutorial berikut menunjukkan cara memperluas SageMaker gambar yang sudah dibuat sebelumnya dan mempublikasikannya ke Amazon ECR.

Topik

- Persyaratan untuk Memperpanjang Kontainer Pra-dibangun
- Perluas SageMaker Kontainer untuk Menjalankan Script Python

Persyaratan untuk Memperpanjang Kontainer Pra-dibangun

Untuk memperluas SageMaker gambar yang sudah dibuat sebelumnya, Anda perlu mengatur variabel lingkungan berikut dalam Dockerfile Anda. Untuk informasi selengkapnya tentang variabel lingkungan dengan SageMaker kontainer, lihat GitHub repoSageMaker Training Toolkit.

- `SAGEMAKER_SUBMIT_DIRECTORY`: Direktori dalam wadah tempat skrip Python untuk pelatihan berada.
- `SAGEMAKER_PROGRAM`: Script Python yang harus dipanggil dan digunakan sebagai titik masuk untuk pelatihan.

Anda juga dapat menginstal pustaka tambahan dengan menyertakan yang berikut ini di Dockerfile Anda:

```
RUN pip install <library>
```

Tutorial berikut menunjukkan cara menggunakan variabel lingkungan ini.

Perluas SageMaker Kontainer untuk Menjalankan Script Python

Dalam tutorial ini, Anda belajar bagaimana untuk memperluas SageMaker PyTorch wadah dengan file Python yang menggunakan dataset CIFAR-10. Dengan memperluas SageMaker PyTorch wadah, Anda memanfaatkan solusi pelatihan yang ada yang dibuat untuk bekerja dengan SageMaker. Tutorial ini memperluas gambar pelatihan, tetapi langkah yang sama dapat diambil untuk memperluas gambar inferensi. Untuk daftar lengkap gambar yang tersedia, lihat Gambar Deep Learning Containers yang Tersedia.

Untuk menjalankan model pelatihan Anda sendiri menggunakan SageMaker kontainer, buat container Docker melalui instance SageMaker Notebook.

Langkah 1: Buat Instans SageMaker Notebook

1. Buka konsol SageMaker .
2. Di panel navigasi kiri, pilih Notebook, pilih Instans notebook, lalu pilih Buat instance notebook.
3. Di halaman Buat instans notebook, berikan informasi berikut:
   a. Untuk nama contoh Notebook, masukkan**`RunScriptNotebookInstance`**.
   b. Untuk jenis Instans Notebook, pilih**`ml.t2.medium`**.
   c. Di bagian Izin dan enkripsi, lakukan hal berikut:
      i. Untuk peran IAM, pilih Buat peran baru.
      ii. Pada halaman Buat peran IAM, pilih Bucket S3 Spesifik, tentukan bucket Amazon S3 bernama**`sagemaker-run-script`**, lalu pilih Buat peran.

      SageMaker menciptakan peran IAM bernama`AmazonSageMaker-ExecutionRole-`*`YYYYMMDD`*`T`*`HHmmSS`*, seperti`AmazonSageMaker-ExecutionRole-20190429T110788`. Perhatikan bahwa konvensi penamaan peran eksekusi menggunakan tanggal dan waktu ketika peran dibuat, dipisahkan oleh `aT`.
   d. Untuk Akses Root, pilih Aktifkan.

e. Pilih Buat instans notebook.

4. Pada halaman Instans Notebook, Status Tertunda. Amazon CloudWatch Internet Monitor memerlukan beberapa menit untuk meluncurkan instance komputasi pembelajaran mesin — dalam hal ini, Amazon Internet Monitor meluncurkan instance notebook—dan melampirkan volume penyimpanan ML-nya. Instance notebook memiliki server notebook Jupyter yang telah dikonfigurasi sebelumnya dan satu set pustaka Anaconda. Untuk informasi selengkapnya, lihat CreateNotebookInstance.

5. Di bagian Izin dan enkripsi, salin nomor ARN peran IAM, dan tempelkan ke file notepad untuk menyimpannya sementara. Anda menggunakan nomor ARN peran IAM ini nanti untuk mengonfigurasi estimator pelatihan lokal di instance notebook. Nomor ARN peran IAM akan terlihat seperti berikut ini:`'arn:aws:iam::111122223333:role/service-role/AmazonSageMaker-ExecutionRole-20190429T110788'`

6. Setelah status instance notebook berubah InService, pilih Buka JupyterLab.

Langkah 2: Buat dan Unggah Skrip Pelatihan Dockerfile dan Python

1. Setelah JupyterLab terbuka, buat folder baru di direktori home Anda JupyterLab. Di pojok kiri atas, pilih ikon Folder Baru, lalu masukkan nama folder`docker_test_folder`.

2. Buat file`Dockerfile` teks di`docker_test_folder` direktori.

   a. Pilih ikon New Launcher (+) di pojok kiri atas.

   b. Di panel kanan bawah Lainnya bagian, pilih File Teks.

   c. Tempelkan kode`Dockerfile` contoh berikut ke dalam file teks Anda.

```
# SageMaker PyTorch image
FROM 763104351884.dkr.ecr.us-east-1.amazonaws.com/pytorch-training:1.5.1-cpu-
py36-ubuntu16.04

ENV PATH="/opt/ml/code:${PATH}"

# this environment variable is used by the SageMaker PyTorch container to
 determine our user code directory.
ENV SAGEMAKER_SUBMIT_DIRECTORY /opt/ml/code

# /opt/ml and all subdirectories are utilized by SageMaker, use the /code
 subdirectory to store your user code.
COPY cifar10.py /opt/ml/code/cifar10.py
```

```
# Defines cifar10.py as script entrypoint
ENV SAGEMAKER_PROGRAM cifar10.py
```

Skrip Dockerfile melakukan tugas-tugas berikut:

- `FROM 763104351884.dkr.ecr.us-east-1.amazonaws.com/pytorch-training:1.5.1-cpu-py36-ubuntu16.04`- Download gambar SageMaker PyTorch dasar. Anda dapat mengganti ini dengan gambar SageMaker dasar apa pun yang ingin Anda bawa untuk membangun kontainer.
- `ENV SAGEMAKER_SUBMIT_DIRECTORY /opt/ml/code`- Set`/opt/ml/code` sebagai direktori skrip pelatihan.
- `COPY cifar10.py /opt/ml/code/cifar10.py`- Menyalin skrip ke lokasi di dalam wadah yang diharapkan oleh SageMaker. Skrip harus ditempatkan di folder ini.
- `ENV SAGEMAKER_PROGRAM cifar10.py`- Menetapkan skrip`cifar10.py` pelatihan Anda sebagai skrip entrypoint.

d. Pada panel navigasi direktori kiri, nama file teks mungkin secara otomatis diberi nama`untitled.txt`. Untuk mengganti nama file, klik kanan file, pilih Ubah nama, ganti nama file sebagai`Dockerfile` tanpa`.txt` ekstensi, lalu tekan`Ctrl+s` atau`Command+s` untuk menyimpan file.

3. Membuat atau meng-upload script pelatihan`cifar10.py` di`docker_test_folder`. Anda dapat menggunakan skrip contoh berikut untuk latihan ini.

```
import ast
import argparse
import logging

import os

import torch
import torch.distributed as dist
import torch.nn as nn
import torch.nn.parallel
import torch.optim
import torch.utils.data
import torch.utils.data.distributed
import torchvision
import torchvision.models
import torchvision.transforms as transforms
```

```
import torch.nn.functional as F

logger=logging.getLogger(__name__)
logger.setLevel(logging.DEBUG)

classes=('plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship',
 'truck')


# https://github.com/pytorch/tutorials/blob/master/beginner_source/blitz/
cifar10_tutorial.py#L118
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1=nn.Conv2d(3, 6, 5)
        self.pool=nn.MaxPool2d(2, 2)
        self.conv2=nn.Conv2d(6, 16, 5)
        self.fc1=nn.Linear(16 * 5 * 5, 120)
        self.fc2=nn.Linear(120, 84)
        self.fc3=nn.Linear(84, 10)

    def forward(self, x):
        x=self.pool(F.relu(self.conv1(x)))
        x=self.pool(F.relu(self.conv2(x)))
        x=x.view(-1, 16 * 5 * 5)
        x=F.relu(self.fc1(x))
        x=F.relu(self.fc2(x))
        x=self.fc3(x)
        return x


def _train(args):
    is_distributed=len(args.hosts) > 1 and args.dist_backend is not None
    logger.debug("Distributed training - {}".format(is_distributed))

    if is_distributed:
        # Initialize the distributed environment.
        world_size=len(args.hosts)
        os.environ['WORLD_SIZE']=str(world_size)
        host_rank=args.hosts.index(args.current_host)
        dist.init_process_group(backend=args.dist_backend, rank=host_rank,
 world_size=world_size)
        logger.info(
```

```
            'Initialized the distributed environment: \'{}\' backend on {} nodes.
'.format(
                args.dist_backend,
                dist.get_world_size()) + 'Current host rank is {}. Using cuda: {}.
Number of gpus: {}'.format(
                dist.get_rank(), torch.cuda.is_available(), args.num_gpus))

    device='cuda' if torch.cuda.is_available() else 'cpu'
    logger.info("Device Type: {}".format(device))

    logger.info("Loading Cifar10 dataset")
    transform=transforms.Compose(
        [transforms.ToTensor(),
         transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])

    trainset=torchvision.datasets.CIFAR10(root=args.data_dir, train=True,
                                          download=False, transform=transform)
    train_loader=torch.utils.data.DataLoader(trainset, batch_size=args.batch_size,
                                             shuffle=True,
num_workers=args.workers)

    testset=torchvision.datasets.CIFAR10(root=args.data_dir, train=False,
                                         download=False, transform=transform)
    test_loader=torch.utils.data.DataLoader(testset, batch_size=args.batch_size,
                                            shuffle=False,
num_workers=args.workers)

    logger.info("Model loaded")
    model=Net()

    if torch.cuda.device_count() > 1:
        logger.info("Gpu count: {}".format(torch.cuda.device_count()))
        model=nn.DataParallel(model)

    model=model.to(device)

    criterion=nn.CrossEntropyLoss().to(device)
    optimizer=torch.optim.SGD(model.parameters(), lr=args.lr,
momentum=args.momentum)

    for epoch in range(0, args.epochs):
        running_loss=0.0
        for i, data in enumerate(train_loader):
            # get the inputs
```

```
            inputs, labels=data
            inputs, labels=inputs.to(device), labels.to(device)

            # zero the parameter gradients
            optimizer.zero_grad()

            # forward + backward + optimize
            outputs=model(inputs)
            loss=criterion(outputs, labels)
            loss.backward()
            optimizer.step()

            # print statistics
            running_loss += loss.item()
            if i % 2000 == 1999:  # print every 2000 mini-batches
                print('[%d, %5d] loss: %.3f' %
                      (epoch + 1, i + 1, running_loss / 2000))
                running_loss=0.0
    print('Finished Training')
    return _save_model(model, args.model_dir)


def _save_model(model, model_dir):
    logger.info("Saving the model.")
    path=os.path.join(model_dir, 'model.pth')
    # recommended way from http://pytorch.org/docs/master/notes/serialization.html
    torch.save(model.cpu().state_dict(), path)


def model_fn(model_dir):
    logger.info('model_fn')
    device="cuda" if torch.cuda.is_available() else "cpu"
    model=Net()
    if torch.cuda.device_count() > 1:
        logger.info("Gpu count: {}".format(torch.cuda.device_count()))
        model=nn.DataParallel(model)

    with open(os.path.join(model_dir, 'model.pth'), 'rb') as f:
        model.load_state_dict(torch.load(f))
    return model.to(device)


if __name__ == '__main__':
    parser=argparse.ArgumentParser()
```

```
    parser.add_argument('--workers', type=int, default=2, metavar='W',
                        help='number of data loading workers (default: 2)')
    parser.add_argument('--epochs', type=int, default=2, metavar='E',
                        help='number of total epochs to run (default: 2)')
    parser.add_argument('--batch-size', type=int, default=4, metavar='BS',
                        help='batch size (default: 4)')
    parser.add_argument('--lr', type=float, default=0.001, metavar='LR',
                        help='initial learning rate (default: 0.001)')
    parser.add_argument('--momentum', type=float, default=0.9, metavar='M',
 help='momentum (default: 0.9)')
    parser.add_argument('--dist-backend', type=str, default='gloo',
 help='distributed backend (default: gloo)')

    # The parameters below retrieve their default values from SageMaker environment
 variables, which are
    # instantiated by the SageMaker containers framework.
    # https://github.com/aws/sagemaker-containers#how-a-script-is-executed-inside-
the-container
    parser.add_argument('--hosts', type=str,
 default=ast.literal_eval(os.environ['SM_HOSTS']))
    parser.add_argument('--current-host', type=str,
 default=os.environ['SM_CURRENT_HOST'])
    parser.add_argument('--model-dir', type=str,
 default=os.environ['SM_MODEL_DIR'])
    parser.add_argument('--data-dir', type=str,
 default=os.environ['SM_CHANNEL_TRAINING'])
    parser.add_argument('--num-gpus', type=int, default=os.environ['SM_NUM_GPUS'])

    _train(parser.parse_args())
```

Langkah 3: Bangun kontainer

1. Di direktori JupyterLab home, buka notebook Jupyter. Untuk membuka notebook baru, pilih ikon New Launch dan kemudian pilih conda_pytorch_p39 di bagian Notebook.
2. Jalankan perintah berikut di sel notebook pertama yang berubah ke`docker_test_folder` direktori:

   ```
   % cd ~/SageMaker/docker_test_folder
   ```

   Ini mengembalikan direktori Anda saat ini sebagai berikut:

```
! pwd
```

`output: /home/ec2-user/SageMaker/docker_test_folder`

3. Masuk ke Docker untuk mengakses kontainer dasar:

```
! aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin 763104351884.dkr.ecr.us-east-1.amazonaws.com
```

4. Untuk membangun container Docker, jalankan perintah Docker build berikut, termasuk ruang yang diikuti dengan periode di akhir:

```
! docker build -t pytorch-extended-container-test .
```

Perintah Docker build harus dijalankan dari direktori Docker yang Anda buat, dalam kasus ini`docker_test_folder`.

> **Note**
>
> Jika Anda mendapatkan pesan galat berikut bahwa Docker tidak dapat menemukan Dockerfile, pastikan Dockerfile memiliki nama yang benar dan telah disimpan ke direktori.
>
> ```
> unable to prepare context: unable to evaluate symlinks in Dockerfile path:
> lstat /home/ec2-user/SageMaker/docker/Dockerfile: no such file or directory
> ```
>
> Ingat bahwa`docker` mencari file yang secara khusus dipanggil`Dockerfile` tanpa ekstensi apa pun dalam direktori saat ini. Jika Anda menamainya sesuatu yang lain, Anda dapat meneruskan nama file secara manual dengan`-f` bendera. Sebagai contoh, jika Anda menamai Dockerfile Anda`Dockerfile-text.txt`, jalankan perintah berikut:
>
> ```
> ! docker build -t tf-custom-container-test -f Dockerfile-text.txt .
> ```

Langkah 4: Uji kontainer

1. Untuk menguji kontainer secara lokal di instance notebook, buka notebook Jupyter. Pilih Peluncur Baru dan pilih Notebook dalam **`conda_pytorch_p39`**kerangka kerja. Sisa cuplikan kode harus dijalankan dari instance notebook Jupyter.

2. Unduh kumpulan data CIFAR-10.

```
import torch
import torchvision
import torchvision.transforms as transforms

def _get_transform():
    return transforms.Compose(
        [transforms.ToTensor(),
         transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])


def get_train_data_loader(data_dir='/tmp/pytorch/cifar-10-data'):
    transform=_get_transform()
    trainset=torchvision.datasets.CIFAR10(root=data_dir, train=True,
                                          download=True, transform=transform)
    return torch.utils.data.DataLoader(trainset, batch_size=4,
                                       shuffle=True, num_workers=2)


def get_test_data_loader(data_dir='/tmp/pytorch/cifar-10-data'):
    transform=_get_transform()
    testset=torchvision.datasets.CIFAR10(root=data_dir, train=False,
                                         download=True, transform=transform)
    return torch.utils.data.DataLoader(testset, batch_size=4,
                                       shuffle=False, num_workers=2)

trainloader=get_train_data_loader('/tmp/pytorch-example/cifar-10-data')
testloader=get_test_data_loader('/tmp/pytorch-example/cifar-10-data')
```

3. Atur`role` ke peran yang digunakan untuk membuat notebook Jupyter Anda. Ini digunakan untuk mengkonfigurasi SageMaker Estimator Anda.

```
from sagemaker import get_execution_role

role=get_execution_role()
```

4. Tempel skrip contoh berikut ke dalam sel kode notebook untuk mengkonfigurasi SageMaker Estimator menggunakan wadah yang diperluas.

```
from sagemaker.estimator import Estimator

hyperparameters={'epochs': 1}
```

```
estimator=Estimator(
    image_uri='pytorch-extended-container-test',
    role=role,
    instance_count=1,
    instance_type='local',
    hyperparameters=hyperparameters
)

estimator.fit('file:///tmp/pytorch-example/cifar-10-data')
```

5. Jalankan sel kode. Tes ini menghasilkan konfigurasi lingkungan pelatihan, nilai yang digunakan untuk variabel lingkungan, sumber data, dan kehilangan dan akurasi yang diperoleh selama pelatihan.

Langkah 5: Tekan Kontainer ke Amazon Elastic Container Registry (Amazon ECR)

1. Setelah berhasil menjalankan pengujian mode lokal, Anda dapat mendorong container Docker ke Amazon ECR dan menggunakannya untuk menjalankan pekerjaan pelatihan.

   Jalankan baris perintah berikut di sel notebook.

```
%%sh

# Specify an algorithm name
algorithm_name=pytorch-extended-container-test

account=$(aws sts get-caller-identity --query Account --output text)

# Get the region defined in the current configuration (default to us-west-2 if none
 defined)
region=$(aws configure get region)

fullname="${account}.dkr.ecr.${region}.amazonaws.com/${algorithm_name}:latest"

# If the repository doesn't exist in ECR, create it.

aws ecr describe-repositories --repository-names "${algorithm_name}" > /dev/null
 2>&1
if [ $? -ne 0 ]
then
aws ecr create-repository --repository-name "${algorithm_name}" > /dev/null
```

```
fi

# Log into Docker
aws ecr get-login-password --region ${region}|docker login --username AWS --
password-stdin ${fullname}

# Build the docker image locally with the image name and then push it to ECR
# with the full name.

docker build -t ${algorithm_name} .
docker tag ${algorithm_name} ${fullname}

docker push ${fullname}
```

2. Setelah Anda mendorong wadah, Anda dapat memanggil gambar Amazon ECR dari mana saja di SageMaker lingkungan. Jalankan contoh kode berikut di sel notebook berikutnya.

   Jika Anda ingin menggunakan wadah pelatihan ini dengan SageMaker Studio untuk menggunakan fitur visualisasi, Anda juga dapat menjalankan kode berikut di sel notebook Studio untuk memanggil gambar Amazon ECR dari wadah pelatihan Anda.

```
import boto3

client=boto3.client('sts')
account=client.get_caller_identity()['Account']

my_session=boto3.session.Session()
region=my_session.region_name

algorithm_name="pytorch-extended-container-test"
ecr_image='{}.dkr.ecr.{}.amazonaws.com/{}:latest'.format(account, region,
 algorithm_name)

ecr_image
# This should return something like
# 12-digits-of-your-account.dkr.ecr.us-east-2.amazonaws.com/tf-2.2-test:latest
```

3. Gunakan`ecr_image` diambil dari langkah sebelumnya untuk mengkonfigurasi objek SageMaker estimator. Contoh kode berikut mengkonfigurasi SageMaker PyTorch estimator.

```
import sagemaker

from sagemaker import get_execution_role
```

```
from sagemaker.estimator import Estimator

estimator=Estimator(
    image_uri=ecr_image,
    role=get_execution_role(),
    base_job_name='pytorch-extended-container-test',
    instance_count=1,
    instance_type='ml.p2.xlarge'
)

# start training
estimator.fit()

# deploy the trained model
predictor=estimator.deploy(1, instance_type)
```

Langkah 6: Bersihkan Sumber Daya

Untuk membersihkan sumber daya saat selesai dengan contoh Memulai

1. Buka SageMaker konsol, pilih instance notebook RunScriptNotebookInstance, pilih Tindakan, dan pilih Berhenti. Ini dapat memerlukan waktu beberapa menit sampai instans berhenti.
2. Setelah status instance berubah menjadi Berhenti, pilih Tindakan, pilih Hapus, lalu pilih Hapus di kotak dialog. Perlu waktu beberapa menit agar instance dihapus. Contoh notebook menghilang dari tabel ketika telah dihapus.
3. Buka konsol Amazon S3 dan hapus bucket yang Anda buat untuk menyimpan artefak model dan set data pelatihan.
4. Buka konsol IAM dan hapus peran IAM. Jika membuat kebijakan izin, Anda dapat menghapusnya juga.

> **Note**
>
> Container Docker mati secara otomatis setelah berjalan. Anda tidak perlu menghapusnya.

Mengadaptasi wadah Docker Anda sendiri untuk bekerja dengan SageMaker

Anda dapat mengadaptasi gambar Docker yang ada untuk bekerja dengannya SageMaker. Anda mungkin perlu menggunakan image Docker eksternal yang sudah ada SageMaker saat Anda memiliki wadah yang memenuhi persyaratan fitur atau keamanan yang saat ini tidak didukung oleh gambar yang dibuat sebelumnya SageMaker. Ada dua toolkit yang memungkinkan Anda membawa wadah Anda sendiri dan menyesuaikannya untuk bekerja dengan: SageMaker

- SageMaker Toolkit Pelatihan
- SageMaker Toolkit Inferensi

Topik berikut menunjukkan cara menyesuaikan gambar Anda yang ada menggunakan toolkit SageMaker Pelatihan dan Inferensi:

Topik

- Perpustakaan Kerangka Individu
- Menggunakan SageMaker Pelatihan dan Inferensi Toolkit
- Mengadaptasi wadah pelatihan Anda sendiri
- Mengadaptasi Wadah Inferensi Anda Sendiri

Perpustakaan Kerangka Individu

Selain SageMaker Training Toolkit dan SageMaker Inference Toolkit, SageMaker juga menyediakan toolkit khusus untuk, TensorFlow MxNet,, dan Chainer. PyTorch Tabel berikut menyediakan tautan ke GitHub repositori yang berisi kode sumber untuk setiap kerangka kerja dan toolkit penyajiannya masing-masing. Instruksi yang ditautkan adalah untuk menggunakan Python SDK untuk menjalankan algoritme pelatihan dan model host. SageMaker Fungsionalitas untuk pustaka individu ini disertakan dalam SageMaker Training Toolkit dan SageMaker Inference Toolkit.

Kerangka Kerja	Kode Sumber Toolkit
`TensorFlow`	SageMaker TensorFlow Pelatihan SageMaker TensorFlow Melayani

Kerangka Kerja	Kode Sumber Toolkit
`MXNet`	SageMaker Pelatihan MxNet SageMaker Inferensi MxNet
`PyTorch`	SageMaker PyTorch Pelatihan SageMaker PyTorch Inferensi
`Chainer`	SageMaker Wadah Chainer SageMaker

Menggunakan SageMaker Pelatihan dan Inferensi Toolkit

KlasterSageMaker PelatihandanSageMaker Inferensitoolkit mengimplementasikan fungsionalitas yang Anda butuhkan untuk menyesuaikan kontainer Anda untuk menjalankan skrip, melatih algoritme, dan menerapkan model SageMaker. Saat diinstal, pustaka mendefinisikan hal berikut untuk pengguna:

- Lokasi untuk menyimpan kode dan sumber daya lainnya.
- Titik masuk yang berisi kode yang akan dijalankan saat kontainer dimulai. Dockerfile Anda harus menyalin kode yang perlu dijalankan ke lokasi yang diharapkan oleh wadah yang kompatibel dengan SageMaker.
- Informasi lain yang dibutuhkan kontainer untuk mengelola penerapan untuk pelatihan dan inferensi.

SageMaker Struktur Kontainer alat

Saat SageMaker melatih model, itu menciptakan struktur folder file berikut dalam wadah`/opt/ml`Direktori.

```
/opt/ml
### input
#   ### config
#   #   ### hyperparameters.json
#   #   ### resourceConfig.json
#   ### data
#       ### <channel_name>
#           ### <input data>
### model
```

```
#
### code
#
### output
#
### failure
```

Saat Anda menjalankan modelpelatihanpekerjaan, yang SageMaker kontainer menggunakan/opt/ml/input/direktori, yang berisi file JSON yang mengkonfigurasi hyperparameters untuk algoritma dan tata letak jaringan yang digunakan untuk pelatihan terdistribusi. Klaster/opt/ml/input/direktori juga berisi file yang menentukan saluran melalui mana SageMaker mengakses data, yang disimpan di Amazon Simple Storage Service (Amazon S3). Klaster SageMaker container library menempatkan script yang kontainer akan berjalan di/opt/ml/code/Direktori. Script Anda harus menulis model yang dihasilkan oleh algoritma Anda ke/opt/ml/model/Direktori. Untuk informasi selengkapnya, lihat Gunakan Algoritma Pelatihan Anda Sendiri.

Saat Andatuan rumahmodel terlatih SageMaker untuk membuat kesimpulan, Anda menyebarkan model ke endpoint HTTP. Model ini membuat prediksi waktu nyata sebagai respons terhadap permintaan inferensi. Container harus berisi tumpukan penyajian untuk memproses permintaan ini.

Dalam wadah hosting atau batch transform, file model terletak di folder yang sama dengan yang ditulis selama pelatihan.

```
/opt/ml/model
#
### <model files>
```

Untuk informasi selengkapnya, lihat Gunakan kode inferensi Anda sendiri.

Tunggal Versus Beberapa Kontainer

Anda dapat menyediakan image Docker terpisah untuk algoritma pelatihan dan kode inferensi atau Anda dapat menggunakan image Docker tunggal untuk keduanya. Saat membuat gambar Docker untuk digunakan dengan SageMaker, pertimbangkan hal berikut:

- Menyediakan dua image Docker dapat meningkatkan persyaratan penyimpanan dan biaya karena pustaka umum mungkin diduplikasi.
- Secara umum, wadah yang lebih kecil mulai lebih cepat untuk pelatihan dan hosting. Model berlatih lebih cepat dan layanan hosting dapat bereaksi terhadap peningkatan lalu lintas dengan penskalaan secara otomatis lebih cepat.

- Anda mungkin dapat menulis wadah inferensi yang jauh lebih kecil dari wadah pelatihan. Ini sangat umum ketika Anda menggunakan GPU untuk pelatihan, tetapi kode inferensi Anda dioptimalkan untuk CPU.
- SageMaker mengharuskan kontainer Docker berjalan tanpa akses istimewa.
- Kedua kontainer Docker yang Anda buat dan yang disediakan oleh SageMaker dapat mengirim pesan ke`Stdout`dan`Stderr`. SageMaker mengirim pesan ini ke Amazon CloudWatch log diAWSakun.

Untuk informasi lebih lanjut tentang cara membuat SageMaker kontainer dan bagaimana skrip dijalankan di dalamnya, lihatSageMakerKit Alat Trainingdan[SageMaker Kit inferensi](https://github.com/aws/sagemaker-inference-toolkit)repositori pada GitHub. Mereka juga menyediakan daftar variabel lingkungan penting dan variabel lingkungan yang disediakan oleh SageMakerkontainer.

Mengadaptasi wadah pelatihan Anda sendiri

Untuk menjalankan model pelatihan Anda sendiri, buat wadah Docker menggunakan[Amazon SageMaker Toolkit Pelatihan](https://github.com/aws/sagemaker-training-toolkit)melalui Amazon SageMaker contoh buku catatan.

Langkah 1: Buat SageMaker contoh notebook

1. Buka Amazon SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi kiri, pilihBuku catatan, pilihInstance Notebook, dan kemudian pilihBuat contoh buku catatan.
3. PadaBuat contoh buku catatanhalaman, memberikan informasi berikut:

    a. UntukNama contoh buku catatan, masukkan**`RunScriptNotebookInstance`**.

    b. UntukJenis Instance Notebook, pilih**`ml.t2.medium`**.

    c. Dilzin dan enkripsibagian, lakukan hal berikut ini:

        i. UntukPeran IAM, pilihBuat peran baru. Ini membuka jendela baru.

        ii. PadaBuat IAM rolehalaman, pilihEmber S3 khusus, tentukan bucket Amazon S3**`sagemaker-run-script`**, dan kemudian pilihBuat peran.

        SageMaker membuat peran IAM bernama`AmazonSageMaker-ExecutionRole-`*`YYYYMMDD`*`T`*`HHmmSS`*. Sebagai contoh, `AmazonSageMaker-ExecutionRole-20190429T110788`. Perhatikan bahwa konvensi penamaan peran eksekusi menggunakan tanggal dan waktu di mana peran itu dibuat, dipisahkan olehT.

d. UntukAkses Root, pilihAktifkan.

e. Pilih Buat instans notebook.

4. PadaInstance Notebookhalaman,StatusadalahTertunda. Ini bisa memakan waktu beberapa menit untuk Amazon SageMaker untuk meluncurkan instance komputasi pembelajaran mesin —dalam hal ini, ia meluncurkan instance notebook—dan melampirkan volume penyimpanan ML ke dalamnya. Instance notebook memiliki server notebook Jupyter yang telah dikonfigurasi sebelumnya dan satu set pustaka Anaconda. Untuk informasi selengkapnya, lihat CreateNotebookInstance.

5. Klik padaNamadari buku catatan yang baru saja kamu buat. Ini membuka halaman baru.

6. DiIzin dan enkripsibagian, salinnomor ARN peran IAM, dan tempelkan ke file notepad untuk menyimpannya sementara. Anda menggunakan nomor ARN peran IAM ini nanti untuk mengonfigurasi estimator pelatihan lokal di instance notebook. Nomor ARN peran IAMterlihat seperti berikut ini:`'arn:aws:iam::111122223333:role/service-role/AmazonSageMaker-ExecutionRole-20190429T110788'`

7. Setelah status instance notebook berubah menjadiInService, pilihTerbuka JupyterLab.

Langkah 2: Buat dan unggah skrip pelatihan Dockerfile dan Python

1. Setelah JupyterLab terbuka, buat folder baru di direktori home JupyterLab. Di sudut kiri atas, pilihFolder Baruikon, dan kemudian masukkan nama folder`docker_test_folder`.

2. Buat`Dockerfile`file teks di`docker_test_folder`direktori.

   a. PilihPeluncur Baruikon (+) di pojok kiri atas.

   b. Di panel kanan di bawahlainnyabagian, pilihFile Teks.

   c. Tempel yang berikut`Dockerfile`kode sampel ke dalam file teks Anda.

```
#Download an open source TensorFlow Docker image
FROM tensorflow/tensorflow:latest-gpu-jupyter

# Install sagemaker-training toolkit that contains the common functionality
 necessary to create a container compatible with SageMaker and the Python SDK.
RUN pip3 install sagemaker-training

# Copies the training code inside the container
COPY train.py /opt/ml/code/train.py
```

```
# Defines train.py as script entrypoint
ENV SAGEMAKER_PROGRAM train.py
```

Skrip Dockerfile melakukan tugas-tugas berikut ini:

- `FROM tensorflow/tensorflow:latest-gpu-jupyter`— Download terbaru TensorFlow Gambar dasar Docker. Anda dapat mengganti ini dengan gambar dasar Docker apa pun yang ingin Anda bawa untuk membangun wadah, serta denganAWSgambar dasar kontainer yang sudah dibangun sebelumnya.
- `RUN pip install sagemaker-training`— MenginstalSageMakerToolkit Pelatihanyang berisi fungsionalitas umum yang diperlukan untuk membuat wadah yang kompatibel dengan SageMaker.
- `COPY train.py /opt/ml/code/train.py`— Menyalin skrip ke lokasi di dalam wadah yang diharapkan oleh SageMaker. Skrip harus berada di folder ini.
- `ENV SAGEMAKER_PROGRAM train.py`— Mengambil skrip pelatihan Anda`train.py`sebagai skrip titik masuk yang disalin di`/opt/ml/code`folder wadah. Ini adalah satu-satunya variabel lingkungan yang harus Anda tentukan ketika Anda membangun wadah Anda sendiri.

d. Pada panel navigasi direktori kiri, nama file teks mungkin secara otomatis diberi nama`untitled.txt`. Untuk mengganti nama file, klik kanan file tersebut, pilihGanti nama, ganti nama file sebagai`Dockerfile`tanpa`.txt`tekstensi, lalu tekan`Ctrl+s`atau`Command +s`untuk menyimpan file.

3. Unggah skrip pelatihan`train.py`ke`docker_test_folder`. Anda dapat menggunakan contoh skrip berikut untuk membuat model yang membaca digit tulisan tangan yang dilatih padaDataset MNISTuntuk latihan ini.

```
import tensorflow as tf
import os

mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
tf.keras.layers.Flatten(input_shape=(28, 28)),
tf.keras.layers.Dense(128, activation='relu'),
tf.keras.layers.Dropout(0.2),
```

```
tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

model.fit(x_train, y_train, epochs=1)
model_save_dir = f"{os.environ.get('SM_MODEL_DIR')}/1"

model.evaluate(x_test, y_test)
tf.saved_model.save(model, model_save_dir)
```

Langkah 3: Bangun wadah

1. Di JupyterLab direktori home, buka notebook Jupyter. Untuk membuka buku catatan baru, pilihPeluncuran baruikon dan kemudian pilih versi terbaruconda_tensorflow2diBuku catatanbagian.
2. Jalankan perintah berikut di sel notebook pertama untuk mengubah ke`docker_test_folder`direktori:

   ```
   cd ~/SageMaker/docker_test_folder
   ```

   Ini mengembalikan direktori Anda saat ini:

   ```
   ! pwd
   ```

   `output: /home/ec2-user/SageMaker/docker_test_folder`
3. Untuk membangun kontainer Docker, jalankan perintah build Docker berikut, termasuk spasi yang diikuti oleh titik di akhir:

   ```
   ! docker build -t tf-custom-container-test .
   ```

   Perintah build Docker harus dijalankan dari direktori Docker yang Anda buat, dalam hal ini`docker_test_folder`.

> **Note**
>
> Jika Anda mendapatkan pesan kesalahan berikut bahwa Docker tidak dapat menemukan Dockerfile, pastikan Dockerfile memiliki nama yang benar dan telah disimpan ke direktori.
>
> ```
> unable to prepare context: unable to evaluate symlinks in Dockerfile path:
> lstat /home/ec2-user/SageMaker/docker/Dockerfile: no such file or directory
> ```
>
> Ingat itu`docker`mencari file yang secara khusus disebut`Dockerfile`tanpa ekstensi apa pun dalam direktori saat ini. Jika Anda menamakannya sesuatu yang lain, Anda dapat meneruskan nama file secara manual dengan`-f`bendera. Misalnya, jika Anda menamai Dockerfile Anda sebagai`Dockerfile-text.txt`, jalankan perintah berikut ini:
>
> ```
> ! docker build -t tf-custom-container-test -f Dockerfile-text.txt .
> ```

Langkah 4: Uji wadahnya

1. Untuk menguji kontainer secara lokal di instance notebook, buka notebook Jupyter. PilihPeluncur Barudan memilih versi terbaruconda_tensorflow2diBuku catatanbagian.
2. Tempelkan contoh skrip berikut ke sel kode notebook untuk mengkonfigurasi SageMaker Penaksir.

```
import sagemaker
from sagemaker.estimator import Estimator

estimator = Estimator(image_uri='tf-custom-container-test',
                      role=sagemaker.get_execution_role(),
                      instance_count=1,
                      instance_type='local')

estimator.fit()
```

Contoh kode sebelumnya,`sagemaker.get_execution_role()`ditentukan untuk`role`argumen untuk secara otomatis mengambil peran yang disiapkan untuk SageMaker sesi. Anda juga dapat menggantinya dengan nilai stringnomor ARN peran IAMAnda menggunakan saat mengkonfigurasi instance notebook. Outputnya akan terlihat seperti

berikut:`'arn:aws:iam::111122223333:role/service-role/AmazonSageMaker-ExecutionRole-20190429T110788'`.

3. Jalankan sel kode. Tes ini menghasilkan konfigurasi lingkungan pelatihan, nilai yang digunakan untuk variabel lingkungan, sumber data, dan kehilangan dan akurasi yang diperoleh selama pelatihan.

Langkah 5: Dorong wadah ke Amazon Elastic Container Registry (Amazon ECR)

1. Setelah Anda berhasil menjalankan tes mode lokal, Anda dapat mendorong wadah Docker keAmazon ECRdan menggunakannya untuk menjalankan pekerjaan pelatihan. Jika Anda ingin menggunakan registri Docker pribadi alih-alih Amazon ECR, lihatDorong wadah pelatihan Anda ke registri pribadi.

   Jalankan baris perintah berikut ini di sel notebook.

```
%%sh

# Specify an algorithm name
algorithm_name=tf-custom-container-test

account=$(aws sts get-caller-identity --query Account --output text)

# Get the region defined in the current configuration (default to us-west-2 if none
 defined)
region=$(aws configure get region)
region=${region:-us-west-2}

fullname="${account}.dkr.ecr.${region}.amazonaws.com/${algorithm_name}:latest"

# If the repository doesn't exist in ECR, create it.

aws ecr describe-repositories --repository-names "${algorithm_name}" > /dev/null
 2>&1
if [ $? -ne 0 ]
then
aws ecr create-repository --repository-name "${algorithm_name}" > /dev/null
fi

# Get the login command from ECR and execute it directly
```

```
aws ecr get-login-password --region ${region}|docker login --username AWS --
password-stdin ${fullname}

# Build the docker image locally with the image name and then push it to ECR
# with the full name.

docker build -t ${algorithm_name} .
docker tag ${algorithm_name} ${fullname}

docker push ${fullname}
```

> **Note**
>
> Skrip bash shell ini dapat menimbulkan masalah izin yang mirip dengan pesan kesalahan berikut:
>
> ```
> "denied: User: [ARN] is not authorized to perform: ecr:InitiateLayerUpload
>  on resource:
> arn:aws:ecr:us-east-1:[id]:repository/tf-custom-container-test"
> ```
>
> Jika kesalahan ini terjadi, Anda harus melampirkanAmazonEC2ContainerRegistryFullAccesskebijakan untuk peran IAM Anda. Pergi keKonsol IAM, pilihPerandari panel navigasi kiri, cari IamRole yang Anda gunakan untuk instance Notebook. Di bawahIzintab, pilihLampirkan kebijakantombol, dan cariAmazonEC2ContainerRegistryFullAccesskebijakan. Tandai kotak centang kebijakan, dan pilihTambahkan izinuntuk menyelesaikan.

2. Jalankan kode berikut di sel notebook Studio untuk memanggil image Amazon ECR dari wadah pelatihan Anda.

```
import boto3

account_id = boto3.client('sts').get_caller_identity().get('Account')
ecr_repository = 'tf-custom-container-test'
tag = ':latest'

region = boto3.session.Session().region_name

uri_suffix = 'amazonaws.com'
if region in ['cn-north-1', 'cn-northwest-1']:
```

```
    uri_suffix = 'amazonaws.com.cn'

byoc_image_uri = '{}.dkr.ecr.{}.{}/{}'.format(account_id, region, uri_suffix,
 ecr_repository + tag)

byoc_image_uri
# This should return something like
# 111122223333.dkr.ecr.us-east-2.amazonaws.com/sagemaker-byoc-test:latest
```

3. Gunakan`ecr_image`diambil dari langkah sebelumnya untuk mengkonfigurasi SageMaker objek estimator. Contoh kode berikut mengkonfigurasi SageMaker estimator dengan`byoc_image_uri`dan memulai pekerjaan pelatihan pada instans Amazon EC2.

   SageMaker Python SDK v1

   ```
   import sagemaker
   from sagemaker import get_execution_role
   from sagemaker.estimator import Estimator

   estimator = Estimator(image_uri=byoc_image_uri,
                         role=get_execution_role(),
                         base_job_name='tf-custom-container-test-job',
                         instance_count=1,
                         instance_type='ml.g4dn.xlarge')

   #train your model
   estimator.fit()
   ```

   SageMaker Python SDK v2

   ```
   import sagemaker
   from sagemaker import get_execution_role
   from sagemaker.estimator import Estimator

   estimator = Estimator(image_uri=byoc_image_uri,
                         role=get_execution_role(),
                         base_job_name='tf-custom-container-test-job',
                         instance_count=1,
                         instance_type='ml.g4dn.xlarge')

   #train your model
   ```

```
estimator.fit()
```

4. Jika Anda ingin menerapkan model Anda menggunakan wadah Anda sendiri, lihatMengadaptasi Wadah Inferensi Anda Sendiri. Anda juga dapat menggunakanAWSwadah kerangka kerja yang dapat menerapkan TensorFlow model. Untuk menerapkan model contoh untuk membaca digit tulisan tangan, masukkan skrip contoh berikut ke dalam buku catatan yang sama yang Anda gunakan untuk melatih model Anda di sub-langkah sebelumnya untuk mendapatkan URI gambar (pengidentifikasi sumber daya universal) yang diperlukan untuk penerapan, dan gunakan model.

```
import boto3
import sagemaker

#obtain image uris
from sagemaker import image_uris
container = image_uris.retrieve(framework='tensorflow',region='us-
west-2',version='2.11.0',
                    image_scope='inference',instance_type='ml.g4dn.xlarge')

#create the model entity, endpoint configuration and endpoint
predictor = estimator.deploy(1,instance_type='ml.g4dn.xlarge',image_uri=container)
```

Uji model Anda menggunakan contoh digit tulisan tangan dari kumpulan data MNIST menggunakan contoh kode berikut.

```
#Retrieve an example test dataset to test
import numpy as np
import matplotlib.pyplot as plt
from keras.datasets import mnist

# Load the MNIST dataset and split it into training and testing sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()
# Select a random example from the training set
example_index = np.random.randint(0, x_train.shape[0])
example_image = x_train[example_index]
example_label = y_train[example_index]

# Print the label and show the image
print(f"Label: {example_label}")
plt.imshow(example_image, cmap='gray')
plt.show()
```

Ubah digit tulisan tangan tes menjadi bentuk yang TensorFlow dapat menelan dan membuat prediksi tes.

```
from sagemaker.serializers import JSONSerializer
data = {"instances": example_image.tolist()}
predictor.serializer=JSONSerializer() #update the predictor to use the
 JSONSerializer
predictor.predict(data) #make the prediction
```

Untuk contoh lengkap yang menunjukkan cara menguji wadah khusus secara lokal dan mendorongnya ke gambar Amazon ECR, lihatMembangun Sendiri TensorFlow Kontainercontoh buku catatan.

> **Tip**
>
> Untuk membuat profil dan men-debug pekerjaan pelatihan untuk memantau masalah pemanfaatan sistem (seperti kemacetan CPU dan kurangnya pemanfaatan GPU) dan mengidentifikasi masalah pelatihan (seperti overfitting, overtraining, tensor meledak, dan gradien menghilang), gunakan Amazon SageMaker Debugger. Untuk informasi selengkapnya, lihat Gunakan Debugger dengan Custom Training Containers.

Langkah 6: Bersihkan Sumber Daya

Untuk membersihkan sumber daya saat selesai dengan contoh memulai

1. BukaSageMaker konsol, pilih contoh notebookRunScriptNotebookInstance, pilihTindakan, dan pilihBerhenti. Ini dapat memerlukan waktu beberapa menit sampai instans berhenti.
2. Setelah contohStatusperubahan keBerhenti, pilihTindakan, pilihHapus, dan kemudian pilihHapusdi kotak dialog. Hal ini dapat memakan waktu beberapa menit untuk menghapus. Contoh notebook menghilang dari tabel ketika telah dihapus.
3. BukaKonsol Amazon S3dan hapus ember yang Anda buat untuk menyimpan artefak model dan kumpulan data pelatihan.
4. BukaKonsol IAMdan hapus peran IAM. Jika Anda membuat kebijakan izin, Anda juga dapat menghapusnya.

> **Note**
>
> Kontainer Docker dimatikan secara otomatis setelah dijalankan. Anda tidak perlu menghapusnya.

Blog dan Studi Kasus

Blog berikut membahas studi kasus tentang penggunaan wadah pelatihan khusus di Amazon SageMaker.

- Mengapa membawa wadah Anda sendiri ke Amazon SageMaker dan bagaimana melakukannya dengan benar,Sedang(20 Januari 2023)

Sesuaikan pekerjaan pelatihan Anda untuk mengakses gambar di registri Docker pribadi

Anda dapat menggunakan pribadiRegistri Dockeralih-alih Amazon Elastic Container Registry (Amazon ECR) untuk meng-host gambar Anda SageMakerPelatihan. Petunjuk berikut menunjukkan kepada Anda cara membuat registri Docker, mengonfigurasi cloud pribadi virtual (VPC) dan pekerjaan pelatihan Anda, menyimpan gambar, dan memberikan SageMaker akses ke gambar pelatihan di registri docker pribadi. Instruksi ini juga menunjukkan cara menggunakan registri Docker yang memerlukan otentikasi SageMaker pekerjaan pelatihan.

Buat dan simpan gambar Anda di registri Docker pribadi

Buat registri Docker pribadi untuk menyimpan gambar Anda. Registri Anda harus:

- gunakanAPI HTTP Docker Registryprotokol
- dapat diakses dari VPC yang sama yang ditentukan dalamVpcConfigparameter di`CreateTrainingJob`API. Masukan`VpcConfig`ketika Anda membuat pekerjaan pelatihan Anda.
- diamankan denganSertifikat TLSdari otoritas sertifikat publik yang dikenal.

Untuk informasi selengkapnya cara membuat registri Docker, lihatMenyebarkan server registri.

Konfigurasikan VPC Anda dan SageMaker pekerjaan pelatihan

SageMaker menggunakan koneksi jaringan dalam VPC Anda untuk mengakses gambar di registri Docker Anda. Untuk menggunakan gambar di registri Docker Anda untuk pelatihan, registri harus dapat diakses dari VPC Amazon di akun Anda. Untuk informasi selengkapnya, lihat Gunakan registri Docker yang memerlukan otentikasi untuk pelatihan.

Anda juga harus mengonfigurasi pekerjaan pelatihan Anda untuk terhubung ke VPC yang sama dengan yang dapat diakses oleh registri Docker Anda. Untuk informasi selengkapnya, lihatKonfigurasikan Training Job untuk Amazon VPC Access.

Buat pekerjaan pelatihan menggunakan gambar dari registri Docker pribadi Anda

Untuk menggunakan gambar dari registri Docker pribadi Anda untuk pelatihan, gunakan panduan berikut untuk mengonfigurasi gambar Anda, mengonfigurasi, dan membuat pekerjaan pelatihan. Contoh kode yang mengikuti menggunakanAWS SDK for Python (Boto3)klien.

1. Buat objek konfigurasi gambar pelatihan dan masukan`Vpc`sang`TrainingRepositoryAccessMode`elapangan sebagai berikut.

```
training_image_config = {
    'TrainingRepositoryAccessMode': 'Vpc'
}
```

> **Note**
>
> Jika registri Docker pribadi Anda memerlukan otentikasi, Anda harus menambahkan`TrainingRepositoryAuthConfig`keberatan dengan objek konfigurasi gambar pelatihan. Anda juga harus menentukan nama sumber daya Amazon (ARN)AWS Lambdafungsi yang menyediakan kredensyal akses ke SageMakermenggunakan`TrainingRepositoryCredentialsProviderArn`bidang`TrainingRe`
> Untuk informasi lebih lanjut, lihat contoh kode berikut ini.
>
> ```
> training_image_config = {
>    'TrainingRepositoryAccessMode': 'Vpc',
>    'TrainingRepositoryAuthConfig': {
>         'TrainingRepositoryCredentialsProviderArn':
>  'arn:aws:lambda:Region:Acct:function:FunctionName'
>    }
> ```

```
}
```

Untuk informasi cara melakukannya, lihat cara melakukan autentikasi, lihatGunakan registri Docker yang memerlukan otentikasi untuk pelatihan.

2. Gunakan klien Boto3 untuk membuat pekerjaan pelatihan dan meneruskan konfigurasi yang benar kecreate_training_jobAPI. Petunjuk berikut menunjukkan cara mengkonfigurasi komponen dan membuat pekerjaan pelatihan.

   a. Buat`AlgorithmSpecification`objek yang ingin Anda berikan`create_training_job`. Gunakan objek konfigurasi gambar yang Anda buat di langkah sebelumnya, seperti yang ditunjukkan dalam contoh kode berikut.

   ```
   algorithm_specification = {
      'TrainingImage': 'myteam.myorg.com/docker-local/my-training-image:<IMAGE-TAG>',
      'TrainingImageConfig': training_image_config,
      'TrainingInputMode': 'File'
   }
   ```

   > **Note**
   >
   > Untuk menggunakan versi tetap, bukan versi terbaru dari gambar, lihat gambarmencernabukan dengan nama atau tag.

   b. Tentukan nama pekerjaan pelatihan dan peran yang ingin Anda berikan`create_training_job`, seperti yang ditunjukkan pada contoh kode berikut ini.

   ```
   training_job_name = 'private-registry-job'
   execution_role_arn = 'arn:aws:iam::123456789012:role/SageMakerExecutionRole'
   ```

   c. Tentukan grup keamanan dan subnet untuk konfigurasi VPC untuk pekerjaan pelatihan Anda. Registri Docker pribadi Anda harus mengizinkan lalu lintas masuk dari grup keamanan yang Anda tentukan, seperti yang ditunjukkan dalam contoh kode berikut.

   ```
   vpc_config = {
       'SecurityGroupIds': ['sg-0123456789abcdef0'],
       'Subnets': ['subnet-0123456789abcdef0','subnet-0123456789abcdef1']
   }
   ```

> **Note**
>
> Jika subnet Anda tidak dalam VPC yang sama dengan registri Docker pribadi Anda, Anda harus mengatur koneksi jaringan antara dua VPC. SeeConnect VPC menggunakanPengintip VPCUntuk informasi lebih lanjut.

d. Tentukan konfigurasi sumber daya, termasuk instance komputasi pembelajaran mesin dan volume penyimpanan yang akan digunakan untuk pelatihan, seperti yang ditunjukkan dalam contoh kode berikut.

```
resource_config = {
    'InstanceType': 'ml.m4.xlarge',
    'InstanceCount': 1,
    'VolumeSizeInGB': 10,
}
```

e. Tentukan konfigurasi data input dan output, tempat dataset pelatihan disimpan, dan di mana Anda ingin menyimpan artefak model, seperti yang ditunjukkan pada contoh kode berikut.

```
input_data_config = [
    {
        "ChannelName": "training",
        "DataSource":
        {
            "S3DataSource":
            {
                "S3DataDistributionType": "FullyReplicated",
                "S3DataType": "S3Prefix",
                "S3Uri": "s3://your-training-data-bucket/training-data-folder"
            }
        }
    }
]

output_data_config = {
    'S3OutputPath': 's3://your-output-data-bucket/model-folder'
}
```

f. Tentukan jumlah detik maksimum yang dapat dijalankan oleh pekerjaan pelatihan model seperti yang ditunjukkan pada contoh kode berikut.

```
stopping_condition = {
    'MaxRuntimeInSeconds': 1800
}
```

g. Terakhir, buat pekerjaan pelatihan menggunakan parameter yang Anda tentukan pada langkah sebelumnya seperti yang ditunjukkan pada contoh kode berikut.

```
import boto3
sm = boto3.client('sagemaker')
try:
    resp = sm.create_training_job(
        TrainingJobName=training_job_name,
        AlgorithmSpecification=algorithm_specification,
        RoleArn=execution_role_arn,
        InputDataConfig=input_data_config,
        OutputDataConfig=output_data_config,
        ResourceConfig=resource_config,
        VpcConfig=vpc_config,
        StoppingCondition=stopping_condition
    )
except Exception as e:
    print(f'error calling CreateTrainingJob operation: {e}')
else:
    print(resp)
```

Gunakan SageMaker estimator untuk menjalankan pekerjaan pelatihan

Anda juga dapat menggunakanpenaksirdari SageMaker Python SDK untuk menangani konfigurasi dan menjalankan SageMaker pekerjaan pelatihan. Contoh kode berikut menunjukkan cara mengkonfigurasi dan menjalankan estimator menggunakan gambar dari registri Docker pribadi.

1. Impor pustaka dan dependensi yang diperlukan, seperti yang ditunjukkan pada contoh kode berikut ini.

```
import boto3
import sagemaker
from sagemaker.estimator import Estimator

session = sagemaker.Session()
```

```
role = sagemaker.get_execution_role()
```

2. Berikan Uniform Resource Identifier (URI) ke image latihan, grup keamanan, dan subnet untuk konfigurasi VPC untuk tugas latihan Anda, seperti yang ditunjukkan pada contoh kode berikut.

```
image_uri = "myteam.myorg.com/docker-local/my-training-image:<IMAGE-TAG>"

security_groups = ["sg-0123456789abcdef0"]
subnets = ["subnet-0123456789abcdef0", "subnet-0123456789abcdef0"]
```

   Untuk informasi lebih lanjut tentang`security_group_ids`dan`subnets`, lihat deskripsi parameter yang sesuai diEstimatorbagian dari SageMaker SDK Python.

> **Note**
>
> SageMaker menggunakan koneksi jaringan dalam VPC Anda untuk mengakses gambar di registri Docker Anda. Untuk menggunakan gambar di registri Docker Anda untuk pelatihan, registri harus dapat diakses dari VPC Amazon di akun Anda.

3. Sebagai pilihan, jika registri Docker Anda memerlukan otentikasi, Anda juga harus menentukan Nama Sumber Daya Amazon (ARN) dariAWS Lambdafungsi yang menyediakan kredensyal akses ke SageMaker. Contoh kode berikut menunjukkan cara menentukan ARN.

```
training_repository_credentials_provider_arn = "arn:aws:lambda:us-
west-2:1234567890:function:test"
```

   Untuk informasi selengkapnya tentang penggunaan gambar di registri Docker yang membutuhkan autentikasi, lihatGunakan registri Docker yang memerlukan otentikasi untuk pelatihandi bawah ini.

4. Gunakan contoh kode dari langkah sebelumnya untuk mengonfigurasi penaksir, seperti yang ditunjukkan dalam contoh kode berikut.

```
# The training repository access mode must be 'Vpc' for private docker registry jobs
training_repository_access_mode = "Vpc"

# Specify the instance type, instance count you want to use
instance_type="ml.m5.xlarge"
instance_count=1

# Specify the maximum number of seconds that a model training job can run
```

```
max_run_time = 1800

# Specify the output path for the model artifacts
output_path = "s3://your-output-bucket/your-output-path"

estimator = Estimator(
    image_uri=image_uri,
    role=role,
    subnets=subnets,
    security_group_ids=security_groups,
    training_repository_access_mode=training_repository_access_mode,

 training_repository_credentials_provider_arn=training_repository_credentials_provider_arn,
  # remove this line if auth is not needed
    instance_type=instance_type,
    instance_count=instance_count,
    output_path=output_path,
    max_run=max_run_time
)
```

5. Mulai pekerjaan pelatihan Anda dengan meneleponestimator.fitdengan nama pekerjaan Anda dan jalur input sebagai parameter, seperti yang ditunjukkan pada contoh kode berikut.

```
input_path = "s3://your-input-bucket/your-input-path"
job_name = "your-job-name"

estimator.fit(
    inputs=input_path,
    job_name=job_name
)
```

Gunakan registri Docker yang memerlukan otentikasi untuk pelatihan

Jika registri Docker Anda memerlukan otentikasi, Anda harus membuatAWS Lambdafungsi yang menyediakan kredensyal akses ke SageMaker. Kemudian, buat pekerjaan pelatihan dan berikan ARN fungsi Lambda ini di dalamcreate_training_jobAPI. Terakhir, Anda dapat secara opsional membuat titik akhir VPC antarmuka sehingga VPC Anda dapat berkomunikasi dengan fungsi Lambda Anda tanpa mengirim lalu lintas melalui internet. Panduan berikut menunjukkan cara membuat fungsi Lambda, menetapkan peran yang benar dan membuat antarmuka VPC endpoint.

Buat fungsi Lambda

Buat sebuahAWS Lambdafungsi yang meneruskan kredensyal akses ke SageMaker dan mengembalikan respon. Contoh kode berikut menciptakan fungsi handler Lambda, sebagai berikut.

```
def handler(event, context):
   response = {
      "Credentials": {"Username": "username", "Password": "password"}
   }
   return response
```

Jenis otentikasi yang digunakan untuk mengatur registri Docker pribadi Anda menentukan isi respons yang dikembalikan oleh fungsi Lambda Anda sebagai berikut.

- Jika registri Docker pribadi Anda menggunakan otentikasi dasar, fungsi Lambda akan mengembalikan nama pengguna dan kata sandi yang diperlukan untuk mengautentikasi ke registri.
- Jika registri Docker pribadi Anda menggunakanotentikasi token pembawa, nama pengguna dan kata sandi dikirim ke server otorisasi Anda, yang kemudian mengembalikan token pembawa. Token ini kemudian digunakan untuk mengautentikasi ke registri Docker pribadi Anda.

> **Note**
>
> Jika Anda memiliki lebih dari satu fungsi Lambda untuk pendaftar Anda di akun yang sama, dan peran pelaksanaannya sama untuk pekerjaan pelatihan Anda, maka pekerjaan pelatihan untuk registri seseorang akan memiliki akses ke fungsi Lambda untuk pendaftar lain.

Berikan izin peran yang benar ke fungsi Lambda Anda

TheIamRoleyang Anda gunakan di`create_training_job`API harus memiliki izinAWS Lambdafungsi. Contoh kode berikut menunjukkan cara melakukan tugas-tugas IAM`myLambdaFunction`.

```
{
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": [
```

```
        "arn:aws:lambda:*:*:function:*myLambdaFunction*"
    ]
}
```

Untuk informasi tentang cara melakukan kebijakan izin peran, lihatMengubah kebijakan izin peran (konsol)diAWSPanduan Pengguna Identity and Access Management.

> **Note**
>
> Peran IAM dengan terlampirAmazonSageMakerFullAccesskebijakan terkelola memiliki izin untuk memanggil fungsi Lambda apa pun dengan”SageMaker“Dalam namanya.

Buat titik akhir VPC antarmuka untuk Lambda

Jika Anda membuat titik akhir antarmuka, VPC Amazon Anda dapat berkomunikasi dengan fungsi Lambda Anda tanpa mengirim lalu lintas melalui internet. Untuk informasi selengkapnya, lihatMengonfigurasi VPC endpoint antarmuka untuk LambdadiAWS LambdaPanduan Pengembang.

Setelah titik akhir antarmuka Anda dibuat, SageMaker pelatihan akan memanggil fungsi Lambda Anda dengan mengirimkan permintaan melalui VPC Anda`lambda.region.amazonaws.com`. Jika Anda memilihAktifkan Nama DNSsaat Anda membuat titik akhir antarmuka Anda,Amazon Route 53merutekan panggilan ke titik akhir antarmuka Lambda. Jika Anda menggunakan penyedia DNS yang berbeda, Anda harus memetakan`lambda.region.amazonaws.com`, ke titik akhir antarmuka Lambda Anda.

Mengadaptasi Wadah Inferensi Anda Sendiri

Jika Anda tidak dapat menggunakan gambar apa pun yang tercantum di Gunakan gambar SageMaker Docker yang dibuat sebelumnya Amazon SageMaker untuk kasus penggunaan Anda, Anda dapat membuat wadah Docker Anda sendiri dan menggunakannya di dalam SageMaker untuk pelatihan dan inferensi. Agar kompatibel SageMaker, wadah Anda harus memiliki karakteristik sebagai berikut:

- Container Anda harus memiliki daftar server web di port`8080`.
- Container Anda harus menerima `POST` permintaan ke titik akhir `/invocations` dan `/ping` real-time. Permintaan yang Anda kirim ke titik akhir ini harus dikembalikan dengan 60 detik dan memiliki ukuran maksimum 6 MB.

Untuk informasi selengkapnya dan contoh cara membuat wadah Docker Anda sendiri untuk pelatihan dan inferensi SageMaker, lihat Membangun wadah algoritme Anda sendiri.

Panduan berikut menunjukkan cara menggunakan `JupyterLab` ruang dengan Amazon SageMaker Studio Classic untuk mengadaptasi wadah inferensi agar berfungsi dengan SageMaker hosting. Contohnya menggunakan server NGINX web, Gunicorn sebagai antarmuka gateway server Python web, dan Flask sebagai kerangka kerja aplikasi web. Anda dapat menggunakan aplikasi yang berbeda untuk menyesuaikan wadah Anda selama memenuhi persyaratan yang tercantum sebelumnya. Untuk informasi selengkapnya tentang menggunakan kode inferensi Anda sendiri, lihatGunakan Kode Inferensi Anda Sendiri dengan Layanan Hosting.

Sesuaikan wadah inferensi Anda

Gunakan langkah-langkah berikut untuk menyesuaikan wadah inferensi Anda sendiri untuk bekerja dengan SageMaker hosting. Contoh yang ditunjukkan dalam langkah-langkah berikut menggunakan model Dinamakan Entity Recognition (NER) pra-terlatih yang menggunakan pustaka pemrosesan bahasa alami (NLP) SPacy untuk `Python` dan berikut ini:

- A Dockerfile untuk membangun wadah yang berisi NER model.
- Skrip inferensi untuk melayani model. NER

Jika Anda mengadaptasi contoh ini untuk kasus penggunaan Anda, Anda harus menggunakan skrip a Dockerfile dan inferensi yang diperlukan untuk menerapkan dan melayani model Anda.

1. Buat JupyterLab ruang dengan Amazon SageMaker Studio Classic (opsional).

   Anda dapat menggunakan notebook apa pun untuk menjalankan skrip untuk menyesuaikan wadah inferensi Anda dengan SageMaker hosting. Contoh ini menunjukkan kepada Anda cara menggunakan JupyterLab ruang dalam Amazon SageMaker Studio Classic untuk meluncurkan JupyterLab aplikasi yang dilengkapi dengan gambar SageMaker Distribusi. Untuk informasi selengkapnya, lihat SageMaker JupyterLab.
2. Unggah Docker file dan skrip inferensi.

   1. Buat folder baru di direktori home Anda. Jika Anda menggunakanJupyterLab, di pojok kiri atas, pilih ikon Folder Baru, dan masukkan nama folder untuk memuat. Dockerfile Dalam contoh ini, folder dipanggil`docker_test_folder`.
   2. Unggah file Dockerfile teks ke folder baru Anda. Berikut ini adalah contoh Dockerfile yang membuat Docker wadah dengan model Named Entity Recognition (NER) yang telah dilatih

sebelumnya dari SPacy, aplikasi dan variabel lingkungan yang diperlukan untuk menjalankan contoh:

```
FROM python:3.8

RUN apt-get -y update && apt-get install -y --no-install-recommends \
         wget \
         python3 \
         nginx \
         ca-certificates \
    && rm -rf /var/lib/apt/lists/*

RUN wget https://bootstrap.pypa.io/get-pip.py && python3 get-pip.py && \
    pip install flask gevent gunicorn && \
        rm -rf /root/.cache

#pre-trained model package installation
RUN pip install spacy
RUN python -m spacy download en


# Set environment variables
ENV PYTHONUNBUFFERED=TRUE
ENV PYTHONDONTWRITEBYTECODE=TRUE
ENV PATH="/opt/program:${PATH}"

COPY NER /opt/program
WORKDIR /opt/program
```

Dalam contoh kode sebelumnya, variabel lingkungan `PYTHONUNBUFFERED` Python mencegah buffering aliran output standar, yang memungkinkan pengiriman log lebih cepat ke pengguna. Variabel lingkungan `PYTHONDONTWRITEBYTECODE` Python mencegah penulisan `.pyc` file bytecode yang dikompilasi, yang tidak diperlukan untuk kasus penggunaan ini. Variabel lingkungan `PATH` digunakan untuk mengidentifikasi lokasi `train` dan `serve` program saat wadah dipanggil.

3. Buat direktori baru di dalam folder baru Anda untuk berisi skrip untuk melayani model Anda. Contoh ini menggunakan direktori yang disebut`NER`, yang berisi skrip berikut yang diperlukan untuk menjalankan contoh ini:

   - `predictor.py`— Python Skrip yang berisi logika untuk memuat dan melakukan inferensi dengan model Anda.

- `nginx.conf`— Skrip untuk mengkonfigurasi server web.
- `serve`— Skrip yang memulai server inferensi.
- `wsgi.py`— Naskah pembantu untuk melayani model.

> **Important**
>
> Jika Anda menyalin skrip inferensi ke buku catatan yang diakhiri `.ipynb` dan mengganti namanya, skrip Anda mungkin berisi karakter pemformatan yang akan mencegah titik akhir Anda diterapkan. Sebagai gantinya, buat file teks dan ganti namanya.

4. Unggah skrip untuk membuat model Anda tersedia untuk inferensi. Berikut ini adalah contoh script `predictor.py` yang disebut yang menggunakan Flask untuk menyediakan `/ping` dan `/invocations` endpoint:

```
from flask import Flask
import flask
import spacy
import os
import json
import logging

#Load in model
nlp = spacy.load('en_core_web_sm')
#If you plan to use a your own model artifacts,
#your model artifacts should be stored in /opt/ml/model/


# The flask app for serving predictions
app = Flask(__name__)
@app.route('/ping', methods=['GET'])
def ping():
    # Check if the classifier was loaded correctly
    health = nlp is not None
    status = 200 if health else 404
    return flask.Response(response= '\n', status=status, mimetype='application/
json')


@app.route('/invocations', methods=['POST'])
def transformation():
```

```
    #Process input
    input_json = flask.request.get_json()
    resp = input_json['input']

    #NER
    doc = nlp(resp)
    entities = [(X.text, X.label_) for X in doc.ents]

    # Transform predictions to JSON
    result = {
        'output': entities
        }

    resultjson = json.dumps(result)
    return flask.Response(response=resultjson, status=200, mimetype='application/
json')
```

`/ping`Titik akhir dalam contoh skrip sebelumnya mengembalikan kode status `200` jika model dimuat dengan benar, dan `404` jika model dimuat dengan tidak benar. `/invocations`Titik akhir memproses permintaan yang diformatJSON, mengekstrak bidang input, dan menggunakan NER model untuk mengidentifikasi dan menyimpan entitas dalam entitas variabel. FlaskAplikasi mengembalikan respon yang berisi entitas ini. Untuk informasi lebih lanjut tentang permintaan kesehatan yang diperlukan ini, lihatBagaimana Kontainer Anda Harus Menanggapi Permintaan Pemeriksaan Kesehatan (Ping).

5. Unggah skrip untuk memulai server inferensi. Contoh skrip berikut memanggil `serve` menggunakan Gunicorn sebagai server aplikasi, dan Nginx sebagai server web:

```
#!/usr/bin/env python

# This file implements the scoring service shell. You don't necessarily need to
 modify it for various
# algorithms. It starts nginx and gunicorn with the correct configurations and
 then simply waits until
# gunicorn exits.
#
# The flask server is specified to be the app object in wsgi.py
#
# We set the following parameters:
#
# Parameter                Environment Variable              Default Value
```

```
# ---------               --------------------             -------------
# number of workers       MODEL_SERVER_WORKERS             the number of CPU
 cores
# timeout                 MODEL_SERVER_TIMEOUT             60 seconds

import multiprocessing
import os
import signal
import subprocess
import sys

cpu_count = multiprocessing.cpu_count()

model_server_timeout = os.environ.get('MODEL_SERVER_TIMEOUT', 60)
model_server_workers = int(os.environ.get('MODEL_SERVER_WORKERS', cpu_count))

def sigterm_handler(nginx_pid, gunicorn_pid):
    try:
        os.kill(nginx_pid, signal.SIGQUIT)
    except OSError:
        pass
    try:
        os.kill(gunicorn_pid, signal.SIGTERM)
    except OSError:
        pass

    sys.exit(0)

def start_server():
    print('Starting the inference server with {}
 workers.'.format(model_server_workers))


    # link the log streams to stdout/err so they will be logged to the container
 logs
    subprocess.check_call(['ln', '-sf', '/dev/stdout', '/var/log/nginx/
access.log'])
    subprocess.check_call(['ln', '-sf', '/dev/stderr', '/var/log/nginx/
error.log'])

    nginx = subprocess.Popen(['nginx', '-c', '/opt/program/nginx.conf'])
    gunicorn = subprocess.Popen(['gunicorn',
                                 '--timeout', str(model_server_timeout),
                                 '-k', 'sync',
```

```
                                    '-b', 'unix:/tmp/gunicorn.sock',
                                    '-w', str(model_server_workers),
                                    'wsgi:app'])

    signal.signal(signal.SIGTERM, lambda a, b: sigterm_handler(nginx.pid,
 gunicorn.pid))

    # Exit the inference server upon exit of either subprocess
    pids = set([nginx.pid, gunicorn.pid])
    while True:
        pid, _ = os.wait()
        if pid in pids:
            break

    sigterm_handler(nginx.pid, gunicorn.pid)
    print('Inference server exiting')

# The main routine to invoke the start function.

if __name__ == '__main__':
    start_server()
```

Contoh skrip sebelumnya mendefinisikan fungsi penangan sinyal`sigterm_handler`, yang mematikan Nginx dan Gunicorn sub-proses ketika menerima sinyal. `SIGTERM` `start_server`Fungsi memulai penangan sinyal, memulai dan memantau dan Gunicorn sub-proses, Nginx dan menangkap aliran log.

6. Unggah skrip untuk mengkonfigurasi server web Anda. Contoh skrip berikut disebut`nginx.conf`, mengkonfigurasi server Nginx web menggunakan Gunicorn sebagai server aplikasi untuk melayani model Anda untuk inferensi:

```
worker_processes 1;
daemon off; # Prevent forking


pid /tmp/nginx.pid;
error_log /var/log/nginx/error.log;

events {
  # defaults
}

http {
```

```
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /var/log/nginx/access.log combined;

  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }

  server {
    listen 8080 deferred;
    client_max_body_size 5m;

    keepalive_timeout 5;
    proxy_read_timeout 1200s;

    location ~ ^/(ping|invocations) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_pass http://gunicorn;
    }

    location / {
      return 404 "{}";
    }
  }
}
```

Contoh skrip sebelumnya mengkonfigurasi Nginx untuk berjalan di latar depan, menetapkan lokasi untuk menangkap`error_log`, dan mendefinisikan`upstream` sebagai kaus kaki soket Gunicorn server. Server mengkonfigurasi blok server untuk mendengarkan pada port`8080`, menetapkan batas pada ukuran badan permintaan klien dan nilai batas waktu. Server memblokir, meneruskan permintaan yang berisi salah satu `/ping` atau `/invocations` jalur ke Gunicorn`server http://gunicorn`, dan mengembalikan `404` kesalahan untuk jalur lain.

7. Unggah skrip lain yang diperlukan untuk melayani model Anda. Contoh ini memerlukan contoh script berikut yang dipanggil `wsgi.py` untuk membantu Gunicorn menemukan aplikasi Anda:

```
import predictor as myapp

# This is just a simple wrapper for gunicorn to find your app.
```

```
# If you want to change the algorithm file, simply change "predictor" above to
 the
# new file.

app = myapp.app
```

Dari folder`docker_test_folder`, struktur direktori Anda harus berisi Dockerfile dan folderNER. NERFolder harus berisi file`nginx.conf`,`predictor.py`,`serve`, dan `wsgi.py` sebagai berikut:

```
/docker_test_folder
|--Dockerfile
|--NER
|  |--nginx.conf
|  |--predictor.py
|  |--serve
|  |--wsgi.py
```

3. Bangun wadah Anda sendiri.

   Dari folder`docker_test_folder`, buat Docker wadah Anda. Perintah contoh berikut akan membangun Docker wadah yang dikonfigurasi dalam AndaDockerfile:

   ```
   ! docker build -t byo-container-test .
   ```

   Perintah sebelumnya akan membangun wadah yang disebut `byo-container-test` di direktori kerja saat ini. Untuk informasi selengkapnya tentang parameter Docker build, lihat Argumen build.

   > **Note**
   >
   > Jika Anda mendapatkan pesan kesalahan berikut yang Docker tidak dapat menemukanDockerfile, pastikan Dockerfile memiliki nama yang benar dan telah disimpan ke direktori.
   >
   > ```
   > unable to prepare context: unable to evaluate symlinks in Dockerfile path:
   > ```

```
lstat /home/ec2-user/SageMaker/docker_test_folder/Dockerfile: no such file
 or directory
```

Dockermencari file yang secara khusus dipanggil Dockerfile tanpa ekstensi apa pun dalam direktori saat ini. Jika Anda menamakannya sesuatu yang lain, Anda dapat meneruskan nama file secara manual dengan flag -f. Misalnya, jika Anda menamai Dockerfile asDockerfile-text.txt, buat Docker penampung Anda menggunakan `-f` bendera yang diikuti oleh file Anda sebagai berikut:

```
! docker build -t byo-container-test -f Dockerfile-text.txt .
```

4. Dorong Docker Gambar Anda ke Amazon Elastic Container Registry (Amazon ECR)

Di sel notebook, dorong Docker gambar Anda ke ECR. Contoh kode berikut menunjukkan cara membangun container Anda secara lokal, login dan mendorongnya ke ECR:

```
%%sh
# Name of algo -> ECR
algorithm_name=sm-pretrained-spacy

#make serve executable
chmod +x NER/serve
account=$(aws sts get-caller-identity --query Account --output text)
# Region, defaults to us-west-2
region=$(aws configure get region)
region=${region:-us-east-1}
fullname="${account}.dkr.ecr.${region}.amazonaws.com/${algorithm_name}:latest"
# If the repository doesn't exist in ECR, create it.
aws ecr describe-repositories --repository-names "${algorithm_name}" > /dev/null
 2>&1
if [ $? -ne 0 ]
then
    aws ecr create-repository --repository-name "${algorithm_name}" > /dev/nullfi
# Get the login command from ECR and execute it directly
aws ecr get-login-password --region ${region}|docker login --username AWS --
password-stdin ${fullname}
# Build the docker image locally with the image name and then push it to ECR
# with the full name.

docker build  -t ${algorithm_name} .
docker tag ${algorithm_name} ${fullname}
```

```
docker push ${fullname}
```

Dalam contoh sebelumnya menunjukkan bagaimana melakukan langkah-langkah berikut yang diperlukan untuk mendorong contoh wadah Docker ke ECR:

a. Tentukan nama algoritma sebagai`sm-pretrained-spacy`.

b. Buat `serve` file di dalam NER folder dapat dieksekusi.

c. MengaturWilayah AWS.

d. Buat ECR jika belum ada.

e. Login ke ECR.

f. Bangun Docker wadah secara lokal.

g. Dorong Docker gambar ke ECR.

5. Siapkan SageMaker klien

Jika Anda ingin menggunakan layanan SageMaker hosting untuk inferensi, Anda harus membuat model, membuat konfigurasi titik akhir, dan membuat titik akhir. Untuk mendapatkan kesimpulan dari titik akhir Anda, Anda dapat menggunakan klien SageMaker boto3 Runtime untuk memanggil titik akhir Anda. Kode berikut menunjukkan cara mengatur SageMaker klien dan klien SageMaker Runtime menggunakan klien SageMaker boto3:

```
import boto3
from sagemaker import get_execution_role

sm_client = boto3.client(service_name='sagemaker')
runtime_sm_client = boto3.client(service_name='sagemaker-runtime')

account_id = boto3.client('sts').get_caller_identity()['Account']
region = boto3.Session().region_name

#used to store model artifacts which SageMaker will extract to /opt/ml/model in the
 container,
#in this example case we will not be making use of S3 to store the model artifacts
#s3_bucket = '<S3Bucket>'

role = get_execution_role()
```

Dalam contoh kode sebelumnya, bucket Amazon S3 tidak digunakan, tetapi dimasukkan sebagai komentar untuk menunjukkan cara menyimpan artefak model.

Jika Anda menerima kesalahan izin setelah menjalankan contoh kode sebelumnya, Anda mungkin perlu menambahkan izin ke peran IAM Anda. Untuk informasi selengkapnya tentang peran IAM, lihatManajer SageMaker Peran Amazon. Untuk informasi selengkapnya tentang menambahkan izin ke peran Anda saat ini, lihatAWSKebijakan Terkelola untuk Amazon SageMaker.

6. Buat model Anda.

   Jika Anda ingin menggunakan layanan SageMaker hosting untuk inferensi, Anda harus membuat model di SageMaker. Contoh kode berikut menunjukkan cara membuat spaCy NER model di dalam SageMaker:

```
from time import gmtime, strftime

model_name = 'spacy-nermodel-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
# MODEL S3 URL containing model atrifacts as either model.tar.gz or extracted
 artifacts.
# Here we are not
#model_url = 's3://{}/spacy/'.format(s3_bucket)

container = '{}.dkr.ecr.{}.amazonaws.com/sm-pretrained-
spacy:latest'.format(account_id, region)
instance_type = 'ml.c5d.18xlarge'

print('Model name: ' + model_name)
#print('Model data Url: ' + model_url)
print('Container image: ' + container)

container = {
'Image': container
}

create_model_response = sm_client.create_model(
    ModelName = model_name,
    ExecutionRoleArn = role,
    Containers = [container])

print("Model Arn: " + create_model_response['ModelArn'])
```

Contoh kode sebelumnya menunjukkan cara mendefinisikan penggunaan `s3_bucket` jika Anda `model_url` menggunakan bucket Amazon S3 dari komentar di Langkah 5, dan mendefinisikan URI ECR untuk image container. Contoh kode sebelumnya mendefinisikan `ml.c5d.18xlarge` sebagai tipe instance. Anda juga dapat memilih jenis instance yang berbeda. Untuk informasi selengkapnya tentang jenis instans yang tersedia, lihat jenis instans Amazon EC2.

Dalam contoh kode sebelumnya, Poin `Image` utama ke URI gambar kontainer. `create_model_response`Definisi menggunakan `create_model method` untuk membuat model, dan mengembalikan nama model, peran dan daftar yang berisi informasi kontainer.

Contoh output dari script sebelumnya berikut:

```
Model name: spacy-nermodel-YYYY-MM-DD-HH-MM-SS
Model data Url: s3://spacy-sagemaker-us-east-1-bucket/spacy/
Container image: 123456789012.dkr.ecr.us-east-2.amazonaws.com/sm-pretrained-
spacy:latest
Model Arn: arn:aws:sagemaker:us-east-2:123456789012:model/spacy-nermodel-YYYY-MM-
DD-HH-MM-SS
```

7. a. Konfigurasikan dan buat titik akhir

   Untuk menggunakan SageMaker hosting untuk inferensi, Anda juga harus mengkonfigurasi dan membuat titik akhir. SageMaker akan menggunakan titik akhir ini untuk inferensi. Contoh konfigurasi berikut menunjukkan cara membuat dan mengonfigurasi titik akhir dengan jenis instans dan nama model yang Anda tentukan sebelumnya:

```
endpoint_config_name = 'spacy-ner-config' + strftime("%Y-%m-%d-%H-%M-%S",
 gmtime())
print('Endpoint config name: ' + endpoint_config_name)

create_endpoint_config_response = sm_client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType': instance_type,
        'InitialInstanceCount': 1,
        'InitialVariantWeight': 1,
        'ModelName': model_name,
        'VariantName': 'AllTraffic'}])
```

```
print("Endpoint config Arn: " +
 create_endpoint_config_response['EndpointConfigArn'])
```

Dalam contoh konfigurasi sebelumnya, `create_endpoint_config_response` mengaitkan `model_name` dengan nama konfigurasi titik akhir unik `endpoint_config_name` yang dibuat dengan stempel waktu.

Contoh output dari script sebelumnya berikut:

```
Endpoint config name: spacy-ner-configYYYY-MM-DD-HH-MM-SS
Endpoint config Arn: arn:aws:sagemaker:us-east-2:123456789012:endpoint-config/
spacy-ner-config-MM-DD-HH-MM-SS
```

Untuk informasi selengkapnya tentang kesalahan titik akhir, lihat Mengapa SageMaker titik akhir Amazon saya masuk ke status gagal saat saya membuat atau memperbarui titik akhir?

b. Buat titik akhir dan tunggu titik akhir dalam layanan.

Contoh kode berikut membuat titik akhir menggunakan konfigurasi dari contoh konfigurasi sebelumnya dan menerapkan model:

```
%%time

import time

endpoint_name = 'spacy-ner-endpoint' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print('Endpoint name: ' + endpoint_name)

create_endpoint_response = sm_client.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name)
print('Endpoint Arn: ' + create_endpoint_response['EndpointArn'])

resp = sm_client.describe_endpoint(EndpointName=endpoint_name)
status = resp['EndpointStatus']
print("Endpoint Status: " + status)

print('Waiting for {} endpoint to be in service...'.format(endpoint_name))
waiter = sm_client.get_waiter('endpoint_in_service')
waiter.wait(EndpointName=endpoint_name)
```

Dalam contoh kode sebelumnya, `create_endpoint` metode membuat titik akhir dengan nama titik akhir yang dihasilkan yang dibuat dalam contoh kode sebelumnya, dan mencetak Nama Sumber Daya Amazon dari titik akhir. `describe_endpoint`Metode mengembalikan informasi tentang endpoint dan statusnya. Seorang SageMaker pelayan menunggu titik akhir dalam pelayanan.

8. Uji titik akhir Anda.

Setelah titik akhir Anda dalam layanan, kirim permintaan pemanggilan ke titik akhir Anda. Contoh kode berikut menunjukkan cara mengirim permintaan pengujian ke titik akhir Anda:

```
import json
content_type = "application/json"
request_body = {"input": "This is a test with NER in America with \
    Amazon and Microsoft in Seattle, writing random stuff."}

#Serialize data for endpoint
#data = json.loads(json.dumps(request_body))
payload = json.dumps(request_body)

#Endpoint invocation
response = runtime_sm_client.invoke_endpoint(
EndpointName=endpoint_name,
ContentType=content_type,
Body=payload)

#Parse results
result = json.loads(response['Body'].read().decode())['output']
result
```

Dalam contoh kode sebelumnya, metode ini `json.dumps` membuat serial `request_body` menjadi string yang diformat dalam JSON dan menyimpannya dalam payload variabel. Kemudian klien SageMaker Runtime menggunakan metode titik akhir pemanggilan untuk mengirim payload ke titik akhir Anda. Hasilnya berisi respons dari titik akhir Anda setelah mengekstraksi bidang output.

Contoh kode sebelumnya harus mengembalikan output berikut:

```
[['NER', 'ORG'],
 ['America', 'GPE'],
 ['Amazon', 'ORG'],
```

```
 ['Microsoft', 'ORG'],
 ['Seattle', 'GPE']]
```

9. Hapus titik akhir Anda

   Setelah Anda menyelesaikan pemanggilan, hapus titik akhir Anda untuk menghemat sumber daya. Contoh kode berikut menunjukkan cara menghapus titik akhir Anda:

```
sm_client.delete_endpoint(EndpointName=endpoint_name)
sm_client.delete_endpoint_config(EndpointConfigName=endpoint_config_name)
sm_client.delete_model(ModelName=model_name)
```

   Untuk buku catatan lengkap yang berisi kode dalam contoh ini, lihat BYOC-Single-model.

Memecahkan masalah penerapan kontainer

Jika titik akhir Anda tidak diterapkan, periksa log Amazon CloudWatch Events sebagai berikut:

1. Dari panel navigasi SageMaker konsol https://console.aws.amazon.com/sagemaker/, pilih Inferensi.
2. Di bawah Inferensi, pilih Endpoints.
3. Temukan titik akhir Anda di bawah Nama, dan klik pada nama titik akhir. Dalam contoh ini, nama akan mengikuti konvensi penamaan`spacy-ner-config`YYYY-MM-DD-HH-MM-SS.
4. Di bawah Ringkasan titik akhir, pilih tautan di bawah Log wadah Model.
5. Pilih aliran Log terbaru di kotak Aliran log.

Gunakan daftar berikut untuk memecahkan masalah penerapan titik akhir Anda. Jika Anda memerlukan bantuan lebih lanjut, hubungi AWSSupport atau Forum AWS Pengembang untuk Amazon SageMaker.

Topik

- Kesalahan nama
- Kuota tidak mencukupi
- Kesalahan waktu habis hulu

Kesalahan nama

Jika log menyatakan`NameError: name 'null' is not defined`, pastikan skrip Anda tidak dibuat di buku catatan yang diakhiri `.ipnyb` dan kemudian diganti namanya menjadi nama file lain seperti. `Dockerfile` Saat Anda membuat buku catatan, memformat karakter dapat mencegah titik akhir Anda diterapkan. Jika Anda mendapatkan kesalahan ini dan Anda mengubah skrip untuk memperbaikinya, Anda mungkin perlu me-restart kernel Anda agar perubahan diterapkan.

Kuota tidak mencukupi

Jika Anda menerima `ResourceLimitExceeded` kesalahan, Anda harus meminta kuota tambahan sebagai berikut:

Minta peningkatan AWS Service Quotas

1. Ambil nama instans, kuota saat ini dan kuota yang diperlukan dari pesan kesalahan di layar. Misalnya, dalam contoh kesalahan berikut:

   - Nama instance-nya adalah`ml.c5d.18xlarge`.
   - Kuota saat ini dari nomor berikut `current utilization` adalah`1 instances`.
   - Tambahan kuota yang diperlukan dari nomor berikut `request delta` adalah`1 instances`.

   Kesalahan sampel berikut:

   ```
   ResourceLimitExceeded: An error occurred (ResourceLimitExceeded)
   when calling the CreateEndpoint operation: The account-level service limit
   'ml.c5d.18xlarge for endpoint usage' is 1 Instances, with current utilization
   of 1 Instances and a request delta of 1 Instances. Please use AWS Service Quotas
   to request an increase for this quota. If  AWS Service Quotas is not available,
   contact AWS support to request an increase for this quota.
   ```

2. Masuk ke AWS Management Console dan buka konsol Service Quotas.
3. Di panel navigasi, di bawah Kelola kuota, masukkan Amazon. SageMaker
4. Pilih Lihat kuota.
5. Di bilah pencarian di bawah Kuota layanan, masukkan nama instance dari Langkah 1. Misalnya, menggunakan informasi yang terkandung dalam pesan kesalahan dari Langkah 1, masukan`ml.c5d.18xlarge`.

6. Pilih nama Kuota yang muncul di sebelah nama instans Anda dan diakhiri dengan untuk penggunaan titik akhir. Misalnya, menggunakan informasi yang terkandung dalam pesan kesalahan dari Langkah 1, pilih `ml.g5.12xlarge` untuk penggunaan titik akhir.
7. Pilih Permintaan peningkatan di tingkat akun.
8. Di bawah Meningkatkan nilai kuota, masukkan kuota yang diperlukan dari informasi yang diberikan dalam pesan kesalahan dari Langkah 1. Masukan total `current utilization` dan`request delta`. Dalam contoh kesalahan sebelumnya, `current utilization` is`1 Instances`, dan `request delta` is`1 Instances`. Dalam contoh ini, minta kuota 2 untuk memasok kuota yang diperlukan.
9. Pilih Minta.
10. Pilih Riwayat permintaan kuota dari panel navigasi.
11. Saat Status berubah dari Tertunda menjadi Disetujui, jalankan kembali pekerjaan Anda. Anda mungkin perlu menyegarkan browser Anda untuk melihat perubahannya.

Untuk informasi selengkapnya tentang meminta peningkatan kuota, lihat Meminta kenaikan kuota.

Kesalahan waktu habis hulu

Jika Anda menerima `upstream timed out (110: Connection timed out)` kesalahan, Anda dapat mencoba yang berikut ini:

- Kurangi latensi wadah atau tingkatkan batas waktu kontainer. SageMaker mengharuskan wadah Anda menanggapi permintaan dalam waktu 60 detik.
- Tingkatkan jumlah waktu sebelum server web Anda menunggu respons dari model.

Untuk informasi lebih lanjut tentang kesalahan waktu habis, lihat Bagaimana cara mengatasi kesalahan SageMaker inferensi Amazon “waktu hulu habis (110: Waktu koneksi habis) saat membaca header respons dari hulu”?

Buat wadah dengan algoritme dan model Anda sendiri

Jika tidak ada SageMaker kontainer yang ada yang memenuhi kebutuhan Anda dan Anda tidak memiliki wadah sendiri, Anda mungkin perlu membuat wadah Docker baru. Bagian berikut menunjukkan cara membuat kontainer Docker dengan algoritma pelatihan dan inferensi Anda untuk digunakan. SageMaker

Topik

Gunakan Algoritma Pelatihan Anda Sendiri

Bagian ini menjelaskan bagaimana Amazon SageMaker berinteraksi dengan wadah Docker yang menjalankan algoritme pelatihan khusus Anda. Gunakan informasi ini untuk menulis kode pelatihan dan membuat gambar Docker untuk algoritme pelatihan Anda.

Topik

Bagaimana Amazon SageMaker Menjalankan Gambar Pelatihan Anda

Anda dapat menggunakan skrip entrypoint khusus untuk mengotomatiskan infrastruktur untuk melatih di lingkungan produksi. Jika Anda meneruskan skrip entrypoint Anda ke wadah Docker Anda, Anda juga dapat menjalankannya sebagai skrip mandiri tanpa membangun kembali gambar Anda. SageMakermemproses gambar pelatihan Anda menggunakan skrip titik masuk kontainer Docker.

Bagian ini menunjukkan cara menggunakan entrypoint khusus tanpa menggunakan toolkit pelatihan. Jika Anda ingin menggunakan entrypoint khusus tetapi tidak terbiasa dengan cara mengonfigurasi wadah Docker secara manual, kami sarankan Anda menggunakan perpustakaan toolkit SageMaker pelatihan sebagai gantinya. Untuk informasi selengkapnya tentang cara menggunakan toolkit pelatihan, lihatMengadaptasi wadah pelatihan Anda sendiri.

Secara default, SageMaker cari skrip yang dipanggil `train` di dalam wadah Anda. Anda juga dapat secara manual memberikan entrypoint kustom Anda sendiri dengan menggunakan `ContainerArguments` dan `ContainerEntrypoint` parameter API. AlgorithmSpecification

Anda memiliki dua opsi berikut untuk mengonfigurasi wadah Docker Anda secara manual untuk menjalankan gambar Anda.

- Gunakan CreateTrainingJobAPI dan wadah Docker dengan instruksi entrypoint yang terdapat di dalamnya.
- Gunakan `CreateTrainingJob` API, dan teruskan skrip pelatihan Anda dari luar wadah Docker Anda.

Jika Anda meneruskan skrip pelatihan Anda dari luar wadah Docker Anda, Anda tidak perlu membangun kembali wadah Docker saat Anda memperbarui skrip Anda. Anda juga dapat menggunakan beberapa skrip berbeda untuk dijalankan di wadah yang sama.

Script entrypoint Anda harus berisi kode pelatihan untuk gambar Anda. Jika Anda menggunakan `source_dir` parameter opsional di dalam estimator, parameter tersebut harus mereferensikan jalur Amazon S3 relatif ke folder yang berisi skrip titik masuk Anda. Anda dapat mereferensikan beberapa file menggunakan `source_dir` parameter. Jika Anda tidak menggunakan`source_dir`, Anda dapat menentukan titik masuk menggunakan parameter. `entry_point` Untuk contoh skrip entrypoint kustom yang berisi estimator, lihat Membawa Model Anda Sendiri dengan SageMaker Mode Skrip.

Jalankan pekerjaan pelatihan dengan skrip entrypoint yang dibundel di dalam wadah Docker

SageMaker dapat menjalankan skrip titik masuk yang dibundel di dalam wadah Docker Anda.

- Secara default, Amazon SageMaker menjalankan penampung berikut.

```
docker run image train
```

- SageMaker mengganti pernyataan CMD default dalam wadah dengan menentukan `train` argumen setelah nama gambar. Dalam wadah Docker Anda, gunakan `exec` bentuk `ENTRYPOINT` instruksi berikut.

```
ENTRYPOINT ["executable", "param1", "param2", ...]
```

  Contoh berikut menunjukkan bagaimana menentukan instruksi entrypoint python yang disebut. `k-means-algorithm.py`

```
ENTRYPOINT ["python", "k-means-algorithm.py"]
```

  `exec`Bentuk `ENTRYPOINT` instruksi memulai eksekusi secara langsung, bukan sebagai anak dari. `/bin/sh` Ini memungkinkannya menerima sinyal seperti `SIGTERM` dan `SIGKILL` dari SageMaker API. Ketentuan berikut berlaku saat menggunakan SageMaker API.

- `CreateTrainingJob`API memiliki kondisi penghentian yang mengarahkan SageMaker untuk menghentikan pelatihan model setelah waktu tertentu.
- Berikut ini menunjukkan `StopTrainingJob`API. API ini mengeluarkan yang setara dengan perintah batas waktu 2 menit untuk menghentikan penampung yang ditentukan dengan anggun. `docker stop`

  ```
  docker stop -t 120
  ```

  Perintah mencoba menghentikan wadah yang sedang berjalan dengan mengirimkan `SIGTERM` sinyal. Setelah batas waktu 2 menit, API mengirim `SIGKILL` dan menghentikan kontainer secara paksa. Jika wadah menangani `SIGTERM` dengan anggun dan keluar dalam waktu 120 detik sejak menerimanya, tidak ada `SIGKILL` yang dikirim.

  Jika Anda ingin akses ke artefak model perantara setelah SageMaker menghentikan pelatihan, tambahkan kode untuk menangani penyimpanan artefak di handler Anda`SIGTERM`.
- Jika Anda berencana menggunakan perangkat GPU untuk pelatihan model, pastikan kontainer Anda `nvidia-docker` kompatibel. Sertakan hanya toolkit CUDA pada wadah; jangan bundel driver NVIDIA dengan gambar. Untuk informasi selengkapnya`nvidia-docker`, lihat NVIDIA/NVIDIA-Docker.
- Anda tidak dapat menggunakan `tini` penginisialisasi sebagai skrip titik masuk Anda dalam SageMaker wadah karena menjadi bingung oleh argumen dan. `train serve`
- `/opt/ml`dan semua subdirektori dicadangkan oleh SageMaker pelatihan. Saat membuat image Docker algoritme Anda, pastikan Anda tidak menempatkan data apa pun yang diperlukan oleh algoritme Anda di direktori ini. Karena jika Anda melakukannya, data mungkin tidak lagi terlihat selama pelatihan.

Untuk menggabungkan skrip shell atau Python Anda di dalam image Docker Anda, atau untuk menyediakan skrip dalam bucket Amazon S3 atau dengan menggunakan (AWS Command Line InterfaceCLI), lanjutkan ke bagian berikut.

Bundel skrip shell Anda dalam wadah Docker

Jika Anda ingin menggabungkan skrip shell khusus di dalam gambar Docker Anda, gunakan langkah-langkah berikut.

1. Salin skrip shell Anda dari direktori kerja Anda ke dalam wadah Docker Anda. Cuplikan kode berikut menyalin skrip entrypoint kustom `custom_entrypoint.sh` dari direktori kerja saat ini ke

wadah Docker yang terletak di. `mydir` Contoh berikut mengasumsikan bahwa image Docker dasar telah diinstal Python.

```
FROM <base-docker-image>:<tag>

# Copy custom entrypoint from current dir to /mydir on container
COPY ./custom_entrypoint.sh /mydir/
```

2. Buat dan dorong wadah Docker ke Amazon Elastic Container Registry (Amazon ECR) dengan mengikuti petunjuk di Mendorong gambar Docker di Panduan Pengguna Amazon ECR.
3. Luncurkan pekerjaan pelatihan dengan menjalankan AWS CLI perintah berikut.

```
aws --region <your-region> sagemaker create-training-job \
--training-job-name <your-training-job-name> \
--role-arn <your-execution-role-arn> \
--algorithm-specification '{ \
    "TrainingInputMode": "File", \
    "TrainingImage": "<your-ecr-image>", \
    "ContainerEntrypoint": ["/bin/sh"], \
    "ContainerArguments": ["/mydir/custom_entrypoint.sh"]}' \
--output-data-config '{"S3OutputPath": "s3://custom-entrypoint-output-bucket/"}' \
--resource-config
 '{"VolumeSizeInGB":10,"InstanceCount":1,"InstanceType":"ml.m5.2xlarge"}' \
--stopping-condition '{"MaxRuntimeInSeconds": 180}'
```

Bundel skrip Python Anda dalam wadah Docker

Untuk menggabungkan skrip Python kustom di dalam gambar Docker Anda, gunakan langkah-langkah berikut.

1. Salin skrip Python Anda dari direktori kerja Anda ke dalam wadah Docker Anda. Cuplikan kode berikut menyalin skrip entrypoint kustom `custom_entrypoint.py` dari direktori kerja saat ini ke wadah Docker yang terletak di. `mydir`

```
FROM <base-docker-image>:<tag>
# Copy custom entrypoint from current dir to /mydir on container
COPY ./custom_entrypoint.py /mydir/
```

2. Luncurkan pekerjaan pelatihan dengan menjalankan AWS CLI perintah berikut.

```
--algorithm-specification '{ \
    "TrainingInputMode": "File", \
    "TrainingImage": "<your-ecr-image>", \
    "ContainerEntrypoint": ["python"], \
    "ContainerArguments": ["/mydir/custom_entrypoint.py"]}' \
```

Jalankan pekerjaan pelatihan dengan skrip titik masuk di luar wadah Docker

Anda dapat menggunakan wadah Docker Anda sendiri untuk pelatihan dan meneruskan skrip titik masuk dari luar wadah Docker. Ada beberapa manfaat untuk menyusun skrip entrypoint Anda di luar wadah. Jika Anda memperbarui skrip entrypoint Anda, Anda tidak perlu membangun kembali wadah Docker. Anda juga dapat menggunakan beberapa skrip berbeda untuk dijalankan di wadah yang sama.

Tentukan lokasi skrip pelatihan Anda menggunakan `ContainerEntrypoint` dan `ContainerArguments` parameter AlgorithmSpecificationAPI. Titik masuk dan argumen ini berperilaku dengan cara yang sama seperti titik masuk dan argumen Docker. Nilai dalam parameter ini mengesampingkan yang sesuai `ENTRYPOINT` atau `CMD` disediakan sebagai bagian dari wadah Docker.

Saat Anda meneruskan skrip entrypoint kustom Anda ke wadah pelatihan Docker Anda, input yang Anda berikan menentukan perilaku penampung.

- Misalnya, jika Anda hanya menyediakan`ContainerEntrypoint`, sintaks permintaan menggunakan CreateTrainingJob API adalah sebagai berikut.

```
{
    "AlgorithmSpecification": {
        "ContainerEntrypoint": ["string"],
        ...
        }
}
```

  Kemudian, backend SageMaker pelatihan menjalankan entrypoint khusus Anda sebagai berikut.

```
docker run --entrypoint <ContainerEntrypoint> image
```

> **Note**
>
> Jika `ContainerEntrypoint` disediakan, backend SageMaker pelatihan menjalankan gambar dengan titik masuk yang diberikan dan mengganti default pada gambar. `ENTRYPOINT`

- Jika Anda hanya menyediakan`ContainerArguments`, SageMaker asumsikan bahwa wadah Docker berisi skrip titik masuk. Sintaks permintaan menggunakan `CreateTrainingJob` API adalah sebagai berikut.

```
{
    "AlgorithmSpecification": {
        "ContainerArguments": ["arg1", "arg2"],
        ...
    }
}
```

Backend SageMaker pelatihan menjalankan entrypoint khusus Anda sebagai berikut.

```
docker run image <ContainerArguments>
```

- Jika Anda memberikan keduanya `ContainerEntrypoint` dan`ContainerArguments`, maka sintaks permintaan menggunakan `CreateTrainingJob` API adalah sebagai berikut.

```
{
    "AlgorithmSpecification": {
        "ContainerEntrypoint": ["string"],
        "ContainerArguments": ["arg1", "arg2"],
        ...
    }
}
```

Backend SageMaker pelatihan menjalankan entrypoint khusus Anda sebagai berikut.

```
docker run --entrypoint <ContainerEntrypoint> image <ContainerArguments>
```

Anda dapat menggunakan `InputDataConfig` sumber apa pun yang didukung di `CreateTrainingJob` API untuk menyediakan skrip titik masuk untuk menjalankan image pelatihan Anda.

Berikan skrip entrypoint Anda di bucket Amazon S3

Untuk menyediakan skrip entrypoint khusus menggunakan bucket S3, gunakan `S3DataSource` parameter DataSourceAPI untuk menentukan lokasi skrip. Jika Anda menggunakan `S3DataSource` parameter, berikut ini diperlukan.

- InputModeHarus dari tipe`File`.
- S3 DataDistributionType harus. `FullyReplicated`

Contoh berikut memiliki skrip bernama custom_entrypoint.sh ditempatkan di jalur ke bucket `s3://<bucket-name>/<bucket prefix>/custom_entrypoint.sh` S3.

```
#!/bin/bash
echo "Running custom_entrypoint.sh"
echo "Hello you have provided the following arguments: " "$@"
```

Selanjutnya, Anda harus mengatur konfigurasi saluran data input untuk menjalankan pekerjaan pelatihan. Lakukan ini baik dengan menggunakan secara AWS CLI langsung atau dengan file JSON.

Konfigurasikan saluran data input menggunakan AWS CLI dengan file JSON

Untuk mengkonfigurasi saluran data input Anda dengan file JSON, gunakan AWS CLI seperti yang ditunjukkan dalam struktur kode berikut. Pastikan bahwa semua bidang berikut menggunakan sintaks permintaan yang ditentukan dalam CreateTrainingJobAPI.

```
// run-my-training-job.json
{
 "AlgorithmSpecification": {
        "ContainerEntrypoint": ["/bin/sh"],
        "ContainerArguments": ["/opt/ml/input/
data/<your_channel_name>/custom_entrypoint.sh"],
         ...
   },
  "InputDataConfig": [
    {
        "ChannelName": "<your_channel_name>",
```

```
        "DataSource": {
            "S3DataSource": {
                "S3DataDistributionType": "FullyReplicated",
                "S3DataType": "S3Prefix",
                "S3Uri": "s3://<bucket-name>/<bucket_prefix>"
            }
        },
        "InputMode": "File",
    },
    ...]
}
```

Selanjutnya, jalankan AWS CLI perintah untuk meluncurkan pekerjaan pelatihan dari file JSON sebagai berikut.

```
aws sagemaker create-training-job --cli-input-json file://run-my-training-job.json
```

Konfigurasikan saluran data input menggunakan AWS CLI langsung

Untuk mengkonfigurasi saluran data input Anda tanpa file JSON, gunakan struktur AWS CLI kode berikut.

```
aws --region <your-region> sagemaker create-training-job \
--training-job-name <your-training-job-name> \
--role-arn <your-execution-role-arn> \
--algorithm-specification '{ \
    "TrainingInputMode": "File", \
    "TrainingImage": "<your-ecr-image>", \
    "ContainerEntrypoint": ["/bin/sh"], \
    "ContainerArguments": ["/opt/ml/input/data/<your_channel_name>/
custom_entrypoint.sh"]}' \
--input-data-config '[{ \
    "ChannelName":"<your_channel_name>", \
    "DataSource":{ \
        "S3DataSource":{ \
            "S3DataType":"S3Prefix", \
            "S3Uri":"s3://<bucket-name>/<bucket_prefix>", \
            "S3DataDistributionType":"FullyReplicated"}}}]' \
--output-data-config '{"S3OutputPath": "s3://custom-entrypoint-output-bucket/"}' \
--resource-config
 '{"VolumeSizeInGB":10,"InstanceCount":1,"InstanceType":"ml.m5.2xlarge"}' \
--stopping-condition '{"MaxRuntimeInSeconds": 180}'
```

Bagaimana Amazon SageMaker Menyediakan Informasi Pelatihan

Bagian ini menjelaskan cara SageMaker membuat informasi pelatihan, seperti data pelatihan, hiperparameter, dan informasi konfigurasi lainnya, tersedia untuk wadah Docker Anda.

Saat Anda mengirim `CreateTrainingJob`permintaan SageMaker untuk memulai pelatihan model, Anda menentukan jalur Amazon Elastic Container Registry (Amazon ECR) Registry ECR) dari image Docker yang berisi algoritme pelatihan. Anda juga menentukan lokasi Amazon Simple Storage Service (Amazon S3) tempat data pelatihan disimpan dan parameter khusus algoritme. SageMaker membuat informasi ini tersedia untuk wadah Docker sehingga algoritma pelatihan Anda dapat menggunakannya. Bagian ini menjelaskan bagaimana kami membuat informasi ini tersedia untuk wadah Docker Anda. Untuk informasi tentang membuat pekerjaan pelatihan, lihat`CreateTrainingJob`. Untuk informasi selengkapnya tentang cara SageMaker kontainer mengatur informasi, lihatMenggunakan SageMaker Pelatihan dan Inferensi Toolkit .

Topik

- Hyperparameter
- Variabel lingkungan
- Konfigurasi Data Masukan
- Data Pelatihan
- Konfigurasi Pelatihan Terdistribusi

Hyperparameter

SageMaker membuat hyperparameters dalam `CreateTrainingJob` permintaan tersedia di wadah Docker dalam file. `/opt/ml/input/config/hyperparameters.json`

Berikut ini adalah contoh konfigurasi hyperparameter `hyperparameters.json` untuk menentukan `num_round` dan `eta` hyperparameters dalam `CreateTrainingJob` operasi untuk XGBoost.

```
{
    "num_round": "128",
    "eta": "0.001"
}
```

Untuk daftar lengkap hyperparameters yang dapat digunakan untuk algoritma XGBoost SageMaker bawaan, lihat XGBoost Hyperparameters.

Hiperparameter yang dapat Anda atur bergantung pada algoritme yang Anda latih. Untuk daftar hiperparameter yang tersedia untuk algoritme SageMaker bawaan, temukan yang tercantum di Hyperparameters di bawah tautan algoritme di Gunakan Algoritma SageMaker Bawaan Amazon atau Model Pra-terlatih.

Variabel lingkungan

SageMaker menetapkan variabel lingkungan berikut dalam wadah Anda:

- TRAINING_JOB_NAME - Ditentukan dalam parameter permintaan. `TrainingJobName` `CreateTrainingJob`
- TRAINING_JOB_ARN — Nama Sumber Daya Amazon (ARN) dari pekerjaan pelatihan dikembalikan sebagai tanggapan. `TrainingJobArn` `CreateTrainingJob`
- Semua variabel lingkungan ditentukan dalam parameter Lingkungan dalam `CreateTrainingJob` permintaan.

Konfigurasi Data Masukan

SageMaker membuat informasi saluran data dalam `InputDataConfig` parameter dari `CreateTrainingJob` permintaan Anda tersedia di `/opt/ml/input/config/inputdataconfig.json` file di wadah Docker Anda.

Misalnya, Anda menentukan tiga saluran data (`train`,`evaluation`, dan`validation`) dalam permintaan Anda. SageMakermenyediakan JSON berikut:

```
{
  "train" : {"ContentType":  "trainingContentType",
             "TrainingInputMode": "File",
             "S3DistributionType": "FullyReplicated",
             "RecordWrapperType": "None"},
  "evaluation" : {"ContentType":  "evalContentType",
                  "TrainingInputMode": "File",
                  "S3DistributionType": "FullyReplicated",
                  "RecordWrapperType": "None"},
  "validation" : {"TrainingInputMode": "File",
                  "S3DistributionType": "FullyReplicated",
                  "RecordWrapperType": "None"}
}
```

> **Note**
>
> SageMaker hanya menyediakan informasi yang relevan tentang setiap saluran data (misalnya, nama saluran dan jenis konten) ke wadah, seperti yang ditunjukkan pada contoh sebelumnya. `S3DistributionType`akan diatur `FullyReplicated` seolah-olah Anda menentukan EFS atau F SxLustre sebagai sumber data input.

Data Pelatihan

`TrainingInputMode`Parameter dalam `AlgorithmSpecification` `CreateTrainingJob`permintaan menentukan bagaimana kumpulan data pelatihan tersedia untuk penampung Anda. Mode input berikut tersedia.

- **`File`**modus

  Jika Anda menggunakan `File` mode sebagai `TrainingInputMode` nilai Anda, SageMaker tetapkan parameter berikut dalam wadah Anda.

  - `TrainingInputMode`Parameter Anda ditulis `inputdataconfig.json` sebagai “File”.
  - Direktori saluran data Anda ditulis ke`/opt/ml/input/data/`*`channel_name`*.

  Jika Anda menggunakan `File` mode, SageMaker buat direktori untuk setiap saluran. Misalnya, jika Anda memiliki tiga saluran bernama`training`,, dan `validation``testing`, SageMaker membuat tiga direktori berikut dalam wadah Docker Anda:

  - `/opt/ml/input/data/training`
  - `/opt/ml/input/data/validation`
  - `/opt/ml/input/data/testing`

  `File`modus juga mendukung sumber data berikut.

  - Amazon Simple Storage Service (Amazon S3)
  - Amazon Elastic File System (Amazon EFS)
  - Amazon FSx for Lustre

> **Note**
>
> Saluran yang menggunakan sumber data sistem file seperti Amazon EFS dan Amazon FSx harus menggunakan `File` mode. Dalam hal ini, jalur direktori yang disediakan di saluran dipasang di`/opt/ml/input/data/channel_name`.

- **`FastFile`**modus

  Jika Anda menggunakan `FastFile` mode sebagai milik Anda`TrainingInputNodeParameter`, SageMaker atur parameter berikut dalam wadah Anda.

  - Mirip dengan `File` `FastFile` mode, dalam mode, `TrainingInputMode` parameter Anda ditulis `inputdataconfig.json` sebagai “File”.
  - Direktori saluran data Anda ditulis ke`/opt/ml/input/data/channel_name`.

  `FastFile`modus mendukung sumber data berikut.

  - Amazon S3

  Jika Anda menggunakan `FastFile` mode, direktori saluran dipasang dengan izin hanya-baca.

  Secara historis, `File` mode mendahului `FastFile` mode. Untuk memastikan kompatibilitas mundur, algoritme yang mendukung `File` mode juga dapat bekerja dengan mulus dengan `FastFile` mode selama `TrainingInputMode` parameter diatur ke dalam. `File` `inputdataconfig.json`.

  > **Note**
  >
  > Saluran yang menggunakan `FastFile` mode harus menggunakan “S3Prefix”. `S3DataType`
  > `FastFile`mode menyajikan tampilan folder yang menggunakan garis miring maju (`/`) sebagai pembatas untuk mengelompokkan objek Amazon S3 ke dalam folder. `S3Uri`awalan tidak harus sesuai dengan nama folder sebagian. Misalnya, jika kumpulan data Amazon S3 berisi`s3://my-bucket/train-01/data.csv`, maka `s3://my-bucket/train` tidak ada yang `s3://my-bucket/train-01` diizinkan sebagai awalan. `S3Uri`
  > Garis miring ke depan disarankan untuk menentukan saluran yang sesuai dengan folder. Misalnya, `s3://my-bucket/train-01/` saluran untuk `train-01` folder. Tanpa garis

miring ke depan, saluran akan menjadi ambigu jika ada folder atau file `s3://my-bucket/train-011/` lain. `s3://my-bucket/train-01.txt/`

- **`Pipe`**modus
  - `TrainingInputMode`parameter ditulis ke`inputdataconfig.json`: “Pipa”
  - Direktori saluran data dalam wadah Docker: `/opt/ml/input/data/`*`channel_name_epoch_number`*
  - Sumber data yang didukung: Amazon S3

  Anda perlu membaca dari pipa terpisah untuk setiap saluran. Misalnya, jika Anda memiliki tiga saluran bernama`training`,`validation`, dan`testing`, Anda perlu membaca dari pipa berikut:

  - `/opt/ml/input/data/training_0, /opt/ml/input/data/training_1, ...`
  - `/opt/ml/input/data/validation_0, /opt/ml/input/data/validation_1, ...`
  - `/opt/ml/input/data/testing_0, /opt/ml/input/data/testing_1, ...`

  Baca pipa secara berurutan. Misalnya, jika Anda memiliki saluran yang disebut`training`, baca pipa dalam urutan ini:

  1. Buka `/opt/ml/input/data/training_0` dalam mode baca dan baca ke end-of-file (EOF) atau, jika Anda selesai dengan epoch pertama, tutup file pipa lebih awal.
  2. Setelah menutup file pipa pertama, cari `/opt/ml/input/data/training_1` dan baca sampai Anda menyelesaikan epoch kedua, dan seterusnya.

  Jika file untuk epoch tertentu belum ada, kode Anda mungkin perlu mencoba lagi sampai pipa dibuat Tidak ada batasan pengurutan di seluruh jenis saluran. Misalnya, Anda dapat membaca beberapa zaman untuk `training` saluran dan hanya mulai membaca `validation` saluran saat Anda siap. Atau, Anda dapat membacanya secara bersamaan jika algoritme Anda mengharuskannya.

  Untuk contoh notebook Jupyter yang menunjukkan cara menggunakan mode Pipa saat membawa wadah Anda sendiri, lihat Membawa algoritme mode pipa Anda sendiri ke Amazon. SageMaker

Konfigurasi Pelatihan Terdistribusi

Jika Anda melakukan pelatihan terdistribusi dengan beberapa kontainer, SageMaker buat informasi tentang semua kontainer tersedia dalam `/opt/ml/input/config/resourceconfig.json` file.

Untuk mengaktifkan komunikasi antar kontainer, file JSON ini berisi informasi untuk semua kontainer. SageMaker membuat file ini tersedia untuk algoritme `Pipe` mode `File` dan keduanya. File ini memberikan informasi berikut:

- `current_host`—Nama kontainer saat ini di jaringan kontainer. Misalnya, `algo-1`. Nilai host dapat berubah kapan saja. Jangan menulis kode dengan nilai spesifik untuk variabel ini.
- `hosts`—Daftar nama semua kontainer di jaringan kontainer, diurutkan secara leksikografis. Misalnya, `["algo-1", "algo-2", "algo-3"]` untuk cluster tiga simpul. Kontainer dapat menggunakan nama-nama ini untuk menangani kontainer lain di jaringan kontainer. Nilai host dapat berubah kapan saja. Jangan menulis kode dengan nilai spesifik untuk variabel-variabel ini.
- `network_interface_name`—Nama antarmuka jaringan yang terpapar ke wadah Anda. Misalnya, kontainer yang menjalankan Message Passing Interface (MPI) dapat menggunakan informasi ini untuk mengatur nama antarmuka jaringan.
- Jangan gunakan informasi di dalam `/etc/hostname` atau `/etc/hosts` karena mungkin tidak akurat.
- Informasi nama host mungkin tidak segera tersedia untuk wadah algoritme. Sebaiknya tambahkan kebijakan coba lagi pada operasi resolusi nama host saat node tersedia di klaster.

Berikut ini adalah contoh file pada node 1 dalam cluster tiga node:

```
{
    "current_host": "algo-1",
    "hosts": ["algo-1","algo-2","algo-3"],
    "network_interface_name":"eth1"
}
```

Jalankan Pelatihan dengan EFA

SageMaker menyediakan integrasi dengan perangkat EFA untuk mempercepat High Performance Computing (HPC) dan aplikasi pembelajaran mesin. Integrasi ini memungkinkan Anda untuk memanfaatkan perangkat EFA saat menjalankan pekerjaan pelatihan terdistribusi Anda. Anda dapat menambahkan integrasi EFA ke wadah Docker yang ada yang Anda bawa. SageMaker Informasi berikut menguraikan cara mengonfigurasi kontainer Anda sendiri untuk menggunakan perangkat EFA untuk pekerjaan pelatihan terdistribusi Anda.

Prasyarat

Wadah Anda harus memenuhi spesifikasi wadah SageMaker Pelatihan.

Instal EFA dan paket yang diperlukan

Wadah Anda harus mengunduh dan menginstal perangkat lunak EFA. Ini memungkinkan penampung Anda mengenali perangkat EFA, dan menyediakan versi Libfabric dan Open MPI yang kompatibel.

Alat apa pun seperti MPI dan NCCL harus diinstal dan dikelola di dalam wadah untuk digunakan sebagai bagian dari pekerjaan pelatihan yang mendukung EFA Anda. Untuk daftar semua versi EFA yang tersedia, lihat Verifikasi penginstal EFA menggunakan checksum. Contoh berikut menunjukkan cara memodifikasi Dockerfile dari wadah berkemampuan EFA Anda untuk menginstal EFA, MPI, OFI, NCCL, dan NCCL-TEST.

> **Note**
>
> Saat menggunakan PyTorch dengan EFA pada penampung Anda, versi NCCL wadah Anda harus cocok dengan versi NCCL instalasi Anda. PyTorch Untuk memverifikasi versi PyTorch NCCL, gunakan perintah berikut:
>
> ```
> torch.cuda.nccl.version()
> ```

```
ARG OPEN_MPI_PATH=/opt/amazon/openmpi/
ENV NCCL_VERSION=2.7.8
ENV EFA_VERSION=1.30.0
ENV BRANCH_OFI=1.1.1

#################################################
## EFA and MPI SETUP
RUN cd $HOME \
  && curl -O https://s3-us-west-2.amazonaws.com/aws-efa-installer/aws-efa-installer-
${EFA_VERSION}.tar.gz \
  && tar -xf aws-efa-installer-${EFA_VERSION}.tar.gz \
  && cd aws-efa-installer \
  && ./efa_installer.sh -y --skip-kmod -g \

ENV PATH="$OPEN_MPI_PATH/bin:$PATH"
ENV LD_LIBRARY_PATH="$OPEN_MPI_PATH/lib/:$LD_LIBRARY_PATH"

#################################################
## NCCL, OFI, NCCL-TEST SETUP
```

```
RUN cd $HOME \
  && git clone https://github.com/NVIDIA/nccl.git -b v${NCCL_VERSION}-1 \
  && cd nccl \
  && make -j64 src.build BUILDDIR=/usr/local

RUN apt-get update && apt-get install -y autoconf
RUN cd $HOME \
  && git clone https://github.com/aws/aws-ofi-nccl.git -b v${BRANCH_OFI} \
  && cd aws-ofi-nccl \
  && ./autogen.sh \
  && ./configure --with-libfabric=/opt/amazon/efa \
       --with-mpi=/opt/amazon/openmpi \
       --with-cuda=/usr/local/cuda \
       --with-nccl=/usr/local --prefix=/usr/local \
  && make && make install

RUN cd $HOME \
  && git clone https://github.com/NVIDIA/nccl-tests \
  && cd nccl-tests \
  && make MPI=1 MPI_HOME=/opt/amazon/openmpi CUDA_HOME=/usr/local/cuda NCCL_HOME=/usr/
local
```

Pertimbangan saat membuat wadah Anda

Perangkat EFA dipasang ke wadah seperti `/dev/infiniband/uverbs0` di bawah daftar perangkat yang dapat diakses ke wadah. Pada instance P4d, wadah memiliki akses ke 4 perangkat EFA. Perangkat EFA dapat ditemukan dalam daftar perangkat yang dapat diakses ke wadah sebagai:

- `/dev/infiniband/uverbs0`
- `/dev/infiniband/uverbs1`
- `/dev/infiniband/uverbs2`
- `/dev/infiniband/uverbs3`

Untuk mendapatkan informasi tentang nama host, nama host peer, dan antarmuka jaringan (untuk MPI) dari `resourceconfig.json` file yang disediakan untuk setiap instance kontainer, lihat Konfigurasi Pelatihan Terdistribusi. Container Anda menangani lalu lintas TCP reguler antar rekan melalui Elastic Network Interfaces (ENI) default, sambil menangani lalu lintas OFI (kernel bypass) melalui perangkat EFA.

Verifikasi bahwa perangkat EFA Anda dikenali

Untuk memverifikasi bahwa perangkat EFA dikenali, jalankan perintah berikut dari dalam container Anda.

```
/opt/amazon/efa/bin/fi_info -p efa
```

Outputnya semestinya mirip dengan yang berikut.

```
provider: efa
    fabric: EFA-fe80::e5:56ff:fe34:56a8
    domain: efa_0-rdm
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
provider: efa
    fabric: EFA-fe80::e5:56ff:fe34:56a8
    domain: efa_0-dgrm
    version: 2.0
    type: FI_EP_DGRAM
    protocol: FI_PROTO_EFA
provider: efa;ofi_rxd
    fabric: EFA-fe80::e5:56ff:fe34:56a8
    domain: efa_0-dgrm
    version: 1.0
    type: FI_EP_RDM
    protocol: FI_PROTO_RXD
```

Menjalankan pekerjaan pelatihan dengan EFA

Setelah Anda membuat wadah berkemampuan EFA, Anda dapat menjalankan pekerjaan pelatihan dengan EFA menggunakan SageMaker Estimator dengan cara yang sama seperti yang Anda lakukan dengan gambar Docker lainnya. Untuk informasi lebih lanjut tentang mendaftarkan kontainer Anda dan menggunakannya untuk pelatihan, lihat Mengadaptasi Wadah Pelatihan Anda Sendiri.

Bagaimana Amazon SageMaker Sinyal Algoritma Sukses dan Kegagalan

Algoritma pelatihan menunjukkan apakah berhasil atau gagal menggunakan kode keluar dari prosesnya.

Eksekusi pelatihan yang berhasil harus keluar dengan kode keluar 0 dan eksekusi pelatihan yang gagal harus keluar dengan kode keluar bukan nol. Ini akan dikonversi ke `Completed` dan `Failed`

`TrainingJobStatus` dikembalikan oleh`DescribeTrainingJob`. Konvensi kode keluar ini standar dan mudah diimplementasikan dalam semua bahasa. Misalnya, dengan Python, Anda dapat menggunakan `sys.exit(1)` sinyal kegagalan keluar, dan hanya berjalan ke akhir rutinitas utama akan menyebabkan Python keluar dengan kode 0.

Dalam kasus kegagalan, algoritma dapat menulis deskripsi kegagalan pada file kegagalan. Lihat bagian selanjutnya untuk detailnya.

Bagaimana Amazon SageMaker Memproses Output Pelatihan

Saat algoritme Anda berjalan dalam wadah, itu menghasilkan output termasuk status pekerjaan pelatihan dan model dan artefak keluaran. Algoritma Anda harus menulis informasi ini ke file-file berikut, yang terletak di `/output` direktori container. Amazon SageMaker memproses informasi yang terkandung dalam direktori ini sebagai berikut:

- `/opt/ml/model`— Algoritma Anda harus menulis semua artefak model akhir ke direktori ini. SageMaker menyalin data ini sebagai objek tunggal dalam format tar terkompresi ke lokasi S3 yang Anda tentukan dalam permintaan. `CreateTrainingJob` Jika beberapa kontainer dalam satu pekerjaan pelatihan menulis ke direktori ini, mereka harus memastikan tidak ada `file/directory` nama yang berbenturan. SageMakermengumpulkan hasil dalam file TAR dan mengunggah ke S3 di akhir pekerjaan pelatihan.
- `/opt/ml/output/data`— Algoritma Anda harus menulis artefak yang ingin Anda simpan selain model akhir ke direktori ini. SageMakermenyalin data ini sebagai objek tunggal dalam format tar terkompresi ke lokasi S3 yang Anda tentukan dalam permintaan. `CreateTrainingJob` Jika beberapa kontainer dalam satu pekerjaan pelatihan menulis ke direktori ini, mereka harus memastikan tidak ada `file/directory` nama yang berbenturan. SageMaker mengumpulkan hasil dalam file TAR dan mengunggah ke S3 di akhir pekerjaan pelatihan.
- `/opt/ml/output/failure`— Jika pelatihan gagal, setelah semua output algoritma (misalnya, logging) selesai, algoritme Anda harus menulis deskripsi kegagalan ke file ini. Sebagai `DescribeTrainingJob` tanggapan, SageMaker mengembalikan 1024 karakter pertama dari file ini sebagai`FailureReason`.

Gunakan kode inferensi Anda sendiri

Anda dapat menggunakan Amazon SageMaker untuk berinteraksi dengan kontainer Docker dan menjalankan kode inferensi Anda sendiri dengan salah satu dari dua cara:

- Untuk menggunakan kode inferensi Anda sendiri dengan titik akhir persisten untuk mendapatkan satu prediksi pada satu waktu, gunakan layanan SageMaker hosting.
- Untuk menggunakan kode inferensi Anda sendiri untuk mendapatkan prediksi untuk seluruh dataset, gunakan SageMaker batch transform.

Topik

Gunakan Kode Inferensi Anda Sendiri dengan Layanan Hosting

Bagian ini menjelaskan bagaimana Amazon SageMaker berinteraksi dengan wadah Docker yang menjalankan kode inferensi Anda sendiri untuk layanan hosting. Gunakan informasi ini untuk menulis kode inferensi dan membuat gambar Docker.

Topik

Bagaimana SageMaker Menjalankan Gambar Inferensi Anda

Untuk mengonfigurasi wadah agar dijalankan sebagai executable, gunakan `ENTRYPOINT` instruksi dalam Dockerfile. Perhatikan hal berikut:

- Untuk inferensi model, SageMaker jalankan wadah sebagai:

  ```
  docker run image serve
  ```

  SageMaker mengganti `CMD` pernyataan default dalam wadah dengan menentukan `serve` argumen setelah nama gambar. `serve`Argumen mengesampingkan argumen yang Anda berikan dengan `CMD` perintah di Dockerfile.

- SageMaker mengharapkan semua kontainer berjalan dengan pengguna root. Buat wadah Anda sehingga hanya menggunakan pengguna root. Saat SageMaker menjalankan penampung Anda, pengguna yang tidak memiliki akses tingkat root dapat menyebabkan masalah izin.

- Kami menyarankan Anda menggunakan `exec` bentuk `ENTRYPOINT` instruksi:

  ```
  ENTRYPOINT ["executable", "param1", "param2"]
  ```

  Sebagai contoh:

  ```
  ENTRYPOINT ["python", "k_means_inference.py"]
  ```

  `exec`Bentuk `ENTRYPOINT` instruksi memulai eksekusi secara langsung, bukan sebagai anak dari. `/bin/sh` Ini memungkinkannya menerima sinyal seperti `SIGTERM` dan `SIGKILL` dari operasi SageMaker API, yang merupakan persyaratan.

  Misalnya, saat Anda menggunakan `CreateEndpoint`API untuk membuat titik akhir, SageMaker berikan jumlah instans komputasi ML yang diperlukan oleh konfigurasi titik akhir, yang Anda tentukan dalam permintaan. SageMaker menjalankan wadah Docker pada instance tersebut.

  Jika Anda mengurangi jumlah instance yang mendukung titik akhir (dengan memanggil `UpdateEndpointWeightsAndCapacities`API), SageMaker jalankan perintah untuk menghentikan wadah Docker pada instance yang sedang dihentikan. Perintah mengirimkan `SIGTERM` sinyal, kemudian mengirimkan `SIGKILL` sinyal tiga puluh detik kemudian.

  Jika Anda memperbarui titik akhir (dengan memanggil `UpdateEndpoint`API), SageMaker luncurkan kumpulan instance komputasi HTML lainnya dan jalankan container Docker yang berisi kode inferensi Anda di dalamnya. Kemudian menjalankan perintah untuk menghentikan kontainer Docker sebelumnya. Untuk menghentikan kontainer Docker, perintah mengirimkan `SIGTERM` sinyal, lalu mengirimkan `SIGKILL` sinyal 30 detik kemudian.

- SageMaker menggunakan definisi kontainer yang Anda berikan dalam `CreateModel`permintaan Anda untuk mengatur variabel lingkungan dan nama host DNS untuk wadah sebagai berikut:

  - Ini menetapkan variabel lingkungan menggunakan `ContainerDefinition.Environment` string-to-string peta.
  - Ini menetapkan nama host DNS menggunakan file. `ContainerDefinition.ContainerHostname`

- Jika Anda berencana menggunakan perangkat GPU untuk inferensi model (dengan menentukan instance komputasi ML berbasis GPU dalam `CreateEndpointConfig` permintaan Anda), pastikan container Anda kompatibel. `nvidia-docker` Jangan bundel driver NVIDIA dengan gambar. Untuk informasi selengkapnya`nvidia-docker`, lihat NVIDIA/NVIDIA-Docker.

- Anda tidak dapat menggunakan `tini` penginisialisasi sebagai titik masuk Anda dalam SageMaker wadah karena menjadi bingung oleh argumen `train` dan`serve`.

Bagaimana SageMaker Memuat Artefak Model Anda

Dalam permintaan `CreateModel`API Anda, Anda dapat menggunakan `S3DataSource` parameter `ModelDataUrl` atau untuk mengidentifikasi lokasi S3 tempat artefak model disimpan. SageMaker menyalin artefak model Anda dari lokasi S3 ke `/opt/ml/model` direktori untuk digunakan oleh kode inferensi Anda. Kontainer Anda memiliki akses hanya-baca ke. `/opt/ml/model` Jangan menulis ke direktori ini.

`ModelDataUrl`Harus menunjuk ke file tar.gz. Jika SageMaker tidak, tidak akan mengunduh file.

Jika Anda melatih model Anda SageMaker, artefak model disimpan sebagai file tar terkompresi tunggal di Amazon S3. Jika Anda melatih model Anda di luar SageMaker, Anda perlu membuat file tar terkompresi tunggal ini dan menyimpannya di lokasi S3. SageMaker mendekompresi file tar ini ke direktori /opt/ml/model sebelum penampung Anda dimulai.

Untuk menerapkan model besar, kami sarankan Anda mengikutiMenyebarkan model yang tidak terkompresi.

Bagaimana Kontainer Anda Harus Menanggapi Permintaan Inferensi

Untuk mendapatkan kesimpulan, aplikasi klien mengirimkan permintaan POST ke titik SageMaker akhir. SageMaker meneruskan permintaan ke wadah, dan mengembalikan hasil inferensi dari wadah ke klien.

Untuk informasi selengkapnya tentang permintaan inferensi yang akan diterima container Anda, lihat tindakan berikut di Referensi Amazon SageMaker API:

- InvokeEndpoint
- InvokeEndpointAsync
- InvokeEndpointWithResponseStream

Persyaratan untuk wadah inferensi

Untuk menanggapi permintaan inferensi, kontainer Anda harus memenuhi persyaratan berikut:

- SageMaker menghapus semua `POST` header kecuali yang didukung oleh`InvokeEndpoint`. SageMaker mungkin menambahkan header tambahan. Wadah inferensi harus dapat dengan aman mengabaikan header tambahan ini.
- Untuk menerima permintaan inferensi, penampung harus memiliki server web yang mendengarkan pada port 8080 dan harus menerima `POST` permintaan ke `/invocations` dan `/ping` titik akhir.
- Kontainer model pelanggan harus menerima permintaan koneksi soket dalam 250 ms.
- Kontainer model pelanggan harus menanggapi permintaan dalam waktu 60 detik. Model itu sendiri dapat memiliki waktu pemrosesan maksimum 60 detik sebelum merespons. `/invocations` Jika model Anda akan memakan waktu 50-60 detik waktu pemrosesan, batas waktu soket SDK harus disetel menjadi 70 detik.

Example fungsi pemanggilan

Contoh berikut menunjukkan bagaimana kode dalam wadah Anda dapat memproses permintaan inferensi. Contoh-contoh ini menangani permintaan yang dikirim aplikasi klien dengan menggunakan InvokeEndpoint tindakan.

FastAPI

FastTapi adalah kerangka kerja web untuk membangun API dengan Python.

```
from fastapi import FastAPI, status, Request, Response
. . .
app = FastAPI()
. . .
@app.post('/invocations')
async def invocations(request: Request):
    # model() is a hypothetical function that gets the inference output:
    model_resp = await model(Request)

    response = Response(
        content=model_resp,
        status_code=status.HTTP_200_OK,
        media_type="text/plain",
    )
    return response
. . .
```

Dalam contoh ini, `invocations` fungsi menangani permintaan inferensi yang SageMaker mengirim ke titik `/invocations` akhir.

Flask

Flask adalah kerangka kerja untuk mengembangkan aplikasi web dengan Python.

```
import flask
. . .
app = flask.Flask(__name__)
. . .
@app.route('/invocations', methods=["POST"])
def invoke(request):
    # model() is a hypothetical function that gets the inference output:
    resp_body = model(request)
    return flask.Response(resp_body, mimetype='text/plain')
```

Dalam contoh ini, `invoke` fungsi menangani permintaan inferensi yang SageMaker mengirim ke titik `/invocations` akhir.

Example fungsi pemanggilan untuk permintaan streaming

Contoh berikut menunjukkan bagaimana kode dalam wadah inferensi Anda dapat memproses permintaan inferensi streaming. Contoh-contoh ini menangani permintaan yang dikirim aplikasi klien dengan menggunakan InvokeEndpointWithResponseStream tindakan.

Ketika sebuah wadah menangani permintaan inferensi streaming, ia mengembalikan inferensi model sebagai serangkaian bagian secara bertahap saat model menghasilkannya. Aplikasi klien mulai menerima tanggapan segera ketika tersedia. Mereka tidak perlu menunggu model untuk menghasilkan seluruh respons. Anda dapat menerapkan streaming untuk mendukung pengalaman interaktif yang cepat, seperti chatbots, asisten virtual, dan generator musik.

FastAPI

FastTapi adalah kerangka kerja web untuk membangun API dengan Python.

```
from starlette.responses import StreamingResponse
from fastapi import FastAPI, status, Request
. . .
app = FastAPI()
. . .
@app.post('/invocations')
async def invocations(request: Request):
    # Streams inference response using HTTP chunked encoding
    async def generate():
        # model() is a hypothetical function that gets the inference output:
        yield await model(Request)
        yield "\n"

    response = StreamingResponse(
        content=generate(),
        status_code=status.HTTP_200_OK,
        media_type="text/plain",
    )
    return response
. . .
```

Dalam contoh ini, `invocations` fungsi menangani permintaan inferensi yang SageMaker mengirim ke titik `/invocations` akhir. Untuk mengalirkan respons, contoh menggunakan `StreamingResponse` kelas dari kerangka Starlette.

Flask

Flask adalah kerangka kerja untuk mengembangkan aplikasi web dengan Python.

```
import flask
. . .
app = flask.Flask(__name__)
. . .
@app.route('/invocations', methods=["POST"])
def invocations(request):
    # Streams inference response using HTTP chunked encoding

    def generate():
        # model() is a hypothetical function that gets the inference output:
        yield model(request)
        yield "\n"
    return flask.Response(
        flask.stream_with_context(generate()), mimetype='text/plain')
. . .
```

Dalam contoh ini, `invocations` fungsi menangani permintaan inferensi yang SageMaker mengirim ke titik `/invocations` akhir. Untuk mengalirkan respons, contoh menggunakan `flask.stream_with_context` fungsi dari kerangka Flask.

Bagaimana Kontainer Anda Harus Menanggapi Permintaan Pemeriksaan Kesehatan (Ping)

SageMaker meluncurkan wadah inferensi baru dalam situasi berikut:

- Menanggapi`CreateEndpoint`,`UpdateEndpoint`, dan panggilan `UpdateEndpointWeightsAndCapacities` API
- Membuat patch keamanan
- Mengganti instance yang tidak sehat

Segera setelah startup container, SageMaker mulai mengirim permintaan GET berkala ke `/ping` titik akhir.

Persyaratan paling sederhana pada wadah adalah merespons dengan kode status HTTP 200 dan badan kosong. Ini menunjukkan SageMaker bahwa penampung siap menerima permintaan inferensi di titik `/invocations` akhir.

Jika wadah tidak mulai lulus pemeriksaan kesehatan dengan merespons secara konsisten dengan 200s selama 8 menit setelah startup, peluncuran instance baru gagal. Hal ini `CreateEndpoint` menyebabkan kegagalan, meninggalkan titik akhir dalam keadaan gagal. Pembaruan yang diminta oleh `UpdateEndpoint` tidak selesai, patch keamanan tidak diterapkan, dan instance yang tidak sehat tidak diganti.

Sementara bilah minimum adalah wadah mengembalikan 200 statis, pengembang kontainer dapat menggunakan fungsi ini untuk melakukan pemeriksaan lebih dalam. Batas waktu permintaan pada `/ping` upaya adalah 2 detik.

Gunakan Private Docker Registry untuk Wadah Inferensi Waktu Nyata

Amazon SageMaker hosting memungkinkan Anda menggunakan gambar yang disimpan di Amazon ECR untuk membangun kontainer Anda untuk inferensi waktu nyata secara default. Secara opsional, Anda dapat membangun kontainer untuk inferensi real-time dari gambar di registri Docker pribadi. Registri pribadi harus dapat diakses dari VPC Amazon di akun Anda. Model yang Anda buat berdasarkan gambar yang disimpan di registri Docker pribadi Anda harus dikonfigurasi untuk terhubung ke VPC yang sama di mana registri Docker pribadi dapat diakses. Untuk informasi tentang menghubungkan model Anda ke VPC, lihatBerikan Akses Endpoint yang SageMaker Dihosting ke Sumber Daya di VPC Amazon Anda.

Registri Docker Anda harus diamankan dengan sertifikat TLS dari otoritas sertifikat publik (CA) yang dikenal.

> **Note**
>
> Registri Docker pribadi Anda harus mengizinkan lalu lintas masuk dari grup keamanan yang Anda tentukan dalam konfigurasi VPC untuk model Anda, sehingga SageMaker hosting dapat menarik gambar model dari registri Anda.
> SageMaker dapat menarik gambar model dari DockerHub jika ada jalur ke internet terbuka di dalam VPC Anda.

Topik

- Simpan Gambar di Private Docker Registry selain Amazon Elastic Container Registry
- Gunakan Gambar dari Private Docker Registry untuk Inferensi Real-time
- Izinkan SageMaker untuk mengautentikasi ke registri Docker pribadi

- Buat fungsi Lambda
- Berikan izin peran eksekusi Anda ke Lambda
- Buat antarmuka VPC untuk Lambda

Simpan Gambar di Private Docker Registry selain Amazon Elastic Container Registry

Untuk menggunakan registri Docker pribadi untuk menyimpan gambar Anda SageMaker inferensi real-time, buat registri pribadi yang dapat diakses dari VPC Amazon Anda. Untuk informasi tentang membuat registri Docker, lihatMenyebarkan server registridalam dokumentasi Docker. Registri Docker harus mematuhi yang berikut:

- Registri harus berupaDocker Registry HTTP API V2registri.
- Registri Docker harus dapat diakses dari VPC yang sama yang Anda tentukan di`VpcConfig`parameter yang Anda tentukan saat Anda membuat model.

Gunakan Gambar dari Private Docker Registry untuk Inferensi Real-time

Saat Anda membuat model dan menerapkannya SageMaker hosting, Anda dapat menentukan bahwa ia menggunakan gambar dari registri Docker pribadi Anda untuk membangun wadah inferensi. Tentukan ini di`ImageConfig`objek di`PrimaryContainer`parameter yang Anda berikan ke panggilan kecreate_modelfungsi.

Untuk menggunakan gambar yang disimpan di registri Docker pribadi Anda untuk wadah inferensi Anda

1. Buat objek konfigurasi gambar dan tentukan nilai`Vpc`untuk`RepositoryAccessMode`lapangan.

   ```
   image_config = {
                       'RepositoryAccessMode': 'Vpc'
                   }
   ```

2. Jika registri Docker pribadi Anda memerlukan otentikasi, tambahkan`RepositoryAuthConfig`objek ke objek konfigurasi gambar. Untuk`RepositoryCredentialsProviderArn`bidang dari`RepositoryAuthConfig`tentukan Amazon Resource Name (ARN) dari sebuahAWS Lambdafungsi yang menyediakan kredensyal yang memungkinkan SageMaker untuk mengautentikasi ke Docker Registry pribadi Anda. Untuk informasi tentang cara membuat fungsi Lambda untuk menyediakan autentikasi, lihatIzinkan SageMaker untuk mengautentikasi ke registri Docker pribadi.

```
image_config = {
                    'RepositoryAccessMode': 'Vpc',
                    'RepositoryAuthConfig': {
                       'RepositoryCredentialsProviderArn':
 'arn:aws:lambda:Region:Acct:function:FunctionName'
                     }
                }
```

3. Buat objek kontainer utama yang ingin Anda lewati`create_model`, gunakan objek konfigurasi gambar yang Anda buat pada langkah sebelumnya.

   Berikan gambar Anda dimencernabentuk. Jika Anda memberikan gambar Anda menggunakan`:latest`Tag, ada risiko bahwa SageMaker menarik versi gambar yang lebih baru dari yang dimaksudkan. Menggunakan formulir intisari memastikan bahwa SageMaker menarik versi gambar yang dimaksud.

   ```
   primary_container = {
       'ContainerHostname': 'ModelContainer',
       'Image': 'myteam.myorg.com/docker-local/my-inference-image:<IMAGE-TAG>',
       'ImageConfig': image_config
   }
   ```

4. Tentukan Nama Model dan Peran Eksekusi yang ingin Anda lewati`create_model`.

   ```
   model_name = 'vpc-model'
   execution_role_arn = 'arn:aws:iam::123456789012:role/SageMakerExecutionRole'
   ```

5. Tentukan satu atau beberapa grup keamanan dan subnet untuk konfigurasi VPC untuk model Anda. Registri Docker pribadi Anda harus mengizinkan lalu lintas masuk dari grup keamanan yang Anda tentukan. Subnet yang Anda tentukan harus berada di VPC yang sama dengan registri Docker pribadi Anda.

   ```
   vpc_config = {
       'SecurityGroupIds': ['sg-0123456789abcdef0'],
       'Subnets': ['subnet-0123456789abcdef0','subnet-0123456789abcdef1']
   }
   ```

6. Dapatkan Boto3 SageMaker klien.

   ```
   import boto3
   ```

```
sm = boto3.client('sagemaker')
```

7. Buat model dengan meneleponcreate_model, menggunakan nilai yang Anda tentukan dalam langkah sebelumnya untukPrimaryContainerdanVpcConfigparameter.

```
try:
    resp = sm.create_model(
        ModelName=model_name,
        PrimaryContainer=primary_container,
        ExecutionRoleArn=execution_role_arn,
        VpcConfig=vpc_config,
    )
except Exception as e:
    print(f'error calling CreateModel operation: {e}')
else:
    print(resp)
```

8. Akhirnya, menelepon[create_endpoint_config](#)dan[create_endpoint](#)untuk membuat endpoint hosting, gunakan model yang Anda buat pada langkah sebelumnya.

```
endpoint_config_name = 'my-endpoint-config'
sm.create_endpoint_config(
    EndpointConfigName=endpoint_config_name,
    ProductionVariants=[
        {
            'VariantName': 'MyVariant',
            'ModelName': model_name,
            'InitialInstanceCount': 1,
            'InstanceType': 'ml.t2.medium'
        },
    ],
)

endpoint_name = 'my-endpoint'
sm.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name,
)

sm.describe_endpoint(EndpointName=endpoint_name)
```

Izinkan SageMaker untuk mengautentikasi ke registri Docker pribadi

Untuk menarik gambar inferensi dari registri Docker pribadi yang memerlukan otentikasi, buatAWS Lambdafungsi Lambda saat Anda memanggil Nama Sumber Daya Amazon (ARN) fungsi Lambda saat Anda memanggilcreate_model. Kapan SageMaker laricreate_model, ia memanggil fungsi Lambda yang Anda tentukan untuk mendapatkan kredensi untuk diautentikasi ke registri Docker Anda.

Buat fungsi Lambda

Buat sebuahAWS Lambdafungsi yang mengembalikan respon dengan bentuk berikut:

```
def handler(event, context):
   response = {
      "Credentials": {"Username": "username", "Password": "password"}
   }
   return response
```

Bergantung pada cara Anda mengatur otentikasi untuk registri Docker pribadi Anda, kredenal yang dikembalikan oleh fungsi Lambda Anda dapat berarti salah satu dari yang berikut:

- Jika Anda mengatur registri Docker pribadi Anda untuk menggunakan otentikasi dasar, berikan kredensi masuk untuk mengautentikasi ke registri.
- Jika Anda mengatur registri Docker pribadi Anda untuk menggunakan otentikasi token pembawa, kredenal masuk dikirim ke server otorisasi Anda, yang mengembalikan token Bearer yang kemudian dapat digunakan untuk mengautentikasi ke registri Docker pribadi.

Berikan izin peran eksekusi Anda ke Lambda

Peran eksekusi yang Anda gunakan untuk memanggilcreate_modelharus memiliki izin untuk meneleponAWS Lambdafungsi. Tambahkan yang berikut ini ke kebijakan izin dari peran eksekusi Anda.

```
{
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": [
        "arn:aws:lambda:*:*:function:*myLambdaFunction*"
```

```
    ]
}
```

DimanamyLambdaFunctionadalah nama fungsi Lambda Anda. Untuk informasi tentang mengedit kebijakan izin peran, lihatMengubah kebijakan izin peran (konsol)diAWS Identity and Access ManagementPanduan Pengguna.

> **Note**
>
> Peran eksekusi dengan`AmazonSageMakerFullAccess`kebijakan terkelola yang dilampirkan padanya memiliki izin untuk memanggil fungsi Lambda apa pun dengan`SageMaker`dalam namanya.

Buat antarmuka VPC untuk Lambda

Buat titik akhir antarmuka sehingga VPC Amazon Anda dapat berkomunikasi dengan AndaAWS Lambdaberfungsi tanpa mengirim lalu lintas melalui internet. Untuk informasi tentang cara melakukannya, lihatMengonfigurasi VPC endpoint antarmuka untuk LambdadiAWS LambdaPanduan Pengembang.

SageMaker hosting mengirimkan permintaan melalui VPC Anda `kelambda.`*`region`*`.amazonaws.com`, untuk memanggil fungsi Lambda Anda. Jika Anda memilih Nama DNS Pribadi saat membuat titik akhir antarmuka, Amazon Route 53 merutekan panggilan ke titik akhir antarmuka Lambda. Jika Anda menggunakan penyedia DNS yang berbeda, pastikan untuk memetakan`lambda.`*`region`*`.amazonaws.com`ke titik akhir Lambda Anda.

Gunakan Kode Inferensi Anda Sendiri dengan Batch Transform

Bagian ini menjelaskan cara Amazon SageMaker berinteraksi dengan container Docker yang menjalankan kode inferensi Anda sendiri untuk transformasi batch. Gunakan informasi ini untuk menulis kode inferensi dan membuat image Docker.

Topik

- Bagaimana SageMaker Menjalankan Gambar Inferensi Anda
- Bagaimana SageMaker Memuat Artefak Model Anda
- Bagaimana Kontainer Melayani Permintaan
- Bagaimana Container Anda Harus Menanggapi Permintaan Inferensi

- Bagaimana Kontainer Anda Harus Menanggapi Permintaan Pemeriksaan Kesehatan (Ping)

Bagaimana SageMaker Menjalankan Gambar Inferensi Anda

Untuk mengonfigurasi kontainer agar dapat dijalankan sebagai executable, gunakan `ENTRYPOINT` instruksi di Dockerfile. Perhatikan hal berikut:

- Untuk transformasi batch, SageMaker memanggil model atas nama Anda. SageMakermenjalankan wadah sebagai:

  ```
  docker run image serve
  ```

  Input untuk mengubah batch harus dari format yang dapat dibagi menjadi file yang lebih kecil untuk diproses secara paralel. Format ini termasuk CSV, JSON, JSON Lines, TFRecord dan RecordO.

  SageMakermenimpa `CMD` pernyataan default dalam wadah dengan menentukan `serve` argumen setelah nama gambar. `serve`Argumen menimpa argumen yang Anda berikan dengan `CMD` perintah di Dockerfile.

- Sebaiknya gunakan `exec` bentuk `ENTRYPOINT` instruksi:

  ```
  ENTRYPOINT ["executable", "param1", "param2"]
  ```

  Misalnya:

  ```
  ENTRYPOINT ["python", "k_means_inference.py"]
  ```

- SageMakermenetapkan variabel lingkungan yang ditentukan dalam `CreateModel`dan `CreateTransformJob`pada wadah Anda. Selain itu, variabel lingkungan berikut diisi:
  - `SAGEMAKER_BATCH`diatur ke `true` saat kontainer menjalankan transformasi batch.
  - `SAGEMAKER_MAX_PAYLOAD_IN_MB`diatur ke payload ukuran terbesar yang dikirim ke wadah melalui HTTP.
  - `SAGEMAKER_BATCH_STRATEGY`diatur ke `SINGLE_RECORD` saat kontainer dikirim satu catatan per panggilan ke pemanggilan dan `MULTI_RECORD` ketika wadah mendapat catatan sebanyak yang akan muat dalam payload.

- `SAGEMAKER_MAX_CONCURRENT_TRANSFORMS`diatur ke jumlah maksimum `/invocations` permintaan yang dapat dibuka secara bersamaan.

> **Note**
>
> Tiga variabel lingkungan terakhir berasal dari panggilan API yang dibuat oleh pengguna. Jika pengguna tidak menetapkan nilai untuk mereka, mereka tidak dilewatkan. Dalam hal ini, baik nilai default atau nilai-nilai yang diminta oleh algoritma (dalam menanggapi`/execution-parameters`) digunakan.

- Jika Anda berencana menggunakan perangkat GPU untuk inferensi model (dengan menentukan instans komputasi berbasis GPU dalam `CreateTransformJob` permintaan Anda), pastikan kontainer Anda kompatibel dengan nvidia-docker. Jangan bundel driver NVIDIA dengan gambar. Untuk informasi lebih lanjut tentang nvidia-docker, lihat NVIDIA/NVIDIA-Docker.

- Anda tidak dapat menggunakan `init` initializer sebagai titik masuk Anda dalam SageMaker kontainer karena menjadi bingung oleh kereta api dan melayani argumen.

Bagaimana SageMaker Memuat Artefak Model Anda

Dalam `CreateModel`permintaan, definisi kontainer menyertakan `ModelDataUrl` parameter, yang mengidentifikasi lokasi di Amazon S3 tempat artefak model disimpan. Ketika Anda menggunakan SageMaker untuk menjalankan kesimpulan, menggunakan informasi ini untuk menentukan dari mana untuk menyalin artefak model. Ini menyalin artefak ke `/opt/ml/model` direktori dalam wadah Docker untuk digunakan oleh kode inferensi Anda.

`ModelDataUrl`Parameter harus menunjuk ke file tar.gz. Jika tidak, tidak SageMaker dapat mengunduh file. Jika Anda melatih modelSageMaker, itu menyimpan artefak sebagai file tar terkompresi tunggal di Amazon S3. Jika Anda melatih model dalam kerangka kerja lain, Anda perlu menyimpan artefak model di Amazon S3 sebagai file tar terkompresi. SageMakerdekompresi file tar ini dan menyimpannya dalam `/opt/ml/model` direktori dalam wadah sebelum pekerjaan batch transform dimulai.

Bagaimana Kontainer Melayani Permintaan

Kontainer harus mengimplementasikan server web yang merespons pemanggilan dan permintaan ping pada port 8080. Untuk transformasi batch, Anda memiliki opsi untuk mengatur algoritme untuk

mengimplementasikan permintaan parameter eksekusi untuk menyediakan konfigurasi runtime dinamis. SageMaker SageMakermenggunakan endpoint berikut:

- `ping`—Digunakan untuk memeriksa kesehatan wadah secara berkala. SageMakermenunggu kode `200` status HTTP dan badan kosong untuk permintaan ping berhasil sebelum mengirim permintaan pemanggilan. Anda dapat menggunakan permintaan ping untuk memuat model ke memori untuk menghasilkan inferensi saat permintaan pemanggilan dikirim.
- (Opsional) `execution-parameters` —Memungkinkan algoritma untuk menyediakan parameter tuning optimal untuk pekerjaan selama runtime. Berdasarkan memori dan CPU yang tersedia untuk wadah, algoritma memilih yang sesuai`MaxConcurrentTransforms`,`BatchStrategy`, dan `MaxPayloadInMB` nilai-nilai untuk pekerjaan.

Sebelum memanggil permintaan pemanggilan, SageMaker mencoba untuk memanggil permintaan eksekusi-parameter. Ketika Anda membuat batch mengubah pekerjaan, Anda dapat memberikan nilai-nilai untuk`MaxConcurrentTransforms`,`BatchStrategy`, dan `MaxPayloadInMB` parameter. SageMakermenentukan nilai-nilai untuk parameter ini menggunakan urutan diutamakan:

1. Nilai parameter yang Anda berikan saat membuat `CreateTransformJob` permintaan.
2. Nilai-nilai yang kontainer model kembali ketika SageMaker memanggil eksekusi-parameter endpoint>
3. Nilai parameter default, tercantum dalam tabel berikut.

Parameter	Nilai Default
`MaxConcurrentTransforms`	1
`BatchStrategy`	`MULTI_RECORD`
`MaxPayloadInMB`	6

Respon untuk permintaan `GET` eksekusi-parameter adalah objek JSON dengan kunci untuk`MaxConcurrentTransforms`,`BatchStrategy`, dan parameter. `MaxPayloadInMB` Ini adalah contoh respons yang valid:

```
{
“MaxConcurrentTransforms”: 8,
“BatchStrategy": "MULTI_RECORD",
```

```
"MaxPayloadInMB": 6
}
```

Bagaimana Container Anda Harus Menanggapi Permintaan Inferensi

Untuk mendapatkan kesimpulan, Amazon SageMaker mengirimkan permintaan POST ke wadah inferensi. Badan permintaan POST berisi data dari Amazon S3. Amazon SageMaker meneruskan permintaan ke kontainer, dan mengembalikan hasil inferensi dari kontainer, menyimpan data dari respons ke Amazon S3.

Untuk menerima permintaan inferensi, wadah harus memiliki server web mendengarkan pada port 8080 dan harus menerima permintaan POST ke titik akhir`/invocations`. Permintaan inferensi timeout dan max percobaan ulang dapat dikonfigurasi melalui. `ModelClientConfig`

Bagaimana Kontainer Anda Harus Menanggapi Permintaan Pemeriksaan Kesehatan (Ping)

Persyaratan paling sederhana pada wadah adalah merespons dengan kode status HTTP 200 dan badan kosong. Hal ini menunjukkan SageMaker bahwa wadah siap menerima permintaan inferensi di titik `/invocations` akhir.

Sementara bar minimum adalah untuk wadah untuk mengembalikan statis 200, pengembang kontainer dapat menggunakan fungsi ini untuk melakukan pemeriksaan lebih dalam. Batas waktu permintaan pada `/ping` upaya adalah 2 detik.

Contoh dan Informasi Lebih Lanjut: Gunakan Algoritma atau Model Anda Sendiri

Notebook Jupyter berikut dan informasi tambahan menunjukkan cara menggunakan algoritme Anda sendiri atau model yang telah dilatih sebelumnya dari instance notebook Amazon. SageMaker Untuk tautan ke GitHub repositori dengan Dockerfiles bawaan untuk, TensorFlow MxNet, Chainer, dan PyTorch kerangka kerja serta instruksi tentang penggunaan AWS SDK for Python (Boto3) estimator untuk menjalankan algoritme pelatihan Anda sendiri di Learner dan model Anda sendiri di hosting, lihat SageMaker SageMaker PrebuiltSageMakerGambar Docker untuk Deep Learning

Pengaturan

1. Buat instance SageMaker notebook. Untuk petunjuk tentang cara membuat dan mengakses instance notebook Jupyter, lihat. Instans SageMaker Notebook Amazon

2. Buka instance notebook yang Anda buat.
3. Pilih tab SageMaker Contoh untuk daftar semua SageMaker contoh buku catatan.
4. Buka contoh buku catatan dari bagian Fungsionalitas Tingkat Lanjut di instance buku catatan Anda atau dari GitHub menggunakan tautan yang disediakan. Untuk membuka buku catatan, pilih tab Use, lalu pilih Create copy.

Model tuan rumah dilatih di Scikit-learn

Untuk mempelajari cara meng-host model yang dilatih dalam Scikit-learn untuk membuat prediksi SageMaker dengan menyuntikkannya ke dalam wadah k-means dan XGBoost pihak pertama, lihat contoh buku catatan berikut.

- kmeans_bring_your_own_model
- xgboost_bring_your_own_model

Package TensorFlow dan model Scikit-learn untuk digunakan di SageMaker

Untuk mempelajari cara mengemas algoritme yang telah Anda kembangkan TensorFlow dan kerangka kerja scikit-learn untuk pelatihan dan penerapan di SageMaker lingkungan, lihat buku catatan berikut. Mereka menunjukkan kepada Anda cara membangun, mendaftar, dan menerapkan wadah Docker Anda sendiri menggunakan Dockerfiles.

- tensorflow_bring_your_own
- scikit_bring_your_own

Melatih dan menyebarkan jaringan saraf SageMaker

Untuk mempelajari cara melatih jaringan saraf secara lokal menggunakan MXNet TensorFlow atau, lalu membuat titik akhir dari model terlatih dan menerapkannya SageMaker, lihat buku catatan berikut. Model MXNet dilatih untuk mengenali angka tulisan tangan dari dataset MNIST. TensorFlow Model ini dilatih untuk mengklasifikasikan iris.

- mxnet_mnist_byom
- Tensorflow_byom_iris

Pelatihan menggunakan mode pipa

Untuk mempelajari cara menggunakan Dockerfile untuk membangun kontainer yang memanggil `train.py script` dan menggunakan mode pipa untuk melatih algoritme secara kustom, lihat buku catatan berikut. Dalam mode pipa, data input ditransfer ke algoritme saat sedang dilatih. Ini dapat mengurangi waktu pelatihan dibandingkan dengan menggunakan mode file.

- pipe_bring_your_own

Bawa model R Anda sendiri

Untuk mempelajari cara menggunakan menambahkan gambar R khusus untuk membuat dan melatih model di buku AWS SMS catatan, lihat posting blog berikut. Posting blog ini menggunakan sampel R Dockerfile dari perpustakaan SageMakerStudio Classic Custom Image Sampel.

- Membawa lingkungan R Anda sendiri ke Amazon SageMaker Studio Classic

Memperpanjang Image PyTorch kontainer yang sudah dibuat sebelumnya

Untuk mempelajari cara memperluas image SageMaker PyTorch kontainer bawaan saat Anda memiliki persyaratan fungsional tambahan untuk algoritme atau model yang tidak didukung oleh image Docker bawaan, lihat buku catatan berikut.

- BertTopic_Extending_Container

Untuk informasi selengkapnya tentang memperluas kontainer, lihat Memperpanjang Kontainer Pra-Built.

Melatih dan men-debug pekerjaan pelatihan pada wadah khusus

Untuk mempelajari cara melatih dan men-debug pekerjaan pelatihan menggunakan SageMaker Debugger, lihat buku catatan berikut. Skrip pelatihan yang diberikan melalui contoh ini menggunakan model TensorFlow Keras ResNet 50 dan dataset CIFAR10. Wadah kustom Docker dibuat dengan skrip pelatihan dan didorong ke Amazon ECR. Saat pekerjaan pelatihan sedang berjalan, Debugger mengumpulkan output tensor dan mengidentifikasi masalah debugging. Dengan alat pustaka `smdebug` klien, Anda dapat menyetel objek `smdebug` uji coba yang memanggil pekerjaan pelatihan

dan informasi debugging, memeriksa status aturan pelatihan dan Debugger, dan mengambil tensor yang disimpan dalam bucket Amazon S3 untuk menganalisis masalah pelatihan.

- build_your_own_container_with_debugger

Mengatasi masalah kontainer Docker

Berikut ini adalah kesalahan umum yang mungkin Anda alami saat menggunakan kontainer Docker SageMaker. Setiap kesalahan diikuti oleh solusi untuk kesalahan.

- Kesalahan: SageMaker telah kehilangan daemon Docker.

  Untuk memperbaiki eror ini, restart Docker menggunakan perintah berikut.

  ```
  sudo service docker restart
  ```

- Kesalahan: Klaster **`/tmp`** direktori container Docker Anda telah kehabisan ruang.

  kontainer Docker menggunakan `/dan/tmp` partisi untuk menyimpan kode. Partisi ini dapat diisi dengan mudah saat menggunakan modul kode besar dalam mode lokal. Klaster SageMakerPython SDK mendukung penentuan direktori temp kustom untuk direktori root mode lokal Anda untuk menghindari masalah ini.

  Untuk menentukan direktori temp kustom di penyimpanan volume EBS, buat file di jalur berikut `~/.sagemaker/config.yaml` dan tambahkan konfigurasi berikut. Direktori yang Anda tentukan sebagai `container_root` harus sudah ada. Klaster SageMakerPython SDK tidak akan mencoba membuatnya.

  ```
  local:
    container_root: /home/ec2-user/SageMaker/temp
  ```

  Dengan konfigurasi ini, mode lokal menggunakan `/temp` direktori dan bukan default `/tmp` direktori.

Konfigurasikan keamanan di Amazon SageMaker

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS. AWS juga memberikan Anda layanan yang dapat digunakan dengan aman. Auditor pihak ketiga secara berkala menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari AWS program kepatuhan. Untuk mempelajari tentang program kepatuhan yang berlaku di Amazon SageMaker, lihat AWSLayanan dalam Cakupan melalui Program Kepatuhan.
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini akan membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan SageMaker. Topik berikut menunjukkan cara mengonfigurasi SageMaker untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan SageMaker sumber daya Anda.

Topik

Privasi Data di Amazon SageMaker

Amazon SageMaker mengumpulkan informasi agregat tentang penggunaan pustaka AWS milik dan open source yang digunakan selama pelatihan. SageMaker menggunakan metadata agregat ini untuk meningkatkan layanan dan pengalaman pelanggan.

Bagian berikut memberikan penjelasan untuk jenis metadata yang SageMaker dikumpulkan dan cara memilih keluar dari koleksi metadata.

Jenis informasi yang dikumpulkan

Informasi Penggunaan

Metadata dari perpustakaan AWS milik dan open source yang digunakan dengan SageMaker pelatihan, seperti yang digunakan untuk pelatihan terdistribusi, kompilasi, dan kuantisasi.

Kesalahan

Kesalahan dari perilaku tak terduga termasuk kegagalan, crash, kaskade, dan kegagalan yang dihasilkan dari interaksi dengan platform pelatihan. SageMaker

Cara memilih keluar dari koleksi metadata

Anda dapat memilih untuk tidak membagikan metadata gabungan dengan SageMaker pelatihan saat membuat pekerjaan pelatihan menggunakan API. `CreateTrainingJob` Jika Anda menggunakan konsol untuk membuat pekerjaan pelatihan, pengumpulan metadata dinonaktifkan secara default.

> **Important**
>
> Anda harus memilih untuk memilih keluar dari pengumpulan metadata untuk setiap pekerjaan pelatihan yang Anda kirimkan. Anda juga harus memilih untuk memilih keluar dalam panggilan API seperti yang ditunjukkan dalam contoh berikut. Anda tidak dapat memilih untuk memilih keluar dalam skrip pelatihan.

Bagian berikut menunjukkan bagaimana Anda dapat memilih keluar dari koleksi metadata menggunakanAWS CLI,AWS SDK for Python (Boto3), atau Python SageMaker SDK.

Menyisih dari koleksi metadata menggunakan () AWS Command Line InterfaceAWS CLI

Untuk memilih keluar dari koleksi metadata menggunakanAWS CLI, setel variabel lingkungan `OPT_OUT_TRACKING` ke 1 dalam `create-training-job` API seperti yang ditunjukkan dalam contoh kode berikut.

```
aws sagemaker create-training-job \
--training-job-name your_job_name \
--algorithm-specification AlgorithmName=your_algorithm_name\
--output-data-config S3OutputPath=s3://bucket-name/key-name-prefix \
--resource-config InstanceType=ml.c5.xlarge, InstanceCount=1 \
--stopping-condition MaxRuntimeInSeconds=100 \
--environment OPT_OUT_TRACKING=1
```

Menyisih dari koleksi metadata menggunakan AWS SDK for Python (Boto3)

Untuk memilih keluar dari koleksi metadata menggunakan SDK for Python (Boto3), setel 1 variabel `OPT_OUT_TRACKING` lingkungan ke dalam API seperti yang ditunjukkan pada contoh kode berikut. `create_training_job`

```
boto3.client('sagemaker').create_training_job(
    TrainingJobName='your_training_job',
    AlgorithmSpecification={
        'AlgorithmName': 'your_algorithm_name',
        'TrainingInputMode': 'File',
    },
    RoleArn='your_arn',
    OutputDataConfig={
        'S3OutputPath': 's3://bucket-name/key-name-prefix',
    },
    ResourceConfig={
        'InstanceType': 'ml.m4.xlarge',
        'InstanceCount': 1,
        'VolumeSizeInGB': 123,
    },
    StoppingCondition={
        'MaxRuntimeInSeconds': 123,
    },
    Environment={
        'OPT_OUT_TRACKING': '1'
```

```
    },
)
```

Menyisih dari koleksi metadata menggunakan Python SageMaker SDK

Untuk memilih keluar dari koleksi metadata menggunakan SageMaker Python SDK, atur variabel lingkungan `OPT_OUT_TRACKING` ke 1 dalam SageMaker estimator seperti yang ditunjukkan pada contoh kode berikut.

```
sagemaker.estimator(
    image_uri='path_to_container',
    role='rolearn',
    instance_count=1,
    instance_type='ml.c5.xlarge',
    environment={
        'OPT_OUT_TRACKING': '1'
    },
)
```

Menyisih dari seluruh akun pengumpulan metadata

Jika Anda ingin memilih keluar dari pengumpulan metadata untuk beberapa akun, Anda dapat mengatur variabel lingkungan untuk memilih keluar dari melacak seluruh akun. Anda harus menggunakan SageMaker Python SDK untuk memilih keluar dari pengumpulan metadata di tingkat akun.

Contoh kode berikut menunjukkan cara memilih keluar dari melacak seluruh akun.

```
SchemaVersion: '1.0'
SageMaker:
  TrainingJob:
    Environment:
      'OPT_OUT_TRACKING': '1'
```

Untuk informasi selengkapnya tentang cara memilih keluar dari melacak seluruh akun, lihat Mengonfigurasi dan menggunakan default dengan Python SDK. SageMaker

Informasi tambahan

Jika layanan hilir Anda tergantung pada pelatihan SageMaker

Jika Anda mengoperasikan layanan yang mengandalkan SageMaker pelatihan, sangat disarankan agar Anda memberi tahu pelanggan Anda tentang pengumpulan metadata agregat di platform SageMaker Pelatihan dan memberi mereka pilihan untuk memilih keluar. Atau, Anda dapat memilih keluar dari pengumpulan metadata atas nama pelanggan Anda.

Jika Anda adalah klien atau pelanggan layanan yang menggunakan SageMaker pelatihan

Jika Anda adalah klien atau pelanggan layanan yang menggunakan SageMaker pelatihan, gunakan metode pilihan Anda di bagian sebelumnya untuk memilih keluar dari pengumpulan metadata.

Perlindungan Data di Amazon SageMaker

Model tanggung jawab AWS bersama diterapkan untuk perlindungan data di Amazon SageMaker. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS.

Untuk tujuan perlindungan data, sebaiknya lindungi kredensial Akun AWS dan siapkan untuk masing-masing pengguna AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya AWS. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pengelogan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama semua kontrol keamanan bawaan dalam Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon SageMaker atau lainnya Layanan AWS menggunakan konsol, APIAWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Topik

Lindungi Data Saat Tidak Digunakan Menggunakan Enkripsi

Untuk melindungi notebook Amazon SageMaker Studio dan instans SageMaker notebook Anda, bersama dengan data pembuatan model dan artefak model Anda, SageMaker enkripsi notebook, serta output dari pekerjaan Pelatihan dan Transformasi Batch. SageMaker mengenkripsi ini secara default menggunakan Kunci AWS Terkelola untuk Amazon S3. Kunci AWS Terkelola untuk Amazon S3 ini tidak dapat dibagikan untuk akses lintas akun. Untuk akses lintas akun, tentukan kunci yang dikelola pelanggan Anda sambil membuat SageMaker sumber daya sehingga dapat dibagikan untuk akses lintas akun. Untuk output data ke Amazon S3 Express One Zone, data dienkripsi dengan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3). Untuk informasi selengkapnyaAWS KMS, lihat Apa itu Layanan Manajemen AWS Kunci? .

Topik

Notebook studio

Di Amazon SageMaker Studio, notebook dan data SageMaker Studio Anda dapat disimpan di lokasi berikut:

- Bucket S3 — Saat Anda melakukan onboard ke Studio dan mengaktifkan sumber daya notebook yang dapat dibagikan, SageMaker bagikan snapshot dan metadata di bucket Amazon Simple Storage Service (Amazon S3).
- Volume EFS — Saat Anda onboard ke Studio, SageMaker lampirkan volume Amazon Elastic File System (Amazon EFS) ke domain Anda untuk menyimpan notebook Studio dan file data Anda. Volume EFS tetap ada setelah domain dihapus.
- Volume EBS — Saat Anda membuka buku catatan di Studio, Amazon Elastic Block Store (Amazon EBS) dilampirkan ke instans tempat notebook berjalan. Volume EBS tetap ada selama durasi instance.

SageMaker menggunakan AWS Key Management Service (AWS KMS) untuk mengenkripsi bucket S3 dan kedua volume. Secara default, ia menggunakan kunci KMS yang dikelola di akun AWS layanan. Untuk kontrol lebih lanjut, Anda dapat menentukan kunci terkelola pelanggan Anda sendiri saat Anda onboard ke Studio atau melalui SageMaker API. Lihat informasi yang lebih lengkap di Ikhtisar SageMaker Domain Amazon dan CreateDomain.

Di `CreateDomain` API, Anda menggunakan `S3KmsKeyId` parameter untuk menentukan kunci terkelola pelanggan untuk buku catatan yang dapat dibagikan. Anda menggunakan `KmsKeyId` parameter untuk menentukan kunci terkelola pelanggan untuk volume EFS dan EBS. Kunci yang dikelola pelanggan yang sama digunakan untuk kedua volume. Kunci yang dikelola pelanggan untuk notebook yang dapat dibagikan dapat menjadi kunci yang dikelola pelanggan yang sama seperti yang digunakan untuk volume atau kunci yang dikelola pelanggan yang berbeda.

Instance, SageMaker lowongan, dan Endpoint Notebook

Untuk mengenkripsi volume penyimpanan machine learning (ML) yang dilampirkan ke notebook, pekerjaan pemrosesan, pekerjaan pelatihan, pekerjaan tuning hyperparameter, pekerjaan transformasi batch, dan titik akhir, Anda dapat meneruskan kunci ke. AWS KMS SageMaker Jika Anda tidak menentukan kunci KMS, SageMaker mengenkripsi volume penyimpanan dengan kunci sementara dan membuangnya segera setelah mengenkripsi volume penyimpanan. Untuk instance notebook, jika Anda tidak menentukan kunci KMS, SageMaker mengenkripsi volume OS dan volume data ML dengan kunci KMS yang dikelola sistem.

Anda dapat menggunakan AWS KMS kunci AWS terkelola untuk mengenkripsi semua volume OS instance. Anda dapat mengenkripsi semua volume data ML untuk semua SageMaker instance dengan AWS KMS kunci yang Anda tentukan. Volume penyimpanan ML dipasang sebagai berikut:

- Notebook - `/home/ec2-user/SageMaker`

- Pemrosesan - `/opt/ml/processing` dan `/tmp/`
- Pelatihan - `/opt/ml/` dan `/tmp/`
- Batch - `/opt/ml/` dan `/tmp/`
- Titik akhir - `/opt/ml/` dan `/tmp/`

Pemrosesan, transformasi batch, dan pelatihan wadah pekerjaan dan penyimpanannya bersifat fana. Ketika pekerjaan selesai, output diunggah ke Amazon S3 AWS KMS menggunakan enkripsi dengan kunci AWS KMS opsional yang Anda tentukan dan instans dirobohkan. Jika AWS KMS Kunci tidak disediakan dalam permintaan pekerjaan, SageMaker gunakan AWS KMS kunci default untuk Amazon S3 untuk akun peran Anda. Jika data keluaran disimpan di Amazon S3 Express One Zone, data tersebut dienkripsi dengan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3).

> **Note**
>
> Kebijakan kunci untuk Kunci AWS Terkelola untuk Amazon S3 tidak dapat diedit, sehingga izin lintas akun tidak dapat diberikan untuk kebijakan utama ini. Jika bucket Amazon S3 keluaran untuk permintaan tersebut berasal dari akun lain, tentukan Kunci AWS KMS Pelanggan Anda sendiri dalam permintaan pekerjaan dan pastikan bahwa peran eksekusi pekerjaan memiliki izin untuk mengenkripsi data dengannya.

> **Important**
>
> Data sensitif yang perlu dienkripsi dengan kunci KMS untuk alasan kepatuhan harus disimpan dalam volume penyimpanan ML atau di Amazon S3, yang keduanya dapat dienkripsi menggunakan kunci KMS yang Anda tentukan.

Saat Anda membuka instance notebook, SageMaker menyimpannya dan file apa pun yang terkait dengannya di SageMaker folder dalam volume penyimpanan ML secara default. Saat Anda menghentikan instance notebook, SageMaker buat snapshot dari volume penyimpanan ML. Setiap penyesuaian ke sistem operasi instance yang dihentikan, seperti pustaka kustom yang diinstal atau pengaturan tingkat sistem operasi, hilang. Pertimbangkan untuk menggunakan konfigurasi siklus hidup untuk mengotomatiskan penyesuaian instance notebook default. Saat Anda menghentikan instance, snapshot dan volume penyimpanan ML akan dihapus. Data apa pun yang Anda perlukan untuk bertahan di luar masa pakai instans notebook harus ditransfer ke bucket Amazon S3.

> **Note**
>
> SageMaker Instans berbasis Nitro tertentu menyertakan penyimpanan lokal, tergantung dari tipe instans. Volume penyimpanan lokal dienkripsi menggunakan modul perangkat keras pada instans. Anda tidak dapat menggunakan kunci KMS pada tipe instans dengan penyimpanan lokal. Untuk daftar tipe instans yang mendukung penyimpanan instans lokal, lihat Volume Penyimpanan Instans. Untuk informasi selengkapnya tentang volume penyimpanan pada instans berbasis Nitro, lihat Amazon EBS dan NVMe di Instans Linux. Untuk informasi lebih lanjut tentang enkripsi penyimpanan instans lokal, lihat Volume Penyimpanan Instans SSD.

SageMaker kemampuan geospasial

Anda dapat melindungi data at rest Anda menggunakan enkripsi untuk SageMaker geospasial.

Enkripsi Sisi Server dengan kunci milik SageMaker geospasial Amazon (Default)

Kemampuan SageMaker geospasial Amazon mengenkripsi semua data Anda, termasuk hasil komputasi dari Anda `EarthObservationJobs` dan `VectorEnrichmentJobs` bersama dengan semua metadata layanan Anda. Tidak ada data yang disimpan dalam Amazon yang SageMaker tidak terenkripsi. Ini menggunakan default Kunci milik AWS untuk mengenkripsi semua data Anda.

Enkripsi Sisi Server dengan Kunci KMS yang Disimpan di (SSE-KMS) AWS Key Management Service

Kemampuan SageMaker geospasial Amazon mendukung enkripsi menggunakan kunci KMS milik pelanggan. Untuk informasi selengkapnya, lihat Menggunakan AWS KMS Izin untuk kemampuan SageMaker geospasial Amazon.

Melindungi Data dalam Transit dengan Enkripsi

Semua data internetwork dalam perjalanan mendukung enkripsi TLS 1.2. Kami merekomendasikan agar Anda menggunakan TLS 1.3.

Dengan Amazon SageMaker, artefak model machine learning (ML) dan artefak sistem lainnya dienkripsi saat transit dan saat istirahat. Permintaan ke SageMaker API dan konsol dibuat melalui koneksi aman (SSL). Anda meneruskan AWS Identity and Access Management peran SageMaker untuk memberikan izin untuk mengakses sumber daya atas nama Anda untuk pelatihan dan penerapan.

Beberapa data intranetwork dalam perjalanan (di dalam platform layanan) tidak dienkripsi. Hal ini mencakup:

- Komunikasi perintah dan kontrol antara pesawat kontrol layanan dan instance pekerjaan pelatihan (bukan data pelanggan).
- Komunikasi antar node dalam pekerjaan pemrosesan terdistribusi (intranetwork).
- Komunikasi antar node dalam pekerjaan pelatihan terdistribusi (intranetwork).

Tidak ada komunikasi antar simpul untuk pemrosesan batch.

Anda dapat memilih untuk mengenkripsi komunikasi antar node dalam cluster pelatihan.

> **Note**
>
> Untuk kasus penggunaan di sektor perawatan kesehatan, praktik terbaik untuk keamanan adalah mengenkripsi komunikasi antar node.

Untuk informasi tentang cara mengenkripsi komunikasi, lihat topik berikutnya diMelindungi Komunikasi Antara Instans Komputasi ML dalam Job Pelatihan Terdistribusi.

> **Note**
>
> Mengenkripsi lalu lintas antar kontainer dapat meningkatkan waktu pelatihan, terutama jika Anda menggunakan algoritme pembelajaran mendalam terdistribusi. Untuk algoritma yang terpengaruh, tingkat keamanan tambahan ini juga meningkatkan biaya. Waktu pelatihan untuk sebagian besar algoritma SageMaker bawaan, seperti XGBoost, DeepAR, dan pembelajar linier, biasanya tidak terpengaruh.

Titik akhir yang divalidasi FIPS tersedia untuk SageMaker API dan meminta router untuk model yang dihosting (runtime). Untuk informasi tentang titik akhir yang sesuai dengan FIPS, lihat Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Lindungi Komunikasi dengan RStudio di Amazon SageMaker

RStudio di Amazon SageMaker menyediakan enkripsi untuk semua komunikasi yang melibatkan SageMaker komponen. Namun, versi sebelumnya tidak mendukung enkripsi antara aplikasi R StudioServerPro dan RSession.

RStudio merilis versi 2022.02.2-485.pro2 pada April 2022. Versi ini mendukung enkripsi antara aplikasi R StudioServerPro dan RSession untuk mengaktifkan end-to-end enkripsi. Upgrade versi, bagaimanapun, tidak sepenuhnya kompatibel ke belakang. Akibatnya, Anda harus memperbarui semua aplikasi R StudioServerPro dan RSession Anda. Untuk informasi tentang cara memperbarui aplikasi Anda, lihatTingkatkan Versi RStudio.

Melindungi Komunikasi Antara Instans Komputasi ML dalam Job Pelatihan Terdistribusi

Secara default, Amazon SageMaker menjalankan tugas pelatihan di Amazon Virtual Private Cloud (Amazon VPC) untuk membantu menjaga keamanan data Anda. Anda dapat menambahkan tingkat keamanan lain untuk melindungi wadah pelatihan dan data Anda dengan mengonfigurasi VPC pribadi. Kerangka kerja dan algoritme ML terdistribusi biasanya mengirimkan informasi yang terkait langsung dengan model, seperti bobot, bukan kumpulan data pelatihan. Saat melakukan pelatihan terdistribusi, Anda dapat lebih melindungi data yang ditransmisikan antar instance. Ini dapat membantu Anda mematuhi persyaratan peraturan. Untuk melakukan ini, gunakan enkripsi lalu lintas antar kontainer.

> **Note**
>
> Untuk kasus penggunaan di sektor perawatan kesehatan, praktik terbaik untuk keamanan adalah mengenkripsi komunikasi antar node.

Mengaktifkan enkripsi lalu lintas antar kontainer dapat meningkatkan waktu pelatihan, terutama jika Anda menggunakan algoritme pembelajaran mendalam terdistribusi. Mengaktifkan enkripsi lalu lintas antar kontainer tidak memengaruhi pekerjaan pelatihan dengan satu instance komputasi. Namun, untuk pekerjaan pelatihan dengan beberapa contoh komputasi, efeknya pada waktu pelatihan tergantung pada jumlah komunikasi antara instance komputasi. Untuk algoritma yang terpengaruh, menambahkan tingkat keamanan tambahan ini juga meningkatkan biaya. Waktu pelatihan untuk sebagian besar algoritma SageMaker bawaan, seperti XGBoost, DeepAR, dan pembelajar linier, biasanya tidak terpengaruh.

Anda dapat mengaktifkan enkripsi lalu lintas antar kontainer untuk pekerjaan pelatihan atau pekerjaan tuning hyperparameter. Anda dapat menggunakan SageMaker API atau konsol untuk mengaktifkan enkripsi lalu lintas antar kontainer.

Untuk informasi tentang menjalankan pekerjaan pelatihan di VPC pribadi, lihat. Berikan Akses Pekerjaan SageMaker Pelatihan ke Sumber Daya di VPC Amazon Anda

Aktifkan Enkripsi Lalu Lintas Antar Kontainer (API)

Sebelum mengaktifkan enkripsi lalu lintas antar kontainer pada pelatihan atau pekerjaan penyetelan hiperparameter dengan API, tambahkan aturan masuk dan keluar ke grup keamanan VPC pribadi Anda.

Untuk mengaktifkan enkripsi lalu lintas antar kontainer (API)

1. Tambahkan aturan jalur masuk dan jalur keluar berikut ini di grup keamanan untuk VPC pribadi Anda:

Protokol	Baris Port	Sumber
UDP	500	*ID Grup Keamanan Diri*
ESP 50	N/A	*ID Grup Keamanan Diri*

2. Saat Anda mengirim permintaan ke `CreateHyperParameterTuningJob`API `CreateTrainingJob`atau, tentukan `True` `EnableInterContainerTrafficEncryption` parameternya.

> **Note**
>
> Untuk `ESP 50` protokol, AWS Security Group Console mungkin menampilkan rentang port sebagai “Semua”. Namun, Amazon EC2 mengabaikan rentang port yang ditentukan karena tidak berlaku untuk protokol IP ESP 50.

Aktifkan Enkripsi Lalu Lintas Antar Kontainer (Konsol)

Aktifkan Enkripsi Lalu Lintas Antar Kontainer dalam Training Job

Untuk mengaktifkan enkripsi lalu lintas antar kontainer dalam pekerjaan pelatihan

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilih Pelatihan, lalu pilih Pekerjaan pelatihan.
3. Pilih Buat pekerjaan pelatihan.
4. Di bawah Jaringan, pilih VPC. Anda dapat menggunakan VPC default atau yang telah Anda buat.

5. Pilih Aktifkan enkripsi lalu lintas antar kontainer.

Setelah Anda mengaktifkan enkripsi lalu lintas antar kontainer, selesaikan pembuatan pekerjaan pelatihan. Untuk informasi selengkapnya, lihat Langkah 4: Latih Model.

Aktifkan Enkripsi Lalu Lintas Antar Kontainer dalam Pekerjaan Tuning Hyperparameter

Untuk mengaktifkan enkripsi lalu lintas antar-kontainer dalam pekerjaan penyetelan hyperparameter

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Di panel navigasi, pilih Pelatihan, lalu pilih Pekerjaan tuning Hyperparameter.
3. Pilih Buat pekerjaan tuning hyperparameter.
4. Di bawah Jaringan, pilih VPC. Anda dapat menggunakan VPC default atau yang Anda buat.
5. Pilih Aktifkan enkripsi lalu lintas antar kontainer.

Setelah mengaktifkan enkripsi lalu lintas antar-kontainer, selesaikan pembuatan pekerjaan penyetelan hyperparameter. Untuk informasi selengkapnya, lihat Konfigurasikan dan Luncurkan Pekerjaan Tuning Hyperparameter.

Manajemen kunci

Pelanggan dapat menentukan AWS KMS kunci, termasuk membawa kunci Anda sendiri (BYOK), untuk digunakan untuk enkripsi amplop dengan bucket input/output Amazon S3 dan volume Amazon EBS pembelajaran mesin (ML) Amazon. Volume ML untuk instance notebook dan untuk pemrosesan, pelatihan, dan kontainer model Docker yang dihosting dapat dienkripsi secara opsional dengan menggunakan kunci milik pelanggan. AWS KMS Semua volume OS instance dienkripsi dengan kunci AWS AWS KMS -managed.

> **Note**
>
> Instans berbasis Nitro tertentu menyertakan penyimpanan lokal, tergantung dari tipe instans. Volume penyimpanan lokal dienkripsi menggunakan modul perangkat keras pada instans. Anda tidak dapat meminta `VolumeKmsKeyId` saat menggunakan tipe instans dengan penyimpanan lokal.
> Untuk daftar tipe instans yang mendukung penyimpanan instans lokal, lihat Volume Penyimpanan Instans.

Untuk informasi lebih lanjut tentang enkripsi penyimpanan instans lokal, lihat Volume Penyimpanan Instans SSD.
Untuk informasi selengkapnya tentang volume penyimpanan pada instans berbasis nitro, lihat Amazon EBS dan NVMe di Instans Linux.

Untuk informasi tentang AWS KMS kunci, lihat Apa itu Layanan Manajemen AWS Kunci? di Panduan AWS Key Management Service Pengembang.

Privasi Lalu Lintas Kerja Internet

Topik ini menjelaskan cara Amazon SageMaker mengamankan koneksi dari layanan ke lokasi lain.

Komunikasi internetwork mendukung enkripsi TLS 1.2 antara semua komponen dan klien. Kami merekomendasikan TLS 1.3.

Instans dapat dihubungkan ke VPC Pelanggan, menyediakan akses ke titik akhir VPC S3 atau repositori pelanggan. Jalan keluar internet dapat dikelola melalui antarmuka ini oleh pelanggan jika platform layanan jalan keluar internet dinonaktifkan untuk notebook. Untuk pelatihan dan hosting, jalan keluar melalui platform layanan tidak tersedia saat terhubung ke VPC pelanggan.

Secara default, panggilan API yang dilakukan ke titik akhir yang dipublikasikan melintasi jaringan publik ke router permintaan. SageMaker mendukung titik akhir antarmuka Amazon Virtual Private Cloud yang didukung oleh AWS PrivateLink konektivitas pribadi antara VPC pelanggan dan router permintaan untuk mengakses titik akhir model yang dihosting. Untuk informasi tentang Amazon VPC, lihat Connect ke SageMaker Dalam VPC

Identity and Access Management untuk Amazon SageMaker

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke sumber daya AWS secara aman. Administrator IAM mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan sumber daya. SageMaker IAM adalah layanan Layanan AWS yang dapat Anda gunakan tanpa dikenakan biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan Identitas

- Mengelola Akses Menggunakan Kebijakan
- Cara Amazon SageMaker Bekerja dengan IAM
- Contoh Kebijakan SageMaker Berbasis Identitas Amazon
- Pencegahan Deputi Bingung Lintas Layanan
- SageMaker Peran
- Manajer SageMaker Peran Amazon
- Kontrol akses untuk notebook
- Izin SageMaker API Amazon: Referensi Tindakan, Izin, dan Sumber Daya
- AWSKebijakan Terkelola untuk Amazon SageMaker
- Pemecahan masalah SageMaker identitas dan akses Amazon

Audiens

Cara menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di SageMaker.

Pengguna layanan — Jika Anda menggunakan SageMaker layanan untuk melakukan tugas Anda, administrator Anda akan memberikan kredensi dan izin yang dibutuhkan. Saat Anda menggunakan lebih banyak SageMaker fitur untuk melakukan pekerjaan, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di SageMaker, lihat Pemecahan masalah SageMaker identitas dan akses Amazon.

Administrator layanan — Jika Anda bertanggung jawab atas SageMaker sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke SageMaker. Tugas Anda adalah menentukan SageMaker fitur dan sumber daya mana yang dapat diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari selengkapnya tentang cara perusahaan Anda dapat menggunakan IAM dengan SageMaker, lihatCara Amazon SageMaker Bekerja dengan IAM.

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke SageMaker. Untuk melihat contoh kebijakan SageMaker berbasis identitas yang dapat Anda gunakan di IAM, lihat. Contoh Kebijakan SageMaker Berbasis Identitas Amazon

Mengautentikasi dengan Identitas

Autentikasi adalah cara Anda untuk masuk ke AWS menggunakan kredensial identitas Anda. Anda harus terautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengambil peran IAM.

Anda dapat masuk ke AWS sebagai identitas terfederasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. Pengguna AWS IAM Identity Center Pengguna (Pusat Identitas IAM), autentikasi Single Sign-On perusahaan Anda, dan kredensial Google atau Facebook Anda merupakan contoh identitas terfederasi. Saat Anda masuk sebagai identitas gabungan, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil suatu peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal akses AWS. Untuk informasi selengkapnya tentang cara masuk ke AWS, lihat Cara masuk ke Akun AWS dalam Panduan Pengguna AWS Sign-In.

Jika Anda mengakses AWS secara terprogram, AWS memberikan Kit Pengembangan Perangkat Lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan peralatan AWS, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang cara menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani permintaan API AWS dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Sebagai contoh, AWS menyarankan Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS dalam Panduan Pengguna IAM.

Pengguna root Akun AWS

Ketika membuat Akun AWS, Anda memulai dengan satu identitas masuk yang memiliki akses penuh ke semua Layanan AWS dan sumber daya di akun tersebut. Identitas ini disebut pengguna root Akun AWS dan diakses dengan cara masuk menggunakan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar tugas

lengkap yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Praktik terbaiknya adalah mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensial temporer.

Identitas terfederasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, AWS Directory Service, direktori Pusat Identitas, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas terfederasi mengakses Akun AWS, identitas tersebut mengambil peran, dan peran ini memberikan kredensial sementara.

Untuk pengelolaan akses terpusat, sebaiknya Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apa yang dimaksud Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center.

Pengguna dan Grup IAM

Pengguna IAM adalah identitas dalam Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya andalkan kredensial temporer, dan bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, sebaiknya rotasikan kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran tersebut dimaksudkan untuk dapat diambil oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk

mempelajari selengkapnya, silakan lihat Kapan harus membuat pengguna IAM (bukan peran) dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM merupakan identitas dalam Akun AWS Anda yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara dalam AWS Management Console dengan berganti peran. Anda dapat mengambil peran dengan cara memanggil operasi API AWS CLI atau AWS atau menggunakan URL kustom. Untuk informasi selengkapnya tentang metode untuk menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi diautentikasi, identitas tersebut dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi sekumpulan izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengaitkan izin yang ditetapkan ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat Rangkaian izin dalam Panduan Pengguna AWS IAM Identity Center.
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, pada beberapa Layanan AWS, Anda dapat menyertakan kebijakan secara langsung ke sumber daya (bukan menggunakan peran sebagai proksi). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan – Sebagian Layanan AWS menggunakan fitur di Layanan AWS lainnya. Contoh, ketika Anda melakukan panggilan dalam layanan, umumnya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran terkait layanan.

- Sesi akses maju (FAS) – Ketika Anda menggunakan pengguna IAM atau peran IAM untuk melakukan tindakan di AWS, Anda akan dianggap sebagai seorang pengguna utama. Saat menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya diajukan saat layanan menerima permintaan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Meneruskan sesi akses.
- Peran IAM – Peran layanan adalah peran IAM yang diambil layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran terkait layanan – Peran terkait layanan adalah tipe peran layanan yang terkait dengan Layanan AWS. Layanan tersebut dapat mengambil peran untuk melakukan sebuah tindakan atas nama Anda. Peran terkait layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan peran IAM untuk mengelola kredensial sementara untuk aplikasi yang berjalan di instans EC2 dan mengajukan permintaan API AWS CLI atau AWS. Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan peran AWS ke instans EC2 dan menyediakannya bagi semua aplikasinya, Anda dapat membuat profil instans yang dilampirkan ke instans tersebut. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam Panduan Pengguna IAM.

Mengelola Akses Menggunakan Kebijakan

Anda mengendalikan akses di AWS dengan membuat kebijakan dan melampirkannya ke identitas atau sumber daya AWS. Kebijakan adalah objek di AWS yang, ketika terkait dengan identitas atau sumber daya, akan menentukan izinnya. AWS mengevaluasi kebijakan-kebijakan tersebut ketika

seorang pengguna utama (pengguna, pengguna root, atau sesi peran) mengajukan permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, silakan lihat Gambaran Umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses terhadap apa. Artinya, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari AWS Management Console, AWS CLI, atau API AWS.

Kebijakan Berbasis Identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di Akun AWS Anda. Kebijakan terkelola meliputi kebijakan yang dikelola AWS dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat Memilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan Berbasis Sumber Daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya,

administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan yang dikelola AWS dari IAM dalam kebijakan berbasis sumber daya.

Daftar Kontrol Akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat Gambaran umum daftar kontrol akses (ACL) di Panduan Developer Layanan Penyimpanan Ringkas Amazon.

Tipe Kebijakan Lainnya

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola beberapa akun Akun AWS yang dimiliki bisnis Anda secara terpusat. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun AWS.

Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations.

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit di salah satu kebijakan ini akan membatalkan izin tersebut. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai Tipe Kebijakan

Jika beberapa jenis kebijakan diberlakukan untuk satu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan ketika ada beberapa jenis kebijakan, lihat Logika evaluasi kebijakan dalam Panduan Pengguna IAM.

Cara Amazon SageMaker Bekerja dengan IAM

Sebelum menggunakan IAM untuk mengelola akses ke SageMaker, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan dengan. SageMaker Untuk mendapatkan tampilan tingkat tinggi tentang cara SageMaker dan AWS layanan lainnya bekerja dengan IAM, lihat AWSLaynan Yang Bekerja dengan IAM di Panduan Pengguna IAM.

Topik

- Kebijakan Berbasis Identitas SageMaker

Kebijakan Berbasis Identitas SageMaker

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta ketentuan terkait jenis tindakan yang diizinkan atau ditolak. SageMaker mendukung tindakan, sumber daya, dan kunci syarat tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat Referensi Elemen Kebijakan JSON IAM dalam Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama seperti operasi API AWS terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin melakukan operasi terkait.

Tindakan kebijakan di SageMaker menggunakan prefiks berikut sebelum tindakan:`sagemaker:`. Misalnya, untuk memberikan izin kepada seseorang untuk menjalankan tugas SageMaker pelatihan dengan operasi SageMaker `CreateTrainingJob` API, Anda mencantumkan `sagemaker:CreateTrainingJob` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction` elemen. SageMaker menentukan serangkaian tindakannya sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
      "sagemaker:action1",
      "sagemaker:action2"
]
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "sagemaker:Describe*"
```

Untuk melihat daftar SageMaker tindakan, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon SageMaker di Referensi Otorisasi Layanan.

Sumber daya

SageMaker tidak mendukung penentuan ARN sumber daya dalam kebijakan.

Kunci Syarat

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi kondisional yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam satu pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya dengan menggunakan operasi `AND` logis. Jika Anda menentukan beberapa niilai untuk satu kunci persyaratan, AWS akan mengevaluasi syarat tersebut menggunakan operasi `OR` yang logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, silakan lihat Elemen kebijakan IAM: variabel dan tanda di Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi spesifik layanan. Untuk melihat semua kunci kondisi global AWS, lihat kunci konteks kondisi global AWS dalam Panduan Pengguna IAM.

SageMaker menentukan set kunci kondisinya sendiri dan juga mendukung penggunaan beberapa kunci kondisi global. Untuk melihat semua kunci syarat global AWS, lihat Kunci Konteks Syarat Global AWS dalam Panduan Pengguna IAM.

SageMaker mendukung sejumlah kunci kondisi khusus layanan yang dapat Anda gunakan untuk kontrol akses berbutir halus untuk operasi berikut:

- `CreateProcessingJob`
- `CreateTrainingJob`
- `CreateModel`
- `CreateEndpointConfig`
- `CreateTransformJob`

- `CreateHyperParameterTuningJob`
- `CreateLabelingJob`
- `CreateNotebookInstance`
- `UpdateNotebookInstance`

Untuk melihat daftar kunci SageMaker kondisi, lihat Kunci syarat untuk Amazon SageMaker di Panduan Pengguna IAM. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan dengan kunci syarat, lihat Tindakan yang ditentukan oleh Amazon SageMaker.

Untuk contoh menggunakan tombol SageMaker kondisi, lihat yang berikut ini: Kontrol Pembuatan Sumber SageMaker Daya dengan Kunci Kondisi

Contoh-contoh

Untuk melihat contoh kebijakan SageMaker berbasis identitas, lihat. Contoh Kebijakan SageMaker Berbasis Identitas Amazon

SageMaker Kebijakan Berbasis Sumber Daya

SageMaker tidak mendukung kebijakan berbasis sumber daya.

Otorisasi Berdasarkan Tag SageMaker

Anda dapat melampirkan tag ke SageMaker sumber daya atau meneruskan tag dalam sebuah permintaan ke SageMaker. Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tanda di elemen syarat dari sebuh kebijakan dengan menggunakan kunci-kunci persyaratan `sagemaker:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`. Untuk informasi selengkapnya tentang penandaan sumber daya SageMaker , lihat Kontrol Akses ke SageMaker Sumber Daya dengan Menggunakan Tag.

Untuk melihat contoh kebijakan berbasis identitas untuk membatasi akses ke sumber daya berdasarkan tag pada sumber daya tersebut, lihat Kontrol Akses ke SageMaker Sumber Daya dengan Menggunakan Tag.

SageMaker IAM Role

IAM role adalah entitas di dalam akun AWS Anda yang memiliki izin tertentu.

Menggunakan kredensi sementara dengan SageMaker

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensyal keamanan sementara dengan memanggil operasi AWS STS API seperti AssumeRoleatau. GetFederationToken

SageMaker mendukung penggunaan kredensi sementara.

Peran Tertaut Layanan

SageMaker sebagian mendukung peran terkait layanan. Peran terkait layanan saat ini tersedia untuk SageMaker Studio Classic dan pekerjaan SageMaker pelatihan.

Peran Layanan

Fitur ini memungkinkan layanan untuk menerima peran layanan atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di akun IAM Anda dan dimiliki oleh akun tersebut. Ini berarti administrator IAM dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

SageMaker mendukung peran layanan.

Memilih IAM Role di SageMaker

Saat Anda membuat instans buku catatan, tugas pemrosesan, tugas pelatihan, titik akhir yang di-hosting, atau sumber daya pekerjaan transformasi batch di SageMaker, Anda harus memilih peran SageMaker agar dapat diakses SageMaker atas nama Anda. Jika sebelumnya Anda telah membuat peran layanan atau peran terkait layanan, maka SageMaker berikan daftar peran yang dapat dipilih. Penting untuk memilih peran yang mengizinkan akses ke AWS operasi dan sumber daya yang Anda butuhkan. Untuk informasi selengkapnya, lihat SageMaker Peran.

Contoh Kebijakan SageMaker Berbasis Identitas Amazon

Secara default, pengguna dan peran IAM tidak memiliki izin untuk membuat atau memodifikasi sumber daya SageMaker . Mereka juga tidak dapat melakukan tugas menggunakan API AWS Management Console, AWS CLI, or AWS. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna IAM atau grup yang memerlukan izin tersebut. Untuk mempelajari cara melampirkan kebijakan

ke pengguna atau grup IAM, lihat Menambahkan dan Menghapus Izin Identitas IAM di Panduan Pengguna IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat Kebijakan pada Tab JSON dalam Panduan Pengguna IAM.

Topik

- Praktik Terbaik Kebijakan
- Menggunakan Konsol SageMaker
- Izinkan Pengguna untuk Melihat Izin Mereka Sendiri
- Kontrol Pembuatan Sumber SageMaker Daya dengan Kunci Kondisi
- Kontrol Akses ke SageMaker API dengan Menggunakan Kebijakan Berbasis Identitas
- Batasi Akses ke SageMaker API dan Panggilan Runtime berdasarkan Alamat IP
- Batasi Akses ke Instance Notebook berdasarkan Alamat IP
- Kontrol Akses ke SageMaker Sumber Daya dengan Menggunakan Tag
- Berikan Izin untuk Menandai SageMaker Sumber Daya Saat Pembuatan

Praktik Terbaik Kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus SageMaker sumber daya di akun Anda. Tindakan ini dikenai biaya untuk Akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulai menggunakan kebijakan yang dikelola AWS dan beralih ke izin dengan hak akses paling rendah – Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan yang dikelola AWS yang memberikan izin untuk banyak kasus penggunaan umum. Kebijakan ini ada di Akun AWS Anda. Sebaiknya Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan AWS yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan yang dikelola AWS atau kebijakan yang dikelola AWS untuk fungsi pekerjaan di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang

cara menggunakan IAM untuk menerapkan izin, lihat Kebijakan dan izin di IAM di Panduan Pengguna IAM.

- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan harus dikirim menggunakan SSL. Anda juga dapat menggunakan kondisi untuk memberi akses ke tindakan layanan jika digunakan melalui Layanan AWS yang spesifk, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Syarat di Panduan Pengguna IAM.
- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat validasi kebijakan Analizer Akses IAM di Panduan Pengguna IAM.
- Wajibkan autentikasi multi-faktor (MFA) – Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Akun AWS Anda, aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Menggunakan Konsol SageMaker

Untuk mengakses SageMaker konsol Amazon, Anda harus memiliki rangkaian izin minimum. Izin ini harus memperbolehkan Anda untuk membuat daftar dan melihat perincian tentang sumber daya SageMaker di akun AWS Anda. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Untuk memastikan bahwa entitas tersebut masih dapat menggunakan SageMaker konsol, lampirkan kebijakan AWS terkelola berikut ini ke entitas. Untuk informasi selengkapnya, lihat Menambahkan izin ke Pengguna dalam Panduan Pengguna IAM:

Anda tidak perlu mengizinkan konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau API AWS. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Topik

- Izin yang Diperlukan untuk Menggunakan Konsol Amazon SageMaker
- Izin yang Diperlukan untuk Menggunakan Konsol Amazon SageMaker Ground Truth
- Izin yang Diperlukan untuk Menggunakan Konsol Amazon Augmented AI (Pratinjau)

Izin yang Diperlukan untuk Menggunakan Konsol Amazon SageMaker

Tabel referensi perizinan mencantumkan operasi Amazon SageMaker API dan menunjukkan izin yang diperlukan untuk setiap operasi. Untuk informasi selengkapnya tentang operasi SageMaker API Amazon, lihatIzin SageMaker API Amazon: Referensi Tindakan, Izin, dan Sumber Daya.

Untuk menggunakan SageMaker konsol Amazon, Anda harus memberikan izin untuk tindakan tambahan. Secara khusus, konsol memerlukan izin yang memungkinkan `ec2` tindakan untuk menampilkan subnet, VPC, dan grup keamanan. Secara opsional, konsol memerlukan izin untuk membuat peran eksekusi untuk tugas-tugas seperti`CreateNotebook`,`CreateTrainingJob`, dan`CreateModel`. Berikan izin ini dengan kebijakan izin berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
          "Sid": "SageMakerApis",
          "Effect": "Allow",
          "Action": [
            "sagemaker:*"
          ],
          "Resource": "*"
        },
        {
          "Sid": "VpcConfigurationForCreateForms",
          "Effect": "Allow",
          "Action": [
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
          ],
```

```
            "Resource": "*"
        },
        {
            "Sid":"KmsKeysForCreateForms",
            "Effect":"Allow",
            "Action":[
              "kms:DescribeKey",
              "kms:ListAliases"
            ],
            "Resource":"*"
        },
        {
          "Sid": "AccessAwsMarketplaceSubscriptions",
          "Effect": "Allow",
          "Action": [
            "aws-marketplace:ViewSubscriptions"
          ],
          "Resource": "*"
        },
        {
          "Effect": "Allow",
          "Action": [
            "codecommit:BatchGetRepositories",
            "codecommit:CreateRepository",
            "codecommit:GetRepository",
            "codecommit:ListRepositories",
            "codecommit:ListBranches",
            "secretsmanager:CreateSecret",
            "secretsmanager:DescribeSecret",
            "secretsmanager:ListSecrets"
          ],
          "Resource": "*"
        },
        {
          "Sid":"ListAndCreateExecutionRoles",
          "Effect":"Allow",
          "Action":[
            "iam:ListRoles",
            "iam:CreateRole",
            "iam:CreatePolicy",
            "iam:AttachRolePolicy"
          ],
          "Resource":"*"
        },
```

```
        {
          "Sid": "DescribeECRMetaData",
          "Effect": "Allow",
          "Action": [
              "ecr:Describe*"
          ],
          "Resource": "*"
        },
        {
          "Sid": "PassRoleForExecutionRoles",
          "Effect": "Allow",
          "Action": [
            "iam:PassRole"
          ],
          "Resource": "*",
          "Condition": {
            "StringEquals": {
                "iam:PassedToService": "sagemaker.amazonaws.com"
            }
          }
        }
    ]
}
```

Izin yang Diperlukan untuk Menggunakan Konsol Amazon SageMaker Ground Truth

Untuk menggunakan konsol Amazon SageMaker Ground Truth, Anda harus memberikan izin untuk sumber daya tambahan. Secara khusus, konsol memerlukan izin bagi AWS Marketplace untuk melihat langganan, operasi Amazon Cognito untuk mengelola tenaga kerja pribadi Anda, tindakan Amazon S3 untuk akses ke file input dan output AndaAWS Lambda, serta tindakan untuk membuat daftar dan menjalankan fungsi. Berikan izin ini dengan kebijakan izin berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GroundTruthConsole",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:DescribeListings",
                "aws-marketplace:ViewSubscriptions",
```

```
                "cognito-idp:AdminAddUserToGroup",
                "cognito-idp:AdminCreateUser",
                "cognito-idp:AdminDeleteUser",
                "cognito-idp:AdminDisableUser",
                "cognito-idp:AdminEnableUser",
                "cognito-idp:AdminRemoveUserFromGroup",
                "cognito-idp:CreateGroup",
                "cognito-idp:CreateUserPool",
                "cognito-idp:CreateUserPoolClient",
                "cognito-idp:CreateUserPoolDomain",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolClient",
                "cognito-idp:ListGroups",
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListUsers",
                "cognito-idp:ListUsersInGroup",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:UpdateUserPool",
                "cognito-idp:UpdateUserPoolClient",

                "groundtruthlabeling:DescribeConsoleJob",
                "groundtruthlabeling:ListDatasetObjects",
                "groundtruthlabeling:RunFilterOrSampleManifestJob",
                "groundtruthlabeling:RunGenerateManifestByCrawlingJob",

                "lambda:InvokeFunction",
                "lambda:ListFunctions",

                "s3:GetObject",
                "s3:PutObject",
                "s3:SelectObjectContent"
            ],
            "Resource": "*"
        }
    ]
}
```

Izin yang Diperlukan untuk Menggunakan Konsol Amazon Augmented AI (Pratinjau)

Untuk menggunakan konsol Augmented AI, Anda harus memberikan izin untuk sumber daya tambahan. Berikan izin ini dengan kebijakan izin berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:*Algorithm",
                "sagemaker:*Algorithms",
                "sagemaker:*App",
                "sagemaker:*Apps",
                "sagemaker:*AutoMLJob",
                "sagemaker:*AutoMLJobs",
                "sagemaker:*CodeRepositories",
                "sagemaker:*CodeRepository",
                "sagemaker:*CompilationJob",
                "sagemaker:*CompilationJobs",
                "sagemaker:*Endpoint",
                "sagemaker:*EndpointConfig",
                "sagemaker:*EndpointConfigs",
                "sagemaker:*EndpointWeightsAndCapacities",
                "sagemaker:*Endpoints",
                "sagemaker:*Environment",
                "sagemaker:*EnvironmentVersion",
                "sagemaker:*EnvironmentVersions",
                "sagemaker:*Environments",
                "sagemaker:*Experiment",
                "sagemaker:*Experiments",
                "sagemaker:*FlowDefinitions",
                "sagemaker:*HumanLoop",
                "sagemaker:*HumanLoops",
                "sagemaker:*HumanTaskUi",
                "sagemaker:*HumanTaskUis",
                "sagemaker:*HyperParameterTuningJob",
                "sagemaker:*HyperParameterTuningJobs",
                "sagemaker:*LabelingJob",
                "sagemaker:*LabelingJobs",
                "sagemaker:*Metrics",
                "sagemaker:*Model",
                "sagemaker:*ModelPackage",
                "sagemaker:*ModelPackages",
                "sagemaker:*Models",
                "sagemaker:*MonitoringExecutions",
                "sagemaker:*MonitoringSchedule",
```

```
                "sagemaker:*MonitoringSchedules",
                "sagemaker:*NotebookInstance",
                "sagemaker:*NotebookInstanceLifecycleConfig",
                "sagemaker:*NotebookInstanceLifecycleConfigs",
                "sagemaker:*NotebookInstanceUrl",
                "sagemaker:*NotebookInstances",
                "sagemaker:*ProcessingJob",
                "sagemaker:*ProcessingJobs",
                "sagemaker:*RenderUiTemplate",
                "sagemaker:*Search",
                "sagemaker:*SearchSuggestions",
                "sagemaker:*Tags",
                "sagemaker:*TrainingJob",
                "sagemaker:*TrainingJobs",
                "sagemaker:*TransformJob",
                "sagemaker:*TransformJobs",
                "sagemaker:*Trial",
                "sagemaker:*TrialComponent",
                "sagemaker:*TrialComponents",
                "sagemaker:*Trials",
                "sagemaker:*Workteam",
                "sagemaker:*Workteams"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:*FlowDefinition"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "sagemaker:WorkteamType": [
                        "private-crowd",
                        "vendor-crowd"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeleteScalingPolicy",
```

```
                "application-autoscaling:DeleteScheduledAction",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:DescribeScheduledActions",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:PutScheduledAction",
                "application-autoscaling:RegisterScalableTarget",
                "aws-marketplace:ViewSubscriptions",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:PutMetricData",
                "codecommit:BatchGetRepositories",
                "codecommit:CreateRepository",
                "codecommit:GetRepository",
                "codecommit:ListBranches",
                "codecommit:ListRepositories",
                "cognito-idp:AdminAddUserToGroup",
                "cognito-idp:AdminCreateUser",
                "cognito-idp:AdminDeleteUser",
                "cognito-idp:AdminDisableUser",
                "cognito-idp:AdminEnableUser",
                "cognito-idp:AdminRemoveUserFromGroup",
                "cognito-idp:CreateGroup",
                "cognito-idp:CreateUserPool",
                "cognito-idp:CreateUserPoolClient",
                "cognito-idp:CreateUserPoolDomain",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolClient",
                "cognito-idp:ListGroups",
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:ListUsers",
                "cognito-idp:ListUsersInGroup",
                "cognito-idp:UpdateUserPool",
                "cognito-idp:UpdateUserPoolClient",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
```

```
                "ec2:CreateVpcEndpoint",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcs",
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:CreateRepository",
                "ecr:Describe*",
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
                "elastic-inference:Connect",
                "elasticfilesystem:DescribeFileSystems",
                "elasticfilesystem:DescribeMountTargets",
                "fsx:DescribeFileSystems",
                "glue:CreateJob",
                "glue:DeleteJob",
                "glue:GetJob",
                "glue:GetJobRun",
                "glue:GetJobRuns",
                "glue:GetJobs",
                "glue:ResetJobBookmark",
                "glue:StartJobRun",
                "glue:UpdateJob",
                "groundtruthlabeling:*",
                "iam:ListRoles",
                "kms:DescribeKey",
                "kms:ListAliases",
                "lambda:ListFunctions",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "logs:PutLogEvents",
                "sns:ListTopics"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:GetLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:SetRepositoryPolicy",
                "ecr:CompleteLayerUpload",
                "ecr:BatchDeleteImage",
                "ecr:UploadLayerPart",
                "ecr:DeleteRepositoryPolicy",
                "ecr:InitiateLayerUpload",
                "ecr:DeleteRepository",
                "ecr:PutImage"
            ],
            "Resource": "arn:aws:ecr:*:*:repository/*sagemaker*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "codecommit:GitPull",
                "codecommit:GitPush"
            ],
            "Resource": [
                "arn:aws:codecommit:*:*:*sagemaker*",
                "arn:aws:codecommit:*:*:*SageMaker*",
                "arn:aws:codecommit:*:*:*Sagemaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecrets"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:CreateSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:*:*:secret:AmazonSageMaker-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:ResourceTag/SageMaker": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "robomaker:CreateSimulationApplication",
                "robomaker:DescribeSimulationApplication",
                "robomaker:DeleteSimulationApplication"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "robomaker:CreateSimulationJob",
                "robomaker:DescribeSimulationJob",
                "robomaker:CancelSimulationJob"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:AbortMultipartUpload",
                "s3:GetBucketCors",
                "s3:PutBucketCors"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*",
                "arn:aws:s3:::*aws-glue*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "s3:ExistingObjectTag/SageMaker": "true"
                }
            }
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:*SageMaker*",
                "arn:aws:lambda:*:*:function:*sagemaker*",
                "arn:aws:lambda:*:*:function:*Sagemaker*",
                "arn:aws:lambda:*:*:function:*LabelingFunction*"
            ]
        },
        {
            "Action": "iam:CreateServiceLinkedRole",
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/aws-service-role/sagemaker.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "sagemaker.application-
autoscaling.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "robomaker.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:Subscribe",
                "sns:CreateTopic"
            ],
            "Resource": [
                "arn:aws:sns:*:*:*SageMaker*",
                "arn:aws:sns:*:*:*Sagemaker*",
                "arn:aws:sns:*:*:*sagemaker*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "sagemaker.amazonaws.com",
                        "glue.amazonaws.com",
                        "robomaker.amazonaws.com",
                        "states.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Izinkan Pengguna untuk Melihat Izin Mereka Sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan pada konsol atau menggunakan AWS CLI atau AWS API secara terprogram.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
```

```
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Kontrol Pembuatan Sumber SageMaker Daya dengan Kunci Kondisi

Kontrol akses berbutir halus untuk memungkinkan pembuatan SageMaker sumber daya dengan menggunakan kunci kondisi SageMaker -spesifik. Untuk informasi tentang penggunaan kunci kondisi dalam kebijakan IAM, lihat elemen kebijakan IAM JSON: Kondisi di Panduan Pengguna IAM.

Kunci kondisi, bersama dengan tindakan API terkait, dan tautan ke dokumentasi yang relevan tercantum dalam Kunci Kondisi untuk SageMaker di Panduan Pengguna IAM.

Contoh berikut menunjukkan cara menggunakan tombol SageMaker kondisi untuk mengontrol akses.

Topik

- Kontrol Akses ke SageMaker Sumber Daya dengan Menggunakan Tombol Kondisi Sistem File
- Batasi Pelatihan ke VPC Tertentu
- Batasi Akses ke Jenis Tenaga Kerja untuk Pekerjaan Pelabelan Ground Truth dan Alur Kerja Tinjauan Manusia Amazon A2I
- Menegakkan Enkripsi Data Input
- Menerapkan Enkripsi Volume Penyimpanan Instans Notebook
- Menegakkan Isolasi Jaringan untuk Pekerjaan Pelatihan
- Menerapkan Jenis Instance Tertentu untuk Pekerjaan Pelatihan
- Menerapkan Akselerator EI Khusus untuk Pekerjaan Pelatihan

- Menegakkan Menonaktifkan Akses Internet dan Akses Root untuk Membuat Instans Notebook

Kontrol Akses ke SageMaker Sumber Daya dengan Menggunakan Tombol Kondisi Sistem File

SageMaker pelatihan menyediakan infrastruktur yang aman untuk menjalankan algoritme pelatihan, tetapi untuk beberapa kasus Anda mungkin ingin meningkatkan pertahanan secara mendalam. Misalnya, Anda meminimalkan risiko menjalankan kode yang tidak tepercaya dalam algoritme Anda, atau Anda memiliki mandat keamanan khusus di organisasi Anda. Untuk skenario ini, Anda dapat menggunakan kunci kondisi khusus layanan dalam elemen Kondisi kebijakan IAM untuk menjangkau pengguna ke sistem file tertentu, direktori, mode akses (baca-tulis, hanya-baca) dan grup keamanan.

Topik

- Batasi Pengguna IAM ke Direktori dan Mode Akses Tertentu
- Membatasi Pengguna ke Sistem File Tertentu

Batasi Pengguna IAM ke Direktori dan Mode Akses Tertentu

Kebijakan di bawah ini membatasi pengguna ke `/sagemaker/xgboost-dm/train` dan `/sagemaker/xgboost-dm/validation` direktori sistem file EFS untuk `ro` (hanya-baca): AccessMode

> **Note**
>
> Ketika sebuah direktori diizinkan, semua subdirektorinya juga dapat diakses oleh algoritma pelatihan. Izin POSIX diabaikan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToElasticFileSystem",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateHyperParameterTuningJob"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "sagemaker:FileSystemId": "fs-12345678",
                    "sagemaker:FileSystemAccessMode": "ro",
                    "sagemaker:FileSystemType": "EFS",
                    "sagemaker:FileSystemDirectoryPath": "/sagemaker/xgboost-dm/train"
                }
            }
        },
        {
            "Sid": "AccessToElasticFileSystemValidation",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateHyperParameterTuningJob"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "sagemaker:FileSystemId": "fs-12345678",
                    "sagemaker:FileSystemAccessMode": "ro",
                    "sagemaker:FileSystemType": "EFS",
                    "sagemaker:FileSystemDirectoryPath": "/sagemaker/xgboost-dm/
validation"
                }
            }
        }
    ]
}
```

Membatasi Pengguna ke Sistem File Tertentu

Untuk mencegah algoritma berbahaya menggunakan klien ruang pengguna mengakses sistem file apa pun secara langsung di akun Anda, Anda dapat membatasi lalu lintas jaringan dengan mengizinkan masuknya dari grup keamanan tertentu. Dalam contoh berikut, pengguna hanya dapat menggunakan grup keamanan yang ditentukan untuk mengakses sistem file:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToLustreFileSystem",
            "Effect": "Allow",
            "Action": [
```

```
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateHyperParameterTuningJob"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "sagemaker:FileSystemId": "fs-12345678",
                    "sagemaker:FileSystemAccessMode": "ro",
                    "sagemaker:FileSystemType": "FSxLustre",
                    "sagemaker:FileSystemDirectoryPath": "/fsx/sagemaker/xgboost/train"
                },
                "ForAllValues:StringEquals": {
                    "sagemaker:VpcSecurityGroupIds": [
                        "sg-12345678"
                    ]
                }
            }
        }
    ]
}
```

Meskipun contoh di atas dapat membatasi algoritma ke sistem file tertentu, itu tidak mencegah algoritma mengakses direktori apa pun dalam sistem file itu menggunakan klien ruang pengguna. Untuk mengurangi ini, Anda dapat:

- Pastikan bahwa sistem file hanya berisi data yang Anda percayai untuk diakses oleh pengguna
- Buat peran IAM yang membatasi pengguna Anda untuk meluncurkan pekerjaan pelatihan dengan algoritme dari repositori ECR yang disetujui

Untuk informasi selengkapnya tentang cara menggunakan peran SageMaker, lihat SageMaker Peran.

Batasi Pelatihan ke VPC Tertentu

Batasi AWS pengguna untuk membuat pekerjaan pelatihan dari dalam VPC Amazon. Ketika pekerjaan pelatihan dibuat dalam VPC, Anda dapat menggunakan log aliran VPC untuk memantau semua lalu lintas ke dan dari cluster pelatihan. Untuk informasi tentang penggunaan log aliran VPC, lihat Log Aliran VPC di Panduan Pengguna Amazon Virtual Private Cloud.

Kebijakan berikut memberlakukan bahwa pekerjaan pelatihan dibuat oleh pengguna yang menelepon `CreateTrainingJob`dari dalam VPC:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFromVpc",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateHyperParameterTuningJob"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "sagemaker:VpcSubnets": ["subnet-a1234"],
                    "sagemaker:VpcSecurityGroupIds": ["sg12345", "sg-67890"]
                },
                "Null": {
                    "sagemaker:VpcSubnets": "false",
                    "sagemaker:VpcSecurityGroupIds": "false"
                }
            }
        }

    ]
}
```

Batasi Akses ke Jenis Tenaga Kerja untuk Pekerjaan Pelabelan Ground Truth dan Alur Kerja Tinjauan Manusia Amazon A2I

Amazon SageMaker Ground Truth dan tim kerja Amazon Augmented AI termasuk dalam salah satu dari tiga jenis tenaga kerja: publik (dengan Amazon Mechanical Turk), swasta, dan vendor. Untuk membatasi akses pengguna ke tim kerja tertentu menggunakan salah satu jenis ini atau ARN tim kerja, gunakan dan/atau `sagemaker:WorkteamType` `sagemaker:WorkteamArn` tombol kondisi. Untuk kunci `sagemaker:WorkteamType` kondisi, gunakan operator kondisi string. Untuk kunci `sagemaker:WorkteamArn` kondisi, gunakan operator ketentuan Amazon Resource Name (ARN). Jika pengguna mencoba membuat pekerjaan pelabelan dengan tim kerja terbatas, SageMaker mengembalikan kesalahan akses ditolak.

Kebijakan di bawah ini menunjukkan berbagai cara untuk menggunakan kunci `sagemaker:WorkteamType` dan `sagemaker:WorkteamArn` kondisi dengan operator kondisi yang sesuai dan nilai kondisi yang valid.

Contoh berikut menggunakan kunci `sagemaker:WorkteamType` kondisi dengan operator `StringEquals` kondisi untuk membatasi akses ke tim kerja umum. Ini menerima nilai kondisi dalam format berikut:*`workforcetype`*`-crowd`, di mana *`workforcetype`* dapat sama,, atau. `public` `private` `vendor`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RestrictWorkteamType",
            "Effect": "Deny",
            "Action": "sagemaker:CreateLabelingJob",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "sagemaker:WorkteamType": "public-crowd"
                }
            }
        }
    ]
}
```

Kebijakan berikut menunjukkan cara membatasi akses ke tim kerja publik menggunakan kunci `sagemaker:WorkteamArn` kondisi. Yang pertama menunjukkan cara menggunakannya dengan varian regex IAM yang valid dari tim kerja ARN dan operator kondisi. `ArnLike` Yang kedua menunjukkan bagaimana menggunakannya dengan operator `ArnEquals` kondisi dan tim kerja ARN.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RestrictWorkteamType",
            "Effect": "Deny",
            "Action": "sagemaker:CreateLabelingJob",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "sagemaker:WorkteamArn": "arn:aws:sagemaker:*:*:workteam/public-
crowd/*"
                }
            }
        }
```

```
    ]
}
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RestrictWorkteamType",
            "Effect": "Deny",
            "Action": "sagemaker:CreateLabelingJob",
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "sagemaker:WorkteamArn": "arn:aws:sagemaker:us-
west-2:394669845002:workteam/public-crowd/default"
                }
            }
        }
    ]
}
```

Menegakkan Enkripsi Data Input

Kebijakan berikut membatasi pengguna untuk menentukan AWS KMS kunci untuk mengenkripsi data input saat membuat tugas pelatihan, penyetelan hiperparameter, dan pelabelan dengan menggunakan kunci kondisi: `sagemaker:VolumeKmsKey`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnforceEncryption",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateHyperParameterTuningJob",
                "sagemaker:CreateLabelingJob",
                "sagemaker:CreateFlowDefiniton"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
```

```
                    "sagemaker:VolumeKmsKey": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
                }
            }
        }

    ]
}
```

Menerapkan Enkripsi Volume Penyimpanan Instans Notebook

Kebijakan berikut membatasi pengguna untuk menentukan AWS KMS kunci untuk mengenkripsi volume penyimpanan terlampir saat membuat atau memperbarui instance notebook menggunakan kunci `sagemaker:VolumeKmsKey` kondisi:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnforceEncryption",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateNotebookInstance"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "sagemaker:VolumeKmsKey": "*key/volume-kms-key-12345"
                }
            }
        }

    ]
}
```

Menegakkan Isolasi Jaringan untuk Pekerjaan Pelatihan

Kebijakan berikut membatasi pengguna untuk mengaktifkan isolasi jaringan saat membuat pekerjaan pelatihan dengan menggunakan kunci `sagemaker:NetworkIsolation` kondisi:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "EnforceIsolation",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateHyperParameterTuningJob"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "sagemaker:NetworkIsolation": "true"
                }
            }
        }
    ]
}
```

Menerapkan Jenis Instance Tertentu untuk Pekerjaan Pelatihan

Kebijakan berikut membatasi pengguna untuk menggunakan jenis instans tertentu saat membuat pekerjaan pelatihan menggunakan kunci `sagemaker:InstanceTypes` kondisi:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnforceInstanceType",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateHyperParameterTuningJob"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "sagemaker:InstanceTypes": ["ml.c5.*"]
                }
            }
        }

     ]
}
```

Menerapkan Akselerator EI Khusus untuk Pekerjaan Pelatihan

Kebijakan berikut membatasi pengguna untuk menggunakan akselerator inferensi elastis (EI) tertentu, jika akselerator disediakan, saat membuat atau memperbarui instance notebook dan saat membuat konfigurasi titik akhir dengan menggunakan kunci kondisi: `sagemaker:AcceleratorTypes`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnforceAcceleratorType",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateNotebookInstance",
                "sagemaker:UpdateNotebookInstance",
                "sagemaker:CreateEndpointConfig"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "sagemaker:AcceleratorTypes": ["ml.eia1.medium"]
                }
            }
        }

     ]
}
```

Menegakkan Menonaktifkan Akses Internet dan Akses Root untuk Membuat Instans Notebook

Anda dapat menonaktifkan akses internet dan akses root ke instance notebook untuk membantu membuatnya lebih aman. Untuk informasi tentang mengontrol akses root ke instance notebook, lihatKontrol akses root ke instance SageMaker notebook. untuk informasi tentang menonaktifkan akses internet untuk instance notebook, lihat. Hubungkan Instance Notebook di VPC ke Sumber Daya Eksternal

Kebijakan berikut mengharuskan pengguna untuk menonaktifkan akses jaringan saat membuat instance, dan menonaktifkan akses root saat membuat atau memperbarui instance notebook.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "LockDownCreateNotebookInstance",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateNotebookInstance"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "sagemaker:DirectInternetAccess": "Disabled",
                    "sagemaker:RootAccess": "Disabled"
                },
                "Null": {
                  "sagemaker:VpcSubnets": "false",
                  "sagemaker:VpcSecurityGroupIds": "false"
                }
            }
        },
        {
            "Sid": "LockDownUpdateNotebookInstance",
            "Effect": "Allow",
            "Action": [
                "sagemaker:UpdateNotebookInstance"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "sagemaker:RootAccess": "Disabled"
                }
            }
        }
     ]
}
```

Kontrol Akses ke SageMaker API dengan Menggunakan Kebijakan Berbasis Identitas

Untuk mengontrol akses ke panggilan SageMaker API dan panggilan ke titik akhir yang SageMaker di-host, gunakan kebijakan IAM berbasis identitas.

Topik

- Batasi Akses ke SageMaker API dan Runtime ke Panggilan dari Dalam VPC Anda

Batasi Akses ke SageMaker API dan Runtime ke Panggilan dari Dalam VPC Anda

Jika Anda menyiapkan titik akhir antarmuka di VPC Anda, individu di luar VPC masih dapat terhubung ke SageMaker API dan runtime melalui internet kecuali Anda melampirkan kebijakan IAM yang membatasi akses ke panggilan yang berasal dari dalam VPC ke semua pengguna dan grup yang memiliki akses ke sumber daya Anda. SageMaker Untuk informasi tentang membuat titik akhir antarmuka VPC untuk SageMaker API dan runtime, lihat. Connect ke SageMaker Dalam VPC

> **Important**
>
> Jika Anda menerapkan kebijakan IAM yang mirip dengan salah satu kebijakan berikut, pengguna tidak dapat mengakses SageMaker API yang ditentukan melalui konsol.

Untuk membatasi akses hanya ke koneksi yang dibuat dari dalam VPC Anda, buat kebijakan AWS Identity and Access Management yang membatasi akses hanya ke panggilan yang berasal dari dalam VPC Anda. Kemudian tambahkan kebijakan tersebut ke setiap AWS Identity and Access Management pengguna, grup, atau peran yang digunakan untuk mengakses SageMaker API atau runtime.

> **Note**
>
> Kebijakan ini mengizinkan koneksi hanya ke penelepon dalam subnet tempat Anda membuat titik akhir antarmuka.

```
{
    "Id": "api-example-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableAPIAccess",
            "Effect": "Allow",
            "Action": [
                "sagemaker:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpc": "vpc-111bbaaa"
```

```
                }
            }
        }
    ]
}
```

Jika Anda ingin membatasi akses ke API untuk hanya panggilan yang dilakukan menggunakan titik akhir antarmuka, gunakan kunci `aws:SourceVpce` kondisi alih-alih: `aws:SourceVpc`

```
{
    "Id": "api-example-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableAPIAccess",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedNotebookInstanceUrl"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpce": [
                        "vpce-111bbccc",
                        "vpce-111bbddd"
                    ]
                }
            }
        }
    ]
}
```

Batasi Akses ke SageMaker API dan Panggilan Runtime berdasarkan Alamat IP

Untuk mengizinkan akses ke panggilan SageMaker API dan pemanggilan runtime hanya dari alamat IP dalam daftar yang Anda tentukan, lampirkan kebijakan IAM yang menolak akses ke API kecuali panggilan berasal dari alamat IP dalam daftar ke setiap AWS Identity and Access Management pengguna, grup, atau peran yang digunakan untuk mengakses API atau runtime. Untuk informasi tentang membuat kebijakan IAM, lihat Membuat Kebijakan IAM di AWS Identity and Access ManagementPanduan Pengguna. Untuk menentukan daftar alamat IP yang ingin Anda akses ke panggilan API, gunakan operator `IpAddress` kondisi dan kunci konteks `aws:SourceIP` kondisi.

Untuk informasi tentang operator kondisi IAM, lihat Elemen Kebijakan IAM JSON: Operator Kondisi di Panduan Pengguna. AWS Identity and Access Management Untuk informasi tentang kunci konteks kondisi IAM, lihat Kunci Konteks Kondisi AWS Global.

Misalnya, kebijakan berikut memungkinkan akses ke `CreateTrainingJob`satu-satunya dari alamat IP dalam rentang `192.0.2.0` - `192.0.2.255` dan `203.0.113.0` -`203.0.113.255`:

```
{
    "Version": "2012-10-17",
    "Statement": [

        {
            "Effect": "Allow",
            "Action": "sagemaker:CreateTrainingJob",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            }
        }
    ]
}
```

Batasi Akses ke Instance Notebook berdasarkan Alamat IP

Untuk mengizinkan akses ke instance notebook hanya dari alamat IP dalam daftar yang Anda tentukan, lampirkan kebijakan IAM yang menolak akses `CreatePresignedNotebookInstanceUrl`kecuali panggilan berasal dari alamat IP dalam daftar ke setiap AWS Identity and Access Management pengguna, grup, atau peran yang digunakan untuk mengakses instance notebook. Untuk informasi tentang membuat kebijakan IAM, lihat Membuat Kebijakan IAM di AWS Identity and Access ManagementPanduan Pengguna. Untuk menentukan daftar alamat IP yang ingin Anda akses ke instance notebook, gunakan operator `IpAddress` kondisi dan kunci konteks `aws:SourceIP` kondisi. Untuk informasi tentang operator kondisi IAM, lihat Elemen Kebijakan IAM JSON: Operator Kondisi di Panduan Pengguna. AWS Identity and Access Management Untuk informasi tentang kunci konteks kondisi IAM, lihat Kunci Konteks Kondisi AWS Global.

Misalnya, kebijakan berikut mengizinkan akses ke instance notebook hanya dari alamat IP dalam rentang `192.0.2.0` - `192.0.2.255` dan `203.0.113.0` -`203.0.113.255`:

```
{
    "Version": "2012-10-17",
    "Statement": [

        {
            "Effect": "Allow",
            "Action": "sagemaker:CreatePresignedNotebookInstanceUrl",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            }
        }
    ]
}
```

Kebijakan membatasi akses ke panggilan ke `CreatePresignedNotebookInstanceUrl` dan ke URL yang ditampilkan panggilan. Kebijakan ini juga membatasi akses untuk membuka instance notebook di konsol dan diberlakukan untuk setiap permintaan HTTP dan WebSocket frame yang mencoba terhubung ke instance notebook.

> **Note**
>
> Menggunakan metode ini untuk memfilter berdasarkan alamat IP tidak kompatibel saat menghubungkan ke SageMaker melalui titik akhir antarmuka VPC. . Untuk informasi tentang membatasi akses ke instance notebook saat menghubungkan melalui titik akhir antarmuka VPC, lihat. Connect ke Instance Notebook Melalui Endpoint Antarmuka VPC

Kontrol Akses ke SageMaker Sumber Daya dengan Menggunakan Tag

Tentukan tag dalam kebijakan IAM untuk mengontrol akses ke grup SageMaker sumber daya. Gunakan tag untuk mengimplementasikan kontrol akses berbasis atribut (ABAC). Menggunakan tag membantu Anda mempartisi akses ke sumber daya ke grup pengguna tertentu. Anda dapat

memiliki satu tim dengan akses ke satu kelompok sumber daya dan tim yang berbeda dengan akses ke kumpulan sumber daya lain. Anda dapat memberikan `ResourceTag` ketentuan dalam kebijakan IAM untuk menyediakan akses bagi setiap grup.

> **Note**
>
> Kebijakan berbasis tag tidak berfungsi untuk membatasi panggilan API berikut:
>
> - DeleteImageVersion
> - DescribeImageVersion
> - ListAlgorithms
> - ListCodeRepositories
> - ListCompilationJobs
> - ListEndpointConfigs
> - ListEndpoints
> - ListFlowDefinitions
> - ListHumanTaskUis
> - ListHyperparameterTuningJobs
> - ListLabelingJobs
> - ListLabelingJobsForWorkteam
> - ListModelPackages
> - ListModels
> - ListNotebookInstanceLifecycleConfigs
> - ListNotebookInstances
> - ListSubscribedWorkteams
> - ListTags
> - ListProcessingJobs
> - ListTrainingJobs
> - ListTrainingJobsForHyperParameterTuningJob
> - ListTransformJobs
> - ListWorkteams
> - Pencarian

Contoh sederhana dapat membantu Anda memahami bagaimana Anda dapat menggunakan tag untuk mempartisi sumber daya. Misalkan Anda telah mendefinisikan dua grup IAM yang berbeda, bernama `DevTeam1` dan`DevTeam2`, di AWS akun Anda. Anda telah membuat 10 instance notebook juga. Anda menggunakan 5 instance notebook untuk satu proyek. Anda menggunakan 5 lainnya untuk proyek kedua. Anda dapat `DevTeam1` memberikan izin untuk melakukan panggilan API pada instance notebook yang Anda gunakan untuk proyek pertama. Anda dapat menyediakan `DevTeam2` untuk melakukan panggilan API pada instance notebook yang digunakan untuk proyek kedua.

Prosedur berikut memberikan contoh sederhana yang membantu Anda memahami konsep menambahkan tag. Anda dapat menggunakannya untuk mengimplementasikan solusi yang dijelaskan dalam paragraf sebelumnya.

Untuk mengontrol akses ke panggilan API (contoh)

1. Tambahkan tag dengan kunci `Project` dan nilai `A` ke instance notebook yang digunakan untuk proyek pertama. Untuk informasi tentang menambahkan tag ke SageMaker sumber daya, lihat `AddTags`.
2. Tambahkan tag dengan kunci `Project` dan nilai `B` ke instance notebook yang digunakan untuk proyek kedua.
3. Buat kebijakan IAM dengan `ResourceTag` kondisi yang menolak akses ke instance buku catatan yang digunakan untuk proyek kedua, dan lampirkan kebijakan tersebut. `DevTeam1` Berikut ini adalah contoh kebijakan yang menolak semua panggilan API pada instance notebook mana pun yang memiliki tag dengan kunci `Project` dan nilai`B`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sagemaker:*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "sagemaker:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "sagemaker:ResourceTag/Project": "B"
        }
```

```
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "sagemaker:AddTags",
        "sagemaker:DeleteTags"
      ],
      "Resource": "*"
    }
  ]
}
```

Untuk informasi tentang membuat kebijakan IAM dan melampirkannya ke identitas, lihat Mengontrol Akses Menggunakan Kebijakan di Panduan Pengguna. AWS Identity and Access Management

4. Buat kebijakan IAM dengan `ResourceTag` kondisi yang menolak akses ke instance buku catatan yang digunakan untuk proyek pertama, dan lampirkan kebijakan tersebut. `DevTeam2` Berikut ini adalah contoh kebijakan yang menolak semua panggilan API pada instance notebook mana pun yang memiliki tag dengan kunci `Project` dan nilai`A`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sagemaker:*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "sagemaker:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "sagemaker:ResourceTag/Project": "A"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
```

```
        "sagemaker:AddTags",
        "sagemaker:DeleteTags"
      ],
      "Resource": "*"
    }
  ]
}
```

Berikan Izin untuk Menandai SageMaker Sumber Daya Saat Pembuatan

Ada banyak SageMaker operasi Amazon yang memungkinkan pengguna Anda menentukan tag saat Anda membuat sumber daya. Gunakan tag sumber daya untuk menerapkan kontrol akses berbasis atribut (ABAC). Menggunakan tag membantu Anda mempartisi akses ke sumber daya ke grup pengguna tertentu. Anda dapat memiliki satu tim dengan akses ke satu kelompok sumber daya dan tim yang berbeda dengan akses ke kumpulan sumber daya lain. Anda dapat memberikan `ResourceTag` ketentuan dalam kebijakan IAM untuk menyediakan akses bagi setiap grup.

Anda harus memberi pengguna Anda akses ke izin yang mereka gunakan untuk membuat sumber daya. Untuk pengguna yang perlu membuat pekerjaan pemrosesan, Anda harus memberi mereka akses ke `sagemaker:CreateProcessingJob` dalam kebijakan. Jika pengguna Anda menandai sumber daya yang mereka buat, Anda juga harus memberikan izin penandaan.

Important

Anda hanya dapat memberikan izin kepada pengguna untuk menambahkan tag ke sumber daya yang mereka buat. Mereka tidak dapat menambahkan tag ke sumber daya yang telah dibuat.
Berikut ini adalah operasi di mana Anda tidak dapat memberi mereka sumber daya untuk menambahkan tag:

- DeleteImageVersion
- DescribeImageVersion
- ListAlgorithms
- ListCodeRepositories
- ListCompilationJobs
- ListEndpointConfigs
- ListEndpoints

- ListFlowDefinitions
- ListHumanTaskUis
- ListHyperparameterTuningJobs
- ListLabelingJobs
- ListLabelingJobsForWorkteam
- ListModelPackages
- ListModels
- ListNotebookInstanceLifecycleConfigs
- ListNotebookInstances
- ListSubscribedWorkteams
- ListTags
- ListProcessingJobs
- ListTrainingJobs
- ListTrainingJobsForHyperParameterTuningJob
- ListTransformJobs
- ListWorkteams
- Pencarian

Anda dapat mengontrol akses ke operasi yang dapat ditandai pengguna atau tag yang dapat mereka gunakan. Anda dapat menentukan izin untuk kasus penggunaan berikut ini:

- Menambahkan izin untuk menandai sumber daya yang dibuat menggunakan operasi dan izin apa pun untuk menggunakan tag apa pun
- Menambahkan izin untuk menandai sumber daya yang dibuat menggunakan operasi dan izin tertentu untuk menggunakan tag apa pun
- Menambahkan izin untuk menandai sumber daya yang dibuat menggunakan operasi dan izin apa pun ke tag tertentu
- Menambahkan izin untuk menandai sumber daya yang dibuat menggunakan operasi dan izin tertentu ke tag tertentu

Permissions for any operation and any tag

Tambahkan pernyataan berikut ke kebijakan IAM untuk memberikan izin kepada pengguna Anda untuk menambahkan tag apa pun untuk operasi apa pun.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:Create*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:AddTags"
            ],
            "Resource": "arn:aws:sagemaker:region:account:*/*",
            "Condition": {
                "Null": {
                    "sagemaker:TaggingAction" : "false"
                }
            }
        }
    ]
}
```

Permissions for a specific operation with specific tags

Pernyataan berikut memberikan izin pengguna untuk menambahkan `cc123` tag ke sumber daya yang dibuat menggunakan `CreateModel` operasi. Anda dapat memodifikasi pernyataan agar sesuai dengan kebutuhan Anda sendiri dan menambahkannya ke kebijakan IAM yang dilampirkan pada peran pengguna Anda.

```
{
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateModel"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/cost-center": [ "cc123" ]
                },
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": [ "purpose" ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:AddTags"
            ],
            "Resource": "arn:aws:sagemaker:region:account:*/*",
            "Condition": {
                "Null": {
                    "sagemaker:TaggingAction" : "false"
                }
            }
        }
    ]
}
```

Permissions for a specific operation and any tag

Pernyataan berikut memberikan izin kepada pengguna Anda untuk menambahkan tag apa pun ke sumber daya yang dibuat menggunakan `CreateModel` operasi.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateModel"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:AddTags"
            ],
            "Resource": "arn:aws:sagemaker:region:account:model/*",
            "Condition": {
                "String": {
                    "sagemaker:TaggingAction" : [ "CreateModel" ]
                }
            }
        }
    ]
}
```

Pencegahan Deputi Bingung Lintas Layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahunAWS, masalah wakil yang bingung dapat muncul karena peniruan identitas lintas layanan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang disebut) dan memanfaatkan izin tinggi layanan yang disebut untuk bertindak atas sumber daya yang tidak memiliki otorisasi untuk diakses oleh layanan panggilan. Untuk mencegah akses tidak sah melalui masalah wakil yang membingungkan, AWS sediakan alat untuk membantu mengamankan data Anda di seluruh layanan. Alat-alat ini membantu Anda mengontrol izin yang diberikan kepada prinsipal layanan, membatasi akses mereka hanya ke sumber daya di akun Anda yang diperlukan. Dengan mengelola hak akses kepala layanan secara hati-hati, Anda dapat membantu mengurangi risiko layanan mengakses data atau sumber daya secara tidak benar yang seharusnya tidak memiliki izin.

Baca terus untuk panduan umum atau arahkan ke contoh untuk SageMaker fitur tertentu:

Topik

- Batasi Izin Dengan Kunci Kondisi Global
- SageMaker Manajer Edge
- SageMaker Gambar

- SageMaker Inferensi
- SageMaker Lowongan Batch Transform
- SageMaker Marketplace
- SageMaker Neo
- SageMaker Pipa
- SageMaker Lowongan Pengolahan
- SageMaker Studio
- SageMaker Lowongan Pelatihan

Batasi Izin Dengan Kunci Kondisi Global

Sebaiknya gunakan kunci kondisi `aws:SourceAccount` global `aws:SourceArn` dan global dalam kebijakan sumber daya untuk membatasi izin ke sumber daya yang SageMaker diberikan Amazon kepada layanan lain. Jika Anda menggunakan kunci kondisi global dan `aws:SourceArn` nilainya berisi ID akun, `aws:SourceAccount` nilai dan akun dalam `aws:SourceArn` nilai harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci kondisi `aws:SourceArn` global dengan ARN penuh sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:sagemaker:*:123456789012:*`.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi `aws:SourceArn` dan `aws:SourceAccount` global SageMaker untuk mencegah masalah wakil yang membingungkan.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "sagemaker.amazonaws.com"
```

```
    },
    # Specify an action and resource policy for another service
    "Action": "service:ActionName",
    "Resource": [
      "arn:aws:service:::ResourceName/*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:partition:sagemaker:region:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

SageMaker Manajer Edge

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi aws:SourceArn global untuk mencegah masalah wakil kebingungan lintas layanan untuk SageMaker Edge Manager yang dibuat dengan nomor akun 123456789012 di Wilayah us-barat-2.

```
{
  "Version": "2012-10-17",
  "Statement": {
      "Effect": "Allow",
      "Principal": { "Service": "sagemaker.amazonaws.com" },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:*"
      }
    }
  }
}
```

Anda dapat mengganti template ini dengan ARN lengkap dari satu pekerjaan pengemasan tertentu untuk membatasi izin lebih lanjut. `aws:SourceArn`

SageMaker Gambar

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi `aws:SourceArn` global untuk mencegah masalah wakil lintas layanan yang membingungkan untuk SageMaker Gambar. Gunakan template ini dengan salah satu `Image` atau`ImageVersion`. *Contoh ini menggunakan ImageVersion catatan ARN dengan nomor akun 123456789012.* Perhatikan bahwa karena nomor akun adalah bagian dari `aws:SourceArn` nilai, Anda tidak perlu menentukan `aws:SourceAccount` nilai.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "sagemaker.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:partition:sagemaker:us-west-2:123456789012:image-version"
      }
    }
  }
}
```

Jangan mengganti `aws:SourceArn` dalam template ini dengan ARN lengkap dari gambar atau versi gambar tertentu. ARN harus dalam format yang disediakan di atas dan tentukan salah satu atau`image`. `image-version` `partition`Placeholder harus menunjuk partisi AWS komersial (`aws`) atau partisi AWS di China (`aws-cn`), tergantung di mana gambar atau versi gambar berjalan. Demikian pula, `region` placeholder di ARN dapat berupa Wilayah yang valid di mana SageMaker gambar tersedia.

SageMaker Inferensi

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi `aws:SourceArn` global untuk mencegah masalah wakil kebingungan lintas layanan untuk inferensi SageMaker real-time, tanpa server, dan asinkron. Perhatikan bahwa karena nomor akun adalah bagian dari `aws:SourceArn` nilai, Anda tidak perlu menentukan `aws:SourceAccount` nilai.

```
{
  "Version": "2012-10-17",
  "Statement": {
```

```
    "Effect": "Allow",
    "Principal": { "Service": "sagemaker.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:*"
      }
    }
  }
}
```

Jangan mengganti template ini dengan ARN lengkap dari model atau titik akhir tertentu. `aws:SourceArn` ARN harus berada dalam format yang tersedia di atas. Tanda bintang di template ARN tidak berarti wildcard dan tidak boleh diubah.

SageMaker Lowongan Batch Transform

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi aws:SourceArn global untuk mencegah masalah deputi kebingungan lintas layanan untuk pekerjaan transformasi SageMaker batch yang dibuat dengan nomor akun 123456789012 di Wilayah us-barat-2. Perhatikan bahwa karena nomor akun ada di ARN, Anda tidak perlu menentukan nilai. `aws:SourceAccount`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:transform-job/*"
        }
      }
    }
  ]
}
```

Anda dapat mengganti `aws:SourceArn` dalam template ini dengan ARN penuh dari satu pekerjaan transformasi batch tertentu untuk membatasi izin lebih lanjut.

SageMaker Marketplace

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi aws:SourceArn global untuk mencegah masalah deputi kebingungan lintas layanan untuk sumber daya SageMaker Marketplace yang dibuat dengan nomor akun 123456789012 di Wilayah us-barat-2. Perhatikan bahwa karena nomor akun ada di ARN, Anda tidak perlu menentukan nilai. `aws:SourceAccount`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:*"
        }
      }
    }
  ]
}
```

Jangan mengganti `aws:SourceArn` dalam template ini dengan ARN lengkap dari algoritma atau paket model tertentu. ARN harus berada dalam format yang tersedia di atas. Tanda bintang dalam template ARN memang merupakan singkatan dari wildcard dan mencakup semua pekerjaan pelatihan, model, dan pekerjaan transformasi batch dari langkah validasi, serta paket algoritma dan model yang diterbitkan ke Marketplace. SageMaker

SageMaker Neo

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi aws:SourceArn global untuk mencegah masalah wakil kebingungan lintas layanan untuk pekerjaan kompilasi SageMaker Neo yang dibuat dengan nomor

akun 123456789012 di Wilayah us-barat-2. Perhatikan bahwa karena nomor akun ada di ARN, Anda tidak perlu menentukan nilai. `aws:SourceAccount`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:compilation-job/*"
        }
      }
    }
  ]
}
```

Anda dapat mengganti `aws:SourceArn` dalam template ini dengan ARN lengkap dari satu pekerjaan kompilasi tertentu untuk membatasi izin lebih lanjut.

SageMaker Pipa

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi `aws:SourceArn` global untuk mencegah masalah wakil lintas layanan yang membingungkan untuk SageMaker Pipelines menggunakan catatan eksekusi pipa dari satu atau lebih saluran pipa. Perhatikan bahwa karena nomor akun ada di ARN, Anda tidak perlu menentukan nilai. `aws:SourceAccount`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
```

```
          "aws:SourceArn": "arn:partition:sagemaker:region:123456789012:pipeline/
mypipeline/*"
        }
      }
    }
  ]
}
```

Jangan mengganti `aws:SourceArn` dalam template ini dengan ARN penuh dari eksekusi pipeline tertentu. ARN harus berada dalam format yang tersedia di atas. `partition`Placeholder harus menunjuk partisi AWS komersial (`aws`) atau partisi AWS di China (`aws-cn`), tergantung di mana pipa berjalan. Demikian pula, `region` placeholder di ARN dapat berupa Wilayah yang valid di mana SageMaker Pipelines tersedia.

Tanda bintang di template ARN memang merupakan singkatan dari wildcard dan mencakup semua eksekusi pipeline dari pipeline bernama. `mypipeline` Jika Anda ingin mengizinkan `AssumeRole` izin untuk semua saluran pipa di akun `123456789012` daripada satu pipa tertentu, maka itu `aws:SourceArn` akan menjadi. `arn:aws:sagemaker:*:123456789012:pipeline/*`

SageMaker Lowongan Pengolahan

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi aws:SourceArn global untuk mencegah masalah wakil kebingungan lintas layanan untuk SageMaker memproses pekerjaan yang dibuat dengan nomor akun 123456789012 di Wilayah us-barat-2. Perhatikan bahwa karena nomor akun ada di ARN, Anda tidak perlu menentukan nilai. `aws:SourceAccount`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:processing-job/*"
        }
      }
```

```
    }
  ]
}
```

Anda dapat mengganti `aws:SourceArn` dalam template ini dengan ARN penuh dari satu pekerjaan pemrosesan tertentu untuk membatasi izin lebih lanjut.

SageMaker Studio

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi aws:SourceArn global untuk mencegah masalah wakil kebingungan lintas layanan untuk SageMaker Studio yang dibuat dengan nomor akun 123456789012 di Wilayah us-barat-2. Perhatikan bahwa karena nomor akun adalah bagian dari `aws:SourceArn` nilai, Anda tidak perlu menentukan `aws:SourceAccount` nilai.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:*"
        }
      }
    }
  ]
}
```

Jangan mengganti `aws:SourceArn` dalam template ini dengan ARN lengkap dari aplikasi Studio tertentu, profil pengguna, atau domain. ARN harus dalam format yang disediakan pada contoh sebelumnya. Tanda bintang di template ARN tidak berarti wildcard dan tidak boleh diubah.

SageMaker Lowongan Pelatihan

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci kondisi aws:SourceArn global untuk mencegah masalah wakil kebingungan lintas

layanan untuk pekerjaan SageMaker pelatihan yang dibuat dengan nomor akun 123456789012 di Wilayah us-barat-2. Perhatikan bahwa karena nomor akun ada di ARN, Anda tidak perlu menentukan nilai. `aws:SourceAccount`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sagemaker:us-west-2:123456789012:training-job/*"
        }
      }
    }
  ]
}
```

Anda dapat mengganti `aws:SourceArn` dalam template ini dengan ARN lengkap dari satu pekerjaan pelatihan tertentu untuk membatasi izin lebih lanjut.

Selanjutnya

Untuk informasi selengkapnya tentang mengelola peran eksekusi, lihat SageMaker Peran.

SageMaker Peran

Amazon SageMaker melakukan operasi atas nama Anda menggunakan AWS layanan lain. Anda harus memberikan SageMaker izin untuk menggunakan layanan ini dan sumber daya yang ditindaklanjutinya. Anda memberikan izin SageMaker ini menggunakan peran eksekusi AWS Identity and Access Management (IAM). Untuk informasi selengkapnya tentang peran IAM, lihat peran IAM.

Untuk membuat dan menggunakan peran eksekusi, Anda dapat menggunakan prosedur berikut.

Buat peran eksekusi

Gunakan prosedur berikut untuk membuat peran eksekusi dengan kebijakan terkelola IAM`AmazonSageMakerFullAccess`, terlampir. Jika kasus penggunaan Anda memerlukan izin yang

lebih terperinci, gunakan bagian lain di halaman ini untuk membuat peran eksekusi yang memenuhi kebutuhan bisnis Anda. Anda dapat membuat peran eksekusi menggunakan SageMaker konsol atau fileAWS CLI.

Important

Kebijakan terkelola IAM`AmazonSageMakerFullAccess`, yang digunakan dalam prosedur berikut hanya memberikan izin peran eksekusi untuk melakukan tindakan Amazon S3 tertentu pada bucket atau objek `SageMaker` dengan`Sagemaker`,,`sagemaker`, `aws-glue` atau dalam nama. Untuk mempelajari cara menambahkan kebijakan tambahan ke peran eksekusi agar akses ke bucket dan objek Amazon S3 lainnya, lihat. Menambahkan Izin Amazon S3 Tambahan ke Peran Eksekusi SageMaker

Note

Anda dapat membuat peran eksekusi secara langsung saat membuat SageMaker domain atau instance notebook.

- Untuk informasi tentang cara membuat SageMaker domain, lihatOrientasi khusus ke SageMaker Domain Amazon menggunakan IAM Identity Center.
- Untuk informasi tentang cara membuat sebuah instans notebook, lihatLangkah 1: Buat Instans SageMaker Notebook Amazon.

Untuk membuat peran eksekusi baru dari SageMaker konsol

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pilih Peran dan kemudian pilih Buat peran.
3. Simpan AWSlayanan sebagai jenis entitas Tepercaya, lalu gunakan panah bawah untuk menemukan SageMakerkasus Penggunaan untuk AWS layanan lain.
4. Pilih SageMaker — Eksekusi dan kemudian pilih Berikutnya.
5. Kebijakan yang dikelola IAM,`AmazonSageMakerFullAccess`, secara otomatis dilampirkan ke peran. Untuk melihat izin yang disertakan dalam kebijakan ini, pilih tanda tambah (+) di samping nama kebijakan. Pilih Berikutnya.
6. Masukkan nama Peran dan Deskripsi.

7. (Opsional) Tambahkan izin dan tag tambahan ke peran.
8. Pilih Buat peran.
9. Di bagian Peran konsol IAM, temukan peran yang baru saja Anda buat. Jika diperlukan, gunakan kotak teks untuk mencari peran menggunakan nama peran.
10. Pada halaman ringkasan peran, catat ARN.

Untuk membuat peran eksekusi baru dari AWS CLI

Sebelum Anda membuat peran eksekusi menggunakanAWS CLI, pastikan untuk memperbarui dan mengonfigurasinya dengan mengikuti instruksi diAWS CLIPrasyarat, lalu lanjutkan dengan instruksi diOnboard dari AWS CLI.

Setelah Anda membuat peran eksekusi, Anda dapat mengaitkannya dengan SageMaker domain, profil pengguna, atau dengan instance notebook Jupyter.

- Untuk mempelajari cara mengaitkan peran eksekusi dengan SageMaker domain yang ada, lihatEdit pengaturan Domain.
- Untuk mempelajari cara mengaitkan peran eksekusi dengan profil pengguna yang ada, lihatTambah dan Hapus Profil Pengguna.
- Untuk mempelajari cara mengaitkan peran eksekusi dengan instance notebook yang ada, lihatMemperbarui Instance Notebook.

Anda juga dapat meneruskan ARN dari peran eksekusi ke panggilan API Anda. Misalnya, menggunakan Amazon SageMaker Python SDK, Anda dapat meneruskan ARN peran eksekusi Anda ke estimator. Dalam contoh kode berikut, kami membuat estimator menggunakan wadah algoritma XGBoost dan meneruskan ARN dari peran eksekusi sebagai parameter. Untuk contoh selengkapnya GitHub, lihat Prediksi Churn Pelanggan dengan XGBoost.

```
import sagemaker, boto3
from sagemaker import image_uris

sess = sagemaker.Session()
region = sess.boto_region_name
bucket = sess.default_bucket()
prefix = "sagemaker/DEMO-xgboost-churn"
container = sagemaker.image_uris.retrieve("xgboost", region, "1.7-1")
```

```
xgb = sagemaker.estimator.Estimator(
    container,
    execution-role-ARN,
    instance_count=1,
    instance_type="ml.m4.xlarge",
    output_path="s3://{}/{}/output".format(bucket, prefix),
    sagemaker_session=sess,
)

...
```

Menambahkan Izin Amazon S3 Tambahan ke Peran Eksekusi SageMaker

Saat Anda menggunakan SageMaker fitur dengan sumber daya di Amazon S3, seperti data input, peran eksekusi yang Anda tentukan dalam permintaan Anda (misalnya`CreateTrainingJob`) digunakan untuk mengakses sumber daya ini.

Jika Anda melampirkan kebijakan terkelola IAM`AmazonSageMakerFullAccess`, ke peran eksekusi, peran tersebut memiliki izin untuk melakukan tindakan Amazon S3 tertentu pada bucket atau objek `SageMaker` dengan`Sagemaker`,,`sagemaker`, `aws-glue` atau dalam nama. Ini juga memiliki izin untuk melakukan tindakan berikut pada sumber daya Amazon S3:

```
"s3:CreateBucket",
"s3:GetBucketLocation",
"s3:ListBucket",
"s3:ListAllMyBuckets",
"s3:GetBucketCors",
"s3:PutBucketCors"
```

Untuk memberikan izin peran eksekusi untuk mengakses satu atau beberapa bucket tertentu di Amazon S3, Anda dapat melampirkan kebijakan yang mirip dengan peran berikut ini. *Kebijakan ini memberikan izin peran IAM untuk melakukan semua tindakan yang AmazonSageMakerFullAccess memungkinkan tetapi membatasi akses ini ke bucket DOC-EXAMPLE-BUCKET1 dan DOC-EXAMPLE-BUCKET2.* Lihat dokumentasi keamanan untuk SageMaker fitur spesifik yang Anda gunakan untuk mempelajari lebih lanjut tentang izin Amazon S3 yang diperlukan untuk fitur tersebut.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:AbortMultipartUpload"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:GetBucketCors",
                "s3:PutBucketCors"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketAcl",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET2"
            ]
        }
    ]
}
```

Dapatkan peran eksekusi

Anda dapat menggunakan SageMaker konsol atau AWS CLI untuk mengambil ARN dari peran eksekusi yang dilampirkan ke SageMaker domain, profil pengguna, atau instance notebook.

- Untuk menemukan ARN dari peran eksekusi IAM yang dilampirkan ke SageMaker domain, lihat. Lihat dan Edit Domain
- Untuk menemukan ARN dari peran eksekusi IAM yang dilampirkan ke profil pengguna, lihat. Lihat Profil Pengguna dan Detail Profil Pengguna
- Untuk menemukan ARN dari peran eksekusi IAM yang dilampirkan ke instance notebook:

  1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
  2. Di panel navigasi kiri, pilih Notebook lalu instance Notebook.
  3. Dari daftar notebook, pilih notebook yang ingin Anda lihat.
  4. ARN ada di bagian Izin dan enkripsi.

Atau, pengguna Amazon SageMaker Python SDK dapat mengambil ARN dari peran eksekusi yang dilampirkan ke profil pengguna mereka atau instance notebook dengan menjalankan kode berikut:

```
import sagemaker
sagemaker_session = sagemaker.Session()
role = sagemaker.get_execution_role()
```

> **Note**
>
> Peran eksekusi hanya tersedia saat menjalankan notebook di dalamnya SageMaker. Jika Anda menjalankan `get_execution_role` di buku catatan tidak SageMaker aktif, harapkan kesalahan “wilayah”.

Peran Lulus

Tindakan seperti melewati peran antar layanan adalah fungsi umum di dalamnya SageMaker. Anda dapat menemukan detail selengkapnya tentang Tindakan, Sumber Daya, dan Kunci Kondisi SageMaker di Panduan Pengguna IAM.

Anda meneruskan role (`iam:PassRole`) saat melakukan panggilan API ini: `CreateAutoMLJob`, `CreateCompilationJob`, `CreateDomain`, `CreateFeatureGroup`, `CreateFlowDefiniton`, `CreateHyperParameterTuningJob`, `CreateImage`, `CreateLabelingJob`, `CreateModelCreateMonitoringSchedule`, `CreateNotebookInstance`, `CreateProcessingJob`, `CreateTrainingJob`, `CreateUserProfile`, `RenderUiTemplate`, dan `UpdateImage`.

Anda melampirkan kebijakan kepercayaan berikut ke peran IAM yang memberikan izin SageMaker utama untuk mengambil peran, dan sama untuk semua peran eksekusi:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Izin yang perlu Anda berikan untuk peran bervariasi tergantung pada API yang Anda panggil. Bagian berikut menjelaskan izin ini.

> **Note**
>
> Alih-alih mengelola izin dengan membuat kebijakan izin, Anda dapat menggunakan kebijakan izin yang AWS dikelola`AmazonSageMakerFullAccess`. Izin dalam kebijakan ini cukup luas, untuk memungkinkan tindakan apa pun yang mungkin ingin Anda lakukan. SageMaker Untuk daftar kebijakan termasuk informasi tentang alasan menambahkan banyak izin, lihatAWSkebijakan terkelola: AmazonSageMakerFullAccess. Jika Anda memilih untuk membuat kebijakan kustom dan mengelola izin untuk cakupan izin hanya untuk tindakan yang perlu Anda lakukan dengan peran eksekusi, lihat topik berikut.

> **Important**
>
> Jika Anda mengalami masalah, lihatPemecahan masalah SageMaker identitas dan akses Amazon.

Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM dalam Panduan Pengguna IAM.

Topik

- CreateAutoMLJob API: Izin Peran Eksekusi
- CreateDomain API: Izin Peran Eksekusi
- CreateImage dan UpdateImage API: Izin Peran Eksekusi
- CreateNotebookInstance API: Izin Peran Eksekusi
- CreateHyperParameterTuningJob API: Izin Peran Eksekusi
- CreateProcessingJob API: Izin Peran Eksekusi
- CreateTrainingJob API: Izin Peran Eksekusi
- CreateModel API: Izin Peran Eksekusi
- SageMaker peran kemampuan geospasial

CreateAutoMLJob API: Izin Peran Eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `CreateAutoMLJob` API, Anda dapat melampirkan kebijakan izin minimum berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeEndpointConfig",
                "sagemaker:DescribeModel",
                "sagemaker:InvokeEndpoint",
                "sagemaker:ListTags",
                "sagemaker:DescribeEndpoint",
```

```
                "sagemaker:CreateModel",
                "sagemaker:CreateEndpointConfig",
                "sagemaker:CreateEndpoint",
                "sagemaker:DeleteModel",
                "sagemaker:DeleteEndpointConfig",
                "sagemaker:DeleteEndpoint",
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "*"
        }
    ]
}
```

Jika Anda menentukan VPC pribadi untuk pekerjaan AutoML Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ]
}
```

Jika input Anda dienkripsi menggunakan enkripsi sisi server dengan kunci yang AWS dikelola KMS (SSE-KMS), tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ]
}
```

Jika Anda menentukan kunci KMS dalam konfigurasi output pekerjaan AutoML Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:Encrypt"
    ]
}
```

Jika Anda menentukan kunci volume KMS dalam konfigurasi sumber daya pekerjaan AutoML Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:CreateGrant"
    ]
}
```

CreateDomain API: Izin Peran Eksekusi

Peran eksekusi untuk domain dengan IAM Identity Center dan peran pengguna/eksekusi untuk domain IAM memerlukan izin berikut saat Anda meneruskan kunci terkelola AWS KMS pelanggan seperti dalam permintaan API. `KmsKeyId` `CreateDomain` Izin diberlakukan selama panggilan `CreateApp` API.

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `CreateDomain` API, Anda dapat melampirkan kebijakan izin berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:region:account-id:key/kms-key-id"
        }
    ]
}
```

Sebagai alternatif, jika izin ditentukan dalam kebijakan KMS, Anda dapat melampirkan kebijakan berikut ke peran tersebut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-id:role/ExecutionRole"
                ]
            },
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

CreateImage dan UpdateImage API: Izin Peran Eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `CreateImage` atau `UpdateImage` API, Anda dapat melampirkan kebijakan izin berikut ke peran tersebut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        }
    ]
}
```

CreateNotebookInstance API: Izin Peran Eksekusi

Izin yang Anda berikan untuk peran eksekusi untuk memanggil `CreateNotebookInstance` API bergantung pada apa yang Anda rencanakan untuk dilakukan dengan instance notebook. Jika Anda berencana menggunakannya untuk memanggil SageMaker API dan meneruskan peran yang sama saat memanggil `CreateTrainingJob` dan `CreateModel` API, lampirkan kebijakan izin berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:*",
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "ecr:SetRepositoryPolicy",
                "ecr:CompleteLayerUpload",
```

```
                "ecr:BatchDeleteImage",
                "ecr:UploadLayerPart",
                "ecr:DeleteRepositoryPolicy",
                "ecr:InitiateLayerUpload",
                "ecr:DeleteRepository",
                "ecr:PutImage",
                "ecr:CreateRepository",
                "cloudwatch:PutMetricData",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents",
                "logs:GetLogEvents",
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "robomaker:CreateSimulationApplication",
                "robomaker:DescribeSimulationApplication",
                "robomaker:DeleteSimulationApplication",
                "robomaker:CreateSimulationJob",
                "robomaker:DescribeSimulationJob",
                "robomaker:CancelSimulationJob",
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeRouteTables",
                "elasticfilesystem:DescribeMountTargets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "codecommit:GitPull",
                "codecommit:GitPush"
            ],
            "Resource": [
                "arn:aws:codecommit:*:*:*sagemaker*",
                "arn:aws:codecommit:*:*:*SageMaker*",
                "arn:aws:codecommit:*:*:*Sagemaker*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        }
    ]
}
```

Untuk memperketat izin, batasi ke sumber daya Amazon S3 dan Amazon ECR tertentu, dengan `"Resource": "*"` membatasi, sebagai berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:*",
                "ecr:GetAuthorizationToken",
                "cloudwatch:PutMetricData",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket/object1",
                "arn:aws:s3:::outputbucket/path",
                "arn:aws:s3:::inputbucket/object2",
                "arn:aws:s3:::inputbucket/object3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": [
                "arn:aws:ecr:region::repository/my-repo1",
                "arn:aws:ecr:region::repository/my-repo2",
                "arn:aws:ecr:region::repository/my-repo3"
            ]
        }
    ]
```

```
}
```

Jika Anda berencana untuk mengakses sumber daya lain, seperti Amazon DynamoDB atau Amazon Relational Database Service, tambahkan izin yang relevan ke kebijakan ini.

Dalam kebijakan sebelumnya, Anda mencakup kebijakan sebagai berikut:

- Cakupan `s3:ListBucket` izin ke bucket tertentu yang Anda tentukan seperti `InputDataConfig.DataSource.S3DataSource.S3Uri` dalam `CreateTrainingJob` permintaan.
- Lingkup `s3:GetObject s3:PutObject`,, dan `s3:DeleteObject` izin sebagai berikut:
  - Cakupan ke nilai berikut yang Anda tentukan dalam `CreateTrainingJob` permintaan:

    `InputDataConfig.DataSource.S3DataSource.S3Uri`

    `OutputDataConfig.S3OutputPath`
  - Cakupan ke nilai berikut yang Anda tentukan dalam `CreateModel` permintaan:

    `PrimaryContainer.ModelDataUrl`

    `SuplementalContainers.ModelDataUrl`
- `ecr`Izin lingkup sebagai berikut:
  - Cakupan ke `AlgorithmSpecification.TrainingImage` nilai yang Anda tentukan dalam `CreateTrainingJob` permintaan.
  - Cakupan ke `PrimaryContainer.Image` nilai yang Anda tentukan dalam `CreateModel` permintaan:

Tindakan `cloudwatch` dan `logs` tindakan berlaku untuk sumber daya “*”. Untuk informasi selengkapnya, lihat CloudWatch Sumber Daya dan Operasi di Panduan CloudWatch Pengguna Amazon.

CreateHyperParameterTuningJob API: Izin Peran Eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `CreateHyperParameterTuningJob` API, Anda dapat melampirkan kebijakan izin berikut ke peran:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "*"
        }
    ]
}
```

Alih-alih menentukan`"Resource": "*"`, Anda dapat mencakup izin ini ke sumber daya Amazon S3, Amazon ECR, CloudWatch dan Amazon Logs tertentu:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::inputbucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket/object",
                "arn:aws:s3:::outputbucket/path"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "arn:aws:ecr:region::repository/my-repo"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/sagemaker/TrainingJobs*"
        }
    ]
}
```

Jika container pelatihan yang terkait dengan tugas penyetelan hyperparameter perlu mengakses sumber data lain, seperti DynamoDB atau sumber daya Amazon RDS, tambahkan izin yang relevan ke kebijakan ini.

Dalam kebijakan sebelumnya, Anda mencakup kebijakan sebagai berikut:

- `s3:ListBucket`Cakupkan izin ke bucket tertentu yang Anda tentukan `InputDataConfig.DataSource.S3DataSource.S3Uri` sebagai `CreateTrainingJob` permintaan.
- Cakupan `s3:GetObject` dan `s3:PutObject` izin ke objek berikut yang Anda tentukan dalam konfigurasi data input dan output dalam `CreateHyperParameterTuningJob` permintaan:

  `InputDataConfig.DataSource.S3DataSource.S3Uri`

  `OutputDataConfig.S3OutputPath`
- Cakupan izin Amazon ECR ke jalur registri (`AlgorithmSpecification.TrainingImage`) yang Anda tentukan dalam permintaan. `CreateHyperParameterTuningJob`
- Cakupan izin Amazon CloudWatch Logs untuk log grup pekerjaan SageMaker pelatihan.

`cloudwatch`Tindakan tersebut berlaku untuk sumber daya “*”. Untuk informasi selengkapnya, lihat CloudWatch Sumber Daya dan Operasi di Panduan CloudWatch Pengguna Amazon.

Jika Anda menentukan VPC pribadi untuk pekerjaan penyetelan hyperparameter Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ]
}
```

Jika input Anda dienkripsi menggunakan enkripsi sisi server dengan kunci yang AWS dikelola KMS (SSE-KMS), tambahkan izin berikut:

```
{
    "Effect": "Allow",
```

```
    "Action": [
        "kms:Decrypt"
    ]
}
```

Jika Anda menentukan kunci KMS dalam konfigurasi output pekerjaan penyetelan hyperparameter Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:Encrypt"
    ]
}
```

Jika Anda menentukan kunci volume KMS dalam konfigurasi sumber daya tugas penyetelan hyperparameter Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:CreateGrant"
    ]
}
```

CreateProcessingJob API: Izin Peran Eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `CreateProcessingJob` API, Anda dapat melampirkan kebijakan izin berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
```

```
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "*"
        }
    ]
}
```

Alih-alih menentukan`"Resource": "*"`, Anda dapat mencakup izin ini ke sumber daya Amazon S3 dan Amazon ECR tertentu:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket/object",
                "arn:aws:s3:::outputbucket/path"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "arn:aws:ecr:region::repository/my-repo"
        }
    ]
}
```

Jika `CreateProcessingJob.AppSpecification.ImageUri` perlu mengakses sumber data lain, seperti DynamoDB atau sumber daya Amazon RDS, tambahkan izin yang relevan ke kebijakan ini.

Dalam kebijakan sebelumnya, Anda mencakup kebijakan sebagai berikut:

- `s3:ListBucket`Cakupkan izin ke bucket tertentu yang Anda tentukan `ProcessingInputs` sebagai `CreateProcessingJob` permintaan.
- Cakupan `s3:GetObject` dan `s3:PutObject` izin untuk objek yang akan diunduh atau diunggah dalam `ProcessingInputs` dan `ProcessingOutputConfig` dalam permintaan. `CreateProcessingJob`
- Cakupan izin Amazon ECR ke jalur registri (`AppSpecification.ImageUri`) yang Anda tentukan dalam permintaan. `CreateProcessingJob`

Tindakan `cloudwatch` dan `logs` tindakan berlaku untuk sumber daya “*”. Untuk informasi selengkapnya, lihat CloudWatch Sumber Daya dan Operasi di Panduan CloudWatch Pengguna Amazon.

Jika Anda menentukan VPC pribadi untuk pekerjaan pemrosesan Anda, tambahkan izin berikut. Jangan cakup dalam kebijakan dengan kondisi atau filter sumber daya apa pun. Jika tidak, pemeriksaan validasi yang terjadi selama pembuatan pekerjaan pemrosesan gagal.

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ]
}
```

Jika input Anda dienkripsi menggunakan enkripsi sisi server dengan kunci yang AWS dikelola KMS (SSE-KMS), tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ]
}
```

Jika Anda menentukan kunci KMS dalam konfigurasi output pekerjaan pemrosesan Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:Encrypt"
    ]
}
```

Jika Anda menentukan kunci volume KMS dalam konfigurasi sumber daya pekerjaan pemrosesan Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
```

```
    "kms:CreateGrant"
    ]
}
```

CreateTrainingJob API: Izin Peran Eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `CreateTrainingJob` API, Anda dapat melampirkan kebijakan izin berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "*"
        }
    ]
}
```

Alih-alih menentukan`"Resource": "*"`, Anda dapat mencakup izin ini ke sumber daya Amazon S3 dan Amazon ECR tertentu:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket/object",
                "arn:aws:s3:::outputbucket/path"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "arn:aws:ecr:region::repository/my-repo"
        }
    ]
}
```

Jika `CreateTrainingJob.AlgorithSpecifications.TrainingImage` perlu mengakses sumber data lain, seperti DynamoDB atau sumber daya Amazon RDS, tambahkan izin yang relevan ke kebijakan ini.

Dalam kebijakan sebelumnya, Anda mencakup kebijakan sebagai berikut:

- `s3:ListBucket`Cakupkan izin ke bucket tertentu yang Anda tentukan `InputDataConfig.DataSource.S3DataSource.S3Uri` sebagai `CreateTrainingJob` permintaan.
- Cakupan `s3:GetObject` dan `s3:PutObject` izin ke objek berikut yang Anda tentukan dalam konfigurasi data input dan output dalam `CreateTrainingJob` permintaan:

  `InputDataConfig.DataSource.S3DataSource.S3Uri`

  `OutputDataConfig.S3OutputPath`
- Cakupan izin Amazon ECR ke jalur registri (`AlgorithmSpecification.TrainingImage`) yang Anda tentukan dalam permintaan. `CreateTrainingJob`

Tindakan `cloudwatch` dan `logs` tindakan berlaku untuk sumber daya “*”. Untuk informasi selengkapnya, lihat CloudWatch Sumber Daya dan Operasi di Panduan CloudWatch Pengguna Amazon.

Jika Anda menentukan VPC pribadi untuk pekerjaan pelatihan Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterface",
      "ec2:CreateNetworkInterfacePermission",
      "ec2:DeleteNetworkInterface",
      "ec2:DeleteNetworkInterfacePermission",
      "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeVpcs",
      "ec2:DescribeDhcpOptions",
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ]
}
```

Jika input Anda dienkripsi menggunakan enkripsi sisi server dengan kunci yang AWS dikelola KMS (SSE-KMS), tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ]
}
```

Jika Anda menentukan kunci KMS dalam konfigurasi output pekerjaan pelatihan Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:Encrypt"
    ]
}
```

Jika Anda menentukan kunci volume KMS dalam konfigurasi sumber daya pekerjaan pelatihan Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:CreateGrant"
    ]
}
```

CreateModel API: Izin Peran Eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `CreateModel` API, Anda dapat melampirkan kebijakan izin berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "s3:GetObject",
                "s3:ListBucket",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "*"
        }
    ]
}
```

Alih-alih menentukan`"Resource": "*"`, Anda dapat mencakup izin ini ke sumber daya Amazon S3 dan Amazon ECR tertentu:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::inputbucket/object"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": [
                "arn:aws:ecr:region::repository/my-repo",
                "arn:aws:ecr:region::repository/my-repo"
             ]
        }
    ]
}
```

Jika `CreateModel.PrimaryContainer.Image` perlu mengakses sumber data lain, seperti Amazon DynamoDB atau sumber daya Amazon RDS, tambahkan izin yang relevan ke kebijakan ini.

Dalam kebijakan sebelumnya, Anda mencakup kebijakan sebagai berikut:

- Cakupan izin S3 untuk objek yang Anda tentukan `PrimaryContainer.ModelDataUrl` dalam permintaan. `CreateModel`
- Cakupan izin Amazon ECR ke jalur registri tertentu yang Anda tentukan sebagai `PrimaryContainer.Image` dan `SecondaryContainer.Image` dalam permintaan. `CreateModel`

Tindakan `cloudwatch` dan `logs` tindakan berlaku untuk sumber daya "*". Untuk informasi selengkapnya, lihat CloudWatch Sumber Daya dan Operasi di Panduan CloudWatch Pengguna Amazon.

> **Note**
>
> Jika Anda berencana untuk menggunakan fitur pagar pembatas SageMaker penerapan untuk penerapan model dalam produksi, pastikan peran eksekusi Anda memiliki izin untuk melakukan `cloudwatch:DescribeAlarms` tindakan pada alarm rollback otomatis Anda.

Jika Anda menentukan VPC pribadi untuk model Anda, tambahkan izin berikut:

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ]
}
```

SageMaker peran kemampuan geospasial

Sebagai layanan terkelola, kemampuan SageMaker geospasial Amazon melakukan operasi atas nama Anda pada AWS perangkat keras yang dikelola oleh SageMaker. Gunakan AWS Identity and Access Management untuk memberi pengguna, grup, dan peran akses ke SageMaker geospasial.

Administrator IAM dapat memberikan izin ini kepada pengguna, grup, atau peran menggunakanAWS Management Console,AWS CLI, atau salah satu SDKAWS.

Untuk menggunakan SageMaker geospasial, Anda memerlukan izin IAM berikut.

1. Peran SageMaker eksekusi.

   Untuk menggunakan operasi API khusus SageMaker geospasial, peran SageMaker eksekusi Anda harus menyertakan prinsip layanan SageMaker geospasial, `sagemaker-geospatial.amazonaws.com` dalam kebijakan kepercayaan peran eksekusi. Hal ini memungkinkan peran SageMaker eksekusi untuk melakukan tindakan atas nama Anda. Akun AWS
2. Pengguna, grup, atau peran yang memiliki akses Amazon SageMaker Studio Classic dan SageMaker geospasial

   Untuk memulai SageMaker geospasial, Anda dapat menggunakan kebijakan AWS terkelola:`AmazonSageMakerGeospatialFullAccess`. Hibah ini akan memberikan pengguna, grup, atau peran akses penuh ke SageMaker geospasial. Untuk melihat kebijakan

dan mempelajari lebih lanjut tentang tindakan, sumber daya, dan ketentuan yang tersedia, lihatAWSkebijakan terkelola: AmazonSageMakerFullAccess.

Untuk memulai Studio Classic dan membuat SageMaker Domain Amazon, lihatIkhtisar SageMaker Domain Amazon.

Gunakan topik berikut untuk membuat peran SageMaker eksekusi baru, memperbarui peran eksekusi yang ada SageMaker, dan mempelajari cara mengelola izin menggunakan tindakan, sumber daya, dan kondisi IAM khusus SageMaker geospasial.

Topik

- Membuat peran SageMaker eksekusi baru
- Menambahkan prinsip layanan SageMaker geospasial ke peran SageMaker eksekusi yang ada
- StartEarthObservationJobAPI: Izin peran eksekusi
- StartVectorEnrichmentJobAPI: Izin peran eksekusi
- ExportEarthObservationJobAPI: Izin peran eksekusi
- ExportVectorEnrichmentJobAPI: Izin Peran Eksekusi

Membuat peran SageMaker eksekusi baru

Untuk bekerja dengan kemampuan SageMaker geospasial, Anda harus mengatur pengguna, grup, atau peran, dan peran eksekusi. Peran pengguna adalah AWS identitas dengan kebijakan izin yang menentukan apa yang dapat dan tidak dapat dilakukan pengguna di dalamnyaAWS. Peran eksekusi adalah peran IAM yang memberikan izin layanan untuk mengakses sumber daya AndaAWS. Peran eksekusi terdiri dari izin dan kebijakan kepercayaan. Kebijakan kepercayaan menentukan prinsipal mana yang memiliki izin untuk mengambil peran.

SageMaker geospasial juga membutuhkan prinsip layanan yang berbeda,`sagemaker-geospatial.amazonaws.com`. Jika Anda adalah SageMaker pelanggan yang sudah ada, Anda harus menambahkan prinsip layanan tambahan ini ke kebijakan kepercayaan Anda.

Gunakan prosedur berikut untuk membuat peran eksekusi baru dengan kebijakan terkelola IAM,,`AmazonSageMakerGeospatialFullAccess`, terlampir. Jika kasus penggunaan Anda memerlukan izin yang lebih terperinci, gunakan bagian lain dari panduan ini untuk membuat peran eksekusi yang memenuhi kebutuhan bisnis Anda.

> **Important**
>
> Kebijakan terkelola IAM`AmazonSageMakerGeospatialFullAccess`, yang digunakan dalam prosedur berikut, hanya memberikan izin peran eksekusi untuk melakukan tindakan Amazon S3 tertentu pada bucket atau objek `SageMaker` dengan`Sagemaker`,,`sagemaker`, `aws-glue` atau dalam nama. Untuk mempelajari cara memperbarui kebijakan peran eksekusi agar akses ke bucket dan objek Amazon S3 lainnya, lihat. Menambahkan Izin Amazon S3 Tambahan ke Peran Eksekusi SageMaker

Untuk membuat peran baru

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Pilih Peran dan kemudian pilih Buat peran.
3. Pilih SageMaker.
4. Pilih Berikutnya: Izin.
5. Kebijakan terkelola IAM, secara otomatis `AmazonSageMakerGeospatialFullAccess` dilampirkan ke peran ini. Untuk melihat izin yang disertakan dalam kebijakan ini, pilih panah menyamping di samping nama kebijakan. Pilih Berikutnya: Tag.
6. (Opsional) Tambahkan tag dan pilih Berikutnya: Tinjauan.
7. Beri nama peran di bidang teks di bawah Nama peran dan pilih Buat peran.
8. Di bagian Peran konsol IAM, pilih peran yang baru saja Anda buat di langkah 7. Jika perlu, gunakan kotak teks untuk mencari peran menggunakan nama peran yang Anda masukkan di langkah 7.
9. Pada halaman ringkasan peran, catat ARN.

Menambahkan prinsip layanan SageMaker geospasial ke peran SageMaker eksekusi yang ada

Untuk menggunakan operasi API khusus SageMaker geospasial, peran SageMaker eksekusi Anda harus menyertakan prinsip layanan SageMaker geospasial, `sagemaker-geospatial.amazonaws.com` dalam kebijakan kepercayaan peran eksekusi. Hal ini memungkinkan peran SageMaker eksekusi untuk melakukan tindakan atas nama Anda. Akun AWS

Tindakan seperti melewati peran antar layanan adalah hal biasa di dalamnya SageMaker. Untuk lebih detailnya,

Untuk menambahkan prinsip layanan SageMaker geospasial ke peran SageMaker eksekusi yang ada, perbarui kebijakan yang ada untuk memasukkan prinsip layanan SageMaker geospasial seperti yang ditunjukkan dalam kebijakan kepercayaan berikut. Dengan melampirkan prinsip layanan ke kebijakan kepercayaan, peran SageMaker eksekusi sekarang dapat menjalankan API khusus SageMaker geospasial atas nama Anda.

Untuk mempelajari lebih lanjut tentang tindakan, sumber daya, dan kondisi IAM khusus SageMaker geospasial, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk SageMaker Panduan Pengguna IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker-geospatial.amazonaws.com",
                    "sagemaker.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

`StartEarthObservationJob`API: Izin peran eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `StartEarthObservationJob` API, Anda dapat melampirkan kebijakan izin minimum berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucketMultipartUploads"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
        "Effect": "Allow",
        "Action": "sagemaker-geospatial:GetEarthObservationJob",
        "Resource":  "arn:aws:sagemaker-geospatial:*:*:earth-observation-job/*"
        },
        {
        "Effect": "Allow",
        "Action": "sagemaker-geospatial:GetRasterDataCollection",
        "Resource": "arn:aws:sagemaker-geospatial:*:*:raster-data-collection/*"
        }
    ]
    }
```

Jika bucket Amazon S3 masukan Anda dienkripsi menggunakan enkripsi sisi server dengan kunci AWS KMS terkelola (SSE-KMS), lihat Menggunakan Kunci Bucket Amazon S3 untuk informasi selengkapnya.

`StartVectorEnrichmentJob`API: Izin peran eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `StartVectorEnrichmentJob` API, Anda dapat melampirkan kebijakan izin minimum berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
```

```
            ]
        },
        {
        "Effect": "Allow",
        "Action": "sagemaker-geospatial:GetVectorEnrichmentJob",
        "Resource":  "arn:aws:sagemaker-geospatial:*:*:vector-enrichment-job/*"
        }
    ]
    }
```

Jika bucket Amazon S3 masukan Anda dienkripsi menggunakan enkripsi sisi server dengan kunci AWS KMS terkelola (SSE-KMS), lihat Menggunakan Kunci Bucket Amazon S3 untuk informasi selengkapnya.

`ExportEarthObservationJob`API: Izin peran eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `ExportEarthObservationJob` API, Anda dapat melampirkan kebijakan izin minimum berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
        "Effect": "Allow",
        "Action": "sagemaker-geospatial:GetEarthObservationJob",
        "Resource":  "arn:aws:sagemaker-geospatial:*:*:earth-observation-job/*"
        }
    ]
    }
```

Jika bucket Amazon S3 masukan Anda dienkripsi menggunakan enkripsi sisi server dengan kunci AWS KMS terkelola (SSE-KMS), lihat Menggunakan Kunci Bucket Amazon S3 untuk informasi selengkapnya.

`ExportVectorEnrichmentJob`API: Izin Peran Eksekusi

Untuk peran eksekusi yang dapat diteruskan dalam permintaan `ExportVectorEnrichmentJob` API, Anda dapat melampirkan kebijakan izin minimum berikut ke peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
        "Effect": "Allow",
        "Action": "sagemaker-geospatial:GetVectorEnrichmentJob",
        "Resource":  "arn:aws:sagemaker-geospatial:*:*:vector-enrichment-job/*"
        }
    ]
    }
```

Jika bucket Amazon S3 masukan Anda dienkripsi menggunakan enkripsi sisi server dengan kunci AWS KMS terkelola (SSE-KMS), lihat Menggunakan Kunci Bucket Amazon S3.

Manajer SageMaker Peran Amazon

Administrator machine learning (ML) yang berjuang untuk mendapatkan izin hak istimewa paling sedikit dengan SageMaker Amazon harus memperhitungkan keragaman perspektif industri, termasuk kebutuhan akses hak istimewa paling unik yang diperlukan untuk persona seperti ilmuwan data,

insinyur operasi pembelajaran mesin (MLOP), dan banyak lagi. Gunakan Amazon SageMaker Role Manager untuk membangun dan mengelola peran IAM berbasis persona untuk kebutuhan pembelajaran mesin umum secara langsung melalui konsol Amazon. SageMaker

Amazon SageMaker Role Manager menyediakan 3 persona peran yang telah dikonfigurasi sebelumnya dan izin yang telah ditentukan sebelumnya untuk 12 aktivitas MLM umum. Jelajahi persona yang disediakan dan kebijakan yang disarankan, atau buat dan pertahankan peran untuk persona yang unik untuk kebutuhan bisnis Anda. Jika Anda memerlukan penyesuaian tambahan, tentukan izin jaringan dan enkripsi untuk sumber daya Amazon Virtual Private Cloud dan kunci AWS Key Management Serviceenkripsi di Langkah 1. Masukkan informasi peran Amazon SageMaker Role Manager.

Topik

- Menggunakan manajer peran (konsol)
- Menggunakan manajer peran (AWS CDK)
- Referensi persona
- Referensi aktivitas MLM
- Luncurkan Studio Classic
- FAQ Role Manager

Menggunakan manajer peran (konsol)

Anda dapat menggunakan Amazon SageMaker Role Manager dari lokasi berikut di navigasi sebelah kiri konsol Amazon SageMaker :

- Memulai — Tambahkan kebijakan izin dengan cepat untuk pengguna Anda.
- Domain — Tambahkan kebijakan izin untuk pengguna dalam Domain Amazon SageMaker .
- Notebook — Tambahkan izin paling sedikit untuk pengguna yang membuat dan menjalankan buku catatan.
- Pelatihan — Tambahkan izin paling sedikit untuk pengguna yang membuat dan mengelola pekerjaan pelatihan.
- Inferensi — Tambahkan izin paling sedikit untuk pengguna yang menerapkan dan mengelola model untuk inferensi.

Anda dapat menggunakan prosedur berikut ini untuk memulai proses pembuatan peran dari lokasi yang berbeda di SageMaker konsol.

Memulai

Jika Anda menggunakan SageMaker untuk pertama kalinya, kami merekomendasikan membuat peran dari bagian Memulai.

Untuk membuat peran menggunakan Amazon SageMaker Role Manager, lakukan hal berikut.

1. Buka SageMaker konsol Amazon.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi Admin.
3. Di bawah Konfigurasi admin, pilih Manajer peran.
4. Pilih Buat peran.

Domain

Anda dapat membuat peran menggunakan Amazon SageMaker Role Manager saat memulai proses pembuatan SageMaker Domain Amazon.

Untuk membuat peran menggunakan Amazon SageMaker Role Manager, lakukan hal berikut.

1. Buka SageMaker konsol Amazon.
2. Di panel navigasi sebelah kiri, pilih Konfigurasi Admin.
3. Di bawah Konfigurasi Admin, pilih Domain.
4. Pilih Create domain (Buat domain).
5. Pilih Buat peran menggunakan panduan pembuatan peran.

Buku catatan

Anda dapat membuat peran menggunakan Amazon SageMaker Role Manager saat memulai proses pembuatan buku catatan.

Untuk membuat peran menggunakan Amazon SageMaker Role Manager, lakukan hal berikut.

1. Buka SageMaker konsol Amazon.
2. Di navigasi kiri, pilih Notebook.
3. Pilih instans Notebook.

4. Pilih Buat instans notebook.
5. Pilih Buat peran menggunakan panduan pembuatan peran.

Pelatihan

Anda dapat membuat peran menggunakan Amazon SageMaker Role Manager saat memulai proses pembuatan pekerjaan pelatihan.

Untuk membuat peran menggunakan Amazon SageMaker Role Manager, lakukan hal berikut.

1. Buka SageMaker konsol Amazon.
2. Di navigasi kiri, pilih Pelatihan.
3. Pilih Pekerjaan Pelatihan.
4. Pilih Buat pekerjaan pelatihan.
5. Pilih Buat peran menggunakan panduan pembuatan peran.

Inferensi

Anda dapat membuat peran menggunakan Amazon SageMaker Role Manager saat memulai proses penerapan model untuk inferensi.

Untuk membuat peran menggunakan Amazon SageMaker Role Manager, lakukan hal berikut.

1. Buka SageMaker konsol Amazon.
2. Di navigasi kiri, pilih Inferensi.
3. Pilih Model.
4. Pilih Buat model.
5. Pilih Buat peran menggunakan panduan pembuatan peran.

Setelah menyelesaikan salah satu prosedur sebelumnya, gunakan informasi di bagian berikut untuk membantu Anda membuat peran.

Prasyarat

Untuk menggunakan Amazon SageMaker Role Manager, Anda harus memiliki izin untuk membuat peran IAM. Izin ini biasanya tersedia untuk administrator dan peran dan peran dengan izin hak istimewa paling sedikit untuk praktisi ML.

Anda dapat mengambil peran IAM untuk sementara waktu dalam AWS Management Console dengan beralih peran. Untuk informasi selengkapnya tentang metode untuk menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Langkah 1. Masukkan informasi peran

Berikan nama untuk digunakan sebagai akhiran unik dari SageMaker peran baru Anda. Secara default, awalan `"sagemaker-"` ditambahkan ke setiap nama peran untuk pencarian yang lebih mudah di konsol IAM. Misalnya, jika Anda memberi nama peran `test-123` selama pembuatan peran, peran Anda akan muncul seperti `sagemaker-test-123` di konsol IAM. Anda dapat menambahkan deskripsi peran Anda secara opsional untuk memberikan detail tambahan.

Kemudian, pilih salah satu persona yang tersedia untuk mendapatkan izin yang disarankan untuk persona seperti ilmuwan data, insinyur data, atau insinyur operasi pembelajaran mesin (MLOP). Untuk informasi tentang persona yang tersedia dan izin yang disarankan, lihat. Referensi persona Untuk membuat peran tanpa izin yang disarankan untuk memandu Anda, pilih Pengaturan Peran Kustom.

> **Note**
>
> Sebaiknya gunakan manajer peran terlebih dahulu untuk membuat Peran SageMaker Komputasi sehingga sumber daya SageMaker komputasi memiliki kemampuan untuk melakukan tugas seperti pelatihan dan inferensi. Gunakan persona Peran SageMaker Komputasi untuk membuat peran ini dengan manajer peran. Setelah membuat Peran SageMaker Komputasi, perhatikan ARN untuk penggunaan masa depan.

Kondisi jaringan dan enkripsi

Kami menyarankan Anda mengaktifkan kustomisasi VPC untuk menggunakan konfigurasi VPC, subnet, dan grup keamanan dengan kebijakan IAM yang terkait dengan peran baru Anda. Saat kustomisasi VPC diaktifkan, kebijakan IAM untuk aktivitas ML yang berinteraksi dengan sumber daya VPC dicakup untuk akses hak istimewa paling sedikit. Kustomisasi VPC tidak diaktifkan secara default. Untuk detail lebih lanjut tentang arsitektur jaringan yang direkomendasikan, lihat Arsitektur jaringan di Panduan AWS Teknis.

Anda juga dapat menggunakan kunci KMS untuk mengenkripsi, mendekripsi, dan mengenkripsi ulang data untuk beban kerja yang diatur dengan data yang sangat sensitif. Saat AWS KMS kustomisasi diaktifkan, kebijakan IAM untuk aktivitas ML yang mendukung kunci enkripsi kustom

dicakup untuk akses hak istimewa paling sedikit. Untuk informasi selengkapnya, lihat Enkripsi dengan AWS KMS di Panduan AWS Teknis.

Langkah 2. Konfigurasikan aktivitas MLM

Setiap aktivitas Amazon SageMaker Role Manager MLL menyertakan izin IAM yang disarankan untuk menyediakan akses ke sumber daya yang relevanAWS. Beberapa aktivitas ML mengharuskan Anda menambahkan ARN peran layanan untuk menyelesaikan penyiapan. Untuk informasi tentang aktivitas ML yang telah ditentukan sebelumnya dan izinnya, lihat. Referensi aktivitas MLM Untuk informasi tentang menambahkan peran layanan, lihatPeran layanan.

Berdasarkan persona yang dipilih, aktivitas ML tertentu sudah dipilih. Anda dapat membatalkan pilihan aktivitas ML yang disarankan atau memilih aktivitas tambahan untuk membuat peran Anda sendiri. Jika Anda memilih persona Pengaturan Peran Kustom, maka tidak ada aktivitas ML yang dipilih sebelumnya dalam langkah ini.

Anda dapat menambahkan kebijakan IAM tambahan AWS atau yang dikelola pelanggan ke peran Anda. Langkah 3: Tambahkan kebijakan dan tag tambahan

Peran layanan

Beberapa AWS layanan memerlukan peran layanan untuk melakukan tindakan atas nama Anda. Jika aktivitas ML yang Anda pilih mengharuskan Anda untuk meneruskan peran layanan, maka Anda harus menyediakan ARN untuk peran layanan tersebut.

Anda dapat membuat peran layanan baru atau menggunakan peran yang sudah ada, seperti peran layanan yang dibuat dengan persona Peran SageMaker Komputasi. Anda dapat menemukan ARN dari peran yang ada dengan memilih nama peran di bagian Peran konsol IAM. Untuk mempelajari lebih lanjut tentang peran layanan, lihat Membuat peran untuk AWS layanan.

Langkah 3: Tambahkan kebijakan dan tag tambahan

Anda dapat menambahkan kebijakan IAM yang sudah ada AWS atau yang dikelola pelanggan ke peran baru Anda. Untuk informasi tentang SageMaker kebijakan yang ada, lihat Kebijakan AWS Terkelola untuk Amazon SageMaker. Anda juga dapat memeriksa kebijakan yang ada di bagian Peran konsol IAM.

Secara opsional, gunakan kondisi kebijakan berbasis tag untuk menetapkan informasi metadata guna mengkategorikan dan mengelola sumber daya. AWS Setiap tag diwakili oleh pasangan nilai kunci. Untuk informasi selengkapnya, lihat Mengontrol akses ke AWS sumber daya menggunakan tag.

Peran tinjau

Luangkan waktu untuk meninjau semua informasi yang terkait dengan peran baru Anda. Pilih Sebelumnya untuk kembali dan mengedit informasi apa pun. Saat Anda siap untuk membuat peran Anda, pilih Buat peran. Ini menghasilkan peran dengan izin untuk aktivitas ML yang Anda pilih. Anda dapat melihat peran baru Anda di bagian Peran konsol IAM.

Menggunakan manajer peran (AWS CDK)

Gunakan AWS Cloud Development Kit (AWS CDK) dengan Amazon SageMaker Role Manager untuk membuat peran secara terprogram dan menetapkan izin. Anda dapat menggunakan AWS CDK untuk menyelesaikan tugas apa pun yang dapat Anda lakukan menggunakan. AWS Management Console Akses terprogram CDK membuatnya lebih mudah untuk memberikan izin yang memberi pengguna Anda akses ke sumber daya tertentu. Untuk informasi selengkapnya tentangAWS CDK, lihat Apa ituAWS CDK?

> **Important**
>
> Anda harus menggunakan persona Peran SageMaker Komputasi untuk membuat Peran SageMaker Komputasi. Untuk informasi selengkapnya tentang komputasi persona, lihat. SageMaker menghitung persona Untuk kode yang dapat Anda gunakan untuk membuat peran komputasi di dalamnyaAWS CDK, lihatBerikan Izin ke Personona Komputasi.

Berikut ini adalah contoh tugas yang dapat Anda lakukan diAWS CDK:

- Buat peran IAM dengan izin granular untuk persona pembelajaran mesin (ML), seperti Ilmuwan Data dan Insinyur MLOP.
- Berikan izin untuk konstruksi CDK dari persona ML atau aktivitas ML.
- Tetapkan parameter kondisi aktivitas ML.
- Aktifkan VPC dan AWS Key Management Service kondisi Amazon global dan tetapkan nilainya.
- Pilih dari semua versi aktivitas ML untuk pengguna Anda tanpa menyebabkan gangguan pada akses mereka.

Ada AWS tugas umum yang terkait dengan pembelajaran mesin (ML) dengan SageMaker yang memerlukan izin IAM tertentu. Izin untuk melakukan tugas didefinisikan sebagai aktivitas ML di Amazon SageMaker Role Manager. Aktivitas ML menentukan satu set izin yang ditautkan ke peran

IAM. Misalnya, aktivitas ML untuk Amazon SageMaker Studio Classic memiliki semua izin yang diperlukan pengguna untuk mengakses Studio Classic. Untuk informasi selengkapnya tentang aktivitas ML, lihatReferensi aktivitas MLM.

Saat Anda membuat peran, pertama-tama Anda menentukan konstruksi untuk persona ML atau aktivitas ML. Konstruksi adalah sumber daya dalam AWS CDK tumpukan. Misalnya, konstruksi bisa berupa bucket Amazon S3, subnet Amazon VPC, atau peran IAM.

Saat Anda membuat persona atau aktivitas, Anda dapat membatasi izin yang terkait dengan persona atau aktivitas tersebut ke sumber daya tertentu. Misalnya, Anda dapat menyesuaikan aktivitas agar hanya memberikan izin untuk subnet tertentu dalam VPC Amazon.

Setelah menetapkan izin, Anda dapat membuat peran dan meneruskan peran tersebut untuk membuat sumber daya lain, seperti instance SageMaker buku catatan.

Berikut ini adalah contoh kode di TypeScript untuk tugas yang dapat Anda selesaikan menggunakan CDK. Saat membuat aktivitas, Anda menentukan ID dan opsi untuk konstruksi aktivitas. Opsi adalah kamus yang menentukan parameter yang diperlukan untuk aktivitas, seperti Amazon S3. Anda meneruskan kamus kosong untuk aktivitas yang tidak memiliki parameter yang diperlukan.

Berikan Izin ke Personona Komputasi

Kode berikut membuat persona Data Scientist MLdengan satu set aktivitas ML khusus untuk persona. Izin dari aktivitas ML hanya berlaku untuk VPC Amazon AWS KMS dan konfigurasi yang ditentukan dalam konstruksi persona. Kode berikut membuat kelas untuk persona Data Scientist. Aktivitas ML didefinisikan dalam daftar aktivitas. Izin VPC dan izin KMS didefinisikan sebagai parameter opsional di luar daftar aktivitas.

Setelah Anda mendefinisikan kelas, Anda dapat membuat peran sebagai konstruksi dalam AWS CDK tumpukan. Anda juga dapat membuat instans notebook. Orang yang menggunakan peran IAM yang Anda buat dalam kode berikut dapat mengakses instance notebook saat mereka masuk ke AWS akun mereka.

```
export class myCDKStack extends cdk.Stack {
  constructor(scope: cdk.App, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    const persona = new Persona(this, 'example-persona-id', {
        activities: [
```

```
            Activity.accessAwsServices(this, 'example-id1', {})
        ]
    });

    const role = persona.createRole(this, 'example-IAM-role-id', 'example-IAM-role-
name');

    }
}
```

Berikan izin kepada persona Ilmuwan Data

Kode berikut membuat persona Data Scientist MLdengan satu set aktivitas ML khusus untuk persona. Izin dari aktivitas ML hanya berlaku untuk konfigurasi VPC dan KMS yang ditentukan dalam konstruksi persona. Kode berikut membuat kelas untuk persona Data Scientist. Aktivitas ML didefinisikan dalam daftar aktivitas. Izin VPC Amazon dan izin didefinisikan sebagai parameter opsional di luar daftar aktivitas. AWS KMS

Setelah Anda mendefinisikan kelas, Anda dapat membuat peran sebagai konstruksi dalam AWS CDK tumpukan. Anda juga dapat membuat instans notebook. Orang yang menggunakan peran IAM yang Anda buat dalam kode berikut dapat mengakses instance notebook saat mereka masuk ke AWS akun mereka.

```
export class myCDKStack extends cdk.Stack {
  constructor(scope: cdk.App, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    const persona = new Persona(this, 'example-persona-id', {
        activities: [
            Activity.runStudioAppsV2(this, 'example-id1', {}),
            Activity.manageJobs(this, 'example-id2', {rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
            Activity.manageModels(this, 'example-id3', {rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
            Activity.manageExperiments(this, 'example-id4', {}),
            Activity.visualizeExperiments(this, 'example-id5', {}),
            Activity.accessS3Buckets(this, 'example-id6', {s3buckets:
 [s3.S3Bucket.fromBucketName('DOC-EXAMPLE-BUCKET')]})
        ],
        // optional: to configure VPC permissions
```

```
        subnets: [ec2.Subnet.fromSubnetId('example-VPC-subnet-id')],
        securityGroups: [ec2.SecurityGroup.fromSecurityGroupId('example-VPC-security-
group-id')],
        // optional: to configure KMS permissions
        dataKeys: [kms.Key.fromKeyArn('example-KMS-key-ARN')],
        volumeKeys: [kms.Key.fromKeyArn('example-KMS-key-ARN')],
    });

    const role = persona.createRole(this, 'example-IAM-role-id', 'example-IAM-role-
name');

    const notebookInstance = new CfnNotebookInstance(this, 'example-notebook-instance-
name', { RoleArn: role.RoleArn, ...});
    }
}
```

Berikan Izin ke Personona Ops

Kode berikut membuat persona Ops MLdengan satu set aktivitas ML khusus untuk persona. Izin dari aktivitas ML hanya berlaku untuk VPC Amazon AWS KMS dan konfigurasi yang ditentukan dalam konstruksi persona. Kode berikut membuat kelas untuk persona Ops Ops. Aktivitas ML didefinisikan dalam daftar aktivitas. Izin VPC dan izin KMS didefinisikan sebagai parameter opsional di luar daftar aktivitas.

Setelah Anda mendefinisikan kelas, Anda dapat membuat peran sebagai konstruksi dalam AWS CDK tumpukan. Anda juga dapat membuat profil pengguna Amazon SageMaker Studio Classic. Orang yang menggunakan peran IAM yang Anda buat dalam kode berikut dapat membuka SageMaker Studio Classic saat mereka masuk ke AWS akun mereka.

```
export class myCDKStack extends cdk.Stack {
  constructor(scope: cdk.App, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    const persona = new Persona(this, 'example-persona-id', {
        activities: [
            Activity.runStudioAppsV2(this, 'example-id1', {}),
            Activity.manageModels(this, 'example-id2', {rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
            Activity.manageEndpoints(this, 'example-id3',{rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
```

```
            Activity.managePipelines(this, 'example-id4', {rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
            Activity.visualizeExperiments(this, 'example-id5', {})
        ],
        subnets: [ec2.Subnet.fromSubnetId('example-VPC-subnet-id')],
        securityGroups: [ec2.SecurityGroup.fromSecurityGroupId('example-VPC-security-
group-id')],
        dataKeys: [kms.Key.fromKeyArn('example-KMS-key-ARN')],
        volumeKeys: [kms.Key.fromKeyArn('example-KMS-key-ARN')],
    });

    const role = persona.createRole(this, 'example-IAM-role-id', 'example-IAM-role-
name');

    let userProfile = new CfnNUserProfile(this, 'example-Studio Classic-profile-name',
 { RoleName: role.RoleName, ... });
    }
}
```

Berikan Izin ke Konstruksi

Kode berikut membuat persona Ops MLdengan satu set aktivitas ML khusus untuk persona. Kode berikut membuat kelas untuk persona Ops MS. Aktivitas ML didefinisikan dalam daftar aktivitas.

Setelah Anda mendefinisikan kelas, Anda dapat membuat peran sebagai konstruksi dalam AWS CDK tumpukan. Anda juga dapat membuat instans notebook. Kode memberikan izin dari aktivitas ML ke peran IAM dari fungsi Lambda.

```
export class myCDKStack extends cdk.Stack {
  constructor(scope: cdk.App, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    const persona = new Persona(this, 'example-persona-id', {
        activities: [
            Activity.runStudioAppsV2(this, 'example-id1', {}),
            Activity.manageModels(this, 'example-id2', {rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
            Activity.manageEndpoints(this, 'example-id3',{rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
            Activity.managePipelines(this, 'example-id4', {rolesToPass:
 [iam.Role.fromRoleName('example-IAM-role-name')]}),
```

```
            Activity.visualizeExperiments(this, 'example-id5', {})
        ],
    });

    const lambdaFn = lambda.Function.fromFunctionName('example-lambda-function-name');
    persona.grantPermissionsTo(lambdaFn);
    }
}
```

Berikan izin untuk satu aktivitas ML

Kode berikut membuat aktivitas ML dan membuat peran dari aktivitas. Izin dari aktivitas hanya berlaku untuk konfigurasi VPC dan KMS yang Anda tentukan untuk pengguna.

```
export class myCDKStack extends cdk.Stack {
  constructor(scope: cdk.App, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    const activity = Activity.manageJobs(this, 'example-activity-id', {
        rolesToPass: [iam.Role.fromRoleName('example-IAM-role-name')],
        subnets: [ec2.Subnet.fromSubnetId('example-VPC-subnet-id')],
        securityGroups: [ec2.SecurityGroup.fromSecurityGroupId('example-VPC-security-
group-id')],
        dataKeys: [kms.Key.fromKeyArn('example-KMS-key-ARN')],
        volumeKeys: [kms.Key.fromKeyArn('example-KMS-key-ARN')],
    });

    const role = activity.createRole(this, 'example-IAM-role-id', 'example-IAM-role-
name');
    }
}
```

Buat peran dan berikan izin untuk satu aktivitas

Kode berikut membuat peran IAM untuk aktivitas MLtunggal.

```
export class myCDKStack extends cdk.Stack {
  constructor(scope: cdk.App, id: string, props?: cdk.StackProps) {
```

```
    super(scope, id, props);

    const activity = Activity.manageJobs(this, 'example-activity-id', {
        rolesToPass: [iam.Role.fromRoleName('example-IAM-role-name')],
    });


    activity.create_role(this, 'example-IAM-role-id', 'example-IAM-role-name')
    }
}
```

Referensi persona

Amazon SageMaker Role Manager memberikan izin yang disarankan untuk sejumlah persona ML. Ini termasuk peran eksekusi pengguna untuk tanggung jawab praktisi MS umum serta peran eksekusi layanan untuk interaksi AWS layanan umum yang diperlukan untuk bekerja dengannya SageMaker.

Setiap persona telah menyarankan izin dalam bentuk aktivitas ML yang dipilih. Untuk informasi tentang aktivitas ML yang telah ditentukan sebelumnya dan izinnya, lihat. Referensi aktivitas MLM

Persona ilmuwan data

Gunakan persona ini untuk mengonfigurasi izin untuk melakukan pengembangan dan eksperimen pembelajaran mesin umum di suatu lingkungan. SageMaker Persona ini mencakup aktivitas ML yang telah dipilih sebelumnya berikut:

- Jalankan Aplikasi Studio Classic
- Mengelola Lowongan Alur
- Kelola Model
- Kelola Eksperimen
- Cari dan Visualisasikan Eksperimen
- Akses Bucket Amazon S3

Persona MLOPs

Pilih persona ini untuk mengonfigurasi izin untuk aktivitas operasional. Persona ini mencakup aktivitas ML yang telah dipilih sebelumnya berikut:

- Jalankan Aplikasi Studio Classic

- Kelola Model
- Kelola Titik Akhir
- Mengelola Alur
- Cari dan Visualisasikan Eksperimen

SageMaker menghitung persona

> **Note**
>
> Sebaiknya gunakan manajer peran terlebih dahulu untuk membuat Peran SageMaker Komputasi sehingga sumber daya SageMaker komputasi dapat melakukan tugas seperti pelatihan dan inferensi. Gunakan persona Peran SageMaker Komputasi untuk membuat peran ini dengan manajer peran. Setelah membuat Peran SageMaker Komputasi, perhatikan ARN untuk penggunaan masa depan.

Persona ini mencakup aktivitas ML yang telah dipilih sebelumnya berikut:

- Akses AWS Layanan yang Diperlukan

Referensi aktivitas MLM

Aktivitas ML adalah AWS tugas umum yang terkait dengan pembelajaran mesin dengan SageMaker yang memerlukan izin IAM tertentu. Setiap persona menyarankan aktivitas ML terkait saat membuat peran dengan Amazon SageMaker Role Manager. Anda dapat memilih aktivitas ML tambahan atau membatalkan pilihan aktivitas ML yang disarankan untuk membuat peran yang memenuhi kebutuhan bisnis unik Anda.

Amazon SageMaker Role Manager menyediakan izin yang telah ditentukan untuk aktivitas ML berikut:

Aktivitas ML	Deskripsi
Akses AWS Layanan yang Diperlukan	Izin untuk mengakses Amazon S3, Amazon ECR, Amazon ECR, CloudWatch Amazon EC2. Diperlukan untuk peran eksekusi untuk pekerjaan dan titik akhir.

Aktivitas ML	Deskripsi
Jalankan Aplikasi Studio Classic	Izin untuk beroperasi dalam lingkungan Studio Classic. Diperlukan untuk peran eksekusi domain dan profil pengguna.
Mengelola Lowongan Alur	Izin untuk mengaudit, menanyakan garis keturunan, dan memvisualisasikan eksperimen.
Kelola Model	Izin untuk mengelola SageMaker pekerjaan di seluruh siklus hidupnya.
Kelola Titik Akhir	Izin untuk mengelola penyebaran dan SageMaker pembaruan titik akhir.
Mengelola Alur	Izin untuk mengelola SageMaker saluran pipa dan eksekusi pipa.
Kelola Eksperimen	Izin untuk mengelola SageMaker eksperimen dan uji coba.
Cari dan Visualisasikan Eksperimen	Izin untuk mengaudit, menanyakan garis keturunan, dan memvisualisasikan eksperimen.
Kelola Pemantauan Model	Izin untuk mengelola jadwal pemantauan untuk SageMaker Model Monitor.
S3 Akses Penuh	Izin untuk melakukan semua operasi Amazon S3.
Akses Bucket S3	Izin untuk melakukan operasi pada bucket S3 tertentu.
Kueri Athena Workgroups	Izin untuk menjalankan dan mengelola kueri Amazon Athena.

Luncurkan Studio Classic

Gunakan peran yang berfokus pada persona untuk meluncurkan Studio Classic. Jika Anda seorang administrator, Anda dapat memberi pengguna Anda akses ke Studio Classic dan meminta mereka mengambil peran persona mereka baik secara langsung melalui AWS Management Console atau melalui. AWS IAM Identity Center

Luncurkan Studio Classic dengan AWS Management Console

Untuk ilmuwan data atau pengguna lain untuk mengasumsikan persona mereka melaluiAWS Management Console, mereka memerlukan peran konsol untuk sampai ke lingkungan Studio Classic.

Anda tidak dapat menggunakan Amazon SageMaker Role Manager untuk membuat peran yang memberikan izin ke. AWS Management Console Namun, setelah membuat peran layanan di manajer peran, Anda dapat pergi ke konsol IAM untuk mengedit peran dan menambahkan peran akses pengguna. Berikut ini adalah contoh peran yang menyediakan akses pengguna keAWS Management Console:

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Sid": "DescribeCurrentDomain",
            "Effect": "Allow",
            "Action": "sagemaker:DescribeDomain",
            "Resource": "arn:aws:sagemaker:<REGION>:<ACCOUNT-ID>:domain/<STUDIO-DOMAIN-
ID>"
        },
        {
            "Sid": "RemoveErrorMessagesFromConsole",
            "Effect": "Allow",
            "Action":
            [
                "servicecatalog:ListAcceptedPortfolioShares",
                "sagemaker:GetSagemakerServicecatalogPortfolioStatus",
                "sagemaker:ListModels",
                "sagemaker:ListTrainingJobs",
                "servicecatalog:ListPrincipalsForPortfolio",
                "sagemaker:ListNotebookInstances",
                "sagemaker:ListEndpoints"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "RequiredForAccess",
            "Effect": "Allow",
            "Action":
            [
                "sagemaker:ListDomains",
                "sagemaker:ListUserProfiles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreatePresignedURLForAccessToDomain",
            "Effect": "Allow",
            "Action": "sagemaker:CreatePresignedDomainUrl",
            "Resource": "arn:aws:sagemaker:<REGION>:<ACCOUNT-ID>:user-profile/<STUDIO-
DOMAIN-ID>/<PERSONA_NAME>"
        }
    ]
}
```

Di panel kontrol Studio Classic, pilih Tambah Pengguna untuk membuat pengguna baru. Di bagian Pengaturan Umum, beri nama pengguna Anda dan setel peran eksekusi default agar pengguna menjadi peran yang Anda buat menggunakan Amazon SageMaker Role Manager.

Pada layar berikutnya, pilih versi Jupyter Lab yang sesuai, dan apakah akan mengaktifkan template SageMaker Jumpstart dan Project. SageMaker Lalu pilih Selanjutnya. Pada halaman pengaturan SageMaker Canvas, pilih apakah akan mengaktifkan dukungan SageMaker Canvas, dan juga apakah akan mengizinkan perkiraan waktu di Canvas. SageMaker Kemudian pilih Kirim.

Pengguna baru Anda sekarang akan terlihat di panel kontrol Studio Classic. Untuk menguji pengguna ini, pilih Studio dari daftar tarik-turun Luncurkan aplikasi di baris yang sama dengan nama pengguna.

Luncurkan Studio Classic dengan IAM Identity Center

Untuk menetapkan pengguna IAM Identity Center ke peran eksekusi, pengguna harus terlebih dahulu ada di direktori IAM Identity Center. Untuk informasi selengkapnya, lihat Mengelola identitas di Pusat Identitas IAM di. AWS IAM Identity Center

> **Note**
>
> Direktori Autentikasi Pusat Identitas IAM Anda dan Domain Studio Klasik harus sama. Wilayah AWS

1. Untuk menetapkan pengguna IAM Identity Center ke Domain Studio Classic Anda, pilih Tetapkan pengguna dan Grup di panel kontrol Studio Classic. Di layar Tetapkan pengguna dan grup pilih pengguna ilmuwan data Anda, lalu pilih Tetapkan Pengguna dan Grup.
2. Setelah pengguna ditambahkan ke panel kontrol Studio Classic, pilih pengguna untuk membuka layar detail pengguna.
3. Pada layar Detail pengguna, pilih Edit.
4. Pada layar Edit profil pengguna, di bawah Pengaturan umum, ubah peran eksekusi default agar sesuai dengan peran eksekusi pengguna yang Anda buat untuk ilmuwan data Anda.
5. Pilih Berikutnya melalui halaman pengaturan lainnya, dan pilih Kirim untuk menyimpan perubahan Anda.

Saat ilmuwan data Anda atau pengguna lain masuk ke portal Pusat Identitas IAM, mereka melihat ubin untuk Domain Klasik Studio ini. Memilih ubin itu akan mencatatnya ke Studio Classic dengan peran eksekusi pengguna yang ditetapkan.

FAQ Role Manager

Lihat item FAQ berikut untuk jawaban atas pertanyaan umum tentang Manajer SageMaker Peran Amazon.

T. Bagaimana cara mengakses Amazon SageMaker Role Manager?

J: Anda dapat mengakses Amazon SageMaker Role Manager melalui beberapa lokasi di SageMaker konsol Amazon. Untuk informasi tentang mengakses manajer peran dan menggunakannya untuk membuat peran, lihatMenggunakan manajer peran (konsol).

Q. Apa itu persona?

J: Persona adalah grup izin yang telah dikonfigurasi sebelumnya berdasarkan tanggung jawab pembelajaran mesin (ML) umum. Misalnya, persona ilmu data menyarankan izin untuk pengembangan dan eksperimen pembelajaran mesin umum di suatu SageMaker lingkungan, sedangkan persona MLOP menyarankan izin untuk aktivitas ML. yang terkait dengan operasi.

Q. Apa itu kegiatan ML?

J: Aktivitas ML adalah AWS tugas umum yang terkait dengan pembelajaran mesin dengan SageMaker yang memerlukan izin IAM tertentu. Setiap persona menyarankan aktivitas ML terkait saat membuat peran dengan Amazon SageMaker Role Manager. Aktivitas ML mencakup tugas-tugas seperti akses penuh Amazon S3 atau mencari dan memvisualisasikan eksperimen. Untuk informasi selengkapnya, lihat Referensi aktivitas MLM.

T. Apakah peran yang saya buat dengan peran manajer peran AWS Identity and Access Management (IAM)?

J: Ya. Peran yang dibuat menggunakan Amazon SageMaker Role Manager adalah peran IAM dengan kebijakan akses yang disesuaikan. Anda dapat melihat peran yang dibuat di bagian Peran konsol IAM.

T. Bagaimana cara melihat peran yang saya buat menggunakan Amazon SageMaker Role Manager?

J: Anda dapat melihat peran yang dibuat di bagian Peran konsol IAM. Secara default, awalan `"sagemaker-"` ditambahkan ke setiap nama peran untuk pencarian yang lebih mudah di konsol IAM. Misalnya, jika Anda memberi nama peran `test-123` selama pembuatan peran, peran Anda akan muncul seperti `sagemaker-test-123` di konsol IAM.

T. Dapatkah saya memodifikasi peran yang dibuat dengan Amazon SageMaker Role Manager setelah dibuat?

J: Ya. Anda dapat mengubah peran dan kebijakan yang dibuat oleh Amazon SageMaker Role Manager melalui konsol IAM. Untuk informasi selengkapnya, lihat Memodifikasi peran dalam Panduan AWS Identity and Access Management Pengguna.

T. Dapatkah saya melampirkan kebijakan saya sendiri ke peran yang dibuat menggunakan Amazon SageMaker Role Manager?

J: Ya. Anda dapat melampirkan kebijakan IAM apa pun AWS atau yang dikelola pelanggan dari akun ke peran yang Anda buat menggunakan Amazon SageMaker Role Manager.

T. Berapa banyak kebijakan yang dapat saya tambahkan ke peran yang saya buat dengan Amazon SageMaker Role Manager?

J: Batas maksimum untuk melampirkan kebijakan terkelola ke peran IAM atau pengguna adalah 20. Batas ukuran karakter maksimum untuk kebijakan terkelola adalah 6.144. Untuk informasi selengkapnya, lihat kuota objek IAM dan persyaratan nama IAM dan AWS Security Token Service kuota, dan batas karakter.

T. Dapatkah saya menambahkan kondisi ke aktivitas ML?

J: Ketentuan apa pun yang Anda berikan di Langkah 1. Masukkan informasi peran Manajer SageMaker Peran Amazon, seperti subnet, grup keamanan, atau kunci KMS, secara otomatis diteruskan ke aktivitas ML yang dipilih. Langkah 2. Konfigurasikan aktivitas MLM Anda juga dapat menambahkan kondisi tambahan ke aktivitas ML jika perlu. Misalnya, Anda juga dapat menambahkan `InstanceTypes` atau `IntercontainerTrafficEncryption` mengkondisikan aktivitas Kelola Pekerjaan Pelatihan.

T. Dapatkah saya menggunakan penandaan untuk mengelola akses ke sumber daya apa pun? AWS

J: Anda dapat menambahkan tag ke peran Anda di Langkah 3: Tambahkan kebijakan dan tag tambahan Manajer SageMaker Peran Amazon. Agar berhasil mengelola AWS sumber daya menggunakan tag, Anda harus menambahkan tag yang sama ke peran dan kebijakan terkait. Misalnya, Anda dapat menambahkan tag ke peran dan ke bucket Amazon S3. Kemudian, karena peran meneruskan tag ke SageMaker sesi, hanya pengguna dengan peran itu yang dapat mengakses bucket S3 itu. Anda dapat menambahkan tag ke kebijakan melalui konsol IAM. Untuk informasi selengkapnya, lihat Menandai peran IAM di AWS Identity and Access ManagementPanduan Pengguna.

T. Dapatkah saya menggunakan Amazon SageMaker Role Manager untuk membuat peran untuk mengakses? AWS Management Console

A: Tidak. Namun, setelah membuat peran layanan di manajer peran, Anda dapat pergi ke konsol IAM untuk mengedit peran dan menambahkan peran akses manusia di konsol IAM.

T. Apa perbedaan antara peran federasi pengguna dan peran SageMaker eksekusi?

J: Peran federasi pengguna secara langsung diasumsikan oleh pengguna untuk mengakses AWS sumber daya seperti akses ke fileAWS Management Console. Peran SageMaker eksekusi diasumsikan oleh SageMaker layanan untuk melakukan fungsi atas nama pengguna atau alat otomatisasi. Misalnya, saat pengguna membuka instance Studio Classic, Studio Classic akan mengasumsikan peran eksekusi yang terkait dengan profil pengguna untuk mengakses AWS sumber daya atas nama pengguna. Jika profil pengguna tidak menentukan peran eksekusi, maka peran eksekusi ditentukan di tingkat SageMaker Domain Amazon.

T. Jika saya menggunakan aplikasi web kustom yang mengakses Studio Classic melalui url presigned, peran apa yang digunakan?

J: Jika Anda menggunakan aplikasi web khusus untuk mengakses Studio Classic, maka Anda memiliki peran federasi pengguna hibrida dan peran SageMaker eksekusi. Pastikan bahwa peran

ini memiliki izin hak istimewa paling sedikit untuk apa yang dapat dilakukan pengguna dan apa yang dapat dilakukan Studio Classic atas nama pengguna terkait.

T: Dapatkah saya menggunakan Amazon SageMaker Role Manager dengan autentikasi AWS IAM Identity Center untuk domain Studio Classic saya?

J: Aplikasi Cloud AWS IAM Identity Center Studio Classic menggunakan peran eksekusi Studio Classic untuk memberikan izin kepada pengguna federasi. Peran eksekusi ini dapat ditentukan di tingkat profil pengguna Studio Classic IAM Identity Center atau tingkat domain default. Identitas dan grup pengguna harus disinkronkan ke Pusat Identitas IAM dan profil pengguna Studio Classic harus dibuat dengan penugasan pengguna IAM Identity Center menggunakan. CreateUserProfile Untuk informasi selengkapnya, lihat Luncurkan Studio Classic dengan IAM Identity Center.

Kontrol akses untuk notebook

Anda harus menggunakan prosedur yang berbeda untuk mengontrol akses ke notebook Amazon SageMaker Studio Classic dan instance SageMaker notebook karena mereka memiliki lingkungan runtime yang berbeda. Studio Classic menggunakan izin sistem file dan kontainer untuk mengontrol akses ke notebook Studio Classic dan isolasi pengguna. Sebuah instance SageMaker notebook memberikan pengguna yang login ke instance notebook akses root default. Topik berikut menjelaskan cara mengubah izin untuk kedua jenis notebook.

Topik

- Kontrol akses dan pengaturan izin untuk notebook SageMaker Studio
- Kontrol akses root ke instance SageMaker notebook

Kontrol akses dan pengaturan izin untuk notebook SageMaker Studio

Amazon SageMaker Studio menggunakan sistem file dan izin kontainer untuk kontrol akses dan isolasi pengguna dan notebook Studio. Ini adalah salah satu perbedaan utama antara notebook Studio dan instance SageMaker notebook. Topik ini menjelaskan bagaimana izin diatur untuk menghindari ancaman keamanan, SageMaker apa yang dilakukan secara default, dan bagaimana pelanggan dapat menyesuaikan izin. Untuk informasi selengkapnya tentang notebook Studio dan lingkungan runtime mereka, lihat. Gunakan Notebook Amazon SageMaker Studio Classic

SageMaker izin aplikasi

Pengguna run-as adalah pengguna/grup POSIX yang digunakan untuk menjalankan JupyterServer aplikasi dan KernelGateway aplikasi di dalam wadah.

Pengguna run-as untuk JupyterServer aplikasi ini adalah sagemaker-user (1000) secara default. Pengguna ini memiliki izin sudo untuk mengaktifkan instalasi dependensi seperti paket yum.

Pengguna run-as untuk KernelGateway aplikasi adalah root (0) secara default. Pengguna ini dapat menginstal dependensi menggunakan pip/apt-get/conda.

Karena pemetaan ulang pengguna, tidak ada pengguna yang dapat mengakses sumber daya atau membuat perubahan pada instance host.

Pemetaan ulang pengguna

SageMaker melakukan pemetaan ulang pengguna untuk memetakan pengguna di dalam wadah ke pengguna pada instance host di luar wadah. Rentang ID pengguna (0 - 65535) dalam wadah dipetakan ke ID pengguna yang tidak memiliki hak istimewa di atas 65535 pada instance. Misalnya, sagemaker-user (1000) di dalam container mungkin memetakan ke user (200001) pada instance, di mana angka dalam tanda kurung adalah ID pengguna. Jika pelanggan membuat pengguna/grup baru di dalam wadah, itu tidak akan mendapat hak istimewa pada instance host terlepas dari ID pengguna/grup. Pengguna root wadah juga dipetakan ke pengguna yang tidak memiliki hak istimewa pada instance. Untuk informasi selengkapnya, lihat Mengisolasi kontainer dengan namespace pengguna.

> **Note**
>
> File yang dibuat oleh pengguna sagemaker-user mungkin terlihat seperti dimiliki oleh sagemaker-studio (uid 65534). Ini adalah efek samping dari mode pembuatan aplikasi cepat di mana gambar SageMaker kontainer ditarik sebelumnya, memungkinkan aplikasi untuk memulai dalam waktu kurang dari satu menit. Jika aplikasi Anda mengharuskan uid pemilik file dan uid pemilik proses untuk mencocokkan, minta layanan pelanggan untuk menghapus nomor akun Anda dari fitur pra-tarik gambar.

Izin gambar kustom

Pelanggan dapat membawa SageMaker gambar kustom mereka sendiri. Gambar-gambar ini dapat menentukan run-as user/group yang berbeda untuk meluncurkan aplikasi. KernelGateway Pelanggan dapat menerapkan kontrol izin berbutir halus di dalam gambar, misalnya, untuk menonaktifkan akses root atau melakukan tindakan lain. Pemetaan ulang pengguna yang sama berlaku di sini. Untuk informasi selengkapnya, lihat Bawa SageMaker gambar Anda sendiri.

Isolasi kontainer

Docker menyimpan daftar kemampuan default yang dapat digunakan penampung. SageMaker tidak menambahkan kemampuan tambahan. SageMaker menambahkan aturan rute khusus untuk memblokir permintaan ke Amazon EFS dan layanan metadata instance (IMDS) dari container. Pelanggan tidak dapat mengubah aturan rute ini dari kontainer. Untuk informasi selengkapnya, lihat:: Keistimewaan waktu aktif dan kemampuan Linux.

Akses metadata aplikasi

Metadata yang digunakan dengan menjalankan aplikasi dipasang ke wadah dengan izin hanya-baca. Pelanggan tidak dapat memodifikasi metadata ini dari wadah. Untuk metadata yang tersedia, lihat. Dapatkan Notebook Studio Classic dan Metadata Aplikasi

Isolasi pengguna di EFS

Saat Anda melakukan onboard ke Studio, SageMaker buat volume Amazon Elastic File System (EFS) untuk domain Anda yang dibagikan oleh semua pengguna Studio di domain tersebut. Setiap pengguna mendapatkan direktori home pribadi mereka sendiri pada volume EFS. Direktori home ini digunakan untuk menyimpan notebook pengguna, repositori Git, dan data lainnya. Untuk mencegah pengguna lain di domain mengakses data pengguna, SageMaker buat ID pengguna yang unik secara global untuk profil pengguna dan menerapkannya sebagai ID pengguna/grup POSIX untuk direktori home pengguna.

Akses EBS

Volume Amazon Elastic Block Store (Amazon EBS) dilampirkan ke instans host dan dibagikan di semua gambar. Ini digunakan untuk volume root notebook dan menyimpan data sementara yang dihasilkan di dalam wadah. Penyimpanan tidak bertahan saat instance yang menjalankan notebook dihapus. Pengguna root di dalam wadah tidak dapat mengakses volume EBS.

Akses IMDS

Karena masalah keamanan, akses ke Layanan Metadata Instans (IMDS) Amazon Elastic Compute Cloud (Amazon EC2) tidak tersedia di Studio. SageMaker Untuk informasi selengkapnya tentang IMDS, lihat Metadata instans dan data pengguna.

Kontrol akses root ke instance SageMaker notebook

Secara default, saat Anda membuat instance notebook, pengguna yang masuk ke instance notebook tersebut memiliki akses root. Ilmu data adalah proses berulang yang mungkin memerlukan ilmuwan

data untuk menguji dan menggunakan alat dan paket perangkat lunak yang berbeda, sehingga banyak pengguna instance notebook perlu memiliki akses root untuk dapat menginstal alat dan paket ini. Karena pengguna dengan akses root memiliki hak administrator, pengguna dapat mengakses dan mengedit semua file pada instance notebook dengan akses root diaktifkan.

Jika Anda tidak ingin pengguna memiliki akses root ke instance notebook, saat Anda memanggil `CreateNotebookInstance`atau `UpdateNotebookInstance`mengoperasikannya, setel `RootAccess` field ke`Disabled`. Anda juga dapat menonaktifkan akses root untuk pengguna saat Anda membuat atau memperbarui instance notebook di SageMaker konsol Amazon. Untuk informasi, lihat Langkah 1: Buat Instans SageMaker Notebook Amazon.

> **Note**
>
> Konfigurasi siklus hidup membutuhkan akses asal untuk dapat mengatur instans notebook. Karena itu, konfigurasi siklus hidup yang terkait dengan instans notebook selalu berjalan dengan akses asal meskipun Anda menonaktifkan akses asal untuk pengguna.

> **Note**
>
> Untuk alasan keamanan, Rootless Docker diinstal pada instance notebook yang dinonaktifkan root, bukan Docker biasa. Untuk informasi selengkapnya, lihat Menjalankan daemon Docker sebagai pengguna non-root (mode Rootless)

Izin SageMaker API Amazon: Referensi Tindakan, Izin, dan Sumber Daya

setiap operasi SageMaker API Amazon, tindakan terkait yang dapat Anda berikan izin untuk melakukan tindakan tersebut, dan AWS sumber daya tempat Anda dapat memberikan izin. Anda menentukan tindakan dalam bidang `Action` kebijakan, dan Anda menentukan nilai sumber daya pada bidang `Resource` kebijakan.

> **Note**
>
> Kecuali untuk `ListTags` API, pembatasan tingkat sumber daya tidak tersedia pada panggilan. `List-` Setiap pengguna yang memanggil `List-` API akan melihat semua sumber daya dari jenis itu di akun.

Untuk menyatakan ketentuan dalam SageMaker kebijakan Amazon, Anda dapat menggunakan kunci kondisi AWS -wide. Untuk daftar lengkap kunci di seluruh AWS, lihat Kunci yang Tersedia di Panduan Pengguna IAM.

> **Warning**
>
> Beberapa tindakan SageMaker API mungkin masih dapat diakses melaluiSearch API. Misalnya, jika pengguna memiliki kebijakan IAM yang menolak izin untuk `Describe` panggilan untuk SageMaker sumber daya tertentu, pengguna tersebut masih dapat mengakses informasi deskripsi melalui API Pencarian. Untuk sepenuhnya membatasi akses pengguna ke `Describe` panggilan, Anda juga harus membatasi akses ke API Pencarian. Untuk daftar SageMaker sumber daya yang dapat diakses melalui API Pencarian, lihat Referensi AWS CLI Perintah SageMaker Pencarian.

Operasi SageMaker API Amazon dan Izin yang Diperlukan untuk Tindakan

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DeleteEarthObservationJob	sagemaker-geospatial:DeleteEarthObservationJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id*
DeleteVectorEnrichmentJob	sagemaker-geospatial:DeleteVectorEnrichmentJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*
ExportEarthObservationJob	sagemaker-geospatial:ExportEarthObservationJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
ExportVectorEnrichmentJob	sagemaker-geospatial:ExportVectorEnrichmentJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*
GetEarthObservationJob	sagemaker-geospatial:GetEarthObservationJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id*
GetRasterDataCollection	sagemaker-geospatial:GetRasterDataCollection	arn:aws:sagemaker-geospatial: *region*:*account-id* :raster-data-collection/public/ *id*
GetTile	sagemaker-geospatial:GetTile	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id*
GetVectorEnrichmentJob	sagemaker-geospatial:GetVectorEnrichmentJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*
ListEarthObservationJobs	sagemaker-geospatial:ListEarthObservationJobs	*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
ListRasterDataCollections	sagemaker-geospatial:ListRasterDataCollections	*
ListTagsForResource	sagemaker-geospatial:ListTagsForResource	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id* arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*
ListVectorEnrichmentJobs	sagemaker-geospatial:ListVectorEnrichmentJobs	*
SearchRasterDataCollection	sagemaker-geospatial:SearchRasterDataCollection	arn:aws:sagemaker-geospatial: *region*:*account-id* :raster-data-collection/public/ *id*
StartEarthObservationJob	sagemaker-geospatial:StartEarthObservationJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
StartVectorEnrichmentJob	sagemaker-geospatial:StartVectorEnrichmentJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*
StopEarthObservationJob	sagemaker-geospatial:StopEarthObservationJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id*
StopVectorEnrichmentJob	sagemaker-geospatial:StopVectorEnrichmentJob	arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*
TagResource	sagemaker-geospatial:TagResource	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id* arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
UntagResource	sagemaker-geospatial:UntagResource	arn:aws:sagemaker-geospatial: *region*:*account-id* :earth-observation-job/*id* arn:aws:sagemaker-geospatial: *region*:*account-id* :vector-enrichment-job/*id*
AddTags	sagemaker:AddTags	arn:aws:sagemaker: *region*:*account-id* :*
CreateApp	sagemaker:CreateApp	arn:aws:sagemaker: *region*:*account-id* :app/*domain-id* /*user-profile-name* /*app-type*/*appName*
CreateApp ImageConfig	sagemaker:CreateAppImageConfig	arn:aws:sagemaker: *region*:*account-id* :app-image-config/ *appImageConfigName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateAutoMLJob	sagemaker:CreateAutoMLJob iam:PassRole Izin berikut hanya diperlukan jika salah satu yang terkait ResourceConfig memiliki peran tertentu VolumeKmsKeyId dan peran terkait tidak memiliki kebijakan yang mengizinkan tindakan ini: kms:CreateGrant	arn:aws:sagemaker:*region*:*account-id*:automl-job/*autoMLJobName*
CreateAutoMLJobV2	sagemaker:CreateAutoMLJobV2 iam:PassRole Izin berikut hanya diperlukan jika salah satu yang terkait ResourceConfig memiliki peran tertentu VolumeKmsKeyId dan peran terkait tidak memiliki kebijakan yang mengizinkan tindakan ini: kms:CreateGrant	arn:aws:sagemaker:*region*:*account-id*:automl-job/*autoMLJobName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateDomain	sagemaker:CreateDomain iam:CreateServiceLinkedRole iam:PassRole Diperlukan jika kunci terkelola pelanggan KMS ditentukan untukKmsKeyId: elasticfilesystem:CreateFileSystem kms:CreateGrant kms:Decrypt kms:DescribeKey kms:GenerateDataKeyWithoutPlainText Diperlukan untuk membuat domain yang mendukung RStudio: sagemaker:CreateApp	arn:aws:sagemaker:*region*:*account-id*:domain/*domain-id*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateEndpoint	sagemaker:CreateEndpoint kms:CreateGrant (diperlukan hanya jika yang terkait EndPointConfig memiliki yang KmsKeyId ditentukan)	arn:aws:sagemaker:*region*:*account-id* :endpoint/ *endpointName* arn:aws:sagemaker:*region*:*account-id* :endpoint-config/*endpointConfigName*
CreateEndpointConfig	sagemaker:CreateEndpointConfig	arn:aws:sagemaker:*region*:*account-id* :endpoint-config/*endpointConfigName*
CreateFlowDefinition	sagemaker:CreateFlowDefinition iam:PassRole	arn:aws:sagemaker:*region*:*account-id* :flow-definition/ *flowDefinitionName*
CreateHumanTaskUi	sagemaker:CreateHumanTaskUi	arn:aws:sagemaker:*region*:*account-id* :human-task-ui/ *humanTaskUiName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateInferenceRecommendationsJob	sagemaker:CreateInferenceRecommendationsJob iam:PassRole Izin berikut diperlukan hanya jika Anda menentukan kunci enkripsi: kms:CreateGrant kms:Decrypt kms:DescribeKey kms:GenerateDataKey	arn:aws:sagemaker:*region*:*account-id* :inference-recommendations-job/ *inferenceRecommendationsJobName*
CreateHyperParameterTuningJob	sagemaker:CreateHyperParameterTuningJob iam:PassRole Izin berikut hanya diperlukan jika salah satu yang terkait ResourceConfig memiliki peran tertentu VolumeKmsKeyId dan peran terkait tidak memiliki kebijakan yang mengizinkan tindakan ini: kms:CreateGrant	arn:aws:sagemaker:*region*:*account-id* :hyper-parameter-tuning-job/ *hyperParameterTuningJobName*
CreateImage	sagemaker:CreateImage iam:PassRole	arn:aws:sagemaker:*region*:*account-id* :image/*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateImageVersion	sagemaker:CreateImageVersion	arn:aws:sagemaker: *region*:*account-id* :image-version/ *imageName* /*
CreateLabelingJob	pembuat sagemaker: CreateLabelingJob saya: PassRole	arn:aws:sagemaker: *region*:*account-id* :labeling-job/ *labelingJobName*
CreateModel	sagemaker:CreateModel iam:PassRole	arn:aws:sagemaker: *region*:*account-id* :model/*modelName*
CreateModelPackage	sagemaker:CreateModelPackage	arn:aws:sagemaker: *region*:*account-id* :model-package/ *modelPackageName*
CreateModelPackageGroup	sagemaker:CreateModelPackageGroup	arn:aws:sagemaker: *region*:*account-id* :model-package-group/ *modelPackageGroupName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateNotebookInstance	sagemaker:CreateNotebookInstance iam:PassRole Izin berikut hanya diperlukan jika Anda menentukan VPC untuk instance notebook Anda: ec2:CreateNetworkInterface ec2:DescribeSecurityGroups ec2:DescribeSubnets ec2:DescribeVpcs Izin berikut hanya diperlukan jika Anda menentukan VPC dan akselerator inferensi elastis untuk instance notebook Anda: ec2:DescribeVpcEndpoints Izin berikut diperlukan hanya jika Anda menentukan kunci enkripsi: kms:DescribeKey kms:CreateGrant Izin berikut hanya diperlukan jika Anda menentukan AWS	arn:aws:sagemaker:*region*:*account-id*:notebook-instance/*notebookInstanceName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
	rahasia Secrets Manager untuk mengakses repositori Git pribadi: `secretsmanager:GetSecretValue`	

CreatePipeline	`sagemaker:CreatePipeline`
	
	`iam:PassRole`
		`arn:aws-partition :sagemaker: region:account-id :pipeline/ pipeline-name`
		
		`arn:aws-partition :iam::account-id :role/role-name`

CreatePresignedDomainUrl	`sagemaker:CreatePresignedDomainUrl`	`arn:aws:sagemaker: region:account-id :app/ domain-id/ userProfileName /*`
CreatePresignedNotebookInstanceUrl	`sagemaker:CreatePresignedNotebookInstanceUrl`	`arn:aws:sagemaker: region:account-id :notebook-instance / notebookInstanceName`

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateProcessingJob	sagemaker:CreateProcessingJob iam:PassRole kms:CreateGrant (diperlukan hanya jika yang terkait ProcessingResources memiliki peran tertentu VolumeKmsKeyId dan peran terkait tidak memiliki kebijakan yang mengizinkan tindakan ini)	arn:aws:sagemaker:*region*:*account-id*:processing-job/*processingJobName*
CreateSpace	sagemaker:CreateSpace	arn:aws:sagemaker:*region*:*account-id*:space/*domain-id*/*spaceName*
CreateStudioLifecycleConfig	sagemaker:CreateStudioLifecycleConfig	arn:aws:sagemaker:*region*:*account-id*:studio-lifecycle-config/.*
CreateTrainingJob	sagemaker:CreateTrainingJob iam:PassRole kms:CreateGrant (diperlukan hanya jika yang terkait ResourceConfig memiliki peran tertentu VolumeKmsKeyId dan peran terkait tidak memiliki kebijakan yang mengizinkan tindakan ini)	arn:aws:sagemaker:*region*:*account-id*:training-job/*trainingJobName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateTransformJob	sagemaker:CreateTransformJob kms:CreateGrant (diperlukan hanya jika yang terkait TransformResources memiliki peran tertentu VolumeKmsKeyId dan peran terkait tidak memiliki kebijakan yang mengizinkan tindakan ini)	arn:aws:sagemaker:*region*:*account-id*:transform-job/*transformJobName*
CreateUserProfile	sagemaker:CreateUserProfile iam:PassRole	arn:aws:sagemaker:*region*:*account-id*:user-profile/domain-id/*userProfileName*
CreateWorkforce	sagemaker:CreateWorkforce cognito-idp:DescribeUserPoolClient cognito-idp:UpdateUserPool cognito-idp:DescribeUserPool cognito-idp:UpdateUserPoolClient	arn:aws:sagemaker:*region*:*account-id*:workforce/*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
CreateWorkteam	sagemaker:CreateWorkteam cognito-idp:DescribeUserPoolClient cognito-idp:UpdateUserPool cognito-idp:DescribeUserPool cognito-idp:UpdateUserPoolClient	arn:aws:sagemaker:*region*:*account-id* :workteam/private-crowd/ *work team name*
DeleteApp	sagemaker:DeleteApp	arn:aws:sagemaker:*region*:*account-id* :app/*domain-id* /*user-profile-name* /*app-type*/*appName*
DeleteAppImageConfig	sagemaker:DeleteAppImageConfig	arn:aws:sagemaker:*region*:*account-id* :app-image-config/ *appImageConfigName*
DeleteDomain	sagemaker:DeleteDomain	arn:aws:sagemaker:*region*:*account-id* :domain/*domainId*
DeleteEndpoint	sagemaker:DeleteEndpoint	arn:aws:sagemaker:*region*:*account-id* :endpoint/ *endpointName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DeleteEndpointConfig	sagemaker:DeleteEndpointConfig	arn:aws:sagemaker: *region*:*account-id* :endpoint-config/*endpointConfigName*
DeleteFlowDefinition	sagemaker:DeleteFlowDefinition	arn:aws:sagemaker: *region*:*account-id* :flow-definition/ *flowDefinitionName*
DeleteHumanLoop	sagemaker:DeleteHumanLoop	arn:aws:sagemaker: *region*:*account-id* :human-loop/ *humanLoopName*
DeleteImage	sagemaker:DeleteImage	arn:aws:sagemaker: *region*:*account-id* :image/*imageName*
DeleteImageVersion	sagemaker:DeleteImageVersion	arn:aws:sagemaker: *region*:*account-id* :image-version/ *imageName* /*versionNumber*
DeleteModel	sagemaker:DeleteModel	arn:aws:sagemaker: *region*:*account-id* :model/*modelName*
DeleteModelPackage	sagemaker:DeleteModelPackage	arn:aws:sagemaker: *region*:*account-id* :model-package/ *modelPackageName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DeleteModelPackageGroup	sagemaker:DeleteModelPackageGroup	arn:aws:sagemaker: *region*:*account-id* :model-package-group/ *modelPackageGroupName*
DeleteModelPackageGroupPolicy	sagemaker:DeleteModelPackageGroupPolicy	arn:aws:sagemaker: *region*:*account-id* :model-package-group/ *modelPackageGroupName*
DeleteNotebookInstance	sagemaker:DeleteNotebookInstance Izin berikut hanya diperlukan jika Anda menetapkan VPC untuk instance notebook Anda: ec2:DeleteNetworkInterface Izin berikut hanya diperlukan jika Anda menetapkan kunci enkripsi saat membuat instance notebook: kms:DescribeKey	arn:aws:sagemaker: *region*:*account-id* :notebook-instance/ *notebookInstanceName*
DeletePipeline	sagemaker:DeletePipeline	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DeleteSpace	sagemaker:DeleteSpace	arn:aws:sagemaker:*region*:*account-id* :space/*domain-id* /*spaceName*
DeleteTags	sagemaker:DeleteTags	arn:aws:sagemaker:*region*:*account-id* :*
DeleteUserProfile	sagemaker:DeleteUserProfile	arn:aws:sagemaker:*region*:*account-id* :user-profile/domain-id/*userProfileName*
DeleteWorkforce	sagemaker:DeleteWorkforce	arn:aws:sagemaker:*region*:*account-id* :workforce/*
DeleteWorkteam	sagemaker:DeleteWorkteam	arn:aws:sagemaker:*region*:*account-id* :workteam/private-crowd/*
DescribeApp	sagemaker:DescribeApp	arn:aws:sagemaker:*region*:*account-id* :app/*domain-id* /*user-profile-name* /*app-type*/*appName*
DescribeAppImageConfig	sagemaker:DescribeAppImageConfig	arn:aws:sagemaker:*region*:*account-id* :app-image-config/ *appImageConfigName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DescribeAutoMLJob	sagemaker:DescribeAutoMLJob	arn:aws:sagemaker:*region*:*account-id* :automl-job/ *autoMLJobName*
DescribeAutoMLJobV2	sagemaker:DescribeAutoMLJobV2	arn:aws:sagemaker:*region*:*account-id* :automl-job/ *autoMLJobName*
DescribeDomain	sagemaker:DescribeDomain	arn:aws:sagemaker:*region*:*account-id* :domain/*domainId*
DescribeEndpoint	sagemaker:DescribeEndpoint	arn:aws:sagemaker:*region*:*account-id* :endpoint/ *endpointName*
DescribeEndpointConfig	sagemaker:DescribeEndpointConfig	arn:aws:sagemaker:*region*:*account-id* :endpoint-config/*endpointConfigName*
DescribeFlowDefinition	sagemaker:DescribeFlowDefinition	arn:aws:sagemaker:*region*:*account-id* :flow-definition/ *flowDefinitionName*
DescribeHumanLoop	sagemaker:DescribeHumanLoop	arn:aws:sagemaker:*region*:*account-id* :human-loop/ *humanLoopName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DescribeHumanTaskUi	sagemaker:DescribeHumanTaskUi	arn:aws:sagemaker:*region*:*account-id* :human-task-ui/ *humanTaskUiName*
DescribeHyperParameterTuningJob	sagemaker:DescribeHyperParameterTuningJob	arn:aws:sagemaker:*region*:*account-id* :hyper-parameter-tuning-job/ *hyperParameterTuningJob*
DescribeImage	sagemaker:DescribeImage	arn:aws:sagemaker:*region*:*account-id* :image/*imageName*
DescribeImageVersion	sagemaker:DescribeImageVersion	arn:aws:sagemaker:*region*:*account-id* :image-version/ *imageName* /*versionNumber*
DescribeLabelingJob	sagemaker:DescribeLabelingJob	arn:aws:sagemaker:*region*:*account-id* :labeling-job/ *labelingJobName*
DescribeModel	sagemaker:DescribeModel	arn:aws:sagemaker:*region*:*account-id* :model/*modelName*
DescribeModelPackage	sagemaker:DescribeModelPackage	arn:aws:sagemaker:*region*:*account-id* :model-package/ *modelPackageName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DescribeModelPackageGroup	sagemaker:DescribeModelPackageGroup	arn:aws:sagemaker:*region*:*account-id* :model-package-group/ *modelPackageGroupName*
DescribeNotebookInstance	sagemaker:DescribeNotebookInstance	arn:aws:sagemaker:*region*:*account-id* :notebook-instance/ *notebookInstanceName*
DescribePipeline	sagemaker:DescribePipeline	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name*
DescribePipelineDefinitionForExecution	sagemaker:DescribePipelineDefinitionForExecution	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* /execution/ *execution-id*
DescribePipelineExecution	sagemaker:DescribePipelineExecution	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* /execution/ *execution-id*
DescribeProcessingJob	sagemaker:DescribeProcessingJob	arn:aws:sagemaker:*region*:*account-id* :processing-job/*processingjobname*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
DescribeSpace	sagemaker:DescribeSpace	arn:aws:sagemaker:*region*:*account-id* :space/*domain-id* /*spaceName*
DescribeSubscribedWorkteam	sagemaker:DescribeSubscribedWorkteam aws-marketplace:ViewSubscriptions	arn:aws:sagemaker:*region*:*account-id* :workteam/vendor-crowd/*
DescribeTrainingJob	sagemaker:DescribeTrainingJob	arn:aws:sagemaker:*region*:*account-id* :training-job/ *trainingjobname*
DescribeTransformJob	sagemaker:DescribeTransformJob	arn:aws:sagemaker:*region*:*account-id* :transform-job/*transformjobname*
DescribeUserProfile	sagemaker:DescribeUserProfile	arn:aws:sagemaker:*region*:*account-id* :user-profile/domain-id/*userProfileName*
DescribeWorkforce	sagemaker:DescribeWorkforce	arn:aws:sagemaker:*region*:*account-id* :workforce/*
DescribeWorkteam	sagemaker:DescribeWorkteam	arn:aws:sagemaker:*region*:*account-id* :workteam/private-crowd/*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
GetModelPackageGroupPolicy	sagemaker:GetModelPackageGroupPolicy	arn:aws:sagemaker: *region*:*account-id* :model-package-group/ *modelPackageGroupName*
InvokeEndpoint	sagemaker:InvokeEndpoint	arn:aws:sagemaker: *region*:*account-id* :endpoint/ *endpointName*
ListAppImageConfigs	sagemaker:ListAppImageConfigs	arn:aws:sagemaker: *region*:*account-id* :app-image-config/*
ListApps	sagemaker:ListApps	arn:aws:sagemaker: *region*:*account-id* :app/*domain-id* /*user-profile-name* /*
ListDomains	sagemaker:ListDomains	arn:aws:sagemaker: *region*:*account-id* :domain/*
ListEndpointConfigs	sagemaker:ListEndpointConfigs	*
ListEndpoints	sagemaker:ListEndpoints	*
ListFlowDefinitions	sagemaker:ListFlowDefinitions	*
ListHumanLoops	sagemaker:ListHumanLoops	*
ListHumanTaskUis	sagemaker:ListHumanTaskUis	*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
ListHyperParameterTuningJobs	sagemaker:ListHyperParameterTuningJobs	arn:aws:sagemaker: *region*:*account-id* :hyper-parameter-tuning-job / *hyperParameterTuningJob*
ListImages	sagemaker:ListImages	*
ListImageVersions	sagemaker:ListImageVersions	arn:aws:sagemaker: *region*:*account-id* :image/ *
ListLabelingJobs	sagemaker:ListLabelingJobs	*
ListLabelingJobsForWorkteam	sagemaker:ListLabelingJobForWorkteam	*
ListModelPackageGroups	sagemaker:ListModelPackageGroups	arn:aws:sagemaker: *region*:*account-id* :model-package-group/ *ModelPackageGroupName*
ListModelPackages	sagemaker:ListModelPackages	arn:aws:sagemaker: *region*:*account-id* :model-package/ *ModelPackageName*
ListModels	sagemaker:ListModels	*
ListNotebookInstances	sagemaker:ListNotebookInstances	*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
ListPipelineExecutions	sagemaker:ListPipelineExecutions	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name*
ListPipelineExecutionSteps	sagemaker:ListPipelineExecutionSteps	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* /execution/ *execution-id*
ListPipelineParametersForExecution	sagemaker:ListPipelineParametersForExecution	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* /execution/ *execution-id*
ListPipelines	sagemaker:ListPipelines	*
ListProcessingJobs	sagemaker:ListProcessingJobs	*
ListSpaces	sagemaker:ListSpaces	arn:aws:sagemaker: *region*:*account-id* :space/*domain-id* /*
ListSubscribedWorkteams	sagemaker:ListSubscribedWorkteams aws-marketplace:ViewSubscriptions	*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
ListTags	sagemaker:ListTags	arn:aws:sagemaker: *region*:*account-id* :*
ListTrainingJobs	sagemaker:ListTrainingJobs	*
ListTrainingJobsForHyperParameterTuningJob	sagemaker:ListTrainingJobsForHyperParameterTuningJob	arn:aws:sagemaker: *region*:*account-id* :hyper-parameter-tuning-job/ *hyperParameterTuningJob*
ListTransformJobs	sagemaker:ListTransformJobs	*
ListUserProfiles	sagemaker:ListUserProfiles	arn:aws:sagemaker: *region*:*account-id* :user-profile/domain-id/*
ListWorkforces	sagemaker:ListWorkforces	*
ListWorkteams	sagemaker:ListWorkteams	*
PutModelPackageGroupPolicy	sagemaker:PutModelPackageGroupPolicy	arn:aws:sagemaker: *region*:*account-id* :model-package-group/ *modelPackageGroupName*
RetryPipelineExecution	sagemaker:RetryPipelineExecution	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* /execution/ *execution-id*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
Search	sagemaker:Search	*
SendPipelineExecutionStepFailure	sagemaker:SendPipelineExecutionStepFailure	*
SendPipelineExecutionStepSuccess	sagemaker:SendPipelineExecutionStepSuccess	*
StartHumanLoop	sagemaker:StartHumanLoop	arn:aws:sagemaker: *region*:*account-id* :human-loop/ *humanLoopName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
StartNotebookInstance	sagemaker:StartNotebookInstance iam:PassRole Izin berikut hanya diperlukan jika Anda menetapkan VPC saat membuat instance notebook: ec2:CreateNetworkInterface ec2:DescribeNetworkInterfaces ec2:DescribeSecurityGroups ec2:DescribeSubnets ec2:DescribeVpcs Izin berikut hanya diperlukan jika Anda menentukan VPC dan akselerator inferensi elastis untuk instance notebook Anda: ec2:DescribeVpcEndpoints Izin berikut hanya diperlukan jika Anda menetapkan kunci enkripsi saat membuat instance notebook: kms:DescribeKey kms:CreateGrant	arn:aws:sagemaker:*region*:*account-id*:notebook-instance/*notebookInstanceName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
	Izin berikut hanya diperlukan jika Anda menetapkan AWS rahasia Secrets Manager untuk mengakses repositori Git pribadi saat Anda membuat instance notebook: `secretsmanager:GetSecretValue`	
StartPipelineExecution	`sagemaker:StartPipelineExecution`	`arn:aws-partition :sagemaker: region:account-id :pipeline/ pipeline-name`
StopHumanLoop	`sagemaker:StopHumanLoop`	`arn:aws:sagemaker: region:account-id :human-loop/ humanLoopName`
StopHyperParameterTuningJob	`sagemaker:StopHyperParameterTuningJob`	`arn:aws:sagemaker: region:account-id :hyper-parameter-tuning-job / hyperParameterTuningJob`
StopLabelingJob	`sagemaker:StopLabelingJob`	`arn:aws:sagemaker: region:account-id :labeling-job/ labelingJobName`
StopNotebookInstance	`sagemaker:StopNotebookInstance`	`arn:aws:sagemaker: region:account-id :notebook-instance / notebookInstanceName`

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
StopPipelineExecution	sagemaker:StopPipelineExecution	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* /execution/ *execution-id*
StopProcessingJob	sagemaker:StopProcessingJob	arn:aws:sagemaker: *region*:*account-id* :processing-job/*processingJobName*
StopTrainingJob	sagemaker:StopTrainingJob	arn:aws:sagemaker: *region*:*account-id* :training-job/ *trainingJobName*
StopTransformJob	sagemaker:StopTransformJob	arn:aws:sagemaker: *region*:*account-id* :transform-job/*transformJobName*
UpdateAppImageConfig	sagemaker:UpdateAppImageConfig	arn:aws:sagemaker: *region*:*account-id* :app-image-config/ *appImageConfigName*
UpdateDomain	sagemaker:UpdateDomain	arn:aws:sagemaker: *region*:*account-id* :domain/*domainId*
UpdateEndpoint	sagemaker:UpdateEndpoint	arn:aws:sagemaker: *region*:*account-id* :endpoint/ *endpointName*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
UpdateEndpointWeightsAndCapacities	sagemaker:UpdateEndpointWeightsAndCapacities	arn:aws:sagemaker:*region*:*account-id* :endpoint/ *endpointName*
UpdateImage	sagemaker:UpdateImage iam:PassRole	arn:aws:sagemaker:*region*:*account-id* :image/*imageName*
UpdateModelPackage	sagemaker:UpdateModelPackage	arn:aws:sagemaker:*region*:*account-id* :model-package/ *modelPackageName*
UpdateNotebookInstance	sagemaker:UpdateNotebookInstance iam:PassRole	arn:aws:sagemaker:*region*:*account-id* :notebook-instance/ *notebookInstanceName*
UpdatePipeline	sagemaker:UpdatePipeline iam:PassRole	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* arn:*aws-partition* :iam::*account-id* :role/*role-name*
UpdatePipelineExecution	sagemaker:UpdatePipelineExecution	arn:*aws-partition* :sagemaker: *region*:*account-id* :pipeline/ *pipeline-name* /execution/ *execution-id*

Operasi SageMaker API Amazon	Izin yang Diperlukan (Tindakan API)	Sumber daya
UpdateSpace	sagemaker:UpdateSpace	arn:aws:sagemaker:*region*:*account-id*:space/*domain-id*/*spaceName*
UpdateUserProfile	sagemaker:UpdateUserProfile	arn:aws:sagemaker:*region*:*account-id*:user-profile/domain-id/*userProfileName*
UpdateWorkforce	sagemaker:UpdateWorkforce	arn:aws:sagemaker:*region*:*account-id*:workforce/*
UpdateWorkteam	sagemaker:UpdateWorkteam	arn:aws:sagemaker:*region*:*account-id*:workteam/private-crowd/*

SageMaker API Amazon dan Izin yang Diperlukan untuk Tindakan

Operasi API: **AddTags**

Izin yang Diperlukan (Tindakan API): `sagemaker:AddTags`

Sumber Daya: *

Operasi API: **CreateEndpoint**

Izin yang Diperlukan (Tindakan API): `sagemaker:CreateEndpoint`

Sumber Daya: `arn:aws:sagemaker:`*region*`:`*account-id*`:endpoint/`*endpointName*

Operasi API: **CreateEndpointConfig**

Izin yang Diperlukan (Tindakan API): `sagemaker:CreateEndpointConfig`

Sumber Daya: `arn:aws:sagemaker:region:account-id:endpoint-config/endpointConfigName`

Operasi API: **CreateModel**

Izin yang Diperlukan (Tindakan API): `sagemaker:CreateModel, iam:PassRole`

Sumber Daya: `arn:aws:sagemaker:region:account-id:model/modelName`

Operasi API: **CreateLabelingJob**

Izin yang Diperlukan (Tindakan API): `sagemaker:CreateLabelingJob, iam:PassRole`

Sumber Daya: `arn:aws:sagemaker:region:account-id:labeling-job/labelingJobName`

Operasi API: **CreateNotebookInstance**

Izin yang Diperlukan (Tindakan API): `sagemaker:CreateNotebookInstance, iam:PassRole, ec2:CreateNetworkInterface, ec2:AttachNetworkInterface, ec2:ModifyNetworkInterfaceAttribute, ec2:DescribeAvailabilityZones, ec2:DescribeInternetGateways, ec2:DescribeSecurityGroups, ec2:DescribeSubnets, ec2:DescribeVpcs, kms:CreateGrant`

Sumber Daya: `arn:aws:sagemaker:region:account-id:notebook-instance/notebookInstanceName`

Operasi API: **CreateTrainingJob**

Izin yang Diperlukan (Tindakan API): `sagemaker:CreateTrainingJob, iam:PassRole`

Sumber Daya: `arn:aws:sagemaker:region:account-id:training-job/trainingJobName`

Operasi API: **CreateWorkforce**

Izin yang Diperlukan (Tindakan API):`sagemaker:CreateWorkforce,cognito-idp:DescribeUserPoolClient,,cognito-idp:UpdateUserPool, cognito-idp:DescribeUserPool cognito-idp:UpdateUserPoolClient`

Sumber Daya: `arn:aws:sagemaker:region:account-id:workforce/*`

Operasi API: **CreateWorkteam**

Izin yang Diperlukan (Tindakan API):`sagemaker:CreateWorkteam`,`cognito-idp:DescribeUserPoolClient`,,`cognito-idp:UpdateUserPool`, `cognito-idp:DescribeUserPool` `cognito-idp:UpdateUserPoolClient`

Sumber daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:workteam/private-crowd/`*`work team name`*

Operasi API: **DeleteEndpoint**

Izin yang Diperlukan (Tindakan API): `sagemaker:DeleteEndpoint`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:endpoint/`*`endpointName`*

Operasi API: **DeleteEndpointConfig**

Izin yang Diperlukan (Tindakan API): `sagemaker:DeleteEndpointConfig`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:endpoint-config/`*`endpointConfigName`*

Operasi API: **DeleteModel**

Izin yang Diperlukan (Tindakan API): `sagemaker:DeleteModel`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:model/`*`modelName`*

Operasi API: **DeleteNotebookInstance**

Izin yang Diperlukan (Tindakan API): `sagemaker:DeleteNotebookInstance`, `ec2:DeleteNetworkInterface`, `ec2:DetachNetworkInterface`, `ec2:DescribeAvailabilityZones`, `ec2:DescribeInternetGateways`, `ec2:DescribeSecurityGroups`, `ec2:DescribeSubnets`, `ec2:DescribeVpcs`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:notebook-instance/`*`notebookInstanceName`*

Operasi API: **DeleteTags**

Izin yang Diperlukan (Tindakan API): `sagemaker:DeleteTags`

Sumber Daya: *

Operasi API: **DeleteWorkteam**

Izin yang Diperlukan (Tindakan API): `sagemaker:DeleteWorkforce`

Sumber Daya: `arn:aws:sagemaker:region:account-id:workforce/private-crowd/*`

Operasi API: **DeleteWorkteam**

Izin yang Diperlukan (Tindakan API): `sagemaker:DeleteWorkteam`

Sumber Daya: `arn:aws:sagemaker:region:account-id:workteam/private-crowd/*`

Operasi API: **DescribeEndpoint**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeEndpoint`

Sumber Daya: `arn:aws:sagemaker:region:account-id:endpoint/endpointName`

Operasi API: **DescribeEndpointConfig**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeEndpointConfig`

Sumber Daya: `arn:aws:sagemaker:region:account-id:endpoint-config/endpointConfigName`

Operasi API: **DescribeLabelingJob**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeLabelingJob`

Sumber Daya: `arn:aws:sagemaker:region:account-id:labeling-job/labelingJobName`

Operasi API: **DescribeModel**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeModel`

Sumber Daya: `arn:aws:sagemaker:region:account-id:model/modelName`

Operasi API: **DescribeNotebookInstance**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeNotebookInstance`

Sumber Daya: `arn:aws:sagemaker:region:account-id:notebook-instance/notebookInstanceName`

Operasi API: **DescribeSubscribedWorkforce**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeSubscribedWorkforce`, `aws-marketplace:ViewSubscriptions`

Sumber Daya: `arn:aws:sagemaker:region:account-id:workforce/*`

Operasi API: **DescribeSubscribedWorkteam**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeSubscribedWorkteam`, `aws-marketplace:ViewSubscriptions`

Sumber Daya: `arn:aws:sagemaker:region:account-id:workteam/vendor-crowd/*`

Operasi API: **DescribeTrainingJob**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeTrainingJob`

Sumber Daya: `arn:aws:sagemaker:region:account-id:training-job/trainingJobName`

Operasi API: **DescribeWorkteam**

Izin yang Diperlukan (Tindakan API): `sagemaker:DescribeWorkteam`

Sumber Daya: `arn:aws:sagemaker:region:account-id:workteam/private-crowd/*`

Operasi API: **CreatePresignedNotebookInstanceUrl**

Izin yang Diperlukan (Tindakan API): `sagemaker:CreatePresignedNotebookInstanceUrl`

Sumber Daya: `arn:aws:sagemaker:region:account-id:notebook-instance/notebookInstanceName`

Operasi API: **runtime_InvokeEndpoint**

Izin yang Diperlukan (Tindakan API): `sagemaker:InvokeEndpoint`

Sumber Daya: `arn:aws:sagemaker:region:account-id:endpoint/endpointName`

Operasi API: **ListEndpointConfigs**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListEndpointConfigs`

Sumber Daya: *

Operasi API: **ListEndpoints**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListEndpoints`

Sumber Daya: *

Operasi API: **ListLabelingJobs**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListLabelingJobs`

Sumber Daya: *

Operasi API: **ListLabelingJobsForWorkteam**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListLabelingJobsForWorkteam`

Sumber Daya: *

Operasi API: **ListModels**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListModels`

Sumber Daya: *

Operasi API: **ListNotebookInstances**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListNotebookInstances`

Sumber Daya: *

Operasi API: **ListSubscribedWorkteams**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListSubscribedWorkteam`, `aws-marketplace:ViewSubscriptions`

Sumber Daya: *

Operasi API: **ListTags**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListTags`

Sumber Daya: *

Operasi API: **ListTrainingJobs**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListTrainingJobs`

Sumber Daya: *

Operasi API: **ListWorkteams**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListWorkforces`

Sumber Daya: *

Operasi API: **ListWorkteams**

Izin yang Diperlukan (Tindakan API): `sagemaker:ListWorkteams`

Sumber Daya: *

Operasi API: **StartNotebookInstance**

Izin yang Diperlukan (Tindakan API): `sagemaker:StartNotebookInstance, iam:PassRole, ec2:CreateNetworkInterface, ec2:AttachNetworkInterface, ec2:ModifyNetworkInterfaceAttribute, ec2:DescribeAvailabilityZones, ec2:DescribeInternetGateways, ec2:DescribeSecurityGroups, ec2:DescribeSubnets, ec2:DescribeVpcs, kms:CreateGrant`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:notebook-instance/`*`notebookInstanceName`*

Operasi API: **StopLabelingJob**

Izin yang Diperlukan (Tindakan API): `sagemaker:StopLabelingJob`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:labeling-job/`*`labelingJobName`*

Operasi API: **StopNotebookInstance**

Izin yang Diperlukan (Tindakan API): `sagemaker:StopNotebookInstance`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:notebook-instance/`*`notebookInstanceName`*

Operasi API: **StopTrainingJob**

Izin yang Diperlukan (Tindakan API): `sagemaker:StopTrainingJob`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:training-job/`*`trainingJobName`*

Operasi API: **UpdateEndpoint**

Izin yang Diperlukan (Tindakan API): `sagemaker:UpdateEndpoints`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:endpoint/`*`endpointName`*

Operasi API: **`UpdateNotebookInstance`**

Izin yang Diperlukan (Tindakan API): `sagemaker:UpdateNotebookInstance, iam:PassRole`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:notebook-instance/`*`notebookInstanceName`*

Operasi API: **`UpdateWorkteam`**

Izin yang Diperlukan (Tindakan API): `sagemaker:UpdateWorkteam`

Sumber Daya: `arn:aws:sagemaker:`*`region`*`:`*`account-id`*`:workteam/private-crowd/*`

AWSKebijakan Terkelola untuk Amazon SageMaker

Menambahkan izin ke para pengguna, grup, dan peran lebih mudah dilakukan dengan menggunakan kebijakan terkelola AWS dibandingkan dengan menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan terkelola pelanggan IAM yang hanya menyediakan izin sesuai kebutuhan tim Anda. Untuk mulai dengan cepat, Anda dapat menggunakan kebijakan-kebijakan terkelola AWS kami. Kebijakan-kebijakan ini mencakup kasus penggunaan umum dan tersedia di akun AWS Anda. Untuk informasi lebih lanjut tentang kebijakan-kebijakan terkelola AWS, lihat kebijakan terkelola AWS di Panduan Pengguna IAM.

Layanan AWS mempertahankan dan memperbarui kebijakan-kebijakan terkelola AWS. Anda tidak dapat mengubah izin yang ada dalam kebijakan-kebijakan yang dikelola AWS. Layanan terkadang menambahkan izin tambahan ke kebijakan yang dikelola AWS untuk mendukung fitur-fitur baru. Jenis pembaruan ini memengaruhi semua identitas (pengguna, grup, dan peran) tempat kebijakan terlampir. Layanan kemungkinan besar akan memperbarui kebijakan yang dikelola AWS saat ada fitur baru yang diluncurkan atau saat ada operasi baru yang tersedia. Layanan tidak menghapus izin yang ada di kebijakan yang dikelola AWS, sehingga pembaruan-pembaruan yang terjadi pada kebijakan tidak akan membuat izin yang ada rusak.

Selain itu, AWS mendukung kebijakan-kebijakan terkelola untuk fungsi tugas yang mencakup beberapa layanan. Sebagai contoh, kebijakan `ReadOnlyAccess` terkelola AWS menyediakan akses hanya-baca ke semua layanan dan sumber daya AWS. Saat layanan meluncurkan fitur baru, AWS menambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk melihat daftar dan deskripsi dari kebijakan fungsi tugas, lihat kebijakan yang dikelola AWS untuk fungsi tugas di Panduan Pengguna IAM.

> **Important**
>
> Kami merekomendasikan agar Anda menggunakan kebijakan paling terbatas yang dapat mengizinkan Anda untuk melakukan kasus penggunaan Anda.

Kebijakan AWS terkelola berikut ini, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk Amazon SageMaker:

- **`AmazonSageMakerFullAccess`**— Memberikan akses penuh ke Amazon SageMaker dan sumber daya SageMaker geospasial dan operasi yang didukung. Ini tidak menyediakan akses Amazon S3 yang tidak terbatas, tetapi mendukung bucket dan objek dengan tag tertentu. `sagemaker` Kebijakan ini memungkinkan semua peran IAM diteruskan ke Amazon SageMaker, tetapi hanya mengizinkan peran IAM dengan "AmazonSageMaker" di dalamnya diteruskan keAWS Glue,AWS Step Functions, dan AWS RoboMaker layanan.
- **`AmazonSageMakerReadOnly`**— Memberikan akses hanya-baca ke sumber daya Amazon. SageMaker

Kebijakan AWS terkelola berikut dapat dilampirkan ke pengguna di akun Anda tetapi tidak disarankan:

- `AdministratorAccess`— Memberikan semua tindakan untuk semua AWS layanan dan untuk semua sumber daya di akun.
- `DataScientist`Memberikan berbagai izin untuk mencakup sebagian besar kasus penggunaan (terutama untuk analitik dan intelijen bisnis) yang dihadapi oleh ilmuwan data.

Anda dapat meninjau kebijakan izin ini dengan masuk ke konsol IAM dan mencarinya.

Anda juga dapat membuat kebijakan IAM khusus Anda sendiri untuk memberikan izin untuk SageMaker tindakan dan sumber daya Amazon saat Anda membutuhkannya. Anda dapat melampirkan kebijakan kustom ini ke pengguna atau grup yang memerlukannya.

Topik

- AWSkebijakan terkelola: AmazonSageMakerFullAccess
- AWSkebijakan terkelola: AmazonSageMakerReadOnly
- AWSkebijakan terkelola untuk Amazon SageMaker Canvas

- AWSkebijakan terkelola untuk Amazon SageMaker Cluster
- AWSkebijakan terkelola untuk Amazon SageMaker Feature Store
- AWSkebijakan terkelola untuk SageMaker geospasial Amazon
- AWSKebijakan Terkelola untuk Amazon SageMaker Ground Truth
- AWSKebijakan Terkelola untuk Tata Kelola SageMaker Model
- AWSKebijakan Terkelola untuk Registri Model
- AWSKebijakan Terkelola untuk SageMaker Notebook
- AWSKebijakan Terkelola untuk SageMaker Saluran Pipa
- AWSKebijakan Terkelola untuk SageMaker proyek dan JumpStart
- SageMaker Pembaruan Kebijakan AWS Terkelola

AWSkebijakan terkelola: AmazonSageMakerFullAccess

Kebijakan ini memberikan izin administratif yang memungkinkan akses penuh utama ke semua Amazon SageMaker dan sumber daya dan operasi SageMaker geospasial. Kebijakan ini juga menyediakan akses terpilih ke layanan terkait. Kebijakan ini memungkinkan semua peran IAM diteruskan ke Amazon SageMaker, tetapi hanya mengizinkan peran IAM dengan "AmazonSageMaker" di dalamnya diteruskan keAWS Glue,AWS Step Functions, dan AWS RoboMaker layanan. Kebijakan ini tidak menyertakan izin untuk membuat SageMaker domain Amazon. Untuk informasi tentang kebijakan yang diperlukan untuk membuat domain, lihatBuat Pengguna Administratif dan Grup .

Detail izin

Kebijakan ini mencakup izin berikut.

- `application-autoscaling`— Memungkinkan prinsipal untuk secara otomatis menskalakan titik akhir inferensi waktu SageMaker nyata.
- `athena`— Memungkinkan prinsipal untuk menanyakan daftar katalog data, database, dan metadata tabel dari. Amazon Athena
- `aws-marketplace`— Memungkinkan kepala sekolah untuk melihat langganan AI AWS Marketplace. Anda memerlukan ini jika Anda ingin mengakses SageMaker perangkat lunak berlangganan. AWS Marketplace
- `cloudformation`— Memungkinkan prinsipal untuk mendapatkan AWS CloudFormation template untuk menggunakan SageMaker JumpStart solusi dan Pipelines. SageMaker

JumpStartmenciptakan sumber daya yang diperlukan untuk menjalankan solusi pembelajaran end-to-end mesin yang SageMaker terkait dengan AWS layanan lain. SageMaker Pipelines membuat proyek baru yang didukung oleh Service Catalog.

- `cloudwatch`— Memungkinkan kepala sekolah untuk memposting CloudWatch metrik, berinteraksi dengan alarm, dan mengunggah log ke CloudWatch Log di akun Anda.
- `codebuild`— Memungkinkan kepala sekolah untuk menyimpan AWS CodeBuild artefak untuk Pipeline dan Proyek. SageMaker
- `codecommit`— Diperlukan untuk AWS CodeCommit integrasi dengan instance SageMaker notebook.
- `cognito-idp`— Diperlukan untuk Amazon SageMaker Ground Truth untuk mendefinisikan tenaga kerja pribadi dan tim kerja.
- `ec2`— Diperlukan SageMaker untuk mengelola sumber daya Amazon EC2 dan antarmuka jaringan saat Anda menentukan VPC Amazon untuk SageMaker pekerjaan, model, titik akhir, dan instans notebook Anda.
- `ecr`— Diperlukan untuk menarik dan menyimpan artefak Docker untuk Amazon SageMaker Studio Classic (gambar khusus), pelatihan, pemrosesan, inferensi batch, dan titik akhir inferensi. Ini juga diperlukan untuk menggunakan wadah Anda sendiri SageMaker. Izin tambahan untuk SageMaker JumpStart solusi diperlukan untuk membuat dan menghapus gambar khusus atas nama pengguna.
- `elastic-inference`— Memungkinkan prinsipal terhubung ke Amazon Elastic Inference untuk SageMaker menggunakan instans dan titik akhir notebook.
- `elasticfilesystem`— Memungkinkan kepala sekolah mengakses Amazon Elastic File System. Ini diperlukan SageMaker untuk menggunakan sumber data di Amazon Elastic File System untuk melatih model pembelajaran mesin.
- `fsx`— Memungkinkan kepala sekolah untuk mengakses Amazon FSx. Ini diperlukan SageMaker untuk menggunakan sumber data di Amazon FSx untuk melatih model pembelajaran mesin.
- `glue`— Diperlukan untuk pra-pemrosesan pipa inferensi dari dalam instance SageMaker notebook.
- `groundtruthlabeling`— Diperlukan untuk pekerjaan pelabelan Ground Truth. `groundtruthlabeling`Titik akhir diakses oleh konsol Ground Truth.
- `iam`— Diperlukan untuk memberikan akses SageMaker konsol ke peran IAM yang tersedia dan membuat peran terkait layanan.
- `kms`— Diperlukan untuk memberikan akses SageMaker konsol ke AWS KMS kunci yang tersedia dan mengambilnya untuk AWS KMS alias tertentu dalam pekerjaan dan titik akhir.
- `lambda`— Memungkinkan kepala sekolah untuk memanggil dan mendapatkan daftar fungsi. AWS Lambda

- `logs`— Diperlukan untuk memungkinkan SageMaker pekerjaan dan titik akhir untuk mempublikasikan aliran log.
- `redshift`— Memungkinkan kepala sekolah mengakses kredensyal cluster Amazon Redshift.
- `redshift-data`— Memungkinkan prinsipal menggunakan data dari Amazon Redshift untuk menjalankan, mendeskripsikan, dan membatalkan pernyataan; mendapatkan hasil pernyataan; dan daftar skema dan tabel.
- `robomaker`— Memungkinkan kepala sekolah memiliki akses penuh untuk membuat, mendapatkan deskripsi, dan menghapus aplikasi dan pekerjaan AWS RoboMaker simulasi. Ini juga diperlukan untuk menjalankan contoh pembelajaran penguatan pada instance notebook.
- `s3, s3express`- Memungkinkan kepala sekolah memiliki akses penuh ke sumber daya Amazon S3 dan Amazon S3 Express yang berkaitan dengan, SageMaker tetapi tidak semua Amazon S3 atau Amazon S3 Express.
- `sagemaker`— Memungkinkan kepala sekolah untuk mencantumkan tag pada profil SageMaker pengguna dan menambahkan tag ke aplikasi. SageMaker Mengizinkan akses hanya ke SageMaker definisi aliran sagemaker: WorkteamType “private-crowd” atau “vendor-crowd”.
- `sagemaker`dan `sagemaker-geospatial` — Memungkinkan kepala sekolah akses hanya-baca ke SageMaker domain dan profil pengguna.
- `secretsmanager`— Memungkinkan kepala sekolah untuk memiliki akses penuh ke. AWS Secrets Manager Prinsipal dapat mengenkripsi, menyimpan, dan mengambil kredensi untuk basis data dan layanan lainnya dengan aman. Ini juga diperlukan untuk instance SageMaker notebook dengan repositori SageMaker kode yang digunakan. GitHub
- `servicecatalog`— Memungkinkan kepala sekolah untuk menggunakan Service Catalog. Prinsipal dapat membuat, mendapatkan daftar, memperbarui, atau menghentikan produk yang disediakan, seperti server, database, situs web, atau aplikasi yang digunakan menggunakan sumber daya. AWS Ini diperlukan untuk SageMaker JumpStart dan Proyek untuk menemukan dan membaca produk katalog layanan dan meluncurkan AWS sumber daya pada pengguna.
- `sns`— Memungkinkan kepala sekolah untuk mendapatkan daftar topik Amazon SNS. Ini diperlukan untuk titik akhir dengan Inferensi Async diaktifkan untuk memberi tahu pengguna bahwa inferensi mereka telah selesai.
- `states`— Diperlukan untuk SageMaker JumpStart dan Pipelines untuk menggunakan katalog layanan untuk membuat sumber daya fungsi langkah.
- `tag`- Diperlukan untuk SageMaker Pipelines untuk dirender di Studio Classic. Studio Classic membutuhkan sumber daya yang ditandai dengan kunci `sagemaker:project-id` tag tertentu. Ini membutuhkan `tag:GetResources` izin.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllNonAdminSageMakerActions",
      "Effect": "Allow",
      "Action": [
        "sagemaker:*",
        "sagemaker-geospatial:*"
      ],
      "NotResource": [
        "arn:aws:sagemaker:*:*:domain/*",
        "arn:aws:sagemaker:*:*:user-profile/*",
        "arn:aws:sagemaker:*:*:app/*",
        "arn:aws:sagemaker:*:*:space/*",
        "arn:aws:sagemaker:*:*:flow-definition/*"
      ]
    },
    {
      "Sid": "AllowAddTagsForApp",
      "Effect": "Allow",
      "Action": [
        "sagemaker:AddTags"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:app/*"
      ]
    },
    {
      "Sid": "AllowStudioActions",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreatePresignedDomainUrl",
        "sagemaker:DescribeDomain",
        "sagemaker:ListDomains",
        "sagemaker:DescribeUserProfile",
        "sagemaker:ListUserProfiles",
        "sagemaker:DescribeSpace",
        "sagemaker:ListSpaces",
        "sagemaker:DescribeApp",
        "sagemaker:ListApps"
      ],
      "Resource": "*"
```

```
    },
    {
      "Sid": "AllowAppActionsForUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateApp",
        "sagemaker:DeleteApp"
      ],
      "Resource": "arn:aws:sagemaker:*:*:app/*/*/*/*",
      "Condition": {
        "Null": {
          "sagemaker:OwnerUserProfileArn": "true"
        }
      }
    },
    {
      "Sid": "AllowAppActionsForSharedSpaces",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateApp",
        "sagemaker:DeleteApp"
      ],
      "Resource": "arn:aws:sagemaker:*:*:app/${sagemaker:DomainId}/*/*/*",
      "Condition": {
        "StringEquals": {
          "sagemaker:SpaceSharingType": [
            "Shared"
          ]
        }
      }
    },
    {
      "Sid": "AllowMutatingActionsOnSharedSpacesWithoutOwner",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateSpace",
        "sagemaker:UpdateSpace",
        "sagemaker:DeleteSpace"
      ],
      "Resource": "arn:aws:sagemaker:*:*:space/${sagemaker:DomainId}/*",
      "Condition": {
        "Null": {
          "sagemaker:OwnerUserProfileArn": "true"
        }
```

```
      }
    },
    {
      "Sid": "RestrictMutatingActionsOnSpacesToOwnerUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateSpace",
        "sagemaker:UpdateSpace",
        "sagemaker:DeleteSpace"
      ],
      "Resource": "arn:aws:sagemaker:*:*:space/${sagemaker:DomainId}/*",
      "Condition": {
        "ArnLike": {
          "sagemaker:OwnerUserProfileArn": "arn:aws:sagemaker:*:*:user-profile/
${sagemaker:DomainId}/${sagemaker:UserProfileName}"
        },
        "StringEquals": {
          "sagemaker:SpaceSharingType": [
            "Private",
            "Shared"
          ]
        }
      }
    },
    {
      "Sid": "RestrictMutatingActionsOnPrivateSpaceAppsToOwnerUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateApp",
        "sagemaker:DeleteApp"
      ],
      "Resource": "arn:aws:sagemaker:*:*:app/${sagemaker:DomainId}/*/*/*",
      "Condition": {
        "ArnLike": {
          "sagemaker:OwnerUserProfileArn": "arn:aws:sagemaker:*:*:user-profile/
${sagemaker:DomainId}/${sagemaker:UserProfileName}"
        },
        "StringEquals": {
          "sagemaker:SpaceSharingType": [
            "Private"
          ]
        }
      }
    },
```

```
    {
      "Sid": "AllowFlowDefinitionActions",
      "Effect": "Allow",
      "Action": "sagemaker:*",
      "Resource": [
        "arn:aws:sagemaker:*:*:flow-definition/*"
      ],
      "Condition": {
        "StringEqualsIfExists": {
          "sagemaker:WorkteamType": [
            "private-crowd",
            "vendor-crowd"
          ]
        }
      }
    },
    {
      "Sid": "AllowAWSServiceActions",
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:DeleteScalingPolicy",
        "application-autoscaling:DeleteScheduledAction",
        "application-autoscaling:DeregisterScalableTarget",
        "application-autoscaling:DescribeScalableTargets",
        "application-autoscaling:DescribeScalingActivities",
        "application-autoscaling:DescribeScalingPolicies",
        "application-autoscaling:DescribeScheduledActions",
        "application-autoscaling:PutScalingPolicy",
        "application-autoscaling:PutScheduledAction",
        "application-autoscaling:RegisterScalableTarget",
        "aws-marketplace:ViewSubscriptions",
        "cloudformation:GetTemplateSummary",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:PutMetricData",
        "codecommit:BatchGetRepositories",
        "codecommit:CreateRepository",
        "codecommit:GetRepository",
        "codecommit:List*",
        "cognito-idp:AdminAddUserToGroup",
```

```
        "cognito-idp:AdminCreateUser",
        "cognito-idp:AdminDeleteUser",
        "cognito-idp:AdminDisableUser",
        "cognito-idp:AdminEnableUser",
        "cognito-idp:AdminRemoveUserFromGroup",
        "cognito-idp:CreateGroup",
        "cognito-idp:CreateUserPool",
        "cognito-idp:CreateUserPoolClient",
        "cognito-idp:CreateUserPoolDomain",
        "cognito-idp:DescribeUserPool",
        "cognito-idp:DescribeUserPoolClient",
        "cognito-idp:List*",
        "cognito-idp:UpdateUserPool",
        "cognito-idp:UpdateUserPoolClient",
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:CreateVpcEndpoint",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcs",
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:CreateRepository",
        "ecr:Describe*",
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer",
        "ecr:StartImageScan",
        "elastic-inference:Connect",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:DescribeMountTargets",
        "fsx:DescribeFileSystems",
        "glue:CreateJob",
        "glue:DeleteJob",
        "glue:GetJob*",
        "glue:GetTable*",
        "glue:GetWorkflowRun",
        "glue:ResetJobBookmark",
        "glue:StartJobRun",
```

```
        "glue:StartWorkflowRun",
        "glue:UpdateJob",
        "groundtruthlabeling:*",
        "iam:ListRoles",
        "kms:DescribeKey",
        "kms:ListAliases",
        "lambda:ListFunctions",
        "logs:CreateLogDelivery",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogDelivery",
        "logs:Describe*",
        "logs:GetLogDelivery",
        "logs:GetLogEvents",
        "logs:ListLogDeliveries",
        "logs:PutLogEvents",
        "logs:PutResourcePolicy",
        "logs:UpdateLogDelivery",
        "robomaker:CreateSimulationApplication",
        "robomaker:DescribeSimulationApplication",
        "robomaker:DeleteSimulationApplication",
        "robomaker:CreateSimulationJob",
        "robomaker:DescribeSimulationJob",
        "robomaker:CancelSimulationJob",
        "secretsmanager:ListSecrets",
        "servicecatalog:Describe*",
        "servicecatalog:List*",
        "servicecatalog:ScanProvisionedProducts",
        "servicecatalog:SearchProducts",
        "servicecatalog:SearchProvisionedProducts",
        "sns:ListTopics",
        "tag:GetResources"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowECRActions",
      "Effect": "Allow",
      "Action": [
        "ecr:SetRepositoryPolicy",
        "ecr:CompleteLayerUpload",
        "ecr:BatchDeleteImage",
        "ecr:UploadLayerPart",
        "ecr:DeleteRepositoryPolicy",
```

```
        "ecr:InitiateLayerUpload",
        "ecr:DeleteRepository",
        "ecr:PutImage"
      ],
      "Resource": [
        "arn:aws:ecr:*:*:repository/*sagemaker*"
      ]
    },
    {
      "Sid": "AllowCodeCommitActions",
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPull",
        "codecommit:GitPush"
      ],
      "Resource": [
        "arn:aws:codecommit:*:*:*sagemaker*",
        "arn:aws:codecommit:*:*:*SageMaker*",
        "arn:aws:codecommit:*:*:*Sagemaker*"
      ]
    },
    {
      "Sid": "AllowCodeBuildActions",
      "Action": [
        "codebuild:BatchGetBuilds",
        "codebuild:StartBuild"
      ],
      "Resource": [
        "arn:aws:codebuild:*:*:project/sagemaker*",
        "arn:aws:codebuild:*:*:build/*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "AllowStepFunctionsActions",
      "Action": [
        "states:DescribeExecution",
        "states:GetExecutionHistory",
        "states:StartExecution",
        "states:StopExecution",
        "states:UpdateStateMachine"
      ],
      "Resource": [
        "arn:aws:states:*:*:statemachine:*sagemaker*",
```

```
        "arn:aws:states:*:*:execution:*sagemaker*:*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "AllowSecretManagerActions",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:CreateSecret"
      ],
      "Resource": [
        "arn:aws:secretsmanager:*:*:secret:AmazonSageMaker-*"
      ]
    },
    {
      "Sid": "AllowReadOnlySecretManagerActions",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "secretsmanager:ResourceTag/SageMaker": "true"
        }
      }
    },
    {
      "Sid": "AllowServiceCatalogProvisionProduct",
      "Effect": "Allow",
      "Action": [
        "servicecatalog:ProvisionProduct"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowServiceCatalogTerminateUpdateProvisionProduct",
      "Effect": "Allow",
      "Action": [
        "servicecatalog:TerminateProvisionedProduct",
        "servicecatalog:UpdateProvisionedProduct"
```

```
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "servicecatalog:userLevel": "self"
        }
      }
    },
    {
      "Sid": "AllowS3ObjectActions",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
        "arn:aws:s3:::*SageMaker*",
        "arn:aws:s3:::*Sagemaker*",
        "arn:aws:s3:::*sagemaker*",
        "arn:aws:s3:::*aws-glue*"
      ]
    },
    {
      "Sid": "AllowS3GetObjectWithSageMakerExistingObjectTag",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Condition": {
        "StringEqualsIgnoreCase": {
          "s3:ExistingObjectTag/SageMaker": "true"
        }
      }
    },
    {
      "Sid": "AllowS3GetObjectWithServiceCatalogProvisioningExistingObjectTag",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
```

```
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
        }
      }
    },
    {
      "Sid": "AllowS3BucketActions",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets",
        "s3:GetBucketCors",
        "s3:PutBucketCors"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowS3BucketACL",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::*SageMaker*",
        "arn:aws:s3:::*Sagemaker*",
        "arn:aws:s3:::*sagemaker*"
      ]
    },
    {
      "Sid": "AllowLambdaInvokeFunction",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:*:*:function:*SageMaker*",
```

```
        "arn:aws:lambda:*:*:function:*sagemaker*",
        "arn:aws:lambda:*:*:function:*Sagemaker*",
        "arn:aws:lambda:*:*:function:*LabelingFunction*"
      ]
    },
    {
      "Sid": "AllowCreateServiceLinkedRoleForSageMakerApplicationAutoscaling",
      "Action": "iam:CreateServiceLinkedRole",
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/aws-service-role/sagemaker.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "sagemaker.application-autoscaling.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowCreateServiceLinkedRoleForRobomaker",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "robomaker.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowSNSActions",
      "Effect": "Allow",
      "Action": [
        "sns:Subscribe",
        "sns:CreateTopic",
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:*:*:*SageMaker*",
        "arn:aws:sns:*:*:*Sagemaker*",
        "arn:aws:sns:*:*:*sagemaker*"
      ]
    },
    {
      "Sid": "AllowPassRoleForSageMakerRoles",
```

```
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/*AmazonSageMaker*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "glue.amazonaws.com",
            "robomaker.amazonaws.com",
            "states.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowPassRoleToSageMaker",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "sagemaker.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowAthenaActions",
      "Effect": "Allow",
      "Action": [
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:ListTableMetadata",
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution"
      ],
      "Resource": [
        "*"
      ]
    },
```

```
    {
      "Sid": "AllowGlueCreateTable",
      "Effect": "Allow",
      "Action": [
        "glue:CreateTable"
      ],
      "Resource": [
        "arn:aws:glue:*:*:table/*/sagemaker_tmp_*",
        "arn:aws:glue:*:*:table/sagemaker_featurestore/*",
        "arn:aws:glue:*:*:catalog",
        "arn:aws:glue:*:*:database/*"
      ]
    },
    {
      "Sid": "AllowGlueUpdateTable",
      "Effect": "Allow",
      "Action": [
        "glue:UpdateTable"
      ],
      "Resource": [
        "arn:aws:glue:*:*:table/sagemaker_featurestore/*",
        "arn:aws:glue:*:*:catalog",
        "arn:aws:glue:*:*:database/sagemaker_featurestore"
      ]
    },
    {
      "Sid": "AllowGlueDeleteTable",
      "Effect": "Allow",
      "Action": [
        "glue:DeleteTable"
      ],
      "Resource": [
        "arn:aws:glue:*:*:table/*/sagemaker_tmp_*",
        "arn:aws:glue:*:*:catalog",
        "arn:aws:glue:*:*:database/*"
      ]
    },
    {
      "Sid": "AllowGlueGetTablesAndDatabases",
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabases",
        "glue:GetTable",
        "glue:GetTables"
```

```
      ],
      "Resource": [
        "arn:aws:glue:*:*:table/*",
        "arn:aws:glue:*:*:catalog",
        "arn:aws:glue:*:*:database/*"
      ]
    },
    {
      "Sid": "AllowGlueGetAndCreateDatabase",
      "Effect": "Allow",
      "Action": [
        "glue:CreateDatabase",
        "glue:GetDatabase"
      ],
      "Resource": [
        "arn:aws:glue:*:*:catalog",
        "arn:aws:glue:*:*:database/sagemaker_featurestore",
        "arn:aws:glue:*:*:database/sagemaker_processing",
        "arn:aws:glue:*:*:database/default",
        "arn:aws:glue:*:*:database/sagemaker_data_wrangler"
      ]
    },
    {
      "Sid": "AllowRedshiftDataActions",
      "Effect": "Allow",
      "Action": [
        "redshift-data:ExecuteStatement",
        "redshift-data:DescribeStatement",
        "redshift-data:CancelStatement",
        "redshift-data:GetStatementResult",
        "redshift-data:ListSchemas",
        "redshift-data:ListTables"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "AllowRedshiftGetClusterCredentials",
      "Effect": "Allow",
      "Action": [
        "redshift:GetClusterCredentials"
      ],
      "Resource": [
```

```
        "arn:aws:redshift:*:*:dbuser:*/sagemaker_access*",
        "arn:aws:redshift:*:*:dbname:*"
      ]
    },
    {
      "Sid": "AllowListTagsForUserProfile",
      "Effect": "Allow",
      "Action": [
        "sagemaker:ListTags"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:user-profile/*"
      ]
    },
    {
      "Sid": "AllowCloudformationListStackResources",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ListStackResources"
      ],
      "Resource": "arn:aws:cloudformation:*:*:stack/SC-*"
    },
    {
      "Sid": "AllowS3ExpressObjectActions",
      "Effect": "Allow",
      "Action": [
        "s3express:CreateSession"
      ],
      "Resource": [
        "arn:aws:s3express:*:*:bucket/*SageMaker*",
        "arn:aws:s3express:*:*:bucket/*Sagemaker*",
        "arn:aws:s3express:*:*:bucket/*sagemaker*",
        "arn:aws:s3express:*:*:bucket/*aws-glue*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowS3ExpressCreateBucketActions",
      "Effect": "Allow",
      "Action": [
```

```
        "s3express:CreateBucket"
      ],
      "Resource": [
        "arn:aws:s3express:*:*:bucket/*SageMaker*",
        "arn:aws:s3express:*:*:bucket/*Sagemaker*",
        "arn:aws:s3express:*:*:bucket/*sagemaker*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowS3ExpressListBucketActions",
      "Effect": "Allow",
      "Action": [
        "s3express:ListAllMyDirectoryBuckets"
      ],
      "Resource": "*"
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerReadOnly

Kebijakan ini memberikan akses hanya-baca ke Amazon SageMaker melalui dan SDKAWS Management Console.

Detail izin

Kebijakan ini mencakup izin berikut.

- `application-autoscaling`— Memungkinkan pengguna untuk menelusuri deskripsi titik akhir inferensi SageMaker real-time yang dapat diskalakan.
- `aws-marketplace`— Memungkinkan pengguna untuk melihat langganan AWS AI Marketplace.
- `cloudwatch`— Memungkinkan pengguna untuk menerima CloudWatch alarm.
- `cognito-idp`— Diperlukan untuk Amazon SageMaker Ground Truth untuk menelusuri deskripsi dan daftar tenaga kerja pribadi dan tim kerja.
- `ecr`— Diperlukan untuk membaca artefak Docker untuk pelatihan dan inferensi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:Describe*",
                "sagemaker:List*",
                "sagemaker:BatchGetMetrics",
                "sagemaker:GetDeviceRegistration",
                "sagemaker:GetDeviceFleetReport",
                "sagemaker:GetSearchSuggestions",
                "sagemaker:BatchGetRecord",
                "sagemaker:GetRecord",
                "sagemaker:Search",
                "sagemaker:QueryLineage",
                "sagemaker:GetLineageGroupPolicy",
                "sagemaker:BatchDescribeModelPackage",
                "sagemaker:GetModelPackageGroupPolicy"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:DescribeScheduledActions",
                "aws-marketplace:ViewSubscriptions",
                "cloudwatch:DescribeAlarms",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolClient",
                "cognito-idp:ListGroups",
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:ListUsers",
                "cognito-idp:ListUsersInGroup",
                "ecr:Describe*"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

AWSkebijakan terkelola untuk Amazon SageMaker Canvas

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan Amazon SageMaker Canvas. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

- AWSkebijakan terkelola: AmazonSageMakerCanvasFullAccess
- AWSkebijakan terkelola: AmazonSageMakerCanvasDataPrepFull Akses
- AWSkebijakan terkelola: AmazonSageMakerCanvasDirectDeployAccess
- AWSkebijakan terkelola: AmazonSageMakerCanvas AI ServicesAccess
- AWSkebijakan terkelola: AmazonSageMakerCanvasBedrockAccess
- AWSkebijakan terkelola: AmazonSageMakerCanvasForecastAccess
- SageMaker Pembaruan Amazon ke kebijakan terkelola Amazon SageMaker Canvas

AWSkebijakan terkelola: AmazonSageMakerCanvasFullAccess

Kebijakan ini memberikan izin yang memungkinkan akses penuh ke Amazon SageMaker Canvas melalui AWS Management Console dan SDK. Kebijakan ini juga menyediakan akses pilih ke layanan terkait [misalnya, Amazon Simple Storage Service (Amazon S3), (IAM), Amazon Virtual Private Cloud (Amazon VPC)AWS Identity and Access Management, Amazon Elastic Container Registry (Amazon ECR), Amazon Elastic Container Registry (Amazon ECR), Amazon CloudWatch Logs, Amazon Redshift, Amazon Autopilot, Registry Model, dan Amazon AWS Secrets Manager Forecast]. SageMaker SageMaker

Kebijakan ini dimaksudkan untuk membantu pelanggan bereksperimen dan memulai dengan semua kemampuan SageMaker Canvas. Untuk kontrol yang lebih halus, kami menyarankan pelanggan membuat versi cakupan mereka sendiri saat mereka beralih ke beban kerja produksi. Untuk informasi selengkapnya, lihat Jenis kebijakan IAM: Bagaimana dan kapan menggunakannya.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `sagemaker`— Memungkinkan prinsipal untuk membuat dan meng-host SageMaker model pada sumber daya yang ARN-nya berisi “Kanvas”, “kanvas”, atau “model-kompilasi-”. Selain itu, pengguna dapat mendaftarkan model SageMaker Canvas mereka ke SageMaker Model Registry di AWS akun yang sama.
- `ec2`— Memungkinkan prinsipal untuk membuat titik akhir Amazon VPC.
- `ecr`— Mengizinkan prinsipal untuk mendapatkan informasi tentang citra kontainer.
- `glue`— Memungkinkan kepala sekolah untuk mengambil tabel dalam katalog.
- `iam`— Memungkinkan kepala sekolah untuk meneruskan peran IAM ke Amazon dan Amazon SageMaker Forecast. Juga mengizinkan prinsipal untuk membuat peran yang terhubung dengan layanan.
- `logs`— Memungkinkan kepala sekolah untuk mempublikasikan log dari pekerjaan pelatihan dan titik akhir.
- `s3`— Mengizinkan prinsipal menambahkan dan mengambil objek dari bucket Amazon S3. Benda-benda ini terbatas pada mereka yang namanya termasuk "SageMaker“, “Sagemaker”, atau “sagemaker”. Juga memungkinkan prinsipal untuk mengambil objek dari ember Amazon S3 yang ARN-nya dimulai dengan "-” di wilayah tertentu. jumpstart-cache-prod
- `secretsmanager`— Memungkinkan kepala sekolah untuk menyimpan kredensil pelanggan untuk terhubung ke database Snowflake menggunakan Secrets Manager.
- `redshift`— Memungkinkan prinsipal untuk mendapatkan kredensil untuk dbuser “sagemaker_access*” di klaster Amazon Redshift mana pun jika pengguna itu ada.
- `redshift-data`— Memungkinkan prinsipal menjalankan kueri di Amazon Redshift menggunakan Amazon Redshift Data API. Ini hanya menyediakan akses ke Redshift Data API itu sendiri dan tidak secara langsung menyediakan akses ke cluster Amazon Redshift Anda. Untuk informasi selengkapnya, lihat Menggunakan API Data Amazon Redshift.
- `forecast`— Memungkinkan prinsipal untuk menggunakan Amazon Forecast.
- `application-autoscaling`— Memungkinkan prinsipal untuk secara otomatis menskalakan titik akhir inferensi. SageMaker
- `rds`— Memungkinkan kepala sekolah mengembalikan informasi tentang instans Amazon RDS yang disediakan.
- `cloudwatch`— Memungkinkan kepala sekolah untuk membuat dan mengelola alarm Amazon. CloudWatch
- `athena`— Memungkinkan kepala sekolah untuk membuat, membaca, dan mengelola kueri, katalog, dan eksekusi Amazon Athena.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SageMakerUserDetailsAndPackageOperations",
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeDomain",
                "sagemaker:DescribeUserProfile",
                "sagemaker:ListTags",
                "sagemaker:ListModelPackages",
                "sagemaker:ListModelPackageGroups",
                "sagemaker:ListEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SageMakerPackageGroupOperations",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateModelPackageGroup",
                "sagemaker:CreateModelPackage",
                "sagemaker:DescribeModelPackageGroup",
                "sagemaker:DescribeModelPackage"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:model-package/*",
                "arn:aws:sagemaker:*:*:model-package-group/*"
            ]
        },
        {
            "Sid": "SageMakerTrainingOperations",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateCompilationJob",
                "sagemaker:CreateEndpoint",
                "sagemaker:CreateEndpointConfig",
                "sagemaker:CreateModel",
                "sagemaker:CreateProcessingJob",
                "sagemaker:CreateAutoMLJob",
                "sagemaker:CreateAutoMLJobV2",
                "sagemaker:DeleteEndpoint",
                "sagemaker:DescribeCompilationJob",
```

```
                "sagemaker:DescribeEndpoint",
                "sagemaker:DescribeEndpointConfig",
                "sagemaker:DescribeModel",
                "sagemaker:DescribeProcessingJob",
                "sagemaker:DescribeAutoMLJob",
                "sagemaker:DescribeAutoMLJobV2",
                "sagemaker:ListCandidatesForAutoMLJob",
                "sagemaker:AddTags",
                "sagemaker:DeleteApp"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:*Canvas*",
                "arn:aws:sagemaker:*:*:*canvas*",
                "arn:aws:sagemaker:*:*:*model-compilation-*"
            ]
        },
        {
            "Sid": "SageMakerHostingOperations",
            "Effect": "Allow",
            "Action": [
                "sagemaker:DeleteEndpointConfig",
                "sagemaker:DeleteModel",
                "sagemaker:InvokeEndpoint",
                "sagemaker:UpdateEndpointWeightsAndCapacities",
                "sagemaker:InvokeEndpointAsync"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:*Canvas*",
                "arn:aws:sagemaker:*:*:*canvas*"
            ]
        },
        {
            "Sid": "EC2VPCOperation",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcEndpointServices"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "ECROperations",
            "Effect": "Allow",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMGetOperations",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Sid": "IAMPassOperation",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "sagemaker.amazonaws.com"
                }
            }
        },
        {
            "Sid": "LoggingOperation",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/sagemaker/*"
        },
        {
            "Sid": "S3Operations",
            "Effect": "Allow",
```

```
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:CreateBucket",
                "s3:GetBucketCors",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
            "Sid": "ReadSageMakerJumpstartArtifacts",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::jumpstart-cache-prod-us-west-2/*",
                "arn:aws:s3:::jumpstart-cache-prod-us-east-1/*",
                "arn:aws:s3:::jumpstart-cache-prod-us-east-2/*",
                "arn:aws:s3:::jumpstart-cache-prod-eu-west-1/*",
                "arn:aws:s3:::jumpstart-cache-prod-eu-central-1/*",
                "arn:aws:s3:::jumpstart-cache-prod-ap-south-1/*",
                "arn:aws:s3:::jumpstart-cache-prod-ap-northeast-2/*",
                "arn:aws:s3:::jumpstart-cache-prod-ap-northeast-1/*",
                "arn:aws:s3:::jumpstart-cache-prod-ap-southeast-1/*",
                "arn:aws:s3:::jumpstart-cache-prod-ap-southeast-2/*"
            ]
        },
        {
            "Sid": "S3ListOperations",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "GlueOperations",
            "Effect": "Allow",
            "Action": "glue:SearchTables",
```

```
            "Resource": [
                "arn:aws:glue:*:*:table/*/*",
                "arn:aws:glue:*:*:database/*",
                "arn:aws:glue:*:*:catalog"
            ]
        },
        {
            "Sid": "SecretsManagerARNBasedOperation",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:CreateSecret",
                "secretsmanager:PutResourcePolicy"
            ],
            "Resource": [
                "arn:aws:secretsmanager:*:*:secret:AmazonSageMaker-*"
            ]
        },
        {
            "Sid": "SecretManagerTagBasedOperation",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:ResourceTag/SageMaker": "true"
                }
            }
        },
        {
            "Sid": "RedshiftOperations",
            "Effect": "Allow",
            "Action": [
                "redshift-data:ExecuteStatement",
                "redshift-data:DescribeStatement",
                "redshift-data:CancelStatement",
                "redshift-data:GetStatementResult",
                "redshift-data:ListSchemas",
                "redshift-data:ListTables",
                "redshift-data:DescribeTable"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "RedshiftGetCredentialsOperation",
            "Effect": "Allow",
            "Action": [
                "redshift:GetClusterCredentials"
            ],
            "Resource": [
                "arn:aws:redshift:*:*:dbuser:*/sagemaker_access*",
                "arn:aws:redshift:*:*:dbname:*"
            ]
        },
        {
            "Sid": "ForecastOperations",
            "Effect": "Allow",
            "Action": [
                "forecast:CreateExplainabilityExport",
                "forecast:CreateExplainability",
                "forecast:CreateForecastEndpoint",
                "forecast:CreateAutoPredictor",
                "forecast:CreateDatasetImportJob",
                "forecast:CreateDatasetGroup",
                "forecast:CreateDataset",
                "forecast:CreateForecast",
                "forecast:CreateForecastExportJob",
                "forecast:CreatePredictorBacktestExportJob",
                "forecast:CreatePredictor",
                "forecast:DescribeExplainabilityExport",
                "forecast:DescribeExplainability",
                "forecast:DescribeAutoPredictor",
                "forecast:DescribeForecastEndpoint",
                "forecast:DescribeDatasetImportJob",
                "forecast:DescribeDataset",
                "forecast:DescribeForecast",
                "forecast:DescribeForecastExportJob",
                "forecast:DescribePredictorBacktestExportJob",
                "forecast:GetAccuracyMetrics",
                "forecast:InvokeForecastEndpoint",
                "forecast:GetRecentForecastContext",
                "forecast:DescribePredictor",
                "forecast:TagResource",
                "forecast:DeleteResourceTree"
```

```
            ],
            "Resource": [
                "arn:aws:forecast:*:*:*Canvas*"
            ]
        },
        {
            "Sid": "RDSOperation",
            "Effect": "Allow",
            "Action": "rds:DescribeDBInstances",
            "Resource": "*"
        },
        {
            "Sid": "IAMPassOperationForForecast",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "forecast.amazonaws.com"
                }
            }
        },
        {
            "Sid": "AutoscalingOperations",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:RegisterScalableTarget"
            ],
            "Resource": "arn:aws:application-autoscaling:*:*:scalable-target/*",
            "Condition": {
                "StringEquals": {
                    "application-autoscaling:service-namespace": "sagemaker",
                    "application-autoscaling:scalable-dimension":
"sagemaker:variant:DesiredInstanceCount"
                }
            }
        },
        {
            "Sid": "AsyncEndpointOperations",
            "Effect": "Allow",
            "Action": [
```

```
                "cloudwatch:DescribeAlarms",
                "sagemaker:DescribeEndpointConfig"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SageMakerCloudWatchUpdate",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:TargetTracking*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaLast": "application-autoscaling.amazonaws.com"
                }
            }
        },
        {
            "Sid": "AutoscalingSageMakerEndpointOperation",
            "Action": "iam:CreateServiceLinkedRole",
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/aws-service-role/sagemaker.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_SageMakerEndpoint",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "sagemaker.application-
autoscaling.amazonaws.com"
                }
            }
        }
    ]
}
```

AWSkebijakan terkelola: AmazonSageMakerCanvasDataPrepFull Akses

Kebijakan ini memberikan izin yang memungkinkan akses penuh ke fungsionalitas persiapan data Amazon SageMaker Canvas. Kebijakan ini juga memberikan izin hak istimewa paling sedikit untuk layanan yang terintegrasi dengan fungsionalitas persiapan data [misalnya, Amazon Simple Storage Service (Amazon S3) Service (IAM), Amazon EMR, Amazon, Amazon Redshift, AWS Identity and

Access Management () dan]. EventBridge AWS Key Management Service AWS KMS AWS Secrets Manager

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `sagemaker`— Memungkinkan kepala sekolah untuk mengakses pekerjaan pemrosesan, pekerjaan pelatihan, saluran inferensi, pekerjaan AutoML, dan grup fitur.
- `athena`— Memungkinkan kepala sekolah untuk menanyakan daftar katalog data, database, dan metadata tabel dari Amazon Athena.
- `elasticmapreduce`— Memungkinkan kepala sekolah untuk membaca dan membuat daftar cluster EMR Amazon.
- `events`— Memungkinkan kepala sekolah untuk membuat, membaca, memperbarui, dan menambahkan target ke EventBridge aturan Amazon untuk pekerjaan terjadwal.
- `glue`— Memungkinkan kepala sekolah untuk mendapatkan dan mencari tabel dari database dalam katalog. AWS Glue
- `iam`— Memungkinkan kepala sekolah untuk meneruskan peran IAM ke Amazon dan. SageMaker EventBridge
- `kms`— Memungkinkan prinsipal untuk mengambil AWS KMS alias yang disimpan dalam pekerjaan dan titik akhir, dan mengakses kunci KMS terkait.
- `logs`— Memungkinkan kepala sekolah untuk mempublikasikan log dari pekerjaan pelatihan dan titik akhir.
- `redshift`— Memungkinkan kepala sekolah mendapatkan kredensyal untuk mengakses database Amazon Redshift.
- `redshift-data`— Memungkinkan prinsipal untuk menjalankan, membatalkan, mendeskripsikan, membuat daftar, dan mendapatkan hasil kueri Amazon Redshift. Juga memungkinkan kepala sekolah untuk membuat daftar skema dan tabel Amazon Redshift.
- `s3`— Mengizinkan prinsipal menambahkan dan mengambil objek dari bucket Amazon S3. Objek-objek ini terbatas pada mereka yang namanya termasuk "SageMaker“, “Sagemaker”, atau “sagemaker”; atau ditandai dengan ""“, case-insensitive. SageMaker
- `secretsmanager`— Memungkinkan kepala sekolah untuk menyimpan dan mengambil kredenal basis data pelanggan menggunakan Secrets Manager.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SageMakerListFeatureGroupOperation",
            "Effect": "Allow",
            "Action": "sagemaker:ListFeatureGroups",
            "Resource": "*"
        },
        {
            "Sid": "SageMakerFeatureGroupOperations",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateFeatureGroup",
                "sagemaker:DescribeFeatureGroup"
            ],
            "Resource": "arn:aws:sagemaker:*:*:feature-group/*"
        },
        {
            "Sid": "SageMakerProcessingJobOperations",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateProcessingJob",
                "sagemaker:DescribeProcessingJob",
                "sagemaker:AddTags"
            ],
            "Resource": "arn:aws:sagemaker:*:*:processing-job/*canvas-data-prep*"
        },
        {
            "Sid": "SageMakerProcessingJobListOperation",
            "Effect": "Allow",
            "Action": "sagemaker:ListProcessingJobs",
            "Resource": "*"
        },
        {
            "Sid": "SageMakerPipelineOperations",
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribePipeline",
                "sagemaker:CreatePipeline",
                "sagemaker:UpdatePipeline",
                "sagemaker:DeletePipeline",
                "sagemaker:StartPipelineExecution",
                "sagemaker:ListPipelineExecutionSteps",
                "sagemaker:DescribePipelineExecution"
```

```
            ],
            "Resource": "arn:aws:sagemaker:*:*:pipeline/*canvas-data-prep*"
        },
        {
            "Sid": "KMSListOperations",
            "Effect": "Allow",
            "Action": "kms:ListAliases",
            "Resource": "*"
        },
        {
            "Sid": "KMSOperations",
            "Effect": "Allow",
            "Action": "kms:DescribeKey",
            "Resource": "arn:aws:kms:*:*:key/*"
        },
        {
            "Sid": "S3Operations",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:GetBucketCors",
                "s3:GetBucketLocation",
                "s3:AbortMultipartUpload"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "S3GetObjectOperation",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::*",
            "Condition": {
                "StringEqualsIgnoreCase": {
```

```
                    "s3:ExistingObjectTag/SageMaker": "true"
                },
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "S3ListOperations",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMListOperations",
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "*"
        },
        {
            "Sid": "IAMGetOperations",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Sid": "IAMPassOperation",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "sagemaker.amazonaws.com",
                        "events.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "EventBridgePutOperation",
```

```
            "Effect": "Allow",
            "Action": [
                "events:PutRule"
            ],
            "Resource": "arn:aws:events:*:*:rule/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/sagemaker:is-canvas-data-prep-job": "true"
                }
            }
        },
        {
            "Sid": "EventBridgeOperations",
            "Effect": "Allow",
            "Action": [
                "events:DescribeRule",
                "events:PutTargets"
            ],
            "Resource": "arn:aws:events:*:*:rule/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/sagemaker:is-canvas-data-prep-job": "true"
                }
            }
        },
        {
            "Sid": "EventBridgeTagBasedOperations",
            "Effect": "Allow",
            "Action": [
                "events:TagResource"
            ],
            "Resource": "arn:aws:events:*:*:rule/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/sagemaker:is-canvas-data-prep-job": "true",
                    "aws:ResourceTag/sagemaker:is-canvas-data-prep-job": "true"
                }
            }
        },
        {
            "Sid": "EventBridgeListTagOperation",
            "Effect": "Allow",
            "Action": "events:ListTagsForResource",
            "Resource": "*"
```

```
        },
        {
            "Sid": "GlueOperations",
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabases",
                "glue:GetTable",
                "glue:GetTables",
                "glue:SearchTables"
            ],
            "Resource": [
                "arn:aws:glue:*:*:table/*",
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*"
            ]
        },
        {
            "Sid": "EMROperations",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups"
            ],
            "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*"
        },
        {
            "Sid": "EMRListOperation",
            "Effect": "Allow",
            "Action": "elasticmapreduce:ListClusters",
            "Resource": "*"
        },
        {
            "Sid": "AthenaListDataCatalogOperation",
            "Effect": "Allow",
            "Action": "athena:ListDataCatalogs",
            "Resource": "*"
        },
        {
            "Sid": "AthenaQueryExecutionOperations",
            "Effect": "Allow",
            "Action": [
                "athena:GetQueryExecution",
                "athena:GetQueryResults",
                "athena:StartQueryExecution",
```

```
                "athena:StopQueryExecution"
            ],
            "Resource": "arn:aws:athena:*:*:workgroup/*"
        },
        {
            "Sid": "AthenaDataCatalogOperations",
            "Effect": "Allow",
            "Action": [
                "athena:ListDatabases",
                "athena:ListTableMetadata"
            ],
            "Resource": "arn:aws:athena:*:*:datacatalog/*"
        },
        {
            "Sid": "RedshiftOperations",
            "Effect": "Allow",
            "Action": [
                "redshift-data:DescribeStatement",
                "redshift-data:CancelStatement",
                "redshift-data:GetStatementResult"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RedshiftArnBasedOperations",
            "Effect": "Allow",
            "Action": [
                "redshift-data:ExecuteStatement",
                "redshift-data:ListSchemas",
                "redshift-data:ListTables"
            ],
            "Resource": "arn:aws:redshift:*:*:cluster:*"
        },
        {
            "Sid": "RedshiftGetCredentialsOperation",
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentials",
            "Resource": [
                "arn:aws:redshift:*:*:dbuser:*/sagemaker_access*",
                "arn:aws:redshift:*:*:dbname:*"
            ]
        },
        {
            "Sid": "SecretsManagerARNBasedOperation",
```

```
            "Effect": "Allow",
            "Action": "secretsmanager:CreateSecret",
            "Resource": "arn:aws:secretsmanager:*:*:secret:AmazonSageMaker-*"
        },
        {
            "Sid": "SecretManagerTagBasedOperation",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:AmazonSageMaker-*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SageMaker": "true",
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "RDSOperation",
            "Effect": "Allow",
            "Action": "rds:DescribeDBInstances",
            "Resource": "*"
        },
        {
            "Sid": "LoggingOperation",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/sagemaker/studio:*"
        }
    ]
}
```

AWSkebijakan terkelola: AmazonSageMakerCanvasDirectDeployAccess

Kebijakan ini memberikan izin yang diperlukan Amazon SageMaker Canvas untuk membuat dan mengelola titik akhir Amazon SageMaker.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `sagemaker`— Memungkinkan prinsipal untuk membuat dan mengelola titik SageMaker akhir dengan nama sumber daya ARN yang dimulai dengan “Kanvas” atau “kanvas”.
- `cloudwatch`— Memungkinkan kepala sekolah untuk mengambil data metrik Amazon. CloudWatch

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SageMakerEndpointPerms",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateEndpoint",
                "sagemaker:CreateEndpointConfig",
                "sagemaker:DeleteEndpoint",
                "sagemaker:DescribeEndpoint",
                "sagemaker:DescribeEndpointConfig",
                "sagemaker:InvokeEndpoint",
                "sagemaker:UpdateEndpoint"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:Canvas*",
                "arn:aws:sagemaker:*:*:canvas*"
            ]
        },
        {
            "Sid": "ReadCWInvocationMetrics",
            "Effect": "Allow",
            "Action": "cloudwatch:GetMetricData",
            "Resource": "*"
        }
    ]
}
```

AWSkebijakan terkelola: AmazonSageMakerCanvas AI ServicesAccess

Kebijakan ini memberikan izin kepada Amazon SageMaker Canvas untuk menggunakan Amazon Ttract, Amazon Rekognition, Amazon Comprehend, dan Amazon Bedrock.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `textract`— Memungkinkan kepala sekolah menggunakan Amazon Ttract untuk mendeteksi dokumen, pengeluaran, dan identitas dalam sebuah gambar.
- `rekognition`— Memungkinkan kepala sekolah menggunakan Amazon Rekognition untuk mendeteksi label dan teks dalam gambar.
- `comprehend`— Memungkinkan kepala sekolah menggunakan Amazon Comprehend untuk mendeteksi sentimen dan bahasa dominan, dan entitas informasi yang dapat diidentifikasi dan diidentifikasi secara pribadi (PII) dalam dokumen teks.
- `bedrock`— Memungkinkan kepala sekolah menggunakan Amazon Bedrock untuk membuat daftar dan memanggil model pondasi.
- `iam`— Mengizinkan prinsip meneruskan peran IAM ke Amazon Bedrock.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Textract",
            "Effect": "Allow",
            "Action": [
                "textract:AnalyzeDocument",
                "textract:AnalyzeExpense",
                "textract:AnalyzeID",
                "textract:StartDocumentAnalysis",
                "textract:StartExpenseAnalysis",
                "textract:GetDocumentAnalysis",
                "textract:GetExpenseAnalysis"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Rekognition",
            "Effect": "Allow",
            "Action": [
                "rekognition:DetectLabels",
                "rekognition:DetectText"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "Comprehend",
            "Effect": "Allow",
            "Action": [
                "comprehend:BatchDetectDominantLanguage",
                "comprehend:BatchDetectEntities",
                "comprehend:BatchDetectSentiment",
                "comprehend:DetectPiiEntities",
                "comprehend:DetectEntities",
                "comprehend:DetectSentiment",
                "comprehend:DetectDominantLanguage"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Bedrock",
            "Effect": "Allow",
            "Action": [
                "bedrock:InvokeModel",
                "bedrock:ListFoundationModels",
                "bedrock:InvokeModelWithResponseStream"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateBedrockResourcesPermission",
            "Effect": "Allow",
            "Action": [
                "bedrock:CreateModelCustomizationJob",
                "bedrock:CreateProvisionedModelThroughput",
                "bedrock:TagResource"
            ],
            "Resource": [
                "arn:aws:bedrock:*:*:model-customization-job/*",
                "arn:aws:bedrock:*:*:custom-model/*",
                "arn:aws:bedrock:*:*:provisioned-model/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": [
                        "SageMaker",
                        "Canvas"
                    ]
```

```
                },
                "StringEquals": {
                    "aws:RequestTag/SageMaker": "true",
                    "aws:RequestTag/Canvas": "true",
                    "aws:ResourceTag/SageMaker": "true",
                    "aws:ResourceTag/Canvas": "true"
                }
            }
        },
        {
            "Sid": "GetStopAndDeleteBedrockResourcesPermission",
            "Effect": "Allow",
            "Action": [
                "bedrock:GetModelCustomizationJob",
                "bedrock:GetCustomModel",
                "bedrock:GetProvisionedModelThroughput",
                "bedrock:StopModelCustomizationJob",
                "bedrock:DeleteProvisionedModelThroughput"
            ],
            "Resource": [
                "arn:aws:bedrock:*:*:model-customization-job/*",
                "arn:aws:bedrock:*:*:custom-model/*",
                "arn:aws:bedrock:*:*:provisioned-model/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SageMaker": "true",
                    "aws:ResourceTag/Canvas": "true"
                }
            }
        },
        {
            "Sid": "FoundationModelPermission",
            "Effect": "Allow",
            "Action": [
                "bedrock:CreateModelCustomizationJob"
            ],
            "Resource": [
                "arn:aws:bedrock:*::foundation-model/*"
            ]
        },
        {
            "Sid": "BedrockFineTuningPassRole",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "bedrock.amazonaws.com"
                }
            }
        }
    ]
}
```

AWSkebijakan terkelola: AmazonSageMakerCanvasBedrockAccess

Kebijakan ini memberikan izin yang biasanya diperlukan untuk menggunakan Amazon SageMaker Canvas dengan Amazon Bedrock.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `s3`— Memungkinkan prinsipal untuk menambahkan dan mengambil objek dari ember Amazon S3 di direktori “Sagemaker-*/Canvas”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3CanvasAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::sagemaker-*/Canvas",
                "arn:aws:s3:::sagemaker-*/Canvas/*"
            ]
        },
```

```
        {
            "Sid": "S3BucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::sagemaker-*"
            ]
        }
    ]
}
```

AWSkebijakan terkelola: AmazonSageMakerCanvasForecastAccess

Kebijakan ini memberikan izin yang biasanya diperlukan untuk menggunakan Amazon SageMaker Canvas dengan Amazon Forecast.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- s3— Mengizinkan prinsipal menambahkan dan mengambil objek dari bucket Amazon S3. Benda-benda ini terbatas pada mereka yang namanya dimulai dengan “sagemaker-”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::sagemaker-*/Canvas",
                "arn:aws:s3:::sagemaker-*/canvas"
            ]
        }
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::sagemaker-*"
            ]
        }
    ]
}
```

SageMaker Pembaruan Amazon ke kebijakan terkelola Amazon SageMaker Canvas

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk SageMaker Canvas karena layanan ini mulai melacak perubahan-perubahan tersebut.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerCanvasBedrockAccess- Kebijakan baru	1	Kebijakan awal	2 Februari 2024
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	9	Tambahkan `sagemaker:ListEndpoints` izin.	24 Januari 2024
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	8	Tambahkan`sagemaker:UpdateEndpointWeightsAndCapacities` ,`sagemaker:DescribeEndpointConfig` ,`sagemaker:InvokeEndpointAsync` ,`athena:ListDataCatalogs` ,`athena:GetQueryExe`	8 Desember 2023

Kebijakan	Versi	Perubahan	Tanggal
		`cution ,athena:GetQueryResults ,athena:StartQueryExecution ,,athena:StopQueryExecution ,athena:ListDatabases ,cloudwatch:DescribeAlarms ,cloudwatch:PutMetricAlarm ,cloudwatch:DeleteAlarms ,` dan `iam:CreateServiceLinkedRole` izin.	
AmazonSageMakerCanvasDataPrepFullAkses - Pembaruan ke kebijakan yang tersedia	2	Pembaruan kecil untuk menegakkan maksud kebijakan sebelumnya, versi 1; tidak ada izin yang ditambahkan atau dihapus.	7 Desember 2023

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerCanvasAI ServicesAccess - Pembaruan ke kebijakan yang tersedia	3	Tambahkan`bedrock:InvokeModelWithResponseStream` ,`bedrock:GetModelCustomizationJob` ,`bedrock:StopModelCustomizationJob` ,`bedrock:GetCustomModel` ,`bedrock:GetProvisionedModelThroughput` ,`bedrock:DeleteProvisionedModelThroughput` ,`bedrock:TagResource` ,`bedrock:CreateModelCustomizationJob` ,`bedrock:CreateProvisionedModelThroughput` , dan `iam:PassRole` izin.	29 November 2023
AmazonSageMakerCanvasDataPrepFullAkses - Kebijakan baru	1	Kebijakan awal	26 Oktober 2023

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerCanvasDirectDeployAccess- Kebijakan baru	1	Kebijakan awal	6 Oktober 2023
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	7	Tambahkan `sagemaker:DeleteEndpointConfig` , `sagemaker:DeleteModel` , dan `sagemaker:InvokeEndpoint` izin. Juga tambahkan `s3:GetObject` izin untuk SageMaker JumpStart sumber daya di wilayah tertentu.	29 September 2023
AmazonSageMakerCanvasAI ServicesAccess - Perbarui ke kebijakan yang ada	2	Tambahkan `bedrock:InvokeModel` dan `bedrock:ListFoundationModels` izin.	29 September 2023
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	6	Tambahkan `rds:DescribeDBInstances` izin.	29 Agustus 2023
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	5	Tambahkan `application-autoscaling:PutScalingPolicy` dan `application-autoscaling:RegisterScalableTarget` izin.	24 Juli 2023

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	4	Tambahkan`sagemaker:CreateModelPackage` ,`sagemaker:CreateModelPackageGroup` ,`sagemaker:DescribeModelPackage` ,`sagemaker:DescribeModelPackageGroup` ,`sagemaker:ListModelPackages` , dan `sagemaker:ListModelPackageGroups` izin.	4 Mei 2023
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	3	Tambahkan`sagemaker:CreateAutoMLJobV2` ,`sagemaker:DescribeAutoMLJobV2` , dan `glue:SearchTables` izin.	24 Maret 2023
AmazonSageMakerCanvasAI ServicesAccess - Kebijakan baru	1	Kebijakan awal	23 Maret 2023

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerCanvasFullAccess - Pembaruan ke kebijakan yang tersedia	2	Tambahkan `forecast:DeleteResourceTree` izin.	6 Desember 2022
AmazonSageMakerCanvasFullAccess - Kebijakan baru	1	Kebijakan awal	8 September 2022
AmazonSageMakerCanvasForecastAccess - Kebijakan baru	1	Kebijakan awal	24 Agustus 2022

AWSkebijakan terkelola untuk Amazon SageMaker Cluster

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan SageMaker Cluster. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

- AWSkebijakan terkelola: AmazonSageMakerClusterInstanceRolePolicy
- SageMaker Pembaruan Amazon ke kebijakan terkelola Amazon SageMaker Cluster

AWSkebijakan terkelola: AmazonSageMakerClusterInstanceRolePolicy

Kebijakan ini memberikan izin yang biasanya diperlukan untuk menggunakan Amazon SageMaker Cluster.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `cloudwatch`— Memungkinkan kepala sekolah untuk memposting metrik Amazon. CloudWatch
- `logs`— Memungkinkan kepala sekolah untuk mempublikasikan CloudWatch aliran log.

- `s3`— Memungkinkan kepala sekolah untuk membuat daftar dan mengambil file skrip siklus hidup dari bucket Amazon S3 di akun Anda. Ember ini terbatas pada mereka yang namanya dimulai dengan “sagemaker-”.
- `ssmmessages`— Memungkinkan kepala sekolah untuk membuka koneksi ke. AWS Systems Manager

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "CloudwatchLogStreamPublishPermissions",
      "Effect" : "Allow",
      "Action" : [
        "logs:PutLogEvents",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams"
      ],
      "Resource" : [
        "arn:aws:logs:*:*:log-group:/aws/sagemaker/Clusters/*:log-stream:*"
      ]
    },
    {
      "Sid" : "CloudwatchLogGroupCreationPermissions",
      "Effect" : "Allow",
      "Action" : [
        "logs:CreateLogGroup"
      ],
      "Resource" : [
        "arn:aws:logs:*:*:log-group:/aws/sagemaker/Clusters/*"
      ]
    },
    {
      "Sid" : "CloudwatchPutMetricDataAccess",
      "Effect" : "Allow",
      "Action" : [
        "cloudwatch:PutMetricData"
      ],
      "Resource" : [
        "*"
      ],
      "Condition" : {
```

```
        "StringEquals" : {
          "cloudwatch:namespace" : "/aws/sagemaker/Clusters"
        }
      }
    },
    {
      "Sid" : "DataRetrievalFromS3BucketPermissions",
      "Effect" : "Allow",
      "Action" : [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource" : [
        "arn:aws:s3:::sagemaker-*"
      ],
      "Condition" : {
        "StringEquals" : {
          "aws:ResourceAccount" : "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid" : "SSMConnectivityPermissions",
      "Effect" : "Allow",
      "Action" : [
        "ssmmessages:CreateControlChannel",
        "ssmmessages:CreateDataChannel",
        "ssmmessages:OpenControlChannel",
        "ssmmessages:OpenDataChannel"
      ],
      "Resource" : "*"
    }
  ]
}
```

SageMaker Pembaruan Amazon ke kebijakan terkelola Amazon SageMaker Cluster

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk SageMaker Claster karena layanan ini mulai melacak perubahan-perubahan tersebut. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat SageMaker dokumen.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerClusterInstanceRolePolicy- Kebijakan baru	1	Kebijakan awal	29 November 2023

AWSkebijakan terkelola untuk Amazon SageMaker Feature Store

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan Toko Fitur. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

- AWSkebijakan terkelola: AmazonSageMakerFeatureStoreAccess
- SageMaker Pembaruan Amazon ke kebijakan terkelola Amazon SageMaker Feature Store

AWSkebijakan terkelola: AmazonSageMakerFeatureStoreAccess

Kebijakan ini memberikan izin yang diperlukan untuk mengaktifkan toko offline untuk grup SageMaker fitur Amazon Feature Store.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `s3`— Mengizinkan prinsipal menulis data ke toko offline Amazon S3. Ember ini terbatas pada mereka yang namanya termasuk "SageMaker“, “Sagemaker”, atau “sagemaker”.
- `s3`— Memungkinkan prinsipal membaca file manifes yang ada yang disimpan di `metadata` folder bucket S3 toko offline.
- `glue`— Memungkinkan kepala sekolah untuk membaca dan memperbarui tabel GlueAWS. Izin ini terbatas pada tabel di `sagemaker_featurestore` folder.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*/metadata/*",
                "arn:aws:s3:::*Sagemaker*/metadata/*",
                "arn:aws:s3:::*sagemaker*/metadata/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTable",
                "glue:UpdateTable"
            ],
            "Resource": [
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/sagemaker_featurestore",
                "arn:aws:glue:*:*:table/sagemaker_featurestore/*"
            ]
        }
    ]
}
```

SageMaker Pembaruan Amazon ke kebijakan terkelola Amazon SageMaker Feature Store

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk Feature Store karena layanan ini mulai melacak perubahan-perubahan tersebut. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat SageMaker dokumen.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerFeatureStoreAccess - Pembaruan ke kebijakan yang tersedia	3	Tambahkan `s3:GetObject` ,`glue:GetTable` , dan `glue:UpdateTable` izin.	5 Desember 2022
AmazonSageMakerFeatureStoreAccess - Pembaruan ke kebijakan yang tersedia	2	Tambahkan `s3:PutObjectAcl` izin.	23 Februari 2021
AmazonSageMakerFeatureStoreAccess - Kebijakan baru	1	Kebijakan awal	1 Desember 2020

AWSkebijakan terkelola untuk SageMaker geospasial Amazon

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan SageMaker geospasial. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

- AWSkebijakan terkelola: AmazonSageMakerGeospatialFullAccess
- AWSkebijakan terkelola: AmazonSageMakerGeospatialExecutionRole
- SageMaker Pembaruan Amazon ke kebijakan terkelola SageMaker geospasial Amazon

AWSkebijakan terkelola: AmazonSageMakerGeospatialFullAccess

Kebijakan ini memberikan izin yang memungkinkan akses penuh ke SageMaker geospasial Amazon melalui dan SDKAWS Management Console.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `sagemaker-geospatial`— Mengizinkan prinsipal memiliki akses penuh ke semua SageMaker sumber daya geospasial.
- `iam`— Mengizinkan prinsipal meneruskan peran IAM ke geospasial. SageMaker

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sagemaker-geospatial:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": ["iam:PassRole"],
      "Resource": "arn:aws:iam::*:role/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "sagemaker-geospatial.amazonaws.com"
           ]
        }
      }
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerGeospatialExecutionRole

Kebijakan ini memberikan izin yang biasanya diperlukan untuk menggunakan SageMaker geospasial.

Detail izin

Kebijakan AWS terkelola ini mencakup izin berikut.

- `s3`— Mengizinkan prinsipal menambahkan dan mengambil objek dari bucket Amazon S3. Benda-benda ini terbatas pada mereka yang namanya mengandung "SageMaker“, “Sagemaker”, atau “sagemaker”.
- `sagemaker-geospatial`— Memungkinkan kepala sekolah untuk mengakses pekerjaan pengamatan Bumi melalui API. `GetEarthObservationJob`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:PutObject",
          "s3:GetObject",
          "s3:ListBucketMultipartUploads"
      ],
      "Resource": [
        "arn:aws:s3:::*SageMaker*",
        "arn:aws:s3:::*Sagemaker*",
        "arn:aws:s3:::*sagemaker*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "sagemaker-geospatial:GetEarthObservationJob",
      "Resource": "arn:aws:sagemaker-geospatial:*:*:earth-observation-job/*"
    },
    {
      "Effect": "Allow",
      "Action": "sagemaker-geospatial:GetRasterDataCollection",
      "Resource": "arn:aws:sagemaker-geospatial:*:*:raster-data-collection/*"
    }
  ]
}
```

SageMaker Pembaruan Amazon ke kebijakan terkelola SageMaker geospasial Amazon

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk SageMaker geospasial karena layanan ini mulai melacak perubahan-perubahan tersebut.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerGeospatialExecutionRole- Kebijakan yang diperbarui	2	Tambahkan `sagemaker-geospatial:GetRas`	10 Mei 2023

Kebijakan	Versi	Perubahan	Tanggal
		`terDataCollection` izin.	
AmazonSageMakerGeospatialFullAccess- Kebijakan baru	1	Kebijakan awal	30 November 2022
AmazonSageMakerGeospatialExecutionRole - Kebijakan baru	1	Kebijakan awal	30 November 2022

AWSKebijakan Terkelola untuk Amazon SageMaker Ground Truth

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan SageMaker Ground Truth. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

- AWSkebijakan terkelola: AmazonSageMakerGroundTruthExecution
- AWSkebijakan terkelola: GroundTruthSyntheticConsoleFullAccess
- AWSkebijakan terkelola: GroundTruthSyntheticConsoleReadOnlyAccess
- Amazon SageMaker memperbarui kebijakan terkelola SageMaker Ground Truth

AWSkebijakan terkelola: AmazonSageMakerGroundTruthExecution

Kebijakan AWS terkelola ini memberikan izin yang biasanya diperlukan untuk menggunakan SageMaker Ground Truth.

Detail izin

Kebijakan ini mencakup izin berikut.

- `lambda`— Memungkinkan kepala sekolah untuk memanggil fungsi Lambda yang namanya mencakup “sagemaker” (case-insensitive), "“, atau"”. GtRecipe LabelingFunction

- `s3`— Mengizinkan prinsipal menambahkan dan mengambil objek dari bucket Amazon S3. Objek-objek ini terbatas pada mereka yang nama case-insensitive mengandung “groundtruth” atau “sagemaker”, atau ditandai dengan "". SageMaker
- `cloudwatch`— Memungkinkan kepala sekolah untuk memposting metrik. CloudWatch
- `logs`— Mengizinkan prinsipal untuk membuat dan mengakses aliran log, dan memposting peristiwa log.
- `sqs`— Memungkinkan kepala sekolah untuk membuat antrian Amazon SQS, dan mengirim serta menerima pesan Amazon SQS. Izin ini terbatas pada antrian yang namanya termasuk "". GroundTruth
- `sns`— Memungkinkan kepala sekolah untuk berlangganan dan mempublikasikan pesan ke topik Amazon SNS yang namanya tidak peka huruf besar/kecil berisi “groundtruth” atau “sagemaker”.
- `ec2`— Memungkinkan prinsipal untuk membuat, mendeskripsikan, dan menghapus titik akhir VPC Amazon VPC yang nama layanan titik akhir VPCnya berisi "" atau “pelabelan”. sagemaker-task-resources

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CustomLabelingJobs",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:*GtRecipe*",
                "arn:aws:lambda:*:*:function:*LabelingFunction*",
                "arn:aws:lambda:*:*:function:*SageMaker*",
                "arn:aws:lambda:*:*:function:*sagemaker*",
                "arn:aws:lambda:*:*:function:*Sagemaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetObject",
                "s3:PutObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::*GroundTruth*",
                "arn:aws:s3:::*Groundtruth*",
                "arn:aws:s3:::*groundtruth*",
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "s3:ExistingObjectTag/SageMaker": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "StreamingQueue",
            "Effect": "Allow",
```

```
            "Action": [
                "sqs:CreateQueue",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage",
                "sqs:SetQueueAttributes"
            ],
            "Resource": "arn:aws:sqs:*:*:*GroundTruth*"
        },
        {
            "Sid": "StreamingTopicSubscribe",
            "Effect": "Allow",
            "Action": "sns:Subscribe",
            "Resource": [
                "arn:aws:sns:*:*:*GroundTruth*",
                "arn:aws:sns:*:*:*Groundtruth*",
                "arn:aws:sns:*:*:*groundTruth*",
                "arn:aws:sns:*:*:*groundtruth*",
                "arn:aws:sns:*:*:*SageMaker*",
                "arn:aws:sns:*:*:*Sagemaker*",
                "arn:aws:sns:*:*:*sageMaker*",
                "arn:aws:sns:*:*:*sagemaker*"
            ],
            "Condition": {
                "StringEquals": {
                    "sns:Protocol": "sqs"
                },
                "StringLike": {
                    "sns:Endpoint": "arn:aws:sqs:*:*:*GroundTruth*"
                }
            }
        },
        {
            "Sid": "StreamingTopic",
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": [
                "arn:aws:sns:*:*:*GroundTruth*",
                "arn:aws:sns:*:*:*Groundtruth*",
                "arn:aws:sns:*:*:*groundTruth*",
```

```
                "arn:aws:sns:*:*:*groundtruth*",
                "arn:aws:sns:*:*:*SageMaker*",
                "arn:aws:sns:*:*:*Sagemaker*",
                "arn:aws:sns:*:*:*sageMaker*",
                "arn:aws:sns:*:*:*sagemaker*"
            ]
        },
        {
            "Sid": "StreamingTopicUnsubscribe",
            "Effect": "Allow",
            "Action": [
                "sns:Unsubscribe"
            ],
            "Resource": "*"
        },
        {
            "Sid": "WorkforceVPC",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeVpcEndpoints",
                "ec2:DeleteVpcEndpoints"
            ],
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "ec2:VpceServiceName": [
                        "*sagemaker-task-resources*",
                        "aws.sagemaker*labeling*"
                    ]
                }
            }
        }
    ]
}
```

AWSkebijakan terkelola: GroundTruthSyntheticConsoleFullAccess

Kebijakan AWS terkelola ini memberikan izin yang diperlukan untuk menggunakan sebagian besar fitur konsol data sintetis SageMaker Ground Truth. Kebijakan ini tersedia di AWS akun Anda. Untuk menggunakan semua fitur konsol, pengguna harus menambahkan izin “s3:GetObject” untuk memungkinkan konsol data sintetis SageMaker Ground Truth menampilkan data dari bucket S3 mereka. Ini dapat dilakukan dengan melampirkan kebijakan ReadOnlyAccess terkelola AmazonS3 ke

peran mereka atau dengan menambahkan “s3:GetObject” untuk sumber daya S3 tertentu ke peran mereka.

Detail izin

Kebijakan ini mencakup izin berikut.

- `s3`— Memungkinkan pengambilan objek dari bucket Amazon S3 untuk menampilkan data di konsol.
- `sagemaker-groundtruth-synthetic`— Memungkinkan konsol memanggil API data sintetis SageMaker Ground Truth.

```
        {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker-groundtruth-synthetic:*",
                "s3:ListBucket"
            ],
            "Resource": "*"
        }
    ]
}
```

AWSkebijakan terkelola: GroundTruthSyntheticConsoleReadOnlyAccess

Kebijakan AWS terkelola ini memberikan akses hanya-baca ke data sintetis SageMaker Ground Truth melalui. AWS Management Console Untuk menggunakan semua fitur konsol, pengguna harus menambahkan izin “s3:GetObject” untuk memungkinkan konsol data sintetis SageMaker Ground Truth menampilkan data dari bucket S3 mereka. Ini dapat dilakukan dengan melampirkan kebijakan ReadOnlyAccess terkelola AmazonS3 ke peran mereka atau dengan menambahkan “s3:GetObject” untuk sumber daya S3 tertentu ke peran mereka.

Detail izin

Kebijakan ini mencakup izin berikut.

- `s3`— Memungkinkan pengambilan objek dari bucket Amazon S3 untuk menampilkan data di konsol.
- `sagemaker-groundtruth-synthetic`— Memungkinkan konsol memanggil ReadOnly API data sintetis SageMaker Ground Truth.

```
        {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker-groundtruth-synthetic:List*",
                "sagemaker-groundtruth-synthetic:Get*",
                "s3:ListBucket"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon SageMaker memperbarui kebijakan terkelola SageMaker Ground Truth

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon SageMaker Ground Truth sejak layanan ini mulai melacak perubahan tersebut.

Kebijakan	Versi	Perubahan	Tanggal
GroundTruthSyntheticConsoleFullAccess- Kebijakan baru	1	Kebijakan awal	25 Agustus 2022
GroundTruthSyntheticConsoleReadOnlyAccess- Kebijakan baru	1	Kebijakan awal	25 Agustus 2022
AmazonSageMakerGroundTruthExecution -	3	Tambahkan`ec2:CreateVpcEndpoint ,ec2:Descr`	29 April 2022

Kebijakan	Versi	Perubahan	Tanggal
Pembaruan ke kebijakan yang tersedia		`ibeVpcEndpoints` , dan `ec2:Delet eVpcEndpoints` izin.	
AmazonSageMakerGro undTruthExecution - Pembaruan ke kebijakan yang tersedia	2	Hapus `sqs:SendM essageBatch` izin.	11 April 2022
AmazonSageMakerGro undTruthExecution - Kebijakan baru	1	Kebijakan awal	20 Juli 2020

AWSKebijakan Terkelola untuk Tata Kelola SageMaker Model

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan Tata Kelola SageMaker Model. Kebijakan ini tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

- AWSkebijakan terkelola: AmazonSageMakerModelGovernanceUseAccess
- SageMaker Pembaruan Amazon ke kebijakan terkelola Tata Kelola SageMaker Model

AWSkebijakan terkelola: AmazonSageMakerModelGovernanceUseAccess

Kebijakan AWS terkelola ini memberikan izin yang diperlukan untuk menggunakan semua fitur SageMaker Tata Kelola Amazon. Kebijakan ini tersedia di AWS akun Anda.

Kebijakan ini mencakup izin berikut.

- `s3`— Ambil objek dari bucket Amazon S3. Objek yang dapat diambil terbatas pada objek yang namanya tidak peka huruf besar/kecil berisi string. `"sagemaker"`
- `kms`— Buat daftar AWS KMS kunci yang akan digunakan untuk enkripsi konten.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:ListMonitoringAlerts",
                "sagemaker:ListMonitoringExecutions",
                "sagemaker:UpdateMonitoringAlert",
                "sagemaker:StartMonitoringSchedule",
                "sagemaker:StopMonitoringSchedule",
                "sagemaker:ListMonitoringAlertHistory",
                "sagemaker:DescribeModelPackage",
                "sagemaker:DescribeModelPackageGroup",
                "sagemaker:CreateModelCard",
                "sagemaker:DescribeModelCard",
                "sagemaker:UpdateModelCard",
                "sagemaker:DeleteModelCard",
                "sagemaker:ListModelCards",
                "sagemaker:ListModelCardVersions",
                "sagemaker:CreateModelCardExportJob",
                "sagemaker:DescribeModelCardExportJob",
                "sagemaker:ListModelCardExportJobs"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:ListTrainingJobs",
                "sagemaker:DescribeTrainingJob",
                "sagemaker:ListModels",
                "sagemaker:DescribeModel",
                "sagemaker:Search",
                "sagemaker:AddTags",
                "sagemaker:DeleteTags",
                "sagemaker:ListTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:CreateBucket",
                "s3:GetBucketLocation",
            ],
            "Resource": [
                "arn:aws:s3:::*SageMaker*",
                "arn:aws:s3:::*Sagemaker*",
                "arn:aws:s3:::*sagemaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```

SageMaker Pembaruan Amazon ke kebijakan terkelola Tata Kelola SageMaker Model

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk SageMaker Tata Kelola karena layanan ini mulai melacak perubahan-perubahan tersebut. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat SageMaker dokumen.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerModelGovernanceUseAccess - Pembaruan ke kebijakan yang tersedia	2	Tambahkan `sagemaker:DescribeModelPackage` dan `DescribeModelPackageGroup` izin.	17 Juli 2023

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerModelGovernanceUseAccess - Kebijakan baru	1	Kebijakan awal	30 November 2022

AWSKebijakan Terkelola untuk Registri Model

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan Model Registry. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol Amazon.

Topik

- AWSkebijakan terkelola: AmazonSageMakerModelRegistryFullAccess
- SageMaker Pembaruan Amazon ke kebijakan terkelola Model Registry

AWSkebijakan terkelola: AmazonSageMakerModelRegistryFullAccess

Kebijakan AWS terkelola ini memberikan izin yang diperlukan untuk menggunakan semua fitur Model Registry di dalam domain Amazon SageMaker . Kebijakan ini dilampirkan ke peran eksekusi saat mengonfigurasi pengaturan Registri Model untuk mengaktifkan izin Registri Model.

Kebijakan ini mencakup izin berikut.

- `ecr`— Memungkinkan kepala sekolah untuk mengambil informasi, termasuk metadata, tentang citra Amazon Elastic Container Registry (Amazon ECR).
- `iam`— Memungkinkan kepala sekolah untuk meneruskan peran eksekusi ke layanan Amazon. SageMaker
- `resource-groups`— Memungkinkan prinsipal untuk membuat, daftar, menandai, dan menghapus. AWS Resource Groups
- `s3`— Memungkinkan prinsipal untuk mengambil objek dari bucket Amazon Simple Storage Service (Amazon S3) tempat versi model disimpan. Objek yang dapat diambil terbatas pada objek yang namanya tidak peka huruf besar/kecil berisi string. `"sagemaker"`
- `sagemaker`— Memungkinkan prinsipal untuk membuat katalog, mengelola, dan menyebarkan model menggunakan registri model. SageMaker

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:DescribeAction",
        "sagemaker:DescribeInferenceRecommendationsJob",
        "sagemaker:DescribeModelPackage",
        "sagemaker:DescribeModelPackageGroup",
        "sagemaker:DescribePipeline",
        "sagemaker:DescribePipelineExecution",
        "sagemaker:ListAssociations",
        "sagemaker:ListArtifacts",
        "sagemaker:ListModelMetadata",
        "sagemaker:ListModelPackages",
        "sagemaker:Search",
        "sagemaker:GetSearchSuggestions"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:AddTags",
        "sagemaker:CreateModel",
        "sagemaker:CreateModelPackage",
        "sagemaker:CreateModelPackageGroup",
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:CreateInferenceRecommendationsJob",
        "sagemaker:DeleteModelPackage",
        "sagemaker:DeleteModelPackageGroup",
        "sagemaker:DeleteTags",
        "sagemaker:UpdateModelPackage"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
```

```
      "Resource": [
        "arn:aws:s3:::*SageMaker*",
        "arn:aws:s3:::*Sagemaker*",
        "arn:aws:s3:::*sagemaker*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:DescribeImages"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "sagemaker.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "resource-groups:GetGroupQuery"
      ],
      "Resource": "arn:aws:resource-groups:*:*:group/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:ListGroupResources"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:Tag"
      ],
      "Resource": "arn:aws:resource-groups:*:*:group/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": "sagemaker:collection"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "resource-groups:DeleteGroup",
      "Resource": "arn:aws:resource-groups:*:*:group/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/sagemaker:collection": "true"
        }
      }
    }
  ]
}
```

SageMaker Pembaruan Amazon ke kebijakan terkelola Model Registry

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk Model Registry karena layanan ini mulai melacak perubahan-perubahan tersebut. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat SageMaker dokumen.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerModelRegistryFullAccess- Kebijakan baru	1	Kebijakan awal	12 April 2023

AWSKebijakan Terkelola untuk SageMaker Notebook

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan Buku SageMaker Catatan. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

- AWSkebijakan terkelola: AmazonSageMakerNotebooksServiceRolePolicy
- Amazon SageMaker memperbarui kebijakan terkelola SageMaker Notebook

AWSkebijakan terkelola: AmazonSageMakerNotebooksServiceRolePolicy

Kebijakan AWS terkelola ini memberikan izin yang biasanya diperlukan untuk menggunakan Notebook Amazon SageMaker . Kebijakan ditambahkan ke kebijakan `AmazonSageMaker-ExecutionRole` yang dibuat saat Anda onboard ke Amazon SageMaker Studio Classic. Untuk informasi selengkapnya tentang peran terkait layanan, lihat. Peran Tertaut Layanan

Detail izin

Kebijakan ini mencakup izin berikut.

- `elasticfilesystem`— Memungkinkan prinsipal untuk membuat dan menghapus sistem file Amazon Elastic File System (EFS), titik akses, dan target mount. Ini terbatas pada yang ditandai dengan kunci ManagedByAmazonSageMakerResource. Memungkinkan prinsipal untuk mendeskripsikan semua sistem file EFS, titik akses, dan target pemasangan. Memungkinkan prinsipal untuk membuat atau menimpa tag untuk titik akses EFS dan memasang target.
- `ec2`— Memungkinkan prinsipal membuat antarmuka jaringan dan grup keamanan untuk instans Amazon Elastic Compute Cloud (EC2). Juga memungkinkan prinsipal untuk membuat dan menimpa tag untuk sumber daya ini.
- `sso`— Memungkinkan prinsipal untuk menambah dan menghapus instance aplikasi terkelola ke. AWS IAM Identity Center

- `sagemaker`— Memungkinkan kepala sekolah untuk membuat dan membaca SageMaker profil pengguna.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticfilesystem:CreateAccessPoint",
            "Resource": "arn:aws:elasticfilesystem:*:*:file-system/*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/ManagedByAmazonSageMakerResource": "*",
                    "aws:RequestTag/ManagedByAmazonSageMakerResource": "*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:DeleteAccessPoint"
            ],
            "Resource": "arn:aws:elasticfilesystem:*:*:access-point/*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/ManagedByAmazonSageMakerResource": "*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "elasticfilesystem:CreateFileSystem",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/ManagedByAmazonSageMakerResource": "*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "elasticfilesystem:CreateMountTarget",
                "elasticfilesystem:DeleteFileSystem",
                "elasticfilesystem:DeleteMountTarget"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/ManagedByAmazonSageMakerResource": "*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:DescribeAccessPoints",
                "elasticfilesystem:DescribeFileSystems",
                "elasticfilesystem:DescribeMountTargets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticfilesystem:TagResource",
            "Resource": [
                "arn:aws:elasticfilesystem:*:*:access-point/*",
                "arn:aws:elasticfilesystem:*:*:file-system/*"
            ],
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/ManagedByAmazonSageMakerResource": "*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DeleteSecurityGroup",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ec2:ResourceTag/ManagedByAmazonSageMakerResource": "*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "sso:CreateManagedApplicationInstance",
                "sso:DeleteManagedApplicationInstance",
                "sso:GetManagedApplicationInstance"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateUserProfile",
                "sagemaker:DescribeUserProfile"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon SageMaker memperbarui kebijakan terkelola SageMaker Notebook

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk Amazon SageMaker karena layanan ini mulai melacak perubahan-perubahan tersebut.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerNotebooksServiceRolePolicy	7	Menambahkan `elasticfilesystem:TagResource` izin.	9 Maret 2023
AmazonSageMakerNotebooksServiceRolePolicy	6	Menambahkan izin untuk `elasticfilesystem:CreateAccessPoint`, `elasticfilesystem:DeleteAccessPoint`, dan `elasticfilesystem:DescribeAccessPoints`.	12 Januari 2023
		SageMaker mulai melacak perubahan untuk kebijakan AWS terkelola	1 Juni 2021

AWSKebijakan Terkelola untuk SageMaker Saluran Pipa

Kebijakan AWS terkelola ini menambahkan izin yang diperlukan untuk menggunakan SageMaker Pipelines. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

Topik

AWSkebijakan terkelola: AmazonSageMakerPipelinesIntegrations

Kebijakan AWS terkelola ini memberikan izin yang biasanya diperlukan untuk menggunakan langkah Callback dan langkah Lambda di Pipelines. SageMaker Kebijakan ditambahkan ke kebijakan `AmazonSageMaker-ExecutionRole` yang dibuat saat Anda onboard ke Amazon SageMaker Studio Classic. Kebijakan dapat dilampirkan ke peran apa pun yang digunakan untuk membuat atau mengeksekusi pipeline.

Kebijakan ini memberikan izin AWS Lambda, Amazon Simple Queue Service (Amazon SQS), EventBridge Amazon, dan IAM yang sesuai yang diperlukan saat membuat pipeline yang menjalankan fungsi Lambda atau menyertakan langkah panggilan balik, yang dapat digunakan untuk langkah persetujuan manual atau menjalankan beban kerja khusus.

Izin Amazon SQS memungkinkan Anda membuat antrean Amazon SQS yang diperlukan untuk menerima pesan panggilan balik, dan juga untuk mengirim pesan ke antrean tersebut.

Izin Lambda memungkinkan Anda membuat, membaca, memperbarui, dan menghapus fungsi Lambda yang digunakan dalam langkah-langkah pipeline, dan juga untuk menjalankan fungsi Lambda tersebut.

Kebijakan ini memberikan izin EMR Amazon yang diperlukan untuk menjalankan langkah EMR Amazon pipeline.

Detail izin

Kebijakan ini mencakup izin berikut.

- `elasticmapreduce`— Baca, tambahkan, dan batalkan langkah-langkah di klaster EMR Amazon yang sedang berjalan. Baca, buat, dan akhiri cluster EMR Amazon baru.

- `events`— Baca, buat, perbarui, dan tambahkan target ke EventBridge aturan bernama `SageMakerPipelineExecutionEMRStepStatusUpdateRule` dan`SageMakerPipelineExecutionEMRClusterStatusUpdateRule`.
- `iam`— Lulus peran IAM ke layanan AWS Lambda, Amazon EMR, dan Amazon EC2.
- `lambda`— Buat, baca, perbarui, hapus, dan panggil fungsi Lambda. Izin ini terbatas pada fungsi yang namanya termasuk “sagemaker”.
- `sqs`— Buat antrian Amazon SQS; kirim pesan Amazon SQS. Izin ini terbatas pada antrian yang namanya termasuk “pembuat sagemaker”.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:UpdateFunctionCode"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:*sagemaker*",
                "arn:aws:lambda:*:*:function:*sageMaker*",
                "arn:aws:lambda:*:*:function:*SageMaker*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:CreateQueue",
                "sqs:SendMessage"
            ],
            "Resource": [
                "arn:aws:sqs:*:*:*sagemaker*",
                "arn:aws:sqs:*:*:*sageMaker*",
                "arn:aws:sqs:*:*:*SageMaker*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "elasticmapreduce.amazonaws.com",
                        "ec2.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:DescribeRule",
                "events:PutRule",
                "events:PutTargets"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/
SageMakerPipelineExecutionEMRStepStatusUpdateRule",
                "arn:aws:events:*:*:rule/
SageMakerPipelineExecutionEMRClusterStatusUpdateRule"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:AddJobFlowSteps",
                "elasticmapreduce:CancelSteps",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:RunJobFlow",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:TerminateJobFlows",
                "elasticmapreduce:ListSteps"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:*:*:cluster/*"
            ]
        }
```

```
    ]
}
```

Amazon SageMaker memperbarui kebijakan terkelola SageMaker Pipelines

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk Amazon SageMaker karena layanan ini mulai melacak perubahan-perubahan tersebut.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerPipelinesIntegrations - Pembaruan ke kebijakan yang tersedia	3	Menambahkan izin untuk `elasticmapreduce:RunJobFlows`, `elasticmapreduce:TerminateJobFlows`, `elasticmapreduce:ListSteps`, dan `elasticmapreduce:DescribeCluster`.	17 Februari 2023
AmazonSageMakerPipelinesIntegrations - Pembaruan ke kebijakan yang tersedia	2	Menambahkan izin untuk `lambda:GetFunction`, `events:DescribeRule`, `events:PutRule`, `events:PutTargets`, `elasticmapreduce:AddJobFlowSteps`, `elasticmapreduce:CancelSteps`, dan `elasticmapreduce`	20 April 2022

Kebijakan	Versi	Perubahan	Tanggal
		`preduce:D escribeStep` .	
AmazonSageMakerPip elinesIntegrations - Kebijakan baru	1	Kebijakan awal	30 Juli 2021

AWSKebijakan Terkelola untuk SageMaker proyek dan JumpStart

Kebijakan AWS terkelola ini menambahkan izin untuk menggunakan templat dan JumpStart solusi SageMaker proyek Amazon bawaan. Kebijakan tersedia di AWS akun Anda dan digunakan oleh peran eksekusi yang dibuat dari SageMaker konsol.

SageMaker memproyeksikan dan JumpStart menggunakan AWS Service Catalog untuk menyediakan AWS sumber daya di akun pelanggan. Beberapa sumber daya yang dibuat perlu mengambil peran eksekusi. Misalnya, jika AWS Service Catalog membuat CodePipeline pipeline atas nama pelanggan untuk proyek CI/CD SageMaker machine learning, maka pipeline tersebut memerlukan peran IAM.

AmazonSageMakerServiceCatalogProductsLaunchPeran Peran memiliki izin yang diperlukan untuk meluncurkan SageMaker portofolio produk dari AWS Service Catalog. AmazonSageMakerServiceCatalogProductsUsePeran Peran memiliki izin yang diperlukan untuk menggunakan SageMaker portofolio produk dari AWS Service Catalog. `AmazonSageMakerServiceCatalogProductsLaunchRole`Peran meneruskan `AmazonSageMakerServiceCatalogProductsUseRole` peran ke sumber daya produk AWS Service Catalog yang disediakan.

Topik

- AWSkebijakan terkelola: AmazonSageMakerAdmin - ServiceCatalogProductsServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerPartnerServiceCatalogProducts ApiGatewayServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerPartnerServiceCatalogProducts CloudFormationServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerPartnerServiceCatalogProducts LambdaServiceRolePolicy

- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsApiGatewayServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsCloudformation ServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsCode BuildServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsCode PipelineServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsEvents ServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsFirehose ServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsGlue ServiceRolePolicy
- AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsLambda ServiceRolePolicy
- Amazon SageMaker memperbarui kebijakan AWS terkelola AWS Service Catalog

AWSkebijakan terkelola: AmazonSageMakerAdmin - ServiceCatalogProductsServiceRolePolicy

Kebijakan peran layanan ini digunakan oleh AWS Service Catalog layanan untuk menyediakan produk dari SageMaker portofolio Amazon. Kebijakan ini memberikan izin ke serangkaian AWS layanan terkait termasukAWS CodePipeline,,, AWS CodeCommit AWS Glue AWS CodeBuildAWS CloudFormation, dan lainnya.

`AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy`Kebijakan ini dimaksudkan untuk digunakan oleh `AmazonSageMakerServiceCatalogProductsLaunchRole` peran yang dibuat dari SageMaker konsol. Kebijakan ini menambahkan izin untuk menyediakan AWS sumber daya untuk SageMaker proyek dan JumpStart menggunakan Service Catalog ke akun pelanggan.

Detail izin

Kebijakan ini mencakup izin berikut.

- `apigateway`— Memungkinkan peran untuk memanggil titik akhir API Gateway yang ditandai dengan. `sagemaker:launch-source`
- `cloudformation`— Memungkinkan AWS Service Catalog untuk membuat, memperbarui, dan menghapus CloudFormation tumpukan.
- `codebuild`— Memungkinkan peran yang diasumsikan oleh AWS Service Catalog dan diteruskan CloudFormation untuk membuat, memperbarui, dan menghapus CodeBuild proyek.

- `codecommit`— Memungkinkan peran yang diasumsikan oleh AWS Service Catalog dan diteruskan CloudFormation untuk membuat, memperbarui, dan menghapus CodeCommit repositori.
- `codepipeline`— Memungkinkan peran yang diasumsikan oleh AWS Service Catalog dan diteruskan CloudFormation untuk membuat, memperbarui, dan menghapus CodePipelines.
- `codestar-connections`— Memungkinkan peran untuk melewati AWS CodeStar koneksi.
- `cognito-idp`— Mengizinkan peran untuk membuat, memperbarui, dan menghapus grup dan kolam pengguna. Juga memungkinkan penandaan sumber daya.
- `ecr`— Memungkinkan peran yang diasumsikan oleh AWS Service Catalog dan diteruskan CloudFormation untuk membuat dan menghapus repositori Amazon ECR. Juga memungkinkan penandaan sumber daya.
- `events`— Memungkinkan peran yang diasumsikan oleh AWS Service Catalog dan diteruskan CloudFormation untuk membuat dan menghapus EventBridge aturan. Digunakan untuk mengikat berbagai komponen pipa CICD.
- `firehose`— Memungkinkan peran untuk berinteraksi dengan aliran Kinesis Data Firehose.
- `glue`— Memungkinkan peran untuk berinteraksi denganAWS Glue.
- `iam`— Memungkinkan peran untuk melewati peran yang ditambahkan. `AmazonSageMakerServiceCatalog` Ini diperlukan ketika Proyek menyediakan AWS Service Catalog produk, sebagai peran perlu diteruskanAWS Service Catalog.
- `lambda`— Memungkinkan peran untuk berinteraksi denganAWS Lambda. Juga memungkinkan penandaan sumber daya.
- `logs`— Memungkinkan peran untuk membuat, menghapus, dan mengakses aliran log.
- `s3`— Memungkinkan peran yang diambil oleh AWS Service Catalog dan diteruskan CloudFormation untuk mengakses bucket Amazon S3 tempat kode template Project disimpan.
- `sagemaker`— Memungkinkan peran untuk berinteraksi dengan berbagai SageMaker layanan. Ini dilakukan baik CloudFormation selama penyediaan template, maupun CodeBuild selama eksekusi pipeline CICD. Juga memungkinkan penandaan sumber daya berikut: titik akhir, konfigurasi titik akhir, model, saluran pipa, proyek, dan paket model.
- `states`— Memungkinkan peran untuk membuat, menghapus, dan memperbarui Step Functions yang ditambahkan. `sagemaker`

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET",
        "apigateway:POST",
        "apigateway:PUT",
        "apigateway:PATCH",
        "apigateway:DELETE"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/sagemaker:launch-source": "*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:POST"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:TagKeys": [
            "sagemaker:launch-source"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:PATCH"
      ],
      "Resource": [
        "arn:aws:apigateway:*::/account"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
```

```
        "cloudformation:UpdateStack",
        "cloudformation:DeleteStack"
      ],
      "Resource": "arn:aws:cloudformation:*:*:stack/SC-*",
      "Condition": {
        "ArnLikeIfExists": {
          "cloudformation:RoleArn": [
            "arn:aws:sts::*:assumed-role/AmazonSageMakerServiceCatalog*"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:aws:cloudformation:*:*:stack/SC-*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:GetTemplateSummary",
        "cloudformation:ValidateTemplate"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:CreateProject",
        "codebuild:DeleteProject",
        "codebuild:UpdateProject"
      ],
      "Resource": [
        "arn:aws:codebuild:*:*:project/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:CreateCommit",
        "codecommit:CreateRepository",
```

```
        "codecommit:DeleteRepository",
        "codecommit:GetRepository",
        "codecommit:TagResource"
      ],
      "Resource": [
        "arn:aws:codecommit:*:*:sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:ListRepositories"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "codepipeline:CreatePipeline",
        "codepipeline:DeletePipeline",
        "codepipeline:GetPipeline",
        "codepipeline:GetPipelineState",
        "codepipeline:StartPipelineExecution",
        "codepipeline:TagResource",
        "codepipeline:UpdatePipeline"
      ],
      "Resource": [
        "arn:aws:codepipeline:*:*:sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cognito-idp:CreateUserPool",
        "cognito-idp:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:TagKeys": [
            "sagemaker:launch-source"
          ]
        }
      }
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "cognito-idp:CreateGroup",
        "cognito-idp:CreateUserPoolDomain",
        "cognito-idp:CreateUserPoolClient",
        "cognito-idp:DeleteGroup",
        "cognito-idp:DeleteUserPool",
        "cognito-idp:DeleteUserPoolClient",
        "cognito-idp:DeleteUserPoolDomain",
        "cognito-idp:DescribeUserPool",
        "cognito-idp:DescribeUserPoolClient",
        "cognito-idp:UpdateUserPool",
        "cognito-idp:UpdateUserPoolClient"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/sagemaker:launch-source": "*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CreateRepository",
        "ecr:DeleteRepository",
        "ecr:TagResource"
      ],
      "Resource": [
        "arn:aws:ecr:*:*:repository/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule",
        "events:DeleteRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets"
```

```
      ],
      "Resource": [
        "arn:aws:events:*:*:rule/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "firehose:CreateDeliveryStream",
        "firehose:DeleteDeliveryStream",
        "firehose:DescribeDeliveryStream",
        "firehose:StartDeliveryStreamEncryption",
        "firehose:StopDeliveryStreamEncryption",
        "firehose:UpdateDestination"
      ],
      "Resource": "arn:aws:firehose:*:*:deliverystream/sagemaker-*"
    },
    {
        "Effect": "Allow",
      "Action": [
        "glue:CreateDatabase",
        "glue:DeleteDatabase"
      ],
      "Resource": [
        "arn:aws:glue:*:*:catalog",
        "arn:aws:glue:*:*:database/sagemaker-*",
        "arn:aws:glue:*:*:table/sagemaker-*",
        "arn:aws:glue:*:*:userDefinedFunction/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateClassifier",
        "glue:DeleteClassifier",
        "glue:DeleteCrawler",
        "glue:DeleteJob",
        "glue:DeleteTrigger",
        "glue:DeleteWorkflow",
        "glue:StopCrawler"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateWorkflow"
      ],
      "Resource": [
        "arn:aws:glue:*:*:workflow/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateJob"
      ],
      "Resource": [
        "arn:aws:glue:*:*:job/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateCrawler",
        "glue:GetCrawler"
      ],
      "Resource": [
        "arn:aws:glue:*:*:crawler/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateTrigger",
        "glue:GetTrigger"
      ],
      "Resource": [
        "arn:aws:glue:*:*:trigger/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
```

```
      "Resource": [
        "arn:aws:iam::*:role/service-role/AmazonSageMakerServiceCatalog*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:AddPermission",
        "lambda:CreateFunction",
        "lambda:DeleteFunction",
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:InvokeFunction",
        "lambda:RemovePermission"
      ],
      "Resource": [
        "arn:aws:lambda:*:*:function:sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "lambda:TagResource",
      "Resource": [
        "arn:aws:lambda:*:*:function:sagemaker-*"
      ],
      "Condition": {
        "ForAllValues:StringLike": {
          "aws:TagKeys": [
            "sagemaker:*"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogGroup",
        "logs:DeleteLogStream",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:PutRetentionPolicy"
      ],
```

```
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/apigateway/AccessLogs/*",
        "arn:aws:logs:*:*:log-group::log-stream:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:DeleteBucketPolicy",
        "s3:GetBucketPolicy",
        "s3:PutBucketAcl",
        "s3:PutBucketNotification",
        "s3:PutBucketPolicy",
        "s3:PutBucketPublicAccessBlock",
        "s3:PutBucketLogging",
        "s3:PutEncryptionConfiguration",
        "s3:PutBucketTagging",
        "s3:PutObjectTagging",
        "s3:PutBucketCORS"
      ],
      "Resource": "arn:aws:s3:::sagemaker-*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:CreateModel",
        "sagemaker:CreateWorkteam",
        "sagemaker:DeleteEndpoint",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DeleteModel",
        "sagemaker:DeleteWorkteam",
        "sagemaker:DescribeModel",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeWorkteam",
        "sagemaker:CreateCodeRepository",
        "sagemaker:DescribeCodeRepository",
        "sagemaker:UpdateCodeRepository",
        "sagemaker:DeleteCodeRepository"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:AddTags"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:endpoint/*",
        "arn:aws:sagemaker:*:*:endpoint-config/*",
        "arn:aws:sagemaker:*:*:model/*",
        "arn:aws:sagemaker:*:*:pipeline/*",
        "arn:aws:sagemaker:*:*:project/*",
        "arn:aws:sagemaker:*:*:model-package/*"
      ],
      "Condition": {
        "ForAllValues:StringLike": {
          "aws:TagKeys": [
            "sagemaker:*"
          ]
        }
      }
  },
  {
      "Effect": "Allow",
```

```
      "Action": [
        "sagemaker:CreateImage",
        "sagemaker:DeleteImage",
        "sagemaker:DescribeImage",
        "sagemaker:UpdateImage",
        "sagemaker:ListTags"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:image/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:CreateStateMachine",
        "states:DeleteStateMachine",
        "states:UpdateStateMachine"
      ],
      "Resource": [
        "arn:aws:states:*:*:stateMachine:sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "codestar-connections:PassConnection",
      "Resource": "arn:aws:codestar-connections:*:*:connection/*",
      "Condition": {
        "StringEquals": {
          "codestar-connections:PassedToService": "codepipeline.amazonaws.com"
        }
      }
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerPartnerServiceCatalogProducts ApiGatewayServiceRolePolicy

Kebijakan ini digunakan oleh Amazon API Gateway dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan ke peran IAM yang diteruskan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh API Gateway yang memerlukan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `lambda`— Memanggil fungsi yang dibuat oleh template mitra.
- `sagemaker`— Memanggil titik akhir yang dibuat oleh template mitra.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:*:*:function:sagemaker-*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/sagemaker:project-name": "false",
          "aws:ResourceTag/sagemaker:partner": "false"
        },
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "sagemaker:InvokeEndpoint",
      "Resource": "arn:aws:sagemaker:*:*:endpoint/*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/sagemaker:project-name": "false",
          "aws:ResourceTag/sagemaker:partner": "false"
        },
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerPartnerServiceCatalogProducts CloudFormationServiceRolePolicy

Kebijakan ini digunakan oleh AWS CloudFormation dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM yang diberikan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh AWS CloudFormation yang membutuhkan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `iam`— Lulus `AmazonSageMakerServiceCatalogProductsLambdaRole` dan `AmazonSageMakerServiceCatalogProductsApiGatewayRole` peran.
- `lambda`— Buat, perbarui, hapus, dan panggil AWS Lambda fungsi; mengambil, menerbitkan, dan menghapus versi lapisan Lambda.
- `apigateway`— Buat, perbarui, dan hapus sumber daya Amazon API Gateway.
- `s3`— Ambil `lambda-auth-code/layer.zip` file dari bucket Amazon Simple Storage Service (Amazon S3)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsLambdaRole"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "lambda.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsApiGatewayRole"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "apigateway.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:DeleteFunction",
        "lambda:UpdateFunctionCode",
        "lambda:ListTags",
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:*:*:function:sagemaker-*"
      ],
      "Condition": {
        "Null": {
          "aws:ResourceTag/sagemaker:project-name": "false",
          "aws:ResourceTag/sagemaker:partner": "false"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:TagResource"
      ],
      "Resource": [
        "arn:aws:lambda:*:*:function:sagemaker-*"
      ],
      "Condition": {
        "Null": {
          "aws:ResourceTag/sagemaker:project-name": "false",
          "aws:ResourceTag/sagemaker:partner": "false"
```

```
        },
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": [
            "sagemaker:project-name",
            "sagemaker:partner"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:PublishLayerVersion",
        "lambda:GetLayerVersion",
        "lambda:DeleteLayerVersion",
        "lambda:GetFunction"
      ],
      "Resource": [
        "arn:aws:lambda:*:*:layer:sagemaker-*",
        "arn:aws:lambda:*:*:function:sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET",
        "apigateway:DELETE",
        "apigateway:PATCH",
        "apigateway:POST",
        "apigateway:PUT"
      ],
      "Resource": [
        "arn:aws:apigateway:*::/restapis/*",
        "arn:aws:apigateway:*::/restapis"
      ],
      "Condition": {
        "Null": {
          "aws:ResourceTag/sagemaker:project-name": "false",
          "aws:ResourceTag/sagemaker:partner": "false"
        }
      }
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "apigateway:POST",
        "apigateway:PUT"
      ],
      "Resource": [
        "arn:aws:apigateway:*::/restapis",
        "arn:aws:apigateway:*::/tags/*"
      ],
      "Condition": {
        "Null": {
          "aws:ResourceTag/sagemaker:project-name": "false",
          "aws:ResourceTag/sagemaker:partner": "false"
        },
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": [
            "sagemaker:project-name",
            "sagemaker:partner"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::sagemaker-*/lambda-auth-code/layer.zip"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerPartnerServiceCatalogProducts LambdaServiceRolePolicy

Kebijakan ini digunakan oleh AWS Lambda dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran

IAM yang diberikan Peran ke AWS sumber daya yang dibuat oleh Lambda yang membutuhkan AmazonSageMakerServiceCatalogProductsLaunchperan.

Detail izin

Kebijakan ini mencakup izin berikut.

- `secretsmanager`— Ambil data dari rahasia yang disediakan mitra untuk template mitra.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:*:*:secret:*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/sagemaker:partner": false
        },
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsApiGatewayServiceRolePolicy

Kebijakan ini digunakan oleh Amazon API Gateway dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan ke peran IAM yang diteruskan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh API Gateway yang memerlukan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `logs`— Buat dan baca grup CloudWatch Log, aliran, dan acara; perbarui acara; jelaskan berbagai sumber daya.

Izin ini terbatas pada sumber daya yang awalan grup lognya dimulai dengan “aws/apigateway/”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogDelivery",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:DescribeResourcePolicies",
        "logs:DescribeDestinations",
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:DescribeQueries",
        "logs:DescribeQueryDefinitions",
        "logs:DescribeSubscriptionFilters",
        "logs:GetLogDelivery",
        "logs:GetLogEvents",
        "logs:PutLogEvents",
        "logs:PutResourcePolicy",
        "logs:UpdateLogDelivery"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/apigateway/*"
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsCloudformation ServiceRolePolicy

Kebijakan ini digunakan oleh AWS CloudFormation dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM yang diberikan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh AWS CloudFormation yang membutuhkan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `sagemaker`— Izinkan akses ke berbagai SageMaker sumber daya tidak termasuk domain, profil pengguna, aplikasi, dan definisi aliran.
- `iam`— Lulus `AmazonSageMakerServiceCatalogProductsCodeBuildRole` dan `AmazonSageMakerServiceCatalogProductsExecutionRole` peran.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:AddAssociation",
        "sagemaker:AddTags",
        "sagemaker:AssociateTrialComponent",
        "sagemaker:BatchDescribeModelPackage",
        "sagemaker:BatchGetMetrics",
        "sagemaker:BatchGetRecord",
        "sagemaker:BatchPutMetrics",
        "sagemaker:CreateAction",
        "sagemaker:CreateAlgorithm",
        "sagemaker:CreateApp",
        "sagemaker:CreateAppImageConfig",
        "sagemaker:CreateArtifact",
        "sagemaker:CreateAutoMLJob",
        "sagemaker:CreateCodeRepository",
        "sagemaker:CreateCompilationJob",
        "sagemaker:CreateContext",
        "sagemaker:CreateDataQualityJobDefinition",
        "sagemaker:CreateDeviceFleet",
        "sagemaker:CreateDomain",
        "sagemaker:CreateEdgePackagingJob",
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:CreateExperiment",
        "sagemaker:CreateFeatureGroup",
        "sagemaker:CreateFlowDefinition",
        "sagemaker:CreateHumanTaskUi",
        "sagemaker:CreateHyperParameterTuningJob",
        "sagemaker:CreateImage",
        "sagemaker:CreateImageVersion",
```

```
        "sagemaker:CreateInferenceRecommendationsJob",
        "sagemaker:CreateLabelingJob",
        "sagemaker:CreateLineageGroupPolicy",
        "sagemaker:CreateModel",
        "sagemaker:CreateModelBiasJobDefinition",
        "sagemaker:CreateModelExplainabilityJobDefinition",
        "sagemaker:CreateModelPackage",
        "sagemaker:CreateModelPackageGroup",
        "sagemaker:CreateModelQualityJobDefinition",
        "sagemaker:CreateMonitoringSchedule",
        "sagemaker:CreateNotebookInstance",
        "sagemaker:CreateNotebookInstanceLifecycleConfig",
        "sagemaker:CreatePipeline",
        "sagemaker:CreatePresignedDomainUrl",
        "sagemaker:CreatePresignedNotebookInstanceUrl",
        "sagemaker:CreateProcessingJob",
        "sagemaker:CreateProject",
        "sagemaker:CreateTrainingJob",
        "sagemaker:CreateTransformJob",
        "sagemaker:CreateTrial",
        "sagemaker:CreateTrialComponent",
        "sagemaker:CreateUserProfile",
        "sagemaker:CreateWorkforce",
        "sagemaker:CreateWorkteam",
        "sagemaker:DeleteAction",
        "sagemaker:DeleteAlgorithm",
        "sagemaker:DeleteApp",
        "sagemaker:DeleteAppImageConfig",
        "sagemaker:DeleteArtifact",
        "sagemaker:DeleteAssociation",
        "sagemaker:DeleteCodeRepository",
        "sagemaker:DeleteContext",
        "sagemaker:DeleteDataQualityJobDefinition",
        "sagemaker:DeleteDeviceFleet",
        "sagemaker:DeleteDomain",
        "sagemaker:DeleteEndpoint",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DeleteExperiment",
        "sagemaker:DeleteFeatureGroup",
        "sagemaker:DeleteFlowDefinition",
        "sagemaker:DeleteHumanLoop",
        "sagemaker:DeleteHumanTaskUi",
        "sagemaker:DeleteImage",
        "sagemaker:DeleteImageVersion",
```

```
        "sagemaker:DeleteLineageGroupPolicy",
        "sagemaker:DeleteModel",
        "sagemaker:DeleteModelBiasJobDefinition",
        "sagemaker:DeleteModelExplainabilityJobDefinition",
        "sagemaker:DeleteModelPackage",
        "sagemaker:DeleteModelPackageGroup",
        "sagemaker:DeleteModelPackageGroupPolicy",
        "sagemaker:DeleteModelQualityJobDefinition",
        "sagemaker:DeleteMonitoringSchedule",
        "sagemaker:DeleteNotebookInstance",
        "sagemaker:DeleteNotebookInstanceLifecycleConfig",
        "sagemaker:DeletePipeline",
        "sagemaker:DeleteProject",
        "sagemaker:DeleteRecord",
        "sagemaker:DeleteTags",
        "sagemaker:DeleteTrial",
        "sagemaker:DeleteTrialComponent",
        "sagemaker:DeleteUserProfile",
        "sagemaker:DeleteWorkforce",
        "sagemaker:DeleteWorkteam",
        "sagemaker:DeregisterDevices",
        "sagemaker:DescribeAction",
        "sagemaker:DescribeAlgorithm",
        "sagemaker:DescribeApp",
        "sagemaker:DescribeAppImageConfig",
        "sagemaker:DescribeArtifact",
        "sagemaker:DescribeAutoMLJob",
        "sagemaker:DescribeCodeRepository",
        "sagemaker:DescribeCompilationJob",
        "sagemaker:DescribeContext",
        "sagemaker:DescribeDataQualityJobDefinition",
        "sagemaker:DescribeDevice",
        "sagemaker:DescribeDeviceFleet",
        "sagemaker:DescribeDomain",
        "sagemaker:DescribeEdgePackagingJob",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeExperiment",
        "sagemaker:DescribeFeatureGroup",
        "sagemaker:DescribeFlowDefinition",
        "sagemaker:DescribeHumanLoop",
        "sagemaker:DescribeHumanTaskUi",
        "sagemaker:DescribeHyperParameterTuningJob",
        "sagemaker:DescribeImage",
```

```
        "sagemaker:DescribeImageVersion",
        "sagemaker:DescribeInferenceRecommendationsJob",
        "sagemaker:DescribeLabelingJob",
        "sagemaker:DescribeLineageGroup",
        "sagemaker:DescribeModel",
        "sagemaker:DescribeModelBiasJobDefinition",
        "sagemaker:DescribeModelExplainabilityJobDefinition",
        "sagemaker:DescribeModelPackage",
        "sagemaker:DescribeModelPackageGroup",
        "sagemaker:DescribeModelQualityJobDefinition",
        "sagemaker:DescribeMonitoringSchedule",
        "sagemaker:DescribeNotebookInstance",
        "sagemaker:DescribeNotebookInstanceLifecycleConfig",
        "sagemaker:DescribePipeline",
        "sagemaker:DescribePipelineDefinitionForExecution",
        "sagemaker:DescribePipelineExecution",
        "sagemaker:DescribeProcessingJob",
        "sagemaker:DescribeProject",
        "sagemaker:DescribeSubscribedWorkteam",
        "sagemaker:DescribeTrainingJob",
        "sagemaker:DescribeTransformJob",
        "sagemaker:DescribeTrial",
        "sagemaker:DescribeTrialComponent",
        "sagemaker:DescribeUserProfile",
        "sagemaker:DescribeWorkforce",
        "sagemaker:DescribeWorkteam",
        "sagemaker:DisableSagemakerServicecatalogPortfolio",
        "sagemaker:DisassociateTrialComponent",
        "sagemaker:EnableSagemakerServicecatalogPortfolio",
        "sagemaker:GetDeviceFleetReport",
        "sagemaker:GetDeviceRegistration",
        "sagemaker:GetLineageGroupPolicy",
        "sagemaker:GetModelPackageGroupPolicy",
        "sagemaker:GetRecord",
        "sagemaker:GetSagemakerServicecatalogPortfolioStatus",
        "sagemaker:GetSearchSuggestions",
        "sagemaker:InvokeEndpoint",
        "sagemaker:InvokeEndpointAsync",
        "sagemaker:ListActions",
        "sagemaker:ListAlgorithms",
        "sagemaker:ListAppImageConfigs",
        "sagemaker:ListApps",
        "sagemaker:ListArtifacts",
        "sagemaker:ListAssociations",
```

```
        "sagemaker:ListAutoMLJobs",
        "sagemaker:ListCandidatesForAutoMLJob",
        "sagemaker:ListCodeRepositories",
        "sagemaker:ListCompilationJobs",
        "sagemaker:ListContexts",
        "sagemaker:ListDataQualityJobDefinitions",
        "sagemaker:ListDeviceFleets",
        "sagemaker:ListDevices",
        "sagemaker:ListDomains",
        "sagemaker:ListEdgePackagingJobs",
        "sagemaker:ListEndpointConfigs",
        "sagemaker:ListEndpoints",
        "sagemaker:ListExperiments",
        "sagemaker:ListFeatureGroups",
        "sagemaker:ListFlowDefinitions",
        "sagemaker:ListHumanLoops",
        "sagemaker:ListHumanTaskUis",
        "sagemaker:ListHyperParameterTuningJobs",
        "sagemaker:ListImageVersions",
        "sagemaker:ListImages",
        "sagemaker:ListInferenceRecommendationsJobs",
        "sagemaker:ListLabelingJobs",
        "sagemaker:ListLabelingJobsForWorkteam",
        "sagemaker:ListLineageGroups",
        "sagemaker:ListModelBiasJobDefinitions",
        "sagemaker:ListModelExplainabilityJobDefinitions",
        "sagemaker:ListModelMetadata",
        "sagemaker:ListModelPackageGroups",
        "sagemaker:ListModelPackages",
        "sagemaker:ListModelQualityJobDefinitions",
        "sagemaker:ListModels",
        "sagemaker:ListMonitoringExecutions",
        "sagemaker:ListMonitoringSchedules",
        "sagemaker:ListNotebookInstanceLifecycleConfigs",
        "sagemaker:ListNotebookInstances",
        "sagemaker:ListPipelineExecutionSteps",
        "sagemaker:ListPipelineExecutions",
        "sagemaker:ListPipelineParametersForExecution",
        "sagemaker:ListPipelines",
        "sagemaker:ListProcessingJobs",
        "sagemaker:ListProjects",
        "sagemaker:ListSubscribedWorkteams",
        "sagemaker:ListTags",
        "sagemaker:ListTrainingJobs",
```

```
        "sagemaker:ListTrainingJobsForHyperParameterTuningJob",
        "sagemaker:ListTransformJobs",
        "sagemaker:ListTrialComponents",
        "sagemaker:ListTrials",
        "sagemaker:ListUserProfiles",
        "sagemaker:ListWorkforces",
        "sagemaker:ListWorkteams",
        "sagemaker:PutLineageGroupPolicy",
        "sagemaker:PutModelPackageGroupPolicy",
        "sagemaker:PutRecord",
        "sagemaker:QueryLineage",
        "sagemaker:RegisterDevices",
        "sagemaker:RenderUiTemplate",
        "sagemaker:Search",
        "sagemaker:SendHeartbeat",
        "sagemaker:SendPipelineExecutionStepFailure",
        "sagemaker:SendPipelineExecutionStepSuccess",
        "sagemaker:StartHumanLoop",
        "sagemaker:StartMonitoringSchedule",
        "sagemaker:StartNotebookInstance",
        "sagemaker:StartPipelineExecution",
        "sagemaker:StopAutoMLJob",
        "sagemaker:StopCompilationJob",
        "sagemaker:StopEdgePackagingJob",
        "sagemaker:StopHumanLoop",
        "sagemaker:StopHyperParameterTuningJob",
        "sagemaker:StopInferenceRecommendationsJob",
        "sagemaker:StopLabelingJob",
        "sagemaker:StopMonitoringSchedule",
        "sagemaker:StopNotebookInstance",
        "sagemaker:StopPipelineExecution",
        "sagemaker:StopProcessingJob",
        "sagemaker:StopTrainingJob",
        "sagemaker:StopTransformJob",
        "sagemaker:UpdateAction",
        "sagemaker:UpdateAppImageConfig",
        "sagemaker:UpdateArtifact",
        "sagemaker:UpdateCodeRepository",
        "sagemaker:UpdateContext",
        "sagemaker:UpdateDeviceFleet",
        "sagemaker:UpdateDevices",
        "sagemaker:UpdateDomain",
        "sagemaker:UpdateEndpoint",
        "sagemaker:UpdateEndpointWeightsAndCapacities",
```

```
        "sagemaker:UpdateExperiment",
        "sagemaker:UpdateImage",
        "sagemaker:UpdateModelPackage",
        "sagemaker:UpdateMonitoringSchedule",
        "sagemaker:UpdateNotebookInstance",
        "sagemaker:UpdateNotebookInstanceLifecycleConfig",
        "sagemaker:UpdatePipeline",
        "sagemaker:UpdatePipelineExecution",
        "sagemaker:UpdateProject",
        "sagemaker:UpdateTrainingJob",
        "sagemaker:UpdateTrial",
        "sagemaker:UpdateTrialComponent",
        "sagemaker:UpdateUserProfile",
        "sagemaker:UpdateWorkforce",
        "sagemaker:UpdateWorkteam"
      ],
      "NotResource": [
        "arn:aws:sagemaker:*:*:domain/*",
        "arn:aws:sagemaker:*:*:user-profile/*",
        "arn:aws:sagemaker:*:*:app/*",
        "arn:aws:sagemaker:*:*:flow-definition/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsCodeBuildRole",
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsExecutionRole"
      ]
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsCode BuildServiceRolePolicy

Kebijakan ini digunakan oleh AWS CodeBuild dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM

yang diberikan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh CodeBuild yang membutuhkan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `sagemaker`— Izinkan akses ke berbagai SageMaker sumber daya.
- `codecommit`— Unggah CodeCommit arsip ke CodeBuild pipeline, dapatkan status unggah, dan batalkan unggahan; dapatkan informasi cabang dan komit. Izin ini terbatas pada sumber daya yang namanya dimulai dengan “sagemaker-”.
- `ecr`— Buat repositori Amazon ECR dan gambar kontainer; unggah lapisan gambar. Izin ini terbatas pada repositori yang namanya dimulai dengan “sagemaker-”.

  `ecr`— Baca semua sumber daya.
- `iam`— Lulus peran berikut:
  - `AmazonSageMakerServiceCatalogProductsCloudformationRole`kepada AWS CloudFormation.
  - `AmazonSageMakerServiceCatalogProductsCodeBuildRole`kepada AWS CodeBuild.
  - `AmazonSageMakerServiceCatalogProductsCodePipelineRole`kepada AWS CodePipeline.
  - `AmazonSageMakerServiceCatalogProductsEventsRole`kepada Amazon EventBridge.
  - `AmazonSageMakerServiceCatalogProductsExecutionRole`kepada Amazon SageMaker.
- `logs`— Buat dan baca grup CloudWatch Log, aliran, dan acara; perbarui acara; jelaskan berbagai sumber daya.

  Izin ini terbatas pada sumber daya yang awalan namanya dimulai dengan “aws/codebuild/”.
- `s3`— Membuat, membaca, dan membuat daftar bucket Amazon S3. Izin ini terbatas pada bucket yang namanya dimulai dengan “sagemaker-”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:CancelUploadArchive",
        "codecommit:GetBranch",
```

```
        "codecommit:GetCommit",
        "codecommit:GetUploadArchiveStatus",
        "codecommit:UploadArchive"
      ],
      "Resource": "arn:aws:codecommit:*:*:sagemaker-*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:DescribeImageScanFindings",
        "ecr:DescribeRegistry",
        "ecr:DescribeImageReplicationStatus",
        "ecr:DescribeRepositories",
        "ecr:DescribeImageReplicationStatus",
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CompleteLayerUpload",
        "ecr:CreateRepository",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": [
        "arn:aws:ecr:*:*:repository/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsEventsRole",
```

```
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsCodePipelineRole",
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsCloudformationRole",
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsCodeBuildRole",
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsExecutionRole"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "events.amazonaws.com",
            "codepipeline.amazonaws.com",
            "cloudformation.amazonaws.com",
            "codebuild.amazonaws.com",
            "sagemaker.amazonaws.com"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogDelivery",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:DescribeResourcePolicies",
        "logs:DescribeDestinations",
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:DescribeQueries",
        "logs:DescribeQueryDefinitions",
        "logs:DescribeSubscriptionFilters",
        "logs:GetLogDelivery",
        "logs:GetLogEvents",
        "logs:ListLogDeliveries",
        "logs:PutLogEvents",
        "logs:PutResourcePolicy",
        "logs:UpdateLogDelivery"
      ],
```

```
      "Resource": "arn:aws:logs:*:*:log-group:/aws/codebuild/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:GetBucketAcl",
        "s3:GetBucketCors",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutBucketCors",
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue-*",
        "arn:aws:s3:::sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:AddAssociation",
        "sagemaker:AddTags",
        "sagemaker:AssociateTrialComponent",
        "sagemaker:BatchDescribeModelPackage",
        "sagemaker:BatchGetMetrics",
        "sagemaker:BatchGetRecord",
        "sagemaker:BatchPutMetrics",
        "sagemaker:CreateAction",
        "sagemaker:CreateAlgorithm",
        "sagemaker:CreateApp",
        "sagemaker:CreateAppImageConfig",
        "sagemaker:CreateArtifact",
        "sagemaker:CreateAutoMLJob",
        "sagemaker:CreateCodeRepository",
        "sagemaker:CreateCompilationJob",
        "sagemaker:CreateContext",
```

```
        "sagemaker:CreateDataQualityJobDefinition",
        "sagemaker:CreateDeviceFleet",
        "sagemaker:CreateDomain",
        "sagemaker:CreateEdgePackagingJob",
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:CreateExperiment",
        "sagemaker:CreateFeatureGroup",
        "sagemaker:CreateFlowDefinition",
        "sagemaker:CreateHumanTaskUi",
        "sagemaker:CreateHyperParameterTuningJob",
        "sagemaker:CreateImage",
        "sagemaker:CreateImageVersion",
        "sagemaker:CreateInferenceRecommendationsJob",
        "sagemaker:CreateLabelingJob",
        "sagemaker:CreateLineageGroupPolicy",
        "sagemaker:CreateModel",
        "sagemaker:CreateModelBiasJobDefinition",
        "sagemaker:CreateModelExplainabilityJobDefinition",
        "sagemaker:CreateModelPackage",
        "sagemaker:CreateModelPackageGroup",
        "sagemaker:CreateModelQualityJobDefinition",
        "sagemaker:CreateMonitoringSchedule",
        "sagemaker:CreateNotebookInstance",
        "sagemaker:CreateNotebookInstanceLifecycleConfig",
        "sagemaker:CreatePipeline",
        "sagemaker:CreatePresignedDomainUrl",
        "sagemaker:CreatePresignedNotebookInstanceUrl",
        "sagemaker:CreateProcessingJob",
        "sagemaker:CreateProject",
        "sagemaker:CreateTrainingJob",
        "sagemaker:CreateTransformJob",
        "sagemaker:CreateTrial",
        "sagemaker:CreateTrialComponent",
        "sagemaker:CreateUserProfile",
        "sagemaker:CreateWorkforce",
        "sagemaker:CreateWorkteam",
        "sagemaker:DeleteAction",
        "sagemaker:DeleteAlgorithm",
        "sagemaker:DeleteApp",
        "sagemaker:DeleteAppImageConfig",
        "sagemaker:DeleteArtifact",
        "sagemaker:DeleteAssociation",
        "sagemaker:DeleteCodeRepository",
```

```
        "sagemaker:DeleteContext",
        "sagemaker:DeleteDataQualityJobDefinition",
        "sagemaker:DeleteDeviceFleet",
        "sagemaker:DeleteDomain",
        "sagemaker:DeleteEndpoint",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DeleteExperiment",
        "sagemaker:DeleteFeatureGroup",
        "sagemaker:DeleteFlowDefinition",
        "sagemaker:DeleteHumanLoop",
        "sagemaker:DeleteHumanTaskUi",
        "sagemaker:DeleteImage",
        "sagemaker:DeleteImageVersion",
        "sagemaker:DeleteLineageGroupPolicy",
        "sagemaker:DeleteModel",
        "sagemaker:DeleteModelBiasJobDefinition",
        "sagemaker:DeleteModelExplainabilityJobDefinition",
        "sagemaker:DeleteModelPackage",
        "sagemaker:DeleteModelPackageGroup",
        "sagemaker:DeleteModelPackageGroupPolicy",
        "sagemaker:DeleteModelQualityJobDefinition",
        "sagemaker:DeleteMonitoringSchedule",
        "sagemaker:DeleteNotebookInstance",
        "sagemaker:DeleteNotebookInstanceLifecycleConfig",
        "sagemaker:DeletePipeline",
        "sagemaker:DeleteProject",
        "sagemaker:DeleteRecord",
        "sagemaker:DeleteTags",
        "sagemaker:DeleteTrial",
        "sagemaker:DeleteTrialComponent",
        "sagemaker:DeleteUserProfile",
        "sagemaker:DeleteWorkforce",
        "sagemaker:DeleteWorkteam",
        "sagemaker:DeregisterDevices",
        "sagemaker:DescribeAction",
        "sagemaker:DescribeAlgorithm",
        "sagemaker:DescribeApp",
        "sagemaker:DescribeAppImageConfig",
        "sagemaker:DescribeArtifact",
        "sagemaker:DescribeAutoMLJob",
        "sagemaker:DescribeCodeRepository",
        "sagemaker:DescribeCompilationJob",
        "sagemaker:DescribeContext",
        "sagemaker:DescribeDataQualityJobDefinition",
```

```
        "sagemaker:DescribeDevice",
        "sagemaker:DescribeDeviceFleet",
        "sagemaker:DescribeDomain",
        "sagemaker:DescribeEdgePackagingJob",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeExperiment",
        "sagemaker:DescribeFeatureGroup",
        "sagemaker:DescribeFlowDefinition",
        "sagemaker:DescribeHumanLoop",
        "sagemaker:DescribeHumanTaskUi",
        "sagemaker:DescribeHyperParameterTuningJob",
        "sagemaker:DescribeImage",
        "sagemaker:DescribeImageVersion",
        "sagemaker:DescribeInferenceRecommendationsJob",
        "sagemaker:DescribeLabelingJob",
        "sagemaker:DescribeLineageGroup",
        "sagemaker:DescribeModel",
        "sagemaker:DescribeModelBiasJobDefinition",
        "sagemaker:DescribeModelExplainabilityJobDefinition",
        "sagemaker:DescribeModelPackage",
        "sagemaker:DescribeModelPackageGroup",
        "sagemaker:DescribeModelQualityJobDefinition",
        "sagemaker:DescribeMonitoringSchedule",
        "sagemaker:DescribeNotebookInstance",
        "sagemaker:DescribeNotebookInstanceLifecycleConfig",
        "sagemaker:DescribePipeline",
        "sagemaker:DescribePipelineDefinitionForExecution",
        "sagemaker:DescribePipelineExecution",
        "sagemaker:DescribeProcessingJob",
        "sagemaker:DescribeProject",
        "sagemaker:DescribeSubscribedWorkteam",
        "sagemaker:DescribeTrainingJob",
        "sagemaker:DescribeTransformJob",
        "sagemaker:DescribeTrial",
        "sagemaker:DescribeTrialComponent",
        "sagemaker:DescribeUserProfile",
        "sagemaker:DescribeWorkforce",
        "sagemaker:DescribeWorkteam",
        "sagemaker:DisableSagemakerServicecatalogPortfolio",
        "sagemaker:DisassociateTrialComponent",
        "sagemaker:EnableSagemakerServicecatalogPortfolio",
        "sagemaker:GetDeviceFleetReport",
        "sagemaker:GetDeviceRegistration",
```

```
        "sagemaker:GetLineageGroupPolicy",
        "sagemaker:GetModelPackageGroupPolicy",
        "sagemaker:GetRecord",
        "sagemaker:GetSagemakerServicecatalogPortfolioStatus",
        "sagemaker:GetSearchSuggestions",
        "sagemaker:InvokeEndpoint",
        "sagemaker:InvokeEndpointAsync",
        "sagemaker:ListActions",
        "sagemaker:ListAlgorithms",
        "sagemaker:ListAppImageConfigs",
        "sagemaker:ListApps",
        "sagemaker:ListArtifacts",
        "sagemaker:ListAssociations",
        "sagemaker:ListAutoMLJobs",
        "sagemaker:ListCandidatesForAutoMLJob",
        "sagemaker:ListCodeRepositories",
        "sagemaker:ListCompilationJobs",
        "sagemaker:ListContexts",
        "sagemaker:ListDataQualityJobDefinitions",
        "sagemaker:ListDeviceFleets",
        "sagemaker:ListDevices",
        "sagemaker:ListDomains",
        "sagemaker:ListEdgePackagingJobs",
        "sagemaker:ListEndpointConfigs",
        "sagemaker:ListEndpoints",
        "sagemaker:ListExperiments",
        "sagemaker:ListFeatureGroups",
        "sagemaker:ListFlowDefinitions",
        "sagemaker:ListHumanLoops",
        "sagemaker:ListHumanTaskUis",
        "sagemaker:ListHyperParameterTuningJobs",
        "sagemaker:ListImageVersions",
        "sagemaker:ListImages",
        "sagemaker:ListInferenceRecommendationsJobs",
        "sagemaker:ListLabelingJobs",
        "sagemaker:ListLabelingJobsForWorkteam",
        "sagemaker:ListLineageGroups",
        "sagemaker:ListModelBiasJobDefinitions",
        "sagemaker:ListModelExplainabilityJobDefinitions",
        "sagemaker:ListModelMetadata",
        "sagemaker:ListModelPackageGroups",
        "sagemaker:ListModelPackages",
        "sagemaker:ListModelQualityJobDefinitions",
        "sagemaker:ListModels",
```

```
        "sagemaker:ListMonitoringExecutions",
        "sagemaker:ListMonitoringSchedules",
        "sagemaker:ListNotebookInstanceLifecycleConfigs",
        "sagemaker:ListNotebookInstances",
        "sagemaker:ListPipelineExecutionSteps",
        "sagemaker:ListPipelineExecutions",
        "sagemaker:ListPipelineParametersForExecution",
        "sagemaker:ListPipelines",
        "sagemaker:ListProcessingJobs",
        "sagemaker:ListProjects",
        "sagemaker:ListSubscribedWorkteams",
        "sagemaker:ListTags",
        "sagemaker:ListTrainingJobs",
        "sagemaker:ListTrainingJobsForHyperParameterTuningJob",
        "sagemaker:ListTransformJobs",
        "sagemaker:ListTrialComponents",
        "sagemaker:ListTrials",
        "sagemaker:ListUserProfiles",
        "sagemaker:ListWorkforces",
        "sagemaker:ListWorkteams",
        "sagemaker:PutLineageGroupPolicy",
        "sagemaker:PutModelPackageGroupPolicy",
        "sagemaker:PutRecord",
        "sagemaker:QueryLineage",
        "sagemaker:RegisterDevices",
        "sagemaker:RenderUiTemplate",
        "sagemaker:Search",
        "sagemaker:SendHeartbeat",
        "sagemaker:SendPipelineExecutionStepFailure",
        "sagemaker:SendPipelineExecutionStepSuccess",
        "sagemaker:StartHumanLoop",
        "sagemaker:StartMonitoringSchedule",
        "sagemaker:StartNotebookInstance",
        "sagemaker:StartPipelineExecution",
        "sagemaker:StopAutoMLJob",
        "sagemaker:StopCompilationJob",
        "sagemaker:StopEdgePackagingJob",
        "sagemaker:StopHumanLoop",
        "sagemaker:StopHyperParameterTuningJob",
        "sagemaker:StopInferenceRecommendationsJob",
        "sagemaker:StopLabelingJob",
        "sagemaker:StopMonitoringSchedule",
        "sagemaker:StopNotebookInstance",
        "sagemaker:StopPipelineExecution",
```

```
        "sagemaker:StopProcessingJob",
        "sagemaker:StopTrainingJob",
        "sagemaker:StopTransformJob",
        "sagemaker:UpdateAction",
        "sagemaker:UpdateAppImageConfig",
        "sagemaker:UpdateArtifact",
        "sagemaker:UpdateCodeRepository",
        "sagemaker:UpdateContext",
        "sagemaker:UpdateDeviceFleet",
        "sagemaker:UpdateDevices",
        "sagemaker:UpdateDomain",
        "sagemaker:UpdateEndpoint",
        "sagemaker:UpdateEndpointWeightsAndCapacities",
        "sagemaker:UpdateExperiment",
        "sagemaker:UpdateImage",
        "sagemaker:UpdateModelPackage",
        "sagemaker:UpdateMonitoringSchedule",
        "sagemaker:UpdateNotebookInstance",
        "sagemaker:UpdateNotebookInstanceLifecycleConfig",
        "sagemaker:UpdatePipeline",
        "sagemaker:UpdatePipelineExecution",
        "sagemaker:UpdateProject",
        "sagemaker:UpdateTrainingJob",
        "sagemaker:UpdateTrial",
        "sagemaker:UpdateTrialComponent",
        "sagemaker:UpdateUserProfile",
        "sagemaker:UpdateWorkforce",
        "sagemaker:UpdateWorkteam"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:endpoint/*",
        "arn:aws:sagemaker:*:*:endpoint-config/*",
        "arn:aws:sagemaker:*:*:model/*",
        "arn:aws:sagemaker:*:*:pipeline/*",
        "arn:aws:sagemaker:*:*:project/*",
        "arn:aws:sagemaker:*:*:model-package/*"
      ]
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsCode PipelineServiceRolePolicy

Kebijakan ini digunakan oleh AWS CodePipeline dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM yang diberikan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh CodePipeline yang membutuhkan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `cloudformation`— Buat, baca, hapus, dan perbarui CloudFormation tumpukan; buat, baca, hapus, dan jalankan set perubahan; atur kebijakan tumpukan. Izin ini terbatas pada sumber daya yang namanya dimulai dengan “sagemaker-”.
- `s3`— Buat, baca, daftar, dan hapus bucket Amazon S3; menambah, membaca, dan menghapus objek dari bucket; membaca dan mengatur konfigurasi CORS; baca daftar kontrol akses (ACL); dan baca Wilayah tempat AWS bucket berada.

  Izin ini terbatas pada ember yang namanya dimulai dengan “sagemaker-” atau “aws-glue-.
- `iam`— Lulus `AmazonSageMakerServiceCatalogProductsCloudformationRole` peran.
- `codebuild`— Dapatkan informasi CodeBuild build dan mulai membangun. Izin ini terbatas pada proyek dan membangun sumber daya yang namanya dimulai dengan “sagemaker-”.
- `codecommit`— Unggah CodeCommit arsip ke CodeBuild pipeline, dapatkan status unggah, dan batalkan unggahan; dapatkan informasi cabang dan komit.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation:CreateStack",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DeleteChangeSet",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStacks",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:SetStackPolicy",
```

```
        "cloudformation:UpdateStack"
      ],
      "Resource": "arn:aws:cloudformation:*:*:stack/sagemaker-*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/service-role/
AmazonSageMakerServiceCatalogProductsCloudformationRole"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:BatchGetBuilds",
        "codebuild:StartBuild"
      ],
      "Resource": [
        "arn:aws:codebuild:*:*:project/sagemaker-*",
        "arn:aws:codebuild:*:*:build/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:CancelUploadArchive",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
```

```
        "codecommit:GetUploadArchiveStatus",
        "codecommit:UploadArchive"
      ],
      "Resource": "arn:aws:codecommit:*:*:sagemaker-*"
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsEvents ServiceRolePolicy

Kebijakan ini digunakan oleh Amazon EventBridge dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM yang diberikan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh EventBridge yang membutuhkan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `codepipeline`— Mulai CodeBuild eksekusi. Izin ini terbatas pada saluran pipa yang namanya dimulai dengan “sagemaker-”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codepipeline:StartPipelineExecution",
      "Resource": "arn:aws:codepipeline:*:*:sagemaker-*"
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsFirehose ServiceRolePolicy

Kebijakan ini digunakan oleh Amazon Kinesis Data Firehose AWS Service Catalog dalam produk yang disediakan dari portofolio Amazon. SageMaker Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM yang diteruskan AmazonSageMakerServiceCatalogProductsLaunchPeran ke AWS sumber daya yang dibuat oleh Kinesis Data Firehose yang memerlukan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `firehose`— Kirim catatan Kinesis Data Firehose. Izin ini terbatas pada sumber daya yang nama aliran pengirimannya dimulai dengan “sagemaker-”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "firehose:PutRecord",
        "firehose:PutRecordBatch"
      ],
      "Resource": "arn:aws:firehose:*:*:deliverystream/sagemaker-*"
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsGlue ServiceRolePolicy

Kebijakan ini digunakan oleh AWS Glue dalam produk yang disediakan oleh AWS Service Catalog dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM yang diberikan AmazonSageMakerServiceCatalogProductsLaunch Peran ke AWS sumber daya yang dibuat oleh Glue yang memerlukan peran.

Detail izin

Kebijakan ini mencakup izin berikut.

- `glue`— Buat, baca, dan hapus AWS Glue partisi, tabel, dan versi tabel. Izin ini terbatas pada sumber daya yang namanya dimulai dengan “sagemaker-”. Buat dan baca database AWS Glue. Izin ini terbatas pada database yang namanya “default”, “global_temp”, atau dimulai dengan “sagemaker-”. Dapatkan fungsi yang ditentukan pengguna.
- `s3`— Buat, baca, daftar, dan hapus bucket Amazon S3; menambah, membaca, dan menghapus objek dari bucket; membaca dan mengatur konfigurasi CORS; baca daftar kontrol akses (ACL), dan baca Wilayah tempat AWS bucket berada.

  Izin ini terbatas pada ember yang namanya dimulai dengan “sagemaker-” atau “aws-glue-”.

- `logs`— Buat, baca, dan hapus grup CloudWatch log log, aliran, dan pengiriman; dan buat kebijakan sumber daya.

  Izin ini terbatas pada sumber daya yang awalan namanya dimulai dengan “aws/glue/”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:BatchCreatePartition",
        "glue:BatchDeletePartition",
        "glue:BatchDeleteTable",
        "glue:BatchDeleteTableVersion",
        "glue:BatchGetPartition",
        "glue:CreateDatabase",
        "glue:CreatePartition",
        "glue:CreateTable",
        "glue:DeletePartition",
        "glue:DeleteTable",
        "glue:DeleteTableVersion",
        "glue:GetDatabase",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetTableVersion",
        "glue:GetTableVersions",
        "glue:SearchTables",
        "glue:UpdatePartition",
        "glue:UpdateTable",
        "glue:GetUserDefinedFunctions"
      ],
      "Resource": [
        "arn:aws:glue:*:*:catalog",
        "arn:aws:glue:*:*:database/default",
        "arn:aws:glue:*:*:database/global_temp",
        "arn:aws:glue:*:*:database/sagemaker-*",
        "arn:aws:glue:*:*:table/sagemaker-*",
        "arn:aws:glue:*:*:tableVersion/sagemaker-*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:GetBucketAcl",
        "s3:GetBucketCors",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutBucketCors"
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue-*",
        "arn:aws:s3:::sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue-*",
        "arn:aws:s3:::sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogDelivery",
        "logs:Describe*",
        "logs:GetLogDelivery",
        "logs:GetLogEvents",
        "logs:ListLogDeliveries",
```

```
        "logs:PutLogEvents",
        "logs:PutResourcePolicy",
        "logs:UpdateLogDelivery"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/glue/*"
    }
  ]
}
```

AWSkebijakan terkelola: AmazonSageMakerServiceCatalogProductsLambda ServiceRolePolicy

Kebijakan ini digunakan oleh AWS Lambda dalam produk yang AWS Service Catalog disediakan dari portofolio Amazon SageMaker . Kebijakan ini dimaksudkan untuk dilampirkan pada peran IAM yang diberikan Peran ke AWS sumber daya yang dibuat oleh Lambda yang membutuhkan AmazonSageMakerServiceCatalogProductsLaunchperan.

Detail izin

Kebijakan ini mencakup izin berikut.

- `sagemaker`— Izinkan akses ke berbagai SageMaker sumber daya.
- `ecr`— Buat dan hapus repositori Amazon ECR; membuat, membaca, dan menghapus gambar kontainer; unggah lapisan gambar. Izin ini terbatas pada repositori yang namanya dimulai dengan “sagemaker-”.
- `events`— Buat, baca, dan hapus EventBridge aturan Amazon; dan buat dan hapus target. Izin ini terbatas pada aturan yang namanya dimulai dengan “sagemaker-”.
- `s3`— Buat, baca, daftar, dan hapus bucket Amazon S3; menambah, membaca, dan menghapus objek dari bucket; membaca dan mengatur konfigurasi CORS; baca daftar kontrol akses (ACL), dan baca Wilayah tempat AWS bucket berada.

  Izin ini terbatas pada ember yang namanya dimulai dengan “sagemaker-” atau “aws-glue-”.
- `iam`— Lulus `AmazonSageMakerServiceCatalogProductsExecutionRole` peran.
- `logs`— Buat, baca, dan hapus grup CloudWatch log log, aliran, dan pengiriman; dan buat kebijakan sumber daya.

  Izin ini terbatas pada sumber daya yang awalan namanya dimulai dengan “aws/lambda/”.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:DescribeImages",
        "ecr:BatchDeleteImage",
        "ecr:CompleteLayerUpload",
        "ecr:CreateRepository",
        "ecr:DeleteRepository",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": [
        "arn:aws:ecr:*:*:repository/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DeleteRule",
        "events:DescribeRule",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets"
      ],
      "Resource": [
        "arn:aws:events:*:*:rule/sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:GetBucketAcl",
        "s3:GetBucketCors",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutBucketCors"
      ],
      "Resource": [
```

```
        "arn:aws:s3:::aws-glue-*",
        "arn:aws:s3:::sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue-*",
        "arn:aws:s3:::sagemaker-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:AddAssociation",
        "sagemaker:AddTags",
        "sagemaker:AssociateTrialComponent",
        "sagemaker:BatchDescribeModelPackage",
        "sagemaker:BatchGetMetrics",
        "sagemaker:BatchGetRecord",
        "sagemaker:BatchPutMetrics",
        "sagemaker:CreateAction",
        "sagemaker:CreateAlgorithm",
        "sagemaker:CreateApp",
        "sagemaker:CreateAppImageConfig",
        "sagemaker:CreateArtifact",
        "sagemaker:CreateAutoMLJob",
        "sagemaker:CreateCodeRepository",
        "sagemaker:CreateCompilationJob",
        "sagemaker:CreateContext",
        "sagemaker:CreateDataQualityJobDefinition",
        "sagemaker:CreateDeviceFleet",
        "sagemaker:CreateDomain",
        "sagemaker:CreateEdgePackagingJob",
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:CreateExperiment",
```

```
        "sagemaker:CreateFeatureGroup",
        "sagemaker:CreateFlowDefinition",
        "sagemaker:CreateHumanTaskUi",
        "sagemaker:CreateHyperParameterTuningJob",
        "sagemaker:CreateImage",
        "sagemaker:CreateImageVersion",
        "sagemaker:CreateInferenceRecommendationsJob",
        "sagemaker:CreateLabelingJob",
        "sagemaker:CreateLineageGroupPolicy",
        "sagemaker:CreateModel",
        "sagemaker:CreateModelBiasJobDefinition",
        "sagemaker:CreateModelExplainabilityJobDefinition",
        "sagemaker:CreateModelPackage",
        "sagemaker:CreateModelPackageGroup",
        "sagemaker:CreateModelQualityJobDefinition",
        "sagemaker:CreateMonitoringSchedule",
        "sagemaker:CreateNotebookInstance",
        "sagemaker:CreateNotebookInstanceLifecycleConfig",
        "sagemaker:CreatePipeline",
        "sagemaker:CreatePresignedDomainUrl",
        "sagemaker:CreatePresignedNotebookInstanceUrl",
        "sagemaker:CreateProcessingJob",
        "sagemaker:CreateProject",
        "sagemaker:CreateTrainingJob",
        "sagemaker:CreateTransformJob",
        "sagemaker:CreateTrial",
        "sagemaker:CreateTrialComponent",
        "sagemaker:CreateUserProfile",
        "sagemaker:CreateWorkforce",
        "sagemaker:CreateWorkteam",
        "sagemaker:DeleteAction",
        "sagemaker:DeleteAlgorithm",
        "sagemaker:DeleteApp",
        "sagemaker:DeleteAppImageConfig",
        "sagemaker:DeleteArtifact",
        "sagemaker:DeleteAssociation",
        "sagemaker:DeleteCodeRepository",
        "sagemaker:DeleteContext",
        "sagemaker:DeleteDataQualityJobDefinition",
        "sagemaker:DeleteDeviceFleet",
        "sagemaker:DeleteDomain",
        "sagemaker:DeleteEndpoint",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DeleteExperiment",
```

```
        "sagemaker:DeleteFeatureGroup",
        "sagemaker:DeleteFlowDefinition",
        "sagemaker:DeleteHumanLoop",
        "sagemaker:DeleteHumanTaskUi",
        "sagemaker:DeleteImage",
        "sagemaker:DeleteImageVersion",
        "sagemaker:DeleteLineageGroupPolicy",
        "sagemaker:DeleteModel",
        "sagemaker:DeleteModelBiasJobDefinition",
        "sagemaker:DeleteModelExplainabilityJobDefinition",
        "sagemaker:DeleteModelPackage",
        "sagemaker:DeleteModelPackageGroup",
        "sagemaker:DeleteModelPackageGroupPolicy",
        "sagemaker:DeleteModelQualityJobDefinition",
        "sagemaker:DeleteMonitoringSchedule",
        "sagemaker:DeleteNotebookInstance",
        "sagemaker:DeleteNotebookInstanceLifecycleConfig",
        "sagemaker:DeletePipeline",
        "sagemaker:DeleteProject",
        "sagemaker:DeleteRecord",
        "sagemaker:DeleteTags",
        "sagemaker:DeleteTrial",
        "sagemaker:DeleteTrialComponent",
        "sagemaker:DeleteUserProfile",
        "sagemaker:DeleteWorkforce",
        "sagemaker:DeleteWorkteam",
        "sagemaker:DeregisterDevices",
        "sagemaker:DescribeAction",
        "sagemaker:DescribeAlgorithm",
        "sagemaker:DescribeApp",
        "sagemaker:DescribeAppImageConfig",
        "sagemaker:DescribeArtifact",
        "sagemaker:DescribeAutoMLJob",
        "sagemaker:DescribeCodeRepository",
        "sagemaker:DescribeCompilationJob",
        "sagemaker:DescribeContext",
        "sagemaker:DescribeDataQualityJobDefinition",
        "sagemaker:DescribeDevice",
        "sagemaker:DescribeDeviceFleet",
        "sagemaker:DescribeDomain",
        "sagemaker:DescribeEdgePackagingJob",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeExperiment",
```

```
        "sagemaker:DescribeFeatureGroup",
        "sagemaker:DescribeFlowDefinition",
        "sagemaker:DescribeHumanLoop",
        "sagemaker:DescribeHumanTaskUi",
        "sagemaker:DescribeHyperParameterTuningJob",
        "sagemaker:DescribeImage",
        "sagemaker:DescribeImageVersion",
        "sagemaker:DescribeInferenceRecommendationsJob",
        "sagemaker:DescribeLabelingJob",
        "sagemaker:DescribeLineageGroup",
        "sagemaker:DescribeModel",
        "sagemaker:DescribeModelBiasJobDefinition",
        "sagemaker:DescribeModelExplainabilityJobDefinition",
        "sagemaker:DescribeModelPackage",
        "sagemaker:DescribeModelPackageGroup",
        "sagemaker:DescribeModelQualityJobDefinition",
        "sagemaker:DescribeMonitoringSchedule",
        "sagemaker:DescribeNotebookInstance",
        "sagemaker:DescribeNotebookInstanceLifecycleConfig",
        "sagemaker:DescribePipeline",
        "sagemaker:DescribePipelineDefinitionForExecution",
        "sagemaker:DescribePipelineExecution",
        "sagemaker:DescribeProcessingJob",
        "sagemaker:DescribeProject",
        "sagemaker:DescribeSubscribedWorkteam",
        "sagemaker:DescribeTrainingJob",
        "sagemaker:DescribeTransformJob",
        "sagemaker:DescribeTrial",
        "sagemaker:DescribeTrialComponent",
        "sagemaker:DescribeUserProfile",
        "sagemaker:DescribeWorkforce",
        "sagemaker:DescribeWorkteam",
        "sagemaker:DisableSagemakerServicecatalogPortfolio",
        "sagemaker:DisassociateTrialComponent",
        "sagemaker:EnableSagemakerServicecatalogPortfolio",
        "sagemaker:GetDeviceFleetReport",
        "sagemaker:GetDeviceRegistration",
        "sagemaker:GetLineageGroupPolicy",
        "sagemaker:GetModelPackageGroupPolicy",
        "sagemaker:GetRecord",
        "sagemaker:GetSagemakerServicecatalogPortfolioStatus",
        "sagemaker:GetSearchSuggestions",
        "sagemaker:InvokeEndpoint",
        "sagemaker:InvokeEndpointAsync",
```

```
        "sagemaker:ListActions",
        "sagemaker:ListAlgorithms",
        "sagemaker:ListAppImageConfigs",
        "sagemaker:ListApps",
        "sagemaker:ListArtifacts",
        "sagemaker:ListAssociations",
        "sagemaker:ListAutoMLJobs",
        "sagemaker:ListCandidatesForAutoMLJob",
        "sagemaker:ListCodeRepositories",
        "sagemaker:ListCompilationJobs",
        "sagemaker:ListContexts",
        "sagemaker:ListDataQualityJobDefinitions",
        "sagemaker:ListDeviceFleets",
        "sagemaker:ListDevices",
        "sagemaker:ListDomains",
        "sagemaker:ListEdgePackagingJobs",
        "sagemaker:ListEndpointConfigs",
        "sagemaker:ListEndpoints",
        "sagemaker:ListExperiments",
        "sagemaker:ListFeatureGroups",
        "sagemaker:ListFlowDefinitions",
        "sagemaker:ListHumanLoops",
        "sagemaker:ListHumanTaskUis",
        "sagemaker:ListHyperParameterTuningJobs",
        "sagemaker:ListImageVersions",
        "sagemaker:ListImages",
        "sagemaker:ListInferenceRecommendationsJobs",
        "sagemaker:ListLabelingJobs",
        "sagemaker:ListLabelingJobsForWorkteam",
        "sagemaker:ListLineageGroups",
        "sagemaker:ListModelBiasJobDefinitions",
        "sagemaker:ListModelExplainabilityJobDefinitions",
        "sagemaker:ListModelMetadata",
        "sagemaker:ListModelPackageGroups",
        "sagemaker:ListModelPackages",
        "sagemaker:ListModelQualityJobDefinitions",
        "sagemaker:ListModels",
        "sagemaker:ListMonitoringExecutions",
        "sagemaker:ListMonitoringSchedules",
        "sagemaker:ListNotebookInstanceLifecycleConfigs",
        "sagemaker:ListNotebookInstances",
        "sagemaker:ListPipelineExecutionSteps",
        "sagemaker:ListPipelineExecutions",
        "sagemaker:ListPipelineParametersForExecution",
```

```
        "sagemaker:ListPipelines",
        "sagemaker:ListProcessingJobs",
        "sagemaker:ListProjects",
        "sagemaker:ListSubscribedWorkteams",
        "sagemaker:ListTags",
        "sagemaker:ListTrainingJobs",
        "sagemaker:ListTrainingJobsForHyperParameterTuningJob",
        "sagemaker:ListTransformJobs",
        "sagemaker:ListTrialComponents",
        "sagemaker:ListTrials",
        "sagemaker:ListUserProfiles",
        "sagemaker:ListWorkforces",
        "sagemaker:ListWorkteams",
        "sagemaker:PutLineageGroupPolicy",
        "sagemaker:PutModelPackageGroupPolicy",
        "sagemaker:PutRecord",
        "sagemaker:QueryLineage",
        "sagemaker:RegisterDevices",
        "sagemaker:RenderUiTemplate",
        "sagemaker:Search",
        "sagemaker:SendHeartbeat",
        "sagemaker:SendPipelineExecutionStepFailure",
        "sagemaker:SendPipelineExecutionStepSuccess",
        "sagemaker:StartHumanLoop",
        "sagemaker:StartMonitoringSchedule",
        "sagemaker:StartNotebookInstance",
        "sagemaker:StartPipelineExecution",
        "sagemaker:StopAutoMLJob",
        "sagemaker:StopCompilationJob",
        "sagemaker:StopEdgePackagingJob",
        "sagemaker:StopHumanLoop",
        "sagemaker:StopHyperParameterTuningJob",
        "sagemaker:StopInferenceRecommendationsJob",
        "sagemaker:StopLabelingJob",
        "sagemaker:StopMonitoringSchedule",
        "sagemaker:StopNotebookInstance",
        "sagemaker:StopPipelineExecution",
        "sagemaker:StopProcessingJob",
        "sagemaker:StopTrainingJob",
        "sagemaker:StopTransformJob",
        "sagemaker:UpdateAction",
        "sagemaker:UpdateAppImageConfig",
        "sagemaker:UpdateArtifact",
        "sagemaker:UpdateCodeRepository",
```

```
        "sagemaker:UpdateContext",
        "sagemaker:UpdateDeviceFleet",
        "sagemaker:UpdateDevices",
        "sagemaker:UpdateDomain",
        "sagemaker:UpdateEndpoint",
        "sagemaker:UpdateEndpointWeightsAndCapacities",
        "sagemaker:UpdateExperiment",
        "sagemaker:UpdateImage",
        "sagemaker:UpdateModelPackage",
        "sagemaker:UpdateMonitoringSchedule",
        "sagemaker:UpdateNotebookInstance",
        "sagemaker:UpdateNotebookInstanceLifecycleConfig",
        "sagemaker:UpdatePipeline",
        "sagemaker:UpdatePipelineExecution",
        "sagemaker:UpdateProject",
        "sagemaker:UpdateTrainingJob",
        "sagemaker:UpdateTrial",
        "sagemaker:UpdateTrialComponent",
        "sagemaker:UpdateUserProfile",
        "sagemaker:UpdateWorkforce",
        "sagemaker:UpdateWorkteam"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:action/*",
        "arn:aws:sagemaker:*:*:algorithm/*",
        "arn:aws:sagemaker:*:*:app-image-config/*",
        "arn:aws:sagemaker:*:*:artifact/*",
        "arn:aws:sagemaker:*:*:automl-job/*",
        "arn:aws:sagemaker:*:*:code-repository/*",
        "arn:aws:sagemaker:*:*:compilation-job/*",
        "arn:aws:sagemaker:*:*:context/*",
        "arn:aws:sagemaker:*:*:data-quality-job-definition/*",
        "arn:aws:sagemaker:*:*:device-fleet/*/device/*",
        "arn:aws:sagemaker:*:*:device-fleet/*",
        "arn:aws:sagemaker:*:*:edge-packaging-job/*",
        "arn:aws:sagemaker:*:*:endpoint/*",
        "arn:aws:sagemaker:*:*:endpoint-config/*",
        "arn:aws:sagemaker:*:*:experiment/*",
        "arn:aws:sagemaker:*:*:experiment-trial/*",
        "arn:aws:sagemaker:*:*:experiment-trial-component/*",
        "arn:aws:sagemaker:*:*:feature-group/*",
        "arn:aws:sagemaker:*:*:human-loop/*",
        "arn:aws:sagemaker:*:*:human-task-ui/*",
        "arn:aws:sagemaker:*:*:hyper-parameter-tuning-job/*",
```

```
        "arn:aws:sagemaker:*:*:image/*",
        "arn:aws:sagemaker:*:*:image-version/*/*",
        "arn:aws:sagemaker:*:*:inference-recommendations-job/*",
        "arn:aws:sagemaker:*:*:labeling-job/*",
        "arn:aws:sagemaker:*:*:model/*",
        "arn:aws:sagemaker:*:*:model-bias-job-definition/*",
        "arn:aws:sagemaker:*:*:model-explainability-job-definition/*",
        "arn:aws:sagemaker:*:*:model-package/*",
        "arn:aws:sagemaker:*:*:model-package-group/*",
        "arn:aws:sagemaker:*:*:model-quality-job-definition/*",
        "arn:aws:sagemaker:*:*:monitoring-schedule/*",
        "arn:aws:sagemaker:*:*:notebook-instance/*",
        "arn:aws:sagemaker:*:*:notebook-instance-lifecycle-config/*",
        "arn:aws:sagemaker:*:*:pipeline/*",
        "arn:aws:sagemaker:*:*:pipeline/*/execution/*",
        "arn:aws:sagemaker:*:*:processing-job/*",
        "arn:aws:sagemaker:*:*:project/*",
        "arn:aws:sagemaker:*:*:training-job/*",
        "arn:aws:sagemaker:*:*:transform-job/*",
        "arn:aws:sagemaker:*:*:workforce/*",
        "arn:aws:sagemaker:*:*:workteam/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:service-role/
AmazonSageMakerServiceCatalogProductsExecutionRole"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogDelivery",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:DescribeResourcePolicies",
        "logs:DescribeDestinations",
```

```
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:DescribeQueries",
        "logs:DescribeQueryDefinitions",
        "logs:DescribeSubscriptionFilters",
        "logs:GetLogDelivery",
        "logs:GetLogEvents",
        "logs:ListLogDeliveries",
        "logs:PutLogEvents",
        "logs:PutResourcePolicy",
        "logs:UpdateLogDelivery"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/lambda/*"
    }
  ]
}
```

Amazon SageMaker memperbarui kebijakan AWS terkelola AWS Service Catalog

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk Amazon SageMaker karena layanan ini mulai melacak perubahan-perubahan tersebut.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerPartnerServiceCatalogProductsApiGatewayServiceRolePolicy	1	Kebijakan awal	1 Agustus 2023
AmazonSageMakerPartnerServiceCatalogProductsCloudFormationServiceRolePolicy	1	Kebijakan awal	1 Agustus 2023
AmazonSageMakerPartnerServiceCatalogProductsLambdaServiceRolePolicy	1	Kebijakan awal	1 Agustus 2023

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerServiceCatalogProductsGlueServiceRolePolicy	2	Tambahkan izin untuk`glue:GetUserDefinedFunctions` .	26 Agustus 2022
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	7	Tambahkan izin untuk`sagemaker:AddTags` .	2 Agustus 2022
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	6	Tambahkan izin untuk`lambda:TagResource` .	14 Juli 2022
AmazonSageMakerServiceCatalogProductsLambdaServiceRolePolicy	1	Kebijakan awal	4 April 2022
AmazonSageMakerServiceCatalogProductsApiGatewayServiceRolePolicy	1	Kebijakan awal	24 Maret 2022
AmazonSageMakerServiceCatalogProductsCloudformationServiceRolePolicy	1	Kebijakan awal	24 Maret 2022
AmazonSageMakerServiceCatalogProductsCodeBuildServiceRolePolicy	1	Kebijakan awal	24 Maret 2022
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	5	Tambahkan izin baru untuk`ecr-idp:TagResource` .	21 Maret 2022

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerServiceCatalogProductsCodePipelineServiceRolePolicy	1	Kebijakan awal	22 Februari 2022
AmazonSageMakerServiceCatalogProductsEventsServiceRolePolicy	1	Kebijakan awal	22 Februari 2022
AmazonSageMakerServiceCatalogProductsFirehoseServiceRolePolicy	1	Kebijakan awal	22 Februari 2022
AmazonSageMakerServiceCatalogProductsGlueServiceRolePolicy	1	Kebijakan awal	22 Februari 2022
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	4	Tambahkan izin untuk `cognito-idp:TagResource dans3:PutBucketCORS` .	16 Februari 2022
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	3	Tambahkan izin baru untuk`sagemaker` . Membuat, membaca, memperbarui, dan menghapus SageMaker Gambar.	15 September 2021

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	2	Tambahkan izin untuk `sagemaker dancodestar-connections .` Membuat, membaca, memperbarui, dan menghapus repositori kode. Lewati AWS CodeStar koneksi keAWS CodePipeline.	1 Juli 2021
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	1	Kebijakan awal	27 November 2020

SageMaker Pembaruan Kebijakan AWS Terkelola

Lihat detail tentang pembaruan terhadap kebijakan AWS terkelola untuk SageMaker karena layanan ini mulai melacak perubahan-perubahan tersebut.

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerFullAccess - Pembaruan ke kebijakan yang tersedia	25	Tambahkan`sagemaker :CreateApp ,sagemaker :DescribeApp ,sagemaker :DeleteApp ,sagemaker :CreateSpace ,sagemaker :UpdateSp`	30 November 2023

Kebijakan	Versi	Perubahan	Tanggal
		`ace` ,`sagemaker:DeleteSpace` ,`s3express:CreateSession` ,`s3express:CreateBucket` , dan `s3express:ListAllMyDirectoryBuckets` izin.	
AmazonSageMakerFullAccess - Pembaruan ke kebijakan yang tersedia	24	Tambahkan`sagemaker-geospatial:*` ,`sagemaker:AddTags` ,`sagemaker-ListTags` ,`sagemaker-DescribeSpace` , dan `sagemaker:ListSpaces` izin.	30 November 2022
AmazonSageMakerFullAccess - Pembaruan ke kebijakan yang tersedia	23	Tambahkan `glue:UpdateTable` .	29 Juni 2022
AmazonSageMakerFullAccess - Pembaruan ke kebijakan yang tersedia	22	Tambahkan `cloudformation:ListStackResources` .	1 Mei 2022

Kebijakan	Versi	Perubahan	Tanggal
AmazonSageMakerReadOnly - Pembaruan ke kebijakan yang tersedia	11	Tambahkan `sagemaker:QueryLineage`, `sagemaker:GetLineageGroupPolicy`, `sagemaker:BatchDescribeModelPackage`, `sagemaker:GetModelPackageGroupPolicy` izin.	1 Desember 2021
AmazonSageMakerFullAccess - Pembaruan ke kebijakan yang tersedia	21	Tambahkan `sns:Publish` izin untuk titik akhir dengan Inferensi Async diaktifkan.	8 September 2021
AmazonSageMakerFullAccess - Pembaruan ke kebijakan yang tersedia	20	Perbarui `iam:PassRole` sumber daya dan izin.	15 Juli 2021
AmazonSageMakerReadOnly - Pembaruan ke kebijakan yang tersedia	10	API baru `BatchGetRecord` ditambahkan untuk SageMaker Feature Store.	10 Juni 2021
		SageMaker mulai melacak perubahan untuk kebijakan AWS terkelola	1 Juni 2021

Pemecahan masalah SageMaker identitas dan akses Amazon

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang Anda mungkin temukan saat bekerja dengan SageMaker dan IAM.

Topik

Saya tidak memiliki izin untuk melakukan tindakan di SageMaker

Jika AWS Management Console memberi tahu bahwa Anda tidak diberi otorisasi untuk melakukan tindakan, Anda harus menghubungi administrator untuk mendapatkan bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Contoh kesalahan berikut terjadi ketika pengguna `mateojackson` IAM mencoba menggunakan konsol untuk melihat detail tentang tugas pelatihan, tetapi tidak memiliki `sagemaker:sagemaker:DescribeTrainingJob` izin.

```
User: arn:aws:iam::123456789012:user/mateojackson is not
            authorized to perform: sagemaker:DescribeTrainingJob on resource: my-
example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya untuk mengizinkan dia mengakses sumber daya `TrainingJob` menggunakan tindakan `sagemaker:DescribeTrainingJob`.

Saya tidak Berwenang untuk Melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak terotorisasi untuk melakukan `iam:PassRole` tindakan tersebut, kebijakan Anda harus diperbarui agar Anda dapat memberikan peran SageMaker.

Sebagian Layanan AWS mengizinkan Anda untuk memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait-layanan. Untuk melakukan tindakan tersebut, Anda harus memiliki izin untuk memberikan peran pada layanan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di SageMaker. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar AWS Akun Saya untuk Mengakses SageMaker Sumber Daya Saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi pengguna akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah SageMaker mendukung fitur-fitur ini, lihatCara Amazon SageMaker Bekerja dengan IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di Akun AWS lainnya yang Anda miliki dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke pihak ketiga Akun AWS, lihat Menyediakan akses ke akun Akun AWS yang dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (gabungan identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Perbedaan antara peran IAM dan kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Pencatatan dan Pemantauan

Anda dapat memantau Amazon SageMaker menggunakan Amazon CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca dan hampir waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Pantau Amazon SageMaker dengan Amazon CloudWatch.

Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses berkas log dari instans Amazon EC2,,,AWS CloudTrail, dan sumber lainnya. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat Log SageMaker Acara Amazon dengan Amazon CloudWatch.

AWS CloudTrail memberikan catatan tindakan yang diambil oleh pengguna, peran, atau layanan AWS di SageMaker. Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke SageMaker, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya. Untuk informasi selengkapnya, Log Panggilan SageMaker API Amazon dengan AWS CloudTrail.

> **Note**
>
> CloudTrail tidak memonitor panggilan ke `runtime_InvokeEndpoint`.

Anda dapat membuat aturan di Amazon CloudWatch Events untuk bereaksi terhadap perubahan status status dalam SageMaker pelatihan, penyetelan hiperperparameter, atau pekerjaan transformasi batch. Untuk informasi selengkapnya, lihat Mengotomatisasi Amazon SageMaker dengan Amazon EventBridge.

Validasi kepatuhan untuk Amazon SageMaker

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan khusus, lihat Layanan AWS di Scope oleh Program Program Kepatuhan yang Anda minati. Untuk informasi umum, lihat Program Kepatuhan AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan berdasarkan sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Mulai Cepat Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.
- Merancang Keamanan dan Kepatuhan HIPAA di Amazon Web Services – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

  > **Note**
  >
  > Tidak semua Layanan AWS memenuhi syarat HIPAA. Untuk informasi selengkapnya, lihat Referensi Layanan yang Memenuhi Syarat HIPAA.

- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Panduan Kepatuhan Pelanggan AWS – Pahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan kontrol keamanan di banyak kerangka kerja (termasuk National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), dan International Organization for Standardization (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- AWS Security Hub – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dalam AWS. Security Hub menggunakan kontrol keamanan untuk

mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.

- AWS Audit Manager – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

Ketahanan di Amazon SageMaker

Infrastruktur global AWS dibangun di seputar Kawasan dan Zona Ketersediaan AWS. AWS Kawasan menyediakan beberapa Zona Ketersediaan yang terpisah dan terisolasi secara fisik, yang tersambung dengan jejaring jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang melakukan secara otomatis pinda saat gagal/failover di antara zona-zona tanpa terputus. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripadai infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Zona Ketersediaan, lihat Infrastruktur Global AWS.

Selain infrastruktur AWS global, Amazon memberi SageMaker penawaran beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan backup Anda.

Keamanan Infrastruktur di Amazon SageMaker

Sebagai layanan terkelola, Amazon SageMaker dilindungi oleh keamanan jaringan AWS global. Lihat informasi tentang layanan keamanan AWS dan cara AWS melindungi infrastruktur di Keamanan Cloud AWS. Untuk mendesain lingkungan AWS Anda dengan menggunakan praktik terbaik bagi keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Pilar Keamanan Kerangka Kerja Berarsitektur Baik AWS.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon SageMaker melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Topik

SageMaker Memindai AWS Marketplace Pelatihan dan Inferensi Wadah untuk Kerentanan Keamanan

Untuk memenuhi persyaratan keamanan kami, semua SageMaker gambar yang telah dibuat sebelumnya, termasuk AWS Deep Learning Containers, wadah kerangka kerja pembelajaran SageMaker mesin, dan wadah algoritme SageMaker bawaan, serta algoritme dan paket model yang AWS Marketplace tercantum dalam dipindai untuk Common Vulnerabilities and Exposures (CVE). CVE adalah daftar informasi yang diketahui publik tentang kerentanan dan eksposur keamanan. National Vulnerability Database (NVD) memberikan rincian CVE seperti tingkat keparahan, peringkat dampak, dan informasi perbaikan. Baik CVE dan NVD tersedia untuk konsumsi publik dan gratis untuk alat dan layanan keamanan untuk digunakan. Untuk informasi selengkapnya, lihat Pertanyaan yang Sering Diajukan CVE (FAQ).

Connect ke Resources Dari Dalam VPC

Instans Amazon SageMaker Studio Classic dan SageMaker notebook memungkinkan akses internet langsung secara default. Ini memungkinkan Anda mengunduh paket dan notebook populer, menyesuaikan lingkungan pengembangan Anda, dan bekerja secara efisien. Namun, ini dapat memberikan jalan tambahan untuk akses tidak sah ke data Anda. Misalnya, jika Anda menginstal kode berbahaya di komputer Anda dalam bentuk buku catatan yang tersedia untuk umum atau pustaka kode sumber yang tersedia untuk umum, kode tersebut dapat mengakses data Anda. Anda dapat memilih untuk membatasi lalu lintas mana yang dapat mengakses internet dengan

meluncurkan instans Amazon SageMaker Studio Classic dan SageMaker notebook di Amazon Virtual Private Cloud (Amazon VPC) yang Anda pilih.

Amazon Virtual Private Cloud adalah jaringan virtual yang khususkan untuk AWS akun Anda. Dengan Amazon VPC, Anda dapat mengontrol akses jaringan dan konektivitas internet dari instans Amazon SageMaker Studio Classic dan notebook Anda. Anda dapat menonaktifkan akses internet langsung untuk menambahkan lapisan keamanan tambahan.

Topik berikut menjelaskan cara menghubungkan instans SageMaker Studio Classic dan instans notebook Anda ke sumber daya di VPC.

Topik

Hubungkan Amazon SageMaker Studio di VPC ke Sumber Daya Eksternal

> **Important**
>
> Pada 30 November 2023, pengalaman Amazon SageMaker Studio sebelumnya sekarang bernama Amazon SageMaker Studio Classic. Bagian berikut khusus untuk menggunakan pengalaman Studio yang diperbarui. Untuk informasi tentang menggunakan aplikasi Studio Classic, lihatAmazon SageMaker Studio Klasik.

Topik berikut memberikan informasi tentang cara menghubungkan Amazon SageMaker Studio di VPC ke sumber daya eksternal.

Topik

Komunikasi default dengan internet

Secara default, Amazon SageMaker Studio menyediakan antarmuka jaringan yang memungkinkan komunikasi dengan internet melalui VPC yang dikelola oleh. SageMaker Lalu lintas ke AWS layanan

seperti Amazon S3 dan CloudWatch melewati gateway internet, seperti halnya lalu lintas yang mengakses SageMaker API dan runtime. SageMaker Lalu lintas antara domain dan volume Amazon EFS Anda melewati VPC yang Anda tentukan saat Anda onboard ke domain atau disebut API. CreateDomain

`VPC only`komunikasi dengan internet

Untuk SageMaker mencegah penyediaan akses internet ke Studio, Anda dapat menonaktifkan akses internet dengan menentukan jenis akses `VPC only` jaringan saat Anda onboard ke Studio Classic atau memanggil API. CreateDomain Akibatnya, Anda tidak akan dapat menjalankan Studio kecuali VPC Anda memiliki titik akhir antarmuka ke SageMaker API dan waktu proses, atau gateway NAT dengan akses internet, dan grup keamanan Anda mengizinkan koneksi keluar.

> **Note**
>
> Jenis akses jaringan tidak dapat diubah setelah pembuatan domain.

Persyaratan untuk menggunakan **`VPC only`** mode

Saat Anda memilih`VpcOnly`, ikuti langkah-langkah ini:

1. Anda harus menggunakan subnet pribadi saja. Anda tidak dapat menggunakan subnet publik dalam `VpcOnly` mode.
2. Pastikan subnet Anda memiliki jumlah alamat IP yang diperlukan. Jumlah yang diharapkan dari alamat IP yang dibutuhkan per pengguna dapat bervariasi berdasarkan kasus penggunaan. Kami merekomendasikan antara 2 dan 4 alamat IP per pengguna. Total kapasitas alamat IP untuk domain adalah jumlah alamat IP yang tersedia untuk setiap subnet yang disediakan saat domain dibuat. Pastikan perkiraan penggunaan alamat IP Anda tidak melebihi kapasitas yang didukung oleh jumlah subnet yang Anda berikan. Selain itu, menggunakan subnet yang didistribusikan di banyak zona ketersediaan dapat membantu ketersediaan alamat IP. Untuk informasi selengkapnya, lihat Ukuran VPC dan subnet untuk IPv4.

   > **Note**
   >
   > Anda hanya dapat mengonfigurasi subnet dengan VPC penyewaan default di mana instans Anda berjalan pada perangkat keras bersama. Untuk informasi selengkapnya tentang atribut tenancy untuk VPC, lihat Instans Khusus.

3.

> **Warning**
>
> Saat menggunakan `VpcOnly` mode, Anda sebagian memiliki konfigurasi jaringan untuk domain. Kami merekomendasikan praktik keamanan terbaik untuk menerapkan izin hak istimewa terkecil ke akses masuk dan keluar yang disediakan aturan grup keamanan. Konfigurasi aturan masuk yang terlalu permisif dapat memungkinkan pengguna dengan akses ke VPC untuk berinteraksi dengan aplikasi profil pengguna lain tanpa otentikasi.

Siapkan satu atau beberapa grup keamanan dengan aturan masuk dan keluar yang memungkinkan lalu lintas berikut:

- Lalu lintas NFS melalui TCP pada port 2049 antara domain dan volume Amazon EFS.
- Lalu lintas TCP dalam grup keamanan. Ini diperlukan untuk konektivitas antara Jupyter Server aplikasi dan Kernel Gateway aplikasi. Anda harus mengizinkan akses ke setidaknya port dalam jangkauan`8192-65535`.

Buat grup keamanan yang berbeda untuk setiap profil pengguna dan tambahkan akses masuk dari grup keamanan yang sama. Kami tidak menyarankan untuk menggunakan kembali grup keamanan tingkat domain untuk profil pengguna. Jika grup keamanan tingkat domain memungkinkan akses masuk ke dirinya sendiri, maka semua aplikasi dalam domain akan memiliki akses ke semua aplikasi lain di domain.

4. Jika Anda ingin mengizinkan akses internet, Anda harus menggunakan gateway NAT dengan akses ke internet, misalnya melalui gateway internet.
5. Jika Anda tidak ingin mengizinkan akses internet, buat antarmuka VPC endpoint (AWS PrivateLink) untuk memungkinkan Studio mengakses layanan berikut dengan nama layanan yang sesuai. Anda juga harus mengaitkan grup keamanan untuk VPC Anda dengan titik akhir ini.

   - SageMaker API: `com.amazonaws.`*`region`*`.sagemaker.api`
   - SageMaker runtime:`com.amazonaws.`*`region`*`.sagemaker.runtime`. Ini diperlukan untuk menjalankan notebook Studio Classic dan untuk melatih dan meng-host model.
   - Amazon S3:. `com.amazonaws.`*`region`*`.s3`
   - Untuk menggunakan SageMaker Proyek:`com.amazonaws.`*`region`*`.servicecatalog`.
   - AWSLayanan lain yang Anda butuhkan.

Jika Anda menggunakan SageMaker Python SDK untuk menjalankan pekerjaan pelatihan jarak jauh, Anda juga harus membuat endpoint Amazon VPC berikut.

- AWS Security Token Service: `com.amazonaws.region.sts`
- Amazon CloudWatch:`com.amazonaws.region.logs`. Ini diperlukan untuk memungkinkan SageMaker Python SDK mendapatkan status pekerjaan pelatihan jarak jauh. Amazon CloudWatch

6. Jika menggunakan domain dalam `VpcOnly` mode dari jaringan lokal, buat konektivitas pribadi dari jaringan host yang menjalankan Studio di browser dan VPC Amazon target. Ini diperlukan karena UI Studio memanggil AWS titik akhir menggunakan panggilan API dengan kredensyal sementaraAWS. Kredenial sementara ini terkait dengan peran eksekusi dari profil pengguna yang dicatat. Jika domain dikonfigurasi dalam `VpcOnly` mode di jaringan lokal, peran eksekusi mungkin menentukan kondisi kebijakan IAM yang menerapkan eksekusi panggilan API AWS layanan hanya melalui titik akhir Amazon VPC yang dikonfigurasi.Hal ini menyebabkan panggilan API yang dijalankan dari UI Studio gagal. Sebaiknya selesaikan ini menggunakan AWS Direct Connectkoneksi AWS Site-to-Site VPNatau.

> **Note**
>
> Untuk pelanggan yang bekerja dalam mode VPC, firewall perusahaan dapat menyebabkan masalah koneksi dengan Studio atau aplikasi. Lakukan pemeriksaan berikut jika Anda mengalami salah satu masalah ini saat menggunakan Studio dari belakang firewall.
>
> - Verifikasi bahwa URL Studio dan URL untuk semua aplikasi Anda ada di daftar izin jaringan Anda. Sebagai contoh:
>
>   ```
>   *.studio.region.sagemaker.aws
>   *.console.aws.a2z.com
>   ```
>
> - Verifikasi bahwa koneksi websocket tidak diblokir. Jupyter menggunakan soket web.

Untuk informasi selengkapnya

- Grup Keamanan untuk VPC Anda
- Connect ke SageMaker Dalam VPC

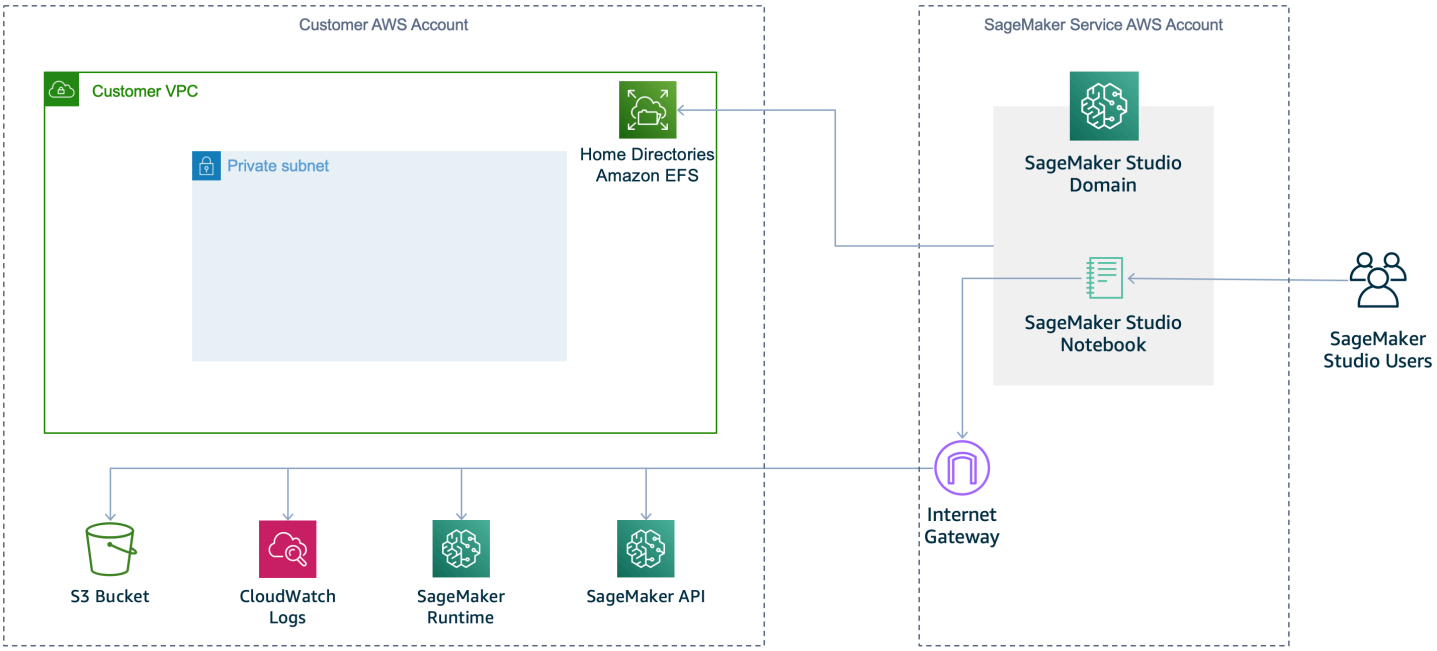
- VPC dengan subnet publik dan privat (NAT)

Connect SageMaker Studio Classic Notebook dalam VPC ke Sumber Daya Eksternal

Topik berikut memberikan informasi tentang cara menghubungkan Studio Classic Notebook di VPC ke sumber daya eksternal.

Komunikasi default dengan internet

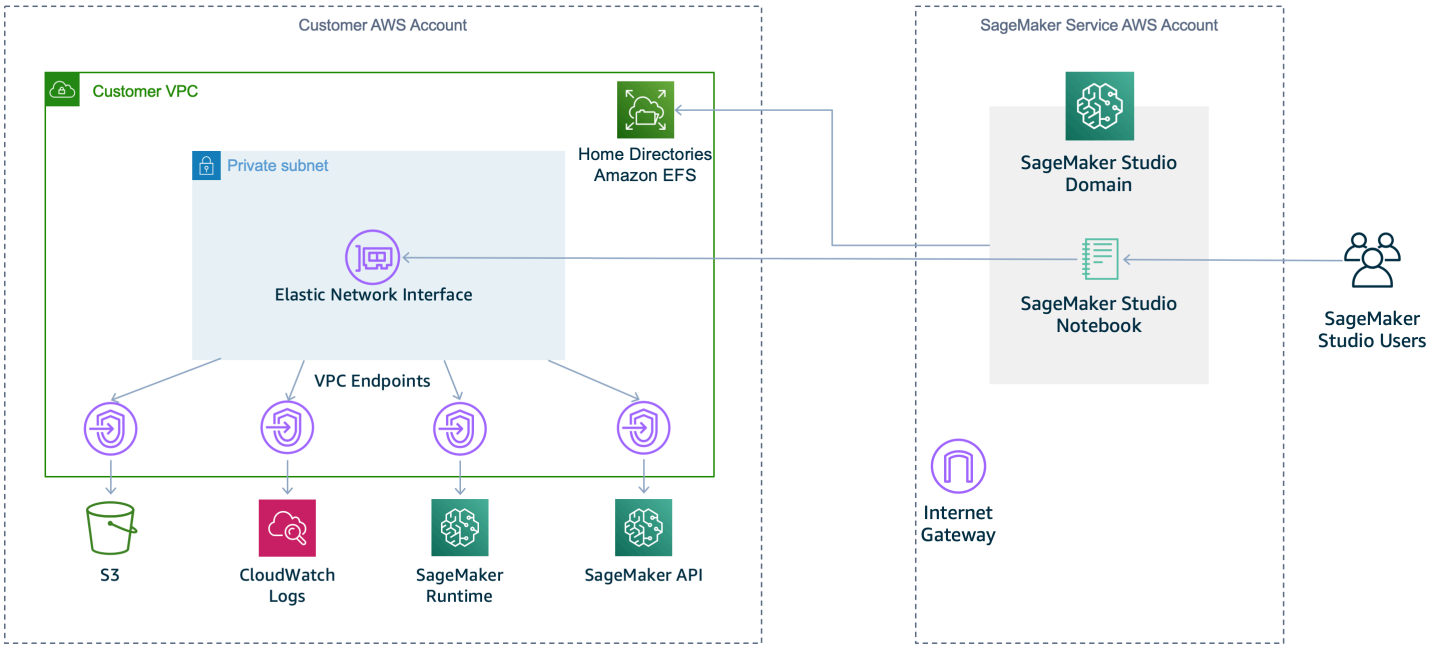
Secara default, SageMaker Studio Classic menyediakan antarmuka jaringan yang memungkinkan komunikasi dengan internet melalui VPC yang dikelola oleh. SageMaker Lalu lintas ke AWS layanan seperti Amazon S3 dan CloudWatch melewati gateway internet, seperti halnya lalu lintas yang mengakses SageMaker API dan runtime. SageMaker Lalu lintas antara domain dan volume Amazon EFS Anda melewati VPC yang Anda tentukan saat Anda onboard ke Studio Classic atau disebut API. CreateDomain Diagram berikut menunjukkan konfigurasi default.



`VPC only`komunikasi dengan internet

Untuk SageMaker mencegah penyediaan akses internet ke notebook Studio Classic, Anda dapat menonaktifkan akses internet dengan menentukan jenis akses `VPC only` jaringan saat Anda onboard ke Studio Classic atau memanggil API. CreateDomain Akibatnya, Anda tidak akan dapat menjalankan buku catatan Studio Classic kecuali VPC Anda memiliki titik akhir antarmuka ke SageMaker API dan waktu proses, atau gateway NAT dengan akses internet, dan grup keamanan

Anda mengizinkan koneksi keluar. Diagram berikut menunjukkan konfigurasi untuk menggunakan mode VPC saja.



Persyaratan untuk menggunakan **`VPC only`** mode

Saat Anda memilih`VpcOnly`, ikuti langkah-langkah ini:

1. Anda harus menggunakan subnet pribadi saja. Anda tidak dapat menggunakan subnet publik dalam `VpcOnly` mode.
2. Pastikan subnet Anda memiliki jumlah alamat IP yang diperlukan. Jumlah yang diharapkan dari alamat IP yang dibutuhkan per pengguna dapat bervariasi berdasarkan kasus penggunaan. Kami merekomendasikan antara 2 dan 4 alamat IP per pengguna. Total kapasitas alamat IP untuk domain Studio Classic adalah jumlah alamat IP yang tersedia untuk setiap subnet yang disediakan saat domain dibuat. Pastikan perkiraan penggunaan alamat IP Anda tidak melebihi kapasitas yang didukung oleh jumlah subnet yang Anda berikan. Selain itu, menggunakan subnet yang didistribusikan di banyak zona ketersediaan dapat membantu ketersediaan alamat IP. Untuk informasi selengkapnya, lihat Ukuran VPC dan subnet untuk IPv4.

> **Note**
>
> Anda hanya dapat mengonfigurasi subnet dengan VPC penyewaan default di mana instans Anda berjalan pada perangkat keras bersama. Untuk informasi selengkapnya tentang atribut tenancy untuk VPC, lihat Instans Khusus.

3.

> **Warning**
>
> Saat menggunakan `VpcOnly` mode, Anda sebagian memiliki konfigurasi jaringan untuk domain. Kami merekomendasikan praktik keamanan terbaik untuk menerapkan izin hak istimewa terkecil ke akses masuk dan keluar yang disediakan aturan grup keamanan. Konfigurasi aturan masuk yang terlalu permisif dapat memungkinkan pengguna dengan akses ke VPC untuk berinteraksi dengan aplikasi profil pengguna lain tanpa otentikasi.

Siapkan satu atau beberapa grup keamanan dengan aturan masuk dan keluar yang memungkinkan lalu lintas berikut:

- Lalu lintas NFS melalui TCP pada port 2049 antara domain dan volume Amazon EFS.
- Lalu lintas TCP dalam grup keamanan. Ini diperlukan untuk konektivitas antara Jupyter Server aplikasi dan Kernel Gateway aplikasi. Anda harus mengizinkan akses ke setidaknya port dalam jangkauan`8192-65535`.

Buat grup keamanan yang berbeda untuk setiap profil pengguna dan tambahkan akses masuk dari grup keamanan yang sama. Kami tidak menyarankan untuk menggunakan kembali grup keamanan tingkat domain untuk profil pengguna. Jika grup keamanan tingkat domain memungkinkan akses masuk ke dirinya sendiri, maka semua aplikasi dalam domain akan memiliki akses ke semua aplikasi lain di domain.

4. Jika Anda ingin mengizinkan akses internet, Anda harus menggunakan gateway NAT dengan akses ke internet, misalnya melalui gateway internet.
5. Jika Anda tidak ingin mengizinkan akses internet, buat antarmuka VPC endpoint (AWS PrivateLink) untuk memungkinkan Studio Classic mengakses layanan berikut dengan nama layanan yang sesuai. Anda juga harus mengaitkan grup keamanan untuk VPC Anda dengan titik akhir ini.

   - SageMaker API: `com.amazonaws.`*`region`*`.sagemaker.api`
   - SageMaker runtime:`com.amazonaws.`*`region`*`.sagemaker.runtime`. Ini diperlukan untuk menjalankan notebook Studio Classic dan untuk melatih dan meng-host model.
   - Amazon S3:. `com.amazonaws.`*`region`*`.s3`
   - Untuk menggunakan SageMaker Proyek:`com.amazonaws.`*`region`*`.servicecatalog`.
   - AWSLayanan lain yang Anda butuhkan.

Jika Anda menggunakan SageMaker Python SDK untuk menjalankan pekerjaan pelatihan jarak jauh, Anda juga harus membuat endpoint Amazon VPC berikut.

- AWS Security Token Service: `com.amazonaws.region.sts`
- Amazon CloudWatch:`com.amazonaws.region.logs`. Ini diperlukan untuk memungkinkan SageMaker Python SDK mendapatkan status pekerjaan pelatihan jarak jauh. Amazon CloudWatch

> **Note**
>
> Untuk pelanggan yang bekerja dalam mode VPC, firewall perusahaan dapat menyebabkan masalah koneksi dengan SageMaker Studio atau antara JupyterServer dan. KernelGateway Lakukan pemeriksaan berikut jika Anda mengalami salah satu masalah ini saat menggunakan SageMaker Studio dari belakang firewall.
>
> - Periksa apakah URL Studio ada di daftar yang diizinkan jaringan Anda.
> - Periksa apakah koneksi websocket tidak diblokir. Jupyter menggunakan websocket di bawah tenda. Jika KernelGateway aplikasi ini InService, JupyterServer mungkin tidak dapat terhubung ke KernelGateway. Anda akan melihat masalah ini saat membuka Terminal Sistem juga.

Untuk informasi selengkapnya

- Mengamankan konektivitas Amazon SageMaker Studio menggunakan VPC pribadi.
- Grup Keamanan untuk VPC Anda
- Connect ke SageMaker Dalam VPC
- VPC dengan subnet publik dan privat (NAT)

Hubungkan Instance Notebook di VPC ke Sumber Daya Eksternal

Topik berikut memberikan informasi tentang cara menghubungkan instance notebook Anda di VPC ke sumber daya eksternal.

Komunikasi default dengan internet

Ketika notebook Anda memungkinkan akses internet langsung, SageMaker menyediakan antarmuka jaringan yang memungkinkan notebook untuk berkomunikasi dengan internet melalui VPC yang dikelola oleh. SageMaker Lalu lintas dalam CIDR VPC Anda melewati elastic network interface yang dibuat di VPC Anda. Semua lalu lintas lainnya melewati antarmuka jaringan yang dibuat oleh SageMaker, yang pada dasarnya melalui internet publik. Lalu lintas ke titik akhir VPC gateway seperti Amazon S3 dan DynamoDB melewati internet publik, sementara lalu lintas ke antarmuka titik akhir antarmuka VPC masih melalui VPC Anda. Jika Anda ingin menggunakan titik akhir VPC gateway, Anda mungkin ingin menonaktifkan akses internet langsung.

Komunikasi VPC dengan internet

Untuk menonaktifkan akses internet langsung, Anda dapat menentukan VPC untuk instance notebook Anda. Dengan demikian, Anda SageMaker mencegah penyediaan akses internet ke instance notebook Anda. Akibatnya, instans notebook tidak dapat melatih atau meng-host model kecuali VPC Anda memiliki titik akhir antarmuka (AWS PrivateLink) atau gateway NAT dan grup keamanan Anda mengizinkan koneksi keluar.

Untuk informasi tentang membuat titik akhir antarmuka VPC yang akan digunakan AWS PrivateLink untuk instance notebook Anda, lihat. Connect ke Instance Notebook Melalui Endpoint Antarmuka VPC Untuk informasi tentang menyiapkan gateway NAT untuk VPC Anda, lihat VPC dengan Subnet Publik dan Pribadi (NAT) di Panduan Pengguna Amazon Virtual Private Cloud. Untuk informasi tentang grup keamanan, lihat Grup Keamanan untuk VPC Anda. Untuk informasi selengkapnya tentang konfigurasi jaringan di setiap mode jaringan dan mengonfigurasi jaringan di lokasi, lihat Memahami konfigurasi jaringan instans SageMaker notebook Amazon dan opsi perutean lanjutan.

Instans Keamanan dan Notebook Bersama

Sebuah instance SageMaker notebook dirancang untuk bekerja paling baik bagi pengguna individu. Ini dirancang untuk memberikan ilmuwan data dan pengguna lain kekuatan paling besar untuk mengelola lingkungan pengembangan mereka.

Pengguna instance notebook memiliki akses root untuk menginstal paket dan perangkat lunak terkait lainnya. Kami menyarankan Anda untuk melakukan penilaian saat memberikan individu akses ke instans notebook yang dilampirkan ke VPC yang berisi informasi sensitif. Misalnya, Anda dapat memberikan akses pengguna ke instans notebook dengan kebijakan IAM, seperti yang ditunjukkan pada contoh berikut:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": "sagemaker:CreatePresignedNotebookInstanceUrl",
      "Resource": "arn:aws:sagemaker:region:account-id:notebook-instance/
myNotebookInstance"
   }
  ]
}
```

Jalankan Kontainer Pelatihan dan Inferensi dalam Mode Bebas Internet

SageMaker pelatihan dan wadah inferensi yang diterapkan diaktifkan internet secara default. Ini memungkinkan kontainer untuk mengakses layanan dan sumber daya eksternal di internet publik sebagai bagian dari beban kerja pelatihan dan inferensi Anda. Namun, ini dapat memberikan jalan untuk akses tidak sah ke data Anda. Misalnya, pengguna atau kode jahat yang tidak sengaja Anda instal di wadah (dalam bentuk pustaka kode sumber yang tersedia untuk umum) dapat mengakses data Anda dan mentransfernya ke host jarak jauh.

Jika Anda menggunakan VPC Amazon dengan menentukan nilai `VpcConfig` parameter saat menelepon `CreateTrainingJob`, atau `CreateHyperParameterTuningJobCreateModel`, Anda dapat melindungi data dan sumber daya Anda dengan mengelola grup keamanan dan membatasi akses internet dari VPC Anda. Namun, ini datang dengan biaya konfigurasi jaringan tambahan, dan memiliki risiko mengkonfigurasi jaringan Anda secara tidak benar. Jika Anda tidak SageMaker ingin memberikan akses jaringan eksternal ke wadah pelatihan atau inferensi Anda, Anda dapat mengaktifkan isolasi jaringan.

Isolasi Jaringan

Anda dapat mengaktifkan isolasi jaringan saat membuat pekerjaan atau model pelatihan dengan menyetel nilai `EnableNetworkIsolation` parameter `True` saat Anda menelepon `CreateTrainingJob`, `CreateHyperParameterTuningJob`, atau `CreateModel`.

> **Note**
>
> Isolasi jaringan diperlukan untuk menjalankan pekerjaan pelatihan dan model menggunakan sumber daya dariAWS Marketplace. Untuk keamanan tambahan, AWS Marketplace gambar

berjalan dalam VPC Amazon. Mereka hanya memiliki akses ke data dalam sistem file lokal mereka.

Jika Anda mengaktifkan isolasi jaringan, kontainer tidak dapat melakukan panggilan jaringan keluar, bahkan ke AWS layanan lain seperti Amazon S3. Selain itu, tidak ada AWS kredensyal yang tersedia untuk lingkungan runtime container. Dalam kasus pekerjaan pelatihan dengan beberapa contoh, lalu lintas masuk dan keluar jaringan terbatas pada rekan-rekan dari setiap wadah pelatihan. SageMaker masih menjalankan operasi pengunduhan dan pengunggahan terhadap Amazon S3 menggunakan peran SageMaker eksekusi Anda secara terpisah dari wadah pelatihan atau inferensi.

SageMaker Kontainer terkelola berikut tidak mendukung isolasi jaringan karena memerlukan akses ke Amazon S3:

- Chainer
- SageMaker Pembelajaran Penguatan

Isolasi jaringan dengan VPC

Isolasi jaringan dapat digunakan bersama dengan VPC. Dalam skenario ini, pengunduhan dan pengunggahan data pelanggan dan artefak model dirutekan melalui subnet VPC Anda. Namun, wadah pelatihan dan inferensi itu sendiri terus diisolasi dari jaringan, dan tidak memiliki akses ke sumber daya apa pun dalam VPC Anda atau di internet.

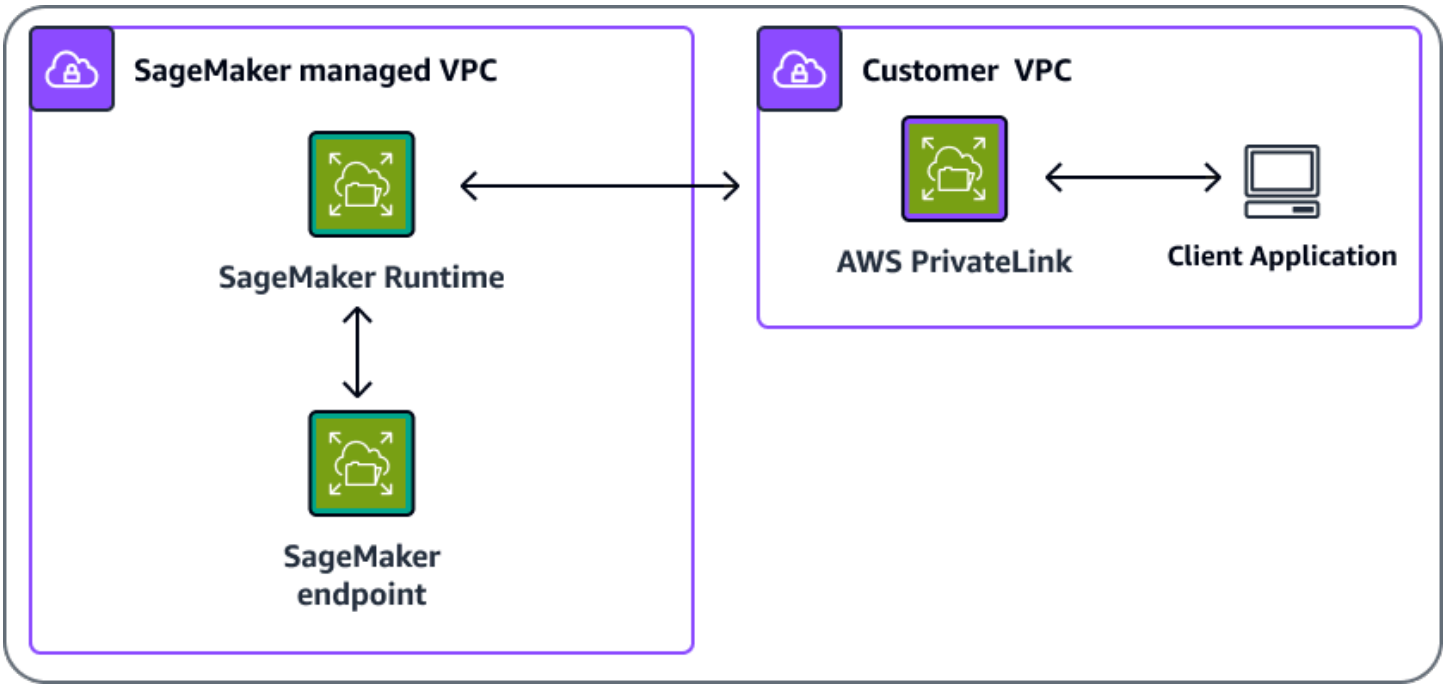
Connect ke SageMaker Dalam VPC

Anda dapat terhubung langsung ke SageMaker API atau Amazon SageMaker Runtime melalui titik akhir antarmuka di virtual private cloud (VPC) alih-alih terhubung melalui internet. Ketika Anda menggunakan titik akhir antarmuka VPC, komunikasi antara VPC dan SageMaker API atau Runtime dilakukan sepenuhnya dan dengan aman di jaringan. AWS

Connect ke SageMaker melalui titik akhir antarmuka VPC

SageMaker API dan SageMaker Runtime mendukung titik akhir antarmuka Amazon Virtual Private Cloud (Amazon VPC) yang didukung oleh. AWS PrivateLink Masing-masing VPC endpoint diwakili oleh satu Antarmuka Jaringan Elastis dengan alamat IP privat di subnet VPC Anda. Misalnya, aplikasi di dalam VPC Anda digunakan AWS PrivateLink untuk berkomunikasi dengan SageMaker Runtime. SageMakerRuntime pada gilirannya berkomunikasi dengan titik akhir. SageMaker Menggunakan

AWS PrivateLink memungkinkan Anda untuk memanggil SageMaker titik akhir Anda dari dalam VPC Anda, seperti yang ditunjukkan pada diagram berikut.



Titik akhir antarmuka VPC menghubungkan VPC Anda langsung ke SageMaker API atau SageMaker Runtime AWS PrivateLink tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak perlu terhubung ke internet publik untuk berkomunikasi dengan API SageMaker atau SageMaker Runtime.

Anda dapat membuat titik akhir AWS PrivateLink antarmuka untuk terhubung ke SageMaker atau ke SageMaker Runtime menggunakan AWS Management Console atau AWS Command Line Interface ()AWS CLI. Untuk petunjuk, lihat Mengakses AWS layanan menggunakan titik akhir VPC antarmuka.

Jika Anda belum mengaktifkan nama host Sistem Nama Domain (DNS) pribadi untuk titik akhir VPC Anda, setelah Anda membuat titik akhir VPC, tentukan URL titik akhir internet ke API atau Runtime. SageMaker SageMaker Contoh kode menggunakan AWS CLI perintah untuk menentukan `endpoint-url` parameter berikut.

```
aws sagemaker list-notebook-instances --endpoint-
url VPC_Endpoint_ID.api.sagemaker.Region.vpce.amazonaws.com

aws sagemaker list-training-jobs --endpoint-
url VPC_Endpoint_ID.api.sagemaker.Region.vpce.amazonaws.com

aws sagemaker-runtime invoke-endpoint --endpoint-url
 https://VPC_Endpoint_ID.runtime.sagemaker.Region.vpce.amazonaws.com  \
```

```
    --endpoint-name Endpoint_Name \
    --body "Endpoint_Body" \
    --content-type "Content_Type" \
            Output_File
```

Jika Anda mengaktifkan nama host DNS privat untuk VPC endpoint Anda, Anda tidak perlu menentukan URL endpoint URL karena nama host default (https://api.sagemaker. *Wilayah* .amazon.com) menyelesaikan ke VPC endpoint Anda. Demikian pula, nama host DNS SageMaker Runtime default (https://runtime.sagemaker. *Region* .amazonaws.com) juga menyelesaikan ke VPC endpoint Anda.

SageMaker API dan SageMaker Runtime mendukung titik akhir VPC di semua Wilayah AWS tempat Amazon VPC dan tersedia. SageMaker SageMaker mendukung panggilan ke semua yang ada `Operations`di dalam VPC Anda. Jika Anda menggunakan `AuthorizedUrl` dari CreatePresignedNotebookInstanceUrlperintah, lalu lintas Anda akan melalui internet publik. Anda tidak hanya dapat menggunakan titik akhir VPC untuk mengakses URL yang telah ditetapkan sebelumnya, permintaan harus melalui gateway internet.

Secara default, pengguna Anda dapat membagikan URL yang telah ditetapkan sebelumnya kepada orang-orang di luar jaringan perusahaan Anda. Untuk keamanan tambahan, Anda harus menambahkan izin IAM untuk membatasi URL hanya dapat digunakan dalam jaringan Anda. Untuk informasi tentang izin IAM, lihat Cara AWS PrivateLink kerja dengan IAM.

> **Note**
>
> Saat menyiapkan titik akhir antarmuka VPC untuk layanan SageMaker Runtime (https://runtime.sagemaker. *Region*.amazonaws.com), Anda harus memastikan bahwa titik akhir antarmuka VPC diaktifkan di Availability Zone klien Anda agar resolusi DNS pribadi berfungsi. Jika tidak, Anda mungkin melihat kegagalan DNS saat mencoba menyelesaikan URL.

Untuk mempelajari selengkapnya tentang AWS PrivateLink, lihat AWS PrivateLink dokumentasi. Lihat AWS PrivateLinkHarga untuk harga VPC endpoints. Untuk mempelajari selengkapnya tentang VPC dan titik akhir, lihat Amazon VPC. Untuk informasi tentang cara menggunakan AWS Identity and Access Management kebijakan berbasis identitas untuk membatasi akses ke SageMaker API dan SageMaker Runtime, lihat. Kontrol Akses ke SageMaker API dengan Menggunakan Kebijakan Berbasis Identitas

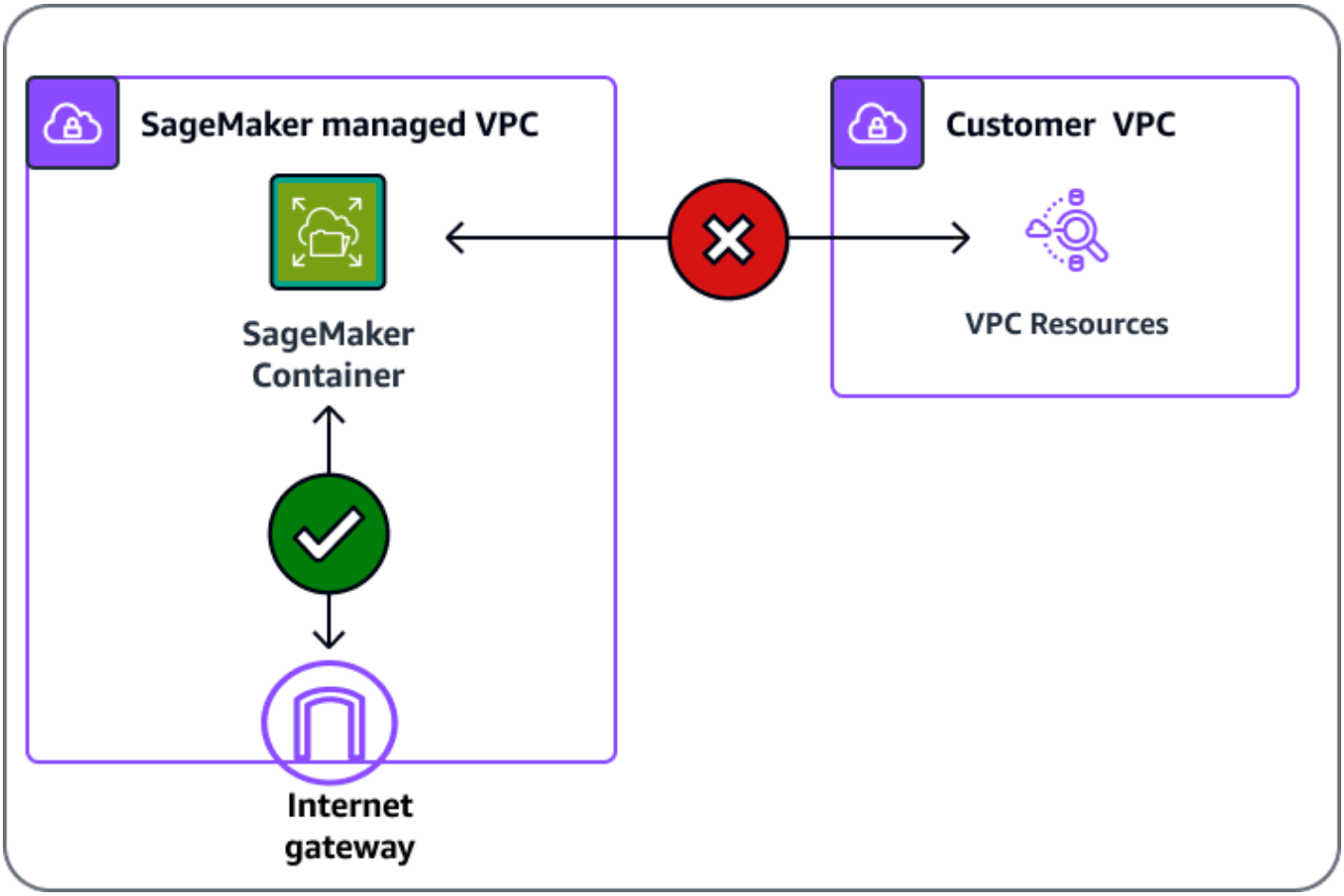
Menggunakan SageMaker pelatihan dan hosting dengan sumber daya di dalam VPC Anda

SageMaker menggunakan peran eksekusi Anda untuk mengunduh dan mengunggah informasi dari bucket Amazon S3 dan Amazon Elastic Container Registry (Amazon ECR) Registry ECR), terpisah dari wadah pelatihan atau inferensi Anda. Jika Anda memiliki sumber daya yang berada di dalam VPC Anda, Anda masih dapat memberikan SageMaker akses ke sumber daya tersebut. Bagian berikut menjelaskan cara membuat sumber daya Anda tersedia SageMaker dengan atau tanpa isolasi jaringan.

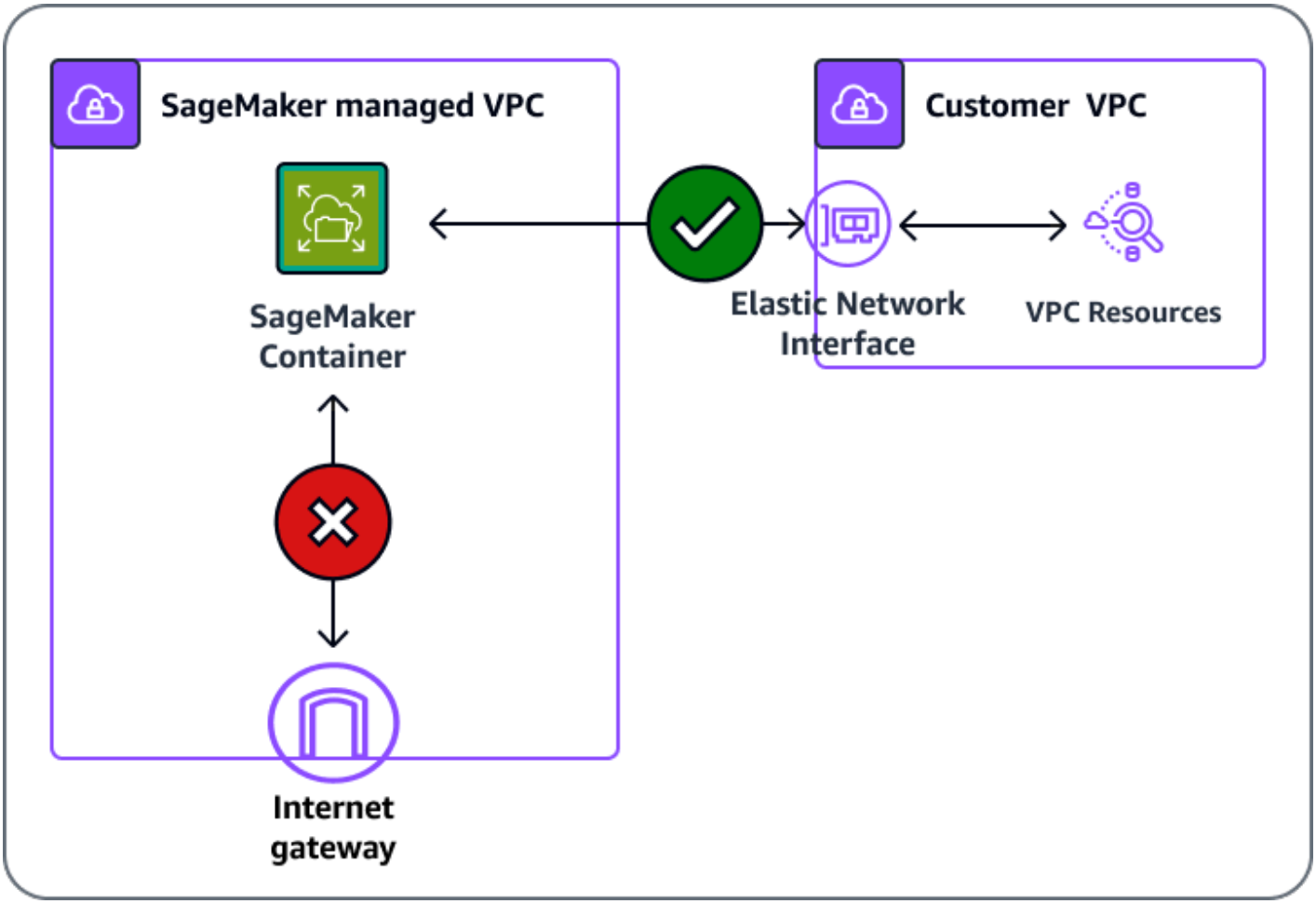
Tanpa isolasi jaringan diaktifkan

Jika Anda belum mengatur isolasi jaringan pada pekerjaan atau model pelatihan Anda, SageMaker dapat mengakses sumber daya menggunakan salah satu metode berikut.

- SageMaker pelatihan dan wadah inferensi yang diterapkan dapat mengakses internet secara default. SageMaker kontainer dapat mengakses layanan dan sumber daya eksternal di internet publik sebagai bagian dari beban kerja pelatihan dan inferensi Anda. SageMaker kontainer tidak dapat mengakses sumber daya di dalam VPC Anda tanpa konfigurasi VPC, seperti yang ditunjukkan pada ilustrasi berikut.

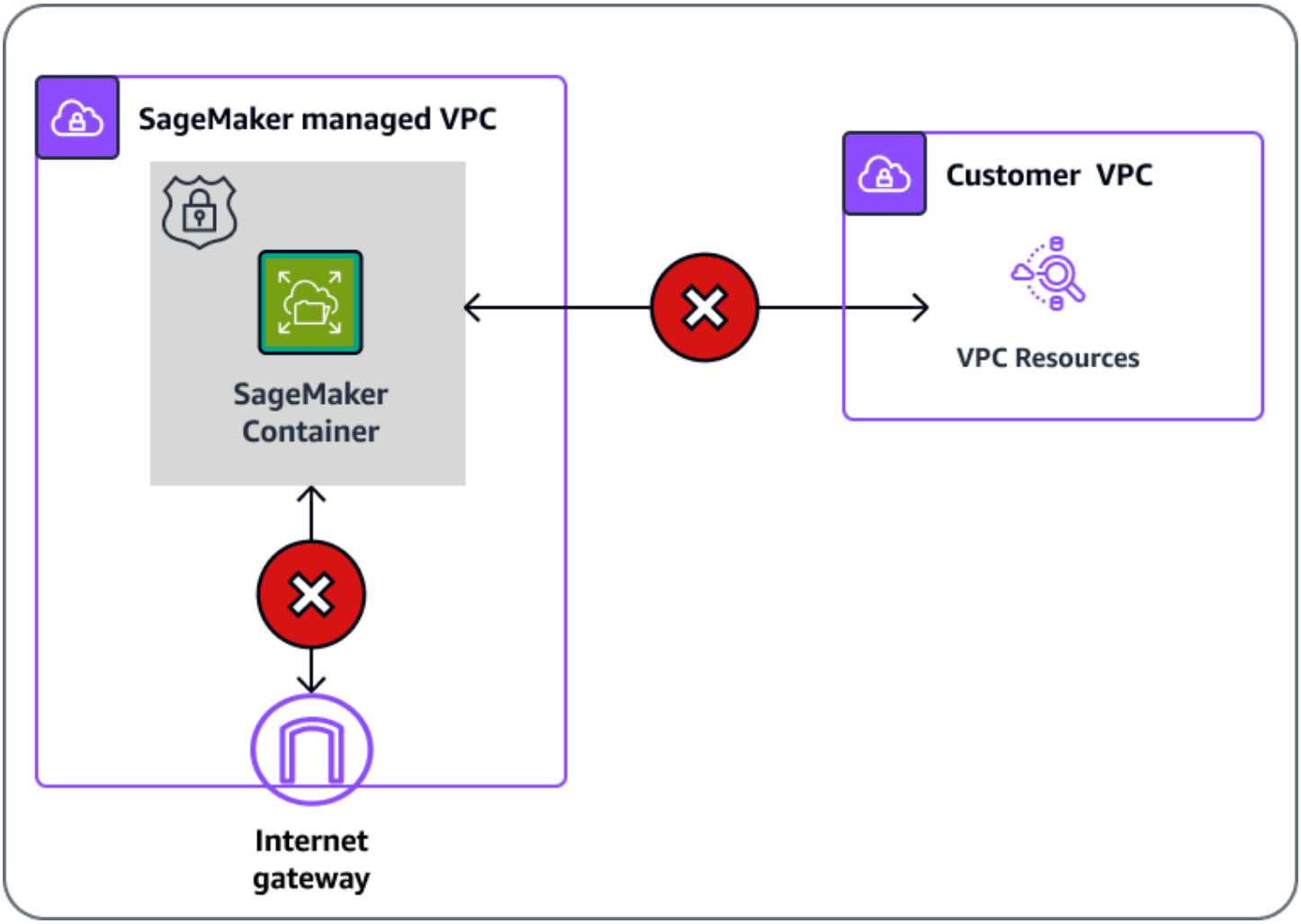


- Gunakan konfigurasi VPC untuk berkomunikasi dengan sumber daya di dalam VPC Anda melalui elastic network interface (ENI). Komunikasi antara wadah dan sumber daya di VPC Anda berlangsung dengan aman di dalam jaringan VPC Anda, seperti yang ditunjukkan pada ilustrasi berikut. Dalam hal ini, Anda mengelola akses jaringan ke sumber daya VPC dan internet Anda.



Dengan isolasi jaringan

Jika Anda menggunakan isolasi jaringan, SageMaker penampung tidak dapat berkomunikasi dengan sumber daya di dalam VPC Anda atau melakukan panggilan jaringan apa pun, seperti yang ditunjukkan pada ilustrasi berikut. Jika Anda menyediakan konfigurasi VPC, operasi pengunduhan dan pengunggahan akan dijalankan melalui VPC Anda. Untuk informasi selengkapnya tentang hosting dan pelatihan dengan isolasi jaringan saat menggunakan VPC, lihat. Isolasi Jaringan



Membuat Kebijakan VPC Endpoint untuk SageMaker

Anda dapat membuat kebijakan untuk Amazon VPC endpoint SageMaker untuk menentukan berikut ini:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat Mengontrol Akses ke Layanan dengan VPC Endpoint dalam Panduan Pengguna Amazon VPC.

> **Note**
>
> Kebijakan VPC endpoint tidak didukung untuk titik akhir waktu proses Federal Information Processing Standard (FIPS) SageMaker . runtime_InvokeEndpoint

Contoh kebijakan VPC endpoint berikut menentukan bahwa semua pengguna yang memiliki akses ke titik akhir antarmuka VPC diizinkan untuk meminta titik akhir yang di-hosting bernama. SageMaker `myEndpoint`

```
{
  "Statement": [
      {
          "Action": "sagemaker:InvokeEndpoint",
          "Effect": "Allow",
          "Resource": "arn:aws:sagemaker:us-west-2:123456789012:endpoint/myEndpoint",
          "Principal": "*"
      }
  ]
}
```

Dalam contoh ini, hal-hal berikut ditolak:

- Tindakan SageMaker API lainnya, seperti `sagemaker:CreateEndpoint` dan`sagemaker:CreateTrainingJob`.
- Memanggil titik akhir yang SageMaker dihosting selain. `myEndpoint`

> **Note**
>
> Dalam contoh ini, pengguna masih dapat mengambil tindakan SageMaker API lainnya dari luar VPC. Untuk informasi tentang cara membatasi panggilan API ke panggilan dari dalam VPC, lihat. Kontrol Akses ke SageMaker API dengan Menggunakan Kebijakan Berbasis Identitas

Buat Kebijakan VPC Endpoint untuk Amazon Feature Store SageMaker

Untuk membuat Endpoint VPC untuk Amazon SageMaker Feature Store, gunakan template endpoint berikut, ganti VPC_ENDPOINT_ID.API dan Region Anda:

```
VPC_Endpoint_ID.api.featurestore-
runtime.sagemaker.Region.vpce.amazonaws.com
```

Connect ke SageMaker Studio Classic Melalui VPC Endpoint Antarmuka

Anda dapat terhubung ke Amazon SageMaker Studio Classic dari Amazon Virtual Private Cloud (Amazon VPC) Anda melalui titik akhir antarmuka di VPC Anda alih-alih terhubung melalui internet. Ketika Anda menggunakan VPC endpoint antarmuka (titik akhir antarmuka), komunikasi antara VPC dan Studio Classic dilakukan sepenuhnya dan dengan aman di jaringan. AWS

SageMaker Studio Classic mendukung titik akhir antarmuka yang didukung oleh AWS PrivateLink. Masing-masing titik akhir antarmuka diwakili oleh satu Antarmuka jaringan elastis dengan alamat IP privat di subnet VPC Anda.

Studio Classic mendukung titik akhir antarmuka di semua AWS Wilayah di mana Amazon SageMaker dan Amazon VPC tersedia.

Topik

- Membuat VPC Endpoint
- Buat Kebijakan VPC Endpoint untuk Studio Classic SageMaker
- Izinkan Akses Hanya dari Dalam VPC Anda

Membuat VPC Endpoint

Anda dapat membuat titik akhir antarmuka untuk terhubung ke Studio Classic dengan AWS konsol atau AWS Command Line Interface (AWS CLI). Untuk instruksi, lihat Membuat titik akhir antarmuka. Pastikan Anda membuat titik akhir antarmuka untuk semua subnet di VPC yang ingin Anda sambungkan ke Studio Classic.

Saat Anda membuat titik akhir antarmuka, pastikan bahwa grup keamanan di titik akhir Anda mengizinkan akses masuk untuk lalu lintas HTTPS dari grup keamanan yang terkait dengan SageMaker Studio Classic. Untuk informasi selengkapnya, lihat Mengontrol akses ke layanan dengan VPC endpoint.

> **Note**
>
> Selain membuat titik akhir antarmuka untuk terhubung ke SageMaker Studio Classic, buat titik akhir antarmuka untuk terhubung ke Amazon SageMaker API. Ketika pengguna

memanggil `CreatePresignedDomainUrl`untuk mendapatkan URL untuk terhubung ke Studio Classic, panggilan itu melewati titik akhir antarmuka yang digunakan untuk terhubung ke SageMaker API.

Saat Anda membuat titik akhir antarmuka, tentukan **`aws.sagemaker.`*`Region`*`.studio`** sebagai nama layanan. Setelah Anda membuat titik akhir antarmuka, aktifkan DNS pribadi untuk titik akhir Anda. Saat Anda terhubung ke SageMaker Studio Classic dari dalam VPC menggunakan SageMaker API, theAWS CLI, atau konsol, Anda terhubung melalui titik akhir antarmuka alih-alih internet publik. Anda juga perlu menyiapkan DNS khusus dengan zona yang dihosting pribadi untuk titik akhir VPC Amazon SageMaker agar Studio Classic dapat mengakses API menggunakan titik SageMaker akhir daripada menggunakan `api.sagemaker.$region.amazonaws.com` URL titik akhir VPC. Untuk petunjuk tentang cara menyiapkan zona host pribadi, lihat Bekerja dengan zona yang dihosting pribadi.

Buat Kebijakan VPC Endpoint untuk Studio Classic SageMaker

Anda dapat melampirkan kebijakan titik akhir VPC Amazon ke titik akhir VPC antarmuka yang Anda gunakan untuk menyambung ke Studio Classic. SageMaker Kebijakan endpoint mengontrol akses ke Studio Classic. Anda dapat menentukan sebagai berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk menggunakan titik akhir VPC dengan SageMaker Studio Classic, kebijakan endpoint Anda harus mengizinkan `CreateApp` pengoperasian pada jenis aplikasi. KernelGateway Ini memungkinkan lalu lintas yang diarahkan ke melalui titik akhir VPC untuk memanggil API. `CreateApp` Contoh kebijakan VPC endpoint berikut menunjukkan cara mengizinkan operasi. `CreateApp`

```
{
 "Statement": [
   {
     "Action": "sagemaker:CreateApp",
     "Effect": "Allow",
     "Resource": "arn:aws:sagemaker:us-west-2:acct-id:app/domain-id/*",
     "Principal": "*"
```

```
    }
  ]
}
```

Untuk informasi selengkapnya, lihat Mengendalikan akses ke layanan dengan VPC endpoint.

Contoh kebijakan VPC endpoint berikut menentukan bahwa semua pengguna yang memiliki akses ke titik akhir diizinkan untuk mengakses profil pengguna di SageMaker domain dengan ID domain tertentu. Akses ke domain lain ditolak.

```
{
  "Statement": [
      {
          "Action": "sagemaker:CreatePresignedDomainUrl",
          "Effect": "Allow",
          "Resource": "arn:aws:sagemaker:us-west-2:acct-id:user-profile/domain-id/*",
          "Principal": "*"
      }
  ]
}
```

Izinkan Akses Hanya dari Dalam VPC Anda

Pengguna di luar VPC Anda dapat terhubung ke SageMaker Studio Classic melalui internet bahkan jika Anda menyiapkan titik akhir antarmuka di VPC Anda.

Untuk mengizinkan akses hanya ke koneksi yang dibuat dari dalam VPC Anda, buat kebijakan AWS Identity and Access Management (IAM) untuk efek tersebut. Tambahkan kebijakan tersebut ke setiap pengguna, grup, atau peran yang digunakan untuk mengakses Studio Classic. Fitur ini hanya didukung dalam mode IAM, dan tidak didukung dalam mode Pusat Identitas IAM. Contoh berikut menunjukkan cara membuat kebijakan tersebut.

> **Important**
>
> Jika Anda menerapkan kebijakan IAM yang mirip dengan salah satu contoh berikut, pengguna tidak dapat mengakses SageMaker Studio Classic atau SageMaker API yang ditentukan melalui SageMaker konsol. Untuk mengakses Studio Classic, pengguna harus menggunakan URL yang telah ditetapkan sebelumnya atau memanggil SageMaker API secara langsung.

Contoh 1: Izinkan koneksi hanya dalam subnet titik akhir antarmuka

Kebijakan berikut hanya mengizinkan koneksi ke penelepon dalam subnet tempat Anda membuat titik akhir antarmuka.

```
{
    "Id": "sagemaker-studio-example-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable SageMaker Studio Access",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedDomainUrl",
                "sagemaker:DescribeUserProfile"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpc": "vpc-111bbaaa"
                }
            }
        }
    ]
}
```

Contoh 2: Izinkan koneksi hanya melalui titik akhir antarmuka menggunakan **`aws:sourceVpce`**

Kebijakan berikut hanya mengizinkan koneksi ke koneksi yang dibuat melalui titik akhir antarmuka yang ditentukan oleh kunci `aws:sourceVpce` kondisi. Misalnya, titik akhir antarmuka pertama dapat memungkinkan akses melalui SageMaker konsol. Titik akhir antarmuka kedua dapat memungkinkan akses melalui SageMaker API.

```
{
    "Id": "sagemaker-studio-example-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable SageMaker Studio Access",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedDomainUrl",
```

```
                "sagemaker:DescribeUserProfile"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:sourceVpce": [
                        "vpce-111bbccc",
                        "vpce-111bbddd"
                    ]
                }
            }
        }
    ]
}
```

Kebijakan ini mencakup DescribeUserProfiletindakan. Biasanya Anda menelepon `DescribeUserProfile` untuk memastikan bahwa status profil pengguna `InService` sebelum Anda mencoba untuk terhubung ke domain. Sebagai contoh:

```
aws sagemaker describe-user-profile \
    --domain-id domain-id \
    --user-profile-name profile-name
```

Respons:

```
{
    "DomainId": "domain-id",
    "UserProfileArn": "arn:aws:sagemaker:us-west-2:acct-id:user-profile/domain-id/
profile-name",
    "UserProfileName": "profile-name",
    "HomeEfsFileSystemUid": "200001",
    "Status": "InService",
    "LastModifiedTime": 1605418785.555,
    "CreationTime": 1605418477.297
}
```

```
aws sagemaker create-presigned-domain-url
    --domain-id domain-id \
    --user-profile-name profile-name
```

Respons:

```
{
    "AuthorizedUrl": "https://domain-id.studio.us-west-2.sagemaker.aws/auth?
token=AuthToken"
}
```

Untuk kedua panggilan ini, jika Anda menggunakan versi AWS SDK yang dirilis sebelum 13 Agustus 2018, Anda harus menentukan URL titik akhir dalam panggilan. Misalnya, contoh berikut menunjukkan panggilan ke`create-presigned-domain-url`:

```
aws sagemaker create-presigned-domain-url
    --domain-id domain-id \
    --user-profile-name profile-name \
    --endpoint-url vpc-endpoint-id.api.sagemaker.Region.vpce.amazonaws.com
```

Contoh 3: Izinkan koneksi dari alamat IP menggunakan **`aws:SourceIp`**

Kebijakan berikut mengizinkan koneksi hanya dari rentang alamat IP yang ditentukan menggunakan kunci `aws:SourceIp` kondisi.

```
{
    "Id": "sagemaker-studio-example-3",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable SageMaker Studio Access",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedDomainUrl",
                "sagemaker:DescribeUserProfile"
            ],
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            }
        }
    ]
```

```
}
```

Contoh 4: Izinkan koneksi dari alamat IP melalui titik akhir antarmuka menggunakan **`aws:VpcSourceIp`**

Jika Anda mengakses SageMaker Studio Classic melalui titik akhir antarmuka, Anda dapat menggunakan tombol `aws:VpcSourceIp` kondisi untuk mengizinkan koneksi hanya dari rentang alamat IP yang ditentukan dalam subnet tempat Anda membuat titik akhir antarmuka seperti yang ditunjukkan dalam kebijakan berikut:

```
{
    "Id": "sagemaker-studio-example-4",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable SageMaker Studio Access",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedDomainUrl",
                "sagemaker:DescribeUserProfile"
            ],
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:VpcSourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                },
                "StringEquals": {
                    "aws:SourceVpc": "vpc-111bbaaa"
                }
            }
        }
    ]
}
```

Connect ke Instance Notebook Melalui Endpoint Antarmuka VPC

Anda dapat connect ke instans notebook Anda dari VPC Anda melalui titik akhir antarmuka di Virtual Private Cloud (VPC) alih-alih terhubung melalui internet publik. Ketika Anda menggunakan titik akhir

antarmuka VPC, komunikasi antara VPC dan instans notebook dilakukan sepenuhnya dan dengan aman di jaringan. AWS

SageMaker instans notebook mendukung titik akhir antarmuka Amazon Virtual Private Cloud (Amazon VPC) yang didukung oleh. AWS PrivateLink Masing-masing VPC endpoint diwakili oleh satu Antarmuka Jaringan Elastis dengan alamat IP privat di subnet VPC Anda.

> **Note**
>
> Sebelum Anda membuat titik akhir VPC antarmuka untuk terhubung ke instance notebook, buat titik akhir VPC antarmuka untuk terhubung ke API. SageMaker Dengan begitu, saat pengguna menelepon CreatePresignedNotebookInstanceUrluntuk mendapatkan URL untuk terhubung ke instance notebook, panggilan itu juga melewati titik akhir VPC antarmuka. Untuk informasi, lihat Connect ke SageMaker Dalam VPC.

Anda dapat membuat titik akhir antarmuka untuk terhubung ke instans notebook Anda dengan perintah AWS Management Console atau AWS Command Line Interface (AWS CLI). Untuk instruksi, lihat Membuat Titik Akhir Antarmuka. Pastikan Anda membuat titik akhir antarmuka untuk semua subnet di VPC yang ingin Anda sambungkan ke instance notebook.

Saat Anda membuat titik akhir antarmuka, tentukan aws.sagemaker. ***Region*** .notebook sebagai nama layanan. Setelah Anda membuat titik akhir VPC, aktifkan DNS pribadi untuk titik akhir VPC Anda. Siapa pun yang menggunakan SageMaker APIAWS CLI, the, atau konsol untuk terhubung ke instance notebook dari dalam VPC terhubung ke instance notebook melalui titik akhir VPC alih-alih internet publik.

SageMaker instance notebook mendukung titik akhir VPC di semua Wilayah AWS tempat Amazon VPC dan tersedia. SageMaker

Topik

- Hubungkan Jaringan Pribadi Anda ke VPC
- Buat Kebijakan VPC Endpoint untuk Instans VPC Endpoint untuk Instans Notebook SageMaker
- Batasi Akses ke Koneksi dari Dalam VPC Anda

Hubungkan Jaringan Pribadi Anda ke VPC

Untuk terhubung ke instans notebook Anda melalui VPC Anda, Anda harus terhubung dari instans yang ada di dalam VPC, atau menghubungkan jaringan privat Anda ke VPC Anda dengan menggunakan () atau. AWS Virtual Private Network AWS VPN AWS Direct Connect Untuk informasi selengkapnyaAWS VPN, lihat Koneksi VPN di Panduan Pengguna Amazon Virtual Private Cloud. Untuk selengkapnyaAWS Direct Connect, lihat Membuat Sambungan di Panduan Pengguna AWS Direct Connect.

Buat Kebijakan VPC Endpoint untuk Instans VPC Endpoint untuk Instans Notebook SageMaker

Anda dapat membuat kebijakan untuk Amazon VPC endpoint untuk instans SageMaker notebook untuk menentukan hal berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat Mengontrol Akses ke Layanan dengan VPC Endpoint dalam Panduan Pengguna Amazon VPC.

Contoh kebijakan VPC endpoint berikut menentukan bahwa semua pengguna yang memiliki akses ke titik akhir diizinkan untuk mengakses instans notebook bernama. `myNotebookInstance`

```
{
  "Statement": [
      {
          "Action": "sagemaker:CreatePresignedNotebookInstanceUrl",
          "Effect": "Allow",
          "Resource": "arn:aws:sagemaker:us-west-2:123456789012:notebook-instance/
myNotebookInstance",
          "Principal": "*"
      }
  ]
}
```

Akses ke instance notebook lainnya ditolak.

Batasi Akses ke Koneksi dari Dalam VPC Anda

Bahkan jika Anda mengatur titik akhir antarmuka di VPC Anda, individu di luar VPC dapat terhubung ke instance notebook melalui internet.

> **Important**
>
> Jika Anda menerapkan kebijakan IAM yang mirip dengan salah satu dari berikut ini, pengguna tidak dapat mengakses SageMaker API yang ditentukan atau instance notebook melalui konsol.

Untuk membatasi akses hanya ke koneksi yang dibuat dari dalam VPC Anda, buat kebijakan AWS Identity and Access Management yang membatasi akses hanya ke panggilan yang berasal dari dalam VPC Anda. Kemudian tambahkan kebijakan tersebut ke setiap AWS Identity and Access Management pengguna, grup, atau peran yang digunakan untuk mengakses instance notebook.

> **Note**
>
> Kebijakan ini mengizinkan koneksi hanya ke penelepon dalam subnet tempat Anda membuat titik akhir antarmuka.

```
{
    "Id": "notebook-example-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable Notebook Access",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedNotebookInstanceUrl",
                "sagemaker:DescribeNotebookInstance"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpc": "vpc-111bbaaa"
                }
            }
        }
```

```
    ]
}
```

Jika Anda ingin membatasi akses ke instance notebook hanya untuk koneksi yang dibuat menggunakan titik akhir antarmuka, gunakan tombol `aws:SourceVpce` kondisi alih-alih `aws:SourceVpc`:

```
{
    "Id": "notebook-example-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable Notebook Access",
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreatePresignedNotebookInstanceUrl",
                "sagemaker:DescribeNotebookInstance"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:sourceVpce": [
                        "vpce-111bbccc",
                        "vpce-111bbddd"
                    ]
                }
            }
        }
    ]
}
```

Kedua contoh kebijakan ini mengasumsikan bahwa Anda juga telah membuat titik akhir antarmuka untuk SageMaker API. Untuk informasi selengkapnya, lihat Connect ke SageMaker Dalam VPC. Pada contoh kedua, salah satu nilai untuk `aws:SourceVpce` adalah ID titik akhir antarmuka untuk instance notebook. Yang lainnya adalah ID titik akhir antarmuka untuk SageMaker API.

Contoh kebijakan di sini termasuk DescribeNotebookInstance, karena biasanya Anda akan menelepon `DescribeNotebookInstance` untuk memastikan bahwa `NotebookInstanceStatus` ada `InService` sebelum Anda mencoba menghubungkannya. Sebagai contoh:

```
aws sagemaker describe-notebook-instance \
```

```
                    --notebook-instance-name myNotebookInstance


{
   "NotebookInstanceArn":
   "arn:aws:sagemaker:us-west-2:1234567890ab:notebook-instance/mynotebookinstance",
   "NotebookInstanceName": "myNotebookInstance",
   "NotebookInstanceStatus": "InService",
   "Url": "mynotebookinstance.notebook.us-west-2.sagemaker.aws",
   "InstanceType": "ml.m4.xlarge",
   "RoleArn":
   "arn:aws:iam::1234567890ab:role/service-role/AmazonSageMaker-
ExecutionRole-12345678T123456",
   "LastModifiedTime": 1540334777.501,
   "CreationTime": 1523050674.078,
   "DirectInternetAccess": "Disabled"
}
aws sagemaker create-presigned-notebook-instance-url --notebook-instance-name
 myNotebookInstance


{
   "AuthorizedUrl": "https://mynotebookinstance.notebook.us-west-2.sagemaker.aws?
authToken=AuthToken
}
```

> **Note**
>
> Yang`presigned-notebook-instance-url`,`AuthorizedUrl`, dihasilkan dapat digunakan dari mana saja di internet.

Untuk kedua panggilan ini, jika Anda tidak mengaktifkan nama host DNS pribadi untuk titik akhir VPC Anda, atau jika Anda menggunakan versi AWS SDK yang dirilis sebelum 13 Agustus 2018, Anda harus menentukan URL titik akhir dalam panggilan. Misalnya, panggilan ke `create-presigned-notebook-instance-url` adalah:

```
aws sagemaker create-presigned-notebook-instance-url
    --notebook-instance-name myNotebookInstance --endpoint-url
    VPC_Endpoint_ID.api.sagemaker.Region.vpce.amazonaws.com
```

Hubungkan Jaringan Pribadi Anda ke VPC

Untuk memanggil SageMaker API dan SageMaker Runtime melalui VPC Anda, Anda harus terhubung dari instans yang ada di dalam VPC atau menghubungkan jaringan privat Anda ke VPC Anda dengan menggunakan () atau. AWS Virtual Private Network AWS VPN AWS Direct Connect Untuk informasi selengkapnyaAWS VPN, lihat Koneksi VPN di Panduan Pengguna Amazon Virtual Private Cloud. Untuk selengkapnyaAWS Direct Connect, lihat Membuat Sambungan di Panduan Pengguna AWS Direct Connect.

Berikan SageMaker Akses ke Sumber Daya di VPC Amazon Anda

SageMaker menjalankan tipe pekerjaan berikut di Amazon Virtual Private Cloud secara default.

- Pemrosesan
- Pelatihan
- Hosting model
- Transformasi Batch
- Amazon SageMaker Klarifikasi
- SageMaker Kompilasi

Namun, wadah untuk pekerjaan ini mengakses AWS sumber daya—seperti bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) tempat Anda menyimpan data pelatihan dan artefak model—melalui internet.

Untuk mengontrol akses ke data dan wadah pekerjaan Anda, kami sarankan Anda membuat VPC pribadi dan mengonfigurasinya sehingga tidak dapat diakses melalui internet. Untuk informasi tentang membuat dan mengonfigurasi VPC, lihat Memulai Amazon VPC di Panduan Pengguna Amazon VPC. Menggunakan VPC membantu melindungi wadah dan data pekerjaan Anda karena Anda dapat mengonfigurasi VPC Anda sehingga tidak terhubung ke internet. Menggunakan VPC juga memungkinkan Anda untuk memantau semua lalu lintas jaringan masuk dan keluar dari wadah pekerjaan Anda dengan menggunakan log aliran VPC. Untuk informasi selengkapnya, lihat Log Alur VPC di Panduan Pengguna Amazon VPC.

Anda menentukan konfigurasi VPC pribadi saat membuat pekerjaan dengan menentukan subnet dan grup keamanan. Saat Anda menentukan subnet dan grup keamanan, SageMaker buat antarmuka jaringan elastis yang terkait dengan grup keamanan Anda di salah satu subnet. Antarmuka jaringan

memungkinkan wadah pekerjaan Anda terhubung ke sumber daya di VPC Anda. Untuk informasi tentang antarmuka jaringan, lihat Antarmuka Jaringan Elastis di Panduan Pengguna Amazon VPC.

Anda menentukan konfigurasi VPC dalam `VpcConfig` objek CreateProcessingJoboperasi atau CreateTrainingJoboperasi. Menentukan konfigurasi VPC saat Anda membuat pekerjaan pelatihan memberi model Anda akses ke sumber daya dalam VPC Anda.

Menentukan konfigurasi VPC saja tidak mengubah jalur pemanggilan. Untuk terhubung ke Amazon SageMaker dalam VPC, buat titik akhir VPC dan panggil. Untuk informasi selengkapnya, lihat Connect ke SageMaker Dalam VPC.

Topik

- Berikan Akses Pekerjaan SageMaker Pemrosesan ke Sumber Daya di VPC Amazon Anda
- Berikan Akses Pekerjaan SageMaker Pelatihan ke Sumber Daya di VPC Amazon Anda
- Berikan Akses Endpoint yang SageMaker Dihosting ke Sumber Daya di VPC Amazon Anda
- Berikan Akses Pekerjaan Transformasi Batch ke Sumber Daya di VPC Amazon Anda
- Berikan Amazon SageMaker Clarify Lowongan Akses ke Sumber Daya di Amazon VPC Anda
- Berikan Akses Pekerjaan SageMaker Kompilasi ke Sumber Daya di VPC Amazon Anda
- Berikan Akses Pekerjaan Inferensi Rekomendasi ke Sumber Daya di VPC Amazon Anda

Berikan Akses Pekerjaan SageMaker Pemrosesan ke Sumber Daya di VPC Amazon Anda

Untuk mengontrol akses ke data dan pekerjaan pemrosesan Anda, buat VPC Amazon dengan subnet pribadi. Untuk informasi tentang membuat dan mengonfigurasi VPC, lihat Memulai Dengan Amazon VPC di Panduan Pengguna Amazon VPC.

Anda dapat memantau semua lalu lintas jaringan masuk dan keluar dari wadah pemrosesan Anda dengan menggunakan log aliran VPC. Untuk informasi selengkapnya, lihat Log Alur VPC di Panduan Pengguna Amazon VPC.

Dokumen ini menjelaskan cara menambahkan konfigurasi VPC Amazon untuk memproses pekerjaan.

Konfigurasikan Pekerjaan Pemrosesan untuk Akses VPC Amazon

Anda mengonfigurasi pekerjaan pemrosesan dengan menentukan subnet dan ID grup keamanan dalam VPC. Anda tidak perlu menentukan subnet untuk wadah pemrosesan. Amazon SageMaker

secara otomatis menarik wadah pemrosesan dari Amazon ECR. Untuk informasi selengkapnya tentang pemrosesan kontainer, lihatMemproses data.

Saat membuat pekerjaan pemrosesan, Anda dapat menentukan subnet dan grup keamanan di VPC menggunakan konsol atau API. SageMaker

Untuk menggunakan API, Anda menentukan subnet dan ID grup keamanan dalam `NetworkConfig.VpcConfig` parameter CreateProcessingJoboperasi. SageMaker menggunakan subnet dan detail grup keamanan untuk membuat antarmuka jaringan dan menempelkannya ke wadah pemrosesan. Antarmuka jaringan menyediakan wadah pemrosesan dengan koneksi jaringan dalam VPC Anda. Ini memungkinkan pekerjaan pemrosesan terhubung ke sumber daya yang ada di VPC Anda.

Berikut ini adalah contoh `VpcConfig` parameter yang Anda sertakan dalam panggilan Anda ke `CreateProcessingJob` operasi:

```
VpcConfig: {
    "Subnets": [
        "subnet-0123456789abcdef0",
        "subnet-0123456789abcdef1",
        "subnet-0123456789abcdef2"
    ],
    "SecurityGroupIds": [
        "sg-0123456789abcdef0"
    ]
}
```

Konfigurasikan VPC Pribadi Anda untuk Pemrosesan SageMaker

Saat mengonfigurasi VPC pribadi untuk pekerjaan pemrosesan SageMaker Anda, gunakan panduan berikut. Untuk informasi tentang menyiapkan VPC, lihat Bekerja dengan VPC dan Subnet di Panduan Pengguna Amazon VPC.

Topik

- Pastikan Subnet Memiliki Alamat IP yang Cukup
- Buat VPC Endpoint Amazon S3
- Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3
- Konfigurasikan Tabel Rute
- Konfigurasikan Grup Keamanan VPC

- Connect ke Sumber Daya di Luar VPC Anda
- Pantau Pekerjaan SageMaker Pemrosesan Amazon dengan CloudWatch Log dan Metrik

Pastikan Subnet Memiliki Alamat IP yang Cukup

Subnet VPC Anda harus memiliki setidaknya dua alamat IP pribadi untuk setiap instance dalam pekerjaan pemrosesan. Untuk informasi selengkapnya, lihat Pengukuran VPC dan subnet untuk IPv4 dalam Panduan Pengguna Amazon VPC.

Buat VPC Endpoint Amazon S3

Jika Anda mengonfigurasi VPC Anda sehingga wadah pemrosesan tidak memiliki akses ke internet, mereka tidak dapat terhubung ke bucket Amazon S3 yang berisi data Anda kecuali Anda membuat titik akhir VPC yang memungkinkan akses. Dengan membuat titik akhir VPC, Anda mengizinkan kontainer pemrosesan mengakses bucket tempat Anda menyimpan data. Kami menyarankan Anda juga membuat kebijakan khusus yang hanya mengizinkan permintaan dari VPC pribadi Anda untuk mengakses bucket S3 Anda. Untuk informasi selengkapnya, lihat Titik Akhir untuk Amazon S3.

Untuk membuat VPC Endpoint S3:

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Buat titik akhir, lalu pilih Buat titik akhir
3. Untuk Nama Layanan, pilih com.amazonaws. ***wilayah*** .s3, di mana *wilayah* adalah nama wilayah tempat VPC Anda berada.
4. Untuk VPC, pilih VPC yang ingin Anda gunakan untuk endpoint ini.
5. Untuk Konfigurasi tabel rute, pilih tabel rute yang akan digunakan oleh titik akhir. Layanan VPC secara otomatis menambahkan rute ke setiap tabel rute yang Anda pilih yang mengarahkan lalu lintas S3 ke titik akhir baru.
6. Untuk Kebijakan, pilih Akses Penuh untuk mengizinkan akses penuh ke layanan S3 oleh pengguna atau layanan apa pun dalam VPC. Pilih Custom untuk membatasi akses lebih lanjut. Untuk informasi, lihat Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3.

Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3

Kebijakan endpoint default memungkinkan akses penuh ke S3 untuk setiap pengguna atau layanan di VPC Anda. Untuk lebih membatasi akses ke S3, buat kebijakan titik akhir kustom. Untuk informasi selengkapnya, lihat Menggunakan Kebijakan titik akhir untuk Amazon S3. Anda juga dapat

menggunakan kebijakan bucket untuk membatasi akses ke bucket S3 hanya untuk lalu lintas yang berasal dari VPC Amazon Anda. Untuk informasi, lihat Menggunakan Kebijakan Bucket Amazon S3.

Batasi Instalasi Package pada Container Processing

Kebijakan endpoint default memungkinkan pengguna untuk menginstal paket dari repositori Amazon Linux dan Amazon Linux 2 pada wadah pemrosesan. Jika Anda tidak ingin pengguna menginstal paket dari repositori itu, buat kebijakan endpoint khusus yang secara eksplisit menolak akses ke repositori Amazon Linux dan Amazon Linux 2. Berikut ini adalah contoh kebijakan yang menolak akses ke repositori ini:

```
{
    "Statement": [
      {
        "Sid": "AmazonLinuxAMIRepositoryAccess",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::packages.*.amazonaws.com/*",
            "arn:aws:s3:::repo.*.amazonaws.com/*"
        ]
      }
    ]
}

{
    "Statement": [
        { "Sid": "AmazonLinux2AMIRepositoryAccess",
          "Principal": "*",
          "Action": [
              "s3:GetObject"
              ],
          "Effect": "Deny",
          "Resource": [
              "arn:aws:s3:::amazonlinux.*.amazonaws.com/*"
              ]
         }
    ]
}
```

Konfigurasikan Tabel Rute

Gunakan pengaturan DNS default untuk tabel rute titik akhir Anda, sehingga URL Amazon S3 standar (misalnya,) teratasi. `http://s3-aws-region.amazonaws.com/MyBucket` Jika Anda tidak menggunakan pengaturan DNS default, pastikan URL yang Anda gunakan untuk menentukan lokasi data dalam pekerjaan pemrosesan Anda diselesaikan dengan mengonfigurasi tabel rute titik akhir. Untuk informasi tentang tabel rute titik akhir VPC, lihat Perutean untuk Titik Akhir Gateway di Panduan Pengguna Amazon VPC.

Konfigurasikan Grup Keamanan VPC

Dalam pemrosesan terdistribusi, Anda harus mengizinkan komunikasi antara wadah yang berbeda dalam pekerjaan pemrosesan yang sama. Untuk melakukan itu, konfigurasikan aturan untuk grup keamanan Anda yang memungkinkan koneksi masuk antara anggota grup keamanan yang sama. Untuk informasi selengkapnya, lihat Aturan Grup Keamanan.

Connect ke Sumber Daya di Luar VPC Anda

Jika Anda menghubungkan model Anda ke sumber daya di luar VPC tempat mereka berjalan, lakukan salah satu hal berikut:

- Connect to other AWS services — Jika model Anda memerlukan akses ke AWS layanan yang mendukung antarmuka Amazon VPC endpoint, buat endpoint untuk terhubung ke layanan tersebut. Untuk daftar layanan yang mendukung titik akhir antarmuka, lihat AWSlayanan yang terintegrasi dengan AWS PrivateLink dalam Panduan AWS PrivateLink Pengguna. Untuk informasi tentang membuat titik akhir VPC antarmuka, lihat Mengakses AWS layanan menggunakan titik akhir VPC antarmuka di Panduan Pengguna. AWS PrivateLink
- Connect to resources through the internet — Jika model Anda berjalan pada instance di VPC Amazon yang tidak memiliki subnet dengan akses ke internet, model tidak akan memiliki akses ke sumber daya di internet. Jika model Anda memerlukan akses ke AWS layanan yang tidak mendukung titik akhir VPC antarmuka, atau ke sumber daya di luarAWS, pastikan Anda menjalankan model Anda di subnet pribadi yang memiliki akses ke internet menggunakan gateway NAT publik di subnet publik. Setelah model Anda berjalan di subnet pribadi, konfigurasikan grup keamanan dan daftar kontrol akses jaringan (NACL) Anda untuk memungkinkan koneksi keluar dari subnet pribadi ke gateway NAT publik di subnet publik. Untuk selengkapnya, lihat gateway NAT di Panduan Pengguna Amazon VPC.

Pantau Pekerjaan SageMaker Pemrosesan Amazon dengan CloudWatch Log dan Metrik

Amazon SageMaker menyediakan CloudWatch log dan metrik Amazon untuk memantau pekerjaan pelatihan. CloudWatch menyediakan CPU, GPU, memori, memori GPU, dan metrik disk, dan pencatatan peristiwa. Untuk informasi selengkapnya tentang memantau pekerjaan SageMaker pemrosesan Amazon, lihat Pantau Amazon SageMaker dengan Amazon CloudWatch danSageMaker Lowongan Kerja dan Metrik Titik Akhir.

Berikan Akses Pekerjaan SageMaker Pelatihan ke Sumber Daya di VPC Amazon Anda

> **Note**
>
> Untuk pekerjaan pelatihan, Anda hanya dapat mengonfigurasi subnet dengan VPC penyewaan default tempat instance Anda berjalan pada perangkat keras bersama. Untuk informasi selengkapnya tentang atribut tenancy untuk VPC, lihat Instans Khusus.

Konfigurasikan Training Job untuk Amazon VPC Access

Untuk mengontrol akses ke pekerjaan pelatihan Anda, jalankan di VPC Amazon dengan subnet pribadi yang tidak memiliki akses internet.

Anda mengonfigurasi tugas pelatihan untuk dijalankan di VPC dengan menentukan subnet dan ID grup keamanannya. Anda tidak perlu menentukan subnet untuk wadah pekerjaan pelatihan. Amazon SageMaker secara otomatis menarik citra kontainer pelatihan dari Amazon ECR.

Saat membuat pekerjaan pelatihan, Anda dapat menentukan subnet dan grup keamanan di VPC menggunakan konsol SageMaker Amazon atau API.

Untuk menggunakan API, Anda menentukan subnet dan ID grup keamanan dalam `VpcConfig` parameter CreateTrainingJoboperasi. SageMaker menggunakan subnet dan detail grup keamanan untuk membuat antarmuka jaringan dan menempelkannya ke wadah pelatihan. Antarmuka jaringan menyediakan wadah pelatihan dengan koneksi jaringan dalam VPC Anda. Ini memungkinkan pekerjaan pelatihan untuk terhubung ke sumber daya yang ada di VPC Anda.

Berikut ini adalah contoh `VpcConfig` parameter yang Anda sertakan dalam panggilan Anda ke `CreateTrainingJob` operasi:

```
VpcConfig: {
```

```
    "Subnets": [
        "subnet-0123456789abcdef0",
        "subnet-0123456789abcdef1",
        "subnet-0123456789abcdef2"
        ],
    "SecurityGroupIds": [
        "sg-0123456789abcdef0"
        ]
      }
```

Konfigurasikan VPC Pribadi Anda untuk Pelatihan SageMaker

Saat mengonfigurasi VPC pribadi untuk pekerjaan pelatihan SageMaker Anda, gunakan panduan berikut. Untuk informasi tentang menyiapkan VPC, lihat Bekerja dengan VPC dan Subnet di Panduan Pengguna Amazon VPC.

Topik

- Pastikan Subnet Memiliki Alamat IP yang Cukup
- Buat VPC Endpoint Amazon S3
- Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3
- Konfigurasikan Tabel Rute
- Konfigurasikan Grup Keamanan VPC
- Connect ke Sumber Daya di Luar VPC Anda
- Pantau Pekerjaan SageMaker Pelatihan Amazon dengan CloudWatch Log dan Metrik

Pastikan Subnet Memiliki Alamat IP yang Cukup

Instans pelatihan yang tidak menggunakan Elastic Fabric Adapter (EFA) harus memiliki setidaknya 2 alamat IP pribadi. Instans pelatihan yang menggunakan EFA harus memiliki setidaknya 5 alamat IP pribadi. Untuk informasi selengkapnya, lihat Beberapa alamat IP di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Subnet VPC Anda harus memiliki setidaknya dua alamat IP pribadi untuk setiap instance dalam pekerjaan pelatihan. Untuk informasi selengkapnya, lihat Pengukuran VPC dan subnet untuk IPv4 dalam Panduan Pengguna Amazon VPC.

Buat VPC Endpoint Amazon S3

Jika Anda mengonfigurasi VPC agar wadah pelatihan tidak memiliki akses ke internet, wadah tersebut tidak dapat terhubung ke bucket Amazon S3 yang berisi data pelatihan kecuali Anda membuat titik akhir VPC yang memungkinkan akses. Dengan membuat titik akhir VPC, Anda mengizinkan wadah pelatihan Anda mengakses bucket tempat Anda menyimpan data dan artefak model. Kami menyarankan Anda juga membuat kebijakan khusus yang hanya mengizinkan permintaan dari VPC pribadi Anda untuk mengakses bucket S3 Anda. Untuk informasi selengkapnya, lihat Titik Akhir untuk Amazon S3.

Untuk membuat VPC Endpoint S3:

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Buat titik akhir, lalu pilih Buat titik akhir
3. Untuk Nama Layanan, cari com.amazonaws. ***wilayah*** .s3, di mana *wilayah* adalah nama wilayah tempat VPC Anda berada.
4. Pilih jenis Gateway.
5. Untuk VPC, pilih VPC yang ingin Anda gunakan untuk endpoint ini.
6. Untuk Konfigurasi tabel rute, pilih tabel rute yang akan digunakan oleh titik akhir. Layanan VPC secara otomatis menambahkan rute ke setiap tabel rute yang Anda pilih yang mengarahkan lalu lintas S3 ke titik akhir baru.
7. Untuk Kebijakan, pilih Akses Penuh untuk mengizinkan akses penuh ke layanan S3 oleh pengguna atau layanan apa pun dalam VPC. Pilih Custom untuk membatasi akses lebih lanjut. Untuk informasi, lihat Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3.

Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3

Kebijakan endpoint default memungkinkan akses penuh ke S3 untuk setiap pengguna atau layanan di VPC Anda. Untuk lebih membatasi akses ke S3, buat kebijakan titik akhir kustom. Untuk informasi selengkapnya, lihat Menggunakan Kebijakan titik akhir untuk Amazon S3. Anda juga dapat menggunakan kebijakan bucket untuk membatasi akses ke bucket S3 hanya untuk lalu lintas yang berasal dari VPC Amazon Anda. Untuk informasi, lihat Menggunakan Kebijakan Bucket Amazon S3.

Batasi Instalasi Package pada Training Container

Kebijakan endpoint default memungkinkan pengguna untuk menginstal paket dari repositori Amazon Linux dan Amazon Linux 2 pada wadah pelatihan. Jika Anda tidak ingin pengguna menginstal paket

dari repositori itu, buat kebijakan endpoint khusus yang secara eksplisit menolak akses ke repositori Amazon Linux dan Amazon Linux 2. Berikut ini adalah contoh kebijakan yang menolak akses ke repositori ini:

```
{
    "Statement": [
      {
        "Sid": "AmazonLinuxAMIRepositoryAccess",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::packages.*.amazonaws.com/*",
            "arn:aws:s3:::repo.*.amazonaws.com/*"
        ]
      }
    ]
}

{
    "Statement": [
        { "Sid": "AmazonLinux2AMIRepositoryAccess",
          "Principal": "*",
          "Action": [
              "s3:GetObject"
              ],
          "Effect": "Deny",
          "Resource": [
              "arn:aws:s3:::amazonlinux.*.amazonaws.com/*"
              ]
         }
    ]
}
```

Konfigurasikan Tabel Rute

Gunakan pengaturan DNS default untuk tabel rute titik akhir Anda, sehingga URL Amazon S3 standar (misalnya,) teratasi. `http://s3-aws-region.amazonaws.com/MyBucket` Jika Anda tidak menggunakan pengaturan DNS default, pastikan URL yang Anda gunakan untuk menentukan lokasi data dalam pekerjaan pelatihan Anda diselesaikan dengan mengonfigurasi tabel rute titik akhir. Untuk

informasi tentang tabel rute titik akhir VPC, lihat Perutean untuk Titik Akhir Gateway di Panduan Pengguna Amazon VPC.

Konfigurasikan Grup Keamanan VPC

Dalam pelatihan terdistribusi, Anda harus mengizinkan komunikasi antara wadah yang berbeda dalam pekerjaan pelatihan yang sama. Untuk melakukan itu, konfigurasikan aturan untuk grup keamanan Anda yang memungkinkan koneksi masuk antara anggota grup keamanan yang sama. Untuk instans yang mendukung EFA, pastikan bahwa koneksi masuk dan keluar memungkinkan semua lalu lintas dari grup keamanan yang sama. Untuk selengkapnya, lihat Aturan Grup Keamanan di Panduan Pengguna Amazon Virtual Private Cloud.

Connect ke Sumber Daya di Luar VPC Anda

Jika Anda mengonfigurasi VPC Anda sehingga tidak memiliki akses internet, pekerjaan pelatihan yang menggunakan VPC tersebut tidak memiliki akses ke sumber daya di luar VPC Anda. Jika pekerjaan pelatihan Anda membutuhkan akses ke sumber daya di luar VPC Anda, berikan akses dengan salah satu opsi berikut:

- Jika pekerjaan pelatihan Anda memerlukan akses ke AWS layanan yang mendukung titik akhir VPC antarmuka, buat titik akhir untuk terhubung ke layanan tersebut. Untuk daftar layanan yang mendukung titik akhir antarmuka, lihat Titik Akhir VPC di Panduan Pengguna Amazon Virtual Private Cloud. Untuk informasi tentang membuat titik akhir VPC antarmuka, lihat Titik Akhir VPC Antarmuka (AWS PrivateLink) di Panduan Pengguna Amazon Virtual Private Cloud.
- Jika tugas pelatihan Anda memerlukan akses ke AWS layanan yang tidak mendukung titik akhir VPC antarmuka atau sumber daya di luarAWS, buat gateway NAT dan konfigurasikan grup keamanan Anda untuk mengizinkan koneksi keluar. Untuk informasi tentang pengaturan gateway NAT untuk VPC Anda, lihat Skenario 2: VPC dengan Subnet Publik dan Privat (NAT) di Panduan Pengguna Amazon Virtual Private Cloud.

Pantau Pekerjaan SageMaker Pelatihan Amazon dengan CloudWatch Log dan Metrik

Amazon SageMaker menyediakan CloudWatch log dan metrik Amazon untuk memantau pekerjaan pelatihan. CloudWatch menyediakan CPU, GPU, memori, memori GPU, dan metrik disk, dan pencatatan peristiwa. Untuk informasi selengkapnya tentang memantau pekerjaan SageMaker pelatihan Amazon, lihat Pantau Amazon SageMaker dengan Amazon CloudWatch danSageMaker Lowongan Kerja dan Metrik Titik Akhir.

Berikan Akses Endpoint yang SageMaker Dihosting ke Sumber Daya di VPC Amazon Anda

Konfigurasikan Model untuk Akses VPC Amazon

Untuk menentukan subnet dan grup keamanan di VPC pribadi Anda, gunakan `VpcConfig` parameter permintaan API, atau berikan informasi ini saat Anda membuat model di SageMaker konsol. `CreateModel` SageMaker menggunakan informasi ini untuk membuat antarmuka jaringan dan melampirkannya ke wadah model Anda. Antarmuka jaringan menyediakan wadah model Anda dengan koneksi jaringan dalam VPC Anda yang tidak terhubung ke internet. Mereka juga memungkinkan model Anda terhubung ke sumber daya di VPC pribadi Anda.

> **Note**
>
> Anda harus membuat setidaknya dua subnet di zona ketersediaan yang berbeda di VPC pribadi Anda, bahkan jika Anda hanya memiliki satu instance hosting.

Berikut ini adalah contoh `VpcConfig` parameter yang Anda sertakan dalam panggilan Anda ke`CreateModel`:

```
VpcConfig: {
      "Subnets": [
          "subnet-0123456789abcdef0",
          "subnet-0123456789abcdef1",
          "subnet-0123456789abcdef2"
          ],
      "SecurityGroupIds": [
          "sg-0123456789abcdef0"
          ]
       }
```

Konfigurasikan VPC Pribadi Anda untuk Hosting SageMaker

Saat mengonfigurasi VPC pribadi untuk model SageMaker Anda, gunakan panduan berikut. Untuk informasi tentang menyiapkan VPC, lihat Bekerja dengan VPC dan Subnet di Panduan Pengguna Amazon VPC.

Topik

- Pastikan Subnet Memiliki Alamat IP yang Cukup
- Buat VPC Endpoint Amazon S3
- Gunakan Kebijakan Titik Akhir Kustom untuk Membatasi Akses ke Amazon S3
- Tambahkan Izin untuk Akses Titik Akhir untuk Kontainer yang Berjalan di VPC ke Kebijakan IAM Kustom
- Konfigurasikan Tabel Rute
- Connect ke Sumber Daya di Luar VPC Anda

Pastikan Subnet Memiliki Alamat IP yang Cukup

Instans pelatihan yang tidak menggunakan Elastic Fabric Adapter (EFA) harus memiliki setidaknya 2 alamat IP pribadi. Instans pelatihan yang menggunakan EFA harus memiliki setidaknya 5 alamat IP pribadi. Untuk informasi selengkapnya, lihat Beberapa alamat IP di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Buat VPC Endpoint Amazon S3

Jika Anda mengonfigurasi VPC Anda sehingga wadah model tidak memiliki akses ke internet, mereka tidak dapat terhubung ke bucket Amazon S3 yang berisi data Anda kecuali Anda membuat titik akhir VPC yang memungkinkan akses. Dengan membuat titik akhir VPC, Anda mengizinkan wadah model Anda mengakses bucket tempat Anda menyimpan data dan artefak model. Kami menyarankan Anda juga membuat kebijakan khusus yang hanya mengizinkan permintaan dari VPC pribadi Anda untuk mengakses bucket S3 Anda. Untuk informasi selengkapnya, lihat Titik Akhir untuk Amazon S3.

Untuk membuat VPC Endpoint Amazon S3:

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Buat titik akhir, lalu pilih Buat titik akhir
3. Untuk Nama Layanan, pilih com.amazonaws. ***region*** .s3, di mana *wilayah* adalah nama AWS Wilayah tempat VPC Anda berada.
4. Untuk VPC, pilih VPC yang ingin Anda gunakan untuk titik akhir ini.
5. Untuk Konfigurasi tabel rute, pilih tabel rute untuk titik akhir yang akan digunakan. Layanan VPC secara otomatis menambahkan rute ke setiap tabel rute yang Anda pilih yang mengarahkan lalu lintas Amazon S3 ke titik akhir yang baru.
6. Untuk Kebijakan, pilih Akses Penuh untuk mengizinkan akses penuh ke layanan Amazon S3 oleh pengguna atau layanan apa pun dalam VPC. Untuk membatasi akses lebih lanjut, pilih

Kustom. Untuk informasi selengkapnya, lihat Gunakan Kebijakan Titik Akhir Kustom untuk Membatasi Akses ke Amazon S3.

Gunakan Kebijakan Titik Akhir Kustom untuk Membatasi Akses ke Amazon S3

Kebijakan endpoint default memungkinkan akses penuh ke Amazon Simple Storage Service (Amazon S3) untuk setiap pengguna atau layanan di VPC Anda. Untuk lebih membatasi akses ke Amazon S3, buat kebijakan titik akhir kustom. Untuk informasi selengkapnya, lihat Menggunakan Kebijakan titik akhir untuk Amazon S3.

Anda juga dapat menggunakan kebijakan bucket untuk membatasi akses ke bucket S3 hanya untuk lalu lintas yang berasal dari VPC Amazon Anda. Untuk informasi, lihat Menggunakan Kebijakan Bucket Amazon S3.

Batasi Instalasi Package pada Container Model dengan Kebijakan Endpoint Kustom

Kebijakan endpoint default memungkinkan pengguna untuk menginstal paket dari repositori Amazon Linux dan Amazon Linux 2 pada wadah model. Jika Anda tidak ingin pengguna menginstal paket dari repositori tersebut, buat kebijakan endpoint khusus yang secara eksplisit menolak akses ke repositori Amazon Linux dan Amazon Linux 2. Berikut ini adalah contoh kebijakan yang menolak akses ke repositori ini:

```
{
    "Statement": [
      {
        "Sid": "AmazonLinuxAMIRepositoryAccess",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::packages.*.amazonaws.com/*",
            "arn:aws:s3:::repo.*.amazonaws.com/*"
        ]
      }
    ]
}

{
    "Statement": [
```

```
        { "Sid": "AmazonLinux2AMIRepositoryAccess",
          "Principal": "*",
          "Action": [
              "s3:GetObject"
              ],
          "Effect": "Deny",
          "Resource": [
              "arn:aws:s3:::amazonlinux.*.amazonaws.com/*"
              ]
         }
    ]
}
```

Tambahkan Izin untuk Akses Titik Akhir untuk Kontainer yang Berjalan di VPC ke Kebijakan IAM Kustom

Kebijakan `SageMakerFullAccess` terkelola mencakup izin yang Anda perlukan untuk menggunakan model yang dikonfigurasi untuk akses VPC Amazon dengan titik akhir. Izin ini memungkinkan SageMaker untuk membuat elastic network interface dan melampirkannya ke wadah model yang berjalan di VPC. Jika Anda menggunakan kebijakan IAM Anda sendiri, Anda harus menambahkan izin berikut ke kebijakan tersebut untuk menggunakan model yang dikonfigurasi untuk akses VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:CreateNetworkInterface"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Untuk informasi selengkapnya tentang kebijakan `SageMakerFullAccess` terkelola, lihatAWSkebijakan terkelola: AmazonSageMakerFullAccess.

Konfigurasikan Tabel Rute

Gunakan pengaturan DNS default untuk tabel rute titik akhir Anda, sehingga URL Amazon S3 standar (misalnya,) teratasi. `http://s3-aws-region.amazonaws.com/MyBucket` Jika Anda tidak menggunakan pengaturan DNS default, pastikan URL yang Anda gunakan untuk menentukan lokasi data dalam model Anda diselesaikan dengan mengonfigurasi tabel rute titik akhir. Untuk informasi tentang tabel rute titik akhir VPC, lihat Perutean untuk Titik Akhir Gateway di Panduan Pengguna Amazon VPC.

Connect ke Sumber Daya di Luar VPC Anda

Jika Anda mengonfigurasi VPC Anda sehingga tidak memiliki akses internet, model yang menggunakan VPC tersebut tidak memiliki akses ke sumber daya di luar VPC Anda. Jika model Anda membutuhkan akses ke sumber daya di luar VPC, berikan akses dengan salah satu opsi berikut:

- Jika model Anda memerlukan akses ke AWS layanan yang mendukung titik akhir VPC antarmuka, buat titik akhir untuk terhubung ke layanan tersebut. Untuk daftar layanan yang mendukung titik akhir antarmuka, lihat Titik Akhir VPC di Panduan Pengguna Amazon VPC. Untuk informasi tentang membuat titik akhir VPC antarmuka, lihat Titik Akhir VPC Antarmuka () di AWS PrivateLink Panduan Pengguna VPC Amazon.
- Jika model Anda memerlukan akses ke AWS layanan yang tidak mendukung titik akhir VPC antarmuka atau sumber daya di luarAWS, buat gateway NAT dan konfigurasikan grup keamanan Anda untuk mengizinkan koneksi keluar. Untuk informasi tentang pengaturan gateway NAT untuk VPC Anda, lihat Skenario 2: VPC dengan Subnet Publik dan Privat (NAT) di Panduan Pengguna Amazon Virtual Private Cloud.

Berikan Akses Pekerjaan Transformasi Batch ke Sumber Daya di VPC Amazon Anda

Untuk mengontrol akses ke data dan pekerjaan transformasi batch, kami sarankan Anda membuat VPC Amazon pribadi dan mengonfigurasinya sehingga pekerjaan Anda tidak dapat diakses melalui internet publik. Anda menentukan konfigurasi VPC pribadi saat membuat model dengan menentukan subnet dan grup keamanan. Anda kemudian menentukan model yang sama ketika Anda membuat pekerjaan transformasi batch. Saat Anda menentukan subnet dan grup keamanan, SageMaker

buat antarmuka jaringan elastis yang terkait dengan grup keamanan Anda di salah satu subnet. Antarmuka jaringan memungkinkan wadah model Anda terhubung ke sumber daya di VPC Anda. Untuk informasi tentang antarmuka jaringan, lihat Antarmuka Jaringan Elastis di Panduan Pengguna Amazon VPC.

Dokumen ini menjelaskan cara menambahkan konfigurasi VPC Amazon untuk pekerjaan transformasi batch.

Konfigurasikan Job Transform Batch untuk Akses VPC Amazon

Untuk menentukan subnet dan grup keamanan di VPC pribadi Anda, gunakan `VpcConfig` parameter permintaan API, atau berikan informasi ini saat Anda membuat model di SageMaker konsol. `CreateModel` Kemudian tentukan model yang sama di parameter `ModelName` permintaan `CreateTransformJob`API, atau di bidang Nama model saat Anda membuat pekerjaan transformasi di SageMaker konsol. SageMaker menggunakan informasi ini untuk membuat antarmuka jaringan dan melampirkannya ke wadah model Anda. Antarmuka jaringan menyediakan wadah model Anda dengan koneksi jaringan dalam VPC Anda yang tidak terhubung ke internet. Mereka juga memungkinkan pekerjaan transformasi Anda untuk terhubung ke sumber daya di VPC pribadi Anda.

Berikut ini adalah contoh `VpcConfig` parameter yang Anda sertakan dalam panggilan Anda ke`CreateModel`:

```
VpcConfig: {
      "Subnets": [
          "subnet-0123456789abcdef0",
          "subnet-0123456789abcdef1",
          "subnet-0123456789abcdef2"
          ],
      "SecurityGroupIds": [
          "sg-0123456789abcdef0"
          ]
        }
```

Jika Anda membuat model menggunakan operasi `CreateModel` API, peran eksekusi IAM yang Anda gunakan untuk membuat model harus menyertakan izin yang dijelaskanCreateModel API: Izin Peran Eksekusi, termasuk izin berikut yang diperlukan untuk VPC pribadi.

Saat membuat model di konsol, jika Anda memilih Buat peran baru di bagian Pengaturan Model, AmazonSageMakerFullAccess kebijakan yang digunakan untuk membuat peran sudah berisi izin ini. Jika Anda memilih Masukkan peran IAM khusus ARN atau Gunakan peran yang ada, ARN peran yang Anda tentukan harus memiliki kebijakan eksekusi yang dilampirkan dengan izin berikut.

```
{
            "Effect": "Allow",
            "Action": [
            "ec2:CreateNetworkInterface",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeVpcs",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
```

Konfigurasikan VPC Pribadi Anda untuk Transformasi Batch SageMaker

Saat mengonfigurasi VPC pribadi untuk pekerjaan transformasi batch SageMaker Anda, gunakan panduan berikut. Untuk informasi tentang menyiapkan VPC, lihat Bekerja dengan VPC dan Subnet di Panduan Pengguna Amazon VPC.

Topik

- Pastikan Subnet Memiliki Alamat IP yang Cukup
- Buat VPC Endpoint Amazon S3
- Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3
- Konfigurasikan Tabel Rute
- Konfigurasikan Grup Keamanan VPC
- Connect ke Sumber Daya di Luar VPC Anda

Pastikan Subnet Memiliki Alamat IP yang Cukup

Subnet VPC Anda harus memiliki setidaknya dua alamat IP pribadi untuk setiap instance dalam pekerjaan transformasi. Untuk informasi selengkapnya, lihat Pengukuran VPC dan subnet untuk IPv4 dalam Panduan Pengguna Amazon VPC.

Buat VPC Endpoint Amazon S3

Jika Anda mengonfigurasi VPC Anda sehingga wadah model tidak memiliki akses ke internet, mereka tidak dapat terhubung ke bucket Amazon S3 yang berisi data Anda kecuali Anda membuat titik akhir VPC yang memungkinkan akses. Dengan membuat titik akhir VPC, Anda mengizinkan wadah model

Anda mengakses bucket tempat Anda menyimpan data dan artefak model. Kami menyarankan Anda juga membuat kebijakan khusus yang hanya mengizinkan permintaan dari VPC pribadi Anda untuk mengakses bucket S3 Anda. Untuk informasi selengkapnya, lihat Titik Akhir untuk Amazon S3.

Untuk membuat VPC Endpoint S3:

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Buat titik akhir, lalu pilih Buat titik akhir
3. Untuk Nama Layanan, pilih com.amazonaws. ***wilayah*** .s3, di mana *wilayah* adalah nama wilayah tempat VPC Anda berada.
4. Untuk VPC, pilih VPC yang ingin Anda gunakan untuk endpoint ini.
5. Untuk Konfigurasi tabel rute, pilih tabel rute yang akan digunakan oleh titik akhir. Layanan VPC secara otomatis menambahkan rute ke setiap tabel rute yang Anda pilih yang mengarahkan lalu lintas S3 ke titik akhir baru.
6. Untuk Kebijakan, pilih Akses Penuh untuk mengizinkan akses penuh ke layanan S3 oleh pengguna atau layanan apa pun dalam VPC. Pilih Custom untuk membatasi akses lebih lanjut. Untuk informasi, lihat Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3.

Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3

Kebijakan endpoint default memungkinkan akses penuh ke S3 untuk setiap pengguna atau layanan di VPC Anda. Untuk lebih membatasi akses ke S3, buat kebijakan titik akhir kustom. Untuk informasi selengkapnya, lihat Menggunakan Kebijakan titik akhir untuk Amazon S3. Anda juga dapat menggunakan kebijakan bucket untuk membatasi akses ke bucket S3 hanya untuk lalu lintas yang berasal dari VPC Amazon Anda. Untuk informasi, lihat Menggunakan Kebijakan Bucket Amazon S3.

Batasi Instalasi Package pada Model Container

Kebijakan endpoint default memungkinkan pengguna untuk menginstal paket dari repositori Amazon Linux dan Amazon Linux 2 pada wadah pelatihan. Jika Anda tidak ingin pengguna menginstal paket dari repositori itu, buat kebijakan endpoint khusus yang secara eksplisit menolak akses ke repositori Amazon Linux dan Amazon Linux 2. Berikut ini adalah contoh kebijakan yang menolak akses ke repositori ini:

```
{
    "Statement": [
      {
```

```
        "Sid": "AmazonLinuxAMIRepositoryAccess",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::packages.*.amazonaws.com/*",
            "arn:aws:s3:::repo.*.amazonaws.com/*"
        ]
      }
    ]
}

{
    "Statement": [
        { "Sid": "AmazonLinux2AMIRepositoryAccess",
          "Principal": "*",
          "Action": [
              "s3:GetObject"
              ],
          "Effect": "Deny",
          "Resource": [
              "arn:aws:s3:::amazonlinux.*.amazonaws.com/*"
              ]
         }
    ]
}
```

Konfigurasikan Tabel Rute

Gunakan pengaturan DNS default untuk tabel rute titik akhir Anda, sehingga URL Amazon S3 standar (misalnya,) teratasi. `http://s3-aws-region.amazonaws.com/MyBucket` Jika Anda tidak menggunakan pengaturan DNS default, pastikan URL yang Anda gunakan untuk menentukan lokasi data dalam pekerjaan transformasi batch Anda diselesaikan dengan mengonfigurasi tabel rute titik akhir. Untuk informasi tentang tabel rute titik akhir VPC, lihat Perutean untuk Titik Akhir Gateway di Panduan Pengguna Amazon VPC.

Konfigurasikan Grup Keamanan VPC

Dalam transformasi batch terdistribusi, Anda harus mengizinkan komunikasi antara kontainer yang berbeda dalam pekerjaan transformasi batch yang sama. Untuk melakukan itu, konfigurasikan aturan

untuk grup keamanan Anda yang memungkinkan koneksi masuk dan keluar antara anggota grup keamanan yang sama. Anggota dari grup keamanan yang sama harus dapat berkomunikasi satu sama lain di semua port. Untuk informasi selengkapnya, lihat Aturan Grup Keamanan.

Connect ke Sumber Daya di Luar VPC Anda

Jika Anda mengonfigurasi VPC Anda sehingga tidak memiliki akses internet, pekerjaan transformasi batch yang menggunakan VPC tersebut tidak memiliki akses ke sumber daya di luar VPC Anda. Jika pekerjaan transformasi batch Anda membutuhkan akses ke sumber daya di luar VPC Anda, berikan akses dengan salah satu opsi berikut:

- Jika pekerjaan transformasi batch Anda memerlukan akses ke AWS layanan yang mendukung titik akhir VPC antarmuka, buat titik akhir untuk terhubung ke layanan tersebut. Untuk daftar layanan yang mendukung titik akhir antarmuka, lihat Titik Akhir VPC di Panduan Pengguna Amazon VPC. Untuk informasi tentang membuat titik akhir VPC antarmuka, lihat Titik Akhir VPC Antarmuka () di AWS PrivateLink Panduan Pengguna VPC Amazon.
- Jika pekerjaan transformasi batch Anda memerlukan akses ke AWS layanan yang tidak mendukung titik akhir VPC antarmuka atau sumber daya di luarAWS, buat gateway NAT dan konfigurasikan grup keamanan Anda untuk mengizinkan koneksi keluar. Untuk informasi tentang pengaturan gateway NAT untuk VPC Anda, lihat Skenario 2: VPC dengan Subnet Publik dan Privat (NAT) di Panduan Pengguna Amazon Virtual Private Cloud.

Berikan Amazon SageMaker Clarify Lowongan Akses ke Sumber Daya di Amazon VPC Anda

Untuk mengontrol akses ke data Anda dan SageMaker Clarify lowongan, kami sarankan Anda membuat VPC Amazon pribadi dan mengonfigurasinya sehingga pekerjaan Anda tidak dapat diakses melalui internet publik. Untuk informasi tentang membuat dan mengonfigurasi VPC Amazon untuk memproses pekerjaan, lihat SageMaker Memberikan Akses Pekerjaan Pemrosesan ke Sumber Daya di VPC Amazon Anda.

Dokumen ini menjelaskan cara menambahkan konfigurasi VPC Amazon tambahan yang memenuhi persyaratan untuk SageMaker pekerjaan Clarify.

Topik

- Mengkonfigurasi SageMaker Clarify Job untuk Amazon VPC Access
- Konfigurasikan VPC Amazon Pribadi Anda untuk SageMaker pekerjaan Clarify

Mengkonfigurasi SageMaker Clarify Job untuk Amazon VPC Access

Anda perlu menentukan subnet dan grup keamanan saat mengonfigurasi VPC Amazon pribadi Anda untuk pekerjaan SageMaker Clarify dan mengaktifkan pekerjaan mendapatkan kesimpulan dari SageMaker model saat menghitung metrik bias pasca-pelatihan dan kontribusi fitur yang membantu menjelaskan prediksi model.

Topik

- SageMaker Klarifikasi Pekerjaan Subnet VPC Amazon dan Grup Keamanan
- Konfigurasikan Model Amazon VPC untuk Inferensi

SageMaker Klarifikasi Pekerjaan Subnet VPC Amazon dan Grup Keamanan

Subnet dan grup keamanan di VPC Amazon pribadi Anda dapat ditetapkan ke pekerjaan SageMaker Clarify dengan berbagai cara, tergantung pada cara Anda membuat pekerjaan.

- SageMaker console: Berikan informasi ini saat Anda membuat pekerjaan di SageMakerDasbor. Dari menu Processing, pilih Processing jobs, lalu pilih Create processing job. Pilih opsi VPC di panel Jaringan dan berikan subnet dan grup keamanan menggunakan daftar drop-down. Pastikan opsi isolasi jaringan yang disediakan di panel ini dimatikan.
- SageMaker API: Gunakan parameter `NetworkConfig.VpcConfig` permintaan `CreateProcessingJob`API, seperti yang ditunjukkan pada contoh berikut:

```
"NetworkConfig": {
    "VpcConfig": {
        "Subnets": [
            "subnet-0123456789abcdef0",
            "subnet-0123456789abcdef1",
            "subnet-0123456789abcdef2"
        ],
        "SecurityGroupIds": [
            "sg-0123456789abcdef0"
        ]
    }
}
```

- SageMaker Python SDK: Gunakan `NetworkConfig` parameter `SageMakerClarifyProcessor`API atau `Processor`API, seperti yang ditunjukkan pada contoh berikut:

```
from sagemaker.network import NetworkConfig
network_config = NetworkConfig(
    subnets=[
        "subnet-0123456789abcdef0",
        "subnet-0123456789abcdef1",
        "subnet-0123456789abcdef2",
    ],
    security_group_ids=[
        "sg-0123456789abcdef0",
    ],
)
```

SageMaker menggunakan informasi untuk membuat antarmuka jaringan dan melampirkannya ke pekerjaan SageMaker Clarify. Antarmuka jaringan menyediakan pekerjaan SageMaker Clarify dengan koneksi jaringan dalam VPC Amazon Anda yang tidak terhubung ke internet publik. Mereka juga memungkinkan pekerjaan SageMaker Clarify untuk terhubung ke sumber daya di VPC Amazon pribadi Anda.

> **Note**
>
> Opsi isolasi jaringan dari pekerjaan SageMaker Clarify harus dimatikan (secara default opsi dimatikan) sehingga pekerjaan SageMaker Clarify dapat berkomunikasi dengan titik akhir bayangan.

Konfigurasikan Model Amazon VPC untuk Inferensi

Untuk menghitung metrik dan penjelasan bias pasca-pelatihan, pekerjaan SageMaker Clarify perlu mendapatkan kesimpulan dari SageMaker model yang ditentukan oleh `model_name` parameter konfigurasi analisis untuk pekerjaan pemrosesan Clarify. SageMaker Atau, jika Anda menggunakan `SageMakerClarifyProcessor` API di SageMaker Python SDK, pekerjaan harus mendapatkan yang `model_name` ditentukan oleh kelas. ModelConfig Untuk mencapai hal ini, pekerjaan SageMaker Clarify membuat titik akhir singkat dengan model, yang dikenal sebagai titik akhir bayangan, dan kemudian menerapkan konfigurasi Amazon VPC model ke titik akhir bayangan.

Untuk menentukan subnet dan grup keamanan di VPC Amazon pribadi Anda ke SageMaker model, gunakan `VpcConfig` parameter permintaan API atau berikan informasi ini saat Anda membuat

model menggunakan dasbor SageMaker di konsol. `CreateModel` Berikut ini adalah contoh `VpcConfig` parameter yang Anda sertakan dalam panggilan Anda ke`CreateModel`:

```
"VpcConfig": {
    "Subnets": [
        "subnet-0123456789abcdef0",
        "subnet-0123456789abcdef1",
        "subnet-0123456789abcdef2"
    ],
    "SecurityGroupIds": [
        "sg-0123456789abcdef0"
    ]
}
```

Anda dapat menentukan jumlah instance titik akhir bayangan yang akan diluncurkan dengan `initial_instance_count` parameter konfigurasi analisis untuk tugas pemrosesan SageMaker Clarify. Atau, jika Anda menggunakan `SageMakerClarifyProcessor` API di SageMaker Python SDK, pekerjaan harus mendapatkan yang `instance_count` ditentukan oleh kelas. ModelConfig

> **Note**
>
> Bahkan jika Anda hanya meminta satu instance saat membuat titik akhir bayangan, Anda memerlukan setidaknya dua subnet dalam model di zona ketersediaan yang berbeda. ModelConfig Jika tidak, pembuatan endpoint bayangan gagal dengan galat berikut: ClientError: Kesalahan hosting titik akhir sagemaker-clarify-endpoint -XXX: Gagal. Alasan: Tidak dapat menemukan setidaknya 2 zona ketersediaan dengan tipe instans yang diminta YYY yang tumpang tindih dengan subnet. SageMaker

Jika model Anda memerlukan file model di Amazon S3, maka model Amazon VPC harus memiliki titik akhir VPC Amazon S3. Untuk informasi selengkapnya tentang membuat dan mengonfigurasi VPC Amazon SageMaker untuk model, lihat. Berikan Akses Endpoint yang SageMaker Dihosting ke Sumber Daya di VPC Amazon Anda

Konfigurasikan VPC Amazon Pribadi Anda untuk SageMaker pekerjaan Clarify

Secara umum, Anda dapat mengikuti langkah-langkah di Konfigurasikan VPC Pribadi Anda untuk SageMaker Pemrosesan untuk mengonfigurasi VPC Amazon pribadi Anda untuk pekerjaan Clarify. SageMaker Berikut adalah beberapa sorotan dan persyaratan khusus untuk pekerjaan SageMaker Clarify.

Topik

- Hubungkan ke Sumber Daya di Luar VPC Amazon Anda
- Konfigurasikan Grup Keamanan Amazon VPC

Hubungkan ke Sumber Daya di Luar VPC Amazon Anda

Jika Anda mengonfigurasi VPC Amazon Anda sehingga tidak memiliki akses internet publik, maka beberapa pengaturan tambahan diperlukan untuk memberikan SageMaker Clarify lowongan akses ke sumber daya dan layanan di luar VPC Amazon Anda. Misalnya, titik akhir VPC Amazon S3 diperlukan karena tugas SageMaker Clarify perlu memuat kumpulan data dari bucket S3 serta menyimpan hasil analisis ke bucket S3. Untuk informasi selengkapnya, lihat Membuat Titik Akhir VPC Amazon S3 untuk panduan pembuatan. Selain itu, jika pekerjaan SageMaker Clarify perlu mendapatkan kesimpulan dari titik akhir bayangan, maka perlu memanggil beberapa layanan lagiAWS.

- Buat titik akhir VPC layanan Amazon SageMaker API: Pekerjaan SageMaker Clarify perlu memanggil layanan Amazon SageMaker API untuk memanipulasi titik akhir bayangan, atau untuk menjelaskan model validasi VPC SageMaker Amazon. Anda dapat mengikuti panduan yang disediakan di blog Mengamankan semua panggilan SageMaker API Amazon dengan AWS PrivateLink blog untuk membuat titik akhir VPC Amazon SageMaker API yang memungkinkan tugas SageMaker Clarify melakukan panggilan layanan. Perhatikan bahwa nama layanan layanan Amazon SageMaker API adalah`com.amazonaws.`*`region`*`.sagemaker.api`, di mana *`wilayah`* adalah nama Wilayah tempat VPC Amazon Anda berada.
- Buat Titik Akhir VPC Amazon SageMaker Runtime: SageMaker Pekerjaan Clarify perlu memanggil layanan runtime SageMaker Amazon, yang merutekan pemanggilan ke titik akhir bayangan. Langkah-langkah penyiapannya mirip dengan yang ada pada layanan Amazon SageMaker API. Perhatikan bahwa nama layanan layanan Amazon SageMaker Runtime adalah`com.amazonaws.`*`region`*`.sagemaker.runtime`, di mana *`wilayah`* adalah nama Wilayah tempat VPC Amazon Anda berada.

Konfigurasikan Grup Keamanan Amazon VPC

SageMaker Klarifikasi pekerjaan mendukung pemrosesan terdistribusi ketika dua atau lebih contoh pemrosesan ditentukan dalam salah satu cara berikut:

- SageMaker console: Jumlah Instans ditentukan di bagian konfigurasi Resource dari panel pengaturan Job pada halaman Create processing job.

- SageMaker API: `InstanceCount` Ini ditentukan saat Anda membuat pekerjaan dengan `CreateProcessingJob`API.
- SageMaker Python SDK: `instance_count` Ini ditentukan saat menggunakan SageMakerClarifyProcessorAPI atau API Processor.

Dalam pemrosesan terdistribusi, Anda harus mengizinkan komunikasi antara instance yang berbeda dalam pekerjaan pemrosesan yang sama. Untuk melakukan itu, konfigurasikan aturan untuk grup keamanan Anda yang memungkinkan koneksi masuk antara anggota grup keamanan yang sama. Untuk selengkapnya, lihat Aturan grup keamanan.

Berikan Akses Pekerjaan SageMaker Kompilasi ke Sumber Daya di VPC Amazon Anda

> **Note**
>
> Untuk pekerjaan kompilasi, Anda hanya dapat mengonfigurasi subnet dengan VPC penyewaan default di mana pekerjaan Anda berjalan pada perangkat keras bersama. Untuk informasi selengkapnya tentang atribut tenancy untuk VPC, lihat Instans Khusus.

Konfigurasikan Pekerjaan Kompilasi untuk Akses VPC Amazon

Untuk menentukan subnet dan grup keamanan di VPC pribadi Anda, gunakan `VpcConfig` parameter permintaan API, atau berikan informasi ini saat Anda membuat pekerjaan kompilasi di SageMaker konsol. `CreateCompilationJob` SageMaker Neo menggunakan informasi ini untuk membuat antarmuka jaringan dan melampirkannya ke pekerjaan kompilasi Anda. Antarmuka jaringan menyediakan pekerjaan kompilasi dengan koneksi jaringan dalam VPC Anda yang tidak terhubung ke internet. Mereka juga memungkinkan pekerjaan kompilasi Anda untuk terhubung ke sumber daya di VPC pribadi Anda. Berikut ini adalah contoh `VpcConfig` parameter yang Anda sertakan dalam panggilan Anda ke`CreateCompilationJob`:

```
VpcConfig: {"Subnets": [
          "subnet-0123456789abcdef0",
          "subnet-0123456789abcdef1",
          "subnet-0123456789abcdef2"
          ],
      "SecurityGroupIds": [
          "sg-0123456789abcdef0"
```

```
        ]
    }
```

Konfigurasikan VPC Pribadi Anda untuk Kompilasi SageMaker

Saat mengonfigurasi VPC pribadi untuk pekerjaan kompilasi SageMaker Anda, gunakan panduan berikut. Untuk informasi tentang menyiapkan VPC, lihat Bekerja dengan VPC dan Subnet di Panduan Pengguna Amazon VPC.

Topik

- Pastikan Subnet Memiliki Alamat IP yang Cukup
- Buat VPC Endpoint Amazon S3
- Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3
- Konfigurasikan Tabel Rute
- Konfigurasikan Grup Keamanan VPC

Pastikan Subnet Memiliki Alamat IP yang Cukup

Subnet VPC Anda harus memiliki setidaknya dua alamat IP pribadi untuk setiap instance dalam pekerjaan kompilasi. Untuk informasi selengkapnya, lihat Pengukuran VPC dan subnet untuk IPv4 dalam Panduan Pengguna Amazon VPC.

Buat VPC Endpoint Amazon S3

Jika Anda mengonfigurasi VPC untuk memblokir akses ke internet, SageMaker Neo tidak dapat terhubung ke bucket Amazon S3 yang berisi model Anda kecuali Anda membuat titik akhir VPC yang memungkinkan akses. Dengan membuat titik akhir VPC, Anda mengizinkan pekerjaan kompilasi SageMaker Neo Anda untuk mengakses bucket tempat Anda menyimpan data dan artefak model. Kami menyarankan Anda juga membuat kebijakan khusus yang hanya mengizinkan permintaan dari VPC pribadi Anda untuk mengakses bucket S3 Anda. Untuk informasi selengkapnya, lihat Titik Akhir untuk Amazon S3.

Untuk membuat VPC Endpoint S3:

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Buat titik akhir, lalu pilih Buat titik akhir
3. Untuk Nama Layanan, cari com.amazonaws. ***wilayah*** .s3, di mana *wilayah* adalah nama wilayah tempat VPC Anda berada.

4. Pilih jenis Gateway.
5. Untuk VPC, pilih VPC yang ingin Anda gunakan untuk endpoint ini.
6. Untuk Konfigurasi tabel rute, pilih tabel rute yang akan digunakan oleh titik akhir. Layanan VPC secara otomatis menambahkan rute ke setiap tabel rute yang Anda pilih yang mengarahkan lalu lintas S3 ke titik akhir baru.
7. Untuk Kebijakan, pilih Akses Penuh untuk mengizinkan akses penuh ke layanan S3 oleh pengguna atau layanan apa pun dalam VPC. Pilih Custom untuk membatasi akses lebih lanjut. Untuk informasi, lihat Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3.

Gunakan Kebijakan Endpoint Kustom untuk Membatasi Akses ke S3

Kebijakan endpoint default memungkinkan akses penuh ke S3 untuk setiap pengguna atau layanan di VPC Anda. Untuk lebih membatasi akses ke S3, buat kebijakan titik akhir kustom. Untuk informasi selengkapnya, lihat Menggunakan Kebijakan titik akhir untuk Amazon S3. Anda juga dapat menggunakan kebijakan bucket untuk membatasi akses ke bucket S3 hanya untuk lalu lintas yang berasal dari VPC Amazon Anda. Untuk informasi, lihat Menggunakan Kebijakan Bucket Amazon S3. Berikut ini adalah contoh kebijakan yang disesuaikan:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::your-sample-bucket",
                "arn:aws:s3:::your-sample-bucket/*"
            ],
            "Condition": {
                "StringNotEquals": {
                    "aws:SourceVpce": [
                        "vpce-01234567890123456"
                    ]
                }
            }
        }
    ]
```

```
}
```

Menambahkan Izin untuk Menjalankan Pekerjaan Kompilasi di VPC Amazon ke Kebijakan IAM Kustom

Kebijakan `SageMakerFullAccess` terkelola mencakup izin yang Anda perlukan untuk menggunakan model yang dikonfigurasi untuk akses VPC Amazon dengan titik akhir. Izin ini memungkinkan SageMaker Neo untuk membuat elastic network interface dan melampirkannya ke pekerjaan kompilasi yang berjalan di Amazon VPC. Jika Anda menggunakan kebijakan IAM Anda sendiri, Anda harus menambahkan izin berikut ke kebijakan tersebut untuk menggunakan model yang dikonfigurasi untuk akses VPC Amazon.

```
{"Version": "2012-10-17",
    "Statement": [
        {"Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:CreateNetworkInterface",
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "*"
        }
    ]
}
```

Untuk informasi selengkapnya tentang kebijakan `SageMakerFullAccess` terkelola, lihatAWSkebijakan terkelola: AmazonSageMakerFullAccess.

Konfigurasikan Tabel Rute

Gunakan pengaturan DNS default untuk tabel rute titik akhir Anda, sehingga URL Amazon S3 standar (misalnya,) teratasi. `http://s3-aws-region.amazonaws.com/MyBucket` Jika Anda tidak menggunakan pengaturan DNS default, pastikan URL yang Anda gunakan untuk menentukan lokasi

data dalam pekerjaan kompilasi Anda diselesaikan dengan mengonfigurasi tabel rute titik akhir. Untuk informasi tentang tabel rute titik akhir VPC, lihat Perutean untuk Titik Akhir Gateway di Panduan Pengguna Amazon VPC.

Konfigurasikan Grup Keamanan VPC

Di grup keamanan untuk pekerjaan kompilasi, Anda harus mengizinkan komunikasi keluar ke titik akhir Amazon S3 Amazon VPC dan rentang CIDR subnet yang digunakan untuk pekerjaan kompilasi. Untuk selengkapnya, lihat Aturan Grup Keamanan dan Kontrol akses ke layanan dengan titik akhir Amazon VPC.

Berikan Akses Pekerjaan Inferensi Rekomendasi ke Sumber Daya di VPC Amazon Anda

> **Note**
>
> Inference Recommender mengharuskan Anda untuk mendaftarkan model Anda dengan Model Registry. Perhatikan bahwa Model Registry tidak mengizinkan artefak model atau gambar Amazon ECR Anda dibatasi oleh VPC.
> Inference Recommender juga memiliki persyaratan bahwa objek Amazon S3 muatan sampel Anda tidak dibatasi oleh VPC. Untuk lowongan rekomendasi inferensi, Anda tidak dapat membuat kebijakan khusus yang hanya mengizinkan permintaan dari VPC pribadi untuk mengakses bucket Amazon S3 Anda.

Untuk menentukan subnet dan grup keamanan di VPC pribadi Anda, gunakan `RecommendationJobVpcConfig` parameter permintaan API, atau tentukan subnet dan grup keamanan saat Anda membuat pekerjaan rekomendasi di konsol. CreateInferenceRecommendationsJob SageMaker

Inference Recommender menggunakan informasi ini untuk membuat endpoint. Saat menyediakan titik akhir, SageMaker buat antarmuka jaringan dan lampirkan ke titik akhir Anda. Antarmuka jaringan menyediakan titik akhir Anda dengan koneksi jaringan ke VPC Anda. Berikut ini adalah contoh `VpcConfig` parameter yang Anda sertakan dalam panggilan ke`CreateInferenceRecommendationsJob`:

```
VpcConfig: {
      "Subnets": [
```

```
            "subnet-0123456789abcdef0",
            "subnet-0123456789abcdef1",
            "subnet-0123456789abcdef2"
            ],
        "SecurityGroupIds": [
            "sg-0123456789abcdef0"
            ]
         }
```

Lihat topik berikut untuk informasi selengkapnya tentang mengonfigurasi VPC Amazon Anda untuk digunakan dengan lowongan Inference Recommendator.

Topik

- Pastikan subnet memiliki alamat IP yang cukup
- Buat VPC Endpoint Amazon S3
- Tambahkan izin untuk pekerjaan Inference Recommender yang berjalan di VPC Amazon ke kebijakan IAM kustom
- Konfigurasikan tabel rute
- Konfigurasikan grup keamanan VPC

Pastikan subnet memiliki alamat IP yang cukup

Subnet VPC Anda harus memiliki setidaknya dua alamat IP pribadi untuk setiap instance dalam pekerjaan rekomendasi inferensi. Untuk informasi selengkapnya tentang subnet dan alamat IP pribadi, lihat Cara kerja Amazon VPC di Panduan Pengguna Amazon VPC.

Buat VPC Endpoint Amazon S3

Jika Anda mengonfigurasi VPC untuk memblokir akses ke internet, Inference Recommender tidak dapat terhubung ke bucket Amazon S3 yang berisi model Anda kecuali Anda membuat titik akhir VPC yang memungkinkan akses. Dengan membuat titik akhir VPC, Anda mengizinkan pekerjaan rekomendasi SageMaker inferensi untuk mengakses bucket tempat Anda menyimpan data dan artefak model.

Untuk membuat VPC endpoint Amazon S3, gunakan prosedur berikut ini:

1. Buka konsol Amazon VPC.
2. Di panel navigasi, pilih titik akhir, lalu pilih Buat titik akhir.

3. Untuk Nama Layanan, cari`com.amazonaws.`*`region`*`.s3`, di *`region`* mana nama Wilayah tempat VPC Anda berada.
4. Pilih jenis Gateway.
5. Untuk VPC, pilih VPC yang ingin Anda gunakan untuk endpoint ini.
6. Untuk Konfigurasi tabel rute, pilih tabel rute yang akan digunakan oleh titik akhir. Layanan VPC secara otomatis menambahkan rute ke setiap tabel rute yang Anda pilih yang mengarahkan lalu lintas Amazon S3 ke titik akhir baru.
7. Untuk Kebijakan, pilih Akses Penuh untuk mengizinkan akses penuh ke layanan Amazon S3 oleh pengguna atau layanan apa pun dalam VPC.

Tambahkan izin untuk pekerjaan Inference Recommender yang berjalan di VPC Amazon ke kebijakan IAM kustom

Kebijakan `AmazonSageMakerFullAccess` terkelola mencakup izin yang Anda perlukan untuk menggunakan model yang dikonfigurasi untuk akses VPC Amazon dengan titik akhir. Izin ini memungkinkan Inference Recommender untuk membuat elastic network interface dan melampirkannya ke tugas rekomendasi inferensi yang berjalan di Amazon VPC. Jika Anda menggunakan kebijakan IAM Anda sendiri, Anda harus menambahkan izin berikut ke kebijakan tersebut untuk menggunakan model yang dikonfigurasi untuk akses VPC Amazon.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {"Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:CreateNetworkInterface",
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Konfigurasikan tabel rute

Gunakan pengaturan DNS default untuk tabel rute titik akhir Anda, sehingga URL Amazon S3 standar (misalnya:) teratasi. *`http://s3-aws-region.amazonaws.com/MyBucket`* Jika Anda tidak menggunakan pengaturan DNS default, pastikan URL yang Anda gunakan untuk menentukan lokasi data dalam pekerjaan rekomendasi inferensi diselesaikan dengan mengonfigurasi tabel rute titik akhir. Untuk informasi tentang tabel rute titik akhir VPC, lihat Titik akhir gateway perutean di Panduan Pengguna Amazon VPC.

Konfigurasikan grup keamanan VPC

Di grup keamanan untuk pekerjaan rekomendasi inferensi, Anda harus mengizinkan komunikasi keluar ke titik akhir VPC Amazon S3 dan rentang CIDR subnet yang digunakan untuk pekerjaan rekomendasi inferensi. Untuk selengkapnya, lihat Aturan Grup Keamanan dan Kontrol akses ke layanan dengan titik akhir Amazon VPC di Panduan Pengguna Amazon VPC.

Jual algoritma dan paket di AWS Marketplace

Amazon SageMaker terintegrasi denganAWS Marketplace, memungkinkan pengembang untuk menagih SageMaker pengguna lain untuk penggunaan algoritme dan paket model mereka. AWS Marketplaceadalah katalog digital yang dikuratori yang memudahkan pelanggan menemukan, membeli, menyebarkan, dan mengelola perangkat lunak dan layanan pihak ketiga yang dibutuhkan pelanggan untuk membangun solusi dan menjalankan bisnis mereka. AWS Marketplacemencakup ribuan daftar perangkat lunak dalam kategori populer, seperti keamanan, jaringan, penyimpanan, pembelajaran mesin, kecerdasan bisnis, database, danDevOps. Ini menyederhanakan lisensi dan pengadaan perangkat lunak dengan opsi harga yang fleksibel dan beberapa metode penyebaran.

Untuk selengkapnya, lihat AWS MarketplaceDokumentasi.

Topik

- SageMakerAlgoritma
- SageMakerPaket Model
- Jual Paket SageMaker Model dan Algoritma Amazon
- Temukan dan Berlangganan Algoritma dan Paket Model di AWS Marketplace
- Gunakan Algoritma dan Sumber Daya Package Model

SageMakerAlgoritma

Algoritma memungkinkan Anda untuk melakukan pembelajaran mesin end-to-end. Ini memiliki dua komponen logis: pelatihan dan inferensi. Pembeli dapat menggunakan komponen pelatihan untuk membuat pekerjaan pelatihan SageMaker dan membangun model pembelajaran mesin. SageMakermenyimpan artefak model yang dihasilkan oleh algoritme selama pelatihan ke bucket Amazon S3. Untuk informasi selengkapnya, lihat Latih Model dengan Amazon SageMaker.

Pembeli menggunakan komponen inferensi dengan artefak model yang dihasilkan selama pekerjaan pelatihan untuk membuat model yang dapat diterapkan di akun mereka. SageMaker Mereka dapat menggunakan model deployable untuk inferensi real-time dengan menggunakan SageMaker layanan hosting. Atau, mereka bisa mendapatkan kesimpulan untuk seluruh dataset dengan menjalankan batch transform jobs. Untuk informasi selengkapnya, lihat Menerapkan Model di Amazon SageMaker.

SageMakerPaket Model

Pembeli menggunakan paket model untuk membangun model deployable di. SageMaker Mereka dapat menggunakan model deployable untuk inferensi real-time dengan menggunakan SageMaker layanan hosting. Atau, mereka bisa mendapatkan kesimpulan untuk seluruh dataset dengan menjalankan batch transform jobs. Untuk informasi selengkapnya, lihat Menerapkan Model di Amazon SageMaker. Sebagai penjual, Anda dapat membuat artefak model Anda dengan berlatihSageMaker, atau Anda dapat menggunakan artefak model Anda sendiri dari model yang Anda latih di luar. SageMaker Anda dapat menagih pembeli untuk inferensi.

Gunakan algoritme dan model Anda sendiri dengan Marketplace AWS

Bagian berikut menunjukkan cara membuat algoritme dan sumber daya paket model yang dapat Anda gunakan secara lokal dan memublikasikan ke AWS Marketplace.

Topik

- Buat Sumber Daya Algoritma dan Package Model
- Gunakan Algoritma dan Sumber Daya Package Model

Buat Sumber Daya Algoritma dan Package Model

Setelah kode latihan dan/atau inferensi Anda dikemas dalam kontainer Docker, buat algoritme dan sumber daya paket model yang dapat Anda gunakan di Amazon SageMaker akun dan, secara opsional, publikasikanAWS Marketplace.

Topik

- Buat Sumber Daya Algoritma
- Membuat Sumber Daya Package Model

Buat Sumber Daya Algoritma

Untuk membuat sumber daya algoritme yang dapat Anda gunakan untuk menjalankan pekerjaan pelatihan di Amazon SageMaker dan terbitanAWS Marketplacetentukan informasi berikut:

- Container Docker yang berisi pelatihan dan, secara opsional, kode inferensi.

- Konfigurasi data input yang diharapkan oleh algoritma Anda untuk pelatihan.
- Hyperparameter yang didukung algoritme Anda.
- Metrik yang dikirim algoritme Anda ke Amazon CloudWatch selama pekerjaan pelatihan.
- Jenis instans yang didukung algoritme Anda untuk pelatihan dan inferensi, dan apakah itu mendukung pelatihan terdistribusi di beberapa instans.
- Profil validasi, yang merupakan pekerjaan pelatihan yang SageMaker digunakan untuk menguji kode pelatihan algoritme Anda dan mengubah pekerjaan batch yang SageMaker berjalan untuk menguji kode inferensi algoritma Anda.

  Untuk memastikan bahwa pembeli dan penjual dapat yakin bahwa produk bekerja SageMaker, kami mengharuskan Anda memvalidasi algoritme Anda sebelum mencantumkannyaAWS Marketplace. Anda dapat mencantumkan produk diAWS Marketplacehanya jika validasi berhasil. Untuk memvalidasi algoritme Anda, SageMaker menggunakan profil validasi dan data sampel Anda untuk menjalankan tugas validasi berikut:

  1. Buat pekerjaan pelatihan di akun Anda untuk memverifikasi bahwa gambar latihan Anda berfungsi SageMaker.
  2. Jika Anda menyertakan kode inferensi dalam algoritme, buat model di akun Anda menggunakan gambar inferensi algoritme dan artefak model yang dihasilkan oleh pekerjaan pelatihan.
  3. Jika Anda menyertakan kode inferensi dalam algoritme, buat pekerjaan transformasi di akun Anda menggunakan model untuk memverifikasi bahwa gambar inferensi Anda berfungsi SageMaker.

  Saat Anda mencantumkan produk AndaAWS Marketplace, input dan output dari proses validasi ini bertahan sebagai bagian dari produk Anda dan tersedia untuk pembeli Anda. Ini membantu pembeli memahami dan mengevaluasi produk sebelum mereka membelinya. Misalnya, pembeli dapat memeriksa data input yang Anda gunakan, output yang dihasilkan, dan log serta metrik yang dipancarkan oleh kode Anda. Semakin komprehensif spesifikasi validasi Anda, semakin mudah bagi pelanggan untuk mengevaluasi produk Anda.

  > **Note**
  >
  > Di profil validasi Anda, berikan hanya data yang ingin Anda paparkan secara publik.

  Validasi bisa memakan waktu hingga beberapa jam. Untuk melihat status pekerjaan di akun Anda, di SageMaker konsol, lihatTugas pelatihandanTugas Transformasihalaman. Jika validasi gagal,

Anda dapat mengakses laporan pemindaian dan validasi dari SageMaker konsol. Jika ada masalah yang ditemukan, Anda harus membuat algoritma lagi.

> **Note**
>
> Untuk mempublikasikan algoritme Anda padaAWS Marketplace, setidaknya satu profil validasi diperlukan.

Anda dapat membuat algoritma dengan menggunakan SageMaker konsol atau SageMakerAPI.

Topik

- Buat Sumber Daya Algoritma (Konsol)
- Buat Resource Algorithm (API)

Buat Sumber Daya Algoritma (Konsol)

Untuk membuat sumber daya algoritma (konsol)

1. Memiilih SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Dari menu sebelah kiri, pilihPelatihan.
3. Dari menu pilihan menurun, pilihAlgoritme, lalu pilihBuat algoritme.
4. PadaSpesifikasi pelatihanhalaman, berikan informasi berikut:
    a. UntukNama algoritme, ketik nama untuk algoritma Anda. Nama algoritma harus unik di akun Anda dan diAWSwilayah. Nama harus memiliki 1 hingga 64 karakter. Karakter yang valid adalah a-z, 0-9, dan - (tanda hubung).
    b. Ketik deskripsi untuk algoritme Anda. Deskripsi ini muncul di SageMaker konsol dan diAWS Marketplace.
    c. UntukGambar latihan, ketik jalur di Amazon ECR tempat wadah latihan Anda disimpan.
    d. UntukSupport pelatihan terdistribusi, Memiilihyajika algoritme Anda mendukung pelatihan pada beberapa instance. Jika tidak, pilih Tidak.
    e. UntukSupport jenis instans untuk pelatihan, pilih jenis instans yang didukung algoritme Anda.

f. UntukSpesifikasi saluran, tentukan hingga 8 saluran data input untuk algoritme Anda. Misalnya, Anda dapat menentukan 3 saluran input yang diberi nama`train`,`validation`, dan`test`. Untuk setiap saluran, tentukan informasi berikut:

i. UntukNama saluran, ketik nama untuk saluran. Nama harus memiliki 1 hingga 64 karakter. Karakter yang valid adalah a-z, 0-9, dan - (tanda hubung).

ii. Untuk meminta saluran untuk algoritme Anda, pilihDiperlukan saluran.

iii. Ketikkan deskripsi untuk saluran tersebut.

iv. UntukMode input yang didukung, MemiilihMode pipajika algoritme Anda mendukung streaming data input, danMode berkasjika algoritme Anda mendukung pengunduhan data input sebagai file. Anda dapat memilih keduanya.

v. UntukJenis konten yang didukung, ketik tipe MIME yang diharapkan algoritme Anda untuk input data.

vi. UntukTipe kompresi yang didukung, MemiilihGzipjika algoritme Anda mendukung kompresi Gzip. Jika tidak, pilihTidak ada.

vii. MemiilihMenambahkan saluranuntuk menambahkan saluran input data lain, atau memilihSelanjutnyajika Anda selesai menambahkan saluran.

5. PadaPenyetelan spesifikasihalaman, berikan informasi berikut:

a. UntukSpesifikasi hiperparameter, tentukan hyperparameters yang didukung algoritme Anda dengan mengedit objek JSON. Untuk setiap hyperparameter yang didukung algoritme Anda, buat blok JSON yang serupa dengan yang berikut ini:

```
{
"DefaultValue": "5",
"Description": "The first hyperparameter",
"IsRequired": true,
"IsTunable": false,
"Name": "intRange",
"Range": {
"IntegerParameterRangeSpecification": {
"MaxValue": "10",
"MinValue": "1"
},
"Type": "Integer"
}
```

Dalam JSON, berikan yang berikut ini:

i. Untuk`DefaultValue`, tentukan nilai default untuk hyperparameter, jika ada.

ii. Untuk`Description`, tentukan deskripsi untuk parameter.

iii. Untuk`IsRequired`, tentukan apakah hyperparameter diperlukan.

iv. Untuk`IsTunable`, tentukan`true`jika hyperparameter ini dapat disetel ketika pengguna menjalankan pekerjaan penyetelan hyperparameter yang menggunakan algoritma ini. Untuk informasi, lihat Lakukan Penyetelan Model Otomatis dengan SageMaker.

v. Untuk`Name`, tentukan nama untuk parameter.

vi. Untuk`Range`, tentukan salah satu hal berikut:

- `IntegerParameterRangeSpecification`- nilai-nilai hyperparameter adalah bilangan bulat. Tentukan nilai minimum dan maksimum untuk hyperparameter.
-
- `ContinuousParameterRangeSpecification`- nilai hyperparameter adalah nilai floating-point. Tentukan nilai minimum dan maksimum untuk hyperparameter.
- `CategoricalParameterRangeSpecification`- nilai hyperparameter adalah nilai kategoris. Tentukan daftar semua nilai yang mungkin.

vii. Untuk`Type`, tentukan`Integer`,`Continuous`, atau`Categorical`. Nilai harus sesuai dengan jenis`Range`yang Anda tentukan.

b. UntukDefinisi metrik, tentukan metrik pelatihan apa pun yang Anda inginkan untuk dikeluarkan oleh algoritme Anda. SageMaker menggunakan ekspresi reguler yang Anda tentukan untuk menemukan metrik dengan menguraikan log dari wadah latihan selama latihan. Pengguna dapat melihat metrik ini ketika mereka menjalankan pekerjaan pelatihan dengan algoritme Anda, dan mereka dapat memantau dan merencanakan metrik di Amazon CloudWatch. Untuk informasi, lihat Memantau dan Menganalisis Pekerjaan Pelatihan Menggunakan Amazon CloudWatch Metrics. Untuk setiap metrik, berikan informasi berikut ini:

i. UntukNama metrik, ketik nama untuk metrik.

ii. Untuk`Regex`, ketik ekspresi reguler yang SageMaker digunakan untuk mengurai log pelatihan sehingga dapat menemukan nilai metrik.

iii. UntukDukungan metrik objektifpilihyajika metrik ini dapat digunakan sebagai metrik objektif untuk pekerjaan penyetelan hyperparameter. Untuk informasi, lihat Lakukan Penyetelan Model Otomatis dengan SageMaker.

iv. MemiilihTambahkan metrikuntuk menambahkan metrik lain, atau pilihSelanjutnyajika Anda selesai menambahkan metrik.

6. PadaSpesifikasi inferensihalaman, berikan informasi berikut ini jika algoritme Anda mendukung kesimpulan:

   a. UntukLokasi gambar inferensi, ketik jalur di Amazon ECR tempat kontainer inferensi Anda disimpan.

   b. UntukNama host DNS kontainer, ketik nama host DNS untuk gambar Anda.

   c. UntukTipe instans yang didukung untuk inferensi waktu nyata, pilih jenis instans yang didukung algoritme Anda untuk model yang digunakan sebagai titik akhir yang di-host SageMaker. Untuk informasi, lihat Menyebarkan model untuk inferensi.

   d. UntukJenis instans yang didukung untuk pekerjaan transformasi batch, pilih jenis instans yang didukung algoritme Anda untuk pekerjaan transformasi batch. Untuk informasi, lihat Gunakan Batch Transform.

   e. UntukJenis konten yang didukung, ketik jenis data input yang diharapkan algoritme Anda untuk permintaan inferensi.

   f. UntukTipe MIME respons yang didukung, ketik tipe MIME yang didukung algoritme Anda untuk respons inferensi.

   g. Pilih Selanjutnya.

7. PadaSpesifikasi validasihalaman, berikan informasi berikut:

   a. UntukPublikasikan algoritma ini padaAWS Marketplace, Memiilihyauntuk mempublikasikan algoritmeAWS Marketplace.

   b. UntukValidasi sumber daya ini, Memiilihyajika Anda ingin SageMaker untuk menjalankan pekerjaan pelatihan dan/atau pekerjaan transformasi batch yang Anda tentukan untuk menguji kode pelatihan dan/atau inferensi algoritme Anda.

      > **Note**
      >
      > Untuk mempublikasikan algoritme Anda padaAWS Marketplace, algoritma Anda harus divalidasi.

   c. UntukPeran IAM, pilih peran IAM yang memiliki izin yang diperlukan untuk menjalankan tugas pelatihan dan mengubah tugas SageMaker, atau pilihMembuat peran barumengizinkan SageMaker untuk menciptakan peran yang

memilikiAmazonSageMakerFullAccesskebijakan yang dikelola terlampir. Untuk informasi, lihat SageMaker Peran.

d. UntukProfil validasi, tentukan hal berikut:

- Nama untuk profil validasi.
- SEBUAHDefinisi tugas pelatihan. Ini adalah blok JSON yang menggambarkan pekerjaan pelatihan. Ini dalam format yang sama dengan`TrainingJobDefinition`parameter masukan dari`CreateAlgorithm`API.
- SEBUAHTransformasi ketentuan tugas Transformasi. Ini adalah blok JSON yang menggambarkan pekerjaan transformasi batch. Ini dalam format yang sama dengan`TransformJobDefinition`parameter masukan dari`CreateAlgorithm`API.

e. MemiilihBuat algoritme.

Buat Resource Algorithm (API)

Untuk membuat sumber daya algoritma dengan menggunakan SageMaker API, panggil`CreateAlgorithm`API.

Membuat Sumber Daya Package Model

Untuk membuat sumber daya paket model yang dapat Anda gunakan untuk membuat model yang dapat diterapkan di Amazon SageMaker dan terbitanAWS Marketplacetentukan informasi berikut:

- Container Docker yang berisi kode inferensi, atau sumber daya algoritme yang digunakan untuk melatih model.
- Lokasi model artefact. Artefak model dapat dikemas dalam wadah Docker yang sama dengan kode inferensi atau disimpan di Amazon S3.
- Jenis instans yang didukung paket model Anda untuk pekerjaan inferensi real-time dan transformasi batch.
- profil validasi, yang batch mengubah pekerjaan yang SageMaker berjalan untuk menguji kode inferensi paket model Anda.

  Sebelum mencantumkan paket model padaAWS Marketplace, Anda harus memvalidasi mereka. Ini memastikan bahwa pembeli dan penjual dapat yakin bahwa produk berfungsi di Amazon SageMaker. Anda dapat mencantumkan produk diAWS Marketplacehanya jika validasi berhasil.

Prosedur validasi menggunakan profil validasi dan data sampel Anda untuk menjalankan tugas validasi berikut:

1. Buat model di akun Anda menggunakan gambar inferensi paket model dan artefak model opsional yang disimpan di Amazon S3.

   > **Note**
   >
   > Paket model khusus untuk wilayah tempat Anda membuatnya. Bucket S3 tempat artefak model disimpan harus berada di wilayah yang sama di mana Anda membuat paket model.

2. Buat pekerjaan transformasi di akun Anda menggunakan model untuk memverifikasi bahwa gambar inferensi Anda berfungsi SageMaker.
3. Buat profil validasi.

> **Note**
>
> Di profil validasi Anda, berikan hanya data yang ingin Anda paparkan secara publik.

Validasi bisa memakan waktu hingga beberapa jam. Untuk melihat status pekerjaan di akun Anda, di SageMaker konsol, lihatLowongan Transformasihalaman. Jika validasi gagal, Anda dapat mengakses laporan pemindaian dan validasi dari SageMaker konsol. Setelah memperbaiki masalah, buat ulang algoritma. Ketika status algoritma adalah`COMPLETED`, menemukannya di SageMaker konsol dan mulai proses listing

> **Note**
>
> Untuk mempublikasikan paket model Anda diAWS Marketplace, setidaknya satu profil validasi diperlukan.

Anda dapat membuat paket model baik dengan menggunakan SageMaker konsol atau dengan menggunakan SageMaker API.

Topik

- Membuat Sumber Daya Package Model (Konsol)

- Membuat Sumber Daya Package Model (API)

Membuat Sumber Daya Package Model (Konsol)

Untuk membuat paket model di SageMaker Konsol:

1. Memiilih SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. Dari menu sebelah kiri, pilihInferensi.
3. MemiilihPaket model Marketplace, lalu pilihBuat paket model marketplace.
4. PadaSpesifikasi inferensihalaman, berikan informasi berikut:

    a. UntukNama paket model, Ketikkan nama untuk paket model Anda. Nama paket model harus unik di akun Anda dan diAWSwilayah. Nama harus memiliki 1 hingga 64 karakter. Karakter yang valid adalah a-z, dan - (tanda hubung).

    b. Ketik deskripsi untuk paket model Anda. Deskripsi ini muncul di SageMaker konsol dan diAWS Marketplace.

    c. UntukOpsi spesifikasi, MemiilihMenyediakan lokasi gambar inferensi dan model artefakuntuk membuat paket model dengan menggunakan wadah inferensi dan model artefak. MemiilihMenyediakan algoritma yang digunakan untuk pelatihan dan artefak modelnyauntuk membuat paket model dari sumber daya algoritma yang Anda buat atau berlangganan dariAWS Marketplace.

    d. Jika Anda memilihMenyediakan lokasi gambar inferensi dan model artefakuntukOpsi spesifikasi, berikan informasi berikut untukDefinisi kontainerdanSumber Daya yang Didukung:

        i. UntukLokasi gambar inferensi, ketik path ke gambar yang berisi kode inferensi Anda. Gambar harus disimpan sebagai wadah Docker di Amazon ECR.

        ii. UntukLokasi artefak data model, ketik lokasi di S3 tempat artefak model Anda disimpan.

        iii. UntukNama host DNS, ketik nama host DNS yang akan digunakan untuk wadah Anda.

        iv. UntukJenis instans yang mendukung untuk inferensi, pilih jenis instans yang didukung paket model Anda untuk inferensi real-time SageMaker titik akhir host.

        v. UntukJenis instans yang didukung untuk pekerjaan transformasi batch, pilih jenis instans yang didukung paket model Anda untuk pekerjaan transformasi batch.

        vi. Jenis konten yang mendukung, ketik jenis konten yang diharapkan paket model Anda untuk permintaan inferensi.

vii. UntukJenis MIME, ketik tipe MIME yang digunakan paket model Anda untuk memberikan inferensi.

e. Jika Anda memilihMenyediakan algoritma yang digunakan untuk pelatihan dan artefak modelnyauntukOpsi spesifikasi, berikan informasi berikut:

i. UntukALGORITHM, ketikkan Amazon Resource Name (ARN) dari sumber daya algoritme yang akan digunakan untuk membuat paket model.

ii. UntukLokasi artefak data model, ketik lokasi di S3 tempat artefak model Anda disimpan.

f. Pilih Selanjutnya.

5. PadaValidasi dan pemindaianhalaman, berikan informasi berikut:

a. UntukPublikasikan paket model ini padaAWS Marketplace, Memiilihyauntuk mempublikasikan paket model padaAWS Marketplace.

b. UntukValidasi sumber daya ini, Memiilihyajika Anda ingin SageMaker untuk menjalankan pekerjaan transformasi batch yang Anda tentukan untuk menguji kode inferensi paket model Anda.

> **Note**
>
> Untuk mempublikasikan paket model Anda diAWS Marketplace, paket model Anda harus divalidasi.

c. UntukPeran IAMMemiilih peran IAM yang memiliki izin yang diperlukan untuk menjalankan tugas transformasi SageMaker, atau MemiilihBuat peran baruuntuk mengizinkan SageMaker untuk menciptakan peran yang memilikiAmazonSageMakerFullAccesskebijakan yang dikelola terlampir. Untuk informasi, lihat SageMaker Peran.

d. UntukProfil validasi, tentukan sebagai berikut:

- Nama untuk profil validasi.
- SEBUAHMengubah ketentuan tugas. Ini adalah blok JSON yang menggambarkan pekerjaan transformasi batch. Ini dalam format yang sama dengan`TransformJobDefinition`parameter masukan dari`CreateAlgorithm`API.

6. MemiilihBuat paket model marketplace.

Membuat Sumber Daya Package Model (API)

Untuk membuat paket model dengan menggunakan SageMaker API, panggil`CreateModelPackage`API.

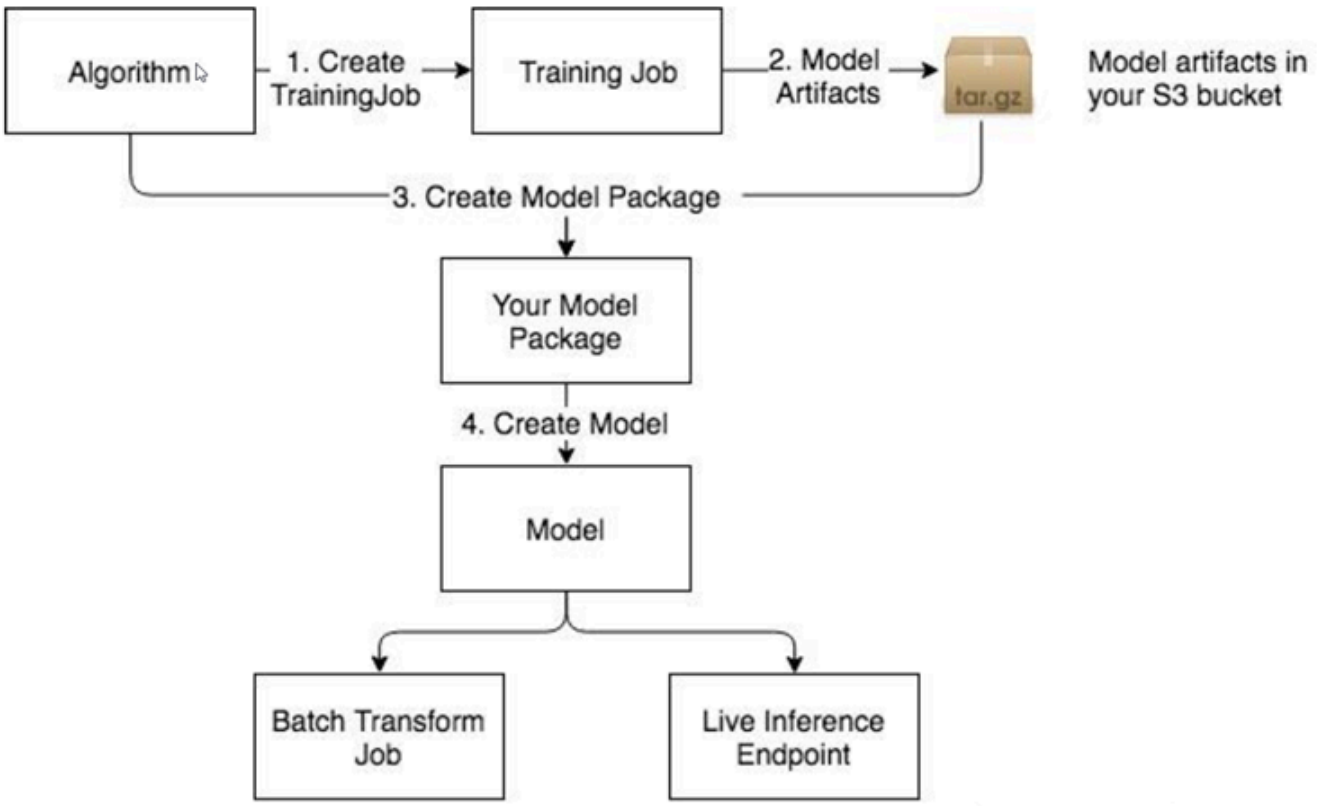
Gunakan Algoritma dan Sumber Daya Package Model

Anda dapat membuat algoritme dan paket model sebagai sumber daya di Amazon SageMaker akun, dan Anda dapat menemukan dan berlangganan algoritma dan paket modelAWS Marketplace.

Gunakan algoritma untuk:

- Jalankan pekerjaan pelatihan. Untuk informasi, lihat Gunakan Algoritma untuk Menjalankan Job Pelatihan.
- Jalankan tugas penyetelan hyperparameter. Untuk informasi, lihat Gunakan Algoritma untuk Menjalankan Job Penyetelan Hyperparameter.
- Buat Paket Model. Setelah Anda menggunakan sumber daya algoritma untuk menjalankan pekerjaan pelatihan atau pekerjaan penyetelan hyperparameter, Anda dapat menggunakan artefak model yang dihasilkan pekerjaan ini bersama dengan algoritma untuk membuat paket model. Untuk informasi, lihat Membuat Sumber Daya Package Model.

> **Note**
>
> Jika Anda berlangganan algoritmaAWS Marketplace, Anda harus membuat paket model sebelum Anda dapat menggunakannya untuk mendapatkan kesimpulan dengan membuat endpoint host atau menjalankan pekerjaan batch transform.



Gunakan paket model untuk:

- Buat model yang dapat Anda gunakan untuk mendapatkan inferensi waktu nyata atau menjalankan pekerjaan transformasi batch. Untuk informasi, lihat Menggunakan Package Model untuk Membuat Model.
- Buat titik akhir yang dihosting untuk mendapatkan inferensi waktu nyata. Untuk informasi, lihat Menyebarkan Model ke SageMaker Layanan Hosting.
- Buat pekerjaan transformasi batch. Untuk informasi, lihat (Opsional) Buat Prediksi dengan Batch Transform.

Topik

- Gunakan Algoritma untuk Menjalankan Job Pelatihan
- Gunakan Algoritma untuk Menjalankan Job Penyetelan Hyperparameter
- Menggunakan Package Model untuk Membuat Model

Gunakan Algoritma untuk Menjalankan Job Pelatihan

Anda dapat membuat menggunakan sumber daya algoritme untuk membuat pekerjaan pelatihan dengan menggunakan Amazon SageMaker konsol, Amazon tingkat rendah SageMaker API, atauAmazon SageMaker Python.

Topik

- Menggunakan Algoritma untuk Menjalankan Job Pelatihan (Konsol)
- Gunakan Algoritma untuk Menjalankan Training Job (API)
- Gunakan Algoritma untuk Menjalankan Job Pelatihan (Amazon SageMaker Python)

Menggunakan Algoritma untuk Menjalankan Job Pelatihan (Konsol)

Untuk menggunakan algoritma untuk menjalankan pekerjaan pelatihan (konsol)

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. MemiilihAlgoritme.
3. Pilih algoritme yang Anda buat dari daftar diAlgoritme sayatab atau pilih algoritma yang Anda berlangganan diAWS Marketplacelangganantab.
4. MemiilihBuat tugas pelatihan.

   Algoritma yang Anda pilih akan dipilih secara otomatis.
5. PadaBuat tugas pelatihanHalaman, berikan informasi berikut:

   a. UntukNama tugas, ketik nama untuk pekerjaan pelatihan.

   b. UntukPeran IAM, pilih peran IAM yang memiliki izin yang diperlukan untuk menjalankan tugas pelatihan di SageMaker, atau pilihMembuat peran baruuntuk mengizinkan SageMaker untuk menciptakan peran yang memilikiAmazonSageMakerFullAccesskebijakan yang dikelola terlampir. Untuk informasi, lihat SageMaker Peran.

   c. Untukkonfigurasi sumber daya, berikan informasi berikut:

      i. UntukJenis instans, pilih jenis instans yang digunakan untuk pelatihan.

      ii. UntukJumlah instans, ketik jumlah instans ML yang akan digunakan untuk pekerjaan pelatihan.

iii. UntukVolume tambahan per instans (GB), ketikkan ukuran volume penyimpanan mL yang ingin Anda sediakan. Volume penyimpanan ML menyimpan artefak model dan status tambahan.

iv. UntukKunci enkripsi, jika Anda ingin Amazon SageMaker untuk menggunakanAWSKunci Layanan Manajemen Kunci untuk mengenkripsi data dalam volume penyimpanan ML yang dilampirkan ke instance pelatihan, tentukan kuncinya.

v. UntukKondisi terhenti, tentukan jumlah maksimum waktu dalam detik, menit, menit, atau hari, yang ingin Anda jalankan tugas pelatihan.

d. UntukVPC, pilih Amazon VPC yang ingin Anda izinkan wadah pelatihan Anda untuk mengakses. Untuk informasi selengkapnya, lihat Berikan Akses Pekerjaan SageMaker Pelatihan ke Sumber Daya di VPC Amazon Anda.

e. UntukHyperparameter, tentukan nilai hiperparameter yang akan digunakan untuk pekerjaan pelatihan.

f. Untukkonfigurasi data input, tentukan nilai berikut untuk setiap saluran data input yang akan digunakan untuk pekerjaan pelatihan. Anda dapat melihat saluran algoritme yang Anda gunakan untuk dukungan pelatihan, dan jenis konten, jenis kompresi yang didukung, dan mode input yang didukung untuk setiap saluran, di bawahSpesifikasi saluranbagianAlgoritmehalaman untuk algoritma.

i. UntukNama saluran, ketikkan nama saluran input.

ii. UntukJenis Konten, ketik jenis konten dari data yang diharapkan algoritma untuk saluran.

iii. UntukJenis kompresi, pilih jenis kompresi data yang akan digunakan, jika ada.

iv. UntukRekam pembungkus, pilih`RecordIO`jika algoritma mengharapkan data di`RecordIO`format.

v. UntukTipe data S3,Tipe distribusi data S3, danLokasi S3, tentukan nilai yang sesuai. Untuk informasi tentang apa arti nilai-nilai ini, lihat`S3DataSource`.

vi. UntukMode input, pilihBerkasuntuk mengunduh data dari volume penyimpanan yang disediakan, dan me-mount direktori ke volume Docker. MemiilihAlurUntuk melakukan streaming data secara langsung dari Amazon S3 ke kontainer.

vii. Untuk menambahkan saluran input lain, pilihMenambahkan Saluran. Jika Anda selesai menambahkan saluran input, pilihSelesai.

g. UntukOutputlocation, tentukan nilai-nilai berikut:

i. UntukPath keluaran S3, pilih lokasi S3 tempat pekerjaan pelatihan menyimpan output, seperti artefak model.

> **Note**
>
> Anda menggunakan artefak model yang disimpan di lokasi ini untuk membuat model atau paket model dari pekerjaan pelatihan ini.

ii. UntukKunci enkripsi, jika Anda ingin SageMaker untuk menggunakanAWS KMSkunci untuk mengenkripsi data keluaran saat istirahat di lokasi S3.

h. UntukTag, tentukan satu tag atau lebih untuk mengelola tugas pelatihan. Setiap tanda terdiri dari kunci dan nilai opsional. Kunci tanda harus unik untuk setiap sumber daya.

i. MemiilihBuat tugas pelatihanuntuk menjalankan pekerjaan pelatihan.

Gunakan Algoritma untuk Menjalankan Training Job (API)

Untuk menggunakan algoritma untuk menjalankan pekerjaan pelatihan dengan menggunakan SageMaker API, tentukan nama atau Amazon Resource Name (ARN) sebagai`AlgorithmName`bidang`AlgorithmSpecification`objek yang Anda lewati`CreateTrainingJob`. Untuk informasi tentang model pelatihan di SageMaker, LihatLatih Model dengan Amazon SageMaker.

Gunakan Algoritma untuk Menjalankan Job Pelatihan (Amazon SageMaker Python)

Gunakan algoritme yang Anda buat atau berlangganAWS Marketplaceuntuk membuat pekerjaan pelatihan, membuat`AlgorithmEstimator`objek dan tentukan Amazon Resource Name (ARN) atau nama algoritme sebagai nilai`algorithm_arn`argumen. Kemudian panggil`fit`dari estimator. Misalnya:

```
from sagemaker import AlgorithmEstimator
data_path = os.path.join(DATA_DIR, 'marketplace', 'training')

algo = AlgorithmEstimator(
algorithm_arn='arn:aws:sagemaker:us-east-2:012345678901:algorithm/my-algorithm',
        role='SageMakerRole',
        instance_count=1,
        instance_type='ml.c4.xlarge',
        sagemaker_session=sagemaker_session,
```

```
        base_job_name='test-marketplace')

train_input = algo.sagemaker_session.upload_data(
path=data_path, key_prefix='integ-test-data/marketplace/train')

algo.fit({'training': train_input})
```

Gunakan Algoritma untuk Menjalankan Job Penyetelan Hyperparameter

Pekerjaan penyetelan hyperparameter menemukan versi terbaik dari model dengan menjalankan banyak pekerjaan pelatihan pada kumpulan data Anda menggunakan algoritme dan rentang hyperparameter yang Anda tentukan. Kemudian memilih nilai hyperparameter yang menghasilkan model yang melakukan yang terbaik, yang diukur dengan metrik yang Anda pilih. Untuk informasi selengkapnya, lihat Lakukan Penyetelan Model Otomatis dengan SageMaker.

Anda dapat membuat menggunakan sumber daya algoritme untuk membuat pekerjaan penyetelan hyperparameter dengan menggunakan Amazon SageMaker konsol, Amazon tingkat rendah SageMaker API, atauAmazon SageMaker Python.

Topik

- Menggunakan Algoritma untuk Menjalankan Hyperparameter Tuning Job (Console)
- Gunakan Algoritma untuk Menjalankan Hyperparameter Tuning Job (API)
- Gunakan Algoritma untuk Menjalankan Job Penyetelan Hyperparameter (Amazon SageMaker Python)

Menggunakan Algoritma untuk Menjalankan Hyperparameter Tuning Job (Console)

Untuk menggunakan algoritma untuk menjalankan pekerjaan penyetelan hyperparameter (konsol)

1. Buka SageMaker konsol dihttps://console.aws.amazon.com/sagemaker/.
2. PilihAlgoritme me.
3. Pilih algoritme yang Anda buat dari daftar diAlgoritme sayatab atau pilih algoritma yang Anda berlangganan diAWS MarketplacelangganTab.
4. PilihBuat pekerjaan penyetelan hyperparameter.

   Algoritma yang Anda pilih akan dipilih secara otomatis.
5. PadaBuat pekerjaan penyetelan hyperparameterhalaman, berikan informasi berikut:

a. UntukMulai hangatPilihAktifkan awal yang hangatuntuk menggunakan informasi dari pekerjaan penyetelan hyperparameter sebelumnya sebagai titik awal untuk pekerjaan penyetelan hyperparameter ini. Untuk informasi selengkapnya, lihat Jalankan Pekerjaan Tuning Hyperparameter Mulai yang Hangat.

   i. PilihData dan algoritma yang identikjika data input Anda sama dengan data input untuk pekerjaan induk dari pekerjaan penyetelan hyperparameter ini, atau pilihPembelajaran transferuntuk menggunakan data input tambahan atau berbeda untuk pekerjaan penyetelan hyperparameter ini.

   ii. UntukPekerjaan penyetelan hyperparameter orang tua, pilih hingga 5 pekerjaan penyetelan hyperparameter untuk digunakan sebagai orang tua untuk pekerjaan penyetelan hyperparameter ini.

b. UntukNama pekerjaan penyetelan hyperparameter, ketik nama untuk pekerjaan tuning.

c. UntukPeran IAMPilih peran IAM yang memiliki izin yang diperlukan untuk menjalankan tugas penyetelan hyperparameter di SageMaker, atau PilihBuat peran baruyang mengizinkan SageMaker untuk menciptakan peran yang memiliki`AmazonSageMakerFullAccess`kebijakan yang dikelola terlampir. Untuk informasi, lihat SageMaker Peran.

d. UntukVPC, pilih Amazon VPC yang ingin Anda izinkan pekerjaan pelatihan yang diluncurkan oleh pekerjaan penyetelan untuk diakses. Untuk informasi selengkapnya, lihat Berikan Akses Pekerjaan SageMaker Pelatihan ke Sumber Daya di VPC Amazon Anda.

e. Pilih Next (Berikutnya).

f. UntukMetrik objektif, pilih metrik yang digunakan pekerjaan penyetelan hyperparameter untuk menentukan kombinasi hyperparameter terbaik, dan pilih apakah akan meminimalkan atau memaksimalkan metrik ini. Untuk informasi selengkapnya, lihat Lihat Training Job Terbaik.

g. UntukKonfigurasi hiperparameter, pilih rentang untuk hyperparameter merdu yang Anda inginkan untuk mencari pekerjaan tuning, dan tetapkan nilai statis untuk hyperparameter yang ingin Anda tetap konstan dalam semua pekerjaan pelatihan yang diluncurkan oleh pekerjaan penyetelan hyperparameter. Untuk informasi selengkapnya, lihat Tentukan Rentang Hyperparameter.

h. Pilih Next (Berikutnya).

i. UntukKonfigurasi data input, tentukan nilai berikut untuk setiap saluran data input yang akan digunakan untuk pekerjaan penyetelan hyperparameter. Anda dapat melihat saluran

apa yang Anda gunakan untuk dukungan penyetelan hyperparameter, dan jenis konten, jenis kompresi yang didukung, dan mode input yang didukung untuk setiap saluran, di bawahSpesifikasi saluranbagianRingkasan algoritma mehalaman untuk algoritma.

i. UntukNama saluran, ketik nama saluran input.

ii. UntukJenis Konten, ketik jenis konten dari data yang diharapkan algoritma untuk saluran.

iii. UntukJenis kompresi baru, pilih jenis kompresi data yang akan digunakan, jika ada.

iv. UntukRekam pembungkusPilih`RecordIO`jika algoritma mengharapkan data di`RecordIO`format.

v. UntukTipe data S3,Tipe distribusi data S3, danLokasi S3, tentukan nilai yang sesuai. Untuk informasi tentang apa arti nilai-nilai ini, lihat`S3DataSource`.

vi. UntukMode inputPilihBerkasuntuk mengunduh data dari volume penyimpanan yang disediakan, dan me-mount direktori ke volume Docker. PilihAlurUntuk melakukan streaming data secara langsung dari Amazon S3 ke wadah.

vii. Untuk menambahkan saluran input lain, pilihTambahkan saluran. Jika Anda selesai menambahkan saluran input, pilihSelesai.

j. UntukOutputlokasi, tentukan nilai-nilai berikut:

i. UntukPath keluaran S3, pilih lokasi S3 di mana pekerjaan pelatihan bahwa pekerjaan penyetelan hyperparameter ini meluncurkan output toko, seperti artefak model.

> **Note**
>
> Anda menggunakan artefak model yang disimpan di lokasi ini untuk membuat model atau model paket dari pekerjaan penyetelan hyperparameter ini.

ii. UntukKunci enkripsi, jika Anda ingin SageMaker untuk menggunakanAWS KMSkunci untuk mengenkripsi data keluaran saat istirahat di lokasi S3.

k. Untukkonfigurasi sumber daya, berikan informasi berikut:

i. UntukJenis instans, pilih jenis instance yang akan digunakan untuk setiap pekerjaan pelatihan yang diluncurkan oleh pekerjaan penyetelan hyperparameter.

ii. UntukJumlah instans, ketik jumlah instans ML yang akan digunakan untuk setiap pekerjaan pelatihan yang diluncurkan oleh pekerjaan penyetelan hyperparameter.

iii. UntukVolume tambahan per instans (GB), ketik ukuran volume penyimpanan mL yang ingin Anda sediakan setiap tugas pelatihan yang diluncurkan tugas penyetelan hyperparameter. Volume penyimpanan ML menyimpan artefak model dan status tambahan.

iv. UntukKunci enkripsi, jika Anda ingin Amazon SageMaker untuk menggunakanAWSKunci Layanan Manajemen Kunci untuk mengenkripsi data dalam volume penyimpanan ML yang dilampirkan ke instans pelatihan, tentukan kuncinya.

l. UntukBatas sumber daya, berikan informasi berikut:

i. UntukTugas pelatihan maksimum, tentukan jumlah maksimum pekerjaan pelatihan yang Anda inginkan untuk memulai pekerjaan penyetelan hyperparameter. Pekerjaan penyetelan hyperparameter dapat meluncurkan maksimal 500 pekerjaan pelatihan.

ii. UntukPekerjaan pelatihan parallel maksimum, tentukan jumlah maksimum pekerjaan pelatihan bersamaan yang dapat diluncurkan oleh pekerjaan penyetelan hyperparameter. Pekerjaan penyetelan hyperparameter dapat meluncurkan maksimal 10 pekerjaan pelatihan bersamaan.

iii. UntukKondisi penghentian, tentukan jumlah maksimum waktu dalam detik, menit, jam, atau hari, yang Anda inginkan setiap pekerjaan pelatihan yang diluncurkan oleh pekerjaan penyetelan hyperparameter untuk dijalankan.

m. UntukTag, tentukan satu atau lebih tag untuk mengelola tugas penyetelan hyperparameter. Setiap tanda terdiri dari kunci dan nilai opsional. Kunci tanda harus unik untuk setiap sumber daya.

n. PilihBuat tugasuntuk menjalankan pekerjaan penyetelan hyperparameter.

Gunakan Algoritma untuk Menjalankan Hyperparameter Tuning Job (API)

Untuk menggunakan algoritma untuk menjalankan pekerjaan penyetelan hyperparameter dengan menggunakan SageMaker API, tentukan nama atau Amazon Resource Name (ARN) algoritme tersebut sebagai`AlgorithmName`Bidang`AlgorithmSpecification`objek yang Anda lewati`CreateHyperParameterTuningJob`. Untuk informasi tentang penyetelan hyperparameter di SageMaker, LihatLakukan Penyetelan Model Otomatis dengan SageMaker.

Gunakan Algoritma untuk Menjalankan Job Penyetelan Hyperparameter (Amazon SageMaker Python)

Gunakan algoritme yang Anda buat atau berlangganan AWS Marketplace untuk membuat pekerjaan penyetelan hyperparameter, buat `AlgorithmEstimator` objek dan tentukan Amazon Resource Name (ARN) atau nama algoritme sebagai nilai algoritme tersebut sebagai nilai `algorithm_arn` argumen. Kemudian initialize a `HyperparameterTuner` objek dengan `AlgorithmEstimator` yang Anda buat sebagai nilai `estimator` argumen. Akhirnya, hubungi `fit` metode `AlgorithmEstimator`. Misalnya:

```
from sagemaker import AlgorithmEstimator
from sagemaker.tuner import HyperparameterTuner

data_path = os.path.join(DATA_DIR, 'marketplace', 'training')

algo = AlgorithmEstimator(
            algorithm_arn='arn:aws:sagemaker:us-east-2:764419575721:algorithm/scikit-
decision-trees-1542410022',
            role='SageMakerRole',
            instance_count=1,
            instance_type='ml.c4.xlarge',
            sagemaker_session=sagemaker_session,
            base_job_name='test-marketplace')

train_input = algo.sagemaker_session.upload_data(
    path=data_path, key_prefix='integ-test-data/marketplace/train')

algo.set_hyperparameters(max_leaf_nodes=10)
tuner = HyperparameterTuner(estimator=algo, base_tuning_job_name='some-name',
                                objective_metric_name='validation:accuracy',
                                hyperparameter_ranges=hyperparameter_ranges,
                                max_jobs=2, max_parallel_jobs=2)

tuner.fit({'training': train_input}, include_cls_metadata=False)
tuner.wait()
```

Menggunakan Package Model untuk Membuat Model

Gunakan paket model untuk membuat model yang dapat diterapkan yang dapat Anda gunakan untuk mendapatkan inferensi waktu nyata dengan membuat titik akhir yang di-host atau untuk menjalankan pekerjaan transformasi batch. Anda dapat membuat model yang dapat diterapkan dari paket model

dengan menggunakan Amazon SageMaker konsol, tingkat rendah SageMaker API), atauAmazon SageMaker Python SDK.

Topik

- Menggunakan Package Model untuk Membuat Model (Konsol)
- Menggunakan Package Model untuk Membuat Model (API)
- Gunakan Package Model untuk Membuat Model (Amazon SageMaker Python SDK)

Menggunakan Package Model untuk Membuat Model (Konsol)

Untuk membuat model deployable dari paket model (konsol)

1. Buka SageMaker konsolhttps://console.aws.amazon.com/sagemaker/.
2. PilihPaket model.
3. Pilih paket model yang Anda buat dari daftarPaket model sayatab atau pilih paket model yang Anda berlangganan diAWS Marketplacelangganantab.
4. PilihBuat Model.
5. UntukNama model, ketik nama untuk model.
6. UntukPeran IAM, pilih peran IAM yang memiliki izin yang diperlukan untuk memanggil layanan lain atas nama Anda, atau memilihBuat peran baruuntuk mengizinkan SageMaker untuk menciptakan peran yang memiliki`AmazonSageMakerFullAccess`kebijakan yang dikelola terlampir. Untuk informasi, lihat SageMaker Peran.
7. UntukVPC, pilih Amazon VPC yang ingin Anda izinkan model untuk mengakses. Untuk informasi selengkapnya, lihat Berikan Akses Endpoint yang SageMaker Dihosting ke Sumber Daya di VPC Amazon Anda.
8. Biarkan nilai default untukOpsi input kontainerdanPilih paket model.
9. Untuk variabel lingkungan, berikan nama dan nilai variabel lingkungan yang ingin Anda sampaikan ke wadah model.
10. UntukTag, tentukan satu atau lebih tag untuk mengelola model. Setiap tanda terdiri dari kunci dan nilai opsional. Kunci tanda harus unik untuk setiap sumber daya.
11. PilihBuat Model.

Setelah membuat model yang dapat diterapkan, Anda dapat menggunakannya untuk menyiapkan titik akhir untuk inferensi waktu nyata atau membuat pekerjaan transformasi batch untuk

mendapatkan kesimpulan di seluruh kumpulan data. Untuk informasi tentang titik akhir hosting SageMaker, LihatMenerapkan Model untuk Inferensi.

Menggunakan Package Model untuk Membuat Model (API)

Untuk menggunakan paket model untuk membuat model yang dapat diterapkan dengan SageMaker API, tentukan nama atau Amazon Resource Name (ARN) dari paket model sebagai`ModelPackageName`bidang`ContainerDefinition`objek yang Anda lewati ke`CreateModel`API

Setelah membuat model yang dapat diterapkan, Anda dapat menggunakannya untuk menyiapkan titik akhir untuk inferensi waktu nyata atau membuat pekerjaan transformasi batch untuk mendapatkan kesimpulan di seluruh kumpulan data. Untuk informasi tentang endpoint yang di-host di SageMaker, LihatMenerapkan Model untuk Inferensi.

Gunakan Package Model untuk Membuat Model (Amazon SageMaker Python SDK)

Untuk menggunakan paket model untuk membuat model yang dapat diterapkan dengan SageMaker Python SDK, menginisialisasi`ModelPackage`objek, dan teruskan Amazon Resource Name (ARN) dari paket model sebagai`model_package_arn`argumen. Misalnya:

```
from sagemaker import ModelPackage
model = ModelPackage(role='SageMakerRole',
         model_package_arn='training-job-scikit-decision-trees-1542660466-6f92',
         sagemaker_session=sagemaker_session)
```

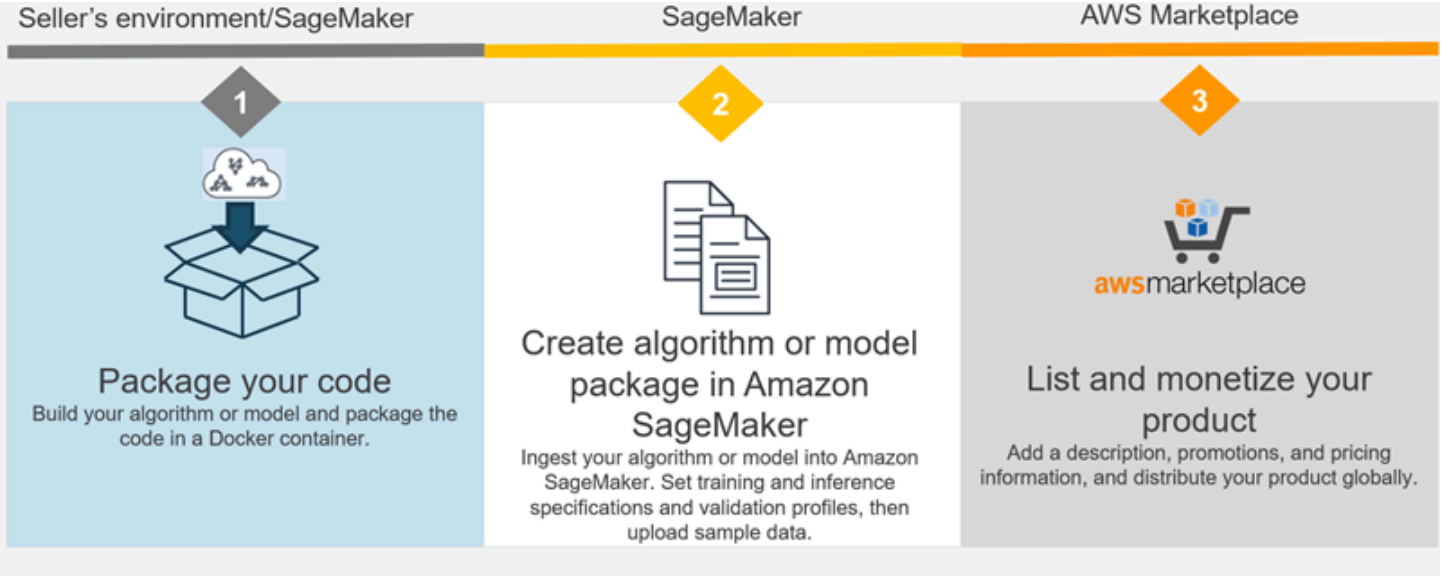
Setelah membuat model yang dapat diterapkan, Anda dapat menggunakannya untuk menyiapkan titik akhir untuk inferensi waktu nyata atau membuat pekerjaan transformasi batch untuk mendapatkan kesimpulan di seluruh kumpulan data. Untuk informasi tentang titik akhir hosting SageMaker, LihatMenerapkan Model untuk Inferensi.

Jual Paket SageMaker Model dan Algoritma Amazon

Menjual SageMaker algoritma Amazon dan paket model adalah proses tiga langkah:

1. Kembangkan algoritme atau model Anda, dan kemaskan dalam wadah Docker. Untuk informasi, lihat Mengembangkan Algoritma dan Model di Amazon SageMaker.
2. Buat sumber daya algoritme atau paket model diSageMaker. Untuk informasi, lihat Buat Sumber Daya Algoritma dan Package Model.

3. Mendaftar sebagai penjual AWS Marketplace dan daftar algoritme atau paket model AndaAWS Marketplace. Untuk informasi tentang mendaftar sebagai penjual, lihat Memulai sebagai Penjual di Panduan Pengguna untuk AWS Marketplace Penyedia. Untuk informasi tentang mencantumkan dan memonetisasi algoritme dan paket model Anda, lihat Mencatat Algoritma dan Paket Model di AWS Marketplace for Machine Learning di Panduan Pengguna untuk Penyedia. AWS Marketplace



Topik

- Mengembangkan Algoritma dan Model di Amazon SageMaker
- Buat Sumber Daya Algoritma dan Package Model
- Cantumkan Algoritma atau Paket Model Anda AWS Marketplace

Mengembangkan Algoritma dan Model di Amazon SageMaker

Sebelum Anda dapat membuat sumber daya algoritme dan model paket untuk digunakan di Amazon SageMaker atau daftarAWS Marketplace, Anda harus mengembangkannya dan mengemasnya dalam kontainer Docker.

> **Note**
>
> Saat algoritme dan paket model dibuat untuk dicantumkanAWS Marketplace, SageMaker memindai kontainer untuk mendapatkan kerentanan keamanan pada sistem operasi yang didukung.

Hanya versi sistem operasi berikut yang didukung:

- Debian: 6.0, 7, 8, 9, 10
- Ubuntu: 12.04, 12.10, 13.04, 14.04, 14.10, 15.04, 15.10, 16.04, 16.10, 17.04, 17.10, 18.04, 18.10
- CentOS: 5, 6, 7
- Oracle Linux: 5, 6, 7
- Alpine: 3.3, 3.4, 3.5
- Amazon Linux

Topik

Mengembangkan Algoritma di SageMaker

Algoritma harus dikemas sebagai wadah buruh pelabuhan dan disimpan di Amazon ECR untuk menggunakannya. SageMaker Container Docker berisi kode pelatihan yang digunakan untuk menjalankan pekerjaan pelatihan dan, secara opsional, kode inferensi yang digunakan untuk mendapatkan kesimpulan dari model yang dilatih dengan menggunakan algoritme.

Untuk informasi tentang mengembangkan algoritma dalam SageMaker dan kemasan mereka sebagai wadah, lihat. Gunakan kontainer Docker untuk membuat model Untuk contoh lengkap tentang cara membuat wadah algoritma, lihat contoh notebook di https://sagemaker-examples.readthedocs.io/en/latest/advanced_functionality/scikit_bring_your_own/scikit_bring_your_own.html. Anda juga dapat menemukan contoh notebook dalam contoh SageMaker notebook. Notebook ini ada di bagian Fungsionalitas Lanjutan, dan diberi nama`scikit_bring_your_own.ipynb`. Untuk informasi tentang menggunakan contoh notebook dalam contoh notebook, lihatContoh Notebook.

Selalu uji algoritma Anda secara menyeluruh sebelum Anda membuat sumber daya algoritme untuk dipublikasikan. AWS Marketplace

> **Note**
>
> Saat pembeli berlangganan produk dalam kontainer Anda, container Docker berjalan di lingkungan yang terisolasi (bebas internet). Saat Anda membuat kontainer, jangan mengandalkan melakukan panggilan keluar melalui internet. Panggilan ke AWS layanan juga tidak diperbolehkan.

Kembangkan Model di SageMaker

Model deployable di SageMaker terdiri dari kode inferensi, model artefak, peran IAM yang digunakan untuk mengakses sumber daya, dan informasi lain yang diperlukan untuk menyebarkan model di. SageMaker Artefak model adalah hasil pelatihan model dengan menggunakan algoritma pembelajaran mesin. Kode inferensi harus dikemas dalam wadah Docker dan disimpan di Amazon ECR. Anda dapat mengemas artefak model dalam wadah yang sama dengan kode inferensi, atau menyimpannya di Amazon S3.

Anda membuat model dengan menjalankan pekerjaan pelatihanSageMaker, atau dengan melatih algoritma pembelajaran mesin di luarSageMaker. Jika Anda menjalankan pekerjaan pelatihan diSageMaker, artefak model yang dihasilkan tersedia di `ModelArtifacts` lapangan dalam menanggapi panggilan ke DescribeTrainingJoboperasi. Untuk informasi tentang cara mengembangkan wadah SageMaker model, lihatGunakan kode inferensi Anda sendiri. Untuk contoh lengkap tentang cara membuat wadah model dari model yang dilatih di luarSageMaker, lihat contoh notebook di https://sagemaker-examples.readthedocs.io/en/latest/advanced_functionality/xgboost_bring_your_own_model/xgboost_bring_your_own_model.html. Anda juga dapat menemukan contoh notebook dalam contoh SageMaker notebook. Notebook ini ada di bagian Fungsionalitas Lanjutan, dan diberi nama`xgboost_bring_your_own_model.ipynb`. Untuk informasi tentang menggunakan contoh notebook dalam contoh notebook, lihatContoh Notebook.

Selalu uji model Anda secara menyeluruh sebelum membuat paket model untuk dipublikasikanAWS Marketplace.

> **Note**
>
> Saat pembeli berlangganan produk dalam kontainer Anda, container Docker berjalan di lingkungan yang terisolasi (bebas internet). Saat Anda membuat kontainer, jangan mengandalkan melakukan panggilan keluar melalui internet. Panggilan ke AWS layanan juga tidak diperbolehkan.

Cantumkan Algoritma atau Paket Model Anda AWS Marketplace

Setelah membuat dan memvalidasi algoritme atau model Anda di AmazonSageMaker, daftarkan produk Anda. AWS Marketplace Proses listing membuat produk Anda tersedia di AWS Marketplace dan SageMaker konsol.

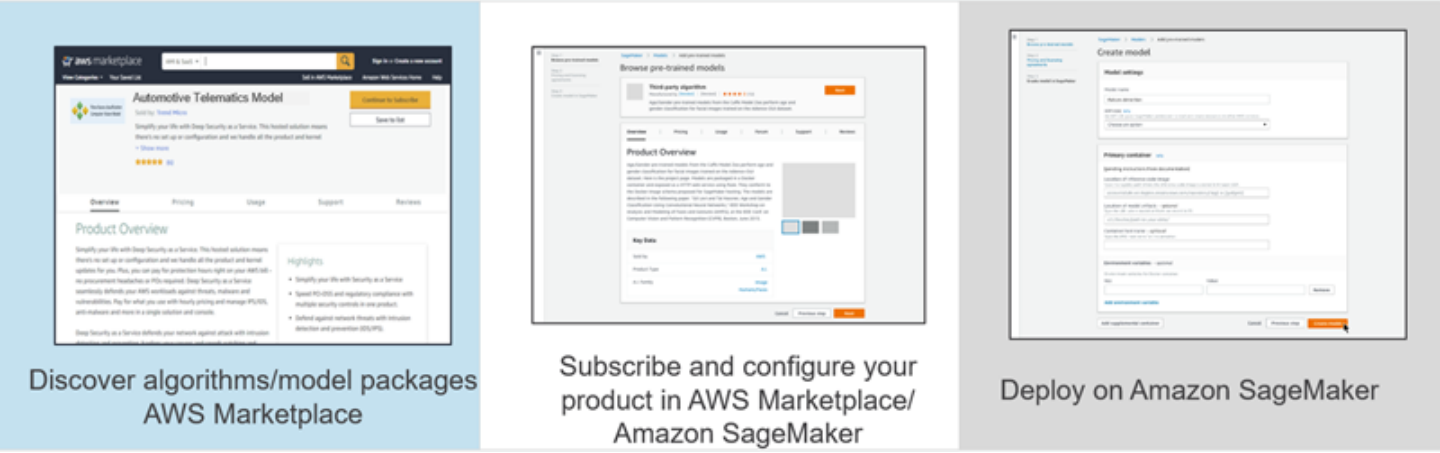
Untuk mencantumkan produkAWS Marketplace, Anda harus menjadi penjual terdaftar. Untuk mendaftar, gunakan proses pendaftaran mandiri dari Portal AWS Marketplace Manajemen (AMMP). Untuk selengkapnya, lihat Memulai sebagai Penjual di Panduan Pengguna untuk AWS Marketplace Penyedia. Saat Anda memulai proses daftar produk dari SageMaker konsol Amazon, kami memeriksa status pendaftaran penjual Anda. Jika Anda belum mendaftar, kami mengarahkan Anda untuk melakukannya.

Untuk memulai proses listing, lakukan salah satu hal berikut:

- Dari SageMaker konsol, pilih produk, pilih Tindakan, dan pilih Publikasikan daftar MLMarketplace baru. Ini membawa referensi produk Anda, Amazon Resource Name (ARN), dan mengarahkan Anda ke AMMP untuk membuat daftar.
- Buka proses listingan ML. masukkan Amazon Resource Name (ARN) secara manual, dan mulai daftar produk Anda. Proses ini membawa metadata produk yang Anda masukkan saat membuat produk. SageMaker Untuk daftar algoritme, informasinya mencakup jenis instans dan hyperparameter yang didukung. Selain itu, Anda dapat memasukkan deskripsi produk, informasi promosi, dan informasi dukungan seperti yang Anda lakukan dengan AWS Marketplace produk lain.

Temukan dan Berlangganan Algoritma dan Paket Model di AWS Marketplace

Dengan ituAWS Marketplace, Anda dapat menelusuri dan mencari ratusan algoritma dan model machine learning dalam berbagai kategori, seperti visi komputer, pemrosesan bahasa alami, pengenalan suara, teks, data, suara, gambar, analisis video, deteksi penipuan, analisis prediktif, dan banyak lagi.



Untuk menemukan algoritma AWS Marketplace

1. Buka SageMaker konsol Amazon di https://console.aws.amazon.com/sagemaker/.
2. Pilih Algoritma, lalu pilih Temukan algoritma.

   Ini membawa Anda ke AWS Marketplace halaman algoritma. Untuk informasi tentang menemukan dan berlangganan algoritmeAWS Marketplace, lihat Produk Pembelajaran Mesin di Panduan AWS Marketplace Pengguna untuk Konsumen. AWS

Untuk menemukan paket model AWS Marketplace

1. Buka SageMaker konsol di https://console.aws.amazon.com/sagemaker/.
2. Pilih Paket model, lalu pilih Temukan paket model.

   Ini akan membawa Anda ke halaman paket AWS Marketplace model. Untuk informasi tentang menemukan dan berlangganan paket modelAWS Marketplace, lihat Produk Pembelajaran Mesin di Panduan AWS Marketplace Pengguna untuk AWS Konsumen.

Gunakan Algoritma dan Paket Model

Untuk informasi tentang penggunaan algoritme dan paket model yang Anda berlangganan SageMaker, lihat. Gunakan Algoritma dan Sumber Daya Package Model

> **Note**
>
> Saat Anda membuat pekerjaan pelatihan, titik akhir inferensi, dan pekerjaan transformasi batch dari algoritme atau paket model yang Anda berlangganan AWS Marketplace, wadah

pelatihan dan inferensi tidak memiliki akses ke internet. Karena wadah tidak memiliki akses ke internet, penjual algoritme atau paket model tidak memiliki akses ke data Anda.

Memantau AWS sumber daya yang disediakan saat menggunakan Amazon SageMaker

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja SageMaker dan AWS solusi Anda yang lain. AWSmenyediakan alat pemantauan berikut untuk menonton SageMaker, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari instans EC2AWS CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- AWS CloudTrail merekam panggilan API dan peristiwa terkait yang dilakukan oleh atau atas nama akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang memanggil AWS, alamat IP sumber yang melakukan panggilan, dan kapan panggilan tersebut terjadi. Untuk mengetahui informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.
- CloudWatch Peristiwa memberikan aliran peristiwa sistem yang mendekati waktu nyata yang menggambarkan perubahan AWS sumber daya. Aturan Create CloudWatch Events bereaksi terhadap perubahan status dalam SageMaker pelatihan, penyetelan hyperparameter, atau pekerjaan transformasi batch

Topik

- Mengotomatisasi Amazon SageMaker dengan Amazon EventBridge

Pantau Amazon SageMaker dengan Amazon CloudWatch

Anda dapat memantau Amazon SageMaker menggunakan Amazon CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Namun, CloudWatch konsol Amazon membatasi pencarian ke metrik yang diperbarui dalam 2 minggu terakhir. Batasan ini memastikan bahwa pekerjaan terbaru ditampilkan di namespace Anda. Untuk membuat grafik metrik tanpa menggunakan pencarian, tentukan nama persisnya di tampilan sumber. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

SageMaker Metrik dan Dimensi

- SageMaker Metrik Pemanggilan Titik Akhir
- SageMaker Metrik Komponen Inferensi
- SageMaker Metrik Titik Akhir Multi-Model
- SageMaker Lowongan Kerja dan Metrik Titik Akhir
- SageMaker Metrik Lowongan Inference Recommender
- SageMaker Metrik Ground Truth
- Metrik Toko SageMaker Fitur Amazon
- SageMaker Metrik Pipelines

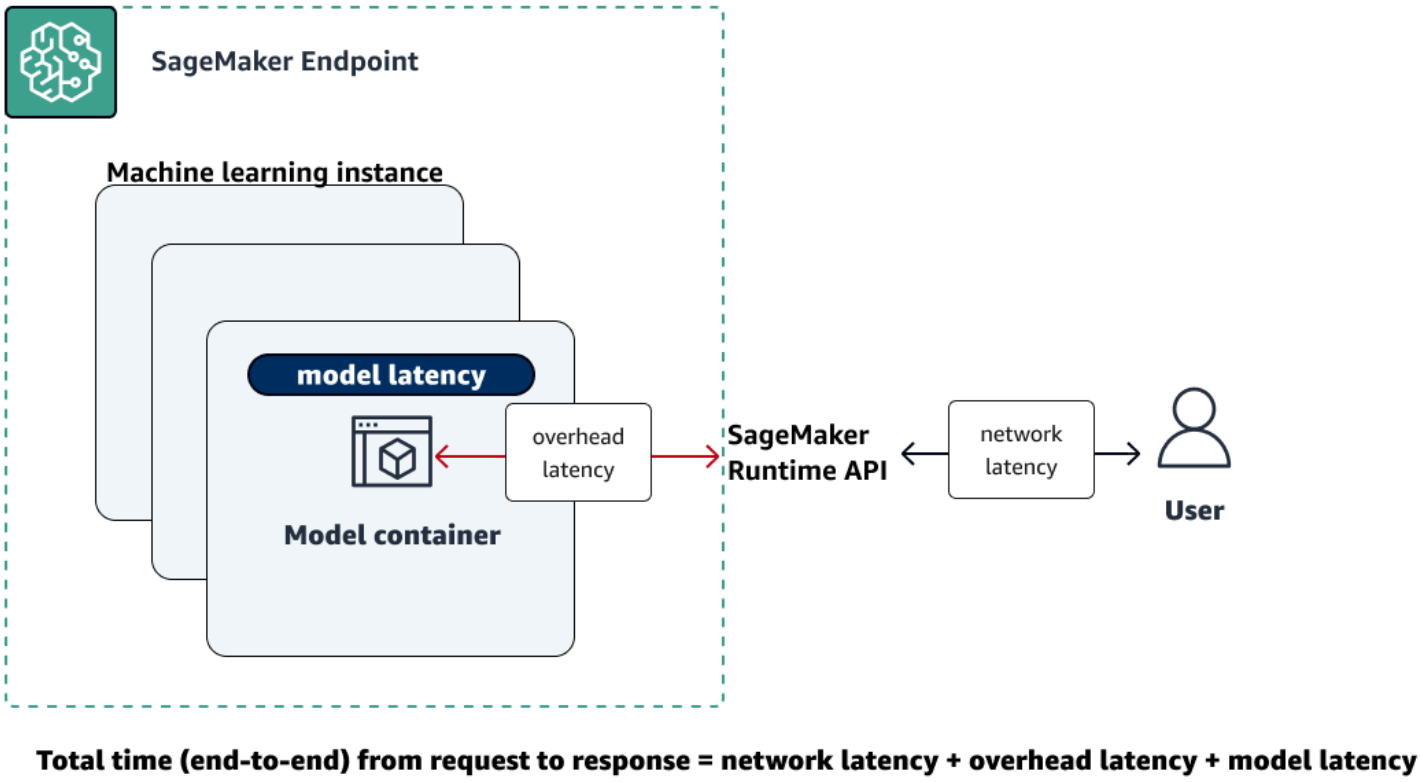
SageMaker Metrik Pemanggilan Titik Akhir

`AWS/SageMaker`Namespace menyertakan metrik permintaan berikut dari panggilan ke. InvokeEndpoint

Metrik tersedia pada frekuensi 1 menit.

Ilustrasi berikut menunjukkan bagaimana SageMaker endpoint berinteraksi dengan Amazon SageMaker Runtime API. Waktu keseluruhan antara mengirim permintaan ke titik akhir dan menerima respons tergantung pada tiga komponen berikut.

- Latensi jaringan — waktu yang diperlukan antara membuat permintaan ke SageMaker Runtime Runtime API dan menerima respons kembali dari SageMaker Runtime Runtime API.
- Latensi overhead — waktu yang diperlukan untuk mengangkut permintaan ke container model dari SageMaker Runtime Runtime API dan mengangkut respons kembali ke Runtime Runtime API. SageMaker
- Latensi model — waktu yang dibutuhkan wadah model untuk memproses permintaan dan mengembalikan respons.



Untuk informasi selengkapnya tentang latensi total, lihat Praktik terbaik untuk pengujian beban titik akhir inferensi SageMaker real-time Amazon. Untuk informasi tentang berapa lama CloudWatch metrik dipertahankan, lihat GetMetricStatisticsdi Referensi Amazon CloudWatch API.

Metrik Pemanggilan Titik Akhir

Metrik	Deskripsi
`Invocation4XXErrors`	Jumlah `InvokeEndpoint` permintaan di mana model mengembalikan kode respons HTTP 4xx. Untuk setiap respons 4xx, 1 dikirim; jika tidak, 0 dikirim. Satuan: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`Invocation5XXErrors`	Jumlah `InvokeEndpoint` permintaan di mana model mengembalikan kode respons HTTP 5xx. Untuk setiap respons 5xx, 1 dikirim; jika tidak, 0 dikirim. Satuan: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`InvocationModelErrors`	Jumlah permintaan pemanggilan model yang tidak menghasilkan respons HTTP 2XX. Ini termasuk kode status 4XX/5XX, kesalahan soket tingkat rendah, respons HTTP yang salah bentuk, dan batas waktu permintaan. Untuk setiap respons kesalahan, 1 dikirim; jika tidak, 0 dikirim. Satuan: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`Invocations`	Jumlah `InvokeEndpoint` permintaan yang dikirim ke titik akhir model. Untuk mendapatkan jumlah total permintaan yang dikirim ke titik akhir model, gunakan statistik Jumlah. Satuan: Tidak ada Statistik yang valid: Jumlah
`InvocationsPerCopy`	Jumlah pemanggilan dinormalisasi oleh setiap salinan komponen inferensi.

Metrik	Deskripsi
	Statistik yang valid: Jumlah
`InvocationsPerInstance`	Jumlah pemanggilan yang dikirim ke model, dinormalisasi oleh `InstanceCount` masing-masing ProductionVariant. 1/ `numberOfInstances` dikirim sebagai nilai pada setiap permintaan, di mana `numberOfInstances` adalah jumlah instance aktif untuk ProductionVariant di belakang titik akhir pada saat permintaan. Satuan: Tidak ada Statistik yang valid: Jumlah
`ModelLatency`	Interval waktu yang dibutuhkan oleh model untuk menanggapi permintaan SageMaker Runtime API. Interval ini mencakup waktu komunikasi lokal yang diambil untuk mengirim permintaan dan untuk mengambil respons dari wadah model dan waktu yang dibutuhkan untuk menyelesaikan inferensi dalam wadah. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`ModelSetupTime`	Waktu yang dibutuhkan untuk meluncurkan sumber daya komputasi baru untuk titik akhir tanpa server. Waktu dapat bervariasi tergantung pada ukuran model, berapa lama waktu yang dibutuhkan untuk mengunduh model, dan waktu start-up wadah. Unit: Mikrodetik Statistik yang valid: Rata-rata, Min, Maks, Jumlah Sampel, Persentil

Metrik	Deskripsi
`OverheadLatency`	Interval waktu ditambahkan ke waktu yang dibutuhkan untuk menanggapi permintaan klien dengan biaya SageMaker overhead. Interval ini diukur dari waktu SageMaker menerima permintaan sampai mengembalikan respons ke klien, dikurangi`ModelLatency` . Latensi overhead dapat bervariasi tergantung pada beberapa faktor, termasuk ukuran payload permintaan dan respons, frekuensi permintaan, dan otentikasi/otorisasi permintaan. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel

Dimensi untuk Metrik Pemanggilan Titik Akhir

Dimensi	Deskripsi
`EndpointName, VariantName`	Memfilter metrik pemanggilan titik akhir untuk titik akhir dan `ProductionVariant` varian yang ditentukan.
`InferenceComponentName`	Filter metrik pemanggilan komponen inferensi.

SageMaker Metrik Komponen Inferensi

`/aws/sagemaker/InferenceComponents`Namespace menyertakan metrik berikut dari panggilan ke titik akhir yang [InvokeEndpoint](#)menghosting komponen inferensi.

Metrik tersedia pada frekuensi 1 menit.

Metrik	Deskripsi
`CPUUtilizationNormalized`	Nilai `CPUUtilizationNormalized` metrik yang dilaporkan oleh setiap salinan komponen inferensi. Nilai berkisar antara 0% — 100%. Jika Anda menyetel `NumberOfCpuCoresRequired` parameter dalam pengaturan untuk salinan komponen inferensi, metrik menyajikan

Metrik	Deskripsi
	pemanfaatan selama reservasi. Jika tidak, metrik menyajikan pemanfaat an di atas batas.
`GPUMemory Utilizati onNormalized`	Nilai `GPUMemoryUtilizationNormalized` metrik yang dilaporkan oleh setiap salinan komponen inferensi.
`GPUUtiliz ationNorm alized`	Nilai `GPUUtilizationNormalized` metrik yang dilaporkan oleh setiap salinan komponen inferensi. Jika Anda menyetel `NumberOfA cceleratorDevicesRequired` parameter dalam pengaturan untuk salinan komponen inferensi, metrik menyajikan pemanfaatan selama reservasi. Jika tidak, metrik menyajikan pemanfaatan di atas batas.
`MemoryUti lizationN ormalized`	Nilai yang `MemoryUtilizationNormalized` dilaporkan oleh setiap salinan komponen inferensi. Jika Anda menyetel `MinMemory RequiredInMb` parameter dalam pengaturan untuk salinan komponen inferensi, metrik menyajikan pemanfaatan selama reservasi. Jika tidak, metrik menyajikan pemanfaatan di atas batas.

Dimensi untuk Metrik Komponen Inferensi

Dimensi	Deskripsi
`Inference ComponentName`	Filter metrik komponen inferensi.

SageMaker Metrik Titik Akhir Multi-Model

`AWS/SageMaker`Namespace menyertakan metrik pemuatan model berikut dari panggilan ke. InvokeEndpoint

Metrik tersedia pada frekuensi 1 menit.

Untuk informasi tentang berapa lama CloudWatch metrik dipertahankan, lihat GetMetricStatisticsdi Referensi Amazon CloudWatch API.

Metrik Pemuatan Model Titik Akhir Multi-Model

Metrik	Deskripsi
`ModelLoadingWaitTime`	Interval waktu permintaan pemanggilan telah menunggu model target diunduh, atau dimuat, atau keduanya untuk melakukan inferensi. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`ModelUnloadingTime`	Interval waktu yang diperlukan untuk membongkar model melalui panggilan `UnloadModel` API container. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`ModelDownloadingTime`	Interval waktu yang dibutuhkan untuk mengunduh model dari Amazon Simple Storage Service (Amazon S3). Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`ModelLoadingTime`	Interval waktu yang diperlukan untuk memuat model melalui panggilan `LoadModel` API container. Unit: Mikrodetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel
`ModelCacheHit`	Jumlah `InvokeEndpoint` permintaan yang dikirim ke titik akhir multi-model yang modelnya sudah dimuat. Statistik rata-rata menunjukkan rasio permintaan yang modelnya sudah dimuat. Satuan: Tidak ada

Metrik	Deskripsi
	Statistik yang valid: Rata-rata, Jumlah, Jumlah Sampel

Dimensi untuk Metrik Pemuatan Model Titik Akhir Multi-Model

Dimensi	Deskripsi
`EndpointName, VariantName`	Memfilter metrik pemanggilan titik akhir untuk titik akhir dan `ProductionVariant` varian yang ditentukan.

`/aws/sagemaker/Endpoints`Ruang nama menyertakan metrik instance berikut dari panggilan ke. InvokeEndpoint

Metrik tersedia pada frekuensi 1 menit.

Untuk informasi tentang berapa lama CloudWatch metrik dipertahankan, lihat GetMetricStatisticsdi Referensi Amazon CloudWatch API.

Metrik Instans Model Titik Akhir Multi-Model

Metrik	Deskripsi
`LoadedModelCount`	Jumlah model yang dimuat dalam wadah titik akhir multi-model. Metrik ini dipancarkan per instance. Statistik rata-rata dengan periode 1 menit memberi tahu Anda jumlah rata-rata model yang dimuat per instance. Statistik Jumlah memberi tahu Anda jumlah total model yang dimuat di semua instance di titik akhir. Model yang dilacak metrik ini belum tentu unik karena model mungkin dimuat dalam beberapa wadah di titik akhir. Satuan: Tidak ada Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel

Dimensi untuk Metrik Pemuatan Model Titik Akhir Multi-Model

Dimensi	Deskripsi
`EndpointName, VariantName`	Memfilter metrik pemanggilan titik akhir untuk titik akhir dan `ProductionVariant` varian yang ditentukan.

SageMaker Lowongan Kerja dan Metrik Titik Akhir

`/aws/sagemaker/Endpoints`Ruang nama `/aws/sagemaker/ProcessingJobs/aws/sagemaker/TrainingJobs`,`/aws/sagemaker/TransformJobs`,, dan menyertakan metrik berikut untuk pekerjaan pelatihan dan instance titik akhir.

Metrik tersedia pada frekuensi 1 menit.

> **Note**
>
> Amazon CloudWatch mendukung metrik kustom resolusi tinggi dan resolusi terbaiknya adalah 1 detik. Namun, semakin halus resolusinya, semakin pendek umur metrik. CloudWatch Untuk resolusi frekuensi 1 detik, CloudWatch metrik tersedia selama 3 jam. Untuk informasi selengkapnya tentang resolusi dan umur CloudWatch metrik, lihat GetMetricStatisticsdi Referensi Amazon CloudWatch API.

> **Tip**
>
> Jika Anda ingin membuat profil pekerjaan pelatihan Anda dengan resolusi yang lebih baik hingga perincian 100 milidetik (0,1 detik) dan menyimpan metrik pelatihan tanpa batas waktu di Amazon S3 untuk analisis khusus kapan saja, pertimbangkan untuk menggunakan Amazon Debugger. SageMaker SageMaker Debugger menyediakan aturan bawaan untuk secara otomatis mendeteksi masalah pelatihan umum; ia mendeteksi masalah pemanfaatan sumber daya perangkat keras (seperti kemacetan CPU, GPU, dan I/O) dan masalah model non-konvergen (seperti overfit, gradien menghilang, dan tensor yang meledak). SageMaker Debugger juga menyediakan visualisasi melalui Studio Classic dan laporan profilingnya. Untuk menjelajahi visualisasi Debugger, lihat Panduan Dasbor Wawasan SageMaker

Debugger, Panduan Laporan Profil Debugger, dan Menganalisis Data Menggunakan Pustaka Klien SMDebug.

Processing Job, Training Job, Batch Transform Job, dan Endpoint Instance Metrics

Metrik	Deskripsi
`CPUReservation`	Jumlah CPU yang dicadangkan oleh kontainer pada sebuah instance. Nilai berkisar antara 0% — 100%. Dalam pengaturan untuk komponen inferensi, Anda mengatur reservasi CPU dengan `NumberOfCpuCoresRequired` parameter. Misalnya, jika ada 4 CPU, dan 2 dicadangkan, `CPUReservation` metriknya adalah 50%.
`CPUUtilization`	Jumlah dari setiap pemanfaatan inti CPU individu. Pemanfaatan CPU dari setiap rentang inti adalah 0-100. Misalnya, jika ada empat CPU, `CPUUtilization` kisarannya adalah 0% - 400%. Untuk pekerjaan pemrosesan, nilainya adalah pemanfaatan CPU dari wadah pemrosesan pada instance. Untuk pekerjaan pelatihan, nilainya adalah pemanfaatan CPU dari wadah algoritma pada instance. Untuk pekerjaan transformasi batch, nilainya adalah pemanfaatan CPU dari wadah transformasi pada instance. Untuk varian endpoint, nilainya adalah jumlah dari pemanfaatan CPU dari wadah primer dan tambahan pada instance. **Note** Untuk pekerjaan multi-instance, setiap instans melaporkan metrik pemanfaatan CPU. Namun, tampilan default CloudWatch menunjukkan pemanfaatan CPU rata-rata di semua instance. Unit: Persen

Metrik	Deskripsi
`CPUUtilizationNormalized`	Jumlah normalisasi pemanfaatan masing-masing inti CPU individu. Nilai berkisar antara 0% — 100%. Misalnya, jika ada empat CPU, dan metriknya 200%, maka `CPUUtilization` `CPUUtilizationNormalized` metriknya adalah 50%.
`DiskUtilization`	Persentase ruang disk yang digunakan oleh kontainer pada sebuah instance menggunakan. Kisaran nilai ini adalah 0% - 100%. Metrik ini tidak didukung untuk pekerjaan transformasi batch. Untuk pekerjaan pemrosesan, nilainya adalah pemanfaatan ruang disk dari wadah pemrosesan pada instance. Untuk pekerjaan pelatihan, nilainya adalah pemanfaatan ruang disk dari wadah algoritma pada instance. Untuk varian endpoint, nilainya adalah jumlah dari pemanfaatan ruang disk dari wadah primer dan tambahan pada instance. Unit: Persen **Note** Untuk pekerjaan multi-instance, setiap instance melaporkan metrik pemanfaatan disk. Namun, tampilan default CloudWatch menunjukkan pemanfaatan disk rata-rata di semua instance.

Metrik	Deskripsi
`GPUMemory Utilization`	Persentase memori GPU yang digunakan oleh kontainer pada sebuah instance. Kisaran nilai adalah 0-100 dan dikalikan dengan jumlah GPU. Misalnya, jika ada empat GPU, `GPUMemoryUtilization` kisarannya adalah 0% - 400%. Untuk pekerjaan pemrosesan, nilainya adalah pemanfaatan memori GPU dari wadah pemrosesan pada instance. Untuk pekerjaan pelatihan, nilainya adalah pemanfaatan memori GPU dari wadah algoritma pada instance. Untuk pekerjaan transformasi batch, nilainya adalah pemanfaatan memori GPU dari wadah transformasi pada instance. Untuk varian titik akhir, nilainya adalah jumlah dari pemanfaatan memori GPU dari wadah primer dan tambahan pada instance. **Note** Untuk pekerjaan multi-instance, setiap instance melaporkan metrik pemanfaatan memori GPU. Namun, tampilan default CloudWatch menunjukkan pemanfaatan memori GPU rata-rata di semua instance. Unit: Persen
`GPUMemory Utilizati onNormalized`	Persentase memori GPU yang dinormalisasi yang digunakan oleh kontainer pada sebuah instance. Nilai berkisar antara 0% — 100%. Misalnya, jika ada empat GPU, dan metriknya 200%, maka `GPUMemory Utilization` `GPUMemoryUtilizationNormalized` metriknya adalah 50%.

Metrik	Deskripsi
GPUReservation	Jumlah GPU yang dicadangkan oleh kontainer pada sebuah instance. Nilai berkisar antara 0% — 100%. Dalam pengaturan untuk komponen inferensi, Anda mengatur reservasi GPU dengan. NumberOfAcceleratorDevicesRequired Misalnya, jika ada 4 GPU dan 2 yang dicadangkan, GPUReservation metriknya adalah 50%.
GPUUtilization	Persentase unit GPU yang digunakan oleh kontainer pada sebuah instance. Nilai dapat berkisar antara rentang adalah 0-100 dan dikalikan dengan jumlah GPU. Misalnya, jika ada empat GPU, GPUUtilization kisarannya adalah 0% - 400%. Untuk pekerjaan pemrosesan, nilainya adalah pemanfaatan GPU dari wadah pemrosesan pada instance. Untuk pekerjaan pelatihan, nilainya adalah pemanfaatan GPU dari wadah algoritma pada instance. Untuk pekerjaan transformasi batch, nilainya adalah pemanfaatan GPU dari wadah transformasi pada instance. Untuk varian endpoint, nilainya adalah jumlah dari pemanfaatan GPU dari wadah primer dan tambahan pada instance. **Note** Untuk pekerjaan multi-instance, setiap instans melaporkan metrik pemanfaatan GPU. Namun, tampilan default CloudWatch menunjukkan pemanfaatan GPU rata-rata di semua instance. Unit: Persen
GPUUtilizationNormalized	Persentase unit GPU yang dinormalisasi yang digunakan oleh kontainer pada sebuah instance. Nilai berkisar antara 0% — 100%. Misalnya, jika ada empat GPU, dan metriknya 200%, maka GPUUtilization GPUUtilizationNormalized metriknya adalah 50%.

Metrik	Deskripsi
`MemoryReservation`	Jumlah memori yang dicadangkan oleh kontainer pada sebuah instance. Nilai berkisar antara 0% — 100%. Dalam pengaturan untuk komponen inferensi, Anda mengatur reservasi memori dengan `MinMemoryRequiredInMb` parameter. Misalnya, jika instance 32 GiB mencadangkan 1024 MB, `MemoryReservation` metriknya adalah 29,8%.
`MemoryUtilization`	Persentase memori yang digunakan oleh kontainer pada sebuah instance. Kisaran nilai ini adalah 0% - 100%. Untuk pekerjaan pemrosesan, nilainya adalah pemanfaatan memori dari wadah pemrosesan pada instance. Untuk pekerjaan pelatihan, nilainya adalah pemanfaatan memori dari wadah algoritma pada instance. Untuk pekerjaan transformasi batch, nilainya adalah pemanfaatan memori dari wadah transformasi pada instance. Untuk varian titik akhir, nilainya adalah jumlah dari pemanfaatan memori wadah primer dan tambahan pada instance. Unit: Persen **Note** Untuk pekerjaan multi-instance, setiap instance melaporkan metrik pemanfaatan memori. Namun, tampilan default CloudWatch menunjukkan pemanfaatan memori rata-rata di semua instance.

Dimensi untuk Metrik Processing Job, Training Job, dan Batch Transform Job Instance

Dimensi	Deskripsi
`Host`	Untuk memproses pekerjaan, nilai untuk dimensi ini memiliki format`[processing-job-name]/algo-[instance-number-in-cluster]` . Gunakan dimensi ini untuk memfilter metrik instance untuk pekerjaan pemrosesan dan instance yang ditentukan. Format dimensi ini hanya ada di `/aws/sagemaker/ProcessingJobs` namespace. Untuk pekerjaan pelatihan, nilai untuk dimensi ini memiliki format`[training-job-name]/algo-[instance-number-in-cluster]` . Gunakan dimensi ini untuk memfilter metrik instance untuk pekerjaan dan instance pelatihan yang ditentukan. Format dimensi ini hanya ada di `/aws/sagemaker/TrainingJobs` namespace. Untuk pekerjaan transformasi batch, nilai untuk dimensi ini memiliki format`[transform-job-name]/[instance-id]` . Gunakan dimensi ini untuk memfilter metrik instance untuk pekerjaan dan instance transformasi batch yang ditentukan. Format dimensi ini hanya ada di `/aws/sagemaker/TransformJobs` namespace.

SageMaker Metrik Lowongan Inference Recommender

`/aws/sagemaker/InferenceRecommendationsJobs`Namespace menyertakan metrik berikut untuk pekerjaan rekomendasi inferensi.

Metrik Rekomendasi Inferensi

Metrik	Deskripsi
`ClientInvocations`	Jumlah `InvokeEndpoint` permintaan yang dikirim ke titik akhir model, seperti yang diamati oleh Inference Recommender. Satuan: Tidak ada Statistik yang valid: Jumlah

Metrik	Deskripsi
`ClientInvocationErrors`	Jumlah `InvokeEndpoint` permintaan yang gagal, seperti yang diamati oleh Inference Recommender. Satuan: Tidak ada Statistik yang valid: Jumlah
`ClientLatency`	Interval waktu yang dibutuhkan antara mengirim `InvokeEndpoint` panggilan dan menerima respons seperti yang diamati oleh Inference Recommender. Perhatikan bahwa waktunya dalam milidetik, sedangkan metrik pemanggilan `ModelLatency` titik akhir dalam mikrodetik. Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel, Persentil
`NumberOfUsers`	Jumlah pengguna bersamaan yang mengirim `InvokeEndpoint` permintaan ke titik akhir model. Satuan: Tidak ada Statistik yang valid: Maks, Min, Rata-rata

Dimensi untuk Metrik Pekerjaan Inference Recommender

Dimensi	Deskripsi
`JobName`	Filter metrik pekerjaan Inference Recommender untuk pekerjaan Inference Recommender yang ditentukan.
`EndpointName`	Filter metrik pekerjaan Inference Recommender untuk titik akhir yang ditentukan.

SageMaker Metrik Ground Truth

Metrik Ground Truth

Metrik	Deskripsi
`ActiveWorkers`	Seorang pekerja aktif tunggal di tim kerja pribadi mengajukan, melepaskan, atau menolak tugas. Untuk mendapatkan jumlah total pekerja aktif, gunakan statistik Jumlah. Ground Truth mencoba untuk menyampaikan setiap `ActiveWorkers` peristiwa individu satu kali. Jika pengiriman ini tidak berhasil, metrik ini mungkin tidak melaporkan jumlah total pekerja aktif Satuan: Tidak ada Statistik yang valid: Jumlah, Jumlah Sampel
`DatasetObjectsAutoAnnotated`	Jumlah objek kumpulan data yang dianotasi secara otomatis dalam pekerjaan pelabelan. Metrik ini hanya dipancarkan saat pelabelan otomatis diaktifkan. Untuk melihat kemajuan pekerjaan pelabelan, gunakan metrik Max. Satuan: Tidak ada Statistik yang valid: Maks
`DatasetObjectsHumanAnnotated`	Jumlah objek dataset yang dianotasi oleh manusia dalam pekerjaan pelabelan. Untuk melihat kemajuan pekerjaan pelabelan, gunakan metrik Max. Satuan: Tidak ada Statistik yang valid: Maks
`DatasetObjectsLabelingFailed`	Jumlah objek dataset yang gagal diberi label dalam pekerjaan pelabelan. Untuk melihat kemajuan pekerjaan pelabelan, gunakan metrik Max. Satuan: Tidak ada Statistik yang valid: Maks

Metrik	Deskripsi
`JobsFailed`	Satu pekerjaan pelabelan gagal. Untuk mendapatkan jumlah total pekerjaan pelabelan yang gagal, gunakan statistik Jumlah. Satuan: Tidak ada Statistik yang valid: Jumlah, Jumlah Sampel
`JobsSucceeded`	Pekerjaan pelabelan tunggal berhasil. Untuk mendapatkan jumlah total pekerjaan pelabelan yang berhasil, gunakan statistik Sum. Satuan: Tidak ada Statistik yang valid: Jumlah, Jumlah Sampel
`JobsStopped`	Satu pekerjaan pelabelan dihentikan. Untuk mendapatkan jumlah total pekerjaan pelabelan yang dihentikan, gunakan statistik Sum. Satuan: Tidak ada Statistik yang valid: Jumlah, Jumlah Sampel
`TasksAccepted`	Satu tugas diterima oleh seorang pekerja. Untuk mendapatkan jumlah total tugas yang diterima oleh pekerja, gunakan statistik Jumlah. Ground Truth mencoba untuk menyampaikan setiap `TaskAccepted` peristiwa individu satu kali. Jika pengiriman ini tidak berhasil, metrik ini mungkin tidak melaporkan jumlah total tugas yang diterima. Satuan: Tidak ada Statistik yang valid: Jumlah, Jumlah Sampel

Metrik	Deskripsi
`TasksDeclined`	Satu tugas ditolak oleh seorang pekerja. Untuk mendapatkan jumlah total tugas yang ditolak oleh pekerja, gunakan statistik Jumlah. Ground Truth mencoba untuk menyampaikan setiap `TasksDeclined` peristiwa individu satu kali. Jika pengiriman ini tidak berhasil, metrik ini mungkin tidak melaporkan jumlah total tugas yang ditolak. Satuan: Tidak ada Statistik yang Valid: Jumlah, Jumlah Sampel
`TasksReturned`	Satu tugas dikembalikan. Untuk mendapatkan jumlah total tugas yang dikembalikan, gunakan statistik Jumlah. Ground Truth mencoba untuk menyampaikan setiap `TasksReturned` peristiwa individu satu kali. Jika pengiriman ini tidak berhasil, metrik ini mungkin tidak melaporkan jumlah total tugas yang dikembalikan. Satuan: Tidak ada Statistik yang valid: Jumlah, Jumlah Sampel
`TasksSubmitted`	Satu tugas diserahkan/diselesaikan oleh pekerja swasta. Untuk mendapatkan jumlah total tugas yang diajukan oleh pekerja, gunakan statistik Jumlah. Ground Truth mencoba untuk menyampaikan setiap `TasksSubmitted` peristiwa individu satu kali. Jika pengiriman ini tidak berhasil, metrik ini mungkin tidak melaporkan jumlah total tugas yang dikirimkan. Satuan: Tidak ada Statistik yang valid: Jumlah, Jumlah Sampel

Metrik	Deskripsi
`TimeSpent`	Waktu yang dihabiskan untuk tugas yang diselesaikan oleh pekerja swasta. Metrik ini tidak termasuk waktu ketika seorang pekerja berhenti atau beristirahat. Ground Truth mencoba untuk menyampaikan setiap `TimeSpent` acara satu kali. Jika pengiriman ini tidak berhasil, metrik ini mungkin tidak melaporkan jumlah total waktu yang dihabiskan. Unit: detik Statistik yang valid: Jumlah, Jumlah Sampel
`TotalDataSetObjectsLabeled`	Jumlah objek dataset yang berhasil dilabeli dalam pekerjaan pelabelan. Untuk melihat kemajuan pekerjaan pelabelan, gunakan metrik Max. Satuan: Tidak ada Statistik yang valid: Maks

Dimensi untuk Metrik Objek Dataset

Dimensi	Deskripsi
`LabelingJobName`	Memfilter metrik jumlah objek kumpulan data untuk pekerjaan pelabelan.

Metrik Toko SageMaker Fitur Amazon

Metrik Konsumsi Toko Fitur

Metrik	Deskripsi
`ConsumedReadRequestsUnits`	Jumlah unit baca yang dikonsumsi selama periode waktu yang ditentukan. Anda dapat mengambil unit baca yang dikonsumsi untuk operasi runtime feature store dan grup fitur yang sesuai. Satuan: Tidak ada

Metrik	Deskripsi
	Statistik yang valid: Semua
`ConsumedWriteRequestsUnits`	Jumlah unit tulis yang dikonsumsi selama periode waktu yang ditentukan. Anda dapat mengambil unit tulis yang dikonsumsi untuk operasi runtime feature store dan grup fitur yang sesuai. Satuan: Tidak ada Statistik yang valid: Semua
`ConsumedReadCapacityUnits`	Jumlah unit kapasitas baca yang disediakan yang dikonsumsi selama periode waktu yang ditentukan. Anda dapat mengambil unit kapasitas baca yang dikonsumsi untuk operasi runtime feature store dan grup fitur yang sesuai. Satuan: Tidak ada Statistik yang valid: Semua
`ConsumedWriteCapacityUnits`	Jumlah unit kapasitas tulis yang disediakan yang dikonsumsi selama periode waktu yang ditentukan. Anda dapat mengambil unit kapasitas tulis yang dikonsumsi untuk operasi runtime feature store dan grup fitur yang sesuai. Satuan: Tidak ada Statistik yang valid: Semua

Dimensi untuk Metrik Konsumsi Toko Fitur

Dimensi	Deskripsi
`FeatureGroupName` , `OperationName`	Memfilter metrik konsumsi runtime feature store dari grup fitur dan operasi yang telah Anda tentukan.

Metrik Operasional Toko Fitur

Metrik	Deskripsi
`Invocations`	Jumlah permintaan yang dibuat untuk operasi runtime feature store selama periode waktu yang ditentukan. Satuan: Tidak ada Statistik yang valid: Jumlah
`Operation 4XXErrors`	Jumlah permintaan yang dibuat untuk operasi runtime Feature Store di mana operasi mengembalikan kode respons HTTP 4xx. Untuk setiap respons 4xx, 1 dikirim; jika tidak, 0 dikirim. Satuan: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`Operation 5XXErrors`	Jumlah permintaan yang dibuat untuk operasi runtime feature store dimana operasi mengembalikan kode respons HTTP 5xx. Untuk setiap respons 5xx, 1 dikirim; jika tidak, 0 dikirim. Satuan: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`Throttled Requests`	Jumlah permintaan yang dibuat untuk operasi runtime feature store tempat permintaan dibatasi. Untuk setiap permintaan yang dibatasi, 1 dikirim; jika tidak, 0 dikirim. Satuan: Tidak ada Statistik yang valid: Rata-rata, Jumlah
`Latency`	Interval waktu untuk memproses permintaan yang dibuat ke operasi runtime Feature Store. Interval ini diukur dari waktu SageMaker menerima permintaan hingga mengembalikan respons ke klien. Unit: Mikrodetik

Metrik	Deskripsi
	Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel, Persentil

Dimensi untuk Metrik Operasional Toko Fitur

Dimensi	Deskripsi
`FeatureGroupName` , `OperationName`	Memfilter metrik operasional runtime feature store dari grup fitur dan operasi yang telah Anda tentukan. Anda dapat menggunakan dimensi ini untuk operasi non batch, seperti GetRecord, PutRecord, dan DeleteRecord.
`OperationName`	Memfilter metrik operasional runtime feature store untuk operasi yang telah Anda tentukan. Anda dapat menggunakan dimensi ini untuk operasi batch seperti BatchGetRecord.

SageMaker Metrik Pipelines

`AWS/Sagemaker/ModelBuildingPipeline`Namespace menyertakan metrik berikut untuk eksekusi pipeline.

Dua kategori metrik eksekusi Pipelines tersedia:

- Metrik Eksekusi di Semua Pipelines — Metrik eksekusi pipeline level akun (untuk semua pipeline di akun saat ini)
- Metrik Eksekusi berdasarkan Pipeline - Metrik eksekusi pipa per pipeline

Metrik tersedia pada frekuensi 1 menit.

Metrik Eksekusi Pipelines

Metrik	Deskripsi
`Execution Started`	Jumlah eksekusi pipa yang dimulai.

Metrik	Deskripsi
	Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`ExecutionFailed`	Jumlah eksekusi pipa yang gagal. Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`Execution Succeeded`	Jumlah eksekusi pipa yang berhasil. Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`Execution Stopped`	Jumlah eksekusi pipa yang berhenti. Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`Execution Duration`	Durasi dalam milidetik eksekusi pipeline berjalan. Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel

Dimensi untuk Metrik Eksekusi berdasarkan Pipeline

Dimensi	Deskripsi
`PipelineName`	Memfilter metrik eksekusi pipeline untuk pipeline tertentu.

Metrik Langkah Pipelines

`AWS/Sagemaker/ModelBuildingPipeline`Namespace menyertakan metrik berikut untuk langkah-langkah pipeline.

Metrik tersedia pada frekuensi 1 menit.

Metrik	Deskripsi
`StepStarted`	Jumlah langkah yang dimulai. Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`StepFailed`	Jumlah langkah yang gagal. Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`StepSucceeded`	Jumlah langkah yang berhasil. Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`StepStopped`	Jumlah langkah yang berhenti. Unit: Hitungan Statistik yang valid: Rata-rata, Jumlah
`StepDuration`	Durasi dalam milidetik langkah berjalan. Unit: Milidetik Statistik yang valid: Rata-rata, Jumlah, Min, Maks, Jumlah Sampel

Dimensi untuk Metrik Langkah Pipelines

Dimensi	Deskripsi
`PipelineName` , `StepName`	Memfilter metrik langkah untuk pipeline dan langkah tertentu.

Log SageMaker Acara Amazon dengan Amazon CloudWatch

Untuk membantu Anda men-debug pekerjaan kompilasi, memproses pekerjaan, pekerjaan pelatihan, titik akhir, mengubah pekerjaan, instance notebook, dan konfigurasi siklus hidup instance notebook, apa pun yang dikirim oleh container algoritme, wadah model, atau konfigurasi siklus hidup instance notebook ke atau juga dikirim ke Amazon Logs. `stdout` `stderr` CloudWatch Selain debugging, Anda dapat menggunakan ini untuk analisis kemajuan.

Log

Tabel berikut mencantumkan semua log yang disediakan oleh Amazon SageMaker.

Log

Catat Nama Grup	Nama Aliran Log
`/aws/sagemaker/CompilationJobs`	`[compilation-job-name]`
`/aws/sagemaker/Endpoints/[EndpointName]`	`[production-variant-name]/[instance-id]`
	(Untuk titik akhir Inferensi Asinkron) `[production-variant-name]/[instance-id]/data-log`
	(Untuk Pipa Inferensi) `[production-variant-name]/[instance-id]/[container-name provided in SageMaker model]`
`/aws/sagemaker/groundtruth/WorkerActivity`	`aws/sagemaker/groundtruth/worker-activity/[requester-AWS-Id]-[region]/[timestamp]`
`/aws/sagemaker/InferenceRecommendationsJobs`	`[inference-recommendations-job-name]/execution`
	`[inference-recommendations-job-name]/CompilationJob/[compilation-job-name]`
	`[inference-recommendations-job-name]/Endpoint/[endpoint-name]`

Catat Nama Grup	Nama Aliran Log
/aws/sagemaker/LabelingJobs	[labeling-job-name]
/aws/sagemaker/NotebookInstances	[notebook-instance-name]/[LifecycleConfigHook]
	[notebook-instance-name]/jupyter.log
/aws/sagemaker/ProcessingJobs	[processing-job-name]/[hostname]-[epoch_timestamp]
/aws/sagemaker/studio	[domain-id]/[user-profile-name]/[app-type]/[app-name]
	[domain-id]/domain-shared/rstudioserverpro/default
/aws/sagemaker/TrainingJobs	[training-job-name]/algo-[instance-number-in-cluster]-[epoch_timestamp]
/aws/sagemaker/TransformJobs	[transform-job-name]/[instance-id]-[epoch_timestamp]
	[transform-job-name]/[instance-id]-[epoch_timestamp]/data-log
	[transform-job-name]/[instance-id]-[epoch_timestamp]/[container-name provided in SageMaker model] (For Inference Pipelines)

Note

1. Aliran /aws/sagemaker/NotebookInstances/[LifecycleConfigHook] log dibuat saat Anda membuat instance notebook dengan konfigurasi siklus hidup. Untuk informasi selengkapnya, lihat Menyesuaikan Instance Notebook Menggunakan Skrip Konfigurasi Siklus Hidup.

2. Untuk Inference Pipelines, jika Anda tidak memberikan nama kontainer, platform menggunakan **container-1, container-2**, dan seterusnya, sesuai dengan urutan yang disediakan dalam model. SageMaker

Untuk informasi selengkapnya tentang peristiwa logging dengan CloudWatch logging, lihat Apa itu CloudWatch Log Amazon? di Panduan CloudWatch Pengguna Amazon.

Log Panggilan SageMaker API Amazon dengan AWS CloudTrail

Amazon SageMaker terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di SageMaker. CloudTrail menangkap semua panggilan API untuk SageMaker, dengan pengecualian InvokeEndpointdan InvokeEndpointAsync, sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari SageMaker konsol dan panggilan kode ke operasi SageMaker API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk. SageMaker Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat SageMaker, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Secara default, data log disimpan di CloudWatch Log tanpa batas waktu. Namun, Anda dapat mengonfigurasi berapa lama data log disimpan dalam grup log. Untuk selengkapnya, lihat Mengubah Penyimpanan Data CloudWatch Log di Log di Panduan Pengguna CloudWatch Log Amazon.

SageMaker Informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di Amazon SageMaker, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Amazon SageMaker, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan

file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat membuat konfigurasi layanan AWS lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di log CloudTrail . Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

Semua SageMaker tindakan, dengan pengecualian InvokeEndpointdan InvokeEndpointAsync, dicatat oleh CloudTrail dan didokumentasikan dalam file `Operations`. Misalnya, panggilan ke`CreateTrainingJob`, `CreateEndpoint` dan `CreateNotebookInstance` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap peristiwa atau entri log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut:

- Apakah permintaan dibuat dengan kredensial akar atau pengguna IAM.
- Apakah permintaan dibuat dengan kredensial keamanan sementara untuk suatu peran atau pengguna gabungan.
- Apakah permintaan dibuat oleh layanan AWS lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

Operasi Dilakukan dengan Penyetelan Model Otomatis

SageMaker mendukung pencatatan peristiwa layanan non-API ke file CloudTrail log Anda untuk pekerjaan penyetelan model otomatis. Peristiwa ini terkait dengan pekerjaan penyetelan Anda, tetapi bukan merupakan hasil langsung dari permintaan pelanggan ke AWS API publik. Misalnya, saat Anda membuat pekerjaan penyetelan hiperparameter dengan menelepon `CreateHyperParameterTuningJob`, SageMaker buat pekerjaan pelatihan untuk mengevaluasi berbagai kombinasi hiperparameter untuk menemukan hasil terbaik. Demikian pula, ketika Anda menelepon `StopHyperParameterTuningJob`untuk menghentikan pekerjaan penyetelan hiperparameter, SageMaker mungkin menghentikan salah satu pekerjaan pelatihan lari terkait. Peristiwa non-API untuk pekerjaan tuning Anda dicatat CloudTrail untuk membantu Anda meningkatkan tata kelola, kepatuhan, dan audit operasional dan risiko akun Anda. AWS

Entri log yang dihasilkan dari peristiwa layanan non-API memiliki sebagai `eventType` `AwsServiceEvent` gantinya. `AwsApiCall`

Memahami Entri File SageMaker Log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Berikut contoh entri log untuk `CreateEndpoint` tindakan, yang menciptakan titik akhir untuk menerapkan model terlatih.

```
{
    "eventVersion":"1.05",
    "userIdentity": {
        "type":"IAMUser",
        "principalId":"AIXDAYQEXAMPLEUMLYNGL",
        "arn":"arn:aws:iam::123456789012:user/intern",
        "accountId":"123456789012",
        "accessKeyId":"ASXIAGXEXAMPLEQULKNXV",
        "userName":"intern"
    },
    "eventTime":"2018-01-02T13:39:06Z",
    "eventSource":"sagemaker.amazonaws.com",
    "eventName":"CreateEndpoint",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"USER_AGENT",
    "requestParameters": {
        "endpointName":"ExampleEndpoint",
        "endpointConfigName":"ExampleEndpointConfig"
    },
    "responseElements": {
        "endpointArn":"arn:aws:sagemaker:us-west-2:123456789012:endpoint/
exampleendpoint"
    },
    "requestID":"6b1b42b9-EXAMPLE",
    "eventID":"a6f85b21-EXAMPLE",
    "eventType":"AwsApiCall",
    "recipientAccountId":"444455556666"
```

```
}
```

Contoh berikut adalah entri log untuk `CreateModel` tindakan, yang membuat satu atau lebih kontainer untuk meng-host model yang dilatih sebelumnya.

```
{
    "eventVersion":"1.05",
    "userIdentity": {
        "type":"IAMUser",
        "principalId":"AIXDAYQEXAMPLEUMLYNGL",
        "arn":"arn:aws:iam::123456789012:user/intern",
        "accountId":"123456789012",
        "accessKeyId":"ASXIAGXEXAMPLEQULKNXV",
        "userName":"intern"
    },
    "eventTime":"2018-01-02T15:23:46Z",
    "eventSource":"sagemaker.amazonaws.com",
    "eventName":"CreateModel",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"USER_AGENT",
    "requestParameters": {
        "modelName":"ExampleModel",
        "primaryContainer": {
            "image":"174872318107.dkr.ecr.us-west-2.amazonaws.com/kmeans:latest"
        },
        "executionRoleArn":"arn:aws:iam::123456789012:role/EXAMPLEARN"
    },
    "responseElements": {
        "modelArn":"arn:aws:sagemaker:us-west-2:123456789012:model/
barkinghappy2018-01-02t15-23-32-275z-ivrdog"
    },
    "requestID":"417b8dab-EXAMPLE",
    "eventID":"0f2b3e81-EXAMPLE",
    "eventType":"AwsApiCall",
    "recipientAccountId":"444455556666"
}
```

Memantau akses sumber daya pengguna dari Amazon SageMaker Studio Classic

Dengan Amazon SageMaker Studio Classic, Anda dapat memantau akses sumber daya pengguna. Untuk melihat aktivitas akses sumber daya, Anda dapat mengonfigurasi AWS CloudTrail untuk memantau dan merekam aktivitas pengguna dengan mengikuti langkah-langkah di Log Amazon SageMaker API Calls dengan AWS CloudTrail.

Namun, AWS CloudTrail log untuk akses sumber daya hanya mencantumkan peran IAM eksekusi Studio Classic sebagai pengenal. Tingkat logging ini cukup untuk mengaudit aktivitas pengguna ketika setiap profil pengguna memiliki peran eksekusi yang berbeda. Namun, ketika peran IAM eksekusi tunggal dibagi antara beberapa profil pengguna, Anda tidak bisa mendapatkan informasi tentang pengguna tertentu yang mengakses AWS sumber daya.

Anda bisa mendapatkan informasi tentang pengguna tertentu yang melakukan tindakan dalam AWS CloudTrail log saat menggunakan peran eksekusi bersama, menggunakan `sourceIdentity` konfigurasi untuk menyebarkan nama profil pengguna Studio Classic. Untuk informasi selengkapnya tentang identitas sumber, lihat Memantau dan mengontrol tindakan yang diambil dengan peran yang diasumsikan.

Prasyarat

- Instal dan konfigurasikan langkah-langkah AWS Command Line Interface berikut dalam Menginstal atau memperbarui versi terbaru AWS CLI.
- Pastikan pengguna Studio Classic di domain Anda tidak memiliki kebijakan yang memungkinkan mereka memperbarui atau memodifikasi domain.
- Untuk mengaktifkan atau menonaktifkan `sourceIdentity` propagasi, semua aplikasi di domain harus dalam `Deleted` status `Stopped` atau. Untuk informasi selengkapnya tentang cara menghentikan dan mematikan aplikasi, lihat Menutup dan Memperbarui Aplikasi Studio Classic.
- Semua peran eksekusi harus memiliki izin kebijakan kepercayaan berikut:
  - Setiap peran yang diasumsikan oleh peran eksekusi domain harus memiliki `sts:SetSourceIdentity` izin dalam kebijakan kepercayaan sebagai berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ]
        }
    ]
}
```

- Ketika Anda mengambil peran dengan peran lain, yang disebut rantai peran, lakukan hal berikut:
  - Izin untuk `sts:SetSourceIdentity` diperlukan baik dalam kebijakan izin prinsipal yang mengambil peran, dan dalam kebijakan kepercayaan peran peran target. Jika tidak, peran operasi asumsi akan gagal.
  - Rantai peran ini dapat terjadi di Studio Classic atau layanan hilir lainnya, seperti Amazon EMR. Untuk informasi selengkapnya tentang rantai peran, lihat Istilah dan konsep peran.

Pertimbangan saat menggunakan **`sourceIdentity`**

Saat Anda melakukan panggilan AWS API dari notebook Studio Classic, SageMaker Canvas, atau Amazon SageMaker Data Wrangler, panggilan hanya akan direkam CloudTrail jika panggilan tersebut dilakukan menggunakan sesi peran eksekusi Studio Classic atau peran berantai apa pun dari sesi tersebut. `sourceIdentity`

Ketika panggilan API ini memanggil layanan lain untuk melakukan operasi tambahan, `sourceIdentity` logging bergantung pada implementasi spesifik dari layanan yang dipanggil.

- Amazon SageMaker Training, Processing, dan Pipeline: Saat Anda membuat lowongan kerja menggunakan fitur-fitur ini, API pembuatan lowongan kerja tidak dapat memanfaatkan `sourceIdentity` yang ada di sesi tersebut. Akibatnya, panggilan AWS API apa pun yang dilakukan dari pekerjaan ini tidak dicatat `sourceIdentity` dalam CloudTrail log.
- Amazon EMR: Saat menghubungkan ke Amazon EMR dari Studio Classic menggunakan peran runtime, administrator harus secara eksplisit menyetel bidang. PropagateSourceIdentity Ini memastikan bahwa Amazon EMR menerapkan `sourceIdentity` dari kredensi panggilan ke sesi pekerjaan atau kueri. Kemudian `sourceIdentity` dicatat dalam CloudTrail log.

> **Note**
>
> Pengecualian berikut berlaku saat menggunakan`sourceIdentity`.
>
> - AWSPanggilan API yang dibuat dari spasi SageMaker bersama tidak merekam `sourceIdentity` dalam CloudTrail log.
> - Jika panggilan AWS API dibuat dari sesi yang dibuat oleh pengguna atau layanan lain dan sesi tidak didasarkan pada sesi peran eksekusi Studio Classic, maka panggilan tersebut `sourceIdentity` tidak direkam dalam CloudTrail log.

Mengaktifkan **`sourceIdentity`**

Kemampuan untuk menyebarkan nama profil pengguna seperti `sourceIdentity` di Studio Classic dimatikan secara default.

Untuk mengaktifkan kemampuan untuk menyebarkan nama profil pengguna sebagai`sourceIdentity`, gunakan AWS CLI selama pembuatan domain dan pembaruan domain. Fitur ini diaktifkan di tingkat domain dan bukan pada tingkat profil pengguna.

Setelah Anda mengaktifkan konfigurasi ini, administrator dapat melihat profil pengguna di AWS CloudTrail log untuk layanan yang diakses. Profil pengguna diberikan sebagai `sourceIdentity` nilai di `userIdentity` bagian. Untuk informasi selengkapnya tentang menggunakan AWS CloudTrail log dengan SageMaker, lihat Log Panggilan SageMaker API Amazon dengan AWS CloudTrail.

Anda dapat menggunakan kode berikut untuk mengaktifkan propagasi nama profil pengguna sebagai `sourceIdentity` selama pembuatan domain menggunakan `create-domain` API.

```
create-domain
--domain-name <value>
--auth-mode <value>
--default-user-settings <value>
--subnet-ids <value>
--vpc-id <value>
[--tags <value>]
[--app-network-access-type <value>]
[--home-efs-file-system-kms-key-id <value>]
```

```
[--kms-key-id <value>]
[--app-security-group-management <value>]
[--domain-settings "ExecutionRoleIdentityConfig=USER_PROFILE_NAME"]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Anda dapat mengaktifkan propagasi nama profil pengguna sebagai `sourceIdentity` selama pembaruan domain menggunakan `update-domain` API.

Untuk memperbarui konfigurasi ini, semua aplikasi di domain harus dalam `Deleted` status `Stopped` atau. Untuk informasi selengkapnya tentang cara menghentikan dan mematikan aplikasi, lihat Menutup dan Memperbarui Aplikasi Studio Classic.

Gunakan kode berikut untuk mengaktifkan propagasi nama profil pengguna sebagai. `sourceIdentity`

```
update-domain
--domain-id <value>
[--default-user-settings <value>]
[--domain-settings-for-update "ExecutionRoleIdentityConfig=USER_PROFILE_NAME"]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Matikan **`sourceIdentity`**

Anda juga dapat mematikan propagasi nama profil pengguna sebagai `sourceIdentity` menggunakan. AWS CLI Ini terjadi selama pembaruan domain dengan meneruskan `ExecutionRoleIdentityConfig=DISABLED` nilai untuk `--domain-settings-for-update` parameter sebagai bagian dari panggilan `update-domain` API.

DalamAWS CLI, gunakan kode berikut untuk menonaktifkan propagasi nama profil pengguna sebagai. `sourceIdentity`

```
update-domain
 --domain-id <value>
[--default-user-settings <value>]
[--domain-settings-for-update "ExecutionRoleIdentityConfig=DISABLED"]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Mengotomatisasi Amazon SageMaker dengan Amazon EventBridge

Amazon EventBridge memantau peristiwa perubahan status di Amazon SageMaker. EventBridge memungkinkan Anda untuk mengotomatisasi SageMaker dan merespons secara otomatis peristiwa seperti perubahan status pekerjaan pelatihan atau perubahan status titik akhir. Acara dari SageMaker dikirim ke EventBridge dalam waktu dekat. Anda dapat menulis aturan sederhana untuk menunjukkan kejadian mana yang sesuai kepentingan Anda, dan tindakan otomatis apa yang diambil ketika suatu kejadian sesuai dengan suatu aturan. Untuk contoh cara membuat aturan, lihatJadwalkan Pipeline dengan Amazon EventBridge.

> **Note**
>
> SageMaker dapat mengirim beberapa acara EventBridge untuk setiap perubahan status. Perilaku ini diharapkan dan tidak selalu menunjukkan kesalahan.

Beberapa contoh tindakan yang dapat dipicu secara otomatis termasuk yang berikut:

- Mengambil fungsi AWS Lambda
- Meminta Perintah Amazon EC2 Run
- Mengirim peristiwa ke Amazon Kinesis Data Streams
- Mengaktifkan mesin keadaan AWS Step Functions
- Memberitahu topik Amazon SNS atau antrian AWS SMS

SageMaker peristiwa yang dipantau oleh EventBridge

- SageMaker perubahan status model
- Perubahan status pekerjaan pelatihan
- Perubahan status pekerjaan tuning hyperparameter
- Ubah perubahan status pekerjaan
- Perubahan status titik akhir
- Perubahan status grup fitur
- Perubahan status paket model
- Perubahan status eksekusi pipa

- Perubahan status langkah pipa
- Memproses perubahan status pekerjaan
- SageMaker perubahan status gambar
- SageMaker perubahan status versi gambar
- Perubahan status penerapan titik akhir
- Perubahan status kartu model

SageMaker perubahan status model

Menunjukkan perubahan keadaan SageMaker model. Status berubah ketika SageMaker model dibuat atau dihapus.

```
{
"source": ["aws.sagemaker"],
"detail-type": ["SageMaker Model State Change"]
"Resources" : ["arn:aws:sagemaker:us-east-1:123456789012:model/model-name"]
}
```

Jika model ditentukan di bawah`Resources`, peristiwa akan dihasilkan dan dikirim ke EventBridge saat keadaan model ini berubah. Jika Anda tidak menentukan nilai untuk`Resources`, peristiwa akan muncul ketika status salah satu SageMaker model yang terkait dengan akun Anda berubah.

Perubahan status pekerjaan pelatihan

Menunjukkan perubahan status pekerjaan SageMaker pelatihan.

Jika nilainya `TrainingJobStatus` adalah`Failed`, acara berisi `FailureReason` bidang, yang memberikan deskripsi mengapa pekerjaan pelatihan gagal.

```
{
    "version": "0",
    "id": "844e2571-85d4-695f-b930-0153b71dcb42",
    "detail-type": "SageMaker Training Job State Change",
    "source": "aws.sagemaker",
    "account": "123456789012",
    "time": "2018-10-06T12:26:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:sagemaker:us-east-1:123456789012:training-job/kmeans-1"
```

```
    ],
    "detail": {
        "TrainingJobName": "89c96cc8-dded-4739-afcc-6f1dc936701d",
        "TrainingJobArn": "arn:aws:sagemaker:us-east-1:123456789012:training-job/
kmeans-1",
        "TrainingJobStatus": "Completed",
        "SecondaryStatus": "Completed",
        "HyperParameters": {
            "Hyper": "Parameters"
        },
        "AlgorithmSpecification": {
            "TrainingImage": "TrainingImage",
            "TrainingInputMode": "TrainingInputMode"
        },
        "RoleArn": "arn:aws:iam::123456789012:role/SMRole",
        "InputDataConfig": [
            {
                "ChannelName": "Train",
                "DataSource": {
                    "S3DataSource": {
                        "S3DataType": "S3DataType",
                        "S3Uri": "S3Uri",
                        "S3DataDistributionType": "S3DataDistributionType"
                    }
                },
                "ContentType": "ContentType",
                "CompressionType": "CompressionType",
                "RecordWrapperType": "RecordWrapperType"
            }
        ],
        "OutputDataConfig": {
            "KmsKeyId": "KmsKeyId",
            "S3OutputPath": "S3OutputPath"
        },
        "ResourceConfig": {
            "InstanceType": "InstanceType",
            "InstanceCount": 3,
            "VolumeSizeInGB": 20,
            "VolumeKmsKeyId": "VolumeKmsKeyId"
        },
        "VpcConfig": {

        },
        "StoppingCondition": {
```

```
            "MaxRuntimeInSeconds": 60
        },
        "CreationTime": "1583831889050",
        "TrainingStartTime": "1583831889050",
        "TrainingEndTime": "1583831889050",
        "LastModifiedTime": "1583831889050",
        "SecondaryStatusTransitions": [

        ],
        "Tags": {

        }
    }
}
```

Perubahan status pekerjaan tuning hyperparameter

Menunjukkan perubahan status pekerjaan tuning SageMaker hyperparameter.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "SageMaker HyperParameter Tuning Job State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:sagemaker:us-east-1:123456789012:tuningJob/x"
  ],
  "detail": {
    "HyperParameterTuningJobName": "016bffd3-6d71-4d3a-9710-0a332b2759fc",
    "HyperParameterTuningJobArn": "arn:aws:sagemaker:us-east-1:123456789012:tuningJob/
x",
    "TrainingJobDefinition": {
      "StaticHyperParameters": {},
      "AlgorithmSpecification": {
        "TrainingImage": "trainingImageName",
        "TrainingInputMode": "inputModeFile",
        "MetricDefinitions": [
          {
            "Name": "metricName",
            "Regex": "regex"
```

```
        }
      ]
    },
    "RoleArn": "roleArn",
    "InputDataConfig": [
      {
        "ChannelName": "channelName",
        "DataSource": {
          "S3DataSource": {
            "S3DataType": "s3DataType",
            "S3Uri": "s3Uri",
            "S3DataDistributionType": "s3DistributionType"
          }
        },
        "ContentType": "contentType",
        "CompressionType": "gz",
        "RecordWrapperType": "RecordWrapper"
      }
    ],
    "VpcConfig": {
      "SecurityGroupIds": [
        "securityGroupIds"
      ],
      "Subnets": [
        "subnets"
      ]
    },
    "OutputDataConfig": {
      "KmsKeyId": "kmsKeyId",
      "S3OutputPath": "s3OutputPath"
    },
    "ResourceConfig": {
      "InstanceType": "instanceType",
      "InstanceCount": 10,
      "VolumeSizeInGB": 500,
      "VolumeKmsKeyId": "volumeKeyId"
    },
    "StoppingCondition": {
      "MaxRuntimeInSeconds": 3600
    }
  },
  "HyperParameterTuningJobStatus": "status",
  "CreationTime": "1583831889050",
  "LastModifiedTime": "1583831889050",
```

```
    "TrainingJobStatusCounters": {
      "Completed": 1,
      "InProgress": 0,
      "RetryableError": 0,
      "NonRetryableError": 0,
      "Stopped": 0
    },
    "ObjectiveStatusCounters": {
      "Succeeded": 1,
      "Pending": 0,
      "Failed": 0
    },
    "Tags": {}
  }
}
```

Ubah perubahan status pekerjaan

Menunjukkan perubahan status pekerjaan transformasi SageMaker batch.

Jika nilainya `TransformJobStatus` adalah`Failed`, acara berisi `FailureReason` bidang, yang memberikan deskripsi mengapa pekerjaan pelatihan gagal.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "SageMaker Transform Job State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": ["arn:aws:sagemaker:us-east-1:123456789012:transform-job/myjob"],
  "detail": {
    "TransformJobName": "4b52bd8f-e034-4345-818d-884bdd7c9724",
    "TransformJobArn": "arn:aws:sagemaker:us-east-1:123456789012:transform-job/myjob",
    "TransformJobStatus": "another status... GO",
    "FailureReason": "failed why 1",
    "ModelName": "i am a beautiful model",
    "MaxConcurrentTransforms": 5,
    "MaxPayloadInMB": 10,
    "BatchStrategy": "Strategizing...",
    "Environment": {
      "environment1": "environment2"
```

```
    },
    "TransformInput": {
      "DataSource": {
        "S3DataSource": {
          "S3DataType": "s3DataType",
          "S3Uri": "s3Uri"
        }
      },
      "ContentType": "content type",
      "CompressionType": "compression type",
      "SplitType": "split type"
    },
    "TransformOutput": {
      "S3OutputPath": "s3Uri",
      "Accept": "accept",
      "AssembleWith": "assemblyType",
      "KmsKeyId": "kmsKeyId"
    },
    "TransformResources": {
      "InstanceType": "instanceType",
      "InstanceCount": 3
    },
    "CreationTime": "2018-10-06T12:26:13Z",
    "TransformStartTime": "2018-10-06T12:26:13Z",
    "TransformEndTime": "2018-10-06T12:26:13Z",
    "Tags": {}
  }
}
```

Perubahan status titik akhir

Menunjukkan perubahan status titik akhir inferensi real-time yang SageMaker dihosting.

Berikut ini menunjukkan peristiwa dengan titik akhir di `IN_SERVICE` negara bagian.

```
{
  "version": "0",
  "id": "d2921b5a-b0ad-cace-a8e3-0f159d018e06",
  "detail-type": "SageMaker Endpoint State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "1583831889050",
  "region": "us-west-2",
```

```
  "resources": [
      "arn:aws:sagemaker:us-west-2:123456789012:endpoint/myendpoint"
  ],
  "detail": {
      "EndpointName": "MyEndpoint",
      "EndpointArn": "arn:aws:sagemaker:us-west-2:123456789012:endpoint/myendpoint",
      "EndpointConfigName": "MyEndpointConfig",
      "ProductionVariants": [
          {
              "DesiredWeight": 1.0,
              "DesiredInstanceCount": 1.0
          }
      ],
      "EndpointStatus": "IN_SERVICE",
      "CreationTime": 1592411992203.0,
      "LastModifiedTime": 1592411994287.0,
      "Tags": {

      }
  }
}
```

Perubahan status grup fitur

Menunjukkan perubahan baik dalam `FeatureGroupStatus` `OfflineStoreStatus` atau grup SageMaker fitur.

```
{
  "version": "0",
  "id": "93201303-abdb-36a4-1b9b-4c1c3e3671c0",
  "detail-type": "SageMaker Feature Group State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2021-01-26T01:22:01Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:sagemaker:us-east-1:123456789012:feature-group/sample-feature-group"
  ],
  "detail": {
    "FeatureGroupArn": "arn:aws:sagemaker:us-east-1:123456789012:feature-group/sample-
feature-group",
    "FeatureGroupName": "sample-feature-group",
    "RecordIdentifierFeatureName": "RecordIdentifier",
```

```
    "EventTimeFeatureName": "EventTime",
    "FeatureDefinitions": [
      {
        "FeatureName": "RecordIdentifier",
        "FeatureType": "Integral"
      },
      {
        "FeatureName": "EventTime",
        "FeatureType": "Fractional"
      }
    ],
    "CreationTime": 1611624059000,
    "OnlineStoreConfig": {
      "EnableOnlineStore": true
    },
    "OfflineStoreConfig": {
      "S3StorageConfig": {
        "S3Uri": "s3://offline/s3/uri"
      },
      "DisableGlueTableCreation": false,
      "DataCatalogConfig": {
        "TableName": "sample-feature-group-1611624059",
        "Catalog": "AwsDataCatalog",
        "Database": "sagemaker_featurestore"
      }
    },
    "RoleArn": "arn:aws:iam::123456789012:role/SageMakerRole",
    "FeatureGroupStatus": "Active",
    "Tags": {}
  }
}
```

Perubahan status paket model

Menunjukkan perubahan status paket SageMaker model.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "SageMaker Model Package State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2021-02-24T17:00:14Z",
```

```
  "region": "us-east-2",
  "resources": [
    "arn:aws:sagemaker:us-east-2:123456789012:model-package/versionedmp-p-
idy6c3e1fiqj/2"
  ],
  "source": [
    "aws.sagemaker"
  ],
  "detail": {
    "ModelPackageGroupName": "versionedmp-p-idy6c3e1fiqj",
    "ModelPackageVersion": 2,
    "ModelPackageArn": "arn:aws:sagemaker:us-east-2:123456789012:model-package/
versionedmp-p-idy6c3e1fiqj/2",
    "CreationTime": "2021-02-24T17:00:14Z",
    "InferenceSpecification": {
      "Containers": [
        {
          "Image": "257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-
xgboost:1.0-1-cpu-py3",
          "ImageDigest":
 "sha256:4dc8a7e4a010a19bb9e0a6b063f355393f6e623603361bd8b105f554d4f0c004",
          "ModelDataUrl": "s3://sagemaker-project-p-idy6c3e1fiqj/versionedmp-p-
idy6c3e1fiqj/AbaloneTrain/pipelines-4r83jejmhorv-TrainAbaloneModel-xw869y8C4a/output/
model.tar.gz"
        }
      ],
      "SupportedContentTypes": [
        "text/csv"
      ],
      "SupportedResponseMIMETypes": [
        "text/csv"
      ]
    },
    "ModelPackageStatus": "Completed",
    "ModelPackageStatusDetails": {
      "ValidationStatuses": [],
      "ImageScanStatuses": []
    },
    "CertifyForMarketplace": false,
    "ModelApprovalStatus": "Rejected",
    "MetadataProperties": {
      "GeneratedBy": "arn:aws:sagemaker:us-east-2:123456789012:pipeline/versionedmp-p-
idy6c3e1fiqj/execution/4r83jejmhorv"
    },
```

```
    "ModelMetrics": {
      "ModelQuality": {
        "Statistics": {
          "ContentType": "application/json",
          "S3Uri": "s3://sagemaker-project-p-idy6c3e1fiqj/versionedmp-p-idy6c3e1fiqj/
script-2021-02-24-10-55-15-413/output/evaluation/evaluation.json"
        }
      }
    },
    "LastModifiedTime": "2021-02-24T17:00:14Z"
  }
}
```

Perubahan status eksekusi pipa

Menunjukkan perubahan status eksekusi SageMaker pipeline.

`currentPipelineExecutionStatus`dan `previousPipelineExecutionStatus` dapat menjadi salah satu nilai berikut:

- Melaksanakan
- Berhasil
- Failed
- Stopping
- Dihentikan

```
{
  "version": "0",
  "id": "315c1398-40ff-a850-213b-158f73kd93ir",
  "detail-type": "SageMaker Model Building Pipeline Execution Status Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2021-03-15T16:10:11Z",
  "region": "us-east-1",
  "resources": ["arn:aws:sagemaker:us-east-1:123456789012:pipeline/myPipeline-123",
 "arn:aws:sagemaker:us-east-1:123456789012:pipeline/myPipeline-123/execution/
p4jn9xou8a8s"],
  "detail": {
    "pipelineExecutionDisplayName": "SomeDisplayName",
    "currentPipelineExecutionStatus": "Succeeded",
```

```
    "previousPipelineExecutionStatus": "Executing",
    "executionStartTime": "2021-03-15T16:03:13Z",
    "executionEndTime": "2021-03-15T16:10:10Z",
    "pipelineExecutionDescription": "SomeDescription",
    "pipelineArn": "arn:aws:sagemaker:us-east-1:123456789012:pipeline/myPipeline-123",
    "pipelineExecutionArn": "arn:aws:sagemaker:us-east-1:123456789012:pipeline/
myPipeline-123/execution/p4jn9xou8a8s"
  }
}
```

Perubahan status langkah pipa

Menunjukkan perubahan status langkah SageMaker pipa.

Jika ada cache hit, acara berisi `cacheHitResult` bidang. `currentStepStatus`dan `previousStepStatus` dapat menjadi salah satu nilai berikut:

- Starting
- Melaksanakan
- Berhasil
- Failed
- Stopping
- Dihentikan

Jika nilainya `currentStepStatus` adalah`Failed`, acara berisi `failureReason` bidang, yang memberikan deskripsi mengapa langkah gagal.

```
{
  "version": "0",
  "id": "ea37ccbb-5e2b-05e9-4073-1daazc940304",
  "detail-type": "SageMaker Model Building Pipeline Execution Step Status Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2021-03-15T16:10:10Z",
  "region": "us-east-1",
  "resources": ["arn:aws:sagemaker:us-east-1:123456789012:pipeline/myPipeline-123",
 "arn:aws:sagemaker:us-east-1:123456789012:pipeline/myPipeline-123/execution/
p4jn9xou8a8s"],
  "detail": {
    "metadata": {
```

```
      "processingJob": {
        "arn": "arn:aws:sagemaker:us-east-1:123456789012:processing-job/pipelines-
p4jn9xou8a8s-myprocessingstep1-tmgxry49ug"
      }
    },
    "stepStartTime": "2021-03-15T16:03:14Z",
    "stepEndTime": "2021-03-15T16:10:09Z",
    "stepName": "myprocessingstep1",
    "stepType": "Processing",
    "previousStepStatus": "Executing",
    "currentStepStatus": "Succeeded",
    "pipelineArn": "arn:aws:sagemaker:us-east-1:123456789012:pipeline/myPipeline-123",
    "pipelineExecutionArn": "arn:aws:sagemaker:us-east-1:123456789012:pipeline/
myPipeline-123/execution/p4jn9xou8a8s"
  }
}
```

Memproses perubahan status pekerjaan

Menunjukkan perubahan status pekerjaan SageMaker Pemrosesan.

Contoh peristiwa berikut adalah untuk pekerjaan Pemrosesan yang gagal, di mana `ProcessingJobStatus` nilainya`Failed`.

```
{
  "version": "0",
  "id": "0a15f67d-aa23-0123-0123-01a23w89r01t",
  "detail-type": "SageMaker Processing Job State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:sagemaker:us-west-2:037210630506:processing-job/integ-test-
analytics-algo-54ee3282-5899-4aa3-afc2-7ce1d02"],
  "detail": {
    "ProcessingInputs": [{
      "InputName": "InputName",
      "S3Input": {
        "S3Uri": "s3://input/s3/uri",
        "LocalPath": "/opt/ml/processing/input/local/path",
        "S3DataType": "MANIFEST_FILE",
        "S3InputMode": "PIPE",
        "S3DataDistributionType": "FULLYREPLICATED"
```

```
      }
    }],
    "ProcessingOutputConfig": {
      "Outputs": [{
        "OutputName": "OutputName",
        "S3Output": {
          "S3Uri": "s3://output/s3/uri",
          "LocalPath": "/opt/ml/processing/output/local/path",
          "S3UploadMode": "CONTINUOUS"
        }
      }],
      "KmsKeyId": "KmsKeyId"
    },
    "ProcessingJobName": "integ-test-analytics-algo-54ee3282-5899-4aa3-afc2-7ce1d02",
    "ProcessingResources": {
      "ClusterConfig": {
        "InstanceCount": 3,
        "InstanceType": "ml.c5.xlarge",
        "VolumeSizeInGB": 5,
        "VolumeKmsKeyId": "VolumeKmsKeyId"
      }
    },
    "StoppingCondition": {
      "MaxRuntimeInSeconds": 2000
    },
    "AppSpecification": {
      "ImageUri": "012345678901.dkr.ecr.us-west-2.amazonaws.com/processing-uri:latest"
    },
    "NetworkConfig": {
      "EnableInterContainerTrafficEncryption": true,
      "EnableNetworkIsolation": false,
      "VpcConfig": {
        "SecurityGroupIds": ["SecurityGroupId1", "SecurityGroupId2",
 "SecurityGroupId3"],
        "Subnets": ["Subnet1", "Subnet2"]
      }
    },
    "RoleArn": "arn:aws:iam::037210630506:role/SageMakerPowerUser",
    "ExperimentConfig": {},
    "ProcessingJobArn": "arn:aws:sagemaker:us-west-2:037210630506:processing-job/integ-
test-analytics-algo-54ee3282-5899-4aa3-afc2-7ce1d02",
    "ProcessingJobStatus":"Failed",
    "FailureReason":"InternalServerError: We encountered an internal error.  Please try
 again.",
```

```
    "ProcessingEndTime":1704320746000,
    "ProcessingStartTime":1704320734000,
    "LastModifiedTime":1704320746000,
    "CreationTime":1704320199000
  }
}
```

SageMaker perubahan status gambar

Menunjukkan perubahan status SageMaker gambar.

```
{
  "version": "0",
  "id": "cee033a3-17d8-49f8-865f-b9ebf485d9ee",
  "detail-type": "SageMaker Image State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2021-04-29T01:29:59Z",
  "region": "us-east-1",
  "resources": ["arn:aws:sagemaker:us-west-2:123456789012:image/
cee033a3-17d8-49f8-865f-b9ebf485d9ee"],
  "detail": {
    "ImageName": "cee033a3-17d8-49f8-865f-b9ebf485d9ee",
    "ImageArn": "arn:aws:sagemaker:us-west-2:123456789012:image/
cee033a3-17d8-49f8-865f-b9ebf485d9ee",
    "ImageStatus": "Creating",
    "Version": 1.0,
    "Tags": {}
  }
}
```

SageMaker perubahan status versi gambar

Menunjukkan perubahan status versi SageMaker gambar.

```
{
  "version": "0",
  "id": "07fc4615-ebd7-15fc-1746-243411f09f04",
  "detail-type": "SageMaker Image Version State Change",
  "source": "aws.sagemaker",
  "account": "123456789012",
  "time": "2021-04-29T01:29:59Z",
```

```
  "region": "us-east-1",
  "resources": ["arn:aws:sagemaker:us-west-2:123456789012:image-
version/07800032-2d29-48b7-8f82-5129225b2a85"],
  "detail": {
    "ImageArn": "arn:aws:sagemaker:us-west-2:123456789012:image/a70ff896-c832-4fe8-
add6-eba25a0f43e6",
    "ImageVersionArn": "arn:aws:sagemaker:us-west-2:123456789012:image-
version/07800032-2d29-48b7-8f82-5129225b2a85",
    "ImageVersionStatus": "Creating",
    "Version": 1.0,
    "Tags": {}
  }
}
```

Untuk informasi selengkapnya tentang nilai status dan artinya untuk SageMaker pekerjaan, titik akhir, dan saluran pipa, lihat tautan berikut:

- AlgorithmStatus
- EndpointStatus
- FeatureGroupStatus
- HyperParameterTuningJobStatus
- LabelingJobStatus
- ModelPackageStatus
- NotebookInstanceStatus
- PipelineExecutionStatus
- StepStatus
- ProcessingJobStatus
- TrainingJobStatus
- TransformJobStatus

Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

Perubahan status penerapan titik akhir

> **Important**
>
> Contoh berikut mungkin tidak berfungsi untuk semua titik akhir. Untuk daftar fitur yang mungkin mengecualikan titik akhir Anda, lihat Pengecualian halaman.

Menunjukkan perubahan status untuk penerapan titik akhir. Contoh berikut menunjukkan pemutakhiran endpoint dengan penerapan kenari biru/hijau.


```
{
    "version": "0",
    "id": "0bd4a141-0a02-9d8a-f977-3924c3fb259c",
    "detail-type": "SageMaker Endpoint Deployment State Change",
    "source": "aws.sagemaker",
    "account": "123456789012",
    "time": "2021-10-25T01:52:12Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:sagemaker:us-west-2:651393343886:endpoint/sample-endpoint"
    ],
    "detail": {
        "EndpointName": "sample-endpoint",
        "EndpointArn": "arn:aws:sagemaker:us-west-2:651393343886:endpoint/sample-
endpoint",
        "EndpointConfigName": "sample-endpoint-config-1",
        "ProductionVariants": [
            {
                "VariantName": "AllTraffic",
                "CurrentWeight": 1,
                "DesiredWeight": 1,
                "CurrentInstanceCount": 3,
                "DesiredInstanceCount": 3
            }
        ],
        "EndpointStatus": "UPDATING",
        "CreationTime": 1635195148181,
        "LastModifiedTime": 1635195148181,
        "Tags": {},
        "PendingDeploymentSummary": {
            "EndpointConfigName": "sample-endpoint-config-2",
```

```
            "StartTime": Timestamp,
            "ProductionVariants": [
                {
                    "VariantName": "AllTraffic",
                    "CurrentWeight": 1,
                    "DesiredWeight": 1,
                    "CurrentInstanceCount": 1,
                    "DesiredInstanceCount": 3,
                    "VariantStatus": [
                        {
                            "Status": "Baking",
                            "StatusMessage": "Baking for 600 seconds
 (TerminationWaitInSeconds) with traffic enabled on canary capacity of 1 instance(s).",
                            "StartTime": 1635195269181,
                        }
                    ]
                }
            ]
        }
    }
}
```

Contoh berikut menunjukkan perubahan status untuk penerapan titik akhir, yang sedang diperbarui dengan kapasitas baru pada konfigurasi titik akhir yang ada.

```
{
    "version": "0",
    "id": "0bd4a141-0a02-9d8a-f977-3924c3fb259c",
    "detail-type": "SageMaker Endpoint Deployment State Change",
    "source": "aws.sagemaker",
    "account": "123456789012",
    "time": "2021-10-25T01:52:12Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:sagemaker:us-west-2:651393343886:endpoint/sample-endpoint"
    ],
    "detail": {
        "EndpointName": "sample-endpoint",
        "EndpointArn": "arn:aws:sagemaker:us-west-2:651393343886:endpoint/sample-
endpoint",
        "EndpointConfigName": "sample-endpoint-config-1",
        "ProductionVariants": [
            {
```

```
                "VariantName": "AllTraffic",
                "CurrentWeight": 1,
                "DesiredWeight": 1,
                "CurrentInstanceCount": 3,
                "DesiredInstanceCount": 6,
                "VariantStatus": [
                    {
                        "Status": "Updating",
                        "StatusMessage": "Scaling out desired instance count to 6.",
                        "StartTime": 1635195269181,
                    }
                ]
            }
        ],
        "EndpointStatus": "UPDATING",
        "CreationTime": 1635195148181,
        "LastModifiedTime": 1635195148181,
        "Tags": {},
    }
```

Status penerapan sekunder berikut juga tersedia untuk titik akhir (ditemukan di objek. `VariantStatus`

- `Creating`: membuat instance untuk varian produksi.

  Contoh pesan: `"Launching X instance(s)."`
- `Deleting`: mengakhiri instance untuk varian produksi.

  Contoh pesan: `"Terminating X instance(s)."`
- `Updating`: memperbarui kapasitas untuk varian produksi.

  Contoh pesan:`"Launching X instance(s)."`, `"Scaling out desired instance count to X."`
- `ActivatingTraffic`: mengaktifkan lalu lintas untuk varian produksi.

  Contoh pesan: `"Activating traffic on canary capacity of X instance(s)."`
- `Baking`: masa tunggu untuk memantau CloudWatch alarm dalam konfigurasi auto-rollback.

  Contoh pesan: `"Baking for X seconds (TerminationWaitInSeconds) with traffic enabled on full capacity of Y instance(s)."`

Perubahan status kartu model

Menunjukkan perubahan status Kartu SageMaker Model Amazon. Untuk informasi lebih lanjut tentang kartu model, lihatKartu SageMaker Model Amazon.


```
{
    "version": "0",
    "id": "aa7a9c4f-2caa-4d04-a6de-e67227ba4302",
    "detail-type": "SageMaker Model Card State Change",
    "source": "aws.sagemaker",
    "account": "123456789012",
    "time": "2022-11-30T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:sagemaker:us-east-1:123456789012:model-card/example-card"
    ],
    "detail": {
        "ModelCardVersion": 2,
        "LastModifiedTime": "2022-12-03T00:09:44.893854735Z",
        "LastModifiedBy": {
            "DomainId": "us-east-1",
            "UserProfileArn": "arn:aws:sagemaker:us-east-1:123456789012:user-profile/
user",
            "UserProfileName": "user"
        },
        "CreationTime": "2022-12-03T00:09:33.084Z",
        "CreatedBy": {
            "DomainId": "us-east-1",
            "UserProfileArn": "arn:aws:sagemaker:us-east-1:123456789012:user-profile/
user",
            "UserProfileName": "user"
        },
        "ModelCardName": "example-card",
        "ModelId": "example-model",
        "ModelCardStatus": "Draft",
        "AccountId": "123456789012",
        "SecurityConfig": {}
    }
}
```

SageMaker Referensi Amazon

Topik

Kerangka Kerja dan Bahasa Machine Learning

Anda dapat menggunakan Python dan R secara native di kernel notebook Amazon SageMaker . Ada juga kernel yang mendukung kerangka kerja tertentu. Cara yang sangat populer untuk memulai SageMaker adalah dengan menggunakan Amazon SageMaker Python SDK. Ini menyediakan API dan wadah Python open source yang memudahkan untuk melatih dan menerapkan model SageMaker, serta contoh untuk digunakan dengan beberapa kerangka kerja pembelajaran mesin dan pembelajaran mendalam yang berbeda.

Untuk informasi tentang penggunaan kerangka kerja tertentu atau cara menggunakan R di SageMaker, lihat topik berikut.

Bahasa SDK dan panduan pengguna:

- Amazon SageMaker Python SDK
- R
- Referensi API

Panduan kerangka pembelajaran mesin dan pembelajaran mendalam:

- Apache MXNet
- Apache Spark
- Chainer
- Hugging Face

- PyTorch
- Scikit-belajar
- Penyajian SparkML
- TensorFlow
- Server Inferensi Triton

Gunakan Apache MxNet dengan Amazon SageMaker

Anda dapat menggunakan SageMaker untuk melatih dan menerapkan model menggunakan kode MXNet kustom. Estimator dan model MXNet Amazon SageMaker Python SDK dan wadah MXNet open-source membuat penulisan SageMaker skrip MXNet dan menjalankannya lebih mudah. SageMaker

Apa yang ingin Anda lakukan?

Saya ingin melatih model MXNet khusus. SageMaker

> Untuk contoh notebook Jupyter, lihat contoh notebook MxNet di repositori Amazon Examples. SageMaker GitHub
>
> Untuk dokumentasi, lihat Melatih Model dengan MxNet.

Saya memiliki model MXNet yang saya latih SageMaker, dan saya ingin menerapkannya ke titik akhir yang dihosting.

> Untuk informasi selengkapnya, lihat Men-deploy model MxNet.

Saya memiliki model MXNet yang saya latih di luar SageMaker, dan saya ingin menerapkannya ke titik akhir SageMaker

> Untuk informasi selengkapnya, lihat Men-deploy Endpoints dari Data Model.

Saya ingin melihat dokumentasi API untuk kelas Amazon SageMaker Python SDK MxNet.

> Untuk informasi selengkapnya, lihat Kelas MXNet.

Saya ingin menemukan repositori SageMaker kontainer MxNet.

> Untuk informasi selengkapnya, lihat repositori SageMaker MxNet Container GitHub .

Saya ingin mencari informasi tentang versi MxNet yang didukung oleh Deep Learning AWS Containers.

Untuk informasi selengkapnya, lihat Gambar Kontainer Pembelajaran Mendalam yang Tersedia.

Untuk informasi umum tentang menulis skrip pelatihan mode skrip MxNet dan menggunakan estimator dan model mode skrip MxNet, lihat Menggunakan MxNet dengan SageMaker Python SDK. SageMaker

Gunakan Apache Spark dengan Amazon SageMaker

Amazon SageMaker Spark adalah pustaka Spark open source yang membantu Anda membuat alur machine learning (ML) Spark dengan. SageMaker Ini menyederhanakan integrasi tahap Spark ML dengan SageMaker tahapan, seperti pelatihan model dan hosting. Untuk informasi tentang SageMaker Spark, lihat repositori SageMaker Spark GitHub.

Pustaka SageMaker Spark tersedia dalam Python dan Scala. Anda dapat menggunakan SageMaker Spark untuk melatih model dalam SageMaker menggunakan bingkai `org.apache.spark.sql.DataFrame` data di cluster Spark Anda. Setelah pelatihan model, Anda juga dapat meng-host model menggunakan layanan SageMaker hosting.

Perpustakaan SageMaker Spark,`com.amazonaws.services.sagemaker.sparksdk`, menyediakan kelas-kelas berikut, antara lain:

- `SageMakerEstimator`—Memperluas antarmuka. `org.apache.spark.ml.Estimator` Anda dapat menggunakan estimator ini untuk pelatihan model di SageMaker.
- `KMeansSageMakerEstimator`,`PCASageMakerEstimator`, dan `XGBoostSageMakerEstimator` —Memperpanjang `SageMakerEstimator` kelas.
- `SageMakerModel`—Memperluas kelas. `org.apache.spark.ml.Model` Anda dapat menggunakan ini `SageMakerModel` untuk hosting model dan mendapatkan kesimpulan di SageMaker.

Anda dapat mengunduh kode sumber untuk pustaka Python Spark (PySpark) dan Scala dari repositori Spark. SageMaker GitHub

Untuk instalasi dan contoh perpustakaan SageMaker Spark, lihat SageMaker Spark untuk contoh Scala atauSageMaker Contoh percikan untuk Python PySpark ().

Jika Anda menggunakan Amazon EMR AWS untuk mengelola cluster Spark, lihat Apache Spark. Untuk informasi selengkapnya tentang menggunakan Amazon EMR di SageMaker, lihat. Siapkan data menggunakan Amazon EMR

Topik

- Integrasikan Aplikasi Apache Spark Anda dengan SageMaker
- SageMaker Spark untuk contoh Scala
- SageMaker Contoh percikan untuk Python PySpark ()

Integrasikan Aplikasi Apache Spark Anda dengan SageMaker

Berikut ini adalah rangkuman tingkat tinggi dari langkah-langkah untuk mengintegrasikan aplikasi Apache Spark Anda dengan. SageMaker

1. Lanjutkan preprocessing data menggunakan pustaka Apache Spark yang Anda kenal. Dataset Anda tetap berada `DataFrame` di cluster Spark Anda. Muat data Anda ke dalam `DataFrame` dan pra-proses sehingga Anda memiliki `features` kolom dengan `org.apache.spark.ml.linalg.Vector` ofDoubles, dan `label` kolom opsional dengan nilai `Double` tipe.
2. Gunakan estimator di perpustakaan SageMaker Spark untuk melatih model Anda. Misalnya, jika Anda memilih algoritma k-means yang disediakan oleh SageMaker untuk pelatihan model, Anda memanggil `KMeansSageMakerEstimator.fit` metode.

   Berikan Anda `DataFrame` sebagai masukan. Estimator mengembalikan `SageMakerModel` objek.

   > **Note**
   >
   > `SageMakerModel`memperluas. `org.apache.spark.ml.Model`

   `fit`Metode ini melakukan hal berikut:

   a. Mengonversi input `DataFrame` ke format protobuf dengan memilih `label` kolom `features` dan dari input `DataFrame` dan mengunggah data protobuf ke bucket Amazon S3. Format protobuf efisien untuk pelatihan model di. SageMaker

b. Memulai pelatihan model SageMaker dengan mengirimkan SageMaker `CreateTrainingJob`permintaan. Setelah pelatihan model selesai, SageMaker simpan artefak model ke ember S3.

   SageMaker mengasumsikan IAM role yang Anda tentukan untuk pelatihan model untuk melakukan tugas atas nama Anda. Misalnya, ia menggunakan peran untuk membaca data pelatihan dari bucket S3 dan menulis artefak model ke ember.

c. Menciptakan dan mengembalikan `SageMakerModel` objek. Konstruktor melakukan tugas-tugas berikut, yang terkait dengan penerapan model Anda. SageMaker

   i. Mengirim `CreateModel`permintaan ke SageMaker.

   ii. Mengirim `CreateEndpointConfig`permintaan ke SageMaker.

   iii. Mengirim `CreateEndpoint`permintaan ke SageMaker, yang kemudian meluncurkan sumber daya yang ditentukan, dan menghosting model pada mereka.

3. Anda bisa mendapatkan kesimpulan dari model Anda yang di-host SageMaker dengan. `SageMakerModel.transform`

   Berikan masukan `DataFrame` dengan fitur sebagai masukan. `transform`Metode mengubahnya menjadi kesimpulan yang `DataFrame` mengandung. Secara internal, `transform` metode mengirimkan permintaan ke `InvokeEndpoint` SageMaker API untuk mendapatkan kesimpulan. `transform`Metode ini menambahkan kesimpulan ke input. `DataFrame`

SageMaker Spark untuk contoh Scala

Amazon SageMaker menyediakan pustaka Apache Spark (SageMakerSpark) yang dapat Anda gunakan untuk mengintegrasikan aplikasi Apache Spark Anda. SageMaker Misalnya, Anda dapat menggunakan Apache Spark untuk preprocessing data dan SageMaker untuk pelatihan model dan hosting. Untuk informasi tentang perpustakaan SageMaker Apache Spark, lihat. Gunakan Apache Spark dengan Amazon SageMaker

Unduh Spark for Scala

Anda dapat mengunduh kode sumber dan contoh untuk pustaka Python Spark (PySpark) dan Scala dari repositori Spark. SageMaker GitHub

Untuk petunjuk rinci tentang menginstal perpustakaan SageMaker Spark, lihat SageMakerSpark.

SageMaker Spark SDK untuk Scala tersedia di repositori pusat Maven. Tambahkan pustaka Spark ke proyek Anda dengan menambahkan dependensi berikut ke file Andapom.xml:

- Jika proyek Anda dibangun dengan Maven, tambahkan yang berikut ini ke file pom.xml Anda:

```
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>sagemaker-spark_2.11</artifactId>
    <version>spark_2.2.0-1.0</version>
</dependency>
```

- Jika proyek Anda bergantung pada Spark 2.1, tambahkan yang berikut ini ke file pom.xml Anda:

```
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>sagemaker-spark_2.11</artifactId>
    <version>spark_2.1.1-1.0</version>
</dependency>
```

Spark untuk contoh Scala

Bagian ini memberikan contoh kode yang menggunakan pustaka Apache Spark Scala yang disediakan oleh SageMaker untuk melatih model dalam SageMaker menggunakan DataFrame s di cluster Spark Anda. Ini kemudian diikuti dengan contoh tentang cara Gunakan Algoritma Kustom untuk Pelatihan Model dan Hosting di Amazon SageMaker dengan Apache Spark danGunakan SageMakerEstimator dalam Pipa Spark.

Contoh berikut menampung artefak model yang dihasilkan menggunakan layanan SageMaker hosting. Untuk detail selengkapnya tentang contoh ini, lihat Memulai: K-Means Clustering on SageMaker with SageMaker Spark SDK Secara khusus, contoh ini melakukan hal berikut:

- Menggunakan KMeansSageMakerEstimator untuk menyesuaikan (atau melatih) model pada data

  Karena contoh menggunakan algoritma k-means yang disediakan oleh SageMaker untuk melatih model, Anda menggunakan. KMeansSageMakerEstimator Anda melatih model menggunakan gambar angka satu digit tulisan tangan (dari kumpulan data MNIST). Anda memberikan gambar sebagai masukanDataFrame. Demi kenyamanan Anda, SageMaker sediakan kumpulan data ini dalam bucket Amazon S3.

Sebagai tanggapan, estimator mengembalikan `SageMakerModel` objek.

- Mendapatkan kesimpulan menggunakan yang terlatih `SageMakerModel`

  Untuk mendapatkan kesimpulan dari model yang di-host SageMaker, Anda memanggil `SageMakerModel.transform` metode. Anda lulus `DataFrame` sebagai input. Metode ini mengubah input `DataFrame` ke yang lain `DataFrame` yang mengandung kesimpulan yang diperoleh dari model.

  Untuk gambar input tertentu dari angka satu digit tulisan tangan, inferensi mengidentifikasi cluster tempat gambar tersebut berada. Untuk informasi selengkapnya, lihat Algoritma K-Means.

```
import org.apache.spark.sql.SparkSession
import com.amazonaws.services.sagemaker.sparksdk.IAMRole
import com.amazonaws.services.sagemaker.sparksdk.algorithms
import com.amazonaws.services.sagemaker.sparksdk.algorithms.KMeansSageMakerEstimator

val spark = SparkSession.builder.getOrCreate

// load mnist data as a dataframe from libsvm
val region = "us-east-1"
val trainingData = spark.read.format("libsvm")
  .option("numFeatures", "784")
  .load(s"s3://sagemaker-sample-data-$region/spark/mnist/train/")
val testData = spark.read.format("libsvm")
  .option("numFeatures", "784")
  .load(s"s3://sagemaker-sample-data-$region/spark/mnist/test/")

val roleArn = "arn:aws:iam::account-id:role/rolename"

val estimator = new KMeansSageMakerEstimator(
  sagemakerRole = IAMRole(roleArn),
  trainingInstanceType = "ml.p2.xlarge",
  trainingInstanceCount = 1,
  endpointInstanceType = "ml.c4.xlarge",
  endpointInitialInstanceCount = 1)
  .setK(10).setFeatureDim(784)

// train
val model = estimator.fit(trainingData)
```

```
val transformedData = model.transform(testData)
transformedData.show
```

Kode contoh melakukan hal berikut:

- Memuat dataset MNIST dari bucket S3 yang disediakan oleh SageMaker (`awsai-sparksdk-dataset`) ke dalam Spark (): `DataFrame` `mnistTrainingDataFrame`

```
// Get a Spark session.

val spark = SparkSession.builder.getOrCreate

// load mnist data as a dataframe from libsvm
val region = "us-east-1"
val trainingData = spark.read.format("libsvm")
  .option("numFeatures", "784")
  .load(s"s3://sagemaker-sample-data-$region/spark/mnist/train/")
val testData = spark.read.format("libsvm")
  .option("numFeatures", "784")
  .load(s"s3://sagemaker-sample-data-$region/spark/mnist/test/")

val roleArn = "arn:aws:iam::account-id:role/rolename"
trainingData.show()
```

  showMetode ini menampilkan 20 baris pertama dalam bingkai data:

```
+-----+--------------------+
|label|            features|
+-----+--------------------+
|  5.0|(784,[152,153,154...|
|  0.0|(784,[127,128,129...|
|  4.0|(784,[160,161,162...|
|  1.0|(784,[158,159,160...|
|  9.0|(784,[208,209,210...|
|  2.0|(784,[155,156,157...|
|  1.0|(784,[124,125,126...|
|  3.0|(784,[151,152,153...|
|  1.0|(784,[152,153,154...|
|  4.0|(784,[134,135,161...|
|  3.0|(784,[123,124,125...|
|  5.0|(784,[216,217,218...|
|  3.0|(784,[143,144,145...|
```

```
|  6.0|(784,[72,73,74,99...|
|  1.0|(784,[151,152,153...|
|  7.0|(784,[211,212,213...|
|  2.0|(784,[151,152,153...|
|  8.0|(784,[159,160,161...|
|  6.0|(784,[100,101,102...|
|  9.0|(784,[209,210,211...|
+-----+--------------------+
only showing top 20 rows
```

Di setiap baris:

- `label`Kolom mengidentifikasi label gambar. Misalnya, jika gambar angka tulisan tangan adalah digit 5, nilai labelnya adalah 5.
- `features`Kolom menyimpan vektor (`org.apache.spark.ml.linalg.Vector`) `Double` nilai. Ini adalah 784 fitur dari nomor tulisan tangan. (Setiap angka tulisan tangan adalah gambar 28 x 28 piksel, membuat 784 fitur.)

- Menciptakan SageMaker estimator () `KMeansSageMakerEstimator`

  `fit`Metode estimator ini menggunakan algoritma k-means yang disediakan oleh SageMaker untuk melatih model menggunakan input. `DataFrame` Sebagai tanggapan, ia mengembalikan `SageMakerModel` objek yang dapat Anda gunakan untuk mendapatkan kesimpulan.

  > **Note**
  >
  > `KMeansSageMakerEstimator`Memperpanjang SageMaker`SageMakerEstimator`, yang memperluas Apache Spark. `Estimator`

```
val estimator = new KMeansSageMakerEstimator(
  sagemakerRole = IAMRole(roleArn),
  trainingInstanceType = "ml.p2.xlarge",
  trainingInstanceCount = 1,
  endpointInstanceType = "ml.c4.xlarge",
  endpointInitialInstanceCount = 1)
  .setK(10).setFeatureDim(784)
```

Parameter konstruktor memberikan informasi yang digunakan untuk melatih model dan menerapkannya pada: SageMaker

- `trainingInstanceType`dan `trainingInstanceCount` —Mengidentifikasi tipe dan jumlah instans komputasi ML untuk digunakan untuk pelatihan model.
- `endpointInstanceType`—Mengidentifikasi tipe instans komputasi ML yang akan digunakan saat menghosting model di. SageMaker Secara default, satu instance komputasi ML diasumsikan.
- `endpointInitialInstanceCount`—Mengidentifikasi jumlah instance komputasi ML yang awalnya mendukung titik akhir yang menghosting model. SageMaker
- `sagemakerRole`— SageMaker mengasumsikan peran IAM ini untuk melakukan tugas atas nama Anda. Misalnya, untuk pelatihan model, ia membaca data dari S3 dan menulis hasil pelatihan (artefak model) ke S3.

  > **Note**
  >
  > Contoh ini secara implisit menciptakan klien. SageMaker Untuk membuat klien ini, Anda harus memberikan kredensyal Anda. API menggunakan kredensyal ini untuk mengautentikasi permintaan ke. SageMaker Misalnya, menggunakan kredensi untuk mengautentikasi permintaan untuk membuat pekerjaan pelatihan dan panggilan API untuk menerapkan model menggunakan layanan hosting. SageMaker

- Setelah `KMeansSageMakerEstimator` objek telah dibuat, Anda mengatur parameter berikut, digunakan dalam pelatihan model:
  - Jumlah cluster yang harus dibuat oleh algoritma k-means selama pelatihan model. Anda menentukan 10 cluster, satu untuk setiap digit, 0 hingga 9.
  - Mengidentifikasi bahwa setiap gambar input memiliki 784 fitur (setiap nomor tulisan tangan adalah gambar 28 x 28 piksel, membuat 784 fitur).
- Memanggil metode estimator `fit`

```
// train
val model = estimator.fit(trainingData)
```

Anda meneruskan input `DataFrame` sebagai parameter. Model melakukan semua pekerjaan melatih model dan SageMaker menerapkannya. Untuk informasi selengkapnya, lihat Integrasikan Aplikasi Apache Spark Anda dengan SageMaker. Sebagai tanggapan, Anda mendapatkan `SageMakerModel` objek, yang dapat Anda gunakan untuk mendapatkan kesimpulan dari model Anda yang digunakan. SageMaker

Anda hanya memberikan masukan`DataFrame`. Anda tidak perlu menentukan jalur registri ke algoritma k-means yang digunakan untuk pelatihan model karena `KMeansSageMakerEstimator` mengetahuinya.

- Memanggil `SageMakerModel.transform` metode untuk mendapatkan kesimpulan dari model yang digunakan. SageMaker

  `transform`Metode ini mengambil input `DataFrame` as, mengubahnya, dan mengembalikan kesimpulan lain `DataFrame` yang mengandung kesimpulan yang diperoleh dari model.

```
val transformedData = model.transform(testData)
transformedData.show
```

  Untuk kesederhanaan, kami menggunakan input yang `DataFrame` sama dengan `transform` metode yang kami gunakan untuk pelatihan model dalam contoh ini. `transform`Metode ini melakukan hal-hal berikut:

  - Serialisasi `features` kolom dalam input `DataFrame` ke protobuf dan mengirimkannya ke titik SageMaker akhir untuk inferensi.
  - Deserialisasi respons protobuf menjadi dua kolom tambahan (`distance_to_clusterdanclosest_cluster`) dalam transformasi. `DataFrame`

  `show`Metode ini mendapatkan kesimpulan ke 20 baris pertama dalam input`DataFrame`:

```
+-----+--------------------+------------------+--------------+
|label|            features|distance_to_cluster|closest_cluster|
+-----+--------------------+------------------+--------------+
|  5.0|(784,[152,153,154...|  1767.897705078125|           4.0|
|  0.0|(784,[127,128,129...|  1392.157470703125|           5.0|
|  4.0|(784,[160,161,162...| 1671.5711669921875|           9.0|
|  1.0|(784,[158,159,160...| 1182.6082763671875|           6.0|
|  9.0|(784,[208,209,210...| 1390.4002685546875|           0.0|
|  2.0|(784,[155,156,157...|  1713.988037109375|           1.0|
|  1.0|(784,[124,125,126...| 1246.3016357421875|           2.0|
|  3.0|(784,[151,152,153...|  1753.229248046875|           4.0|
|  1.0|(784,[152,153,154...|  978.8394165039062|           2.0|
|  4.0|(784,[134,135,161...|  1623.176513671875|           3.0|
|  3.0|(784,[123,124,125...|  1533.863525390625|           4.0|
|  5.0|(784,[216,217,218...|  1469.357177734375|           6.0|
|  3.0|(784,[143,144,145...|  1736.765869140625|           4.0|
|  6.0|(784,[72,73,74,99...|   1473.69384765625|           8.0|
```

```
|  1.0|(784,[151,152,153...|    944.88720703125|            2.0|
|  7.0|(784,[211,212,213...| 1285.9071044921875|            3.0|
|  2.0|(784,[151,152,153...| 1635.0125732421875|            1.0|
|  8.0|(784,[159,160,161...| 1436.3162841796875|            6.0|
|  6.0|(784,[100,101,102...| 1499.7366943359375|            7.0|
|  9.0|(784,[209,210,211...| 1364.6319580078125|            6.0|
+-----+--------------------+-------------------+---------------+
```

Anda dapat menafsirkan data, sebagai berikut:

- Angka tulisan tangan dengan `label` 5 milik cluster 4 (`closest_cluster`).
- Angka tulisan tangan dengan `label` 0 milik cluster 5.
- Nomor tulisan tangan dengan `label` 4 milik cluster 9.
- Nomor tulisan tangan dengan `label` 1 milik cluster 6.

Topik

- Gunakan Algoritma Kustom untuk Pelatihan Model dan Hosting di Amazon SageMaker dengan Apache Spark
- Gunakan SageMakerEstimator dalam Pipa Spark

Gunakan Algoritma Kustom untuk Pelatihan Model dan Hosting di Amazon SageMaker dengan Apache Spark

DiSageMaker Spark untuk contoh Scala, Anda menggunakan `kMeansSageMakerEstimator` karena contoh menggunakan algoritma k-means yang disediakan oleh Amazon SageMaker untuk pelatihan model. Anda dapat memilih untuk menggunakan algoritma kustom Anda sendiri untuk pelatihan model sebagai gantinya. Dengan asumsi bahwa Anda telah membuat gambar Docker, Anda dapat membuat sendiri `SageMakerEstimator` dan menentukan jalur Amazon Elastic Container Registry untuk gambar kustom Anda.

Contoh berikut menunjukkan cara membuat `KMeansSageMakerEstimator` dari`SageMakerEstimator`. Di estimator baru, Anda secara eksplisit menentukan jalur registri Docker ke gambar kode pelatihan dan inferensi Anda.

```
import com.amazonaws.services.sagemaker.sparksdk.IAMRole
import com.amazonaws.services.sagemaker.sparksdk.SageMakerEstimator
import
 com.amazonaws.services.sagemaker.sparksdk.transformation.serializers.ProtobufRequestRowSeriali
```

```
import
 com.amazonaws.services.sagemaker.sparksdk.transformation.deserializers.KMeansProtobufResponseR

val estimator = new SageMakerEstimator(
  trainingImage =
    "811284229777.dkr.ecr.us-east-1.amazonaws.com/kmeans:1",
  modelImage =
    "811284229777.dkr.ecr.us-east-1.amazonaws.com/kmeans:1",
  requestRowSerializer = new ProtobufRequestRowSerializer(),
  responseRowDeserializer = new KMeansProtobufResponseRowDeserializer(),
  hyperParameters = Map("k" -> "10", "feature_dim" -> "784"),
  sagemakerRole = IAMRole(roleArn),
  trainingInstanceType = "ml.p2.xlarge",
  trainingInstanceCount = 1,
  endpointInstanceType = "ml.c4.xlarge",
  endpointInitialInstanceCount = 1,
  trainingSparkDataFormat = "sagemaker")
```

Dalam kode, parameter dalam `SageMakerEstimator` konstruktor meliputi:

- `trainingImage`—Mengidentifikasi jalur registri Docker ke gambar pelatihan yang berisi kode kustom Anda.
- `modelImage`—Mengidentifikasi jalur registri Docker ke gambar yang berisi kode inferensi.
- `requestRowSerializercom.amazonaws.services.sagemaker.sparksdk.transformation.Re` Menerapkan.

  Parameter ini membuat serial baris dalam input `DataFrame` untuk mengirimkannya ke model yang dihosting SageMaker untuk inferensi.
- `responseRowDeserializer`—Menerapkan

  `com.amazonaws.services.sagemaker.sparksdk.transformation.ResponseRowDeserialize`

  Parameter ini mendeserialisasi respons dari model, di-host SageMaker, kembali ke file. `DataFrame`
- `trainingSparkDataFormat`—Menentukan format data yang digunakan Spark saat mengunggah data pelatihan dari a ke S3. `DataFrame` Misalnya, `"sagemaker"` untuk format protobuf, untuk nilai yang dipisahkan koma, dan `"csv"` untuk `"libsvm"` format libSVM.

Anda dapat mengimplementasikan sendiri `RequestRowSerializer` dan `ResponseRowDeserializer` untuk membuat serial dan deserialisasi baris dari format data yang didukung kode inferensi Anda, seperti.libsvm atau.. csv.

Gunakan SageMakerEstimator dalam Pipa Spark

Anda dapat menggunakan `org.apache.spark.ml.Estimator` estimator dan `org.apache.spark.ml.Model` model, dan `SageMakerEstimator` estimator dan `SageMakerModel` model dalam `org.apache.spark.ml.Pipeline` pipeline, seperti yang ditunjukkan pada contoh berikut:

```
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.feature.PCA
import org.apache.spark.sql.SparkSession
import com.amazonaws.services.sagemaker.sparksdk.IAMRole
import com.amazonaws.services.sagemaker.sparksdk.algorithms
import com.amazonaws.services.sagemaker.sparksdk.algorithms.KMeansSageMakerEstimator

val spark = SparkSession.builder.getOrCreate

// load mnist data as a dataframe from libsvm
val region = "us-east-1"
val trainingData = spark.read.format("libsvm")
  .option("numFeatures", "784")
  .load(s"s3://sagemaker-sample-data-$region/spark/mnist/train/")
val testData = spark.read.format("libsvm")
  .option("numFeatures", "784")
  .load(s"s3://sagemaker-sample-data-$region/spark/mnist/test/")

// substitute your SageMaker IAM role here
val roleArn = "arn:aws:iam::account-id:role/rolename"

val pcaEstimator = new PCA()
  .setInputCol("features")
  .setOutputCol("projectedFeatures")
  .setK(50)

val kMeansSageMakerEstimator = new KMeansSageMakerEstimator(
  sagemakerRole = IAMRole(integTestingRole),
  requestRowSerializer =
    new ProtobufRequestRowSerializer(featuresColumnName = "projectedFeatures"),
  trainingSparkDataFormatOptions = Map("featuresColumnName" -> "projectedFeatures"),
  trainingInstanceType = "ml.p2.xlarge",
```

```
  trainingInstanceCount = 1,
  endpointInstanceType = "ml.c4.xlarge",
  endpointInitialInstanceCount = 1)
  .setK(10).setFeatureDim(50)

val pipeline = new Pipeline().setStages(Array(pcaEstimator, kMeansSageMakerEstimator))

// train
val pipelineModel = pipeline.fit(trainingData)

val transformedData = pipelineModel.transform(testData)
transformedData.show()
```

Parameter `trainingSparkDataFormatOptions` mengonfigurasi Spark untuk membuat serial ke protobuf kolom “ProjectedFeatures” untuk pelatihan model. Selain itu, Spark membuat serial untuk membuat protobuf kolom “label” secara default.

Karena kita ingin membuat kesimpulan menggunakan kolom “ProjectedFeatures”, kita meneruskan nama kolom ke dalam kolom. `ProtobufRequestRowSerializer`

Contoh berikut menunjukkan transformasi`DataFrame`:

```
+-----+-------------------+-------------------+------------------+---------------+
|label|           features|   projectedFeatures|distance_to_cluster|closest_cluster|
+-----+-------------------+-------------------+------------------+---------------+
|  5.0|(784,[152,153,154...|[880.731433034386...|     1500.470703125|            0.0|
|  0.0|(784,[127,128,129...|[1768.51722024166...|      1142.18359375|            4.0|
|  4.0|(784,[160,161,162...|[704.949236329314...|  1386.246826171875|            9.0|
|  1.0|(784,[158,159,160...|[-42.328192193771...| 1277.0736083984375|            5.0|
|  9.0|(784,[208,209,210...|[374.043902028333...|   1211.00927734375|            3.0|
|  2.0|(784,[155,156,157...|[941.267714528850...|  1496.157958984375|            8.0|
|  1.0|(784,[124,125,126...|[30.2848596410594...| 1327.6766357421875|            5.0|
|  3.0|(784,[151,152,153...|[1270.14374062052...| 1570.7674560546875|            0.0|
|  1.0|(784,[152,153,154...|[-112.10792566485...|     1037.568359375|            5.0|
|  4.0|(784,[134,135,161...|[452.068280676606...| 1165.1236572265625|            3.0|
|  3.0|(784,[123,124,125...|[610.596447285397...|  1325.953369140625|            7.0|
|  5.0|(784,[216,217,218...|[142.959601818422...| 1353.4930419921875|            5.0|
|  3.0|(784,[143,144,145...|[1036.71862533658...| 1460.4315185546875|            7.0|
|  6.0|(784,[72,73,74,99...|[996.740157435754...| 1159.8631591796875|            2.0|
|  1.0|(784,[151,152,153...|[-107.26076167417...|   960.963623046875|            5.0|
|  7.0|(784,[211,212,213...|[619.771820430940...|   1245.13623046875|            6.0|
|  2.0|(784,[151,152,153...|[850.152101817161...|  1304.437744140625|            8.0|
|  8.0|(784,[159,160,161...|[370.041887230547...| 1192.4781494140625|            0.0|
```

```
|  6.0|(784,[100,101,102...|[546.674328209335...|    1277.0908203125|           2.0|
|  9.0|(784,[209,210,211...|[-29.259112927426...| 1245.8182373046875|           6.0|
+-----+--------------------+--------------------+-------------------+--------------+
```

SageMaker Contoh percikan untuk Python PySpark ()

Amazon SageMaker menyediakan pustaka Apache Spark Python (SageMaker PySpark) yang dapat Anda gunakan untuk mengintegrasikan aplikasi Apache Spark Anda. SageMaker Misalnya, Anda dapat menggunakan Apache Spark untuk preprocessing data dan SageMaker untuk pelatihan model dan hosting. Untuk informasi tentang perpustakaan SageMaker Apache Spark, lihat. Gunakan Apache Spark dengan Amazon SageMaker

Unduh PySpark

Anda dapat mengunduh kode sumber untuk pustaka Python Spark (PySpark) dan Scala dari repositori Spark. SageMaker GitHub

Untuk petunjuk tentang menginstal pustaka SageMaker Spark, gunakan opsi berikut atau kunjungi SageMaker PySpark.

- Instal menggunakan pip:

```
pip install sagemaker_pyspark
```

- Instal dari sumbernya:

```
git clone git@github.com:aws/sagemaker-spark.git
cd sagemaker-pyspark-sdk
python setup.py install
```

- Anda juga dapat membuat notebook baru dalam instance notebook yang menggunakan kernel `Sparkmagic (PySpark)` atau `Sparkmagic (PySpark3)` kernel dan terhubung ke cluster EMR Amazon jarak jauh.

> **Note**
>
> Cluster EMR Amazon harus dikonfigurasi dengan peran IAM yang memiliki kebijakan terlampir. `AmazonSageMakerFullAccess` Untuk informasi tentang mengonfigurasi peran untuk kluster EMR, lihat Mengonfigurasi Peran IAM untuk Izin EMR Amazon ke AWS Layanan di Panduan Manajemen EMR Amazon.

PySpark contoh

Untuk contoh tentang penggunaan SageMaker PySpark, lihat:

- Menggunakan Amazon SageMaker dengan Apache Spark di Baca Dokumen.
- SageMaker GitHubRepositori percikan.

Untuk menjalankan notebook pada instance notebook, lihatContoh Notebook. Untuk menjalankan notebook di Studio, lihatMembuat atau Membuka Notebook Amazon SageMaker Studio Classic.

Gunakan Chainer dengan Amazon SageMaker

Anda dapat menggunakan SageMaker untuk melatih dan menerapkan model menggunakan kode Chainer khusus. Estimator dan model SageMaker Python SDK Chainer dan wadah Chainer SageMaker open-source membuat penulisan skrip Chainer dan menjalankannya lebih mudah. SageMaker

Apa yang ingin Anda lakukan?

Saya ingin melatih model Chainer khusus. SageMaker

> Untuk contoh notebook Jupyter, lihat contoh notebook Chainer di repositori Amazon Examples. SageMaker GitHub
>
> Untuk dokumentasi, lihat Melatih Model dengan Chainer.

Saya memiliki model Chainer yang saya latih SageMaker, dan saya ingin menerapkannya ke titik akhir yang dihosting.

> Untuk informasi selengkapnya, lihat Men-deploy model Chainer.

Saya memiliki model Chainer yang saya latih di luar SageMaker, dan saya ingin menerapkannya ke titik akhir SageMaker

> Untuk informasi selengkapnya, lihat Men-deploy Endpoints dari Data Model.

Saya ingin melihat dokumentasi API untuk kelas Amazon SageMaker Python SDK Chainer.

> Untuk informasi selengkapnya, lihat Kelas Chainer.

Saya ingin mencari informasi tentang wadah SageMaker Chainer.

> Untuk informasi selengkapnya, lihat GitHub repositori SageMaker Chainer Container.

Untuk informasi tentang versi Chainer yang didukung, dan untuk informasi umum tentang menulis skrip pelatihan Chainer dan menggunakan estimator dan model Chainer dengan SageMaker, lihat Menggunakan Chainer dengan Python SDK. SageMaker

Gunakan Hugging Face dengan Amazon SageMaker

Amazon SageMaker memungkinkan pelanggan untuk melatih, menyempurnakan, dan menjalankan inferensi menggunakan model Hugging Face untuk Natural Language Processing (NLP). SageMaker Anda dapat menggunakan Hugging Face untuk pelatihan dan inferensi. Fungsi ini tersedia melalui pengembangan Hugging AWSFace Deep Learning Containers. Wadah ini termasuk Hugging Face Transformers, Tokenizers, dan pustaka Datasets, yang memungkinkan Anda menggunakan sumber daya ini untuk pekerjaan pelatihan dan inferensi Anda. Untuk daftar gambar Deep Learning Containers yang tersedia, lihat Gambar Deep Learning Containers yang Tersedia. Gambar Deep Learning Containers ini dipelihara dan diperbarui secara berkala dengan patch keamanan.

Untuk menggunakan Hugging Face Deep Learning Containers dengan SageMaker Python SDK untuk pelatihan, lihat Hugging Face Estimator. SageMaker Dengan Hugging Face Estimator, Anda dapat menggunakan model Hugging Face seperti halnya Estimator lainnya. SageMaker Namun, menggunakan SageMaker Python SDK adalah opsional. Anda juga dapat mengatur penggunaan Hugging Face Deep Learning Containers dengan dan. AWS CLI AWS SDK for Python (Boto3)

Untuk informasi lebih lanjut tentang Hugging Face dan model yang tersedia di dalamnya, lihat dokumentasi Hugging Face.

Pelatihan

Untuk menjalankan pelatihan, Anda dapat menggunakan salah satu dari ribuan model yang tersedia di Hugging Face dan menyempurnakannya untuk kasus penggunaan khusus Anda dengan pelatihan tambahan. Dengan SageMaker, Anda dapat menggunakan pelatihan standar atau memanfaatkan Data SageMaker Terdistribusi dan pelatihan Paralel Model. Seperti pekerjaan SageMaker pelatihan lainnya menggunakan kode kustom, Anda dapat menangkap metrik Anda sendiri dengan meneruskan definisi metrik ke SDK Python seperti yang ditunjukkan dalam Mendefinisikan Metrik Pelatihan (SageMaker Python SDK). SageMaker Metrik yang ditangkap kemudian dapat diakses melalui CloudWatchdan sebagai Panda `DataFrame` melalui metode. TrainingJobAnalytics Setelah model Anda dilatih dan disetel dengan baik, Anda dapat menggunakannya seperti model lain untuk menjalankan pekerjaan inferensi.

Cara menjalankan pelatihan dengan Hugging Face Estimator

Anda dapat mengimplementasikan Hugging Face Estimator untuk pekerjaan pelatihan menggunakan SageMaker Python SDK. SageMaker Python SDK adalah pustaka open source untuk pelatihan dan penerapan model pembelajaran mesin. SageMaker Untuk informasi selengkapnya tentang Hugging Face Estimator, lihat dokumentasi SageMakerPython SDK.

Dengan SageMaker Python SDK, Anda dapat menjalankan pekerjaan pelatihan menggunakan Hugging Face Estimator di lingkungan berikut:

- SageMakerStudio: Amazon SageMaker Studio adalah lingkungan pengembangan terintegrasi penuh pertama (IDE) untuk pembelajaran mesin (ML). SageMaker Studio menyediakan antarmuka visual tunggal berbasis web di mana Anda dapat melakukan semua langkah pengembangan ML yang diperlukan untuk menyiapkan, membangun, melatih, dan menyetel, menyebarkan, dan mengelola model. Untuk informasi tentang penggunaan Notebook Jupyter di Studio, lihat Menggunakan Notebook Amazon SageMaker Studio.
- SageMaker Instans Notebook: Instans SageMaker notebook Amazon adalah instans komputasi pembelajaran mesin (MS) yang menjalankan Aplikasi Notebook Jupyter. Aplikasi ini memungkinkan Anda menjalankan Jupyter Notebooks di instance notebook Anda untuk menyiapkan dan memproses data, menulis kode untuk melatih model, menyebarkan model ke SageMaker hosting, dan menguji atau memvalidasi model Anda tanpa fitur SageMaker Studio seperti Debugger, Model Monitoring, dan IDE berbasis web.
- Secara lokal: Jika Anda memiliki konektivitas AWS dan memiliki SageMaker izin yang sesuai, Anda dapat menggunakan SageMaker Python SDK secara lokal untuk meluncurkan pelatihan jarak jauh dan pekerjaan inferensi untuk Hugging Face in on on. SageMaker AWS Ini berfungsi pada mesin lokal Anda, serta AWS layanan lain dengan SDK SageMaker Python yang terhubung dan izin yang sesuai.

Inferensi

Untuk inferensi, Anda dapat menggunakan model Hugging Face terlatih atau salah satu model Hugging Face yang telah dilatih sebelumnya untuk menggunakan pekerjaan inferensi. SageMaker Dengan kolaborasi ini, Anda hanya perlu satu baris kode untuk menyebarkan model terlatih dan model pra-terlatih. SageMaker Anda juga dapat menjalankan pekerjaan inferensi tanpa harus menulis kode inferensi khusus apa pun. Dengan kode inferensi kustom, Anda dapat menyesuaikan logika inferensi dengan menyediakan skrip Python Anda sendiri.

Cara menerapkan pekerjaan inferensi menggunakan Hugging Face Deep Learning Containers

Anda memiliki dua pilihan untuk menjalankan inferensi dengan SageMaker. Anda dapat menjalankan inferensi menggunakan model yang Anda latih, atau menerapkan model Hugging Face yang telah dilatih sebelumnya.

- Jalankan inferensi dengan model terlatih Anda: Anda memiliki dua opsi untuk menjalankan inferensi dengan model terlatih Anda sendiri. Anda dapat menjalankan inferensi dengan model yang Anda latih menggunakan model Hugging Face yang ada dengan Hugging Face Deep Learning Containers, atau Anda dapat membawa model Hugging Face Anda sendiri dan menggunakannya. SageMaker SageMaker Saat menjalankan inferensi dengan model yang Anda latih dengan Pengukur Wajah SageMaker Pelukan, Anda dapat menerapkan model segera setelah pelatihan selesai atau Anda dapat mengunggah model terlatih ke bucket Amazon S3 dan menelannya saat menjalankan inferensi nanti. Jika Anda membawa model Hugging Face sendiri yang sudah ada, Anda harus mengunggah model terlatih ke bucket Amazon S3 dan menelan bucket tersebut saat menjalankan inferensi seperti yang ditunjukkan pada Deploy your Hugging Face Transformers untuk contoh inferensi.
- Jalankan inferensi dengan HuggingFace model pra-terlatih: Anda dapat menggunakan salah satu dari ribuan model Hugging Face yang telah dilatih sebelumnya untuk menjalankan pekerjaan inferensi Anda tanpa memerlukan pelatihan tambahan. Untuk menjalankan inferensi, Anda memilih model yang telah dilatih sebelumnya dari daftar model Hugging Face, seperti yang diuraikan dalam Contoh Deploy Hugging Face Transformers yang telah dilatih sebelumnya untuk inferensi.

Apa yang ingin Anda lakukan?

Notebook Jupyter berikut di repositori notebook Hugging Face menggambarkan cara menggunakan Hugging Face Deep Learning Containers dengan dalam berbagai kasus penggunaan. SageMaker

Saya ingin melatih dan menerapkan model klasifikasi teks menggunakan Hugging Face in with. SageMaker PyTorch

> Untuk contoh Notebook Jupyter, lihat Demo PyTorch Memulai.

Saya ingin melatih dan menerapkan model klasifikasi teks menggunakan Hugging Face in with. SageMaker TensorFlow

> Untuk contoh Notebook Jupyter, lihat contoh TensorFlow Memulai.

Saya ingin menjalankan pelatihan terdistribusi dengan paralelisme data menggunakan Hugging Face dan Distributed. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Pelatihan Terdistribusi.

Saya ingin menjalankan pelatihan terdistribusi dengan paralelisme model menggunakan Hugging Face dan Distributed. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Paralelisme Model.

Saya ingin menggunakan instance spot untuk melatih dan menerapkan model menggunakan Hugging Face in. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Instans Spot.

Saya ingin menangkap metrik khusus dan menggunakan SageMaker Checkpointing saat melatih model klasifikasi teks menggunakan Hugging Face in. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Pelatihan dengan Metrik Kustom.

Saya ingin melatih model penjawab pertanyaan terdistribusi menggunakan TensorFlow Hugging Face in. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh TensorFlow Pelatihan Terdistribusi.

Saya ingin melatih model ringkasan terdistribusi menggunakan Hugging Face in. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Pelatihan Ringkasan Terdistribusi.

Saya ingin melatih model klasifikasi gambar menggunakan Hugging Face SageMaker in.

Untuk contoh Notebook Jupyter, lihat contoh Pelatihan Transformer Visi.

Saya ingin menerapkan model Hugging Face terlatih saya di. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Deploy your Hugging Face Transformers untuk inferensi.

Saya ingin menerapkan model Hugging Face yang telah dilatih sebelumnya. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Deploy Hugging Face Transformers yang telah dilatih sebelumnya untuk contoh inferensi.

Gunakan PyTorch dengan Amazon SageMaker

Anda dapat menggunakan Amazon SageMaker untuk melatih dan menerapkan model menggunakan PyTorch kode khusus. PyTorch Penaksir dan model SDK SageMaker Python serta PyTorch wadah

SageMaker sumber terbuka membuat penulisan PyTorch skrip dan menjalankannya menjadi lebih mudah. SageMaker

Apa yang ingin Anda lakukan?

Saya ingin melatih PyTorch model khusus SageMaker.

Untuk contoh notebook Jupyter, lihat PyTorch contoh notebook di repositori Amazon SageMaker Examples GitHub.

Untuk dokumentasi, lihat Melatih Model dengan PyTorch.

Saya memiliki PyTorch model yang saya latih SageMaker, dan saya ingin menerapkannya ke titik akhir yang dihosting.

Untuk informasi selengkapnya, lihat Men-deploy PyTorch model.

Saya memiliki PyTorch model yang saya latih di luar SageMaker, dan saya ingin menerapkannya ke titik akhir SageMaker

Untuk informasi selengkapnya, lihat Men-deploy PyTorch model Anda sendiri.

Saya ingin melihat dokumentasi API untuk kelas Amazon SageMaker Python SDK PyTorch.

Untuk informasi selengkapnya, lihat PyTorch Kelas.

Saya ingin menemukan repositori SageMaker PyTorch kontainer.

Untuk informasi selengkapnya, lihat GitHub Repositori SageMaker PyTorch kontainer.

Saya ingin mencari informasi tentang PyTorch versi yang didukung oleh AWS Deep Learning Containers.

Untuk informasi selengkapnya, lihat Gambar Kontainer Pembelajaran Mendalam yang Tersedia.

Untuk informasi umum tentang menulis skrip PyTorch pelatihan dan menggunakan PyTorch estimator dan model dengan SageMaker, lihat Menggunakan PyTorch dengan Python SageMaker SDK.

R Panduan Pengguna untuk Amazon SageMaker

Dokumen ini akan memandu Anda melalui cara-cara memanfaatkan SageMaker fitur Amazon menggunakan R. Panduan ini memperkenalkan SageMaker kernel R bawaan, cara memulai dengan R on SageMaker, dan akhirnya beberapa contoh notebook.

Contoh disusun dalam tiga tingkatan, Pemula, Menengah, dan Lanjutan. Mereka mulai dari Memulai dengan R on SageMaker, melanjutkan pembelajaran end-to-end mesin dengan R on SageMaker, dan kemudian menyelesaikan dengan topik yang lebih maju seperti SageMaker Processing with R script, dan algoritma Bring-Your-Own (BYO) R ke. SageMaker

Untuk informasi tentang cara membawa gambar R kustom Anda sendiri ke Studio, lihatBawa SageMaker gambar Anda sendiri. Untuk artikel blog serupa, lihat Membawa lingkungan R Anda sendiri ke Amazon SageMaker Studio.

RStudio Support di SageMaker

Amazon SageMaker mendukung RStudio sebagai lingkungan pengembangan terintegrasi (IDE) yang dikelola sepenuhnya yang terintegrasi dengan Domain Amazon. SageMaker Dengan integrasi RStudio, Anda dapat meluncurkan lingkungan RStudio di Domain untuk menjalankan alur kerja RStudio Anda pada sumber daya. SageMaker Untuk informasi selengkapnya, lihat RStudio di Amazon SageMaker.

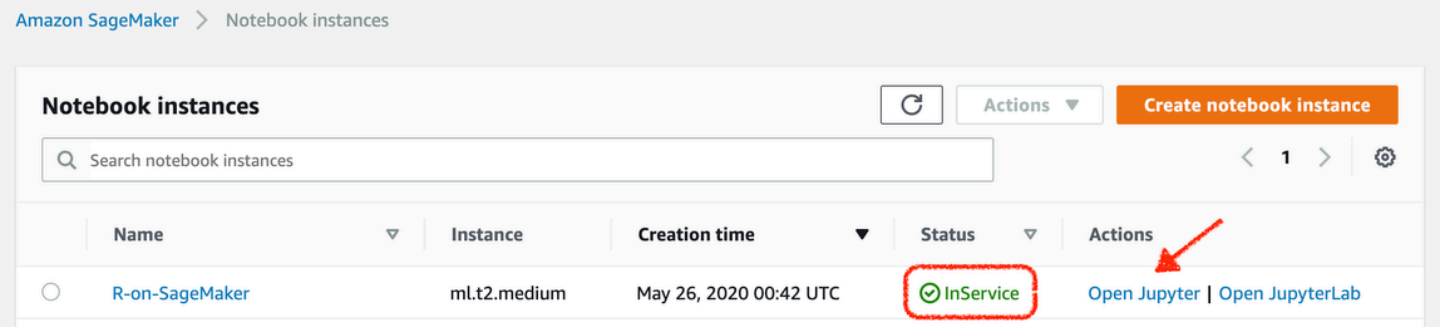
R Kernel di SageMaker

SageMaker instance notebook mendukung R menggunakan kernel R yang sudah diinstal sebelumnya. Selain itu, kernel R memiliki perpustakaan retikulat, antarmuka R ke Python, sehingga Anda dapat menggunakan fitur SageMaker Python SDK dari dalam skrip R.

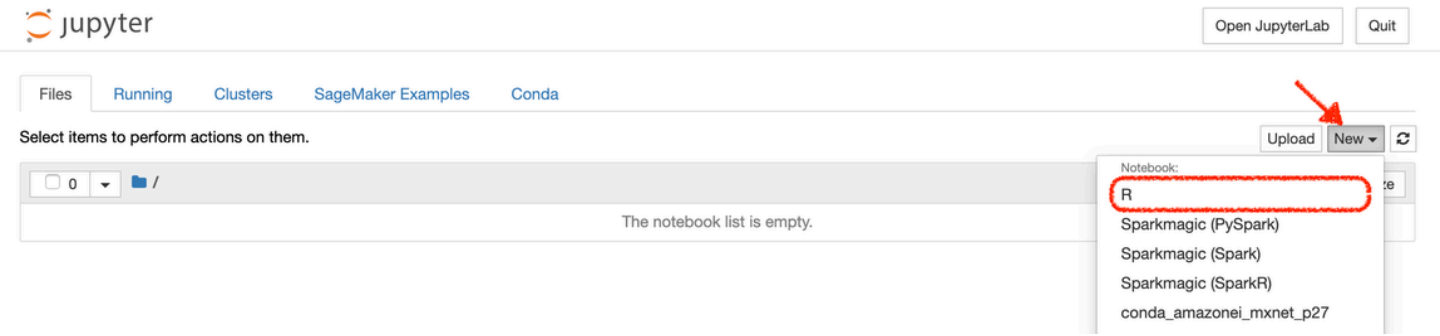
- reticulatelibrary: menyediakan antarmuka R ke Amazon Python SDK. SageMaker Paket retikulat menerjemahkan antara objek R dan Python.

Memulai dengan R di SageMaker

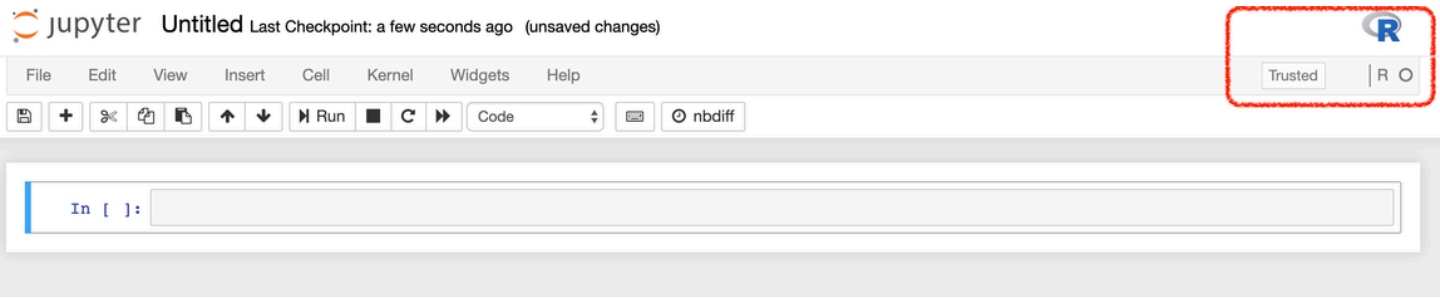
- Buat Instance Notebook menggunakan tipe instans t2.medium dan ukuran penyimpanan default. Anda dapat memilih instans yang lebih cepat dan lebih banyak penyimpanan jika Anda berencana untuk terus menggunakan instance untuk contoh yang lebih maju, atau membuat instance yang lebih besar nanti.
- Tunggu hingga status buku catatan dalam layanan, lalu klik Buka Jupyter.



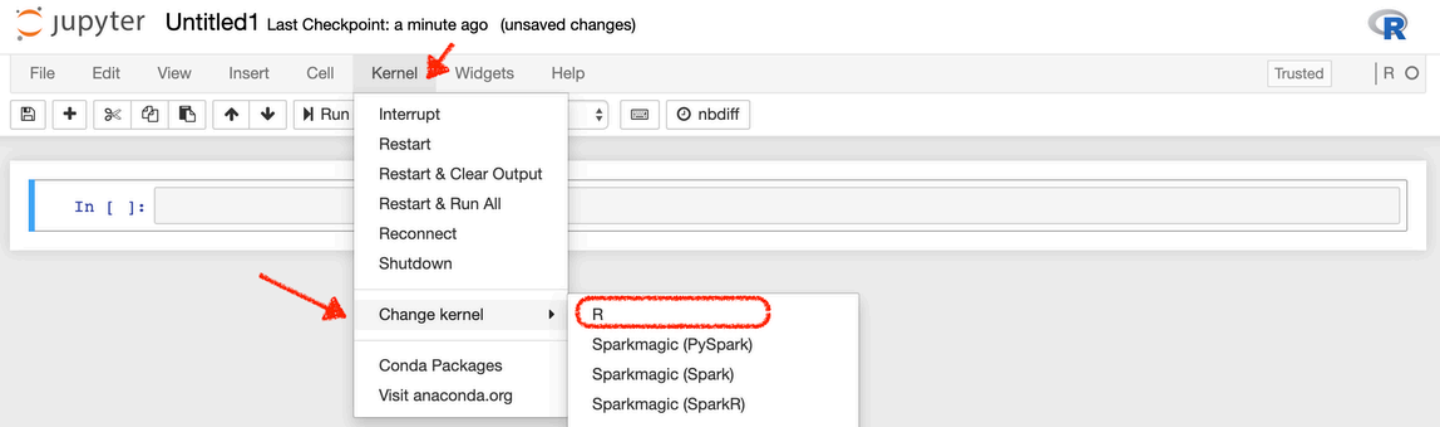
- Buat notebook baru dengan kernel R dari daftar lingkungan yang tersedia.



- Ketika notebook baru dibuat, Anda akan melihat logo R di sudut kanan atas lingkungan notebook, dan juga R sebagai kernel di bawah logo itu. Ini menunjukkan bahwa SageMaker telah berhasil meluncurkan kernel R untuk notebook ini.



- Atau, ketika Anda berada di notebook Jupyter, Anda dapat menggunakan menu Kernel, dan kemudian pilih R dari opsi Ubah Kernel.



Contoh Notebook

Prasyarat

Memulai R aktif SageMaker: Contoh buku catatan ini menjelaskan bagaimana Anda dapat mengembangkan skrip R menggunakan kernel R SageMaker Amazon. Di buku catatan ini Anda mengatur SageMaker lingkungan dan izin, mengunduh dataset abalon dari UCI Machine Learning Repository, lakukan beberapa pemrosesan dasar dan visualisasi pada data, lalu simpan data sebagai format.csv ke S3.

Tingkat Pemula

SageMakerBatch Transform menggunakan R Kernel: Contoh Notebook ini menjelaskan cara melakukan pekerjaan transformasi batch menggunakan SageMaker Transformer API dan algoritma XGBoost. Notebook ini juga menggunakan dataset Abalone.

Tingkat Menengah

Optimasi Hyperparameter untuk XGBoost di R: Notebook contoh ini memperluas notebook pemula sebelumnya yang menggunakan dataset abalone dan XGBoost. Ini menjelaskan bagaimana melakukan penyetelan model dengan optimasi hyperparameter. Anda juga akan belajar cara menggunakan transformasi batch untuk prediksi batch, serta cara membuat titik akhir model untuk membuat prediksi waktu nyata.

Amazon SageMaker Processing dengan R: SageMakerProcessing memungkinkan Anda melakukan pra-proses, pasca-proses, dan menjalankan beban kerja evaluasi model. Contoh ini menunjukkan kepada Anda cara membuat skrip R untuk mengatur pekerjaan Processing.

Tingkat Lanjut

Latih dan Terapkan Algoritma R Anda Sendiri di SageMaker: Apakah Anda sudah memiliki algoritma R, dan Anda ingin membawanya SageMaker untuk menyetel, melatih, atau menerapkannya? Contoh ini memandu Anda melalui cara menyesuaikan SageMaker kontainer dengan paket R khusus, hingga menggunakan titik akhir yang dihosting untuk inferensi pada model R-origin Anda.

Gunakan Scikit-Learn dengan Amazon SageMaker

Anda dapat menggunakan Amazon SageMaker untuk melatih dan menerapkan model menggunakan kode Scikit-learn kustom. Penaksir dan model SageMaker Python SDK Scikit-learn serta wadah Scikit-learn SageMaker sumber terbuka membuat penulisan skrip Scikit-learn dan menjalankannya menjadi lebih mudah. SageMaker

Persyaratan

Scikit-learn 1.2 memiliki dependensi berikut.

Dependensi	Versi minimum
Python	3.8
NumPy	1.17.3
SciPy	1.3.2
joblib	1.1.1
threadpoolctl	2.0.0

Container SageMaker Scikit-learn mendukung versi Scikit-learn berikut.

Versi Scikit-learn yang didukung	Versi Python minimum
`1.2-1`	`3.8`
`1.0-1`	`3.7`
`0.23-1`	`3.6`
`0.20.0`	`2.7` atau `3.4`

Untuk informasi umum tentang menulis skrip pelatihan Scikit-learn dan menggunakan estimator dan model Scikit-learn dengan, SageMaker lihat Menggunakan Scikit-learn dengan Python SDK. SageMaker

Apa yang ingin Anda lakukan?

> **Note**
>
> Matplotlib v2.2.3 atau yang lebih baru diperlukan untuk menjalankan contoh notebook Scikit-learn. SageMaker

Saya ingin menggunakan Scikit-learn untuk pemrosesan data, rekayasa fitur, atau evaluasi model di. SageMaker

Untuk contoh notebook Jupyter, lihat https://github.com/awslabs/ amazon-sagemaker-examples / tree/master/sagemaker_processing/scikit_learn_data_processing_and_model_evaluation.

Untuk dokumentasi, lihat ReadTheDocs.

Saya ingin melatih model Scikit-learn khusus di. SageMaker

Untuk contoh notebook Jupyter, lihat https://github.com/awslabs/ amazon-sagemaker-examples / tree/master/ sagemaker-python-sdk /scikit_learn_iris.

Untuk dokumentasi, lihat Melatih Model dengan Scikit-learn.

Saya memiliki model Scikit-learn yang saya latih SageMaker, dan saya ingin menerapkannya ke titik akhir yang dihosting.

Untuk informasi selengkapnya, lihat Men-deploy model Scikit-learn.

Saya memiliki model Scikit-learn yang saya latih di luar SageMaker, dan saya ingin menerapkannya ke titik akhir SageMaker

Untuk informasi selengkapnya, lihat Men-deploy Endpoints dari Data Model.

Saya ingin melihat dokumentasi API untuk kelas Amazon SageMaker Python SDK Scikit-learn.

Untuk informasi selengkapnya, lihat Kelas Scikit-Learn.

Saya ingin melihat informasi tentang wadah SageMaker Scikit-learn.

Untuk informasi selengkapnya, lihat SageMaker Scikit-learn Container repository. GitHub

Gunakan Penyajian SparkMl dengan Amazon SageMaker

Model dan prediktor Penyajian Amazon SageMaker Python SDK SparkMl dan wadah Penyajian SparkMl sumber terbuka SageMaker Amazon mendukung penerapan pipeline Apache Spark ML yang diserialkan dengan mLeap untuk mendapatkan kesimpulan. SageMaker

Untuk informasi tentang penggunaan kontainer SparkMl Serving untuk menyebarkan model SageMaker, SageMaker lihat Repositori Spark ML Container. GitHub Untuk informasi tentang model dan prediktor Penyajian Amazon SageMaker Python SDK SparkMl, lihat dokumentasi SparkMl Serving Model dan Predictor API.

Gunakan TensorFlow dengan Amazon SageMaker

Anda dapat menggunakan Amazon SageMaker untuk melatih dan menerapkan model menggunakan TensorFlow kode khusus. TensorFlow Penaksir dan model SDK SageMaker Python serta TensorFlow wadah SageMaker sumber terbuka membuat penulisan TensorFlow skrip dan menjalankannya menjadi lebih mudah. SageMaker

Gunakan TensorFlow Versi 1.11 dan yang lebih baru

Untuk TensorFlow versi 1.11 dan yang lebih baru, Amazon SageMaker Python SDK mendukung skrip pelatihan mode skrip.

Apa yang ingin Anda lakukan?

Saya ingin melatih TensorFlow model khusus SageMaker.

Untuk contoh notebook Jupyter, lihat pelatihan dan TensorFlow penyajian mode skrip.

Untuk dokumentasi, lihat Melatih Model dengan TensorFlow.

Saya memiliki TensorFlow model yang saya latih SageMaker, dan saya ingin menerapkannya ke titik akhir yang dihosting.

Untuk informasi selengkapnya, lihat Men-deploy model TensorFlow Serving.

Saya memiliki TensorFlow model yang saya latih di luar SageMaker, dan saya ingin menerapkannya ke titik akhir SageMaker

Untuk informasi selengkapnya, lihat Men-deploy langsung dari artefak model.

Saya ingin melihat dokumentasi API untuk kelas Amazon SageMaker Python SDK TensorFlow.

Untuk informasi selengkapnya, lihat TensorFlow Penaksir.

Saya ingin menemukan repositori SageMaker TensorFlow kontainer.

Untuk informasi selengkapnya, lihat GitHub Repositori SageMaker TensorFlow kontainer.

Saya ingin mencari informasi tentang TensorFlow versi yang didukung oleh AWS Deep Learning Containers.

Untuk informasi selengkapnya, lihat Gambar Kontainer Pembelajaran Mendalam yang Tersedia.

Untuk informasi umum tentang menulis TensorFlow skrip pelatihan mode skrip dan menggunakan estimator mode TensorFlow skrip dan model dengan SageMaker, lihat Menggunakan TensorFlow dengan Python SageMaker SDK.

Gunakan Mode TensorFlow Legacy untuk Versi 1.11 dan Sebelumnya

Amazon SageMaker Python SDK menyediakan mode lama yang mendukung TensorFlow versi 1.11 dan sebelumnya. Gunakan skrip TensorFlow pelatihan mode lama untuk menjalankan TensorFlow pekerjaan di SageMaker if:

- Anda memiliki skrip mode lama yang tidak ingin Anda konversi ke mode skrip.
- Anda ingin menggunakan TensorFlow versi lebih awal dari 1.11.

Untuk informasi tentang menulis TensorFlow skrip mode lama yang akan digunakan dengan SageMaker Python SDK, lihat Estimator dan Model. TensorFlow SageMaker

Gunakan Server Inferensi Triton dengan Amazon SageMaker

SageMaker memungkinkan pelanggan untuk menerapkan model menggunakan kode khusus dengan NVIDIA Triton Inference Server. Fungsi ini tersedia melalui pengembangan Triton Inference Server Containers. Kontainer ini termasuk NVIDIA Triton Inference Server, dukungan untuk kerangka kerja MS umum, dan variabel lingkungan berguna yang memungkinkan Anda mengoptimalkan kinerja. SageMaker Untuk daftar semua gambar Deep Learning Containers yang tersedia, lihat Gambar Deep Learning Containers yang Tersedia. Gambar Deep Learning Containers dipertahankan dan diperbarui secara berkala dengan patch keamanan.

Anda dapat menggunakan Triton Inference Server Container dengan SageMaker Python SDK seperti halnya wadah lain dalam model Anda. SageMaker Namun, menggunakan SageMaker Python SDK adalah opsional. Anda dapat menggunakan Triton Inference Server Containers dengan dan. AWS CLI AWS SDK for Python (Boto3)

Untuk informasi lebih lanjut tentang NVIDIA Triton Inference Server lihat dokumentasi Triton.

Inferensi

> **Note**
>
> Backend Triton Python menggunakan memori bersama (SHMEM) untuk menghubungkan kode Anda ke Triton. SageMaker Inferensi menyediakan hingga setengah dari memori instance sebagai SHMEM sehingga Anda dapat menggunakan instance dengan lebih banyak memori untuk ukuran SHMEM yang lebih besar.

Untuk inferensi, Anda dapat menggunakan model MLmu yang terlatih dengan Triton Inference Server untuk menerapkan pekerjaan inferensi. SageMaker

Beberapa fitur utama dari Triton Inference Server Container adalah:

- Support untuk beberapa kerangka kerja: Triton dapat digunakan untuk menyebarkan model dari semua kerangka kerja MS utama. Triton mendukung TensorFlow GraphDef dan SavedModel, ONNX,, PyTorch TorchScript TensorRT, dan format model Python/C++ kustom.
- Pipa model: Ansambel model Triton mewakili pipa dari satu model dengan logika pemrosesan pra/pasca dan koneksi tensor input dan output di antara keduanya. Permintaan inferensi tunggal ke ansambel memicu eksekusi seluruh pipeline.
- Eksekusi model bersamaan: Beberapa instance dari model yang sama dapat berjalan secara bersamaan pada GPU yang sama atau pada beberapa GPU.
- Batching dinamis: Untuk model yang mendukung batching, Triton memiliki beberapa algoritma penjadwalan dan batching bawaan yang menggabungkan permintaan inferensi individu bersama-sama untuk meningkatkan throughput inferensi. Keputusan penjadwalan dan batching ini transparan bagi klien yang meminta inferensi.
- Dukungan CPU dan GPU yang beragam: Model dapat dijalankan pada CPU atau GPU untuk fleksibilitas maksimum dan untuk mendukung persyaratan komputasi heterogen.

Apa yang ingin Anda lakukan?

Saya ingin menerapkan PyTorch model terlatih saya di SageMaker.

Untuk contoh Notebook Jupyter, lihat contoh Menerapkan model PyTorch Resnet50 Anda dengan Triton Inference Server contoh.

Saya ingin menerapkan model Hugging Face terlatih saya di. SageMaker

Untuk contoh Notebook Jupyter, lihat contoh Menerapkan model PyTorch BERT Anda dengan Triton Inference Server contoh.

Referensi API

Membuat panggilan API langsung dari kode tidak praktis, dan mengharuskan Anda menulis kode untuk mengautentikasi permintaan Anda. Amazon SageMaker menyediakan alternatif berikut:

Topik

- Model Pemrograman untuk Amazon SageMaker
- API, CLI, dan SDKs

Model Pemrograman untuk Amazon SageMaker

Membuat panggilan API langsung dari kode tidak praktis, dan mengharuskan Anda menulis kode untuk mengautentikasi permintaan Anda. Amazon SageMaker menyediakan alternatif berikut:

- Gunakan SageMaker konsol —Dengan konsol, Anda tidak menulis kode apa pun. Anda menggunakan UI konsol untuk memulai pelatihan model atau menerapkan model. Konsol berfungsi dengan baik untuk pekerjaan sederhana, di mana Anda menggunakan algoritme pelatihan bawaan dan Anda tidak perlu memproses data pelatihan terlebih dahulu.

- Ubah contoh notebook Jupyter — SageMaker menyediakan beberapa notebook Jupyter yang melatih dan menyebarkan model menggunakan algoritme dan kumpulan data tertentu. Mulailah dengan notebook yang memiliki algoritme yang sesuai dan modifikasi untuk mengakomodasi sumber data dan kebutuhan spesifik Anda.

- Menulis pelatihan model dan kode inferensi dari awal — SageMaker menyediakan beberapa bahasa AWS SDK (tercantum dalam ikhtisar) dan Amazon SageMaker Python SDK, pustaka Python tingkat tinggi yang dapat Anda gunakan dalam kode Anda untuk memulai pekerjaan pelatihan model dan menerapkan model yang dihasilkan.

- SageMaker Python SDK —Pustaka Python ini menyederhanakan pelatihan dan penerapan model. Selain mengautentikasi permintaan Anda, pustaka mengabstraksi spesifik platform dengan menyediakan metode sederhana dan parameter default. Sebagai contoh:

  - Untuk menerapkan model Anda, Anda hanya memanggil `deploy()` metode. Metode ini membuat artefak SageMaker model, konfigurasi titik akhir, lalu menyebarkan model pada titik akhir.

  - Jika Anda menggunakan skrip kerangka kerja khusus untuk pelatihan model, Anda memanggil `fit()` metode. Metode ini membuat file.gzip skrip Anda, mengunggahnya ke lokasi Amazon S3, lalu menjalankannya untuk pelatihan model, dan tugas lainnya. Untuk informasi selengkapnya, lihat Kerangka Kerja dan Bahasa Machine Learning.

  - Untuk menetapkan default panggilan SageMaker API yang dibuat oleh SageMaker Python SDK, Anda menggunakan kamus konfigurasi default. Untuk informasi selengkapnya, lihat Mengkonfigurasi dan menggunakan default dengan Python SDK. SageMaker

- AWSSDK — SDK menyediakan metode yang sesuai dengan SageMaker API (lihat `Operations`). Gunakan SDK untuk memulai pekerjaan pelatihan model secara terprogram dan menjadi tuan rumah model. SageMaker Klien SDK menangani otentikasi untuk Anda, jadi Anda tidak perlu menulis kode otentikasi. Mereka tersedia dalam berbagai bahasa dan platform. Untuk informasi selengkapnya, lihat daftar sebelumnya di ikhtisar.

DiMemulai, Anda melatih dan menerapkan model menggunakan algoritme yang disediakan oleh SageMaker. Latihan itu menunjukkan cara menggunakan kedua perpustakaan ini. Untuk informasi selengkapnya, lihat Memulai.

- Integrasikan SageMaker ke dalam alur kerja Apache Spark Anda — SageMaker menyediakan perpustakaan untuk memanggil API-nya dari Apache Spark. Dengan itu, Anda dapat menggunakan estimator SageMaker berbasis di pipeline Apache Spark. Untuk informasi selengkapnya, lihat Gunakan Apache Spark dengan Amazon SageMaker.

API, CLI, dan SDKs

Amazon SageMaker menyediakan API, SDK, dan antarmuka baris perintah yang dapat Anda gunakan untuk membuat dan mengelola instance notebook serta melatih serta menerapkan model.

- Amazon SageMaker Python SDK (Direkomendasikan)
- Amazon SageMaker API Referensi
- Referensi API AI Augmented AI Amazon
- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto)
- AWS SDK for Ruby
- Amazon SageMaker Spark

Anda juga bisa mendapatkan contoh kode dari GitHub repositori notebook SageMaker contoh Amazon.

- Contoh notebook

SageMaker Gambar Distribusi

SageMaker Distribusi adalah kumpulan gambar Docker, yang mencakup perpustakaan dan paket populer untuk pembelajaran mesin, ilmu data, dan visualisasi analitik data. Gambar Docker mencakup kerangka kerja pembelajaran mendalam seperti berikut ini:

- PyTorch
- TensorFlow
- Keras

Ini juga termasuk paket Python populer seperti berikut ini:

- numpy
- scikit-belajar
- panda

Di dalam wadah, Anda dapat menggunakan IDE berikut:

- JupyterLab
- Editor Kode, berdasarkan Code-OSS (Visual Studio Code Open Source)

Setiap gambar SageMaker Distribusi memiliki varian GPU dan varian CPU.

SageMaker Distribusi tersedia di:

- Studio
- Lab Studio

Paket yang disertakan dalam wadah dijamin kompatibel satu sama lain dan runtime dibuat untuk bekerja di mana saja. Anda dapat menggunakan penampung untuk menjalankan notebook Amazon SageMaker Studio atau pekerjaan SageMaker pelatihan. Anda juga dapat menjalankan wadah di laptop lokal. Gunakan SageMaker Distribusi untuk memulai pengembangan ML dengan cepat di lingkungan lokal Anda. Transisi mulus ke tugas seperti eksekusi batch pekerjaan pelatihan tanpa perlu mengkonfigurasi ulang lingkungan runtime Anda.

Untuk daftar semua pustaka yang didukung dalam SageMaker distribusi dan versi yang sesuai, lihat SageMakerDistribusi GitHub. Anda juga dapat menggunakan gambar pra-bangun dan ready-to-use SageMaker Distribusi dari Amazon Elastic Container Registry Gallery.

Paket dan versi yang didukung

Untuk daftar paket yang diinstal dalam versi SageMaker Distribusi, lihat file Release.md di direktori build_artifacts dari repositori Distribution. SageMaker GitHub

SageMaker Kebijakan Dukungan Gambar Distribusi

Mayor	Rilis versi utama SageMaker	Setengah tahun

	Distribusi Amazon meningkatkan semua dependensi intinya ke versi terbaru yang kompatibel. SageMaker Distribusi dapat menambah atau menghapus paket dalam rilis versi utama. Versi utama dilambangkan dengan angka pertama dalam string versi. Misalnya, 1.0, 2.0, 3.0.	
Minor	Rilis versi minor SageMaker Distribusi Amazon memastikan bahwa semua dependensi intinya diperbarui ke versi minor terbaru yang kompatibel dalam versi utama yang sama. SageMaker Distribusi dapat menambahkan paket baru selama rilis versi minor. Versi minor dilambangkan dengan angka kedua dalam string versi. Misalnya, 1.1, 1.2, atau 2.1	Bulanan (versi minor tambahan dirilis berdasarkan kebutuhan tambahan juga)

Patch	Rilis versi patch dari Amazon SageMaker Distribution memastikan bahwa semua dependensi intinya diperbarui ke versi patch terbaru yang kompatibel dalam versi minor yang sama. SageMaker Distribusi tidak menambah atau menghapus paket selama rilis versi patch.	7 hari (perbaikan semalam juga diterapkan berdasarkan tingkat keparahan)

Important

- SageMaker Distribusi v0.xy hanya digunakan di Studio Classic. SageMaker Distribusi v1.x.y hanya digunakan di. JupyterLab
- Kami mencoba memperbarui gambar Studio dengan versi baru secara teratur. Jika paket dalam gambar Distribusi sudah kedaluwarsa, kami sarankan menunggu pembaruan berikutnya.
- Beberapa dependensi, seperti Python, diperlakukan berbeda. SageMakerDistribusi Amazon memungkinkan peningkatan kecil Python dengan rilis. Misalnya, Anda dapat meng-upgrade Python 3.10 ke Python 3.11 ketika Anda meng-upgrade dari versi 4.8 ke 5.0.

Riwayat Dokumen untuk Amazon SageMaker

Perubahan	Deskripsi	Tanggal

AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut. • AmazonSageMakerCanvasBedrockAccess	2 Februari 2024
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasFullAccess	24 Januari 2024
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasFullAccess	8 Desember 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasDataPrepFullAkses	7 Desember 2023

Fitur baru re: Invent 2023	Fitur-fitur baru berikut diperkenalkan di re:Invent 2023. • SageMaker Obrolan Kanvas untuk persiapan data • Editor Kode • Wadah pembelajaran mendalam untuk inferensi model besar • Terapkan model untuk inferensi waktu nyata • SageMaker Gambar Distribusi • Penyederhanaan orientasi domain • Amazon S3 Express Satu Zona • Evaluasi model pondasi (FMEval) • SageMakerHyperPod • Jupyterai • JupyterLab di Studio • SageMakerNotebook Lowongan • SageMaker Pipa • SageMakersmart menyaring • SageMakerStudio	30 November 2023

AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut di re:Invent 2023. • AmazonSageMakerFullAccess	30 November 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut di re:Invent 2023. • AmazonSageMakerCanvasAI ServicesAccess • AmazonSageMakerCanvasDataPrepFullAkses	29 November 2023
AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut di re:Invent 2023. • AmazonSageMakerClusterInstanceRolePolicy	29 November 2023
AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut. • AmazonSageMakerCanvasDataPrepFullAkses	26 Oktober 2023
AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut. • AmazonSageMakerCanvasDirectDeployAccess	6 Oktober 2023

AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasFullAccess • AmazonSageMakerCanvasAI ServicesAccess	29 September 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasFullAccess	29 Agustus 2023
AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut. • AmazonSageMakerPartnerServiceCatalogProductsApiGatewayServiceRolePolicy • AmazonSageMakerPartnerServiceCatalogProductsCloudFormationServiceRolePolicy • AmazonSageMakerPartnerServiceCatalogProductsLambdaServiceRolePolicy	1 Agustus 2023

AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasFullAccess	24 Juli 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerModelGovernanceUseAccess	17 Juli 2023
Daftar Isi Refactored	SageMaker Panduan Pengembang Daftar Isi difaktorkan ulang untuk lebih mencerminkan konten baru.	1 Juni 2023
SageMaker Jalur ECR	Jalur Registri Docker dan Kode Contoh diterbitkan.	25 Mei 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerGeospatialExecutionRole.	10 Mei 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasFullAccess	4 Mei 2023

AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut. • AmazonSageMakerModelRegistryFullAccess	12 April 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerCanvasFullAccess	24 Maret 2023
AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut. • AmazonSageMakerCanvasAI ServicesAccess	23 Maret 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerNotebooksServiceRolePolicy	9 Maret 2023
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut. • AmazonSageMakerNotebooksServiceRolePolicy	12 Januari 2023

Fitur baru re: Invent 2022	Fitur-fitur baru berikut diperkenalkan di re:Invent 2022. • SageMaker kemampuan geospasial • SageMaker Kartu Model • SageMaker Dasbor Model • SageMaker Manajer Peran • Kolaborasi dengan ruang bersama • Tes bayangan inferensi • Alur Kerja Berbasis Notebook • Widget persiapan data Wrangler Data • Langkah AutoML di Amazon SageMaker Model Building Pipelines • Ekstensi Git Klasik Studio	30 November 2022
AWSpembaruan kebijakan terkelola - Pembaruan kebijakan yang ada	SageMaker memperbarui kebijakan AWS terkelola berikut di re:Invent 2022. • AmazonSageMakerFullAccess • AmazonSageMakerFeatureStoreAccess • AmazonSageMakerCanvasFullAccess	30 November 2022

AWSpembaruan kebijakan terkelola - Kebijakan baru	SageMaker menambahkan kebijakan AWS terkelola baru berikut di re:Invent 2022. • AmazonSageMakerGeospatialFullAccess • AmazonSageMakerGeospatialExecutionRole • AmazonSageMakerModelGovernanceUseAccess	30 November 2022
Fitur baru re: Invent 2021	Fitur-fitur baru berikut diperkenalkan di re:Invent 2021. • SageMaker Kanvas • SageMaker Ground Truth Plus • SageMaker Rekomendasi Inferensi • SageMaker Titik Akhir Tanpa Server • SageMaker Studio Lab • SageMaker Studio Notebook dan Amazon EMR • SageMaker Kompiler Pelatihan	1 Desember 2021
Data deret waktu autopilot	Amazon SageMaker Autopilot menerima deret waktu sebagai input model. Untuk informasi selengkapnya, lihat data Amazon SageMaker Autopilot dan jenis masalah.	25 Oktober 2021

AWSkebijakan terkelola	Mulai melacak perubahan untuk kebijakan SageMaker terkelola.	10 Juni 2021
Fitur baru re: Invent 2020	Fitur-fitur baru berikut diperkenalkan di re:Invent 2020. • Pipa Bangunan SageMaker Model Amazon • Otomatiskan MLOP dengan Proyek SageMaker • SageMaker Manajer Tepi • SageMaker Klarifikasi • SageMaker Data Wrangler • SageMaker Toko Fitur • SageMaker Studio JumpStart • Daftarkan dan Terapkan Model dengan Registri Model • SageMaker Didistribusikan • Profiling Deep dengan SageMaker Debugger	1 Desember 2020
Notebook Studio	SageMaker Notebook Studio	28 April 2020

Fitur baru re:Invent 2019	Fitur baru berikut ini diperkenalkan di re:Invent 2019. • SageMaker Studio • SageMaker Studio Notebook (pratinjau) • SageMaker Eksperimen • SageMaker Autopilot • SageMaker Debugger • SageMaker Model Monitor	3 Desember 2019
Fitur baru re:Invent 2018	Fitur-fitur baru berikut diperkenalkan di re:Invent 2018. • Amazon SageMaker Ground Truth • Amazon Elastic Inference • SageMaker Sumber daya di AWS Marketplace • SageMaker Pipa Inferensi • SageMaker Neo • Cari SageMaker Eksperimen Amazon • Pembelajaran Penguatan • Kaitkan Repositori Git dengan SageMaker Instans Notebook • Algoritma Segmentasi Semantik • File Manifes Tertambah dalam Pekerjaan Pelatihan	28 November 2018

Mengkonfigurasi instance notebook	Gunakan skrip shell untuk mengonfigurasi instance notebook saat Anda membuat atau memulainya. Untuk informasi selengkapnya, lihat Menyesuaikan Instans Notebook.	1 Mei 2018
Dukungan Application Auto Scaling	Amazon SageMaker sekarang mendukung Application Auto Scaling untuk varian produksi. Untuk selengkapnya, lihat Penskalaan Model SageMaker Secara Otomatis	28 Februari 2018
TensorFlow 1.5 dan dukungan MXNet 1.0	Wadah Amazon SageMaker Deep Learning sekarang mendukung TensorFlow 1.5 dan Apache MXNet 1.0.	27 Februari 2018
BlazingText algoritma	Amazon SageMaker sekarang mendukung BlazingTextalgoritme.	18 Januari 2018
Enkripsi KMS	Amazon SageMaker sekarang mendukung enkripsi KMS untuk hosting instance dan melatih artefak model saat istirahat.	17 Januari 2018
CloudTrail dukungan	Amazon SageMaker sekarang mendukung logging dengan AWS CloudTrail.	11 Januari 2018
Algoritma Peramalan DeepAR	Amazon SageMaker sekarang mendukung algoritma DeepAR untuk peramalan deret waktu.	8 Januari 2018

SageMaker peluncuran	Amazon SageMaker diluncurkan di re:Invent 2017.	28 November 2017

Glosarium AWS

Lihat terminologi AWS terbaru di AWS glosarium dalam Referensi Glosarium AWS.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.